Cyberpunk 間 Architecture

A humble attempt to give visual expression to new models of reality by Samo Bertok

Diplomarbeit

Cyberpunk 間 Architecture

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Diplom-Ingenieurs unter der Leitung

O.Univ.Prof. Dipl.-Ing. William Alsop
Institut für Architektur und Entwerfen
Hochbau und Entwerfen 253/4

eingereicht an der Technischen Universität Wien
Fakultät für Architektur und Raumplanung von

Samuel Bertok
Matrikelnummer 0026052
Erlachgasse 75/14
1100 Wien

Wien, am _____ _____

Contents

Each of these Books has a unique size and proportion which has to do with the content it regards. Corresponding with the idea of punk and open source this Books should be easy to handle and not bind into a hard book cover. However for formal purposes of this thesis submission, these different formats have been converted to an A4 format with all the small issues it means for the layout.


CYBER SPACE

Intro

The *'punk'* (as in *'cyberpunk'*) is an attitude. It is a method to view, live and manipulate the social space or the factors which make it up.

Considering city space and cyberspace, both equally artificialities, dependent on real time changing factors from which they are both generated in real time. By manipulating these factors in the generation process, deforming the rendered space we can have a direct impact on the space we are living in or which surrounds us. It's not much different than to design a building or a square in the city, build it and leave it to the public to be used. Later it turns out to be used the intended way (as the architect intended) or it might as well end up abandoned, rotten and adopted by an graffiti artist. I do not think this is necessary the wrong way (there is no right or wrong in this case). Usurping the public space for one's needs is a fully understandable and agreeable position in this social dialogue. This is *'punk'*. Punk in this meaning is a way to approach a social issue and solve this spacial problem. This is what I want to do with the cyberspace (cyberpunk) and real space.

Cyberpunk 間 Architecture

A humble attempt to give visual expression to new models of reality by Samo Bertok

Book 1 –
Analysis
of History
and Theory
regarding
cyberspace-
architecture
and cyberpunk

The future has already happened, it just isn't very well distributed.
William Gibson

The key observation I made, which is sort of a premise for this diploma, has quite well been formulated by Greg Lynn, in his newest book *Form*. He says *'... we have been dreaming of a robotic future since we were first introduced to Robby in Forbidden Planet (1956). But, curiously, that future has arrived without anyone noticing. And more ironically, many young architects today dream a tomorrow that is already here.'*

The 21st century has shown already the potential of digitalized spaces and structures. I have set my goal for this diploma to discover the '*space*' in '*cyberspace*' and to transform it to a layer that could be displayed on top of the real space. Merging this two into the so called Augmented reality. Once William Gibson has noticed, that what is happening behind the video screen was far more interesting than what was happening in the space in front of it. In another words cyberspace pulls the user into the receding space of the electronic matrix in total withdrawal from the world. The cyber space has an impact on real space and vice versa.

I argue that the way we have been using internet (cyberspace) has dramatically changed our understanding of space and spacial relations as such. This has been a tremendous shift from traditional understanding of space like for example: 3D, perspective, movement, gravity, horizontal, vertical, genus loci... shift towards: MultiD, graphical representation of information and communication, hyperlink, attractors, space with equal directions, generic space...

In such new circumstances we have to rethink the role architecture plays in the world. We as architects simply have to include this variables into our design process, in search for a specific formulated example for an architectural design which would contain the idea of cyberspace-architecture.

The *'punk'* (as in *'cyberpunk'*) is an attitude. It is a method to view, live and manipulate the social space or the factors which make it up.

Considering city space and cyberspace, both equally artificialities, dependent on real time changing factors from which they are both generated in real time. By manipulating these factors in the generation process, deforming the rendered space we can have a direct impact on the space we are living in or which surrounds us. It's not much different than to design a building or a square in the city, build it and leave it to the public to be used. Later it turns out to be used the intended way (as the architect intended) or it might as well end up abandoned, rotten and adopted by an graffiti artist. I do not think this is necessary the wrong way (there is no right or wrong in this case). Usurping the public space for one's needs is a fully understandable and agreeable position in this social dialogue. This is *'punk'*. Punk in this meaning is a way to approach a social issue and solve this spacial problem. This is what I want to do with the cyberspace (cyberpunk) and real space.

The Information Age

as seen by Manuel Castells in 1996:
'We have entered a new age, the Information Age. Spatial transformation is a fundamental dimension of the overall process of structural change. We need a new theory of spatial forms and processes, adapted to the new social, technological, and spatial context where we live.'

I will try to focus on above mentioned theory of urbanism in the Information Age by proposing here some elements of urban infrastructure (such as technology, internet...) as a concrete manifestation of the *cyberspace-architecture* idea. All these in order to explore different facets of the relationship between the space (building, city, environment...), the body and the human state of mind as a subject of this discussion. I'll be speculating on how the possible worlds of artificial intelligence and cyberspace might affect the material reality of design, architectural and urban intuition as well as the conceptual models of space. My aim is not to foreclose discussions surrounding the '*cyborg city*' but to open up a series of dialogues in order to explore contemporary thinking, visions and resulting possibilities around these questions.

The main motivation for this work will be the discussion about how far we already invaded the territory of the fantastic imagination of science fiction authors. Is it possible that we didn't even notice under the disguise of everyday routine, that we entered the world of *Star Trek*, cyborgs, space ships, almighty AI's and cloud castles of the fiction known from books and movies? How much of the fiction of *Neuromancer* is happening today? How long will it take to fulfil the rest of the dark visions of cyberpunk? And what does that mean for us? ... The more interesting question for us architects, although seems to be: ... What can architecture take from cyberpunk theory and how could it affect the world (the space concept) we live in, its design and the way we perceive this space?

It is interesting to witness the reality (even if partly only virtual) which declines from *'the clean lines and smooth surfaces of Star Trek and 2001: A Space Odyssey in favour of the grimy, posturban realism of Batman, Neuromancer, and Bladerunner, in which computers do not simplify human issues but expose and even amplify the obvious faults in our systems of logic and social engineering.'* **1 Douglas Rushkoff, "Cyberia," 1994**

Maybe that is why we can no longer call it science fiction but more appropriate would be just simply call it the way things are actually happening.

The Information Age

as seen by Manuel Castells in 1996:
'We have entered a new age, the Information Age. Spatial transformation is a fundamental dimension of the overall process of structural change. We need a new theory of spatial forms and processes, adapted to the new social, technological, and spatial context where we live.'

I will try to focus on above mentioned theory of urbanism in the Information Age by proposing here some elements of urban infrastructure (such as technology, internet...) as a concrete manifestation of the *cyberspace-architecture* idea. All these in order to explore different facets of the relationship between the space (building, city, environment...), the body and the human state of mind as a subject of this discussion. I'll be speculating on how the possible worlds of artificial intelligence and cyberspace might affect the material reality of design, architectural and urban intuition as well as the conceptual models of space. My aim is not to foreclose discussions surrounding the '*cyborg city*' but to open up a series of dialogues in order to explore contemporary thinking, visions and resulting possibilities around these questions.

The main motivation for this work will be the discussion about how far we already invaded the territory of the fantastic imagination of science fiction authors. Is it possible that we didn't even notice under the disguise of everyday routine, that we entered the world of *Star Trek*, cyborgs, space ships, almighty AI's and cloud castles of the fiction known from books and movies? How much of the fiction of *Neuromancer* is happening today? How long will it take to fulfil the rest of the dark visions of cyberpunk? And what does that mean for us? ... The more interesting question for us architects, although seems to be: ... What can architecture take from cyberpunk theory and how could it affect the world (the space concept) we live in, its design and the way we perceive this space?

It is interesting to witness the reality (even if partly only virtual) which declines from '*the clean lines and smooth surfaces of Star Trek and 2001: A Space Odyssey in favour of the grimy, posturban realism of Batman, Neuromancer, and Bladerunner, in which computers do not simplify human issues but expose and even amplify the obvious faults in our systems of logic and social engineering.*' **1 Douglas Rushkoff, "Cyberia," 1994**

Maybe that is why we can no longer call it science fiction but more appropriate would be just simply call it the way things are actually happening.

Cyberpunk

or how the virtual reality has been described by science-fictional authors since the 80's

First of all let us define some key concepts before we can go deeper into understanding the aim of this discussion.

Cyberspace. A consensual hallucination experienced daily by millions of legitimate operators. ... a graphic representation of data abstracted from the banks of every computer in the human system. Unthinkable complexity. Lines of light ranged in the nonspace of the mind, clusters and constellations of data. Like city lights, receding. [2] William Gibson, "Neuromancer," London: Grafton, 1984

Cyberia. Is the place a businessperson goes when involved in a phone conversation, the place a shamanic warrior goes when travelling out of body, the place an *'acid house'* dancer goes when experiencing the bliss of a techno-acid trance. Cyberia is the place alluded to by the mystical teachings of every religion, the theoretical tangents of every science, and the wildest speculations of every imagination. Now, however, unlike any other time in history, Cyberia is thought to be within our reach. The technological strides of our postmodern culture, coupled with the rebirth of ancient spiritual ideas, have convinced a growing number of people that Cyberia is the dimensional plane in which humanity will soon find itself. [3] Douglas Rushkoff, "Cyberia," 1994

Cyberculture. A far-flung, loosely knit complex of sublegitimate, alternative, and oppositional subcultures whose common project is the subversive use of techno-commodities often framed by radical body politics ... cyberculture is divisible into several major territories: visionary technology, fringe science, avant-garde art, and pop culture. [4] Mark Dery, "Cyberculture," South Atlantic quarterly 91, 1992

Cyberpunk. Is a sub-genre of science fiction literature and film, with it's origins in 1980's, and associated with writers such as William Gibson (*Neuromancer,* 1984), Bruce Sterling and Neal Stephenson (though it is arguably prefigured in work by, among others, Philip K. Dick, William Burroughs, J. G. Ballard). In cinema, films like *Blade Runner* (1982) are seen as encapsulating the spirit and aesthetic of cyberpunk. Cyberpunk centres on the impacts of new technologies such as computers and virtual reality, and with propagating popular images of cyberspace, cyborgs, artificial life forms, and so on. Like many literary and filmic genres, it has split and recombined, with sub-sub-genres including *steam punk*, *biopunk*, and *cyberprep*, the latter offering a slick, clean, rosy view of the future to contrast cyberpunk's dirty, grim dystopias. [5] David Bell, "Cyberculture Theorists," Routledge critical thinkers, p. 4, 2007

History of Cyberpunk

'*Beat*' hero William Burroughs didn't start the cyberpunk movement in literature, but he foresaw it, most notably in his novel *Naked Lunch* (1959). Although written long before video games or the personal computer existed, Burroughs's works utilize a pre-cybernetic hallucinatory dimension called the Interzone, where machines mutate into creatures, and people can be controlled telepathically by *senders* who communicate messages via psychedelics introduced into the victims' bloodstreams.

Burroughs's description of the psychic interface prophesizes a virtual reality nightmare: *senders* gain '*control of physical movements, mental processes, emotional responses, and apparent sensory impressions by means of bioelectrical signals injected into the nervous system of the subject. ... The biocontrol apparatus [is] the prototype of one-way telepathic control.*' Once indoctrinated, the drug user becomes an unwilling agent for one of the Interzone's two main rivalling powers. The battle is fought entirely in the hallucinatory dimension, and involves **jacking in** (as William Gibson will later call it) through intelligent mutated typewriters.

'*Evolution did not come to a reverent halt with homo sapiens. An evolutionary step that involves biologic alterations is irreversible. We now must take such a step if we are to survive at all. And it had better be good. ... We have the technology to recreate a flawed artefact, and to produce improved and variegated models of the body designed for space conditions. I have predicted that the transition from time into space will involve biologic alteration. Such alterations are already manifest.*'

Cyberpunk stories have also been seen as fictional forecasts of the evolution of the Internet. The virtual world of what is now known as the Internet often appears under various names, including *cyberspace, the Wired, the Metaverse,* and *the Matrix.* **6 http://en.wikipedia.org/wiki/Cyberpunk**

Theory of Cyberpunk

Cyberpunk author Bruce Sterling summarized the cyberpunk ethos in *Cyberpunk in the Nineties* as follows:

'*Like punk music, cyberpunk is in some sense a return to roots. The cyberpunks are perhaps the first SF generation to grow up not only within the literary tradition of science fiction but in a truly science-fictional world. For them, the techniques of classical 'hard SF' – extrapolation, technological literacy – are not just literary tools but an aid to daily life. They are a means of understanding and highly valued.*' **7 Bruce Sterling, In "Mirrorshades: The Cyberpunk Anthology," Arbor House, preface, 1986**

'*Anything that can be done to a rat can be done to a human being. And we can do most anything to rats. This is a hard thing to think about, but it's the truth. It won't go away because we cover our eyes. This is cyberpunk.*'
8 Bruce Sterling, "Cyberpunk in the Nineties," http://cyber.eserver.org/sterling/interzon.txt

All of these science fiction concepts are based on the idea that human beings are basically programmable. They are as much determined from the outside as from the inside. They can't resist but reproduce the schemes inserted into their minds by both parents and society, as much as they are determined by the genetics or the DNA. They are helpless against the advertisement and screaming voices from the screens around them (which indeed take too much of the space one can perceive – I will explain this point a little bit later). As we read such books, we know this means that the character can't act otherwise. Those characters must behave absolutely in accordance with their programming, having no choice but following the instructions of their emotional templates.

What this implies is simple. We all act like programs to some extend. Facing certain situation we make a decision based on our experience and genetic background and we act. This is how we read the space we happen to find ourselves in. How we navigate. How we create and deform the space. What I'm interested in at this point are the rules that drives us to do this. To find and isolate them and to manipulate them in order to create a virtual-to-real interface.

Also like in Burroughs's prismatic style, the feeling of these books is more textural than structural. Like fantasy role-playing, computer games, or Nintendo adventures, these books are to be appreciated for the ride. Take the opening of Gibson and Sterling's novel, *The Difference Engine:*

'... Composite image, optically encoded by escort-craft of the trans-Channel airship Lord Brunel: aerial view of suburban Cherbourg, October 14, 1905.
A villa, a garden, a balcony.
Erase the balcony's wrought-iron curves, exposing a bath-chair and its occupant. Reflected sunset glints from the nickel-plate of the chair's wheel-spokes.
The occupant, owner of the villa, rests her arthritic hands upon fabric woven by a Jacquard loom.
These hands consist of tendons, tissue, jointed bone. Through quiet processes of time and information, threads within the human cells have woven themselves into a woman.
Her name is Sybil Gerard.' **9 William Gibson, Bruce Sterling, "The Difference Engine," Victor Gollancz Ltd., 1990**

Like the characters in *Fantastic Voyage,* we move through a multilayered fractal reality, enjoying the lens of a camera, the dexterity of a computer design program, the precision of a microscope, the information access of an eternal historical database, the intimacy of a shared consciousness, and finally the distance and objectivity of a narrative voice that can identify this entity by its name.

The way in which we move through the text says as much, if not more, about the cyberpunk worldview than does its particular post-sci-fi aesthetic. Writers like Gibson and Sterling hate to be called *cyberpunk* because they know their writing is not just an atmosphere or flavour. While this branch of fiction may have launched the cyberpunk milieu, it also embodies some of the principles of the current renaissance in its thematic implications.

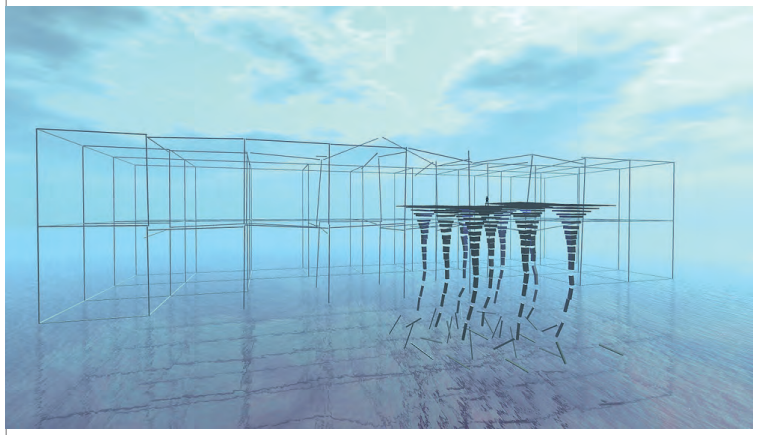
Even the above passage from *The Difference Engine* demonstrates a sense of holographic reality (this passage describes a *'picture'*, an image, made by an *'engine'*, a machine), where identity is defined by the consensual hallucination of a being's component parts. Similarly, like a DMT trip, a shamanic journey, or a hypertext computer program, reality in these books unfolds in a nonlinear fashion. A minor point may explode into the primary adventure at hand, or a character may appear, drop a clue or warning, and then vanish. Furthermore, these stories boldly contrast the old with the new, and the biological with the technical, reminding us that society does not progress in a smooth, curvilinear fashion. **10 Douglas Rushkoff, "Cyberia," 1994**

And isn't this much alike to how we experience our reality in the presence, past and future? Let's imagine the space we live in as a *'hollow room'* with definite borders (much in the sense of the quantum physics universe) and we observe this border, with for example our eyes, as a surface of infinite size and detail (remember

the TV screen-surface?). We have now the technology to zoom in and out this surface, to explore it's layers (we even do think in layers), to imagine and follow the '*history*' of it's creation (similar to the history in every web browser) and the networks and capacity to exchange and store this information. '*We live ... in the encyclopaedia of historical experience,...*' (Castells, 2002). Let's take for example '*hyperlinks*' or '*windowing*'. '*Our minds are full of hyperlinks...*' (Bell, 2007). The point is that we are actually getting used to it. We use this point of view, those networks and this way of looking at things every day, but we still aren't quite aware of the change in our lives that comes with it. Even though we use the TV like this, or the newspapers or we search up the internet exactly this way. We have established '*windowing*' as the way of working with computers, clicking between screens – windows, as between files and applications. It allows us to be in various contexts at the same time, in real time. This experience has migrated out of the computers and subsequently turned our real life yet another '*window*' (thanks Bill :-)). As we moved from the world of representation to simulation, the '*...cycling through has become the way we think about life itself*' (Turkle, 1995). The world around us doesn't flow in a linear fashion; at least we don't look at it that way anymore. As much as we for example got used to *'surf the web'* via hypertext connections (*'the hypertext is inside us'* – Castells, 2001), windows and nonlinear histories, we have to admit that cyberpunk visions have caught up with our reality.

Theory of cyberspace

or how we see virtual reality in the 21st century



From fireplace to digital hub

If we look back in the history of architecture and our spacial behaviour, in the beginning the turning point of the household was the fireplace. In fact the function of the fireplace as a source of heat and food was central and the everyday life was organized around it. In more abstract way one could argue that the social life of the community was dependent on the fireplace.

Observing the continuing historical development focusing in particular on the understanding of space, functionality in space and acting in space, there is a clearly visible tendency from the fireplace through workplace, cooking place, towards the modern media as telegraph, telephone, TV, computer... internet.

I argue that at the centre of the living space of the 21st century definitely is the information in the virtual space, know as internet or mobile communication.

Actual trends

According to Bolter and Grusin (2000) the '*cyberspace*' is very much a part of our contemporary world and ... it is constituted through a series of remediations. As a digital network, cyberspace remediates the electric communications networks of the past 150 years, the telegraph and the telephone; as virtual reality it remediates the visual space of painting, film, and television; and as social space, it remediates such historical places as cities and parks and such '*nonplaces*' as theme parks and shopping malls. Like other contemporary telemediated spaces, cyberspace refashions and extends earlier media, which are themselves embedded in material and social environments. **11 Stephen Graham, "The Cybercities reader," Urban Reader Series, p. 11, 2004**

William Gibson (2001) describes Japan, as the reality nearest to his books as following: *'Modern Japan simply was cyberpunk. The Japanese themselves knew it and delighted in it. I remember my first glimpse of Shibuya, when one of the young Tokyo journalists who had taken me there, his face drenched with the light of a thousand media-suns (all that towering, animated crawl of commercial information) said, "You see? You see? It is Blade Runner town." And it was. It so evidently was.'*

I will try to use the looking at current trends as a method to implicate the future common use of technology and how we will '*re-think*' the world. This is all about '*pattern recognition*'. *'Homo sapiens is about pattern recognition... both a gift and a trap.'* **12 William Gibson, "Pattern Recognition," G. P. Putnam's Sons, p. 22–23, 2003**

Pattern recognition in this case means to label the input data received by a computer to make it accessible by the next program which determines the correct output value for new data instances. If we look at us as people working with the same kind of programs in our minds, scanning the unknown space around us searching for patterns we do understand (which we can label) and we can work with, all we do is a pattern recognition. The question is how do new patterns evolve and if we are able to label them.

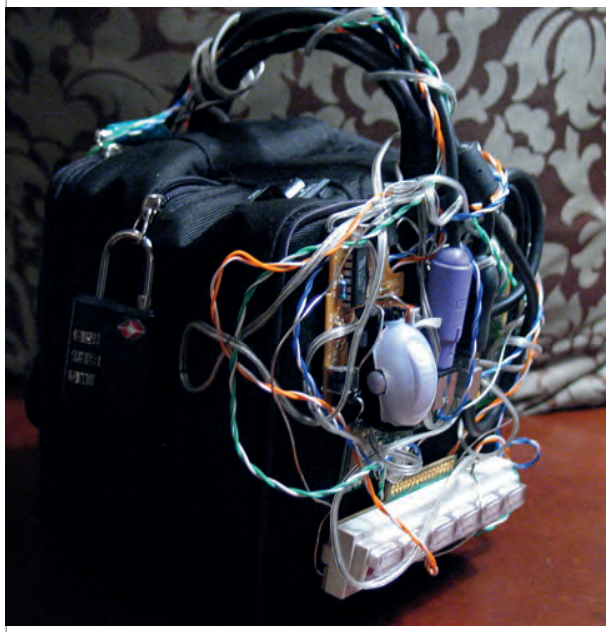
The cyberian paradigm finds its way to our unsuspecting minds through new kind of arts and entertainment that rely less on structure and linear progression than on textural experience and moment-to-moment awareness. Role-playing games, for example, have no beginning or end, but instead celebrate the inventiveness of their players, who wind their way through complex fantasies together, testing strategies that they may later use in their own lives, which have in turn begun to resemble the wild adventures of their game characters. [13] Douglas Rushkoff, "Cyberia," 1994

And once we take virtuality seriously as a way of life, we need a new language for talking about the simplest things. [14] Sherry Turkle, "Who am we?," wired, 4.01, 1996

The new language mentioned by Turkle is a paradigmatic issue. If we analyse it deeper we will have to accept the position that virtuality is a spacial issue, in which connections, communication, freedom and self-expression are the rudimentary components. Similar to architectural components. This I think is a area where architecture comes into play. Architects do have the best position to formulate the key principles and rules for (let's say) 'the interface' of the intersection between the digital and analog or between the virtual and real or the cyberspace and real space.

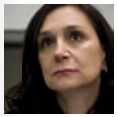
Technology

or where are the
technical boundaries
in cyberspace



John Barlow (the lyricist for the Grateful Dead and cofounder of the Electronic Frontiers Foundation) says on *'the kind of technology'*: *'On the most rudimentary level there is simply terror of feeling like an immigrant in a place where your children are natives, where you're always going to be behind the 8-ball because they can develop the technology faster than you can learn it. It's what I call 'the learning curve of Sisyphus'. And the only people who are going to be comfortable with that are people who don't mind confusion and ambiguity. I look at confusing circumstances as an opportunity, but not everybody feels that way. That's not the standard neurotic response. We've got a culture that's based on the ability of people to control everything. Once you start to embrace confusion as a way of life, concomitant with that is the assumption that you really don't control anything. At best it's a matter of surfing the white-water.'* **15 Douglas Rushkoff, "Cyberia," introduction, 1994**

Children's point of view



Granting a psychology to computers can mean that objects in the category *'machine'*, like objects in the categories *'people'* and *'pets'*, are fitting partners for dialog and relationship. Although children increasingly regard computers as mere machines, they are also increasingly likely to attribute qualities to them that undermine the machine/person distinction. **16 Sherry Turkle, "Who am we?," wired, 4.01, 1996**

We witness a revolution in socializing with machines. It goes probably dangerously unnoticed that a digital device is used as commonly as *'talking'*. Every person has at least a smartphone or a notebook or some other personal digital device at hand, often regarded as part of one´s body or a dear possession, bearing life dependant personal data inside or at least accessible through it. Even there is the idea that the space surrounding us could act like such a portal to access this data. Be it a connection spot, node, wall, building or a city. How would we then see such a *'building'* and how would we use it or communicate with it?

We are not experiencing some wholesale, discrete, break with the urban past ushered by the *'impacts'* of new technology. Rather, we are experiencing a complex and infinitely diverse range of transformations where new and old practices and media technologies become mutually linked and fused in an ongoing blizzard of change. **17 Stephen Graham, "The Cybercities reader," Urban Reader Series, p. 11, 2004**

The digital us

In a sense, ICTs have now '*produced the ordinary*' to be woven so completely into the fabric of everyday urban life that they become more and more ignored (Amin and Thrift, 2002). This now leads to new means of social control even better than Bentham's Panopticon. I'm speaking of people becoming so called *'digital subjects'*, *'files'* monitored, tracked, controlled, commodified and stored. In this sense we watch ourselves (similar to the idea of Bentham) as we participate within the webs of infinite of digital connections, always leaving our digital tracks as objects of further analysis.

Or in the words of Johnny Mnemonic (the main character of Gibson's novel of the same name): *'we are an information economy. They teach you that in school. What they don't tell you is that it's impossible to move, to live, to operate at any level without leaving traces, bits, seemingly meaningless fragments of personal information. Fragments that can be retrieved, amplified...'* **18 Wiliam Gibson, "Johnny Mnemonic," Omni magazine, 1981**

One can buy any information. That means anyone can pick up that trace. Your *'digital trace'*. It's here (there) for all of us accessible.

Denis Cosgrove, a cultural geographer, suggests that the urban world networked by Bill Gates' technologies, strung out on the wire, is not disconnected, abstract, and inhuman; it is bound in places and times of our lives, into human existences that are as connected, sensuous and personal as they have ever been. **19 Stephen Graham, "The Cybercities reader," Urban Reader Series, p. 15, 2004**

The informational networks are, by now, so very much woven into the very core of our lives, that for example we have to consider that, *'money is increasingly experienced as informational patterns stored in computer banks rather than as the presence of cash'*. **20 Katherine Hayles, "How we became Posthuman," The University of Chicago press, p. 27, 1999**

And I don't just mean the current situation of the financial crisis, much more I'm trying to point at the changed principles, which came so smooth from actual physical value (like trade goods or gold or so on) to virtual value stored digitally. Now money has in a sense become just the mere information about possession and the right to possess.

This information age has never been (only) a technological matter. It has always been a matter of social transformation, a process of social change in which technology is an element that is inseparable from social, economic, cultural and political trends. **21 Manuel Castells with Catterall, "Cyberculture Theorists," Routledge critical thinkers, by David Bell, p. 59, 2007**

What will be?

Ha ha. This is how science fiction was imagined in 2008

Maros Krivy, November 30th 2008

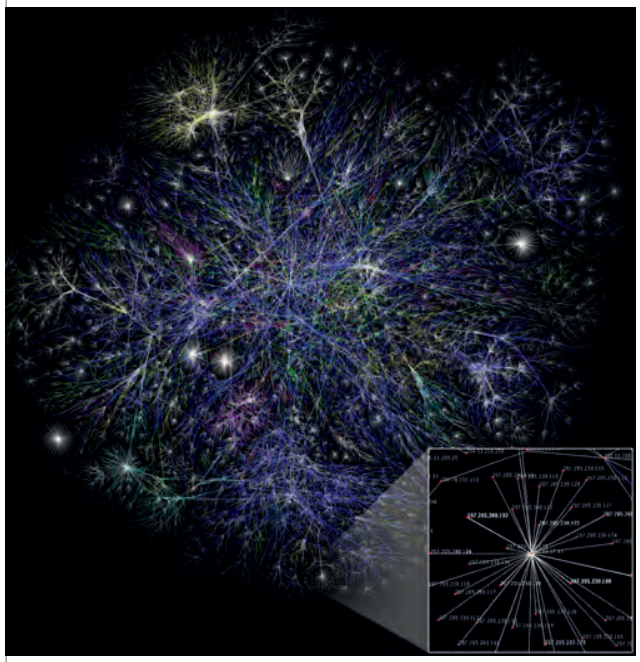
The future of the cybercities is a most uncertain and complex prediction to make, as the speed of change and frequency of new technologies emerging, is growing unmatchable with anything we have seen in the past. To think of the future shape of cybercities we have to consider a vast field of inputs. Such as architecture, urbanism, science fiction, future studies, geography, urban planning, engineering, art, business studies, economics, philosophy, anthropology, sociology, cultural studies, journalism, programming and theology. Given the near impossibility of predicting future directions of social and technological change, futuristic speculations about technology and cities tend to say as much about our own times (and lives) as it does about the nature of the future worlds. We are and we will remain part of a digital whole. As Douglas Rushkoff point out in his latest book *Program or be Programmed: '... we are living in an age when thinking itself in no longer a personal activity but a collective one. We are immersed in media and swimming in the ideas of other people all the time.'* **22 Douglas Rushkoff, "Program or be Programmed," Soft Skull Press, p. 124, 2011**

A French analyst Paul Virilio follows a theory that our future is the model of the '*online disabled citizen*'. He goes so far as to say that we will be physically unable to move and will be connected into sets of prosthetic devices which will connect global universes of electronic space straight into users' senses and bodies. Much the same as the human body is seen as a slave of '*the machines*' in the movie *The Matrix* (1999).

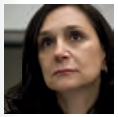
The figure of the cyborg, as represented in science fiction cinema, is not an automaton or robot but a sophisticated creation that seems to simultaneously extend but also threaten our understanding of what it means to be human. **23 Mathew Gandy, "Cyborg Urbanization," International Journal of Urban and Regional Research, 2005**

Internet

or where is the *space* in cyberspace, how does it function and what is affecting it? How do we form it and how does it form us?



The actor-network theory assumes that there are multiple, heterogeneous networks. According to this approach, electronic domains and cybercities, therefore need to be considered as fragmented, divided, and contested multiplicities of heterogeneous infrastructures and actor-networks. For example, there are tens of thousands of specialised corporate networks and intranets. The Internet provides the basis for countless Usenet groups, Listservers, corporate advertising sites, specialised web sites, MUDs, corporate intranets, virtual communities and increasingly sophisticated flows of media and video. PSTNs, and the many competing telecoms infrastructures, support global systems of private ATM networks, credit card and electronic clearing systems, as well as blossoming applications for CCTV, telehealth, teleshopping and telebanking, global logistics, remote monitoring, back office and telesales flows, as well as data and telephony flows. And specialised systems for satellite, broadband, cable and broadcasting networks support burgeoning arrays of television flows. [24] **Stephen Graham, "The Cybercities reader," Urban Reader Series, p. 69, 2004**



The Internet links millions of people in new spaces that are changing the way we think and the way we form our communities. Which means, that we are moving from *'a modernist culture of calculation toward a postmodernist culture of simulation.'* [25] **Sherry Turkle, "Who am we?," wired, 4.01, 1996**

The internet (or Cyberspace) has become *'the domain of virtual collectivism'*. Building communities just like in real-world. It is being constituted as a resultant of millions of vectors representing the diverse and more than often conflicting interests of the actors involved inside or outside, of humans or artificial intelligences linked together through this giant computer network.

Here are some of the aspects of Internet I would like to point out:

The Digital Divide is still a notable factor even if one wouldn't think so. If we look at the coverage, accessibility, capacity, literacy, speed of connection, etc., we discover a digital separation between the rich and poor almost mapping the distinction of the society in real space. The same goes for public and private space as for open and closed access to parts of internet.

The internet has had a unparalleled impact on the **process of thinking and orientation.** We witness substantial influence of internet on our spacial understanding and navigation in space including several digital inventions like *hyperlinks* and *windowing* to name just a few.

The internet has become a new, **virtual world**, we *'visit'* everyday not even realising the journey anymore. Using it like a tool rather than a space to solve our instant tasks be it communicating, reading the news, watching movies, ordering food or buying clothes. **Bookmarking** web pages we notoriously visit makes it even easier to feel like in familiar environment. **Personalised space**. Exactly as we would tune up the space of a room we live in, to comfort us and suit our needs. This becomes now so apparent that it renders the term *'at home'* new. The question of *'home'* becomes really important.

How do we define *'home'* in the Information Age? Is it the most familiar place to us? Where we are most present? The one we visit most and spend the most time in? The most influenced by us? The most influencing us? The one where we have the most friends? To be at home in internet? Because with these in mind it might as well be *Facebook*.

'... it is a way now, approx, of being at home. The Forum has become one of the most consistent places, in her life, like a familiar café that exists somehow outside geography and beyond time zones.' **26 William Gibson, "Pattern Recognition," G. P. Putnam's Sons, 2003**

The issue of shortening of the time to reach a distant place or to get an information across vast distance has been brought up across the whole history of human civilisation. Every time a new technology has been implemented, it resulted into the phenomenon called **reduction of distances.** From wheel and horseback riding, through sailing, steam power, railroads, combustion, automobile, airplane to telegraph, telephone and email.

Travel? Well to be honest we do not travel in the same way or sense as we were used to 50 years ago. *'Reality is increasingly immaterial, and our modes of travel become static terminal transmissions.'* **27 M. Christine Boyer, "CyberCities," Princeton Architectural Press, p. 11, 1996**

Marcos Novak describes travelling through hypertext in following manner: *'Every paragraph an idea, every idea an image, every image an index, indices strung together along dimensions of my choosing, and I travel through them, sometimes with them, sometimes across them. I produce new sense, nonsense, and nuisance by combination and variation, and I follow the scent of a quality through sand dunes of information. Hints of an attribute attach themselves to my sensors and guide me past irrelevant, into the company of the important; or I choose to browse the unfamiliar and tumble through volumes and volumes of knowledge still in making.'* **28 Marcos Novak, "Liquid architectures in cyberspace," In Cyberspace: First Steps, MIT Press, p. 230, 1991**

In this way a **hyperlink** is a virtual connection based on meaning and common sense reducing the time and space between two places in the virtual space to minimum. This is how the virtual space is *'woven'* and build up. It's fluidity comes from hyperlinks.

The **email** has generally replaced our written communication for good. Then came the **chat** and grew. Now we have Twitter and Facebook. The written communication mutated from asynchron to synchron which means we do response to incoming information immediately or constantly check if the information arrived so we can answer back. Everybody wants to be able to acquire the most fresh information in real time and be able to reply instantly. This sets high standards for designing the virtual environment for us simple because it is hugely complicated to filter out the *'right'* information for a specific person. This is what the biggest companies like Google and Facebook are doing business with and throwing immense amount of resources into. If you ask yourself what the value in social networks is, the simplest way to find out is to look who's paying for it.

In the early 1970s, the face-to-face role-playing game Dungeons and Dragons swept the game culture. The term *'dungeon'* persisted in the high-tech culture to connote a virtual place. So when virtual spaces were created that many computer users could share and collaborate within, they were deemed Multi-User Dungeons or **MUDs**, a new kind of social virtual reality. **29 Sherry Turckle, "Who am we?," wired, 4.01, 1996**

As players who participate on the MUD, become authors not only of text but of themselves, constructing new selves through social interaction.

In the MUDs, virtual characters converse with each other, exchange gestures, express emotions, win and lose virtual money, and rise and fall in social status. A virtual character can also die. Some die of *'natural'* causes (a player decides to close them down), or they can have their virtual lives snuffed out.

The anonymity of MUDs gives people the chance to express multiple and often unexplored aspects of the self, to play with their identity and to try out new ones. MUDs make possible the creation of an identity so fluid and multiple that it strains the limits of the notion.

On a MUD one actually gets to build character and environment and then to live within the toy situation. A MUD can become a context for discovering who one is and wishes to be. In this way, the games are laboratories for the construction of identity.

Gender-swapping on a MUD is not a small part of the game action. By some estimates, '*Habitat*', (a Japanese MUD), has 1.5 million users and is operated for profit. Among the registered members of Habitat, there is a ratio of four real-life men to each real-life woman. But inside the MUD the ratio is only three male characters to one female character. In other words, a significant number of players, many tens of thousands of them, are virtually cross-dressing. [30] Sherry Turkle, "Who am we?," wired, 4.01, 1996

Lyrics of a Jethro Tull song *Dot Com* describing the internet as a '*lived space*':

It's a wide world out there
So much wider than imagined
I can't quite put my finger on the pulse
Of your heart softly beating
Just beneath the raw silk sheen
That reflects the tints of autumn from the hills.

So punch my name.
And in case you wonder – I'll be yours – yours, dot com.
Executive accommodation
Bland but nonetheless appealing
Waiters discretely at your back and call
Place the tall sun-down potion
Lightly by your velvet elbow
While you compose a message on the wall.
So punch my name.
And in case you wonder – I'll be yours – yours, dot com.

With your handmade leather valise
Packed and ready, ready waiting
Showered and dressed down lightly for the heat
Give a clue; leave a kind word
Hint as to a destination
A domain where our cyber souls might meet.
So punch my name.
And in case you wonder – I'll be yours – yours, dot com.

Globalization

Globalization is highly selective.

Castells, 2000

The global cities are the network cities for the network society. Given the disembedding qualities of the network society, given that time-space compression '*frees*' people from constraints of geography, and given all the futurology about new ways of '*wired living*', why do cities still exist? Even more the fact is that the world is becoming more urbanised. [33] David Bell, "Cyberculture Theorists," Routledge critical thinkers, p. 71, 2007

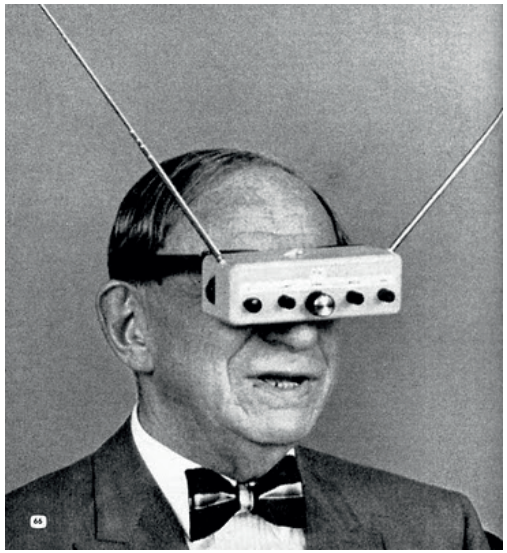
USA

Townsend (2004) says that more optic fibre underlines the island of Manhattan than is treaded across the whole of Africa. And, in 2003, the Internet geographer Matthew Zook showed that only 5 metropolitan regions – New York, Los Angeles, San Francisco, London and Washington DC – accounted for 17.7% of the world's total Internet domains. These are the familiar .coms, .nets, .orgs etc. that are widely portrayed as being '*placeless*' and '*without a geography*'. [34] Stephen Graham, "The Cybercities reader," Urban Reader Series, p. 12, 2004

China

Experts estimate that in Beijing alone, 1 bill. Sq. feet of offices, shops and apartments will be added to the skyline by 2008 (the equivalent of 3 Manhattans) totalling $160 bill. worth of construction. [35] William Mellor and Allen Cheng, "Beijing Home to $160b Worth of Construction," Vancouver Sun, march 4, 2006

China's spectacular urban revolution now consumes one-half of the world's annual production of concrete and one-third of its steel output, pushing up world prices to the point that long-closed iron-ore mines throughout the world are now being reopened. [36] Deyan Sudjic, "The City That Ate the World," Guardian, October 16, 2005



Augmented reality

Augmented reality (according to Wikipedia) is an environment that includes both virtual reality and real-world elements. For instance, an AR user might wear translucent goggles; through these, he could see the real world, as well as computer generated images projected on top of that world. Ronald Azuma (Research Leader at Nokia Research Centre) defines an augmented reality system as one that

∴ *combines real and virtual*
∴ *is interactive in real-time*
∴ *is registered in three dimensions*

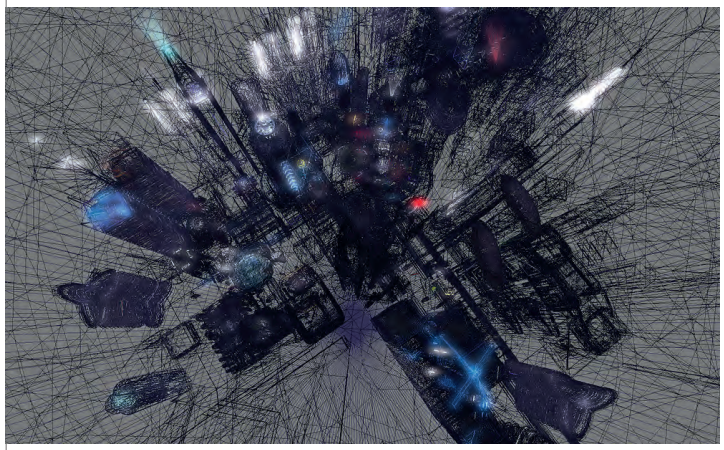
The potential which the augmented reality could have for the, not so far, future is suggested in one of the latest books from William Gibson *The Spook Country*. Although, he uses the term *'locative art'* here: *'Would it all be like this, in Alberto's new world of locative? Would it mean that the untagged, unscripted world would gradually fill with virtual things, as beautiful or ugly or banal as anything one encountered on the web already?'* [31] **William Gibson, "Spook Country," Penguin Group, p. 142, 2008**

As recently stated on The Wired magazine Google is currently advertising for a designer and engineer responsible for augmented-reality mapping. The designer position requires the ability to *'integrate mobile platforms, augmented reality mapping, geo-location, and real-time interaction.'* And according to the article it is rumoured that they are already busy working on a pair of head-up display (HUD) glasses to be released later this year. The technology used could include either a small, discrete LCD screen that sits at the bottom of the right-hand lens, or the augmented reality data will be displayed directly in the lenses of the glasses. The more futuristic and challenging possibility already in discussion is a virtual retinal display (VRD) which could use lasers to draw images directly onto the retina of the eye. [32] **Roberto Baldwin, "Google Hiring Augmented-Reality Experts Amid Rumors of HUD Glasses," wired.com, February 27, 2012**

Augmented reality could be the very next step in how do we define space and how do we use it. We can place virtual objects (like virtual copies of real objects or utterly non-real objects) into real space and actually use them for whatever purpose we want most arguably for purposes they were not meant to be used as the were real. But the most interesting question would be how do we design the real space which should act as the best environment for these kind of *'interventions'*. I argue the architecture should come up with an answer to that.

Architecture

or how the cyberspace influences the actual trends in architecture



Karl Popper proposed the idea of 3 *'worlds'* based on the classification of how we *'understand'* the reality. World 1, of physical objects, events and biological entities; World 2, of mental events and objects; World 3, of products of human mind, or abstract objects. We also tend to use the concept of World 3 as we talk about cyberspace and cyberculture, as an emblematic abstract '*mind space*'.

Michael Benedikt argues that architecture is also part of World 3, the world of patterns of communication – architecture is communication in build form. **37 David Bell, "Cyberculture Theorists," Routledge critical thinkers, p. 19, 2007**

And he also thinks that '*... building cyberspace will require cyberspace architects, designers of electronic edifices, the liquid architecture of information flows.*'

Major crossovers have already occurred between cyberpunk and critical urban social studies, by writers like Mike Davis, who expresses a major debt to cyberpunk imaginings of near-future urban conditions, and Roger Burrows who even argues that: *'... I think that one gets a clearer analytical understanding of contemporary urban processes from reading of Gibson or Stephenson than one does from a reading of Sassen or Castells.'* **38 Roger Burrows In "The Cybercities Reader," ed. Stephen Graham, Routledge, p. 389, 2004**

According to David Bell the cyberculture studies include among others:

- ∴ *Studies of changing work patterns.*
- ∴ *Studies of aesthetics of new technologies.*
- ∴ *Cross-disciplinary futurology that predicts way of living yet to come.*
- ∴ *Cultural studies approaches to understanding the material, symbolic and experimental dimensions of cyberspace, to cybercultural forms, practices, politics and identities, and to cybercultural production and consumption.* **39 David Bell, "Cyberculture Theorists," Routledge critical thinkers, p. 9, 2007**

All of these authors suggest *why* architecture...

Computers are already an established part of our lives. We use them from listening to music, watching movies and storing our memories to working in the office and most of all for communication of various kinds. Moreover we have to consider that, not only we may be transformed into data and lodged in databases (as for example companies, monitoring our shopping and working practices and habits, do), but equally do have *'intimate'* relationships with digital devices (such as computers, ipods, mobiles, pads...) that become not only part of our lives but even more part of our bodies. Leading us to the questions of cyborgs and posthumans, including territories

such as cyborg studies, feminist cyberstudies, posthumanism or postbiology, body vs. mind… all of which are simply issues of identity. Identity in times of cyberspace or better to say *'identity in cyberspace'*. And by this identity we have to understand that we are talking about *'the constructed identity'* more than a natural one. We have to acknowledge the fact that both the cyberspace and city space are artifices, intentionally constructed with the help and mark of computer algorithms (or building restriction, light issues, owner relations, etc. in case of city space) which have to function exactly in the way their programs or design codes allow (which means how the creators design them obeying the laws of computer algorithms). Every computer program is design by people who have ideas in mind which naturally come from their experience. Experience with the real world and real space which subsequently means that the laws for the *'code'* to obey are reality dependant. This is why I argue that the computer laws (same as the building restrictions) are determined from the physical reality. In the end we are still only talking about space.

This is a very important point to consider. Both cyberspace and city space end up in the hands of architects to be *'constructed'* rather than anyone else, simply because they happen to be most trained in constructing space and spacial relations. This I regard as the biggest challenge and biggest opportunity equally. We as architects have a responsibility towards the user of our creation in order to give him the full possibility of using it.

But before we can speculate on how possible worlds of AI and cyberspace might affect the material reality of design, conceptual models of space and architectural or urban intuitions, or before we talk about identity in cyberspace, we have to understand, how we might think and build cyberspace and also need to understand how we have developed ways of acting in the world around us (phenomenological). Benedikt and Bell suggest seven key principles of cyberspace design and build:

∴ *Principle of Exclusion* – two things cannot be in the same place at the same time.
∴ *Principle of Maximal Exclusion* – rules over how 'big' and how 'dense' can cyberspace become.
∴ *Principle of Indifference* – cyberspace has an existence independent of users.
∴ *Principle of Scale* – relationship between the amount of information in space and the amount of space in space.
∴ *Principle of Transit* – travel as experience is important as is navigation.
∴ *Principle of Personal Visibility* – users in cyberspace should be seen, at some minimal level, by other users… 'cyberspace must have a street life'.

∴ *Principle of Commonality* – the needs to be an objective, shared social 'reality' in cyberspace, so that people see and hear the same things (at least partially). **40 David Bell, "Cyberculture Theorists," Routledge critical thinkers, p. 22, 2007**

I would suggest some for myself.

∴ *Principle of Open-Source* – the spirit of informationalism captured in the famous hacker slogan *'information wants to be free'*.
∴ *Principle of Difference* – if I see one more orthogonal office or boring façade, I'm going to log out… wouldn't you?

The architect Marcos Novak suggests in his work (*Augmented space and transarchitecture, 1994*) redefining the urban field by challenging the three rudimentary assumptions of urban studies. In first place the space is three-dimensional and shared between the subjects acting in it. Second, that space is either solid or void. And finally that one can only be at one place at one time. He thinks that, with distanciated contact, mediated presence and software agents acting in our names, these conditions no longer hold. The assumption of autonomous objects having localized presence with enduring substance, as been taught for ages in the architectural canon dominated by metaphysics of presence is past. At this moment the least secure thing in Architecture of today, is the one which has been thought to be the most granted for ages and that is the question of *'presence'*. *'The development of temporally simultaneous activities in spatially discontinuous locations calls for a different imagination of presence and the shape of urban time-space'*. **41 Mike Crang In "The Cybercities Reader," ed. Stephen Graham, Routledge, p. 130, 2004**

Is it possible to design real space and architecture using these rules? And who for?

Who is the person we are talking about these days? Who are the people living and working in the cyberspace? To answer this question we have to look at the work and life patterns of the *'networkers'* (Castells calls them also *'self programmable labour'*). First of all they work and live in the cyberspace, which means they adapted the computer as an everyday tool. They love the city. In their everyday *'stereotype'* they physically move while they constantly keep their network connection, to everything they do, alive. This is important because they connectivity and ability to react instantly to whatever new informational status they face is *'life dependant'*. In their job (not necessarily in their private life) they are *'footloose'* – they do not feel to belong to a particular place, have no sense of local or national identity, nor do they feel a loyalty to a firm (which can change if they are the firm). Their skills are highly transferable

so they can adapt quickly and smoothly to the tasks given to them (ever changing networks, different projects, multitasking). They don't wear white collars and ties any more; they are casual in attire and casual in their attitude, even though they can be dead serious about their work. Young and restless. They are the *'metropolitan nation'* living the cosmopolitan lifestyles on a high consumption basis. When they are not working in an office or having a business meeting in a conference room, they can be found in habitats like boutique hotels, or loft apartments, airport VIP lounges, exclusive and expensive restaurants or on high-culture events like operas and theatre. Castells describes their lifestyle, following: *'... the regular use of spa installations (even when travelling), and the practice of jogging; the mandatory diet of grilled salmon and green salad, with udon and sashimi providing a Japanese functional equivalent; the 'pale chamois' wall colour intended to create the cosy atmosphere of the inner space; the ubiquitous laptop computer; the combination of business suit and sportswear; the unisex dressing style; and so on.'* **42 David Bell, "Cyberculture Theorists," Routledge critical thinkers, p. 66, 2007**

Now as we are currently observing this group has already made their mark on the cities they live in or travel through. But the question is if the generic architecture style, which is serially reproduced to make these *'nomads'* feel at home everywhere they land, is the right direction to go.

The modern home, for example, has become a complex exoskeleton for the human body with its provision of water, warmth, light and other essential needs. The home can be conceived as *'prosthesis and prophylactic'* in which modernist distinctions between nature and culture, and between the organic and the inorganic, become blurred (Vidler, 1990). And beyond the boundaries of the home itself we find a vast interlinked system of networks, pipes and wires that enable the modern city to function. These interstitial spaces of connectivity within individual buildings extend through urban space to produce a multi-layered structure of extraordinary complexity and utility.
43 Mathew Gandy, "Cyborg Urbanization," International Journal of Urban and Regional Research, 2005

The distinction between the architecture and furniture and space is blurring constantly since 20 years. We are now left with a space free of form or other restrictions with let's say *'floating information-technology systems'* such as wireless networks or mobile networks. This is the urban realm we learned to exist with in a spatial sense and we are becoming much better to navigate through. We have to consider that the aim of the design of this environment is the higher accessibility, speed and volume of information we can acquire as well as easier and more effective way to filter it and navigate through it (for example using these networks for the navigation itself).
The emphasis on the city as a self-contained body or machine has

been challenged by a hybridized conception of space as a system of technological devices that enhances human productive and imaginative capabilities. **44 Mathew Gandy, "Cyborg Urbanization," International Journal of Urban and Regional Research, 2005**

Castells sees a major change in the recent trend in architecture, which signals it's transformation from an intervention of *'the space of places'* to an intervention of *'the space of flows'* (which he assumes to be the dominant space of the Information Age), by acting on space dedicated to museums, convention centres and transportation nodes. Here the architecture acts as mean of transformation of spaces of great cultural and communicational potential into forms of cultural expression and meaningful exchange of information. Or in other words, let us value architecture as it's ability to generate a discussion around itself, for at that point it has been inserted into the endless circulation of cultural signs.

'The most spectacular example is Frank Gehry's Guggenheim Museum in Bilbao that symbolized the will of life of a city immersed in a serious economic crisis and dramatic political conflict. Calatrava's bridges (Seville, Bilbao), telecommunication towers (Barcelona), airports (Bilbao) or Convention Centres (Valencia) mark the space of flows with sculpted engineering. Bofill's Barcelona airport, Moneo's AVE railway station in Madrid and Kursaal Convention Center in San Sebastian, Meier's Modern Art Museum in Barcelona, or Koolhaas' Lille Grand Palais (or CCTV in Beijing), are all examples of these new cathedrals of information age, where the pilgrims gather to search for the meaning of their wandering.' **45 Stephen Graham, "The Cybercities reader," Urban Reader Series, p. 91, 2004**

Critics however, would point at the disconnection of these symbolic buildings and the city as whole. They show us that the lack of integration of this architecture of the space of flows into the public space would be an example of spatial meaninglessness and nonsubstantial symbolic punctuation of the urban structure.

This is quite a different perspective on the role of *'chaos'*, for example, as from that associated with the modernist city. It is perceived as a field of evolutionary and radical experimentation. Chaos is no longer seen as an anomaly to the urban experience (where everything must have its given order) to cause problems or be excluded from analysis but a rich vein of social and spatial interaction as such. Through which we may perceive signs of alternative or until now overlooked urban forms of bottom-up hierarchies like street art as a form of a free and public self expression. Chaos may also be characterized as a more sophisticated, resilient and adaptable form of order, as Rem Koolhaas has suggested (Koolhaas *et al.*, 2001; Koolhaas, 2002).

Focusing on wireless infrastructures William Mitchell explores the extraordinary technological potentialities of the cyber city and the increasing interaction of digital code with virtually every sphere of human activity (Mitchell, 2003). Possibilities are almost endless. The use of cyber technologies in the city space are vast. Partly we can see it coming in form of smartphones applications useful to navigate through the city or finding the desired restaurant or bank or underground gallery. But let's think far beyond this. Every message we want to deliver to the public user of the city space we can bind to a certain location and save it on a '*separate layer*' in the cyberspace. When the user reaches the desired location in the city and turns on the layer with the information stored there, he will actually see what we left him there in the right place in form of an augmented layer on top of the reality. If the information we are talking about is for example an art project we call it '*locative art*' and this concept has been described in the novel *Spook Country* by William Gibson in 2007.

Location-based media (LBM) delivers multimedia and other content directly to the user of a mobile device dependent upon their location. Location information determined by means such as mobile phone tracking, GPS and other emerging Real-time locating system technologies like Wi-Fi. [46] http://en.wikipedia.org/wiki/Locative_media

The physicist Michio Kaku, for example, describes a world emerging between 2020 and 2050 in which computers as we know them will be superseded by an invisible network with the power of artificial intelligence (Broderick, 2002). But who owns these complex network patterns ergo who controls them? A person? A corporation? Are they actually under control? Here it has to be said that the cliché about the ownership and the control of power seems to be still present. We witness the moguls of media empires (means to have '*the power over the information*') to be the puppet masters of the politics and economics of our time. As far as the exchange of information has supplanted the role of material exchange, the power to dominate these networks and their design, inevitably means to become the dominant dynamic behind the shaping of urban space. Even more with '*the distributed cognition*' (Bateson cited in Gandy, 2005) the built space of the city has not only become part of the human body but has begun to impinge upon the process of thought itself. As soon as we understand that we all are '*cyborg*', we will be fascinated by the '*cyborg landscape*' because we can relate to us from it. It tells us something about ourselves as well as about the complex changes underway in the structure and meaning of contemporary cities. [47] Mathew Gandy, "Cyborg Urbanization," International Journal of Urban and Regional Research, 2005

The cyborg concept can in this sense be enlisted into an architectural project to '*rematerialize*' the city and establish substantive connections between the body, technology and space. This means we can design the space to suit our body as well as the technology we are part of. Let's say using the same principles cyberspace is build of, we could design a sort of '*compatible space*' for us to live in. As we learned to use the cyberspace based on some principles, it will be easier for us if the environment we inhabit is working the same way.

But there are quite a few dangerous scenarios in this theory as well. As for example the question of '*service or servitude*' posted by Virilio. The old public services are in danger of being replaced by a domestic enslavement whose crowning glory would surely be home automation. Achieving a domiciliary inertia, the widespread use of techniques of '*environmental control*' will end in behavioural isolation, in intensifying the insularity that has always threatened the town, the difference between the (separate) '*block*' and the (segregated) '*ghetto*' remaining precarious. [48] Stephen Graham, "The Cybercities reader," Urban Reader Series, p. 80, 2004

Let's not forget about **Archigram**...

The very core of *Archigram's* belief was that architectural form and the tight bound between place and architecture needed to be erased. '*By dissolving place into a nexus of servicing points joined by free-roving human receptors, it too threatened to dissolve place,*' Mr. Sadler writes. '*Archigram sought the solution to modernism's shortcoming in making modernism more extreme; the appetite, postmodernists were discovering, was for the opposite.*' [49] Simon Sadler, cited from article by Clay Risen "Rebellious Brit Architects Pushed Modernity to the Limit," 2005

The modern architecture is no more fighting the private/public combination based on the concept of possession; rather the key concern is the question of access to information or the right to get the information. Whereas possession implies the existence of private life based in physical exclusion or inclusion, access implies the existence of practices that use patterns rather than presences to distinguish between those who do and those who don't have the right to enter (this is where computerized surveillance enters the stage of architecture). Moreover, entering is itself constituted as access to data rather than as a change in physical location. [50] M. Christine Boyer, "CyberCities," Princeton Architectural Press, p. 11, 1996

Conclusion

Now, although Gibson thinks the future has already happened let us try to picture what it could look like in this case.

As we enter the Information Age and with Castells theory about the space of places vs. the space of flows (best demonstrated on museums, convention centres and places where people meet together open-minded), I have chosen a cyberspace gallery to be the function of my project, as much real as virtual.

This choice is not random. The gallery is a space where people meet open-minded and therefore it can be a place where experiments with space and new ways of using it, can be demonstrated more than somewhere else. Furthermore art as a language is suitable for digitalization and virtualisation. I was thinking about a way to seamlessly include a virtual node into the city fabric and an art gallery looked just the right object for that. I imagine everyday users as much as foreign tourists using it to get in touch with the actual city it is placed in. The art on display will comprise both local artists and well-known masters. The former ones will be displaying their art on the spot they know and prefer, while the art of the letter ones could be simply placed in never before tried out context, places never designed or though of. Who wouldn't like to see a 10 m high '*hologram*' of a Rodin sculpture in the middle of his local street, with cars and busses riding right through it?

The idea is to create a network of different nodes (spots) around the cityscape where art can be displayed in different settings. A tourist can use this network to navigate through the cityscape (he is not used to) and explore as much the well known spots as the hidden corners. For the local this can be a source of inspiration on art he is unable to find in the local museum as much as it gives one the opportunity to look at everyday space in a different way.

What if there will be no need for a physical access point, in terms of a building or a gate-away. What if we have a node (area) in which we are automatically connected, let's say just through simply be there. Imagine a space constantly communicating with us and reacting to our needs and wishes. If we would be able to alter the space (e.g. shape, light, sound...) freely to the current need of the user using semi AI controlling the environment communicating with the person via wireless devices connected to the human body or simply by intelligent scanning of the person. This goes far beyond what *Archigram* suggested that architectural form isn't bound between place and architecture and still it doesn't sound as science fiction anymore. When '*place*' is dissolved in '*endless waters of the cyberspace*' we will have to come up with '*place less*' solutions.

First of all, this project will be settled into the world of Cyberpunk as known from the literature like Neuromancer or movies like Blade Runner, Brazil, Fifth Element etc. into a globalized, digitalized virtualized, technologized, miniaturized world, accessible at high-

speed, into a brave new world as the future of possible. I will assume everything this might offer as a given for the terms of this diploma.

This architecture design, embedded into the modern computerized city, will demonstrate why cyberspace-architecture can offer the possible solutions or ways out.

The term cyberspace-architecture, where cyberspace (and cyberculture) will merge with reality to form a compatible space and the objects (the art) at display (even if real or virtual art) will be presented let's say on the edge of the merged boundary of cyberspace. The user will see himself at the same time in the real and virtual space. In such a setup, the virtual will enforce and interact with the real and vice versa. A perfect interaction, mutual representation, completion of one to the other, not the opposite. I'm still not quite sure what it will look like but it is a point to be demonstrated. After all I'm trying to rematerialize virtuality. There will be two parts of the project merging into one object. The first one a virtual building (*The Beast*) of the gallery on a real spot in the cityscape visible for every potential random user just walking by. The second a virtual twin (inside of the first) coming out once activated by an active user. This part acts like an interface between the real user and the vast information inside the database of the gallery. Allowing the user to browse the data, pick the desired art object and place it in the real space around him. The user will create his identity in cyberspace through surfing the gallery web in nonlinear fashion just to be appreciated for the ride as we are used by now on the Internet (note that gender-swapping is more than allowed :-)). The user will himself experience the navigation via layers or for example the learning curve of Sisyphus, as he adopts his process of thinking and orientation to the cyberspace gallery, only to realize that this can change immediately. He will communicate with other users (even physically) and internet users as the cyberspace gallery will be connected to the web showing him all the art outside the real gallery but still inside the virtual one. To demonstrate the idea of Reduction of distances he will have access to the art not physically present in the gallery via Internet and be able to perceive art virtually from distance 0 to ∞ through zooming in. The art will be transformed into data to show that any content is capable of being transformed this way.

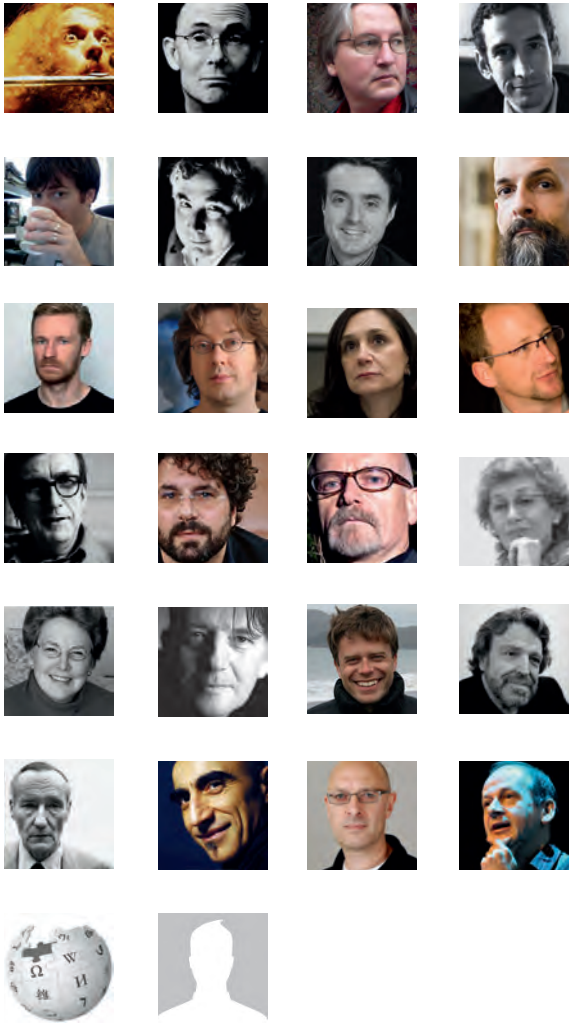
As a goal of working against The Digital Divide, the access shouldn't be just for the metropolitan nation, the networkers (even if mostly for them rightly so), but the cyberspace gallery is opened for the web user as well as the rare gallery visitor who can be animated to enter as much as he is to go onto the net. This includes the decisions about the spots in the cityscape, urban and digital connection. For some it might even become something similar

to a meeting place on day to day basis, an everyday forum like Facebook as long as it fully opens up it's potential.

Chaos is a legitimate way for the visitor to grasp the creation of cyberspace. Through computerized surveillance it will be possible to show the quality of cyberspace on many different examples, to create a space where one can focus on thinking cyberspace. I'm not trying to talk just about self-reflection, moreover self-realisation, not to show one does something good or wrong but to show what one does in a number of different views, aspects, forms and examples, to represent the digital trace and movement of the visitor through the virtual world and the actual space of the gallery.

1 Douglas Rushkoff, “Cyberia,” 1994
2 William Gibson, “Neuromancer,” *London: Grafton,* 1984
3 Douglas Rushkoff, “Cyberia,” 1994
4 Mark Dery, “Cyberculture,” *South Atlantic quarterly 91,* 1992
5 David Bell, “Cyberculture Theorists,” *Routledge critical thinkers,* p. 4, 2007
6 Wikipedia, “Cyberpunk,” *http://en.wikipedia.org/wiki/Cyberpunk*
7 Bruce Sterling, In “Mirrorshades: The Cyberpunk Anthology,” *Arbor House,* preface, 1986
8 Bruce Sterling, “Cyberpunk in the Nineties,” *http://cyber.eserver.org/sterling/interzon.txt*
9 William Gibson, Bruce Sterling, “The Difference Engine,” *Victor Gollancz Ltd.*, 1990
10 Douglas Rushkoff, “Cyberia,” 1994
11 Stephen Graham, “The Cybercities reader,” *Urban Reader Series,* p. 11, 2004
12 William Gibson, “Pattern Recognition,” *G. P. Putnam’s Sons,* p. 22–23, 2003
13 Douglas Rushkoff, “Cyberia,” 1994
14 Sherry Turkle, “Who am we?,” *wired,* 4.01, 1996
15 Douglas Rushkoff, “Cyberia,” introduction, 1994
16 Sherry Turkle, “Who am we?,” *wired,* 4.01, 1996
17 Stephen Graham, “The Cybercities reader,” *Urban Reader Series,* p. 11, 2004
18 Wiliam Gibson, “Johnny Mnemonic,” *Omni magazine*, 1981
19 Stephen Graham, “The Cybercities reader,” *Urban Reader Series,* p. 15, 2004
20 Katherine Hayles, “How we became Posthuman,” *The University of Chicago press,* p. 27, 1999
21 Manuel Castells with Catterall, “Cyberculture Theorists,” *Routledge critical thinkers,* by David Bell, p. 59, 2007
22 Douglas Rushkoff, “Program or be Programmed,” *Soft Skull Press,* p. 124, 2011
23 Mathew Gandy, “Cyborg Urbanization,” *International Journal of Urban and Regional Research,* 2005
24 Stephen Graham, “The Cybercities reader,” *Urban Reader Series,* p. 69, 2004
25 Sherry Turkle, “Who am we?,” *wired,* 4.01, 1996
26 William Gibson, “Pattern Recognition,” *G. P. Putnam’s Sons*, 2003
27 M. Christine Boyer, “CyberCities,” *Princeton Architectural Press,* p. 11, 1996
28 Marcos Novak, “Liquid architectures in cyberspace,” In Cyberspace: First Steps, *MIT Press,* p. 230, 1991
29 Sherry Turkle, “Who am we?,” *wired,* 4.01, 1996
30 Sherry Turkle, “Who am we?,” *wired,* 4.01, 1996
31 William Gibson, “Spook Country,” Penguin Group, p. 142, 2008
32 Roberto Baldwin, “Google Hiring Augmented-Reality Experts Amid Rumors of HUD Glasses,” *wired.com*, February 27, 2012
33 David Bell, “Cyberculture Theorists,” *Routledge critical thinkers,* p. 71, 2007
34 Stephen Graham, “The Cybercities reader,” *Urban Reader Series,* p. 12, 2004
35 William Mellor and Allen Cheng, “Beijing Home to $160b Worth of Construction,” *Vancouver Sun,* march 4, 2006
36 Deyan Sudjic, “The City That Ate the World,” *Guardian,* October 16, 2005
37 David Bell, “Cyberculture Theorists,” *Routledge critical thinkers,* p. 19, 2007
38 Roger Burrows In “The Cybercities Reader,” ed. Stephen Graham, *Routledge,* p. 389, 2004
39 David Bell, “Cyberculture Theorists,” *Routledge critical thinkers,* p. 9, 2007
40 David Bell, “Cyberculture Theorists,” *Routledge critical thinkers,* p. 22, 2007
41 Mike Crang In “The Cybercities Reader,” ed. Stephen Graham, *Routledge*, p. 130, 2004
42 David Bell, “Cyberculture Theorists,” *Routledge critical thinkers,* p. 66, 2007
43 Matthew Gandy, “Cyborg Urbanization,” *International Journal of Urban and Regional Research,* 2005
44 Matthew Gandy, “Cyborg Urbanization,” *International Journal of Urban and Regional Research,* 2005
45 Stephen Graham, “The Cybercities reader,” *Urban Reader Series,* p. 91, 2004
46 Wikipedia, “Locative media,” http://en.wikipedia.org/wiki/Locative_media
47 Matthew Gandy, “Cyborg Urbanization,” *International Journal of Urban and Regional Research,* 2005
48 Stephen Graham, “The Cybercities reader,” *Urban Reader Series,* p. 80, 2004
49 Simon Sadler, cited from article by Clay Risen “Rebellious Brit Architects Pushed Modernity to the Limit,” *http://www.observer.com/node/37855*, 2005
50 M. Christine Boyer, “CyberCities,” *Princeton Architectural Press,* p. 11, 1996

www.wikipedia.com
http://project.cyberpunk.ru/
Greg Lynn, “Form,” *Rizzoli International Publications,* 2008
Bruno Latour, “Wir sind nie modern gewesen,” *Fischer,* 1998
Ortlos, “Architecture of the networks,” *Hatje Cantz,* 2005



Book 1 – Analysis of History and Theory regarding cyberspace-architecture and cyberpunk
Book 2 – Vocabulary of cyberspace-architecture and cyberpunk
Book 3 – Notes and sketches
Book 4 – Scripts and codes
Book 5 – Presentation

Cyberpunk 間 Architecture

A humble attempt to give visual expression to new models of reality by Samo Bertok

Book 2 – Vocabulary of cyberspace-architecture and cyberpunk

First step towards anything,
is a list of the keywords
related to subject >>

cyberspace

A consensual hallucination… a graphic representation of data abstracted from the banks of every computer in the human system. Unthinkable complexity. Lines of light ranged in the nonspace of the mind, clusters and constellations of data. Like city lights, receding.[1]
This term is used to describe the space created through communication networks such as Internet. Enabling computer mediated communication between users present in this public space.

> it's a wide world out there so much wider than imagined. Ian Anderson, *Jet Cam*, Jethro Tull

1
Neuromancer, William Gibson, 1984

<img>
Wormser Straße Cyberspace
Pierre Vlcek / www.flickr.com 10|12|08

virtual world

Is a digitally simulated environment inhabited by users interacting via representative personas *avatars*. The virtual world of what is now known as the Internet often appears under various names, including *cyberspace*, *wired*, *Metaverse*, *Second Life* or *matrix*. Often it happens the users claim the virtual to be more real than the *actual* real or in other words the distinction between the real and the virtual is vanishing gradually.

<img>
The Construct from the Wachowski brothers movie Matrix
http://matrix.wikia.com/wiki/Construct
10|01|10

augmented reality

Is an environment that includes both virtual reality and real-world elements and the surrounding real world of the user becomes interactive and digitally usable.[2] For instance, an AR user might wear translucent goggles; through which, he could see the real world as well as computer generated images projected on top of that world. An augmented reality system as one that combines the real and the virtual, is interactive in real-time and can be registered in three dimensions.

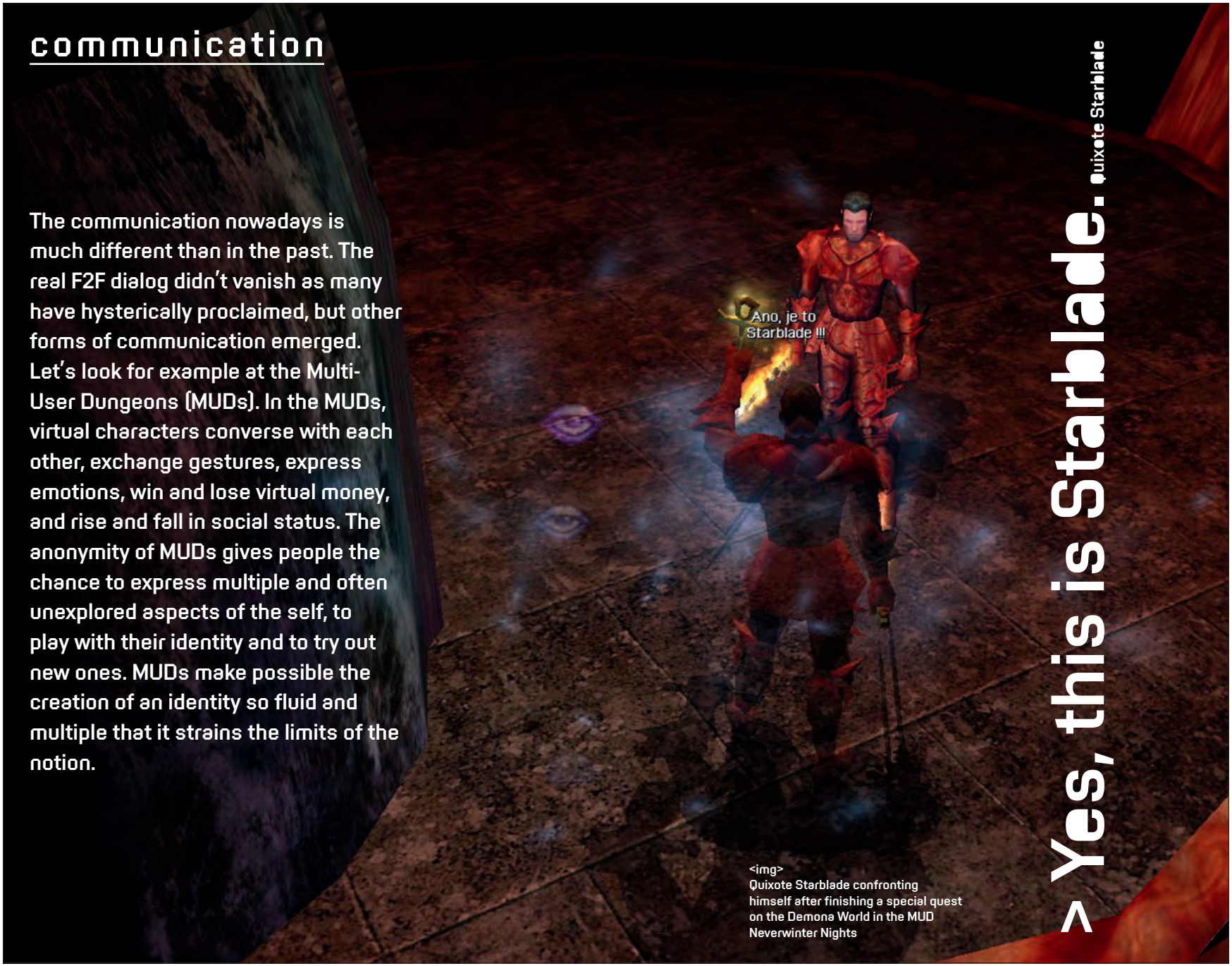
> Would it all be like this in Alberta's new world of locative? Would it mean that the untagged, unscripted world would gradually fill with virtual things, as beautiful or ugly or banal as anything encountered on the web already. William Gibson, *Spook Country*

2
http://en.wikipedia.org/wiki/Augmented_reality
08|01|10

<img>
oximoron studio,
Cvernovka, Bratislava

communication

The communication nowadays is much different than in the past. The real F2F dialog didn't vanish as many have hysterically proclaimed, but other forms of communication emerged. Let's look for example at the Multi-User Dungeons (MUDs). In the MUDs, virtual characters converse with each other, exchange gestures, express emotions, win and lose virtual money, and rise and fall in social status. The anonymity of MUDs gives people the chance to express multiple and often unexplored aspects of the self, to play with their identity and to try out new ones. MUDs make possible the creation of an identity so fluid and multiple that it strains the limits of the notion.



<img>
Quixote Starblade confronting himself after finishing a special quest on the Demona World in the MUD Neverwinter Nights

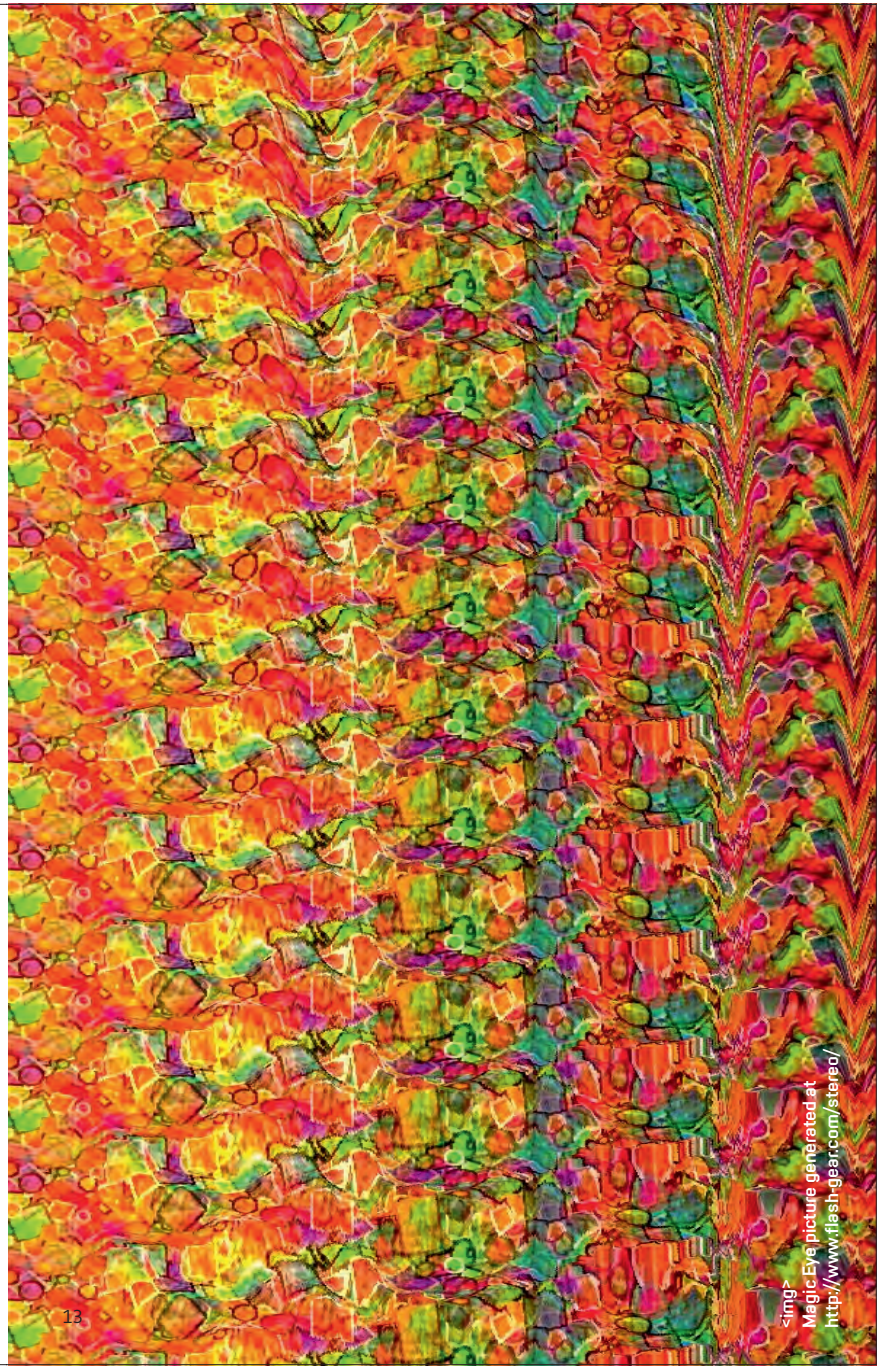
> Yes, this is Starblade. Quixote Starblade

chaos

As from that is associated with the modernist city *chaos* can be perceived as a field of evolutionary and radical experimentation. Chaos is no longer seen as an anomaly to the urban experience (where everything must have its given order) to cause problems or be excluded from analysis but a rich vein of social and spatial interaction as such. Through which we may perceive signs of alternative or until now overlooked urban forms. Chaos may also be characterized as a more sophisticated, resilient and adaptable form of order.[4]

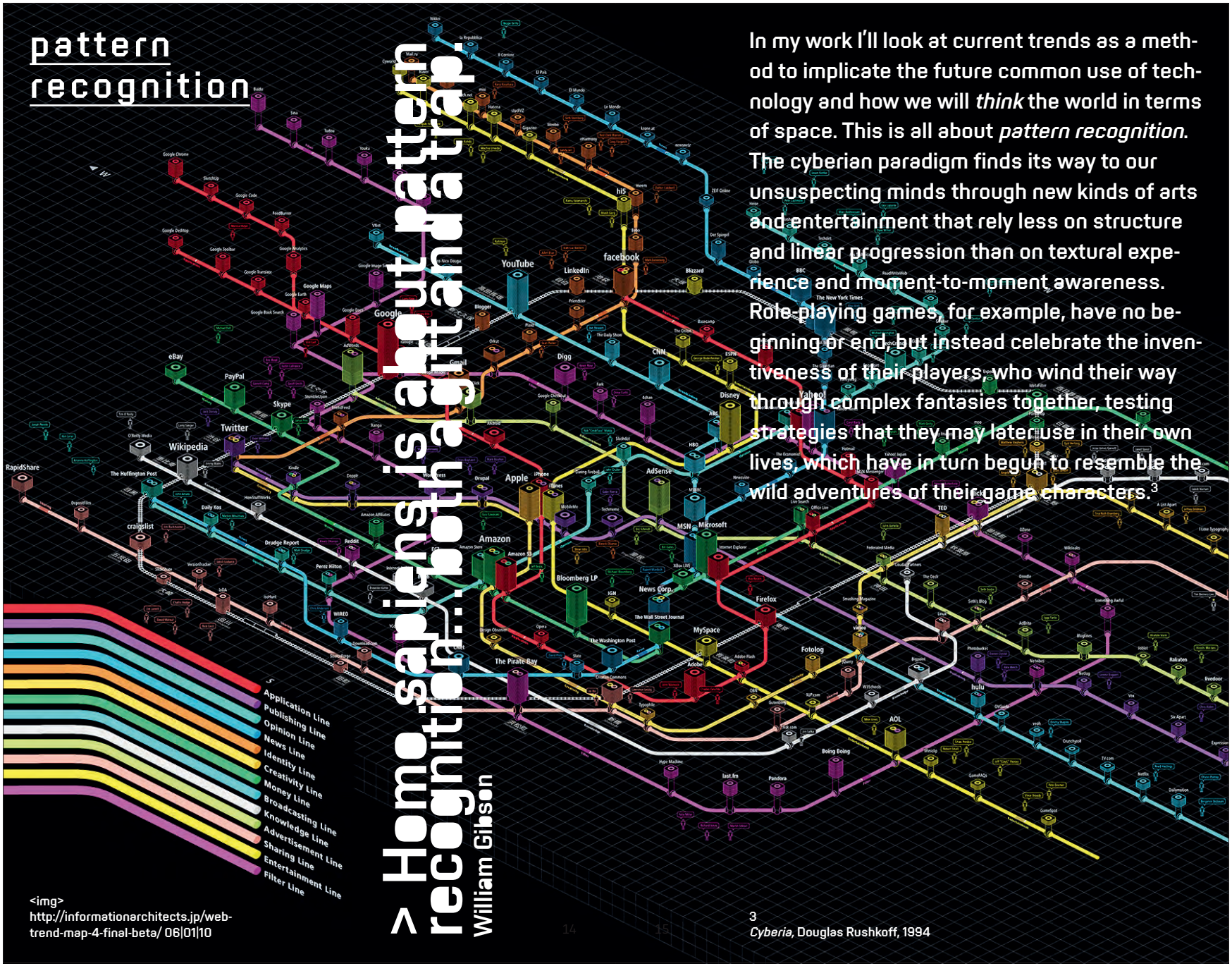
4
Koolhaas et al. , 2001
Koolhaas, 2002

> Architecture can't fully represent the chaos and turmoil that are part of the human personality, but you need to put some of that turmoil into the architecture, or it isn't real. Frank Stella



<image> Magic Eye picture generated at http://www.flash-gear.com/stereo/

pattern recognition



<img>
http://informationarchitects.jp/web-trend-map-4-final-beta/ 06|01|10

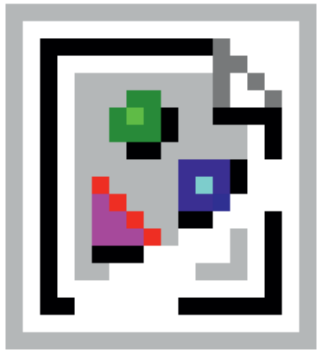
> Homo sapiens is about pattern recognition... both at and in a trap.
William Gibson

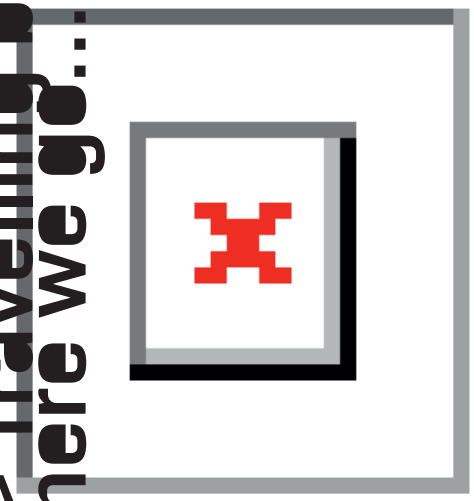
In my work I'll look at current trends as a method to implicate the future common use of technology and how we will *think* the world in terms of space. This is all about *pattern recognition*. The cyberian paradigm finds its way to our unsuspecting minds through new kinds of arts and entertainment that rely less on structure and linear progression than on textural experience and moment-to-moment awareness. Role-playing games, for example, have no beginning or end, but instead celebrate the inventiveness of their players, who wind their way through complex fantasies together, testing strategies that they may later use in their own lives, which have in turn begun to resemble the wild adventures of their game characters.[3]

3
Cyberia, Douglas Rushkoff, 1994

presence

Is the now. Means to be in one place at a specific time. That is how we as architects are skilled to think of a real body/object in space. But there is not the body in the cyberspace to be concerned with at the first place. There is other kind of presence one that is no more bound with a place in particular. Regarding the mind or the user as the body/object moving and taking presence it is not a now or one place far more a now and everywhere. Travelling by hyperlink and email, showing presence in a chat or MUD's.





> Travelling by telephone, hey ho here we go...
Syd Barrett, *The Flaming*, Pink Floyd

<img>
icons shown in various web browsers
when an image fails to load properly

neuromancer

The most famous and influential novel [1984] by the *guru* of the cyberpunk literature, William Gibson. Not only defining the term cyberspace but crucial for the whole genre and next generation of sci-fi literature. Acts as the first part of the Sprawl Trilogy, followed by *Count Zero* [1986] and *Mona Lisa Overdrive* [1988]. Describes a world full of hackers, AI's and new subcultures, who are trying to claim space in the new digital world dominated by corporate capital, by *jacking into the cyberspace*.

<img>
Splashscreen from Neuromancer game from 1988 by Troy A. Miles, Interplay Production Inc.

cyberpunk

A cyberpunk is a person who takes/ having control of cyberspace/ electronic computer and uses it for the army or the government or Lufthansa Airline, but to give his own purpose. Timothy Leary

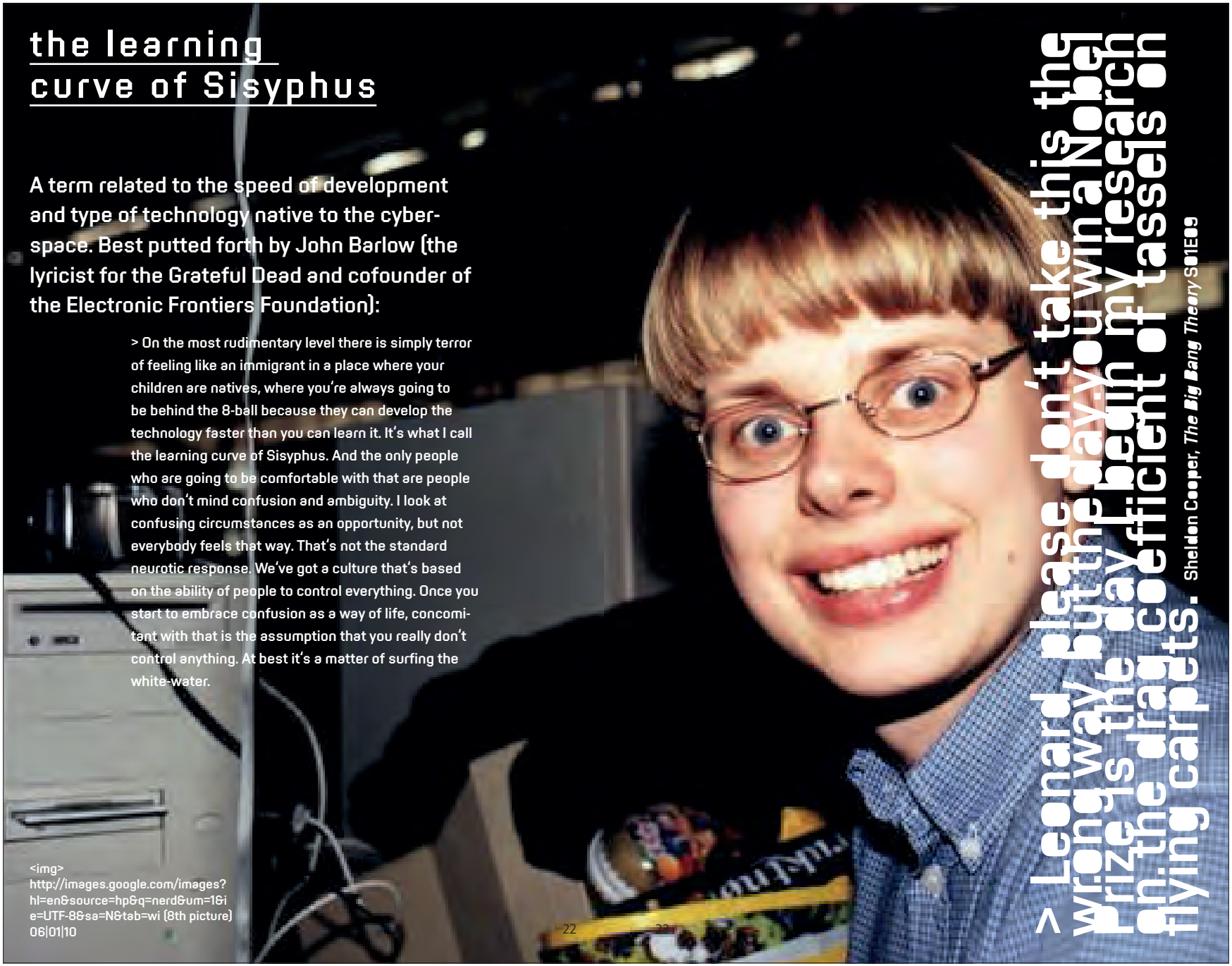
Is a subgenre of science fiction literature and film, with it's origins in 1980's, and associated with writers such as William Gibson, Bruce Sterling, Jon Courtenay Grimwood and Neal Stephenson. In cinema, films like *Blade Runner* [1982] are seen as encapsulating the spirit and aesthetic of cyberpunk. Cyberpunk centres on the impacts of new technologies such as computers and virtual reality, and with propagating popular images of cyberspace, cyborgs, artificial life forms, and so on.

<img>
Kowloon Walled City
http://www.archidose.org/KWC/

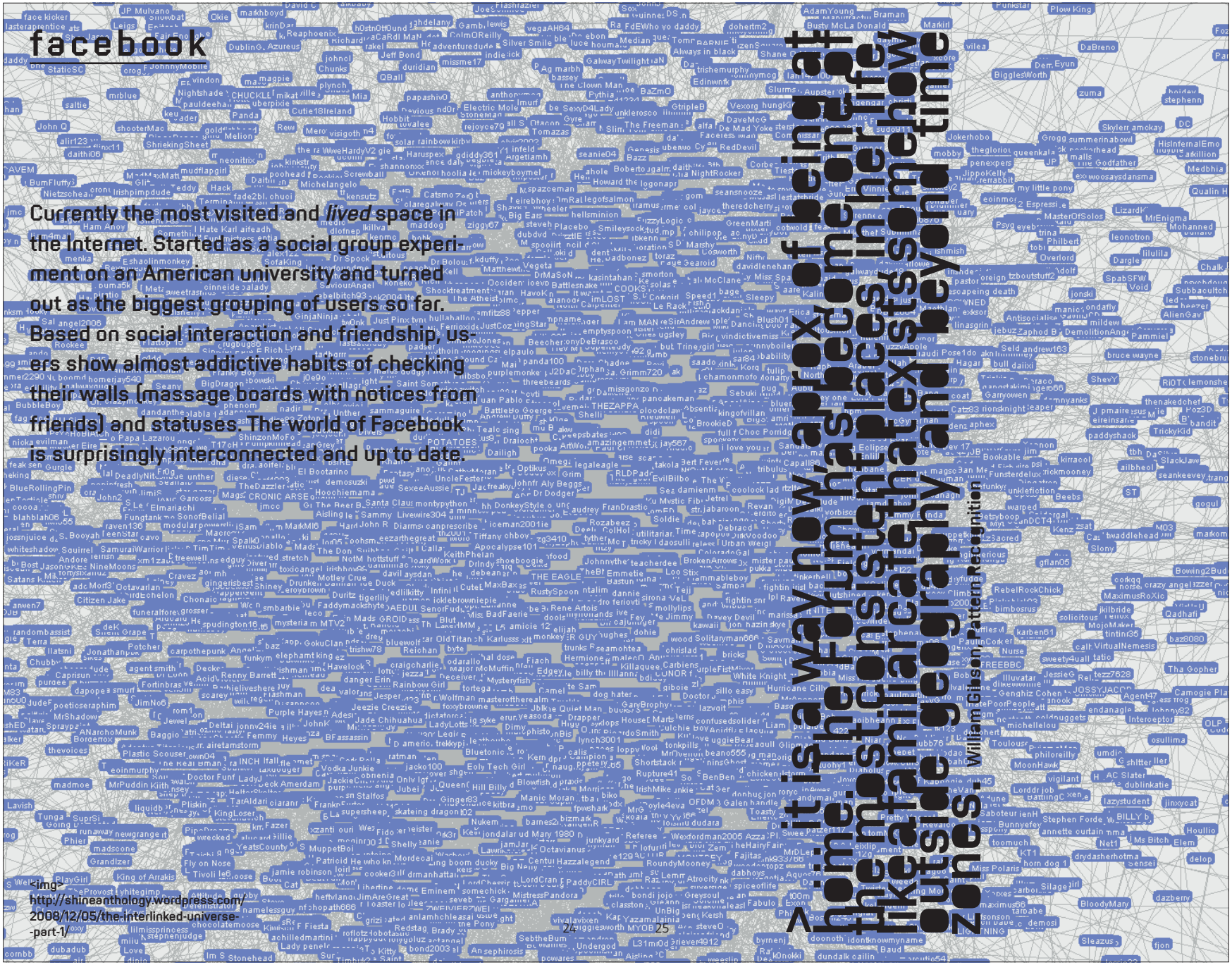
the learning curve of Sisyphus

A term related to the speed of development and type of technology native to the cyber-space. Best putted forth by John Barlow (the lyricist for the Grateful Dead and cofounder of the Electronic Frontiers Foundation):

> On the most rudimentary level there is simply terror of feeling like an immigrant in a place where your children are natives, where you're always going to be behind the 8-ball because they can develop the technology faster than you can learn it. It's what I call the learning curve of Sisyphus. And the only people who are going to be comfortable with that are people who don't mind confusion and ambiguity. I look at confusing circumstances as an opportunity, but not everybody feels that way. That's not the standard neurotic response. We've got a culture that's based on the ability of people to control everything. Once you start to embrace confusion as a way of life, concomitant with that is the assumption that you really don't control anything. At best it's a matter of surfing the white-water.



> Leonard, please don't take this the wrong way, but the day you win a Nobel Prize is the day I begin my research in the drag coefficient of tassels on flying carpets. Sheldon Cooper, *The Big Bang Theory* S1E09

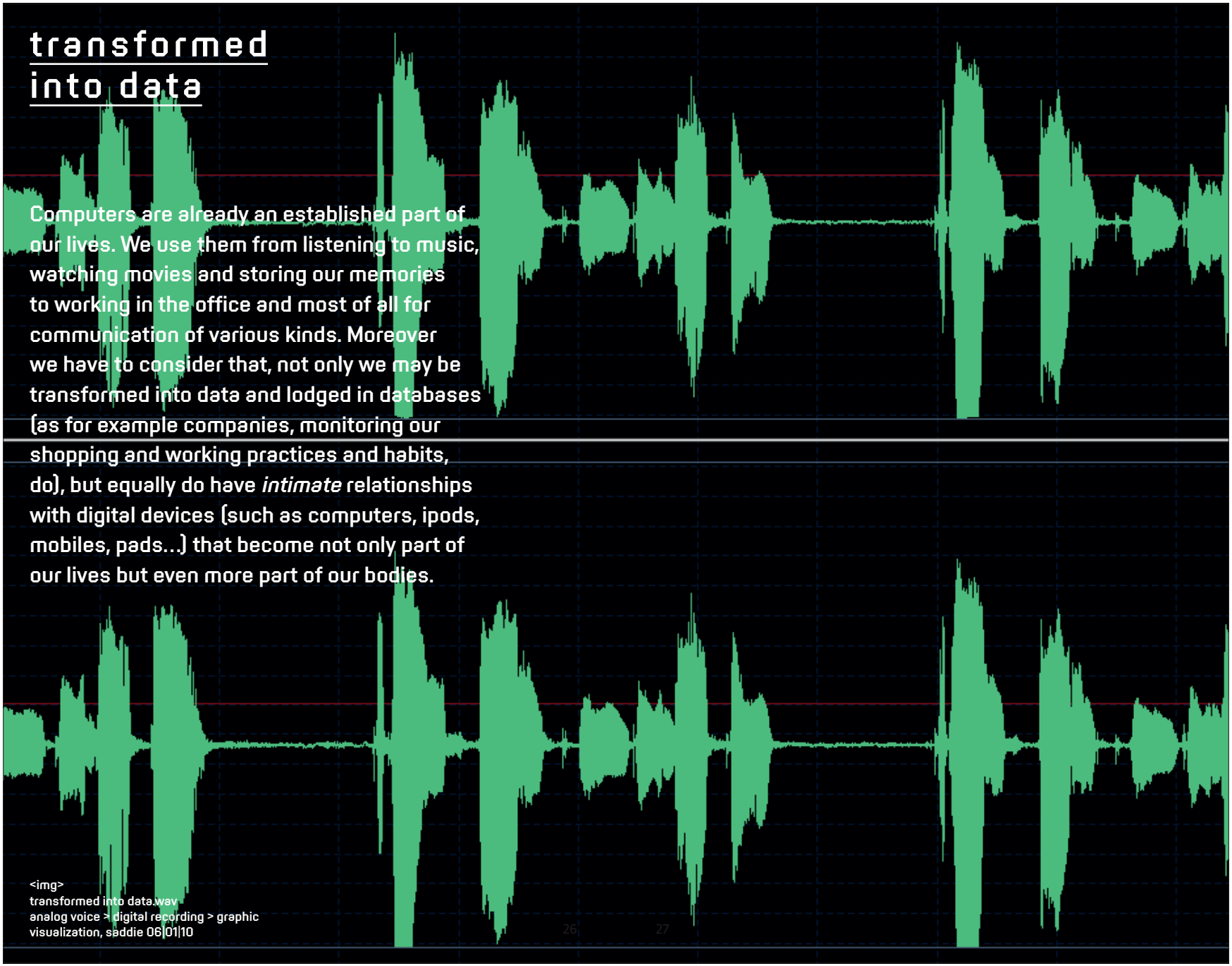


Currently the most visited and *lived* space in the Internet. Started as a social group experiment on an American university and turned out as the biggest grouping of users so far. Based on social interaction and friendship, users show almost addictive habits of checking their walls (massage boards with notices from friends) and statuses. The world of Facebook is surprisingly interconnected and up to date.

img: http://shineanthology.wordpress.com/2008/12/05/the-interlinked-universe--part-1/

> "It is a way of approximating the most sophisticated places that exist somewhere outside geography and beyond time zones."
>
> William Gibson, Pattern Recognition

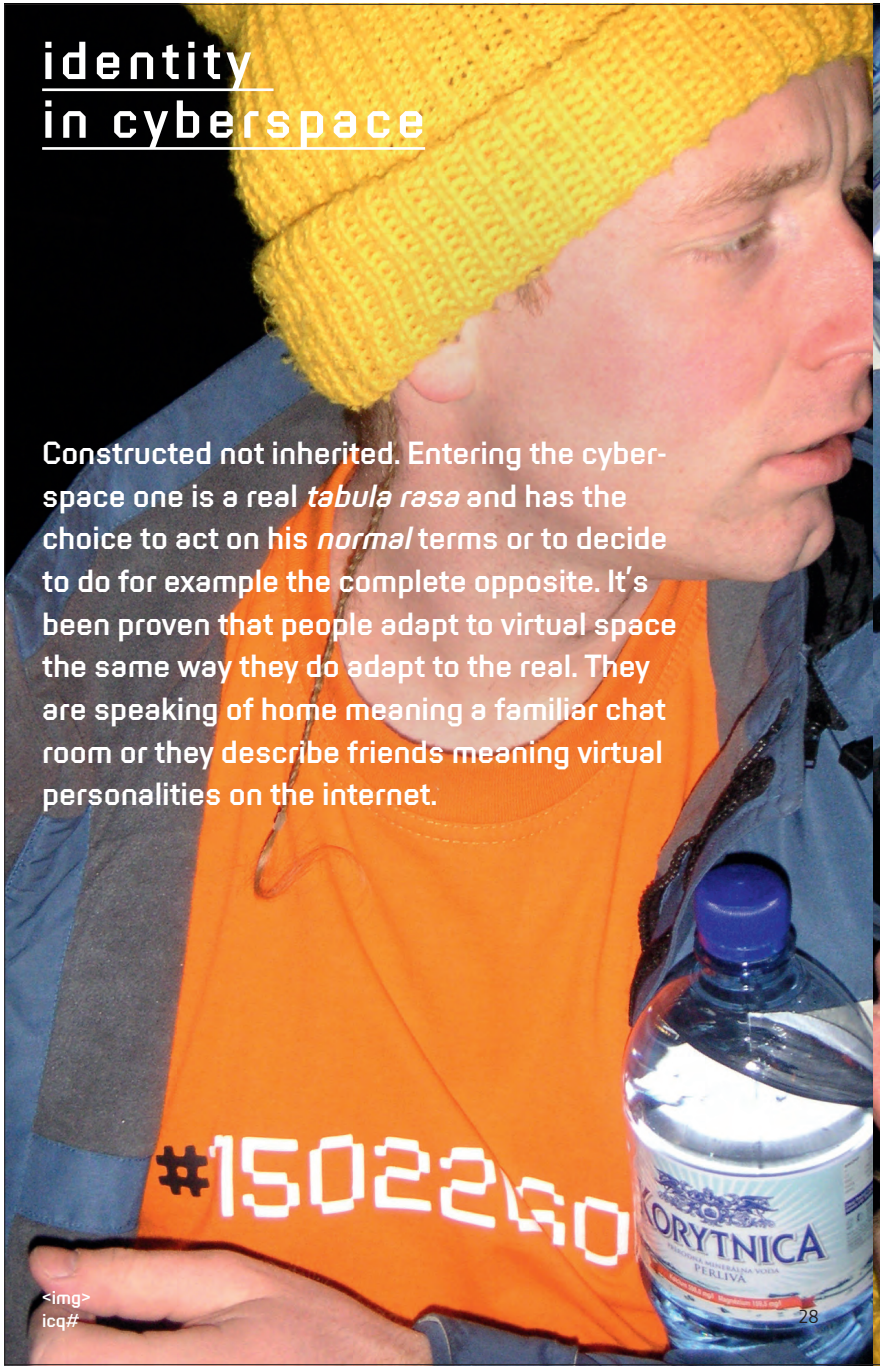
transformed into data



Computers are already an established part of our lives. We use them from listening to music, watching movies and storing our memories to working in the office and most of all for communication of various kinds. Moreover we have to consider that, not only we may be transformed into data and lodged in databases (as for example companies, monitoring our shopping and working practices and habits, do), but equally do have *intimate* relationships with digital devices (such as computers, ipods, mobiles, pads...) that become not only part of our lives but even more part of our bodies.

identity in cyberspace

Constructed not inherited. Entering the cyber-space one is a real *tabula rasa* and has the choice to act on his *normal* terms or to decide to do for example the complete opposite. It's been proven that people adapt to virtual space the same way they do adapt to the real. They are speaking of home meaning a familiar chat room or they describe friends meaning virtual personalities on the internet.



> The past week reminded us that users feel a real sense of ownership over Facebook itself, not just the information they share. Mark Zuckerberg, on user voting on terms of service of FB

digital trace

> Money beats soul everytime.

Jim Morrison, ***Roadhouse Blues, Morrison Hotel***

One can buy any information. This now leads to new means of social control even better than Bentham's Panopticon. I'm speaking of people becoming so called *digital subjects*, *files* monitored, tracked, controlled, commodified and stored. In this sense we watch ourselves (similar to the idea of Bentham) as we participate within the webs of infinite digital connections, always leaving our digital tracks as objects further analysis. Unlike Carbon-footprint the digital trace has no impact on the earth environment, on the other hand it leaves an uncleanable trace of the identity surfing the cyberspace.

<img>
Books offered by recommender system at Amazon.com after choosing David Bell's Cyberculture Theorists 06|01|10

gender-swapping

Is not a small part of the game action on a MUD. By some estimates, *Habitat*, (a Japanese MUD), has 1.5 million users and is operated for profit. Among the registered members of Habitat, there is a ratio of four real-life men to each real-life woman. But inside the MUD the ratio is only three male characters to one female character.[5] In other words, a significant number of players, many tens of thousands of them, are virtually cross-dressing. Which means the space designed for such frequent 180° changes must be equally flexible.

> A female is more likely to get what she wants and male characters are more generous when they deal with females. unknown man, regarding playing as female avatars

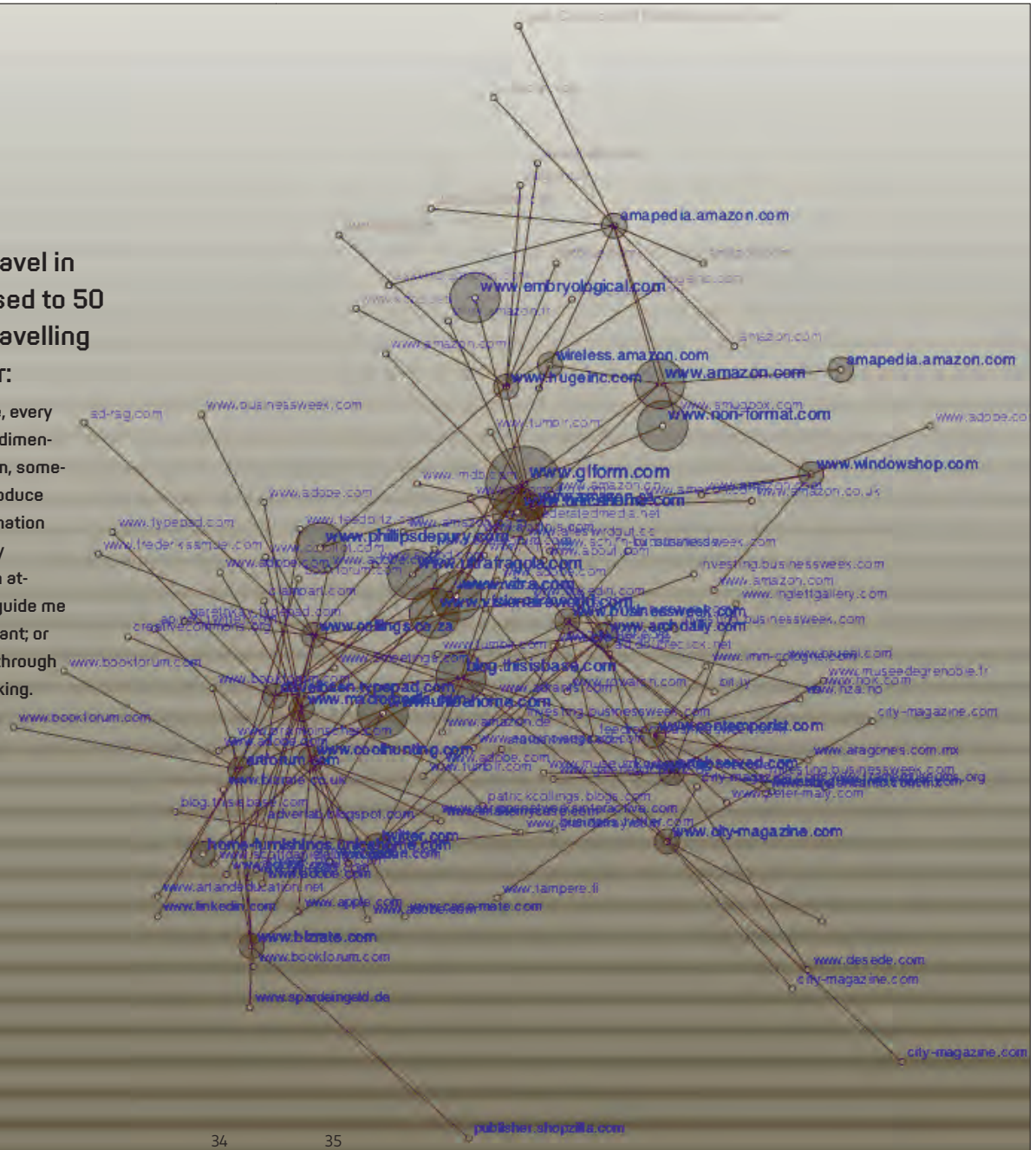
5
Who am we?, Sherry Turkle, wired, 4.01, 1996

<img>
http://fhg.sheanimale.com/sheanimale/204_3D_Military_Shemales_Fucking/index.php?id=1420732

hyperlink – reduction of distances

Travel? Well to be honest we do not travel in the same way or sense as we were used to 50 years ago. Marcos Novak describes travelling through hypertext in following manner:

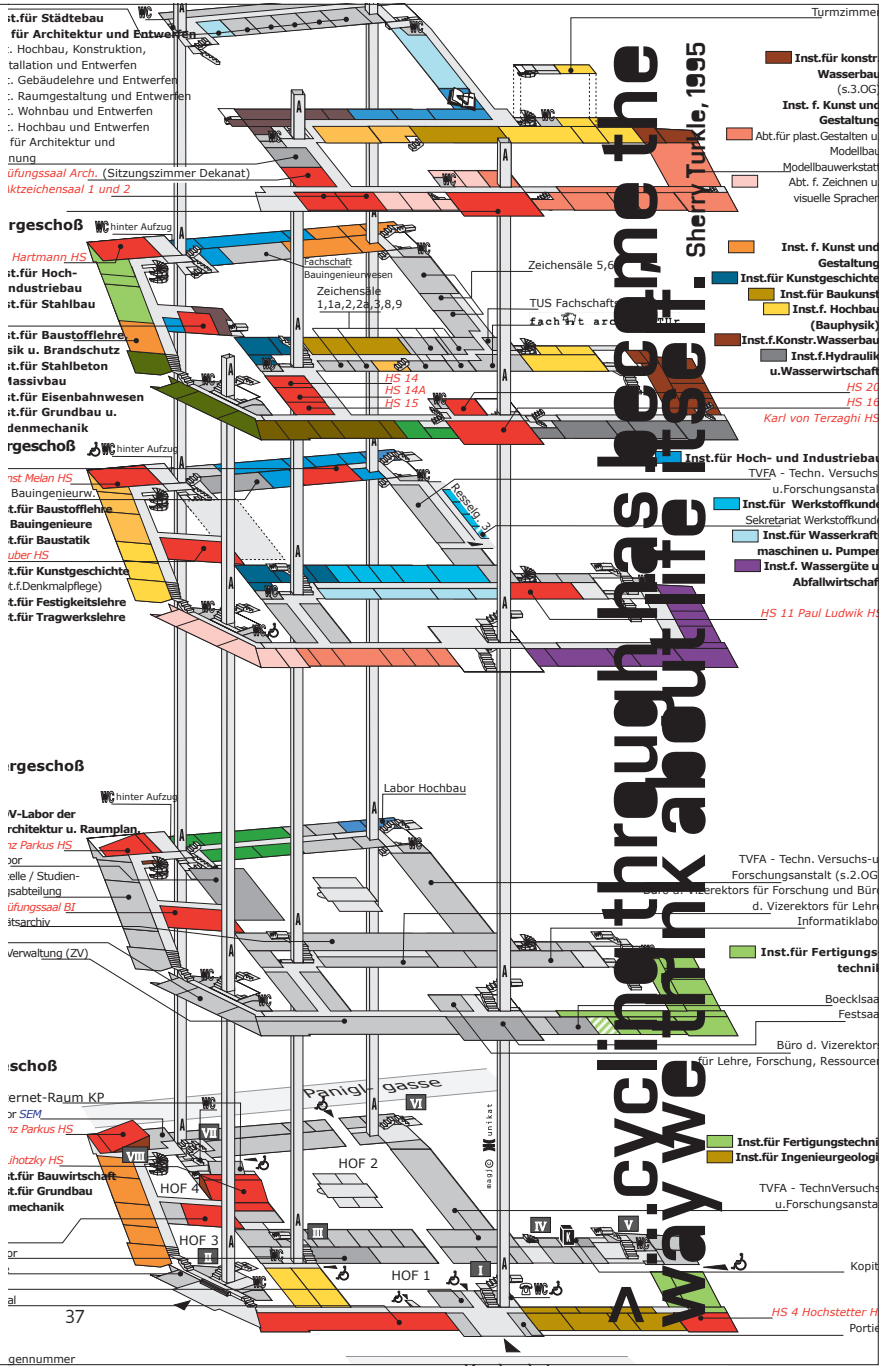
> Every paragraph an idea, every idea an image, every image an index, indices strung together along dimensions of my choosing, and I travel through them, sometimes with them, sometimes across them. I produce new sense, nonsense, and nuisance by combination and variation, and I follow the scent of a quality through sand dunes of information. Hints of an attribute attach themselves to my sensors and guide me past irrelevant, into the company of the important; or I choose to browse the unfamiliar and tumble through volumes and volumes of knowledge still in making.



<img>
Visualization of hyperlink connections
from http://www.glform.com
generated on 20|12|09 by NodeBox
with depht parameter set to 3

the world
in nonlinear fashion

We have established *windowing* as the way of working with computers, clicking between screens – windows, as between files and applications. It allows us to be in various contexts at same time, in real time. This experience has migrated out of the computers and subsequently turned our real life yet another *window* (thanks Bill :-)). As we moved from the world of representation to simulation, this is how we experience the reality. The way in which we move through the text says as much, if not more, about the cyberpunk worldview than does its particular post-sci-fi aesthetic.



<img>
http://www.wegweiser.ac.at/static/
plaene/pdf/E_Hauptgeb.pdf 28|11|09

Layers are good servants but bad masters.

Secret Rainbow Proverbs Conglomerate

digital divide

Refers to the gap between people with effective access to digital and information technology and those with very limited or no access at all. It includes the imbalances in physical access to technology as well as the imbalances in resources and skills needed to effectively participate as a digital citizen.[6] Includes problems with coverage, accessibility, literacy, speed of connection, open vs. closed... etc. Virtual borders same as borders in the urban space marking for example the public and private space.

> Computing is not about computers any more. It is about living. Nicholas Negroponte

6
http://en.wikipedia.org/wiki/Digital_divide 06|01|10

<img>
http://wiki.laptop.org/go/One_Laptop_per_Child 10|01|10

the space of places / the space of flows

As seen by Castells, there is a major change in the recent trend in architecture, which signals it's transformation from an intervention of *the space of places* to an intervention of *the space of flows* (which he assumes to be the dominant space of the informational age), by acting on space dedicated to museums, convention centres and transportation nodes. Here the architecture acts as mean of transformation of spaces of great cultural and communicational potential into forms of cultural expression and meaningful exchange of information. Or in other words, let us value architecture as it's ability to generate a discussion around itself, for at that point it has been inserted into the endless circulation of cultural signs.

> "...to reconceptualize new forms of spatial arrangements under the new technological paradigm" Manuel Castells

<img>
3295078060_cea34c036c_b.jpg
author unknown / www.flickr.com 10|12|08

background

Maria Bakardjieva, “Internet Society: The Internet in Everyday Life,” SAGE, 2005

Banksy, “Wall and piece,” Century, 2006

David Bell, “Cyberculture: the key concepts,” Routledge, 2004

David Bell, “Cyberculture Theorists,” Routledge critical thinkers, 2007

Michael Benedikt, “For an Architecture of Reality,” Lumen. Inc, 1987

M. Christine Boyer, “CyberCities,” Princeton Architectural Press, 1996

Mike Davis, Daniel Bertrand Monk, “Evil Paradises: Dreamworlds of Neoliberalism,” New Press, 2008

Richard Dawkins, “The Blind Watchmaker,” Norton & Company. Inc,1986

Richard Dawkins, “The Selfish Gene,” Oxford University Press, 1976

Mark Dery, “Cyberculture,” South Atlantic quarterly 91, 1992

Mark Dery, “Escape Velocity: Cyberculture at the End of the Century,” Grove Press, 1997

Mathew Gandy, “Cyborg Urbanization,” International Journal of Urban and Regional Research, 2005

William Gibson, “All Tomorrow’s Parties,” Ace Books, 2000

William Gibson, “Idoru, ” G.P. Putnam’s Sons, 1996

William Gibson, “Neuromancer,” Ace Books, 1984

William Gibson, “Pattern Recognition,” Berkley Books, 2005

William Gibson, “Spook Country,” Penguin Group, 2008

William Gibson, Bruce Sterling, “The Difference Engine,” Bantam Books, 1991

Stephen Graham, “The Cybercities reader,” Urban Reader Series, 2004

Jon Courtenay Grimwood, “9tail Fox,” Night Shade Books, 2007

Jon Courtenay Grimwood, “Lucifer’s Dragon,” Pocket Books, 2004

Jon Courtenay Grimwood, “Redrobe,” Simon & Schuster UK, 2001

Jon Courtenay Grimwood, “Remix,” Simon & Schuster UK, 2001

Jon Courtenay Grimwood, “Stamping Butterflies,” Bantam Books, 2006

Katherine Hayles, “How we became Posthuman,” The University of Chicago press, 1999

Christian Hundertmark (aka C100), “Art of Rebellion: The World of Street Art,” Gingko, 2005

Rem Koolhaas, “Content,” Taschen, 2004

Rem Koolhaas, “Delirious New York: a retroactive manifesto for Manhattan,” Monacelli Press, 1994

Bruno Latour, “Wir sind nie modern gewesen,” Fischer, 1998

Greg Lynn, “Form,” Rizzoli International Publications, 2008

William Mellor and Allen Cheng, “Beijing Home to $160b Worth of Construction,” Vancouver Sun, march 4, 2006

Ortlos, “Architecture of the networks,” Hatje Cantz, 2005

Sebastian Peiter, “Guerilla Art,” Laurence King Publishers, 2009

Douglas Rushkoff, “Cyberia: life in the trenches of hyperspace,” Clinamen Press, 2002

David Silver, “Critical cyberculture studies,” New York University Press, 2006

Deyan Sudjic, “The City That Ate the World,” Guardian, October 16, 2005

Edward R. Tufte, “Envisioning information,” Graphics Press, 1990

Sherry Turkle, “Who am we?,” wired, 4.01, 1996

Robert Venturi, Denise Scott Brown, Steven Izenour, “Learning from Las Vegas: the forgotten symbolism of architectural form,” MIT Press, 1977

http://project.cyberpunk.ru/

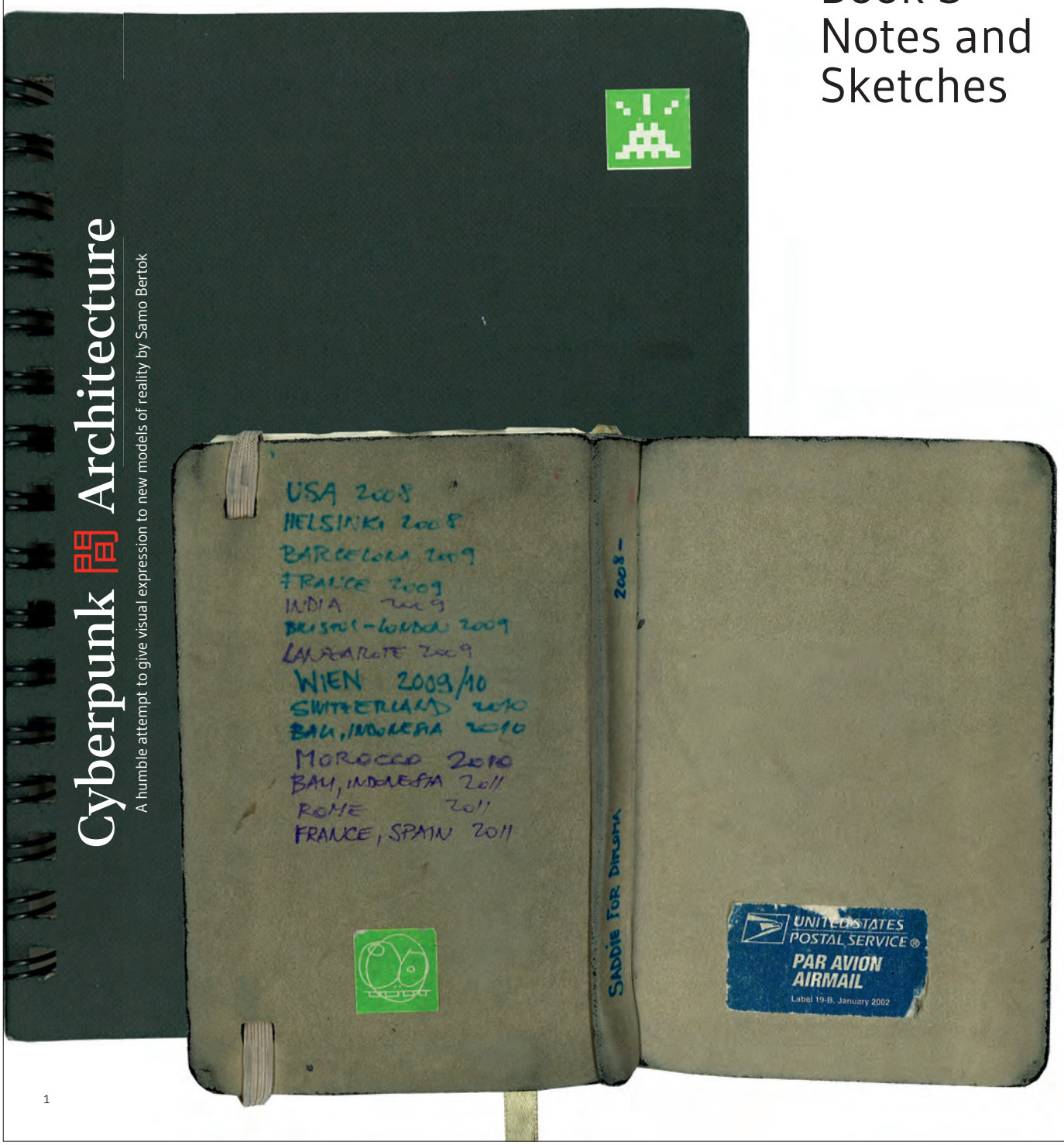
http://www.wikipedia.org/

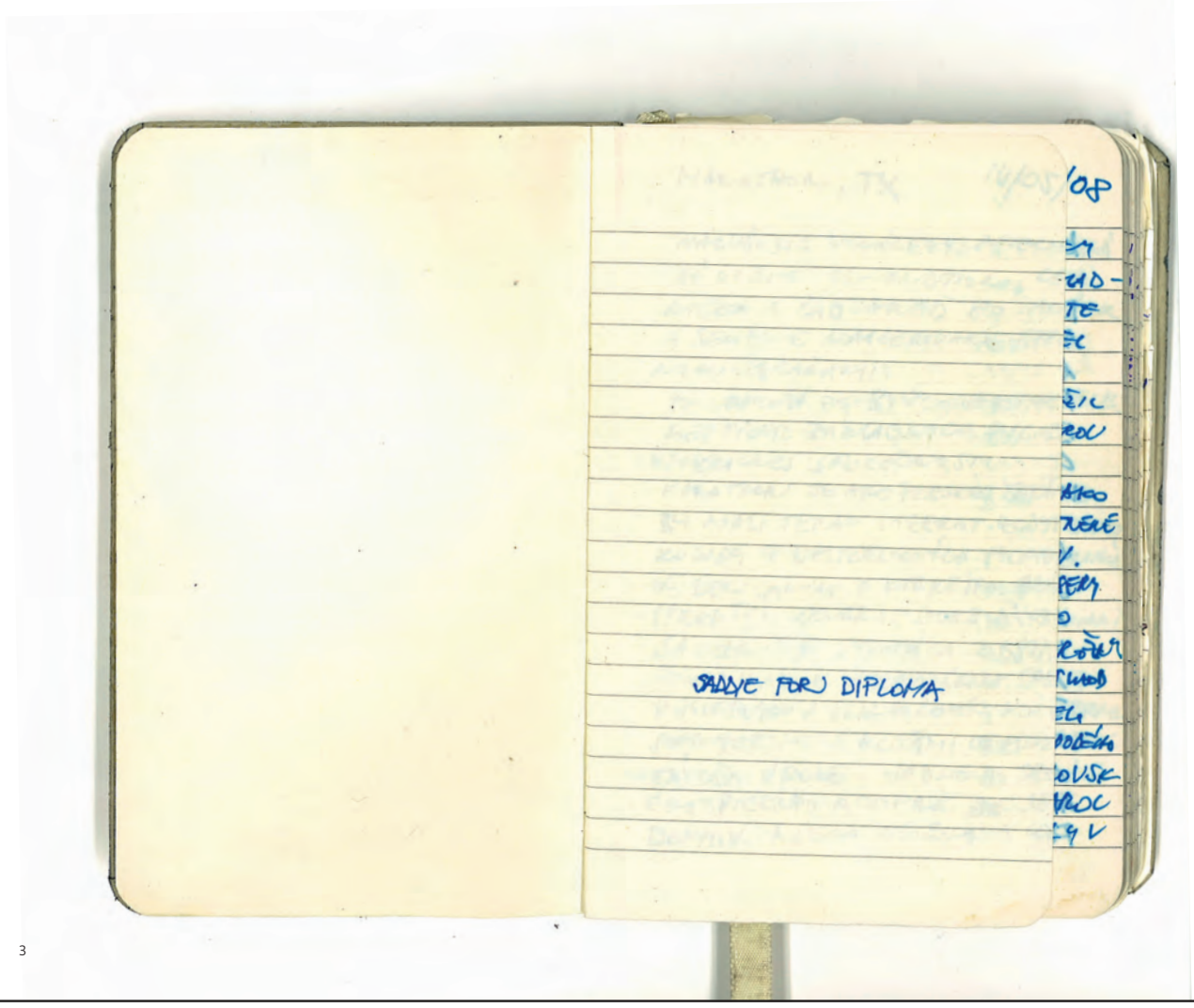
Book 1 – Analysis of History and Theory regarding cyberspace-architecture and cyberpunk
Book 2 – Vocabulary of cyberspace-architecture and cyberpunk
Book 3 – Notes and sketches
Book 4 – Scripts and codes
Book 5 – Presentation

Cyberpunk 問 Architecture

A humble attempt to give visual expression to new models of reality by Samo Bertok

Book 3 – Notes and Sketches




SKAVE FOR DIPLOMA

Carlsbad, TX 17/05/2008

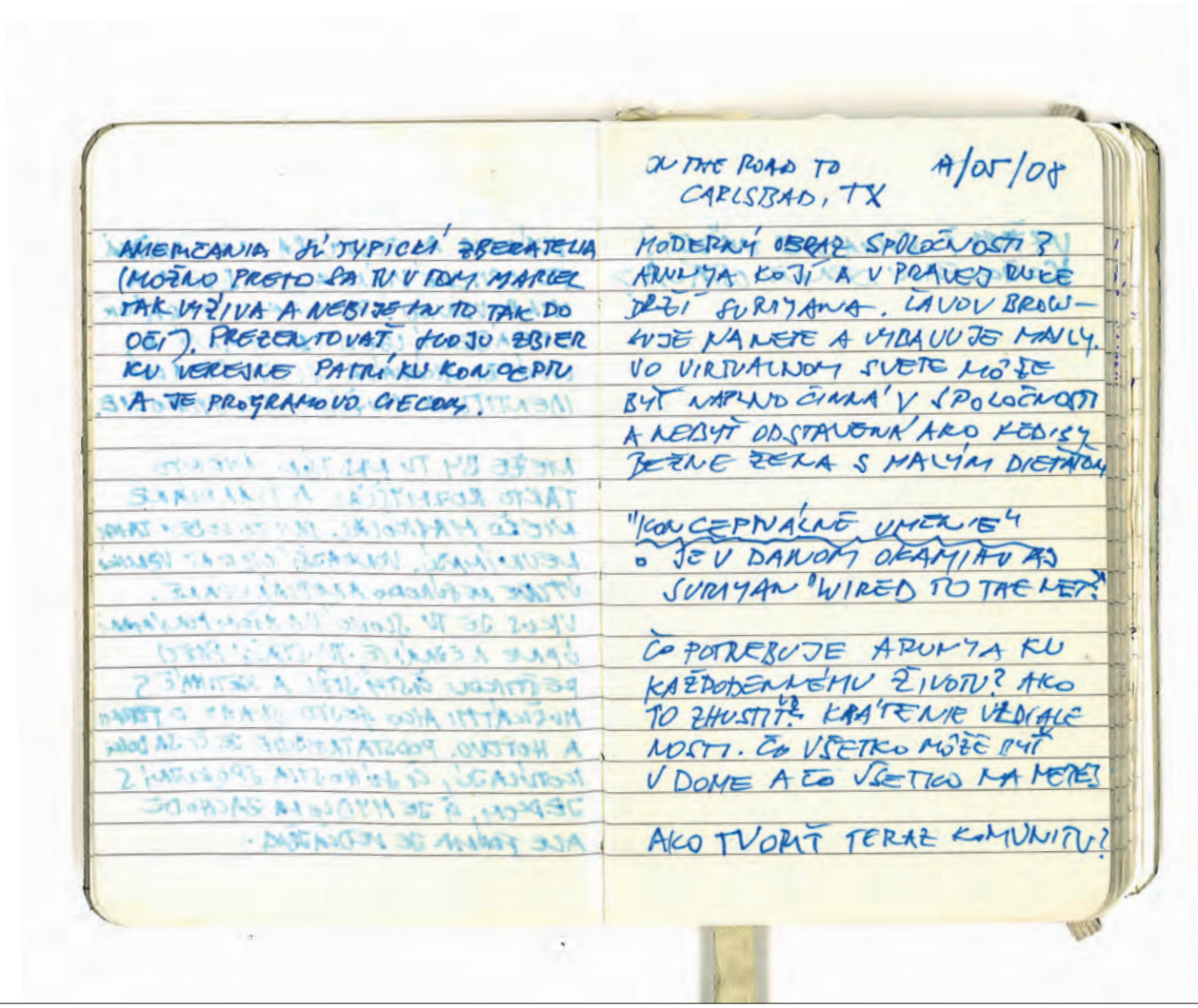
modern view of society? Arunya breast-feeds Suriyan holding him in her right hand. She uses her left hand for browsing the net, she looks through her mailbox. She can be fully active in the virtual world and society despite she is at home and has a newborn baby.

‚conceptual art' – is in this act Suriyan also ‚wired to the web'?

What does she need for every day life? How to compress it? Shortening of distances. What can she have in the house and what on the web?

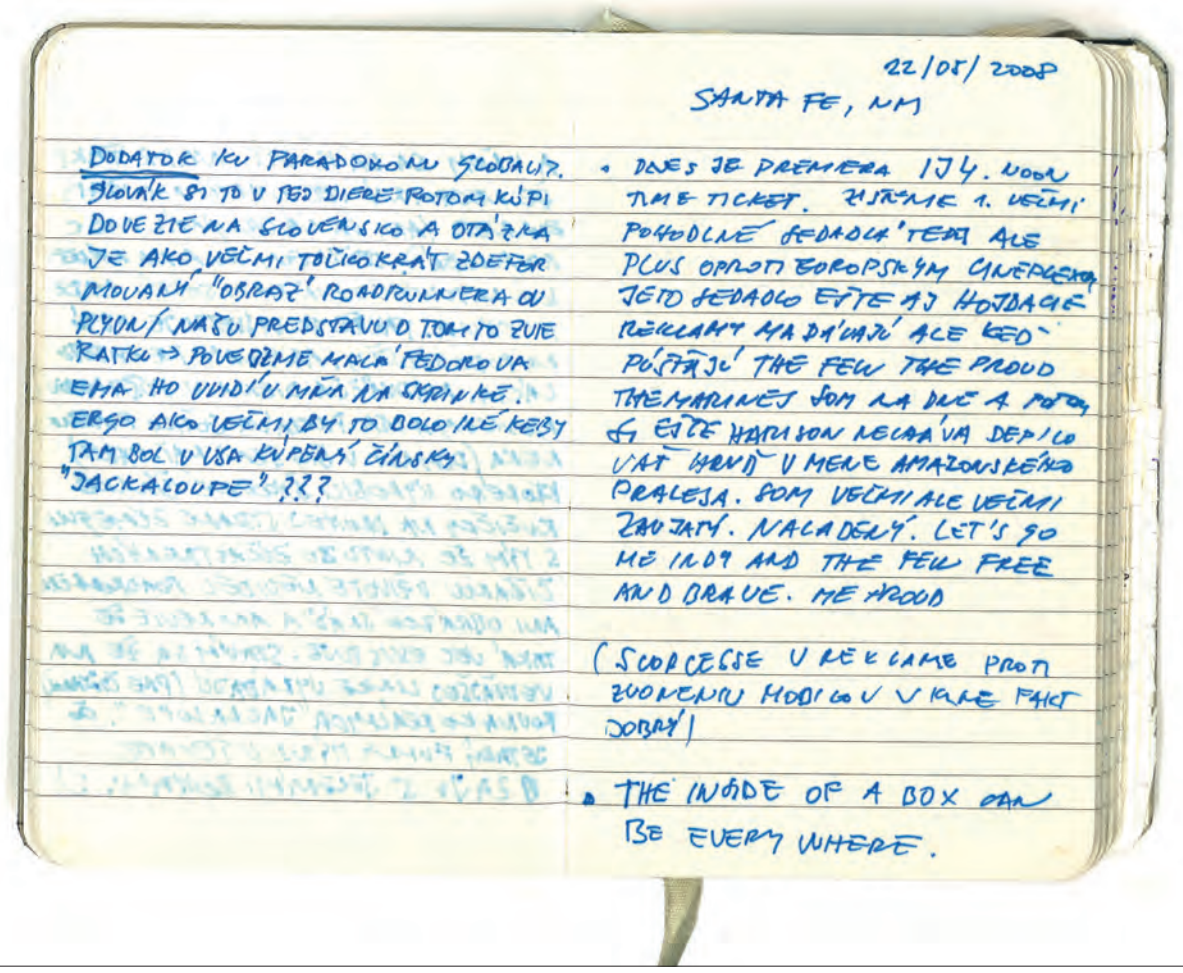
How is community nowadays created?

Relations? Social connections? How much time is left a day?



Santa Fe, NM 22/05/2008

the inside of a box can be everywhere



Vienna, AT 23/10/2008

we have never been modern

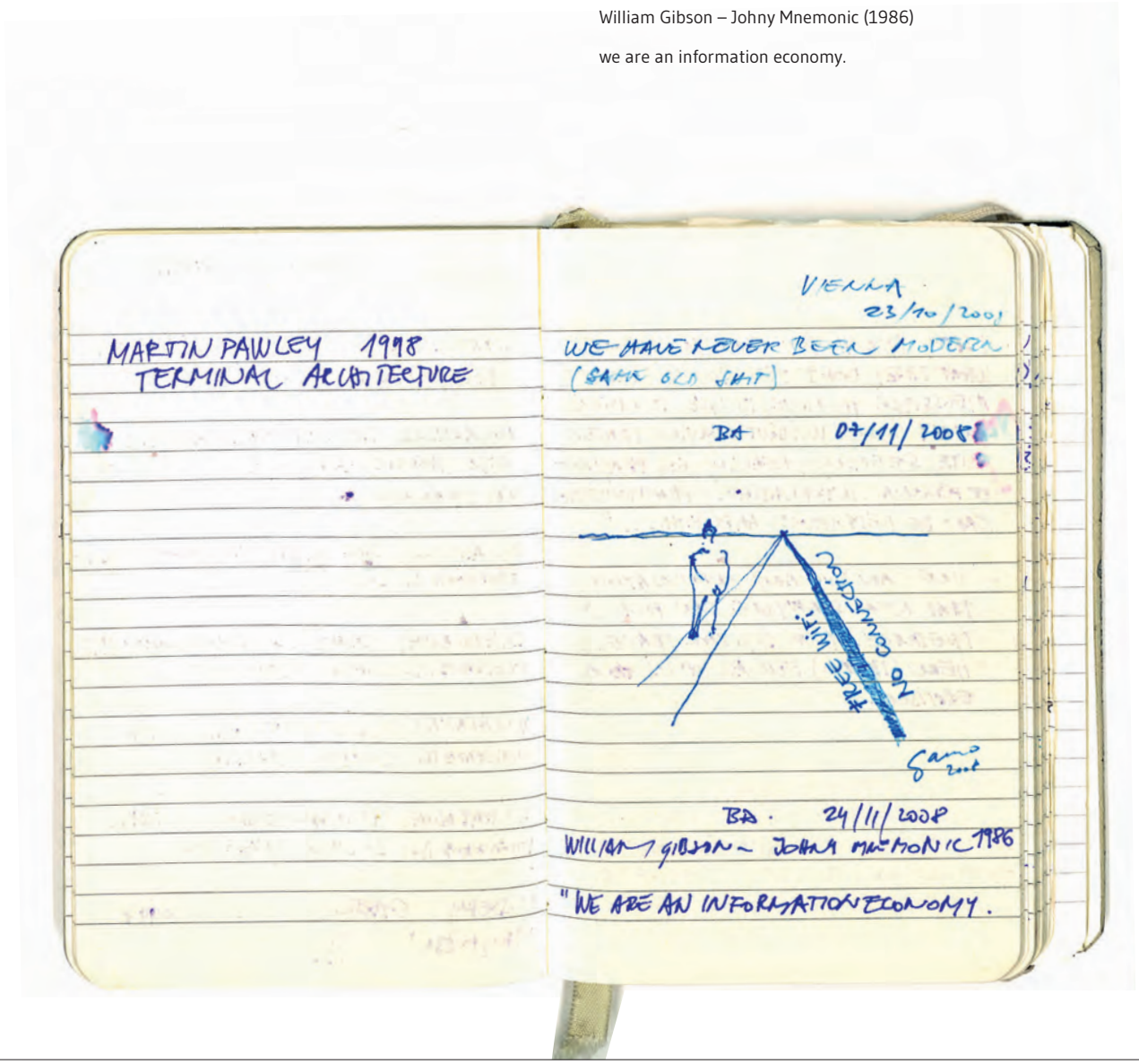
Bratislava, SK 07/11/2008

free wifi connection project

Bratislava, SK 24/11/2008

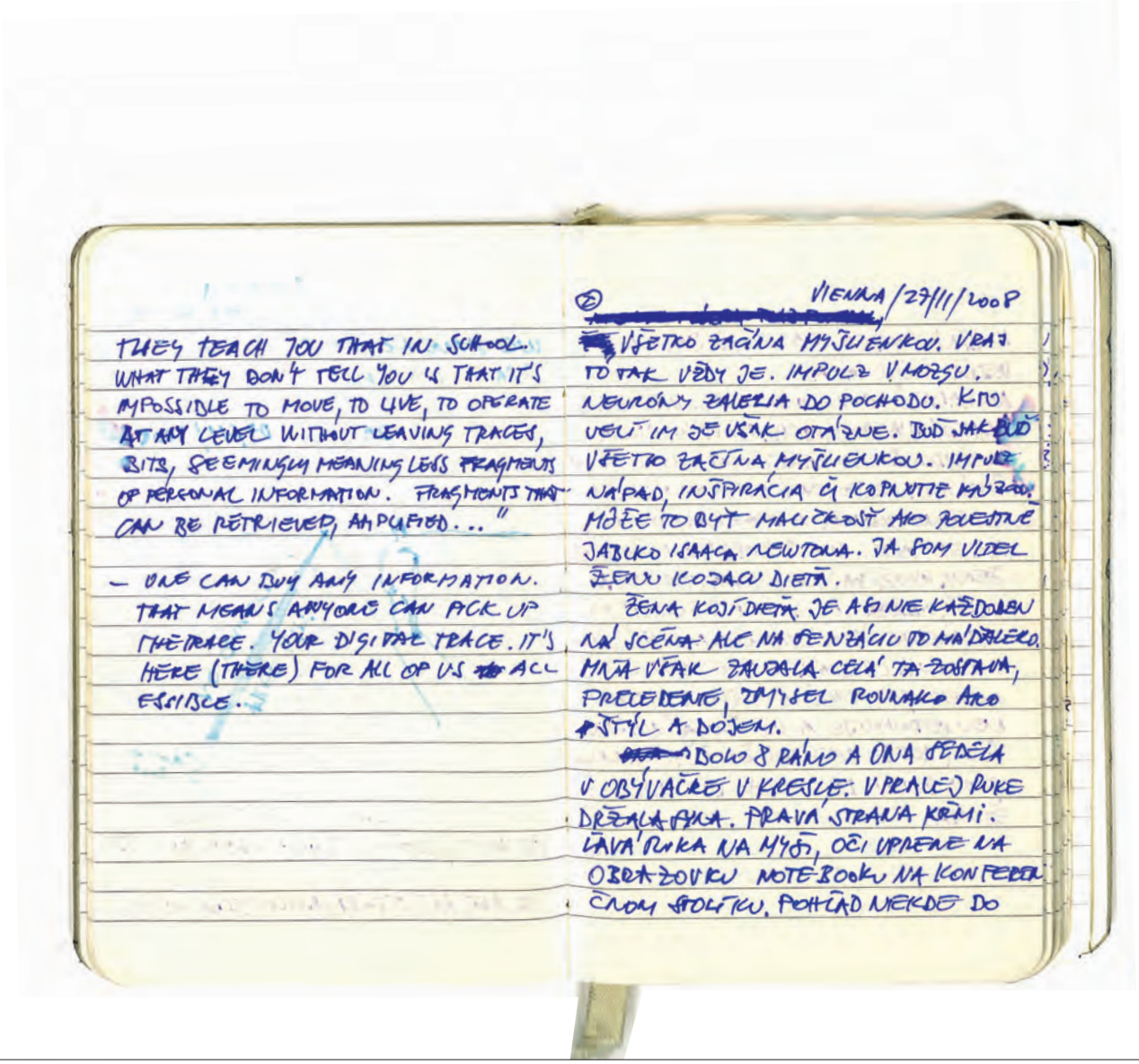
William Gibson – Johny Mnemonic (1986)

we are an information economy.



They teach you that in school. What they don't tell you is that it's impossible to move, to live, to operate at any level without leaving traces, bits, seemingly meaningless fragments of personal information. Fragments that can be retrieved, amplified…'

One can buy any information. That means anyone can pick up that trace. Your *digital trace*. It's here (there) for all of us accessible.

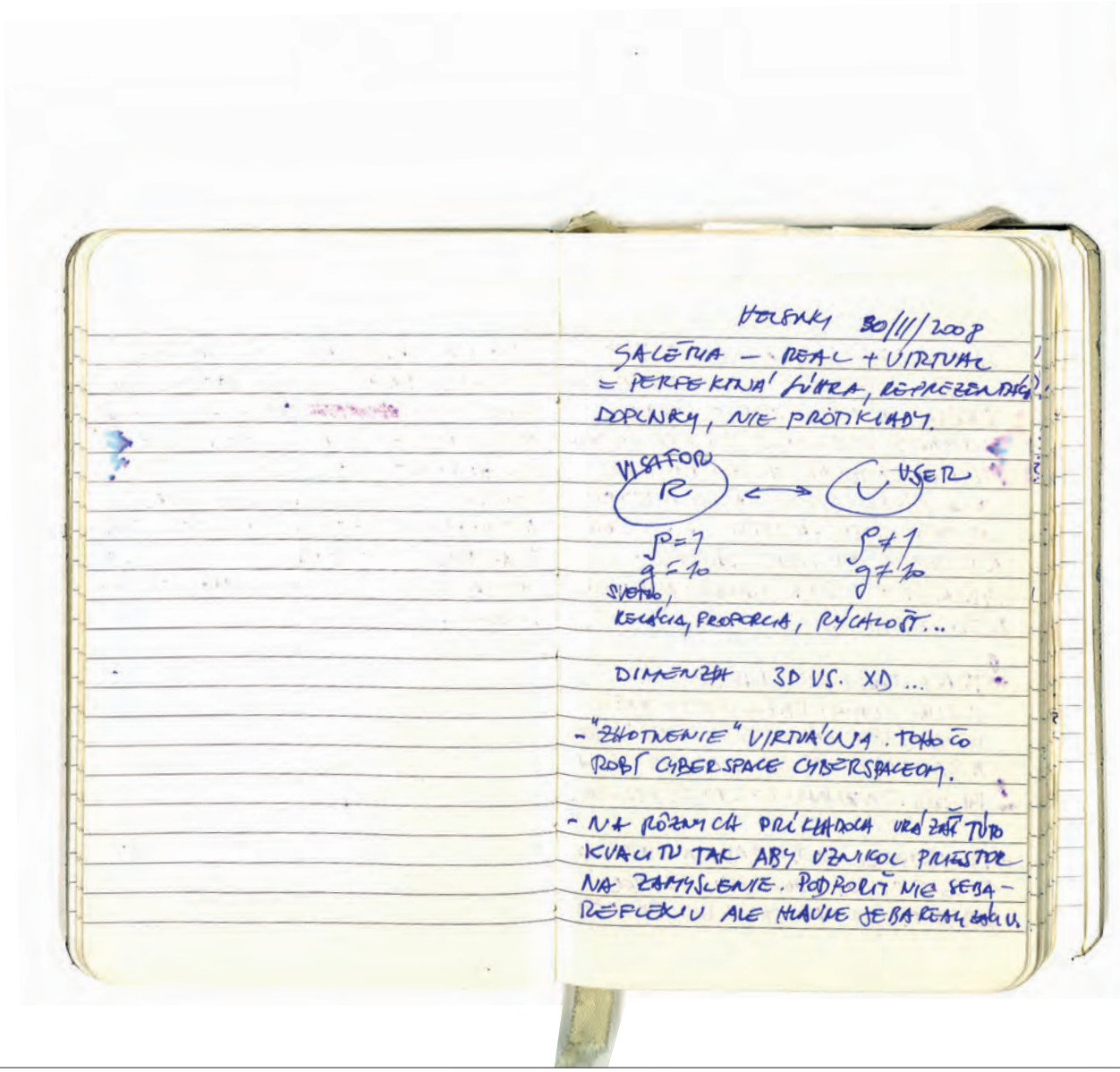


Helsinki, FI 30/11/2008

Gallery = real + virtual → perfect interplay, representation, complementary not contrariety

materialize of the virtual. Of what makes cyberspace.

Showing this quality on different examples in such way that the spectator experiences a space where he can re-think what it means cyberspace. Not to encourage one's self-reflection but self-realization.



The aim is not to show the user doing something in a right or wrong way but to show, materialize, represent and document on different aspects and forms what the user actually really does. I see myself retrospectively what I did or am doing and choose different scenarios to represent me most in that moment or simply create my own.

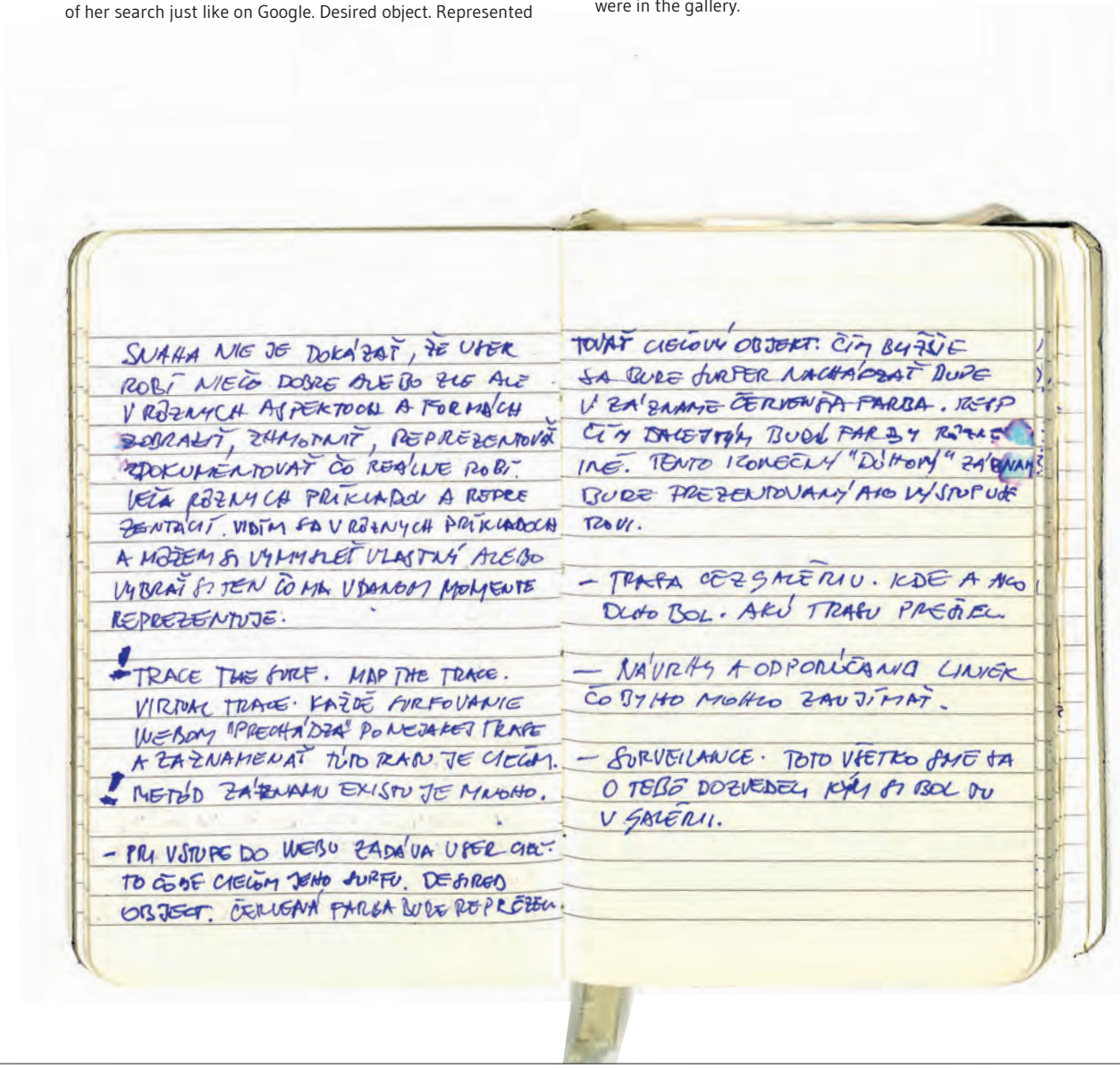
Trace the surf. Map the trace. Every surfing in the web is like *pathwalking* and to map ones path is the objective here. There are many methods to do so.

As soon as the user enters the space she writes down a target of her search just like on Google. Desired object. Represented by let's say red colour. The nearer he comes to this object the more red colour will there be in his record. The more away he strays the more will the colour be different from red. This final ‚rainbow' record he will obtain when he exits the space.

Path through the gallery. Where and for how long he stayed. Which path he took.

Links recommendations for him according to what he browsed and look at.

Surveillance. All of this we learned about you in the time you were in the gallery.



Helsinki, FI 01/12/2008

Ha-ha this is how science-fiction was imagined in 2008. Maros Krivy (on cyberspace)

Helsinki (airport), FI 01/12/2008

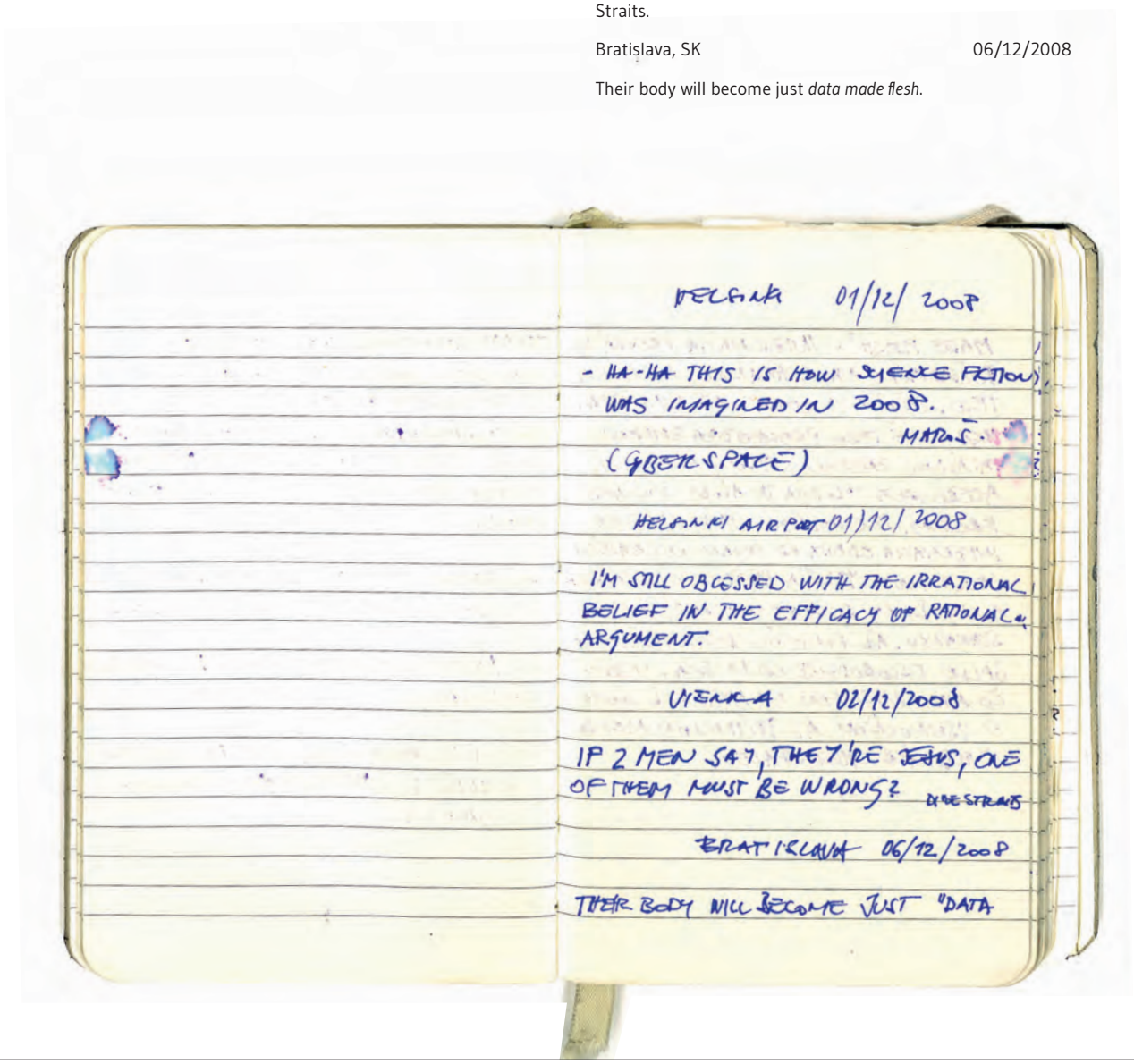
I'm still obsessed with the irrational belief in the efficacy of a rational argument.

Vienna, AT 02/12/2008

If 2 men say, they're Jesus, one of them must be wrong? Dire Straits.

Bratislava, SK 06/12/2008

Their body will become just *data made flesh*.



The information doesn't require an envelope, medium or *body*. It's beginning to exist outside of the medium. The organic body is going through a transformation. Fatal transformation. Metamorphosis. Cyborg as such, or the idea of a cyborg, is just some intermediary stage, in-between state where the information unleashes itself from organic matter, leaves it's *body* a transfers from the organic cell through a *bridge* towards an anorganic artificial shell. Only through digitalization it can totally free itself from the medium. This is what had W. Gibson in mind as he speaks about the almighty AI lurking somewhere in the vastness of the cyberspace.

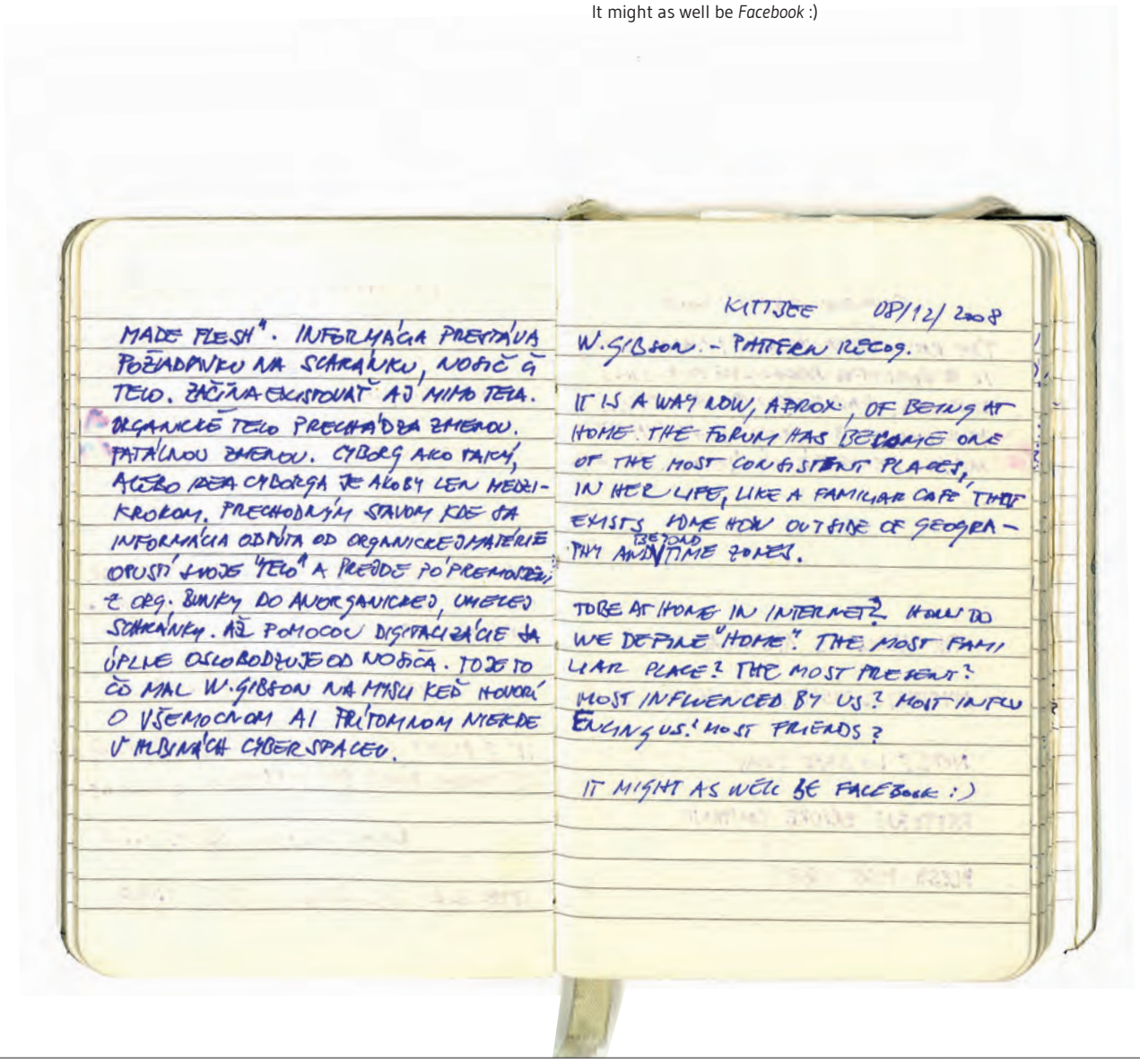
Kittsee, AT 08/12/2008

W. Gibson – Pattern recognition.

'... it is a way now, approx, of being at home. The Forum has become one of the most consistent places, in her life, like a familiar café that exists somehow outside geography and beyond time zones.'

To be at home in internet? How do we define *home*? The most familiar place? The most present? The most influenced by us? The most influencing us? Most friends?

It might as well be *Facebook* :)



Šenkvice, SK 24/12/2008

The key observation I made, which is sort of a premise for this diploma, has quite well been formulated by Greg Lynn himself, in his newest book *Form*. He says...

the world is artificial

mystery is the very root of cool secrets

nerds do make jokes

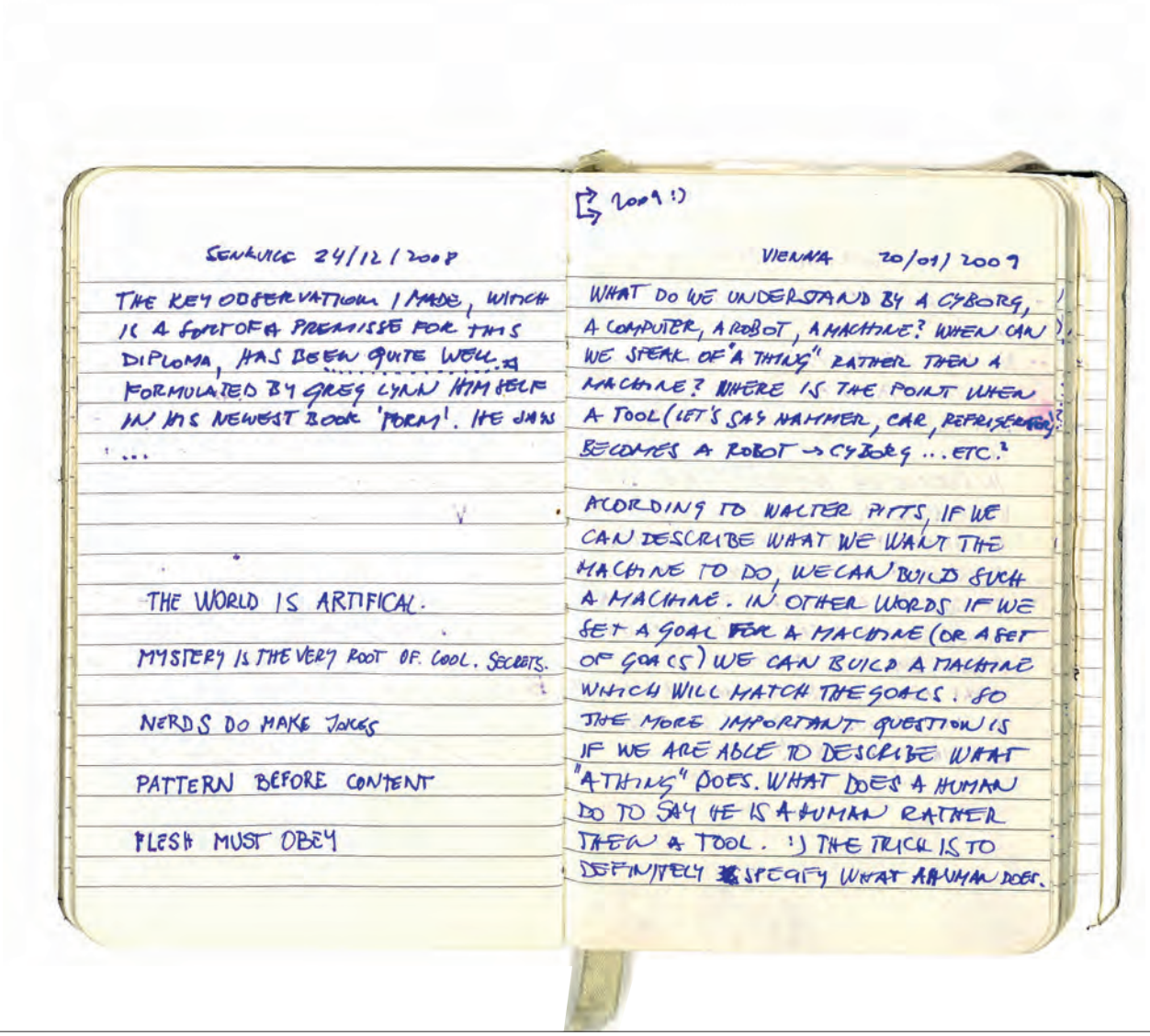
pattern before content

flesh must obey

Vienna, AT 20/01/2009

What do we understand by a cyborg, a computer, a robot, a machine? When can we speak of a *thing* rather than a machine? Where is the point where a tool (let's say hammer, car, refrigerator) becomes a robot → cyborg ... etc?

According to Walter Pitts, if we can describe what we want a machine to do, we can build such a machine. In other words if we set a goal for a machine (or a set of goals) we can build a machine which will match the goals. So the more important question is if we are able to describe what a *thing* does. What does a human do to say he is human rather than a tool. The trick is to definitely specify what a human does.



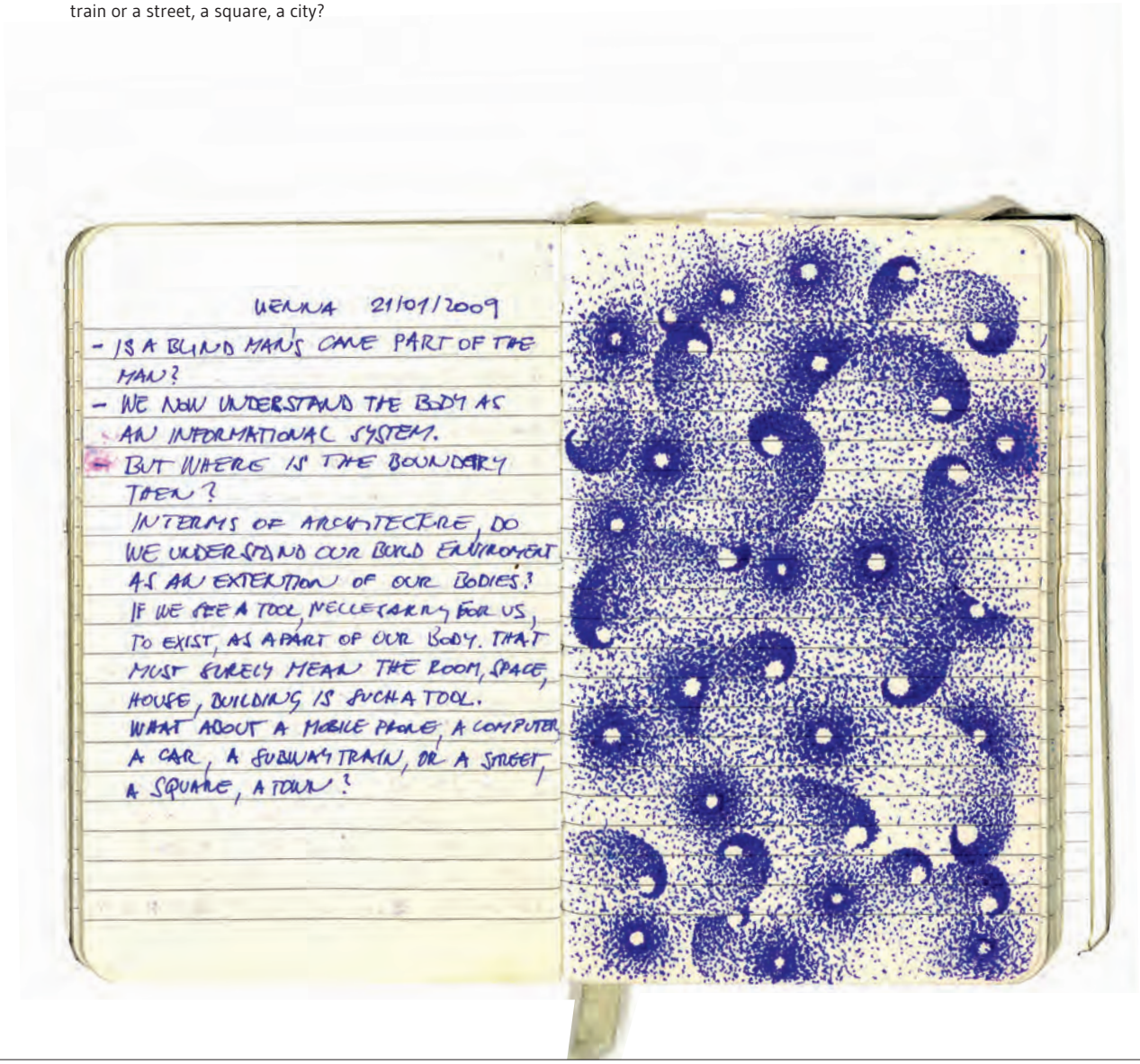
Vienna, AT 21/01/2009

Is a blind man's cane part of the man?

We now understand the body as an informational system.

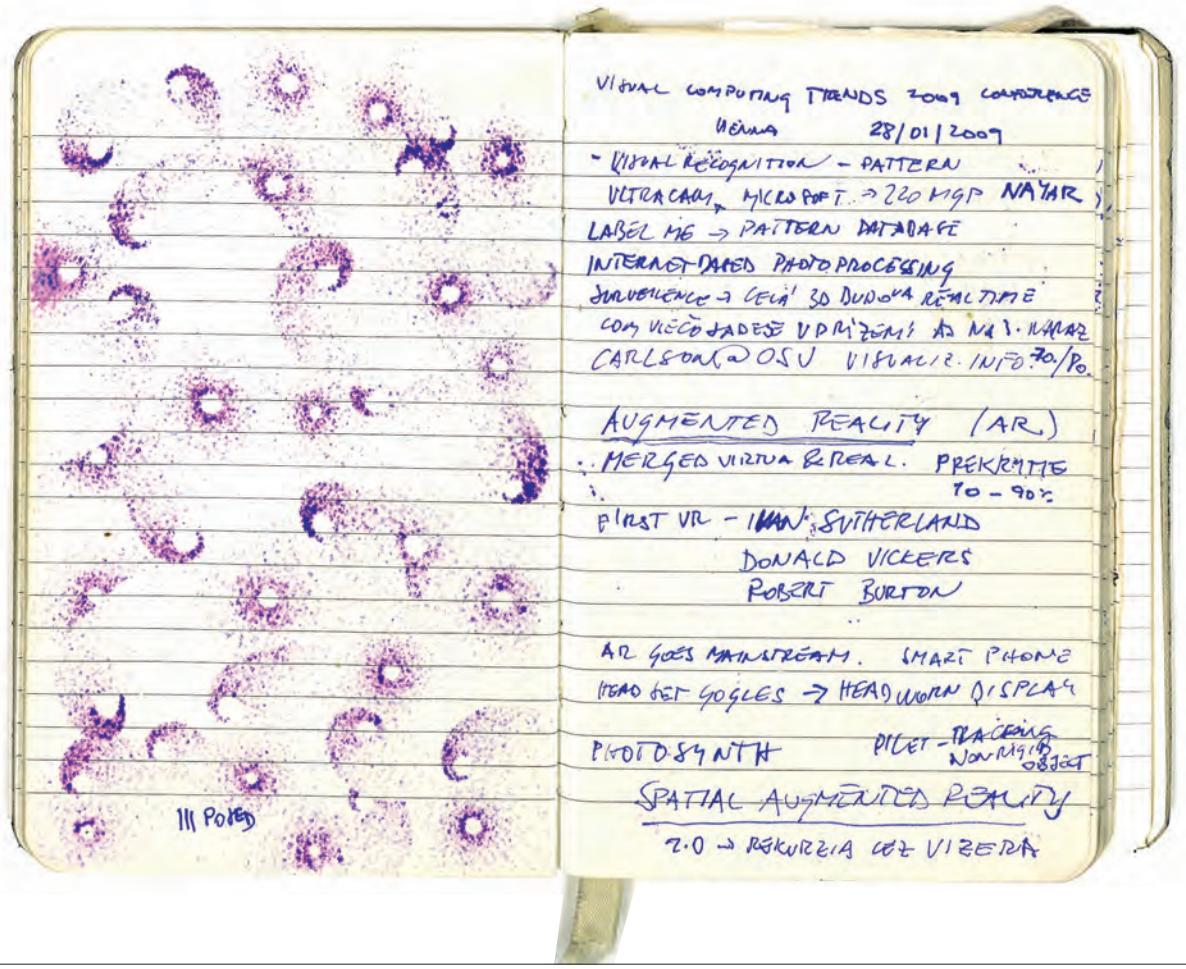
But where is the boundary then?

In terms of architecture, do we understand our build environment as an extension of our bodies? If we see a tool necessary for us, to exist, as a part of our body. That must surely mean a room, a space, a house or a building is such a tool. What about a mobile phone, a computer, a car, a subway train or a street, a square, a city?



Vienna, AT 28/01/2009

Visual Computing Trends 2009 conference



Vienna, AT 11/02/2009

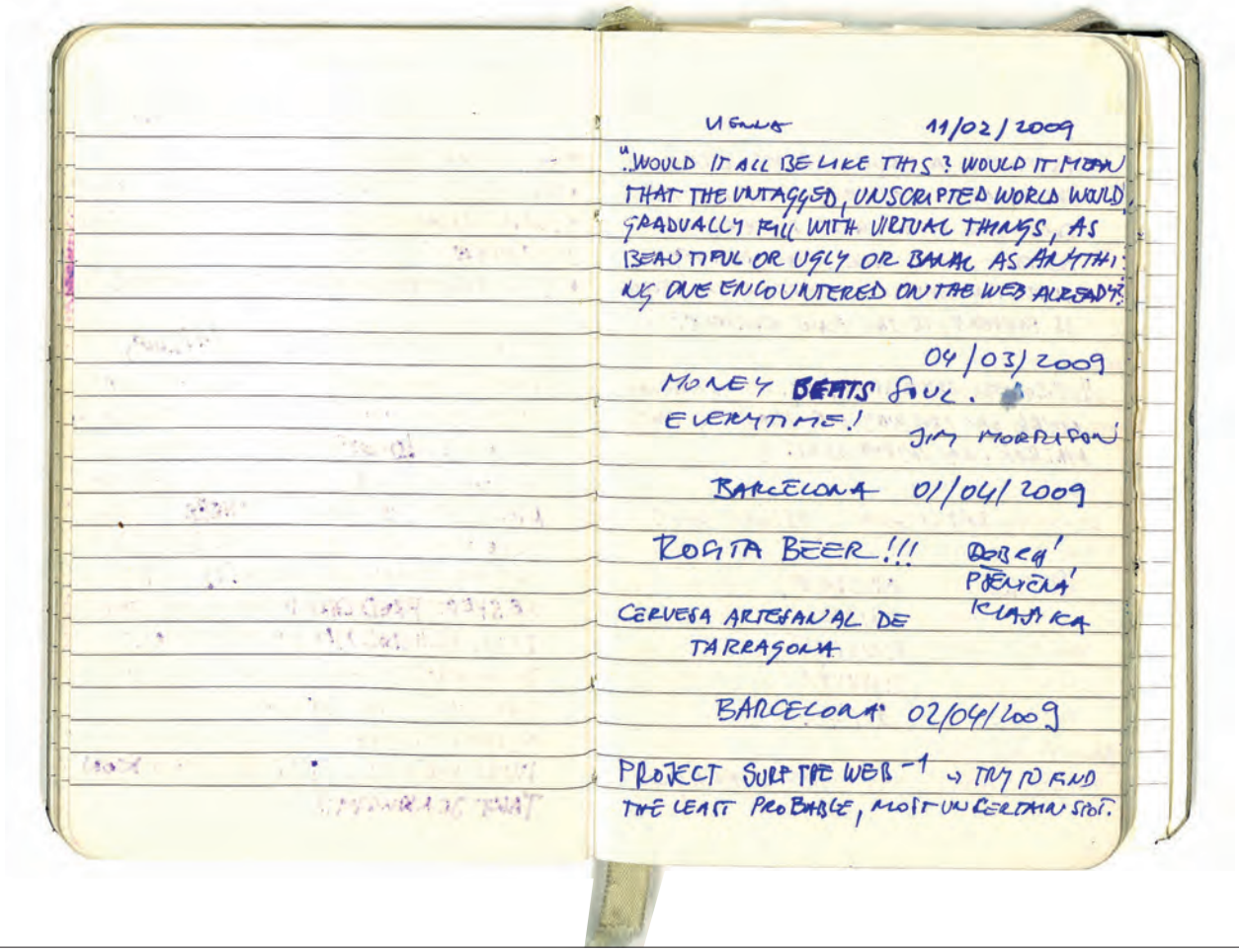
‚Would it all be like this? Would it mean that the untagged, unscripted world would gradually fill with virtual things, as beautiful or ugly or banal as anything one encountered on the web already?

Vienna, AT 04/03/2009

Money beats soul. Every time. Jim Morrison.

Barcelona, ES 02/04/2009

Project surf web-[1] → try to find the least probable, most uncertain spot.



Is there a page, a place with no or almost no hints to it? How does one find those places? How do they look like? How do they fit into the matrix? Where no connection is present, is the place still existent (in the web)?

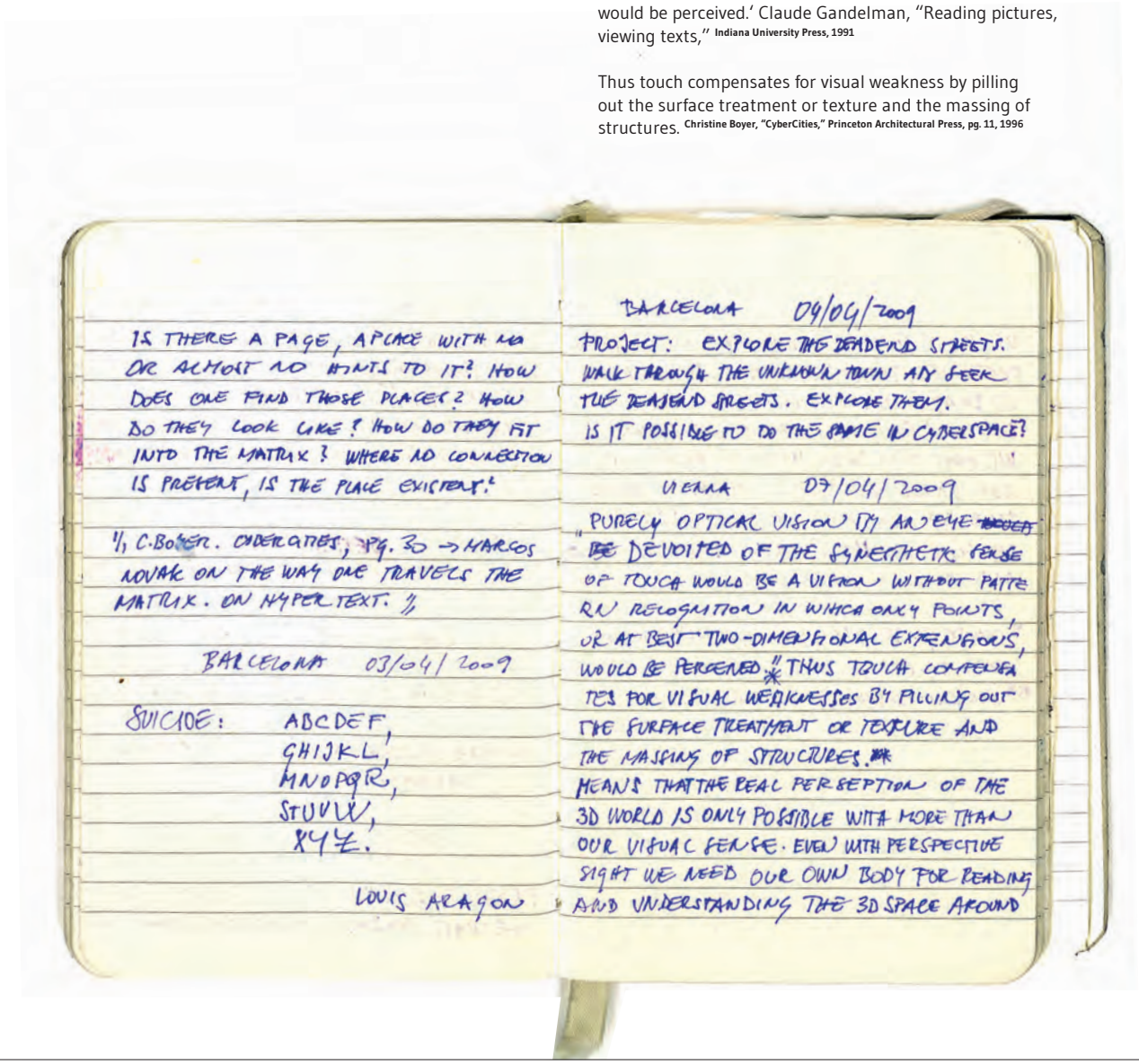
Barcelona, ES 04/04/2009

Project: explore the dead-end streets. Walk through the unknown town and look for the dead-end streets. Explore them. Is it possible to do the same in cyberspace?

Vienna, AT 07/04/2009

‚Purely optical vision by an eye stripped of the synesthetic sense of touch would be a vision without pattern recognition in which only points or at best 2-dimensional extensions, would be perceived.' Claude Gandelman, "Reading pictures, viewing texts," Indiana University Press, 1991

Thus touch compensates for visual weakness by pilling out the surface treatment or texture and the massing of structures. Christine Boyer, "CyberCities," Princeton Architectural Press, pg. 11, 1996



Means that the real perception of the 3D world is only possible with more than our visual sense. Even with perspective sight we need our own body for reading and understanding the 3D space around us, be it for the understanding of depth, size, relation or presence or simply to be able to perceive hardness or softness in sense of spatial statements. Without our body in relation of the world we *see*, we can't understand concept of time. Is that the point of diversification between human and animal organism?

Bratislava, SK 11/04/2009

Facebook project 01

read the notes about a user, but see no photo. Things like: I like cats. I'm a fan of Abba. I like T-shirts. My favourite movies are... I love my mum.

Now try to describe the person in your imagination.

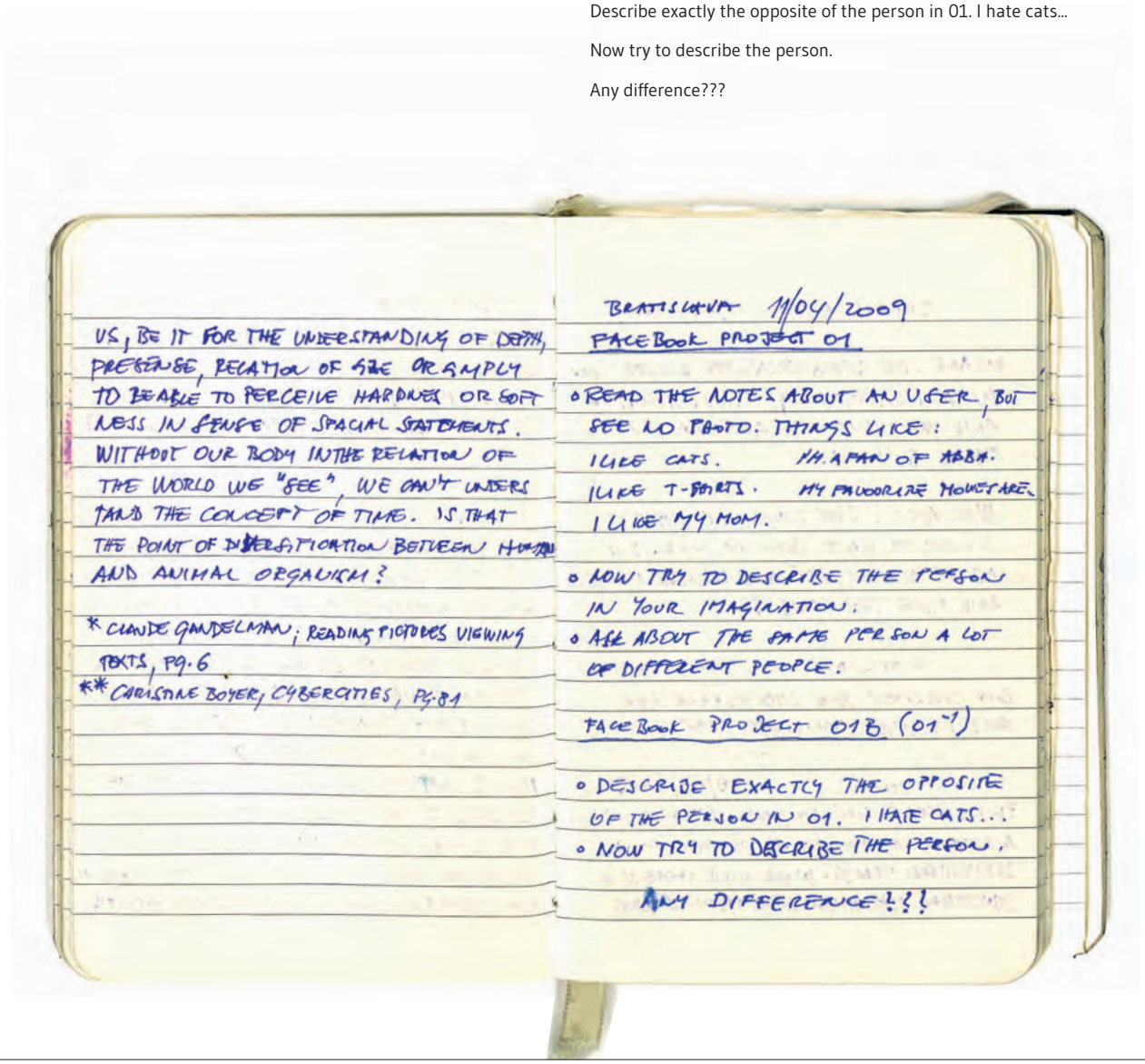
Ask about the same person a lot of different people.

Facebook project 01B

Describe exactly the opposite of the person in 01. I hate cats...

Now try to describe the person.

Any difference???



Bratislava, SK 11/04/2009

We are the dematerialized bodies living in the disappearing cities. Identity and individuality are rare artefacts by now.

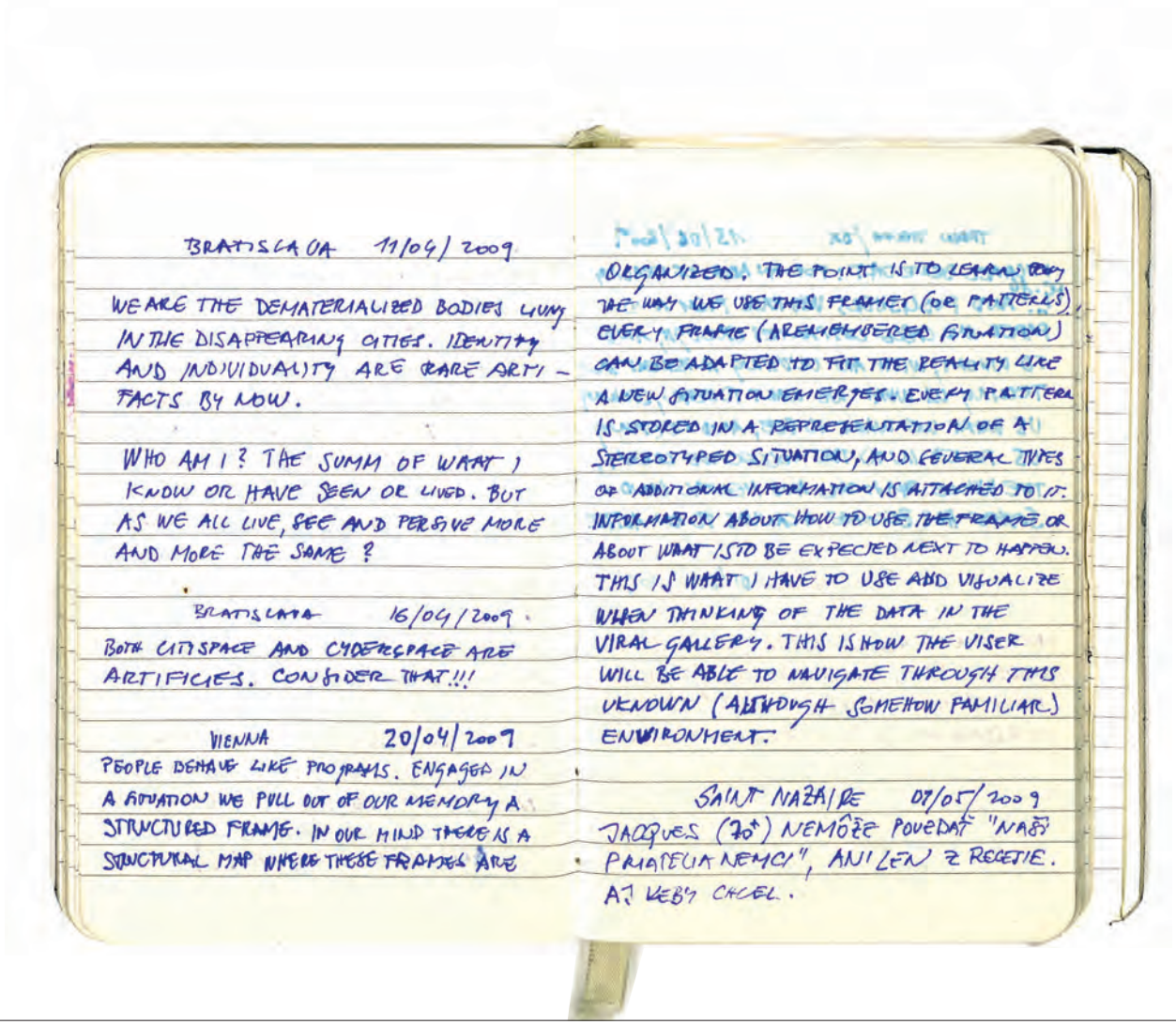
Who am I? The sum of what I know or have seen or lived. But as we all live, see and perceive more and more the same?

Bratislava, SK 16/04/2009

Both cityspace and cyberspace are artifices. Consider that!!!

Vienna, AT 20/04/2009

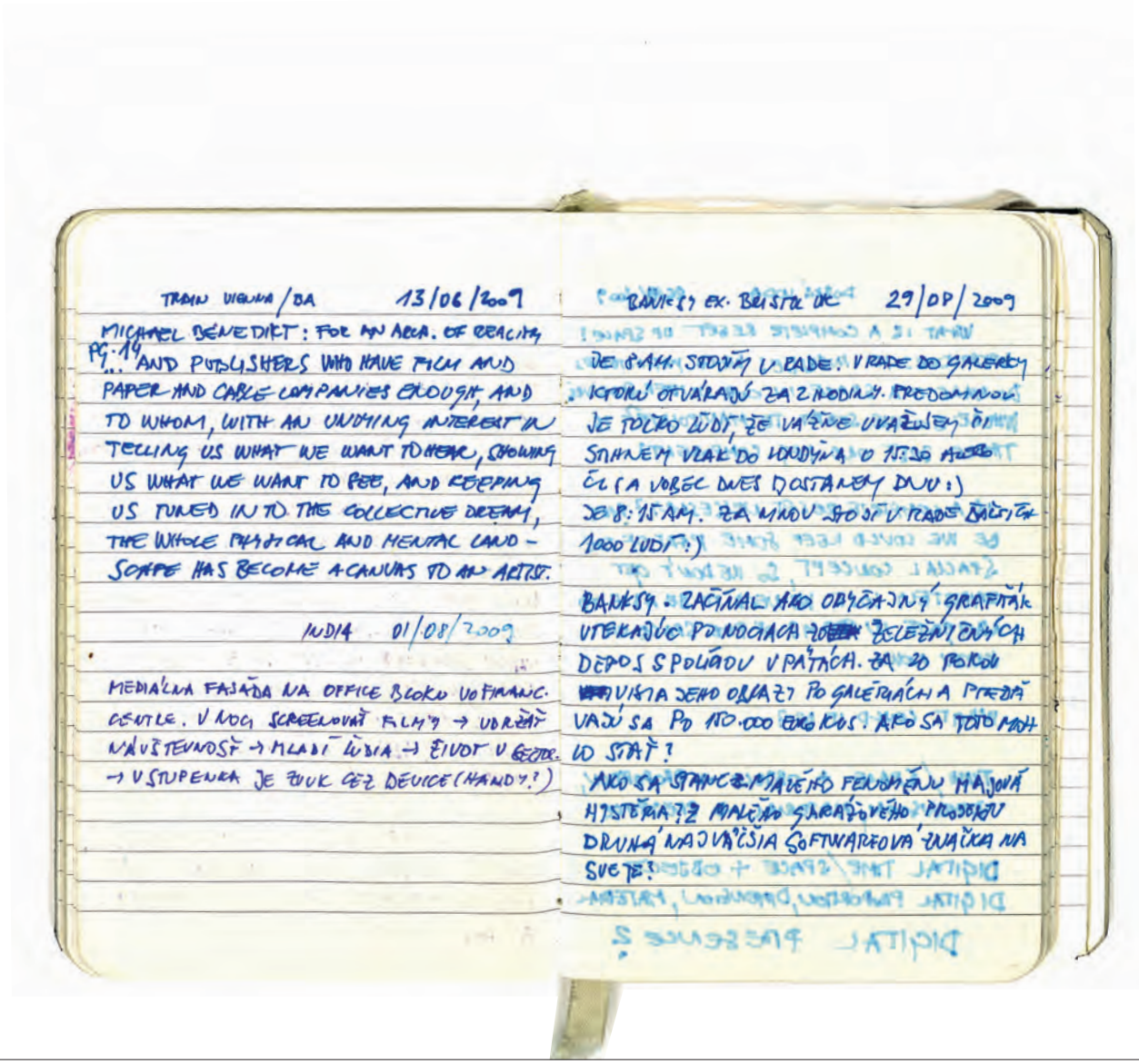
People behave like programs. Engaged in a situation we pull out of our memory a structured frame. In our mind there is a structural map where these frames are organised. The point is to learn from the way we use these frames (or patterns?). Every frame (a remembered situation) can be adapted to fit the reality alike when a new situation emerges. Every pattern is stored in a representation of a stereotyped situation and several types of additional information is attached to it. Information about how to use the frame or about what is to be expected next to happen. This is what I have to use and visualize when thinking of the data in the virtual gallery. This is how the viser will be able to navigate through this unknown (although somehow familiar) environment.



Train Vienna – Bratilava, AT 13/06/2009

Michael Benedikt: For an Architecture of reality, pg. 14

...and publishers who have film and paper and cable companies enough, and to whom, with an undying interest in telling us what we want to hear, showing us what we want to see, and keeping us tuned into the collective dream, the whole physical and mental landscape has become a canvas to an artist.



Dobrá Voda, SK 26/08/2009

What is a complete reset of space? What does it include? Which parameters do make it a space in our understanding? Where do we start to manipulate? Is there just one key component?

Is a complete reset necessary? Maybe we could keep some parts of the spacial concept, so we don't get completely lost. We need to be able to navigate in terms of the space we know now.

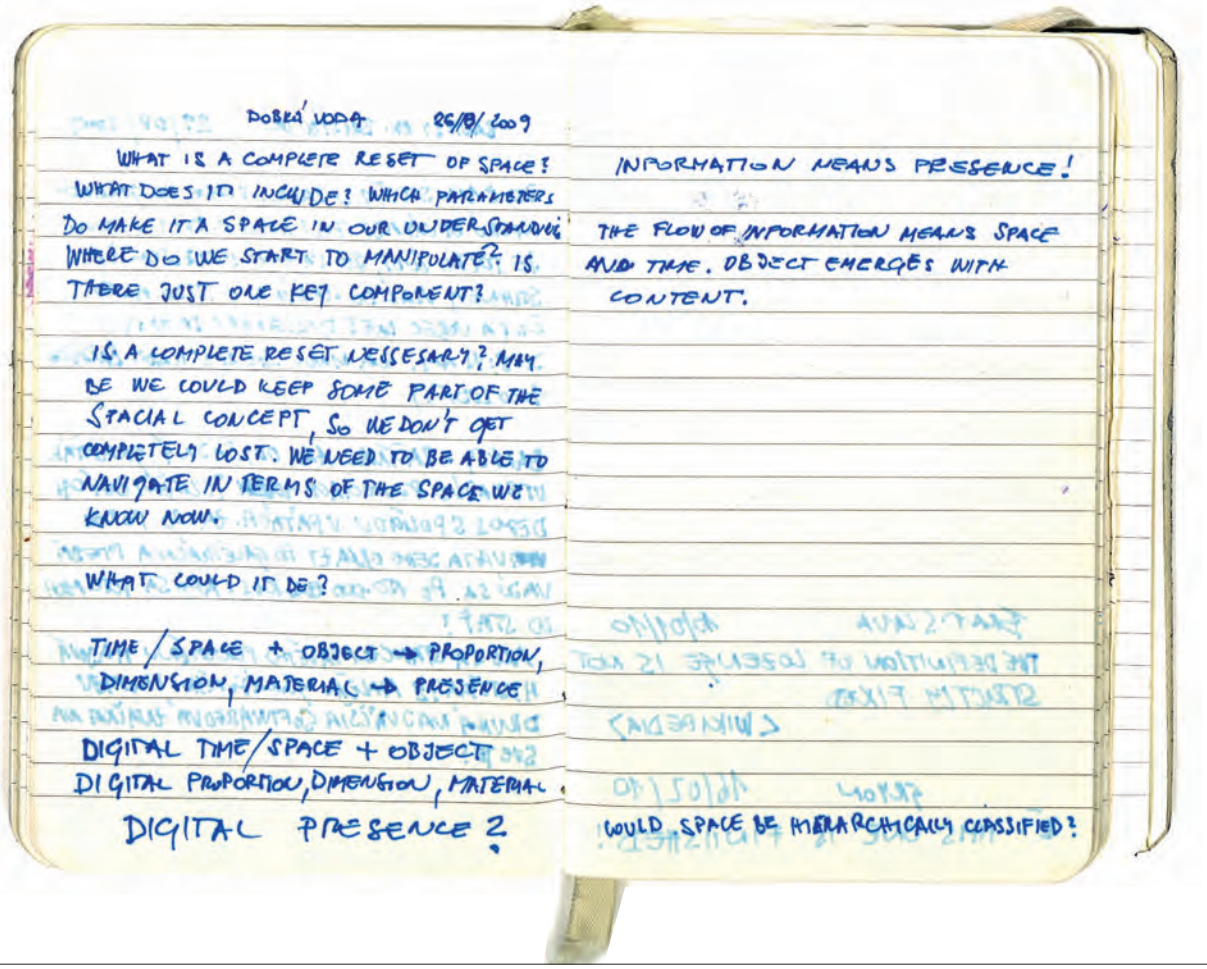
What could it be?

time/space + object → proportion, dimension, material → presence

digital time/space + object → digital proportion, dimension, material → digital presence?

Information means presence!

The flow of information means space and time. Object emerges with content. Would space be hierarchically classified?



Famara beach/Lanzarote, ES 16/10/2009

Cybergallery features: will be possible to vid. The desired object from any angle, store the vid and take it.

Concept book: every double page will be a keyword or keynote or idea. Text 20-30 words. Picture. 20-30 pages.

Optionally: 4-6 pages about the gallery as a possibility of practical use (practical diploma)

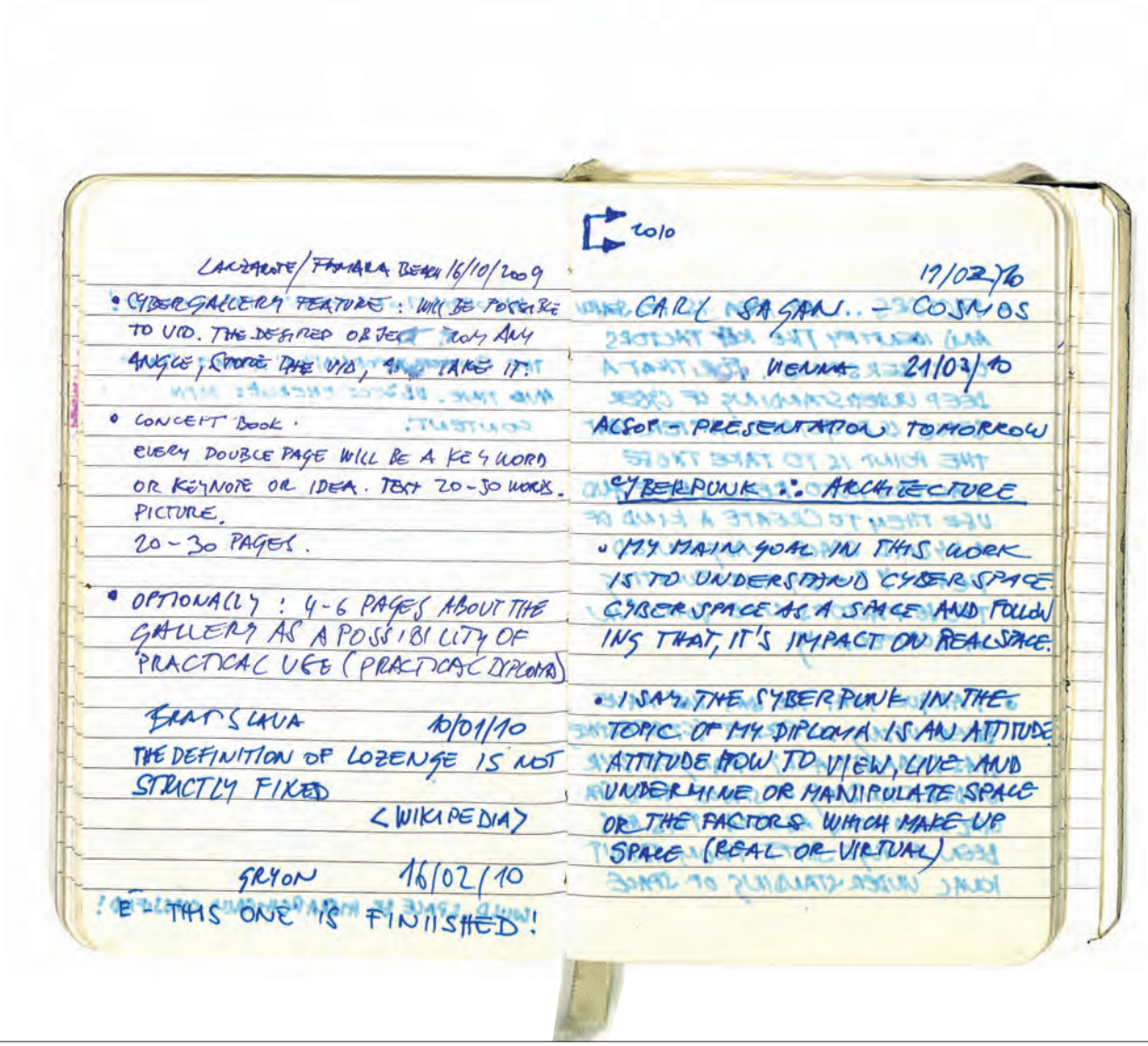
Vienna, AT 21/03/2010

Alsop-presentation tomorrow.

Cyberpunk Architecture

My main goal in this work is to understand cyberspace. Cyberspace as a space and following that, it's impact on *real* space.

I say the cyberpunk in the topic of my diploma is an attitude. Attitude how to view, live and undermine or manipulate space or the factors which make up space (real or virtual)



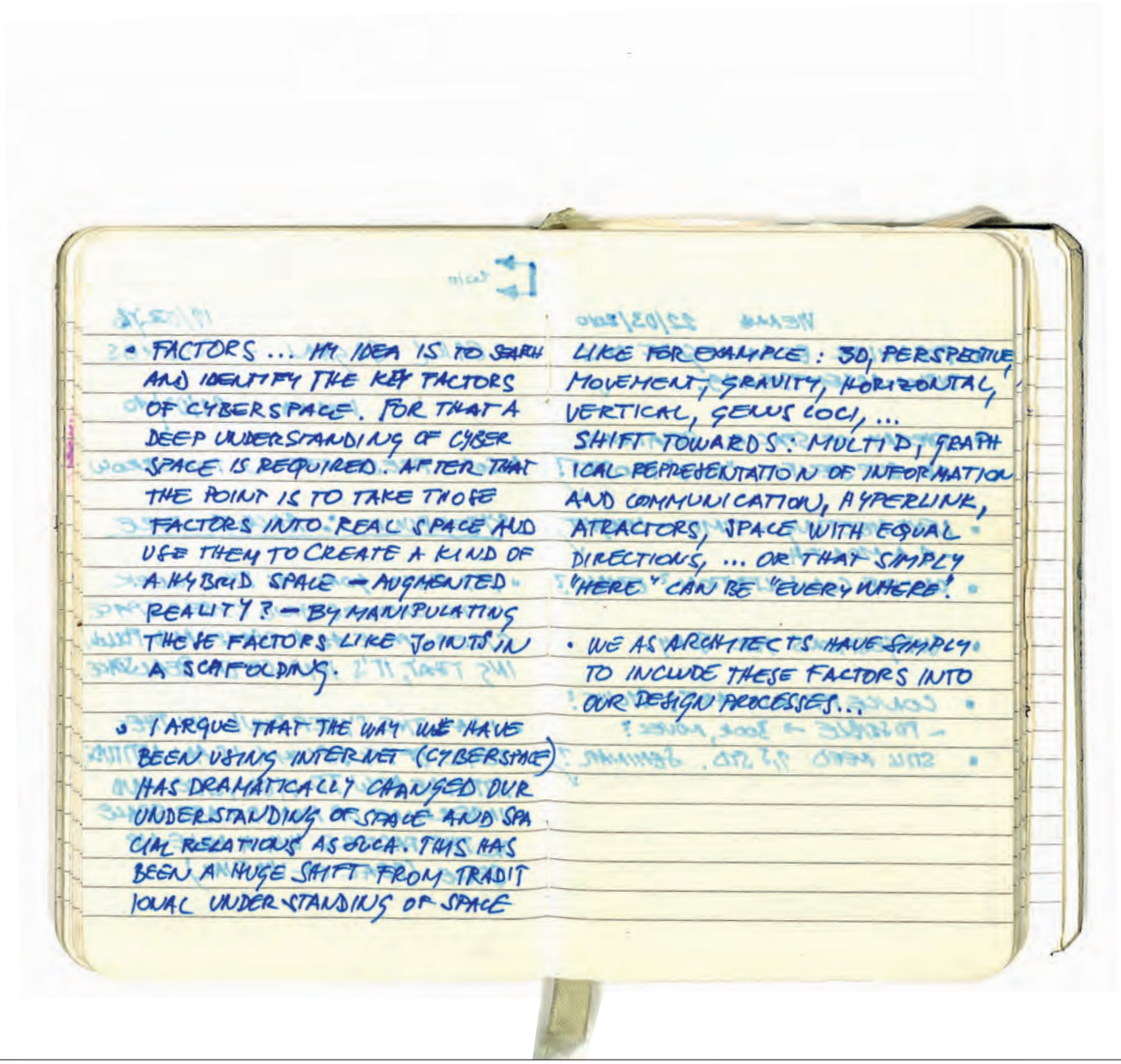
Factors ... my idea is to search and identify the key factors of cyberspace. For that a deep understanding of cyberspace is required. After that the point is to take those factors into real space and use them to create a kind of a hybrid space – augmented reality? – by manipulating these factors like joints in a scaffolding.

I argue that the way we have been using internet (cyberspace) has dramatically changed our understanding of space and spacial relations as such. This has been a huge shift from traditional understanding of space.

Like for example: 3D, perspective, movement, gravity, horizontal, vertical, genius loci, ...

Shift towards: multiD, graphical representation of information and communication, hyperlink, attractors, space with equal directions, ... or that simply *here* can be *everywhere*.

We as architects simply have to include these factors into our design processes...



Vienna, AT 22/03/2010

1. Session Alsop:

I would like to be more specific.

Ubud, Bali, ID 02/05/2010

Skp: 3D graph of the net from the cyber-gallery, different people, different colours

Ubud, Bali, ID 03/05/2010

"Antiprakriti" as opposed to the material nature.

Augmented installations

main: ants, termites eating the columns
spider, web
monkey's playing
starwars battle

top: huge fish floating
chess game
giant wave tubing forever

screen: inside out!

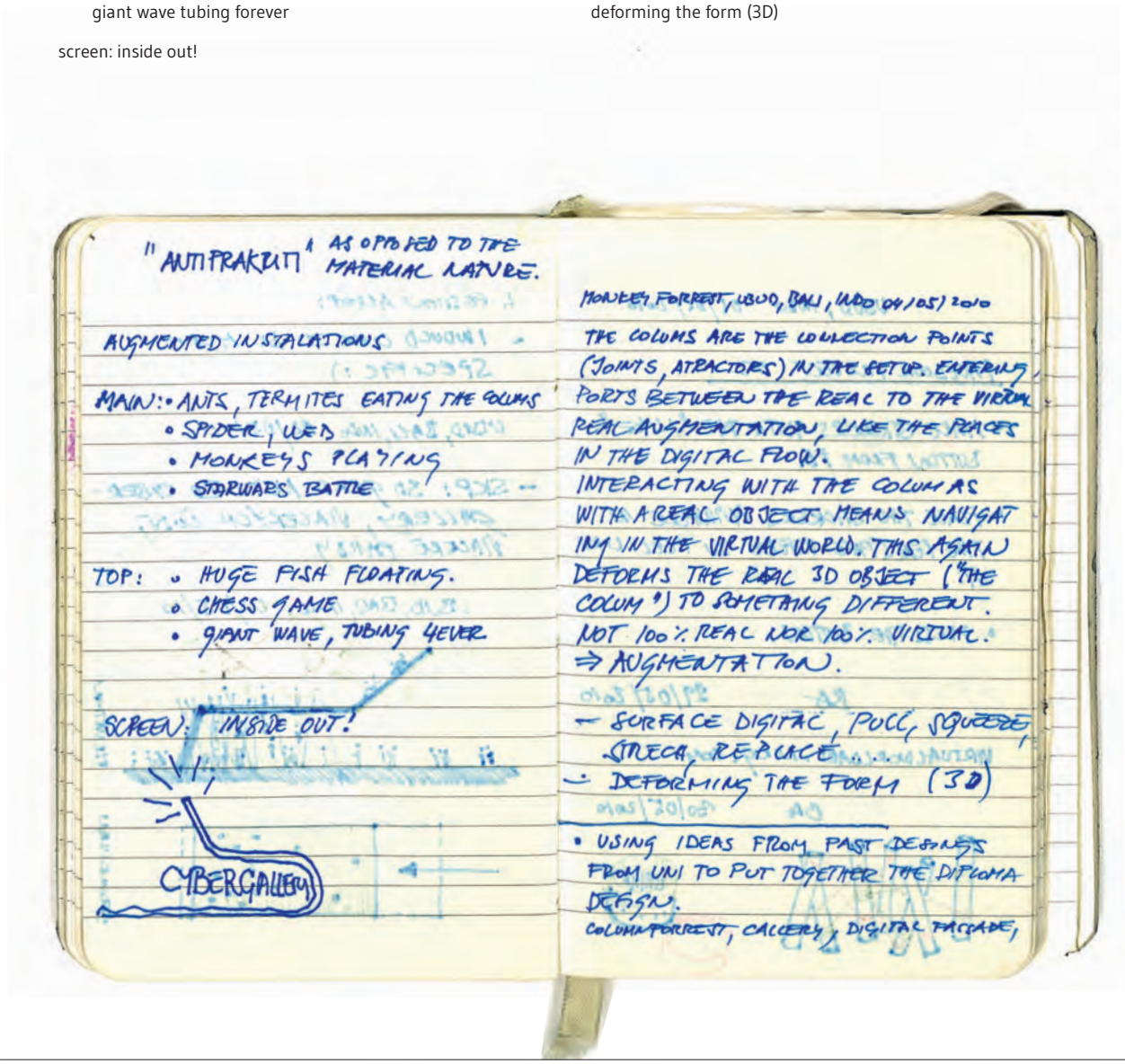
Monkey Forrest, Ubud, Bali, ID 04/05/2010

The columns are the connection points (joints, attractors) in the set up. Entering ports between the real and thee virtual. Real augmentation, like the places in the digital flow.

Interacting with the columns as with a real object means navigating in the virtual world. This again deforms the real 3D object (the column) to something different. Not 100% real nor 100% virtual → augmentation

digital surface, pull, squeeze, stretch, replace...

deforming the form (3D)



Ubud, Bali, ID 05/05/2010

Facebook project 02

Make stickers with the ‚I like' button from FB.

Place the stickers anywhere in the real space where you feel like it.

Press the button.

Bratislava, SK 30/05/2010

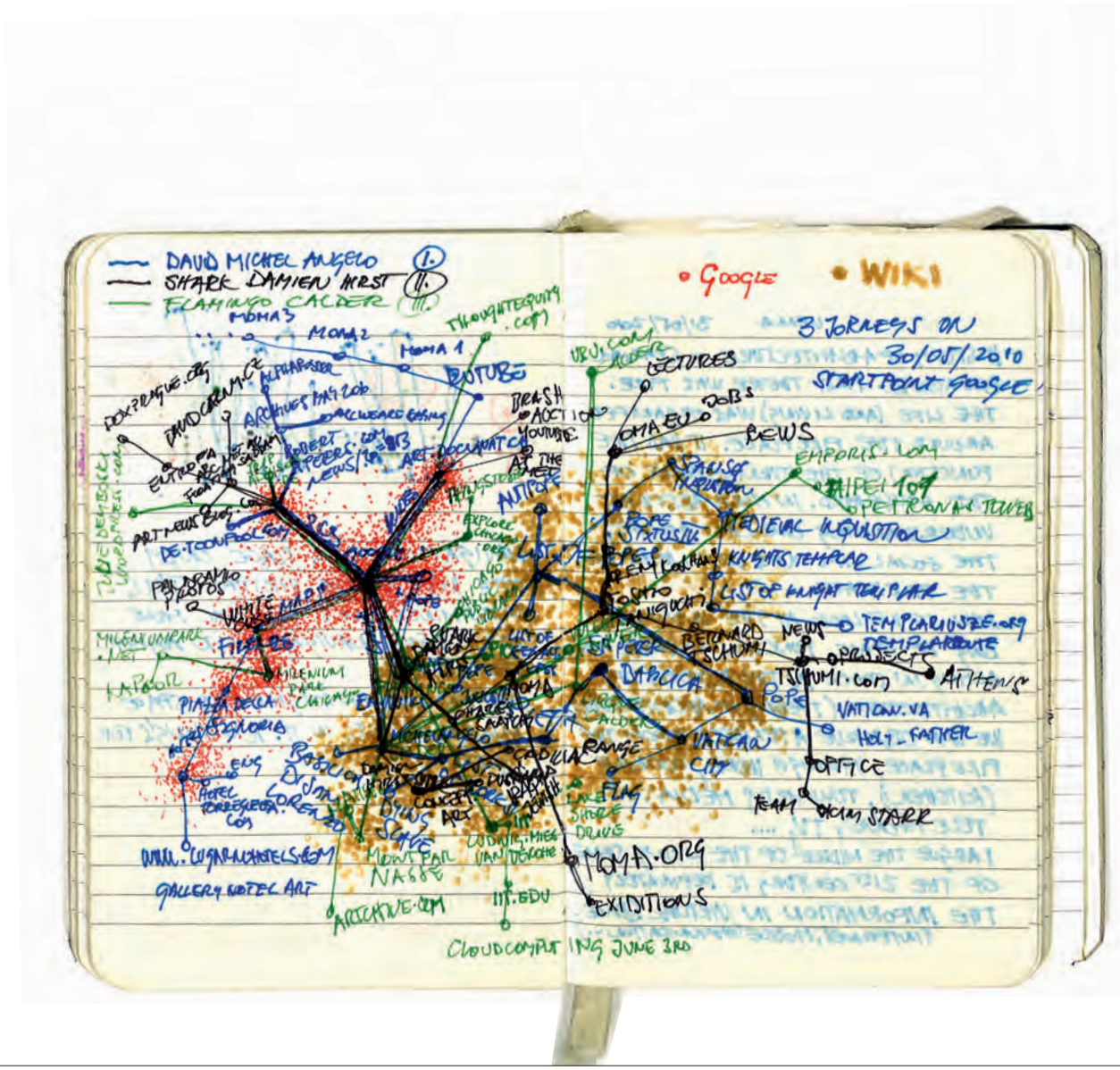
Navigation takes place only with help of the virtual. This determines one's physical movement. Visual navigation in augmented space.

One has to investigate the space and the objects in it. Nothing's for sure.

History of browsing from 3 keywords in Google. This is a visual attempt to record the journey in the cyberspace of the internet. Given 3 keywords:

david michelangelo
shark damien hirst
flamingo calder

The space of Google and Wikipedia are marked in red and brown.



Vienna, AT 31/05/2010

History of architecture – spacial behaviour.

First there was fire. The life (and living) was organised around the fireplace. It had the function of the turning point of the household. In more abstract way one could argue that the social life was dependent on the fireplace.

Now if we look at the understanding of space, function in space and acting in space in the history of architecture (theory of architecture), we can observe a tendency from the fireplace, through work, cooking (kitchen), towards media. Telephone, TV,...

I argue the middle of the living space of the 21st century is definitely the information in the virtual space (internet, mobile communication...).

Vienna, AT 02/06/2010

For enabling movement, navigation and orientation (in both real and virtual) the user will use layers. Switching them on and off gives the space meaning, usefulness and versatility. Control of the informational density. Layers such as:

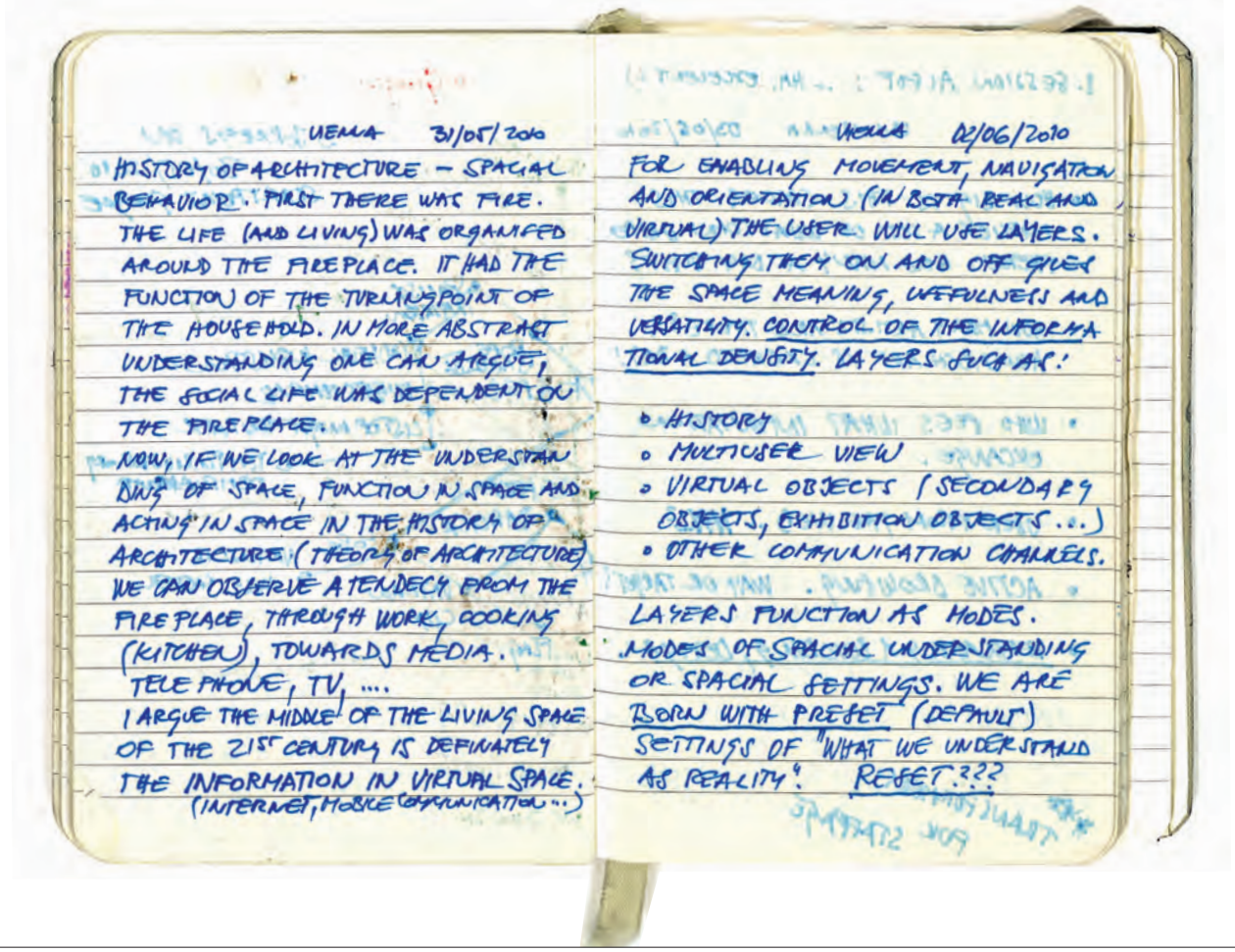
history

multiuser view

virtual objects (secondary objects, exhibition objects,...)

other communication channels

Layers function as modes. Modes of spacial understanding or spacial settings. We are born with preset (default) settings of *what we understand as reality.* Reset???



Vienna, AT 03/06/2010

2. Session Alsop:

How big is this space? This implies a lot of what can happen there. How many a day visit?

How many artists are there? How much art is created a day?

Who sees what? Information exchange.

User from the web?

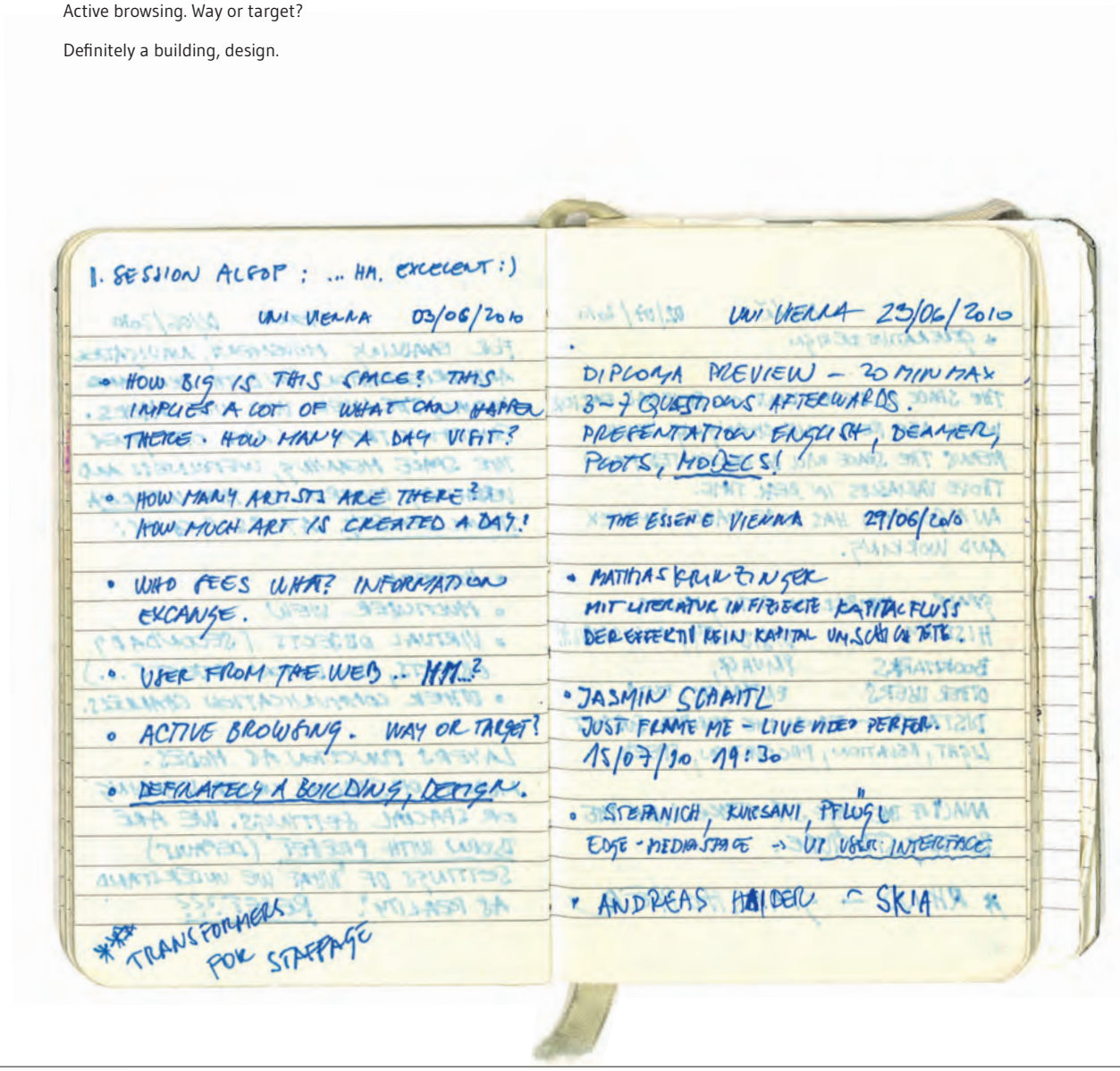
Active browsing. Way or target?

Definitely a building, design.

Vienna, AT 23/06/2010

diploma preview

20mins max, 3–7 questions afterwards. Presentation english, beamer, plots, models?!



Murán, SK 02/07/2010

Generative design

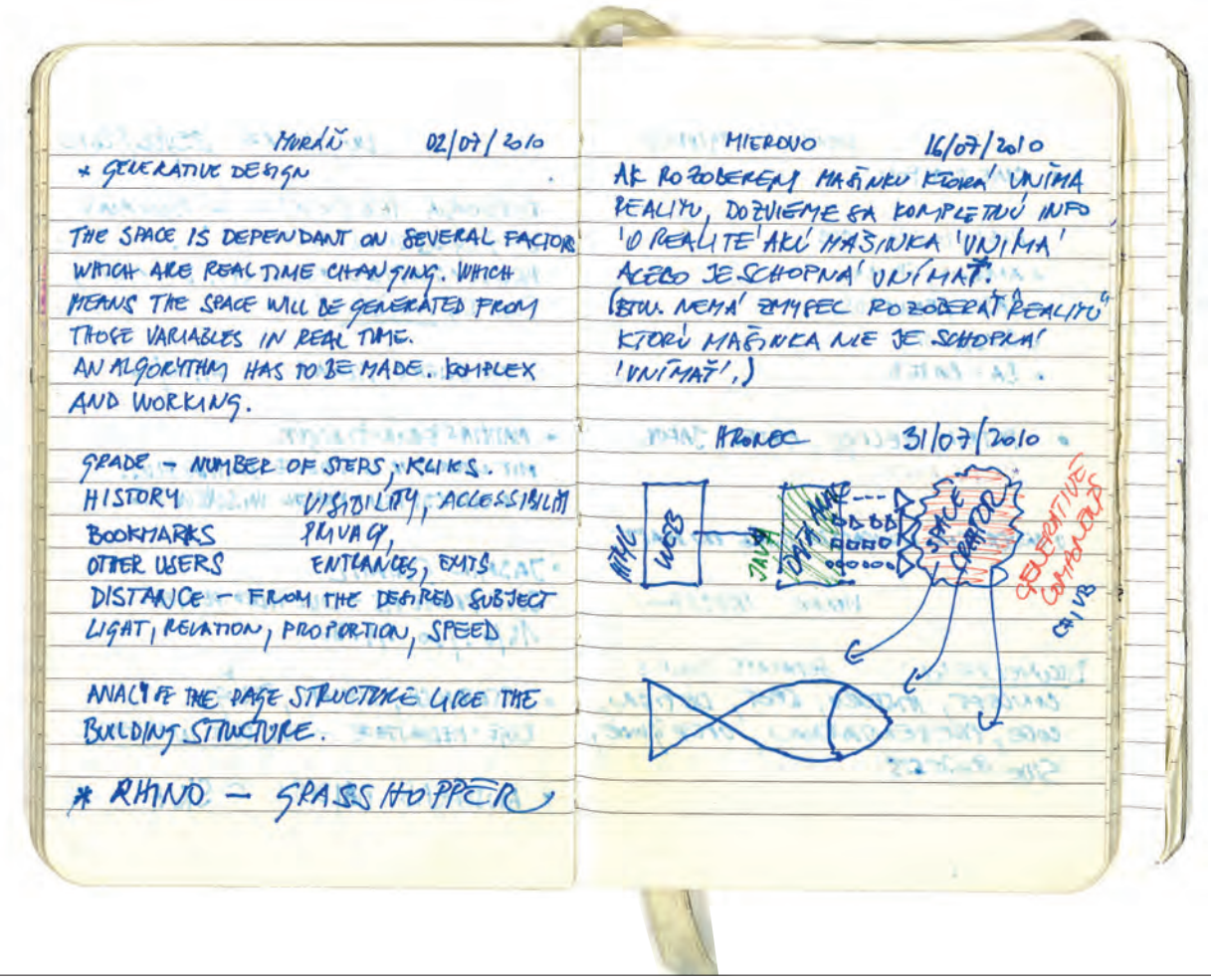
The space is dependant on several factors which are realtime changing. Which means the space will be generated from those variables in realtime.

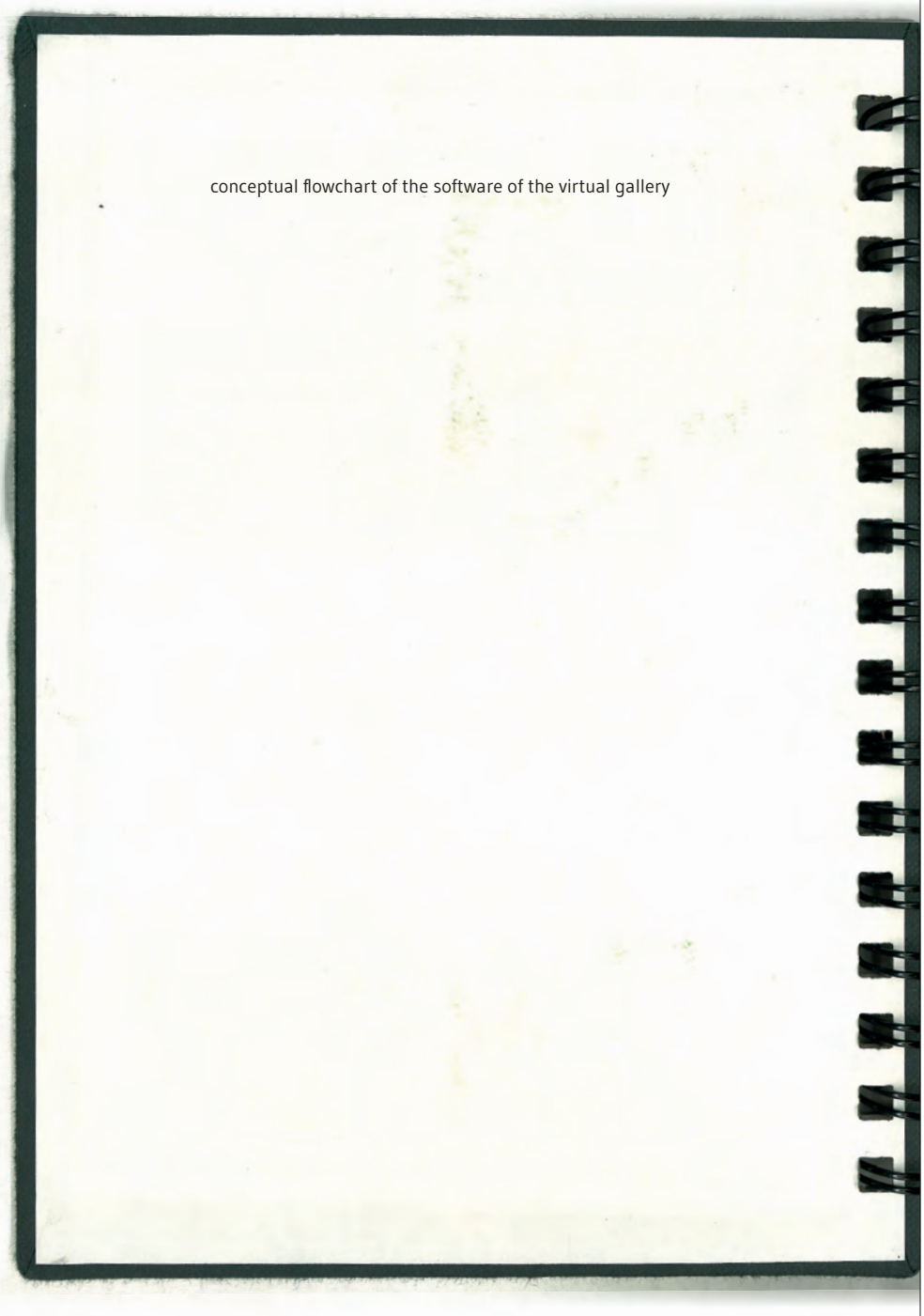
An algorythm has to be made. Complex and working.

Grade – number of steps, cliks
history – visibility, accessibility
bookmarks – individuality and privacy
other users – entrances, exits
distance – from the desired subject
light, relation, proportion, speed.

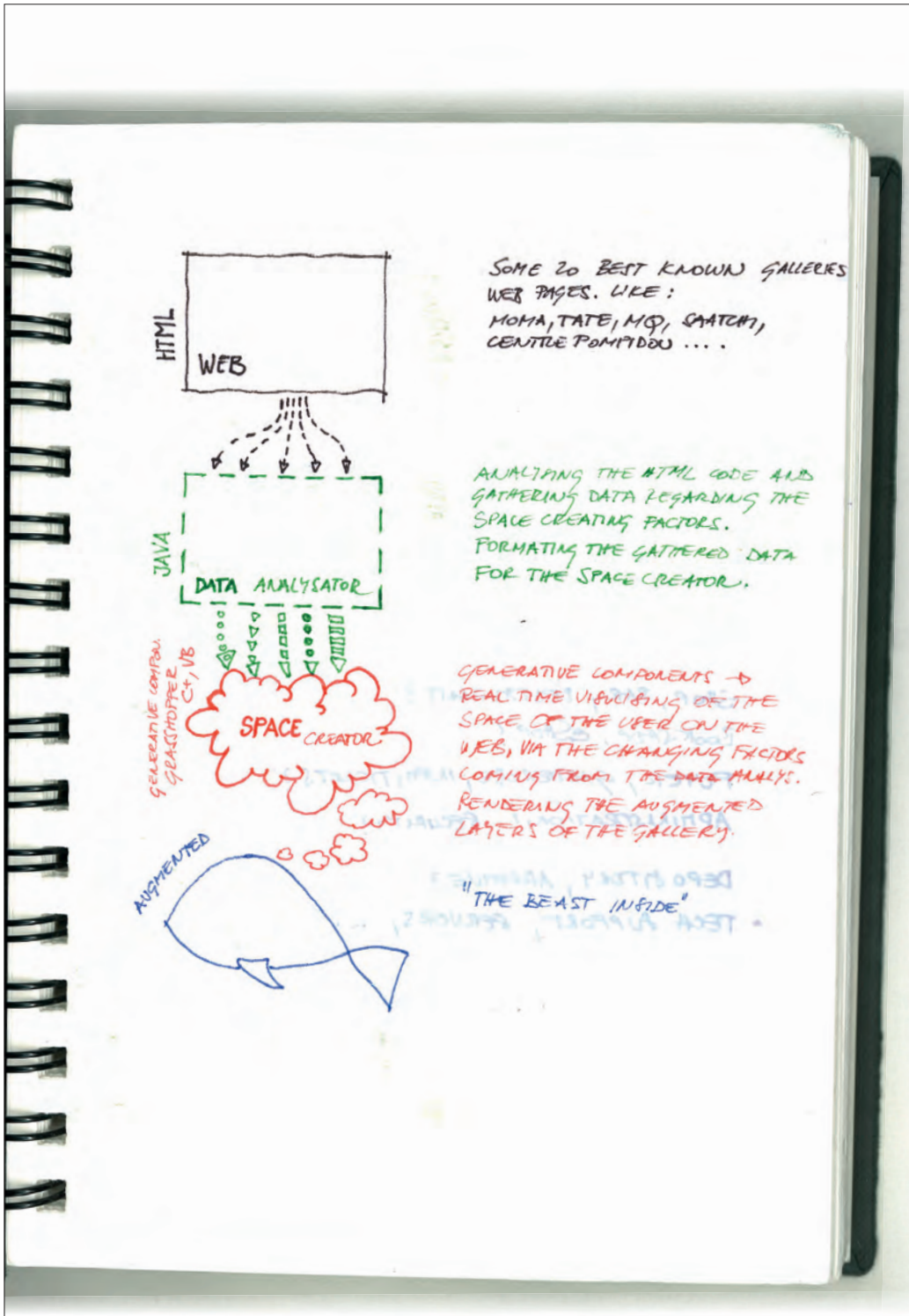
Analyse the page structure like the building structure.

Rhino – Grasshopper





conceptual flowchart of the software of the virtual gallery


HTML
WEB
JAVA
DATA ANALYSATOR
GENERATIVE COMPON.
GRASSHOPPER
C+, VB
SPACE CREATOR
AUGMENTED
SOME 20 BEST KNOWN GALLERIES WEB PAGES. LIKE:
MOMA, TATE, MOD, GAATCH, CENTRE POMPIDOU ....
ANALYSING THE HTML CODE AND GATHERING DATA REGARDING THE SPACE CREATING FACTORS.
FORMATING THE GATHERED DATA FOR THE SPACE CREATOR.
GENERATIVE COMPONENTS →
REAL TIME VISUALISING OF THE SPACE OF THE USER ON THE WEB, VIA THE CHANGING FACTORS COMING FROM THE ~~DATA~~ ANALYS.
RENDERING THE AUGMENTED LAYERS OF THE GALLERY.
"THE BEAST INSIDE"

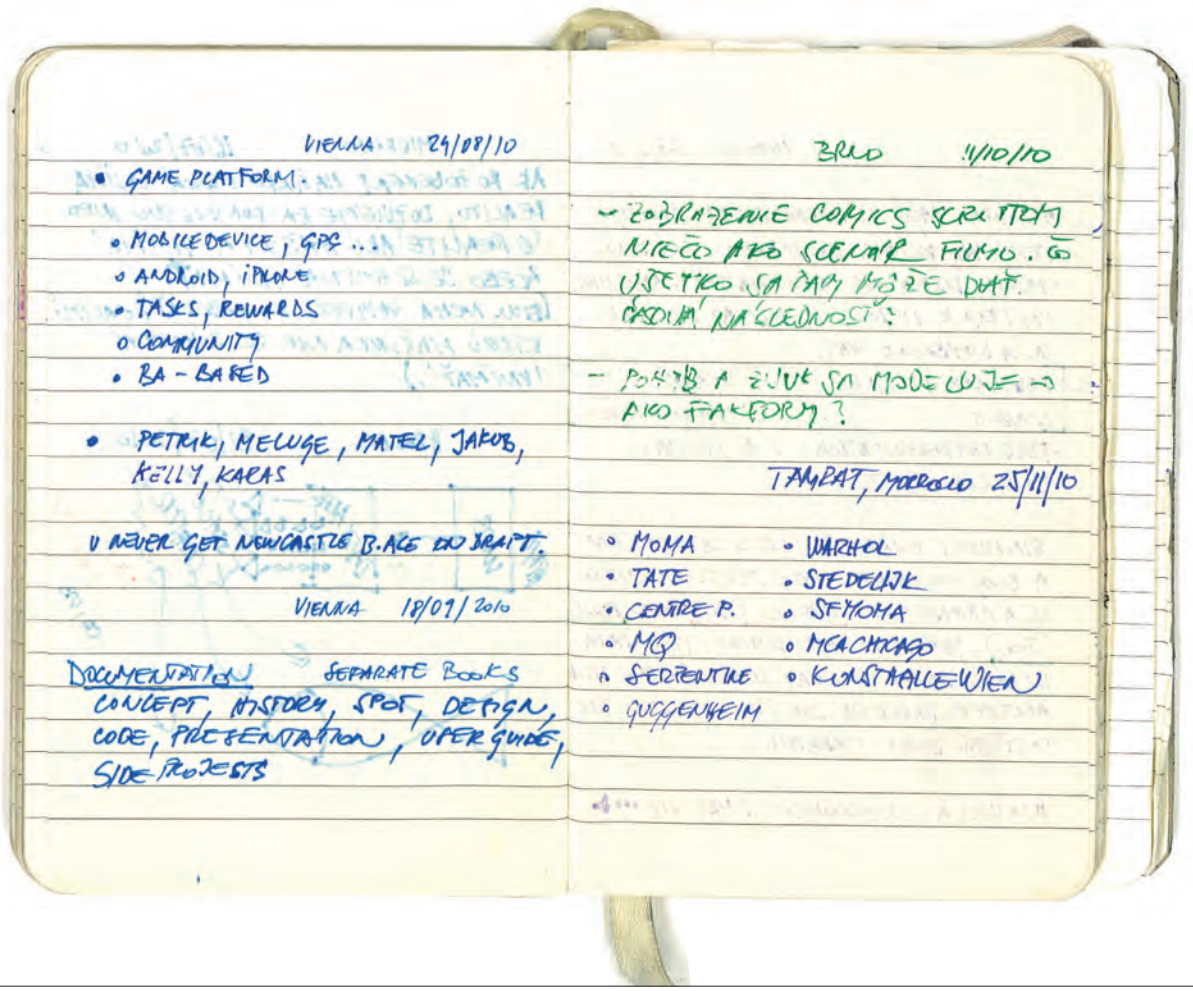
Vienna, AT 18/09/2010

Documentation (separate books)

concept, history, spot, design, code, presentation, user guide, side projects

Tamrat, MA 25/11/2010

MoMA, Tate, CentrePompidou, MQ, Serpentine, Guggenheim, Warhol, Stedelijk, SFMoMA, MCA Chicago, Kunsthalle Wien



Tamrat, MA 27/11/2010

An HTML page is represented through text and pictures on screen. Building links. Displaying information. Now what I'm thinking to do is to display this information in a different way.

Text and pictures – space

links – connections

This representation is a virtual environment.

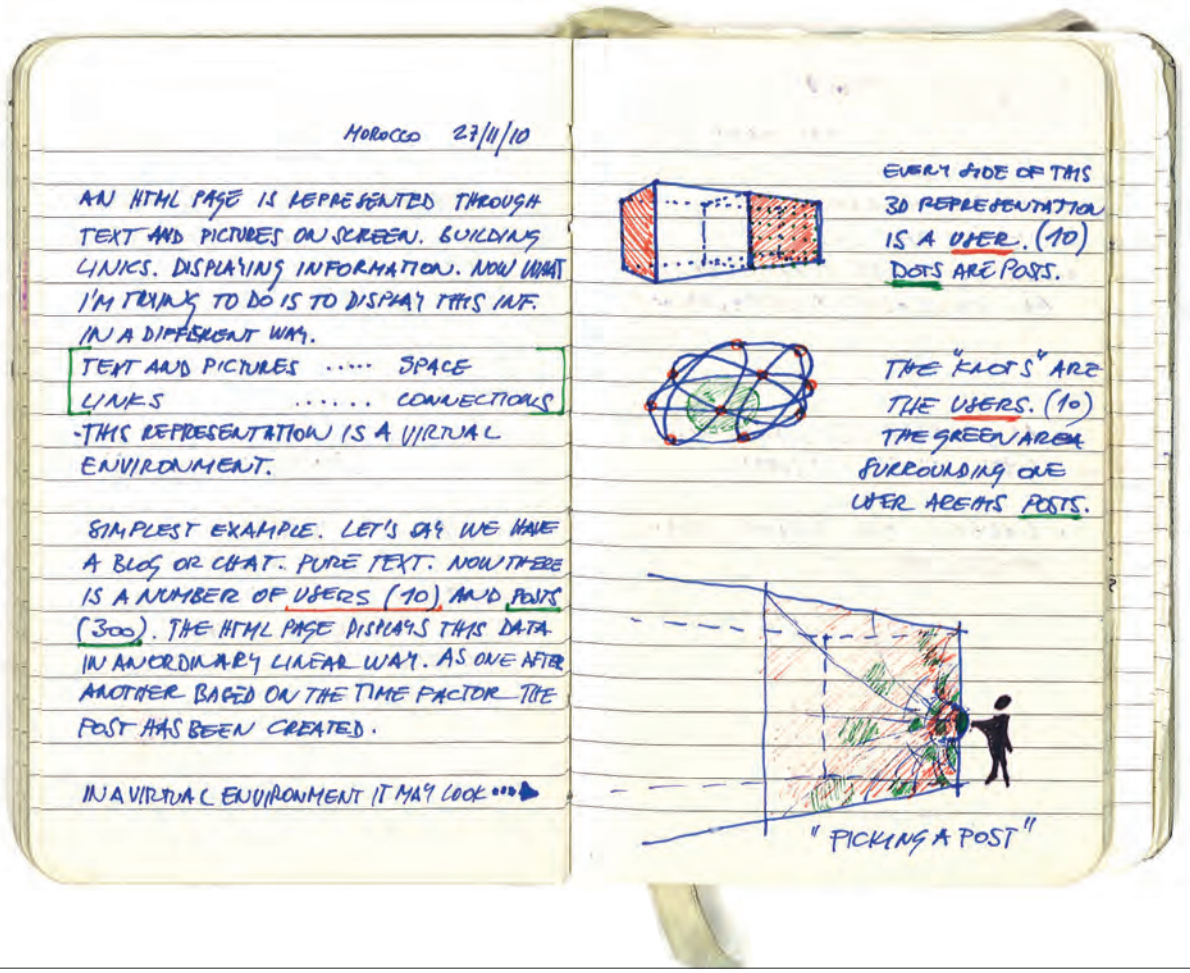
Simplest example. Let's say we have a blog or chat. Pure text. Now there is a number of users (10) and posts (300). the HTML page displays this data in an ordinary linear way. As one after another based on the time factor the post has been created.

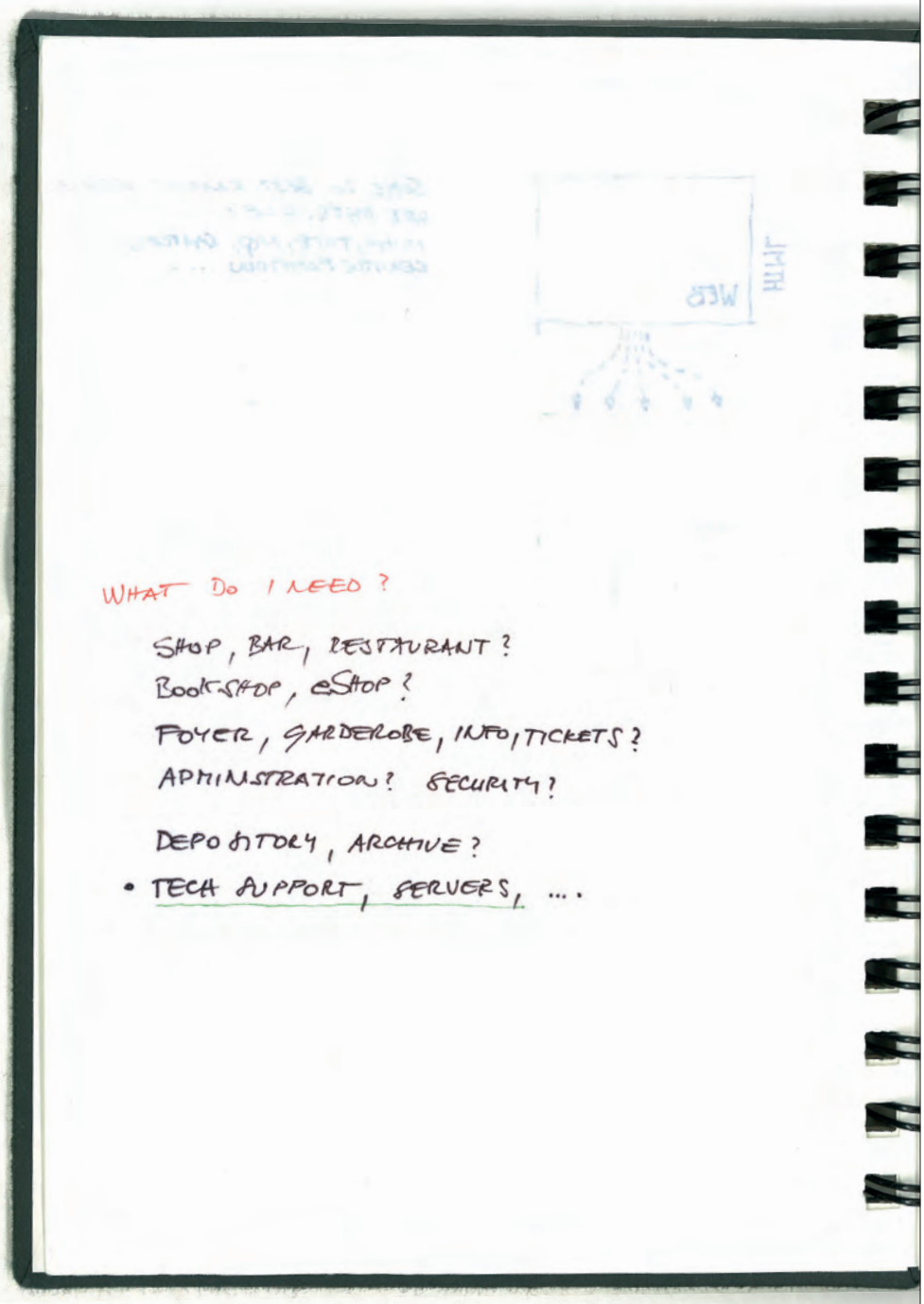
In an virtual environment it may look → ... pictures

Every side of this 3D representation is a user. (10) dots are posts.

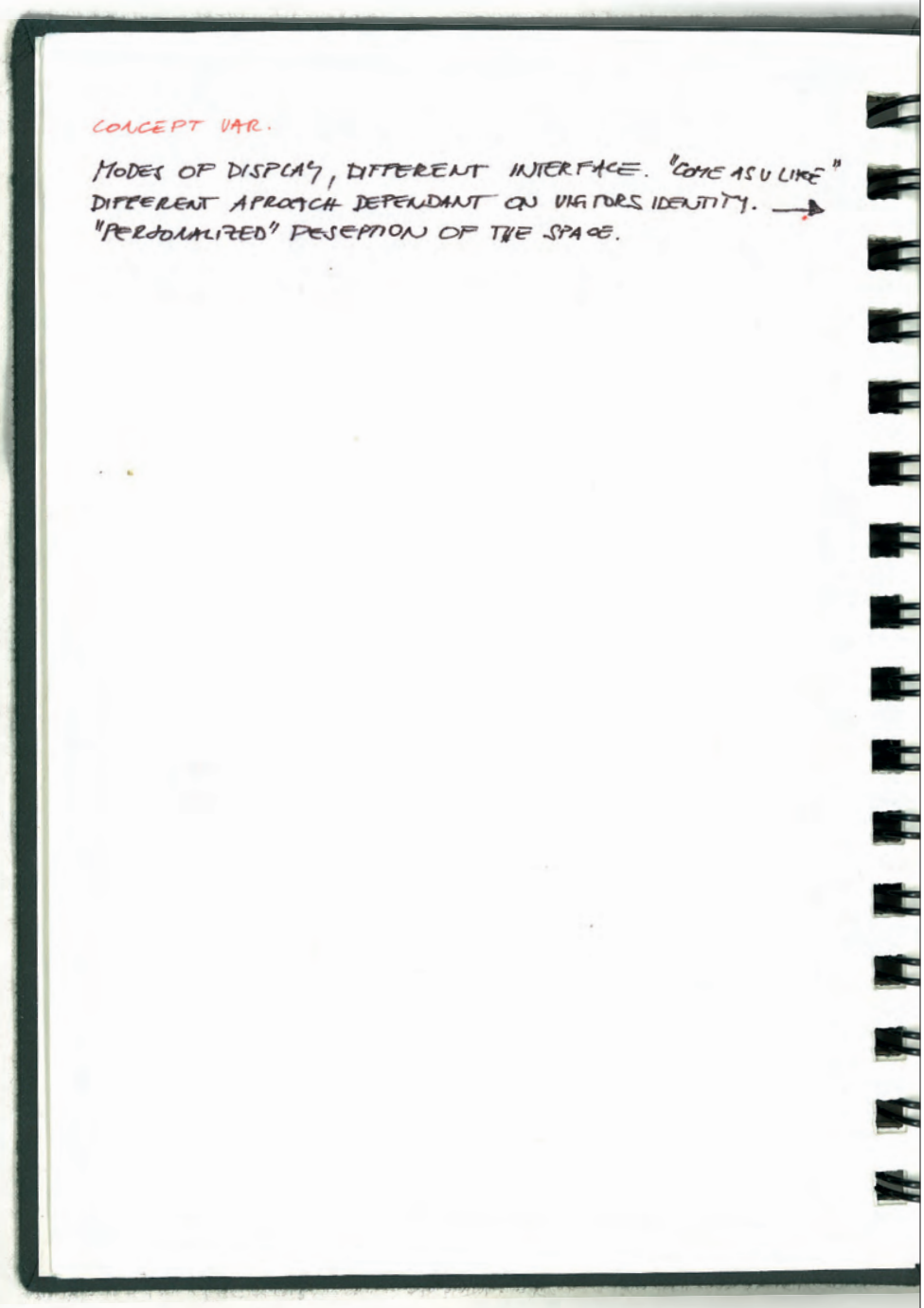
The *knots* are the the users. (10) The green are surrounding one user are his posts.

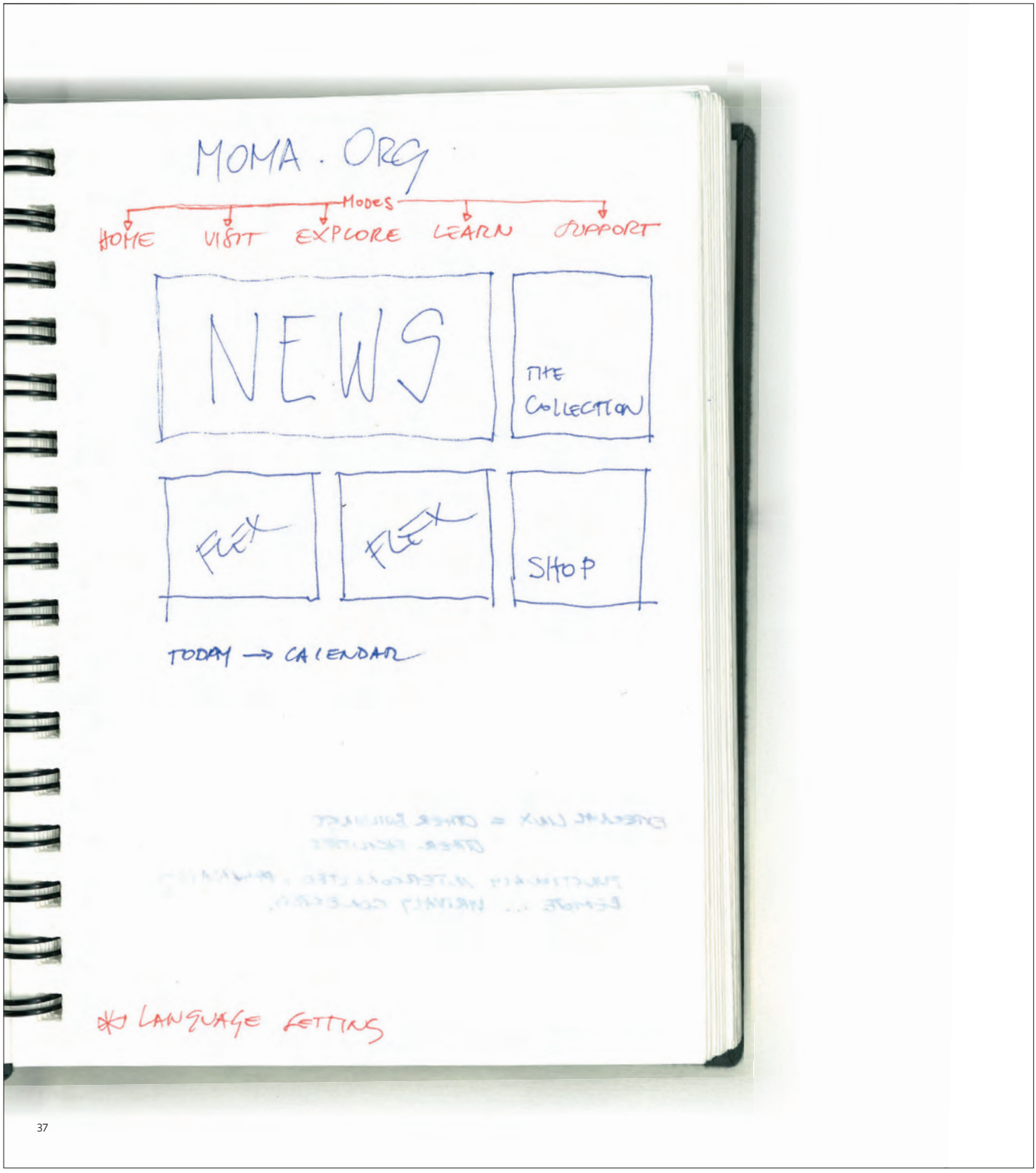
Picking a post

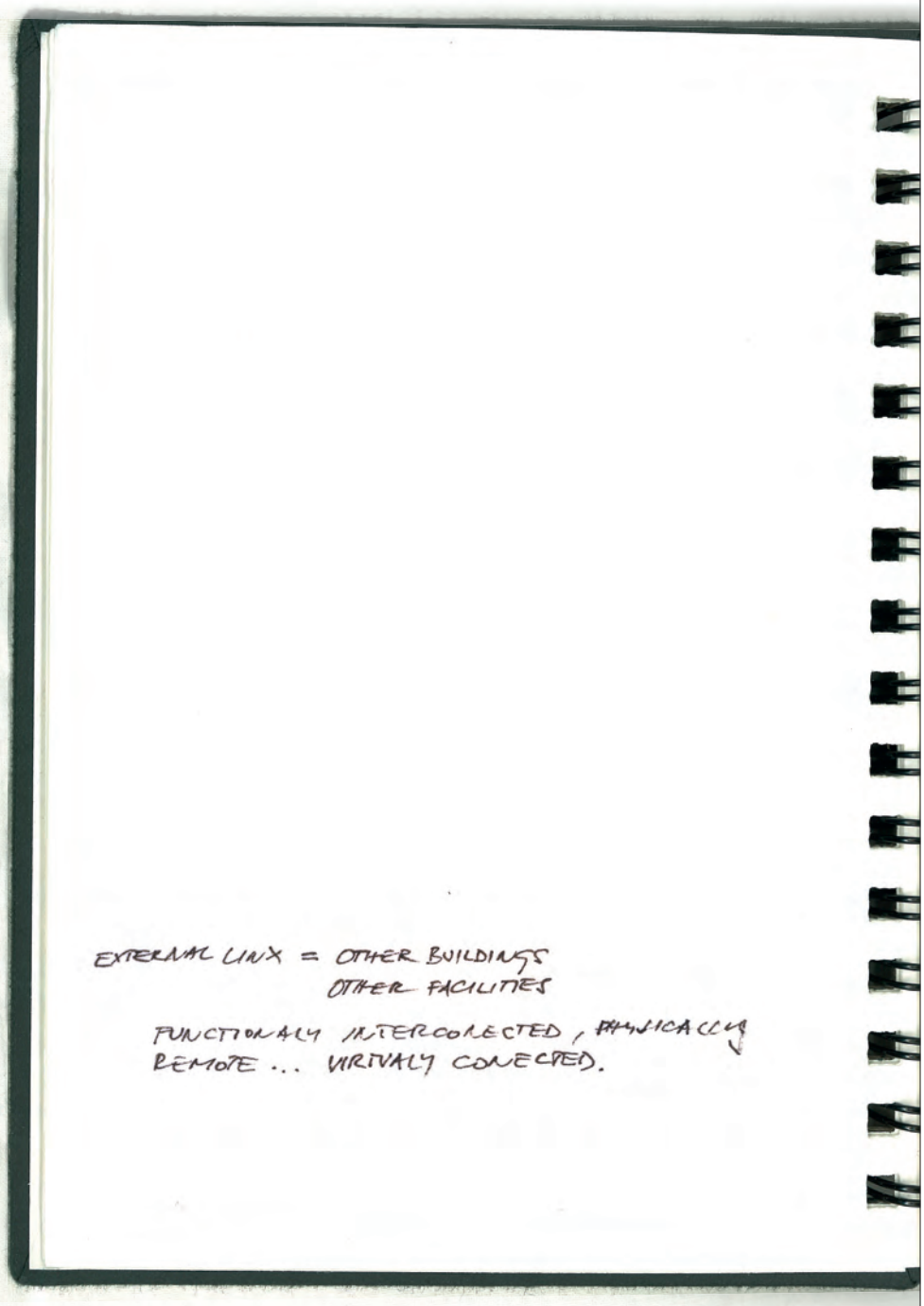


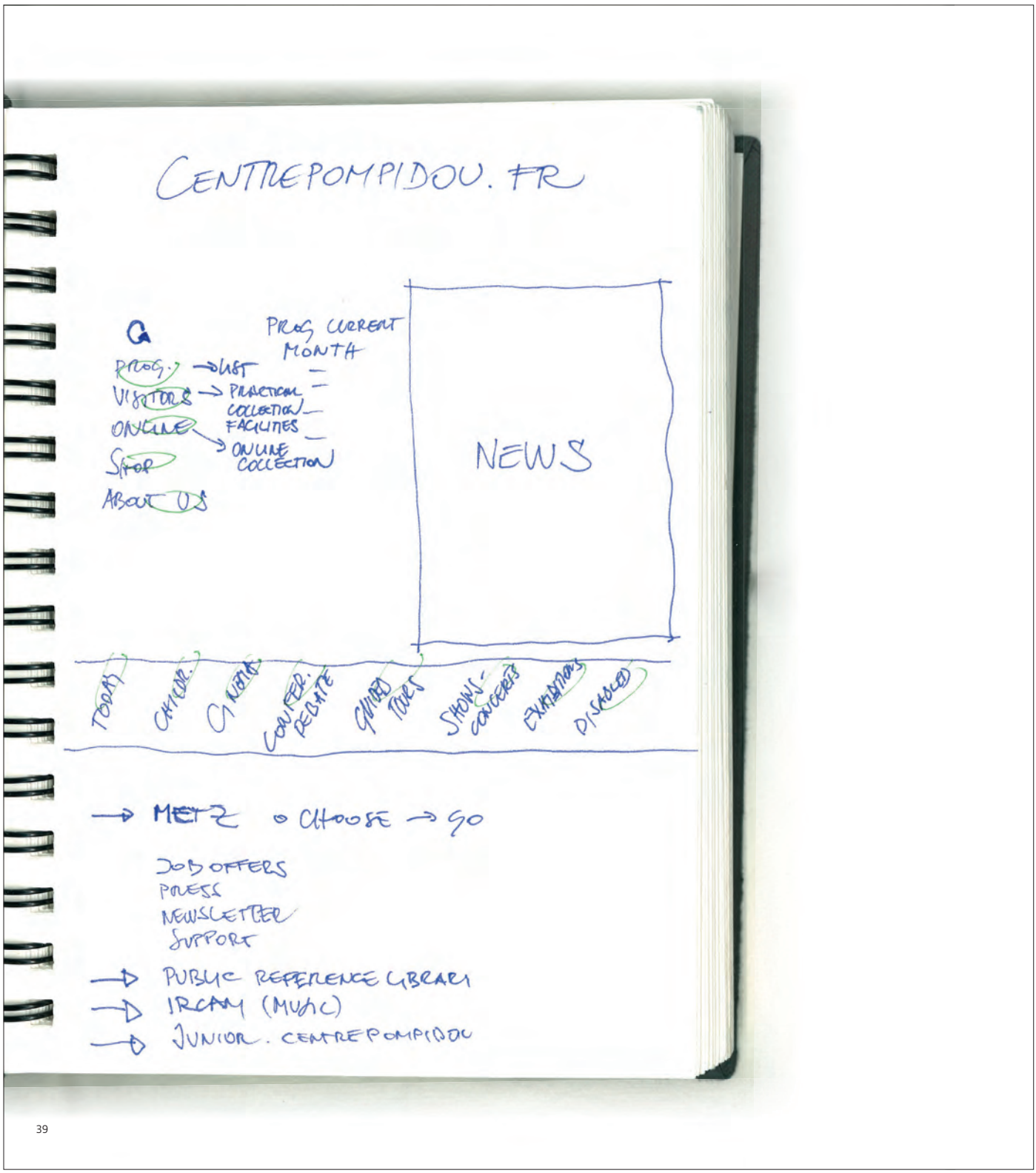

WHAT DO I NEED ?
SHOP, BAR, RESTAURANT ?
BOOK-SHOP, eSHOP ?
FOYER, GARDEROBE, INFO, TICKETS ?
APMINISTRATION ? SECURITY ?
DEPOSITORY, ARCHIVE ?
• TECH SUPPORT, SERVERS, ...

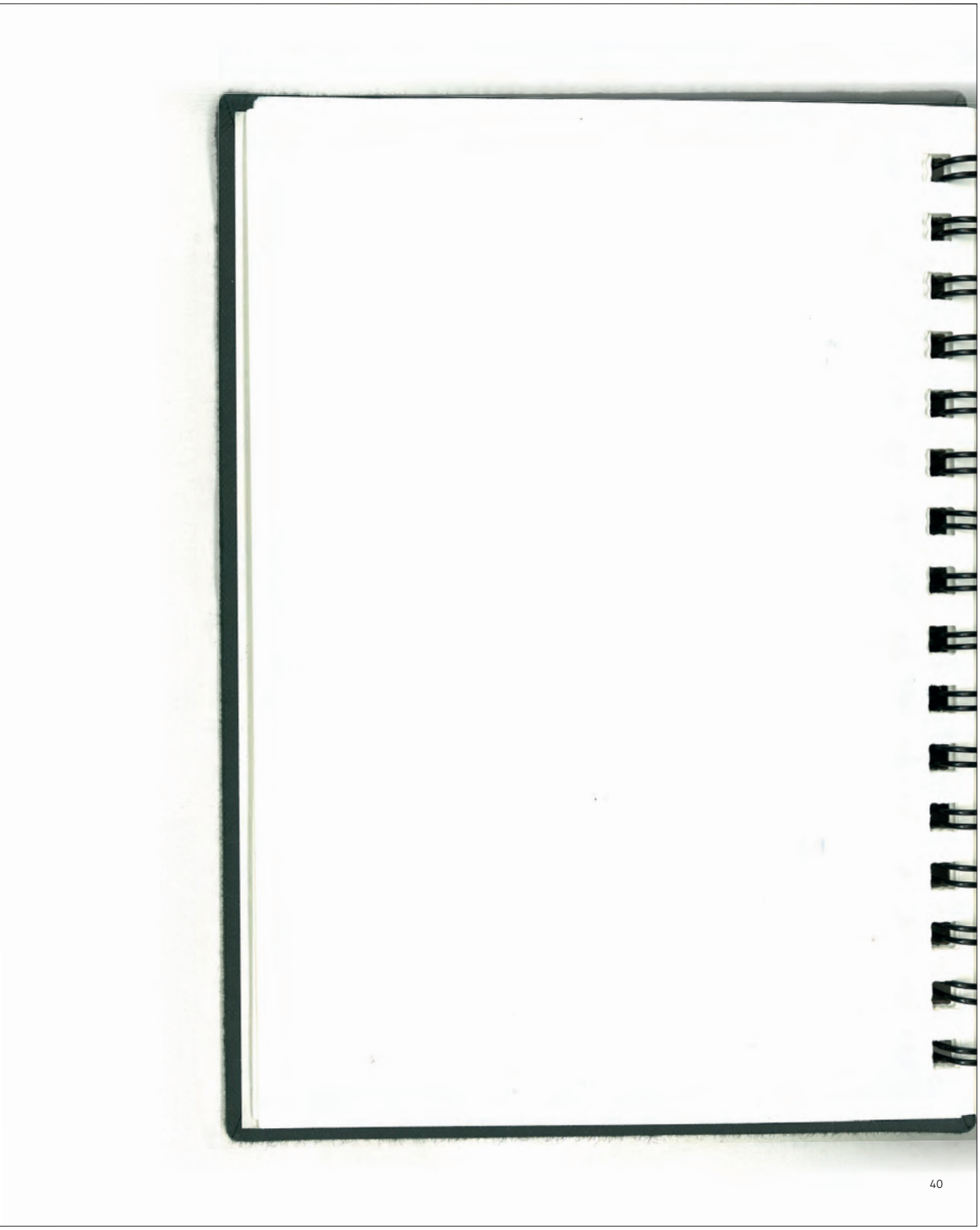

TATE . ORG . UK
LANGUAGE SETTINGS
COLLECTION
ABOUT
SEARCH
ARTISTS
TURNER
NEWS
FEEDBACK
SURVEY
EMAIL BULLET.
BLOG
RESEARCH
LOCATIONS
BRIT
MOD
LIV
ST IVES
EXHIBITIONS
CURRENT
FUTURE
PAST ?
CALENDAR
FOR VISITOR
DIRECTIONS
FAMILY INFO
TICKETS
ABOUT US
CONTACT
WORK
PRESS
SHOP
ENTERTAIN.
SUPPORT
DONATIONS..
CORPORATE..
PATRONS...
MEMBERS
OTHER ENTITIES
TATE CHANNEL
TURNER PRIZE
PUBLISHER
BOOKS
TRADE
LEARN ONLINE
COURSES
LEARNING INSIDE
KIDS, YOUNG...
EMAIL
TWITTER
FACEBOOK
YOUTUBE
FLICKR


CONCEPT VAR.
MODES OF DISPLAY, DIFFERENT INTERFACE. "COME AS U LIKE"
DIFFERENT APROACH DEPENDANT ON VISITORS IDENTITY.
"PERSONALIZED" PERCEPTION OF THE SPACE.

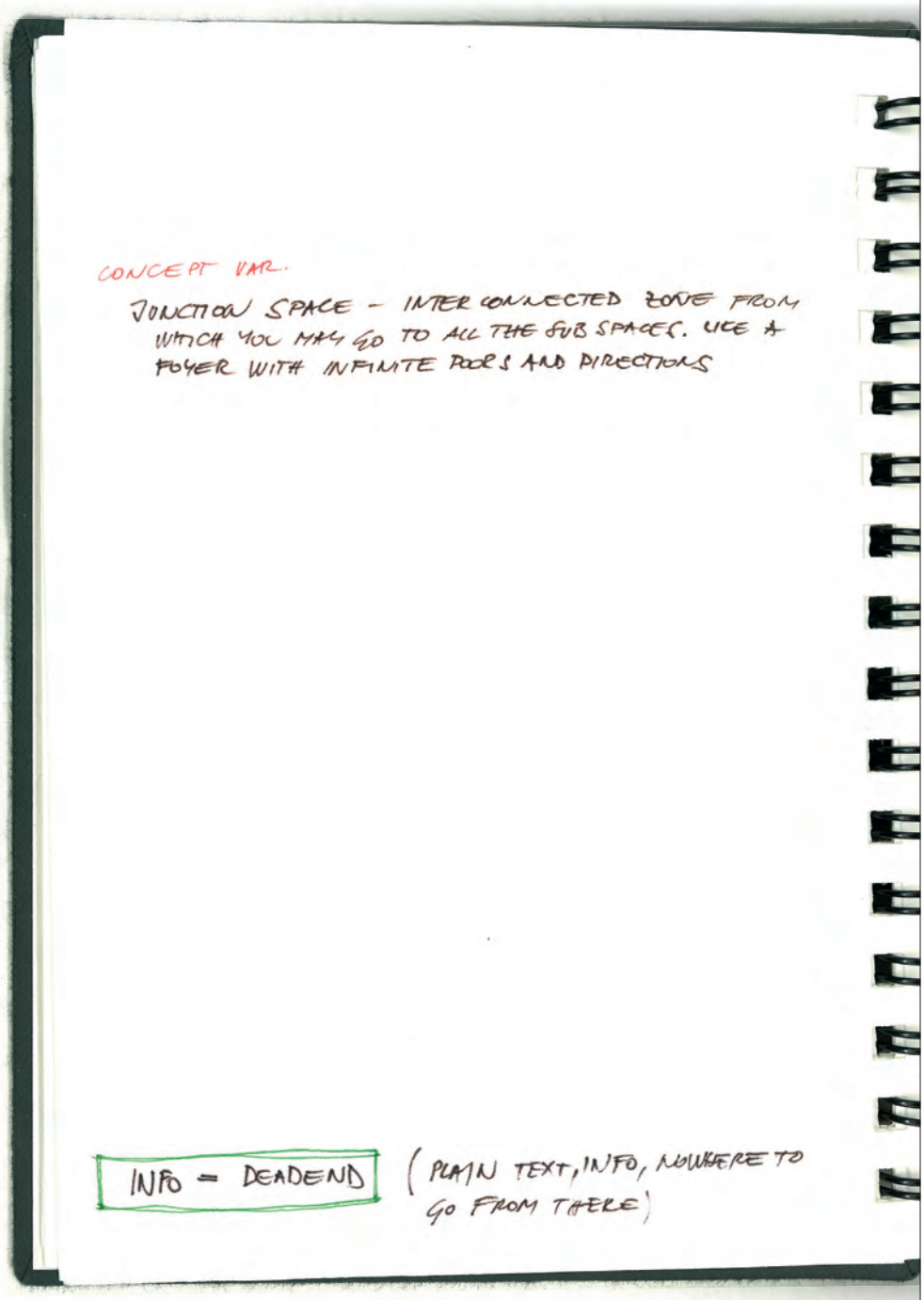

MOMA.ORG
Modes
HOME
VISIT
EXPLORE
LEARN
SUPPORT
NEWS
THE COLLECTION
FLEX
FLEX
SHOP
TODAY → CALENDAR
* LANGUAGE SETTING

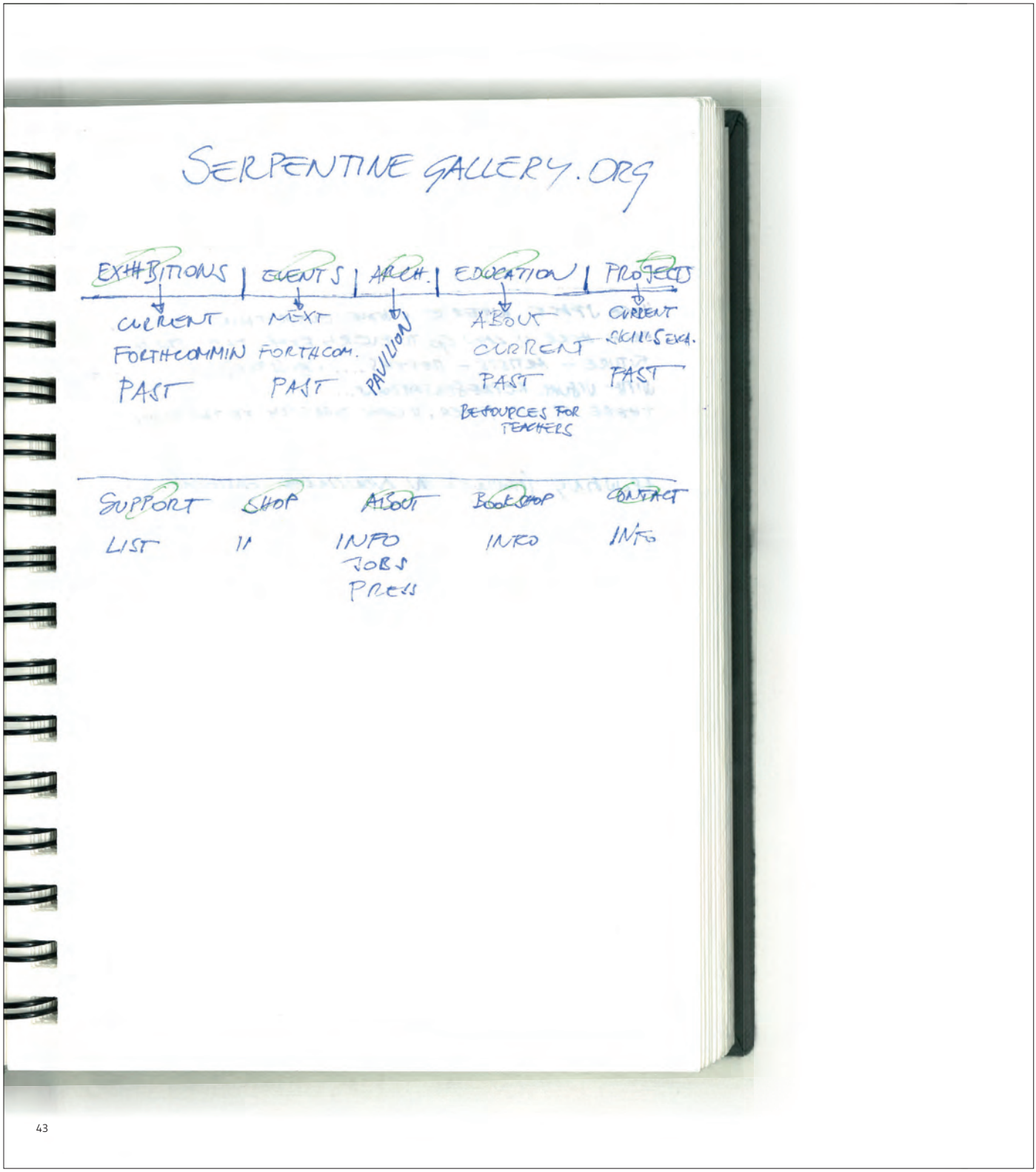

EXTERNAL LINK = OTHER BUILDINGS
OTHER FACILITIES
FUNCTIONALLY INTERCONECTED, PHYSICALLY
REMOTE ... VIRTUALLY CONECTED.

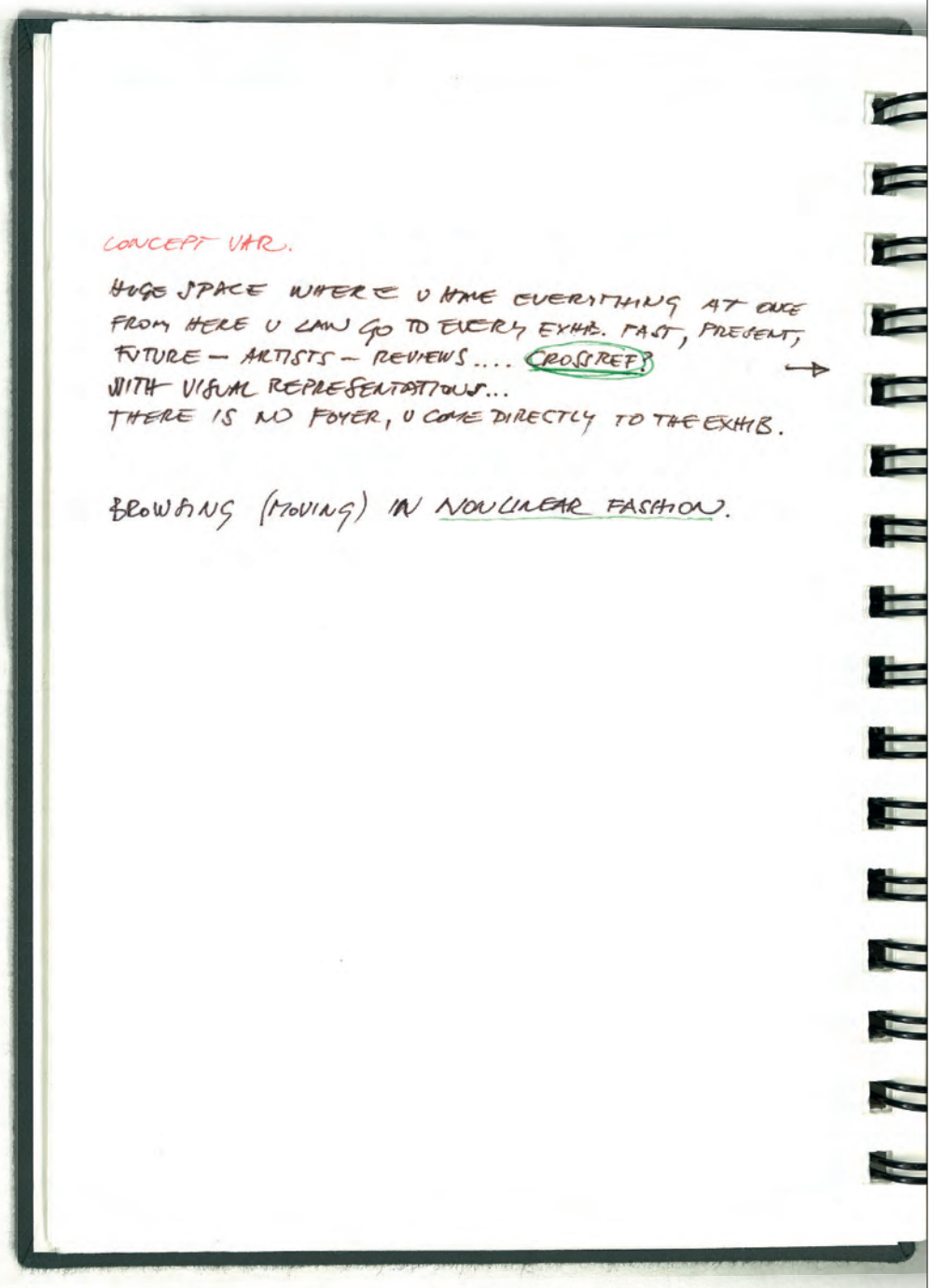

CENTREPOMPIDOU.FR
PROG CURRENT MONTH
PROG. → LIST
VISITORS → PRACTICAL
COLLECTION
FACILITIES
ONLINE
ONLINE COLLECTION
SHOP
ABOUT US
NEWS
TODAY
CALENDR.
CONFER. DEBATE
GUIDED TOURS
SHOWS CONCERTS
EXHIBITIONS
DISABLED
→ METZ ○ CHOOSE → GO
JOB OFFERS
PRESS
NEWSLETTER
SUPPORT
→ PUBLIC REFERENCE LIBRARY
→ IRCAM (MUSIC)
→ JUNIOR. CENTREPOMPIDOU

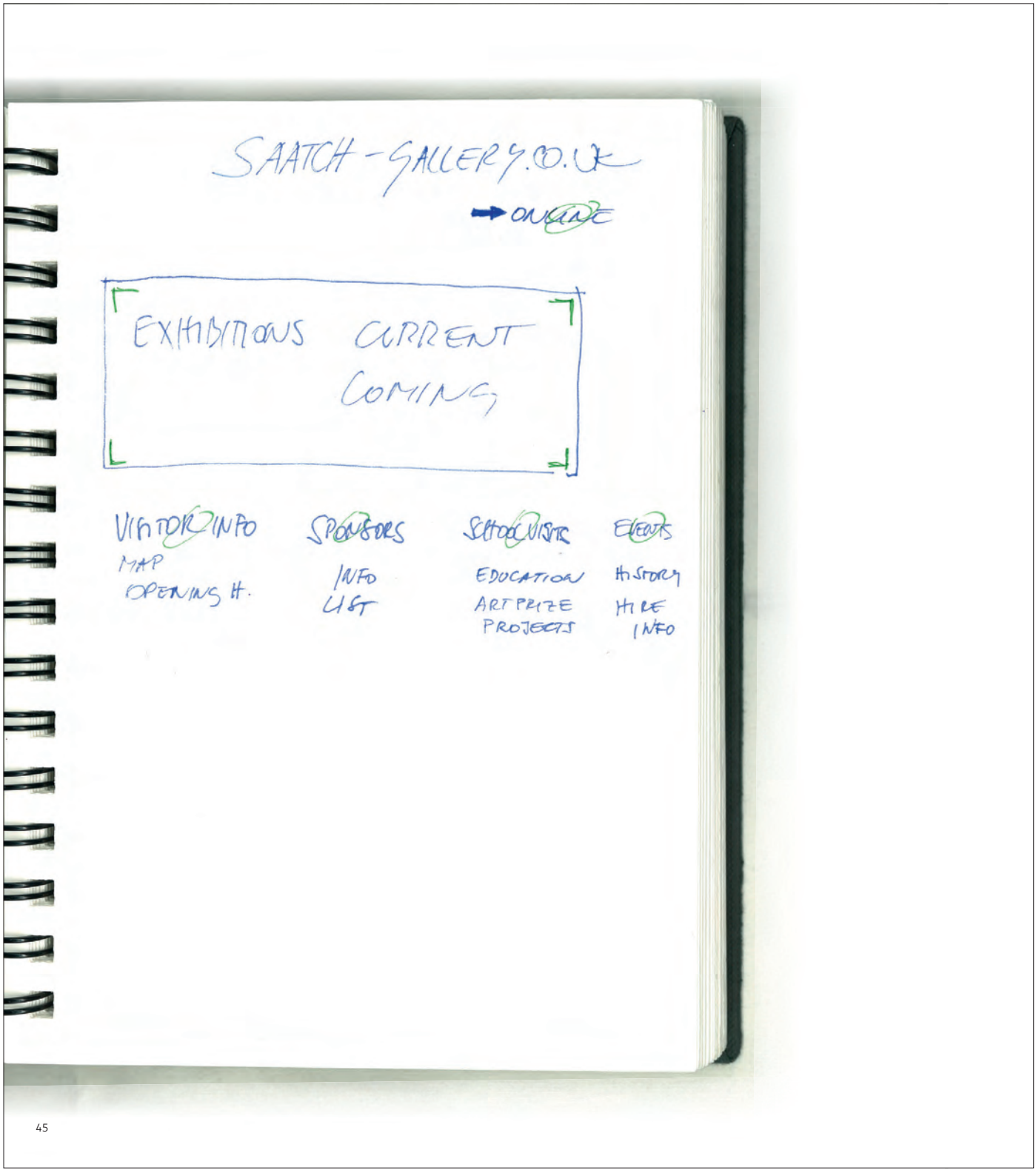



MQW.AT
LANGUAGE SETTING
CURRENT EXHIBITION
VISITOR
– INSTITUTIONS
– OPENING H. PRICES
– GUIDED TOURS
PROGRAM
– SORT DATE, CATEG., TITLE
THE MQ
– ABOUT
– STATEMENT
– FACTS...
EAT & DRINK
SHOP
EVENTS
PRESS
NEWSLETTER


CONCEPT VAR.
JUNCTION SPACE - INTERCONNECTED ZONE FROM WHICH YOU MAY GO TO ALL THE SUB SPACES. LIKE A FOYER WITH INFINITE DOORS AND DIRECTIONS
INFO = DEADEND
(PLAIN TEXT, INFO, NOWHERE TO GO FROM THERE)


SERPENTINE GALLERY.ORG
EXHIBITIONS
EVENTS
ARCH.
EDUCATION
PROJECTS
CURRENT
FORTHCOMMIN
PAST
NEXT
FORTHCOM.
PAST
PAVILION
ABOUT
CURRENT
PAST
RESOURCES FOR TEACHERS
CURRENT
SKILLS EXA.
PAST
SUPPORT
LIST
SHOP
//
ABOUT
INFO
JOBS
PRESS
BOOKSHOP
INFO
CONTACT
INFO


CONCEPT VAR.
HUGE SPACE WHERE U HAVE EVERYTHING AT ONCE
FROM HERE U CAN GO TO EVERY EXHIB. PAST, PRESENT,
FUTURE — ARTISTS — REVIEWS .... CROSSREF?
WITH VISUAL REPRESENTATIONS...
THERE IS NO FOYER, U COME DIRECTLY TO THE EXHIB.
BROWSING (MOVING) IN NONLINEAR FASHION.


SAATCH-GALLERY.CO.UK
ONLINE
EXHIBITIONS CURRENT
COMING
VISITOR INFO
MAP
OPENING H.
SPONSORS
INFO
LIST
SCHOOL VISIT
EDUCATION
ART PRIZE
PROJECTS
EVENTS
HISTORY
HIRE
INFO

HTML

WWW. MoMA . ORG

- USER / WINDOW / DATA
- AUTHOR / OBJECT
- CATEGORY
- +++

- BROWSING THROUGH THE COLLECTION OF MOMA ON THEIR WEB. EVERY VISITOR / USER LIKE EVERY WEBBROWSER WINDOW.

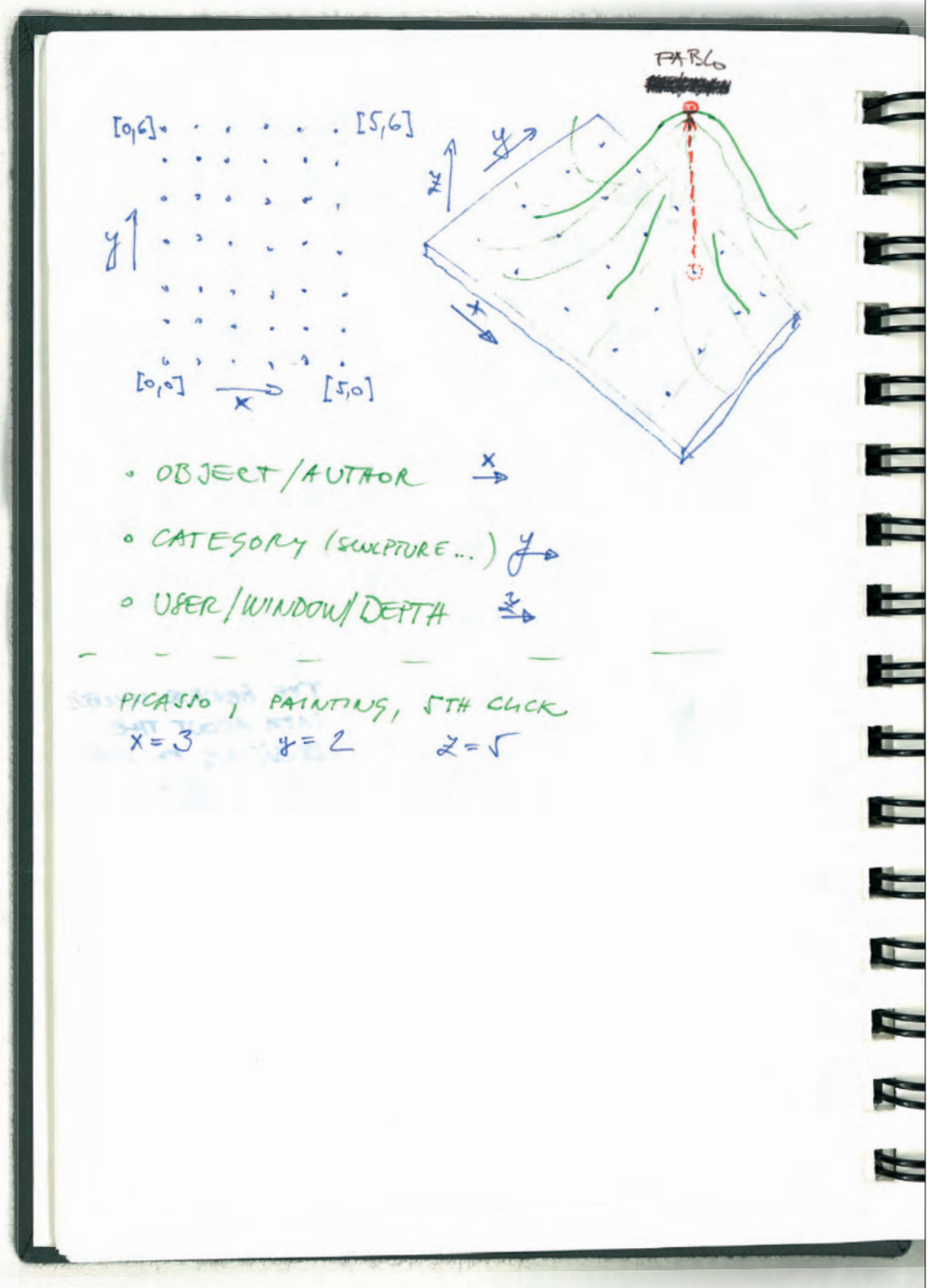
- BROWSING THE COLLECTION REPRESENTS MOVEMENT IN THE VIRTUAL WORLD. EVERY CLICK ON A LINK (ITEM IN THE COLLECTION) REPRESENTS A CHANGE → MOVEMENT IN THE SPACE.

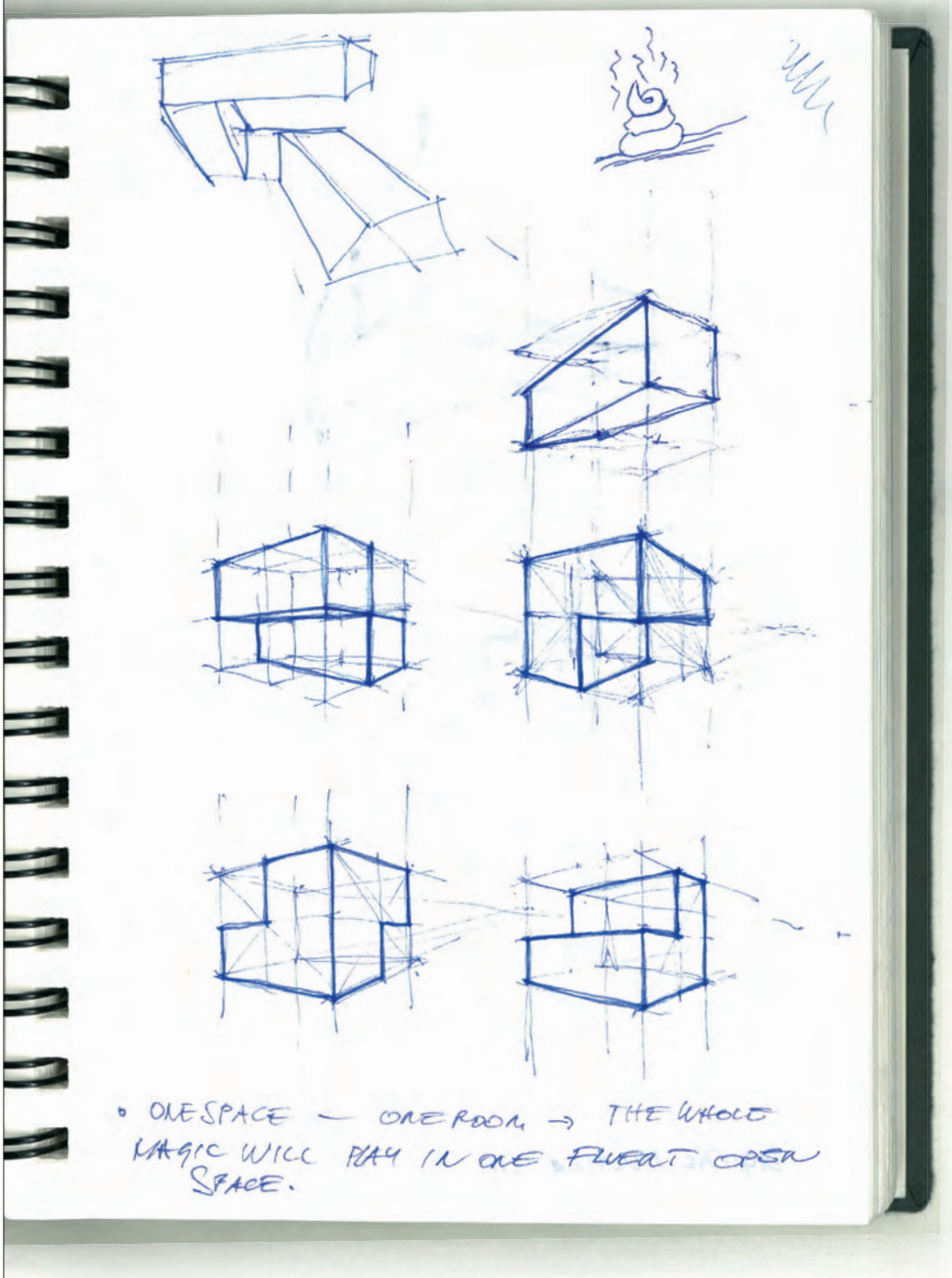
- THE REAL SPACE ~~THE~~ OF THE GALLERY REACTS TO THIS KIND OF MOVEMENT. BY CHANGE, ADAPTATION, DEFORMATION, REPRESENTATION...

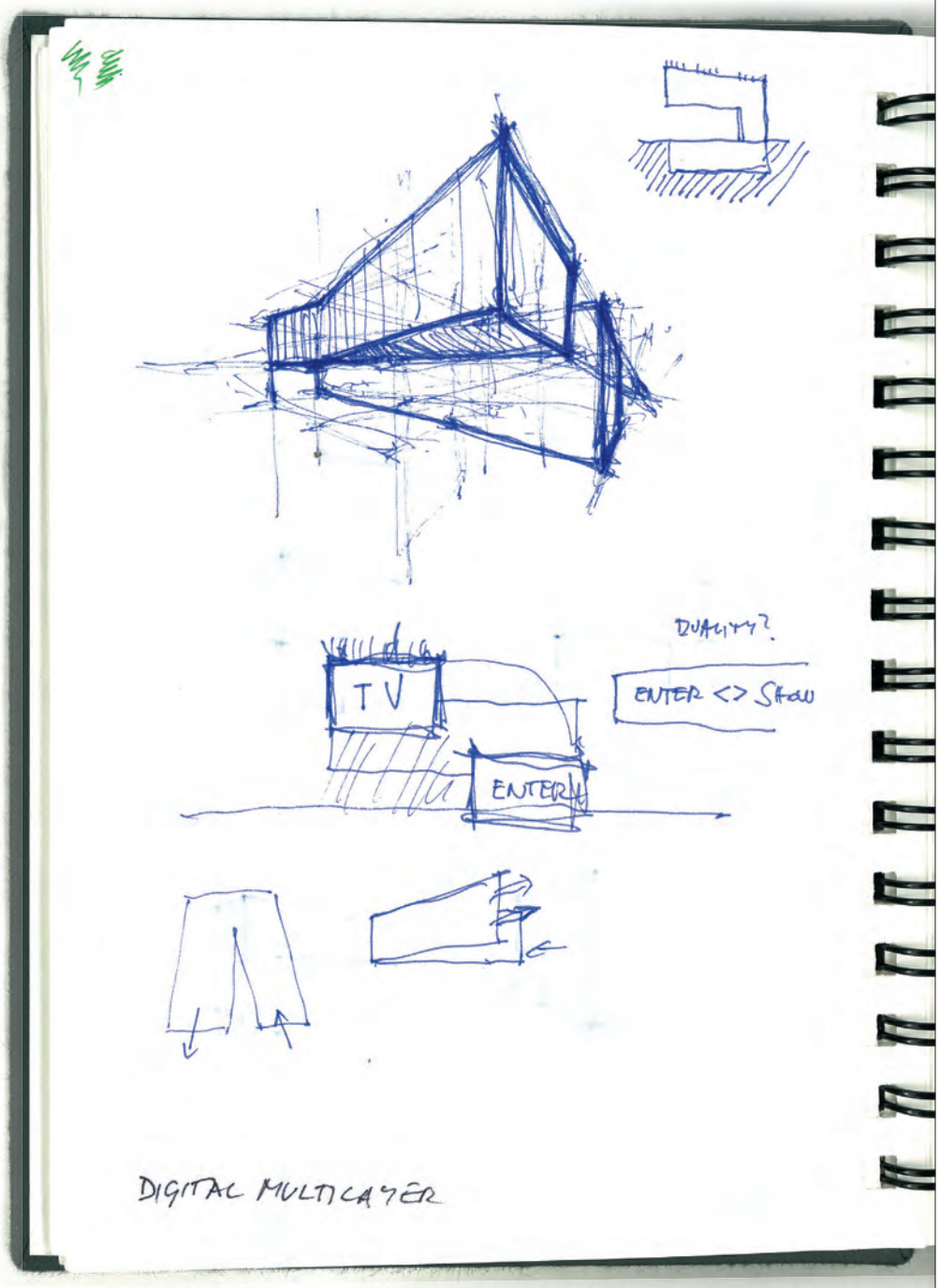
THE SERVER COLLECTS DATA ABOUT THE BROWSING FACTORS AUTHOR/OBJECT; CATEGORY; DEPTH.

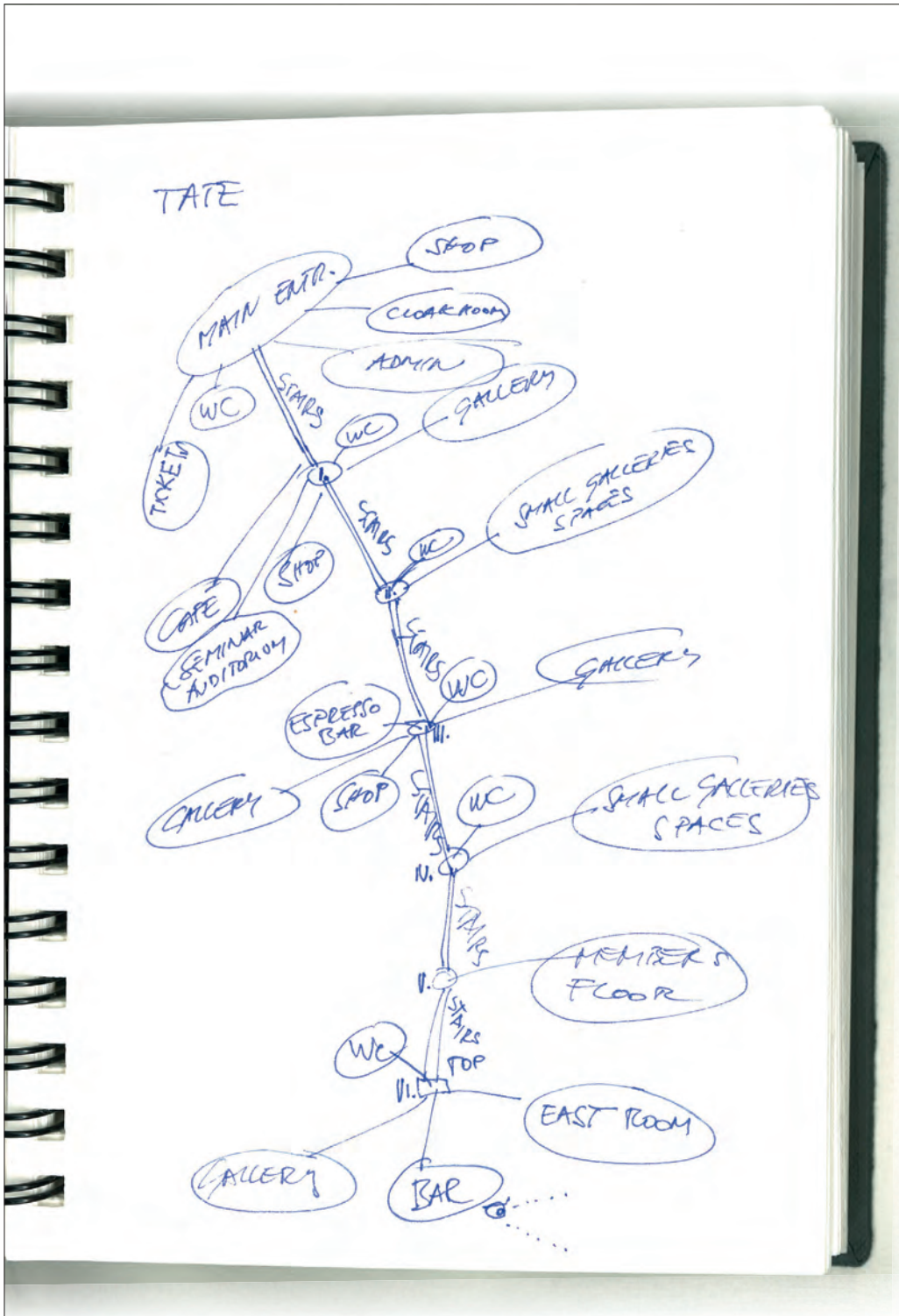
→ THEN SENDS THIS INFORM. TO THE DATA ANALYSIS.

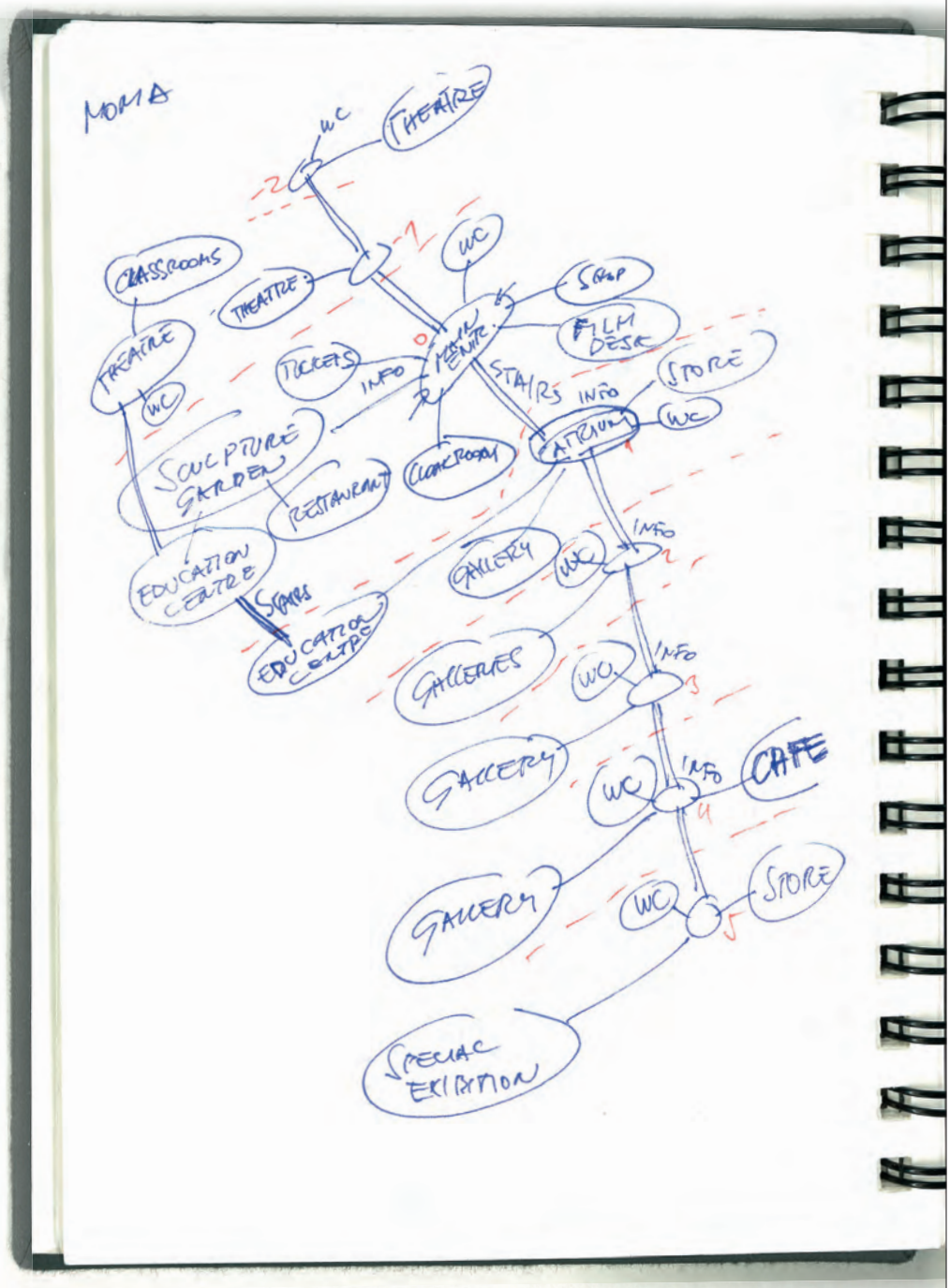
→ THERE IT IS TRANSFORMED INTO VARIABLES AND THEIR VALUES WHICH CAN THE SPACE CREATOR USE REALTIME TO "BUILD" THE SPACIAL REPRESENTATION OF THE ACTION.


PABLO
[0,6]
[5,6]
[0,0]
[5,0]
y
x
z
• OBJECT/AUTHOR
• CATEGORY (SCULPTURE...)
• USER/WINDOW/DEPTH
PICASSO, PAINTING, 5TH CLICK
x = 3
y = 2
z = 5


• ONE SPACE — ONE ROOM → THE WHOLE
MAGIC WILL PLAY IN ONE GREAT OPEN
SPACE.


TV
ENTER
ENTER <> STAY
DIGITAL MULTICAYER


TATE
SHOP
MAIN ENTR.
CLOAKROOM
ADMIN
GALLERY
WC
STAIRS
WC
TICKETS
SMALL GALLERIES SPACES
STAIRS
WC
SHOP
CAFE
SEMINAR AUDITORIUM
STAIRS
WC
GALLERY
ESPRESSO BAR
III.
GALLERY
SHOP
STAIRS
WC
SMALL GALLERIES SPACES
IV.
STAIRS
MEMBERS FLOOR
V.
STAIRS
WC
TOP
VI.
EAST ROOM
GALLERY
BAR


MOMA
WC
THEATRE
CLASSROOMS
THEATRE
WC
SHOP
FILM DESK
TICKETS
INFO
MAIN ENTR.
STAIRS
INFO
STORE
THEATRE
WC
ATRIUM
WC
SCULPTURE GARDEN
RESTAURANT
CLASSROOM
EDUCATION CENTRE
STAIRS
EDUCATION CENTRE
GALLERY
WC
INFO
GALLERIES
WC
INFO
GALLERY
WC
INFO
CAFE
GALLERY
WC
STORE
SPECIAL EXHIBITION


• NO TICKETS
• CLOAKROOM
• TOILETS
• BACKUP SPACES (SEMINAR, EDUCATION ...)
• NO SHOP
• CAFE, RESTAURANT, BAR
• ROOF TERRACE, SCULPTURE GARDEN?, LOOKOUT
ONE ROOM DOES IT ALL.

Vienna, AT 15/12/2010

3. Session Alsop:

why do we need this?

What is the benefit?

What is better here than in an ordinary gallery space?

Answering questions and reasons. Show us the vision.

Scenarios for kurators, owners, artists and activists.

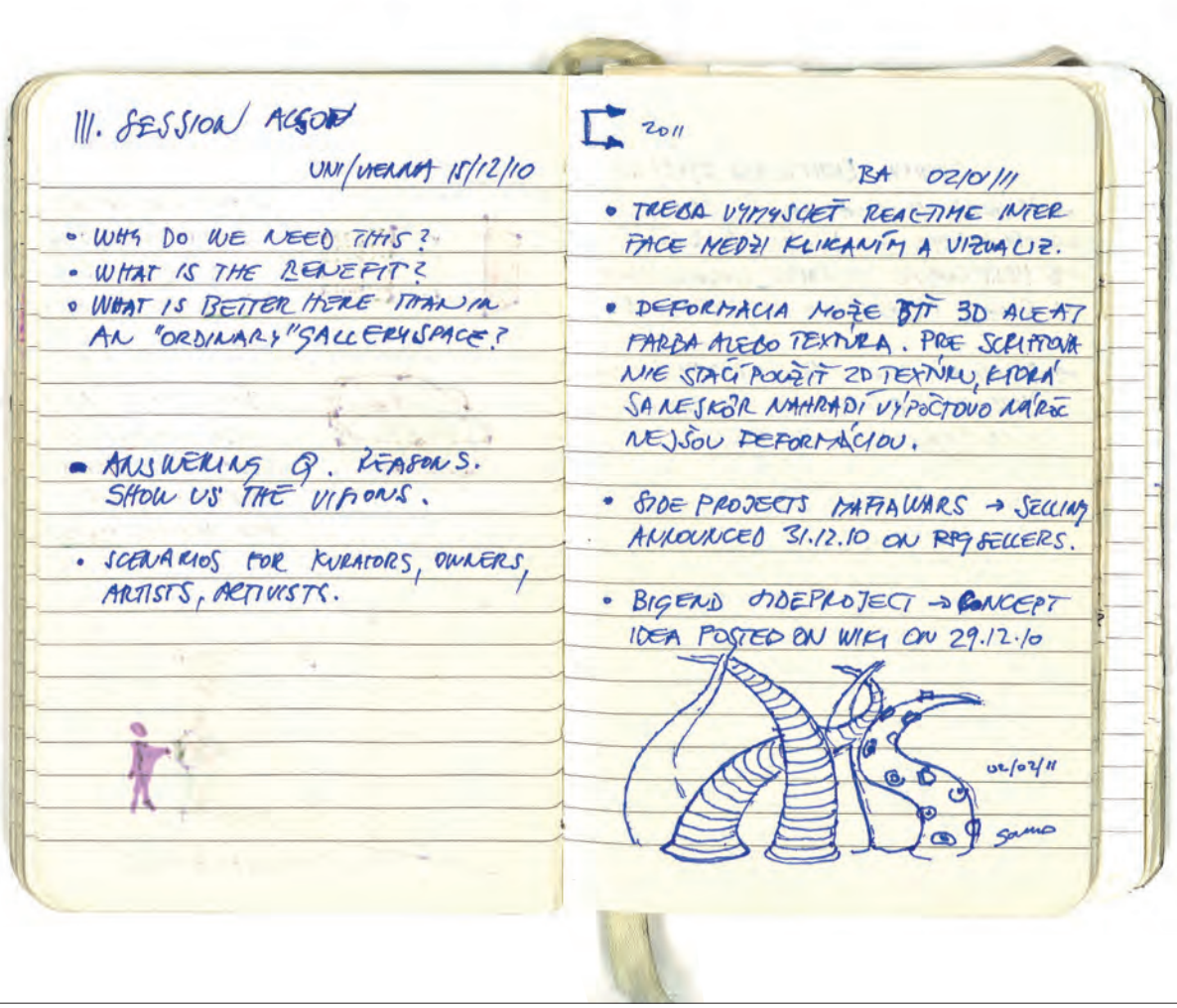
Bratislava, SK 02/01/2011

A realtime interface between clicking and visualisation is needed.

The deformation could be 3D or colour or texture. For the script a 2D texture, which will later be replaced by computational more difficult deformation, is enough.

somewhere 02/02/2011

the idea of the tentacles for the first time



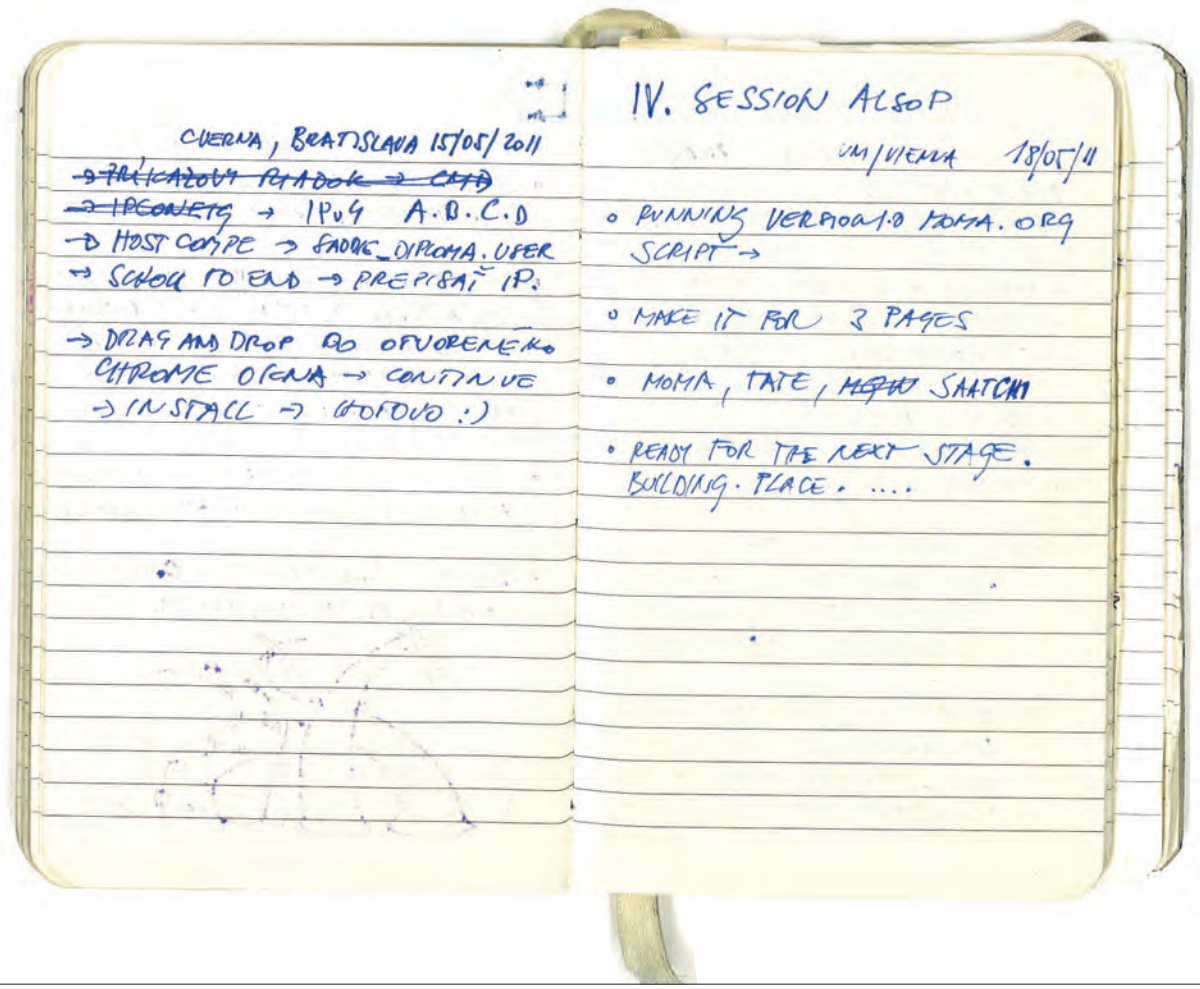
Vienna, AT 18/05/2011

4. Session Alsop:

running version of moma.org script

make it for 3 pages (MoMA, Tate, Saatchi)

ready for the next stage → building, place



19/10/2011 Rem Koolhaas on Ch. Rose

‚Architecture is about performance. The building is not about how it looks or functions rather about how it performs.'

Cverna, Bratislava, SK 21/10/2011

boundary, wall, limits → space restrictions

to manipulate space means to manipulate its limits and laws. This is how the public user of space is operating and always has. Now, we have to consider that this has not changed. What has changed though, is what constitutes space, which limits and laws. By manipulating these we will manipulate the space itself. And exactly to show this manipulation is what I'm thinking to do.

By using space we stretch the limits of it.

Šenkvice, SK 12/11/2011

Books for presentation

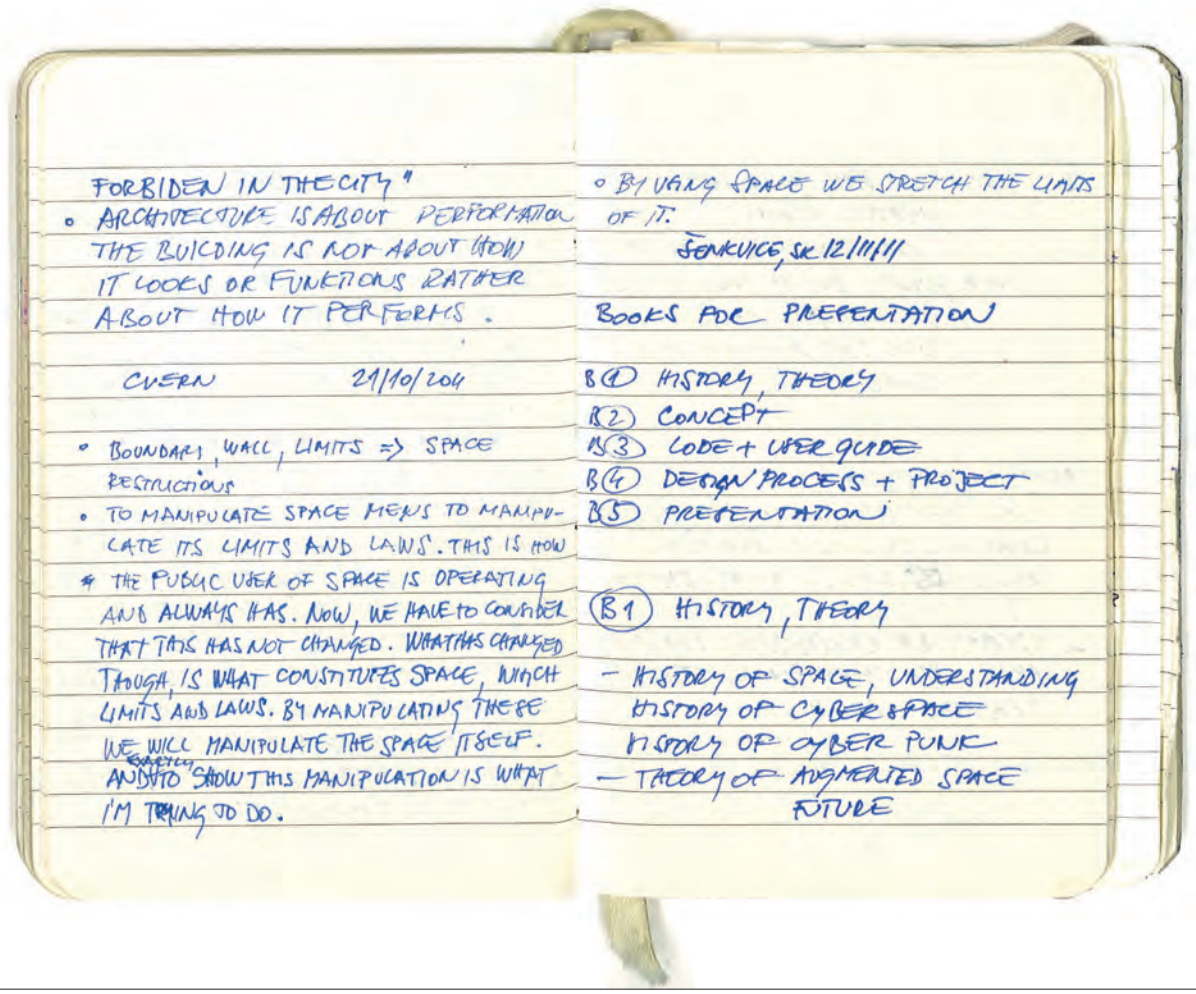
B1 – history, theory

B2 – concept

B3 – code + user guide

B4 – design process + project

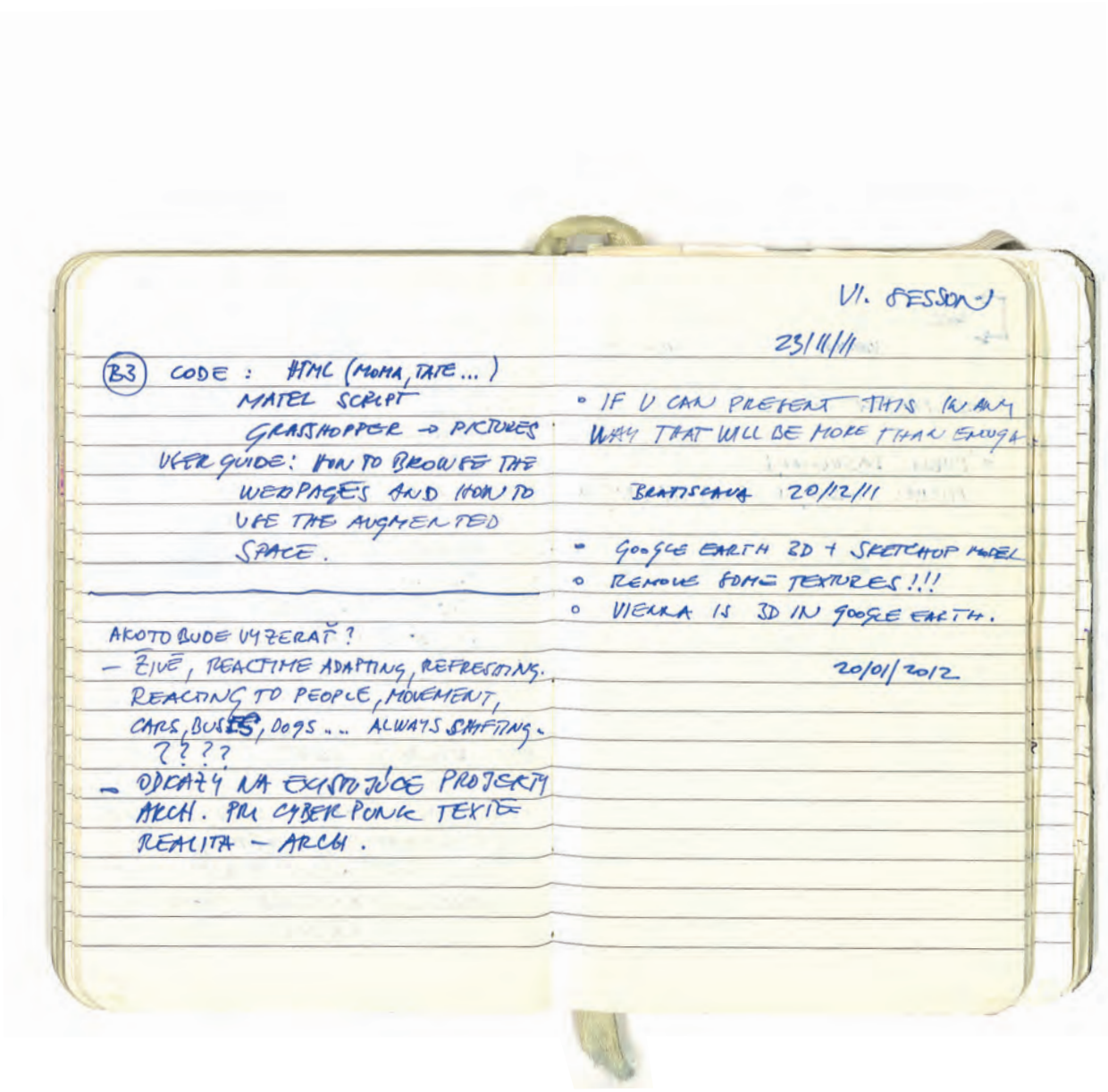
B5 – presentation

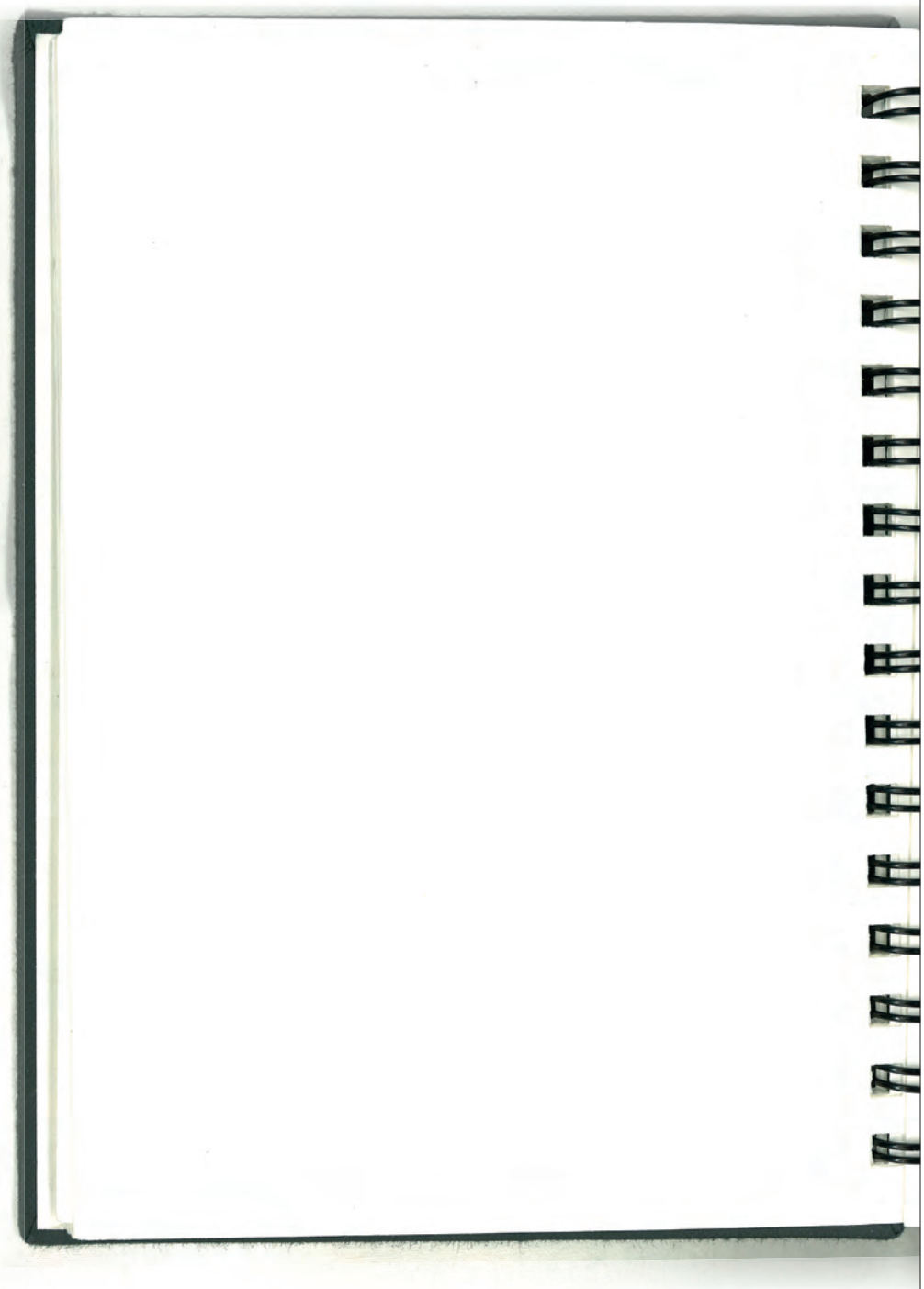


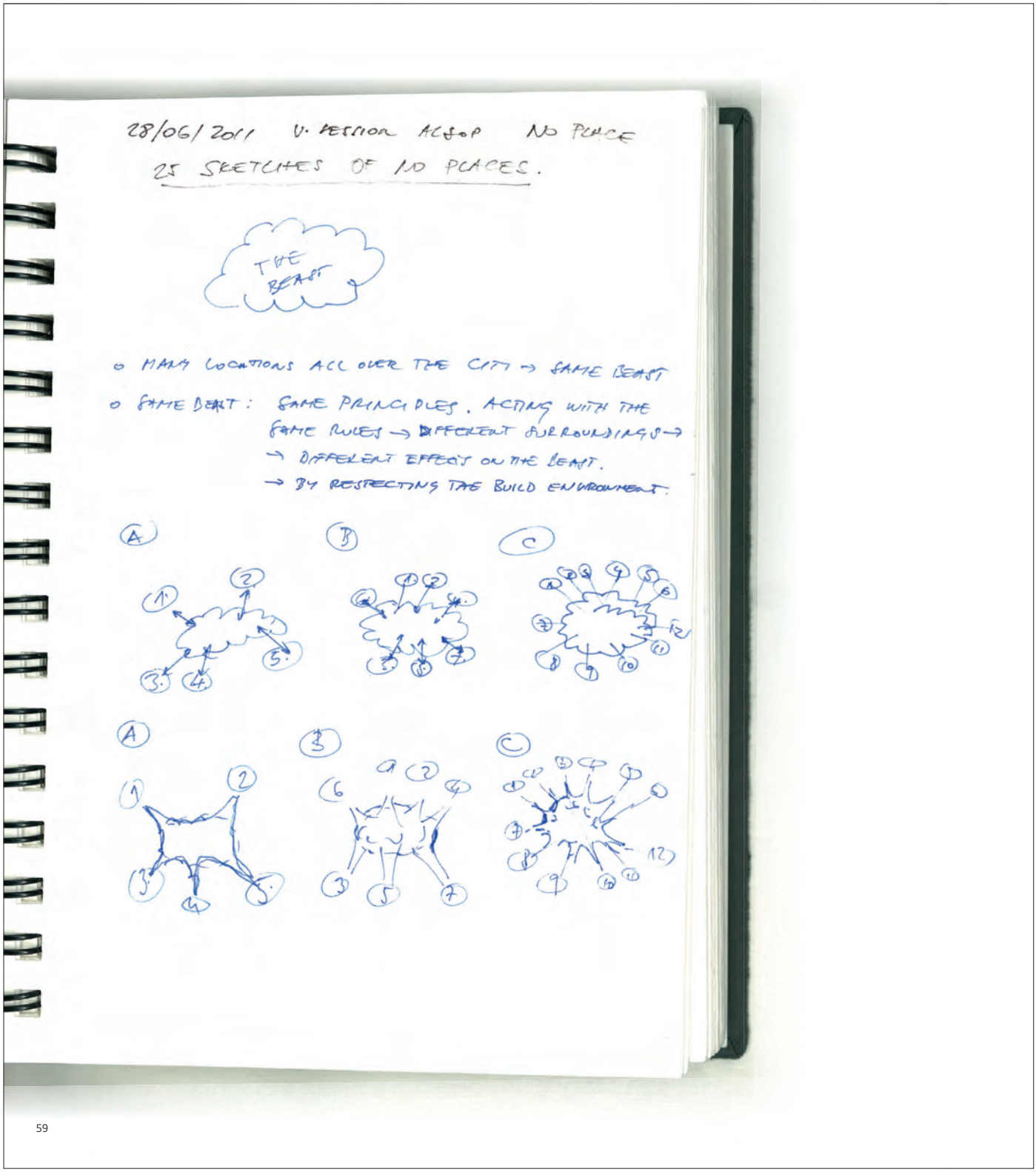
Vienna, AT 23/11/2011

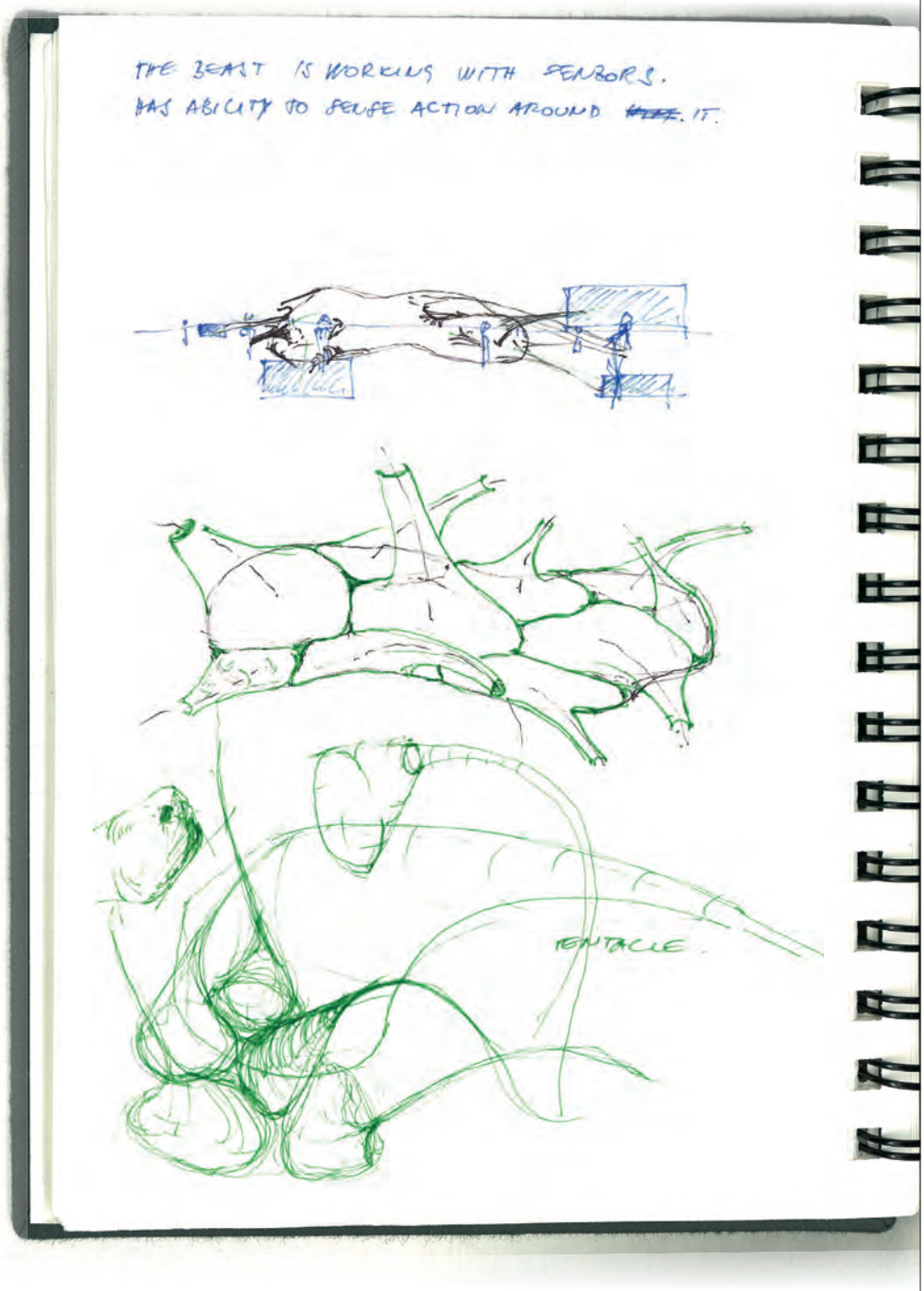
6. Session Alsop:

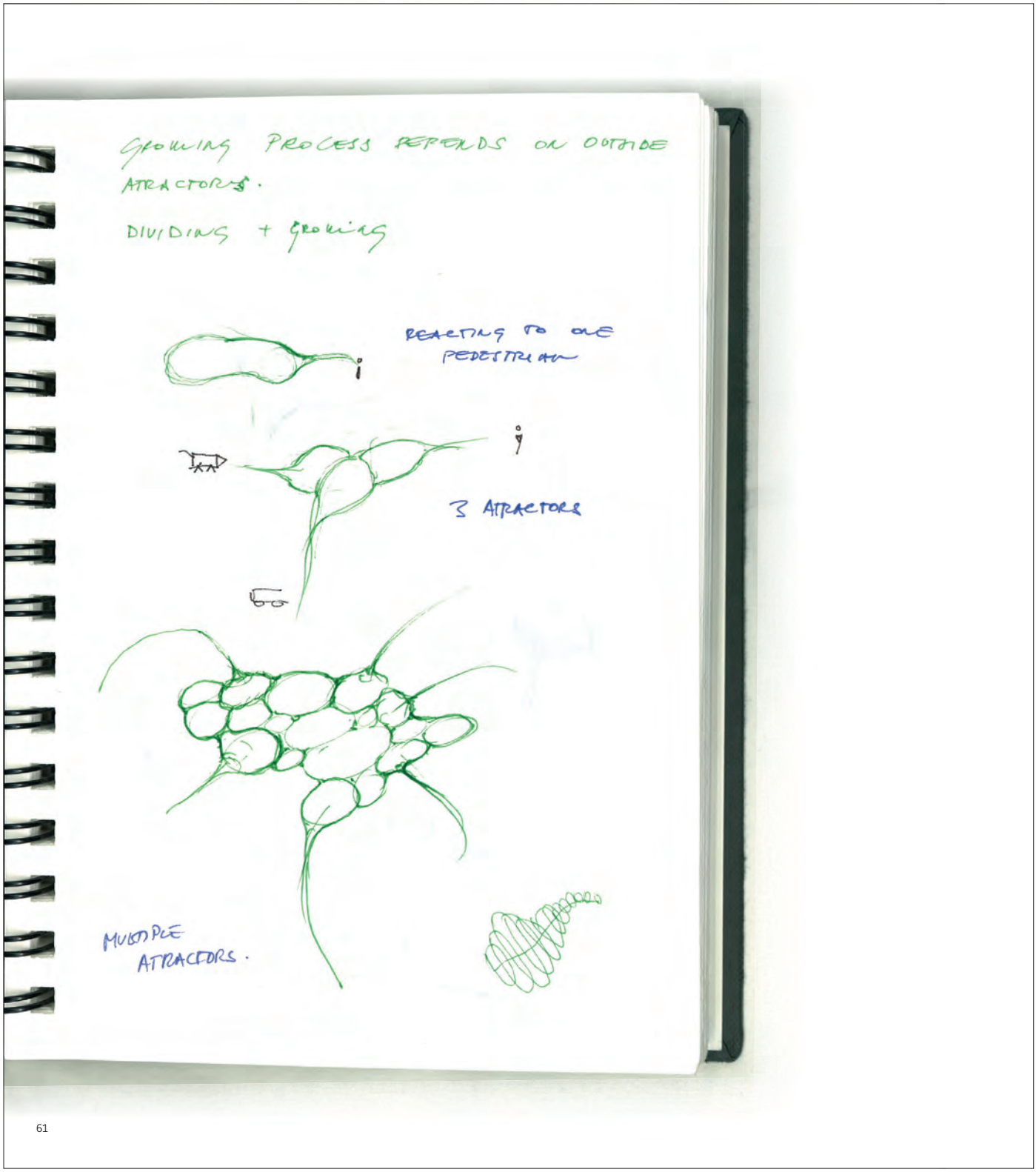
if u can present this in any way that will be more than enough

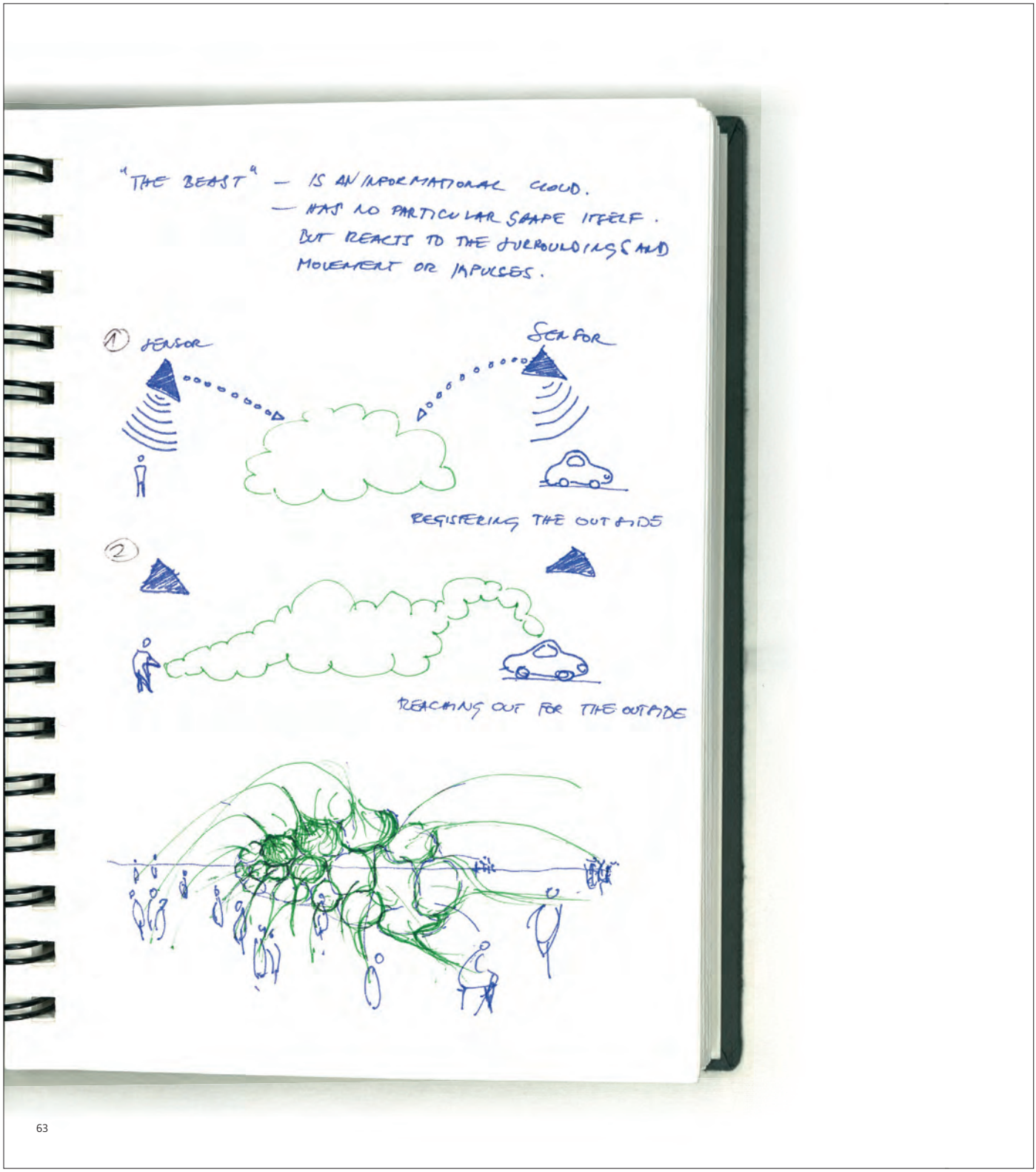





28/06/2011 V. PESTION ACSOP NO PLACE
25 SKETCHES OF NO PLACES.
THE BEAST
o MANY LOCATIONS ALL OVER THE CITY → SAME BEAST
o SAME BEAST: SAME PRINCIPLES. ACTING WITH THE SAME RULES → DIFFERENT SURROUNDINGS →
→ DIFFERENT EFFECTS ON THE BEAST.
→ BY RESTRICTING THE BUILD ENVIRONMENT.
A
B
C
A
B
C


THE BEAST IS WORKING WITH SENSORS.
HAS ABILITY TO SENSE ACTION AROUND ~~HIM~~. IT.
TENTACLE


GROWING PROCESS DEPENDS ON OUTSIDE ATTRACTORS.
DIVIDING + GROWING
REACTING TO ONE PEDESTRIAN
3 ATTRACTORS
MULTIPLE ATTRACTORS.


"THE BEAST" – IS AN INFORMATIONAL CLOUD.
– HAS NO PARTICULAR SHAPE ITSELF.
BUT REACTS TO THE SURROUNDINGS AND
MOVEMENT OR IMPULSES.
1 SENSOR
SEA FOR
REGISTERING THE OUTSIDE
2
REACHING OUT FOR THE OUTSIDE

- INTERACTION BETWEEN THE BEAST AND THE USER → USER BROWSES THE CYBERSPACE AND SELECTS THE DESIRED INFORMATION → PAGE → ARTIST INSTALLATION
- THE BEAST → DISPLAYS THE ART INSTALLATION VIA AUGMENTED REALITY IN THE SPACE AROUND

(3)

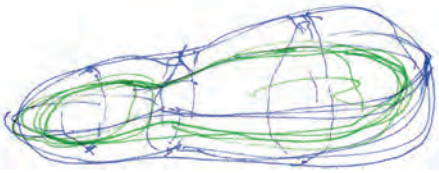
- 4 DIFFERENT "PLACES" ACTIVATE INSIDE OF THE INFORMATIONAL CLOUD.

INTERACTION – COMMUNICATION

(4)

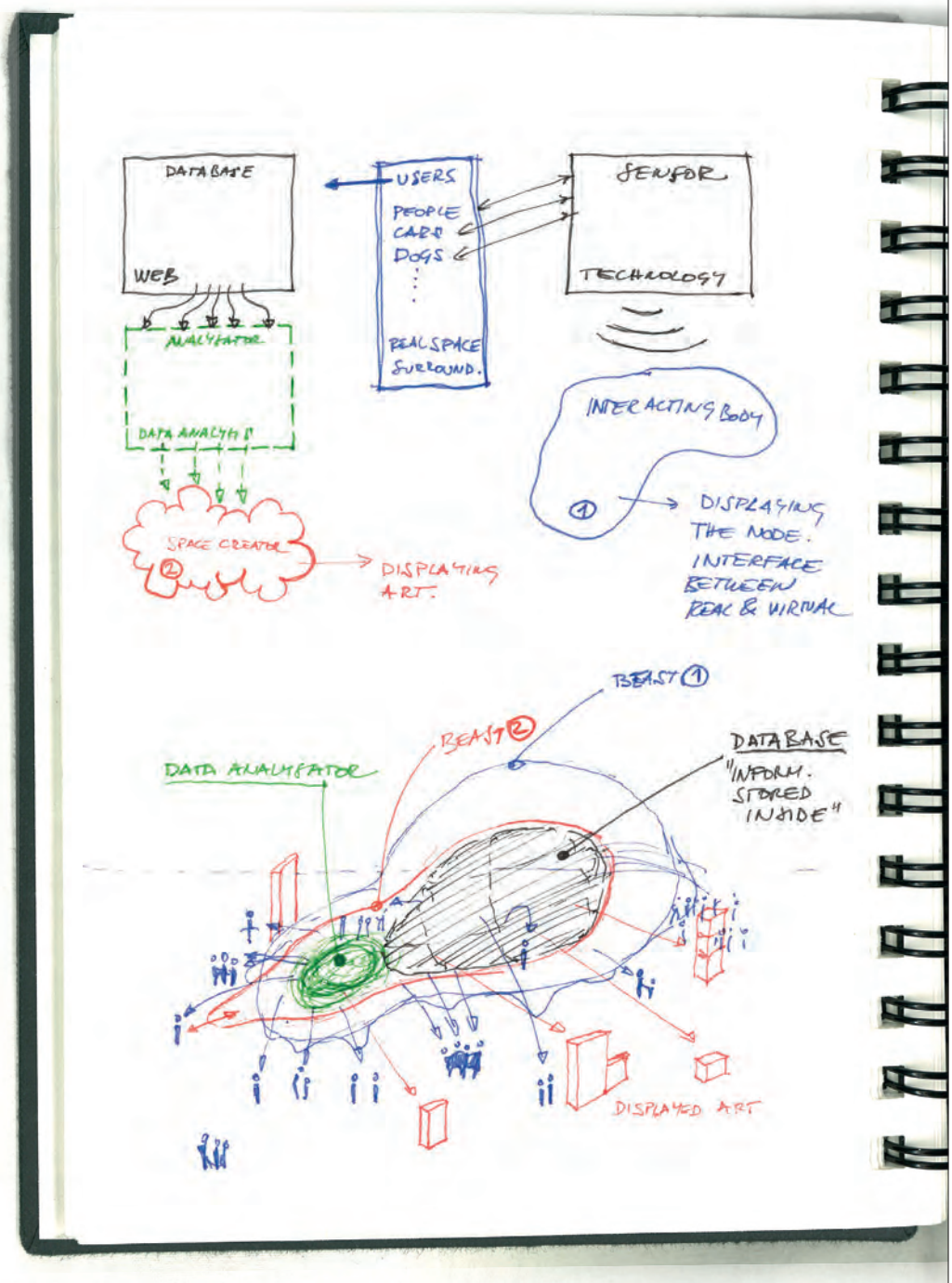
"LOVE" IS ANSWER.

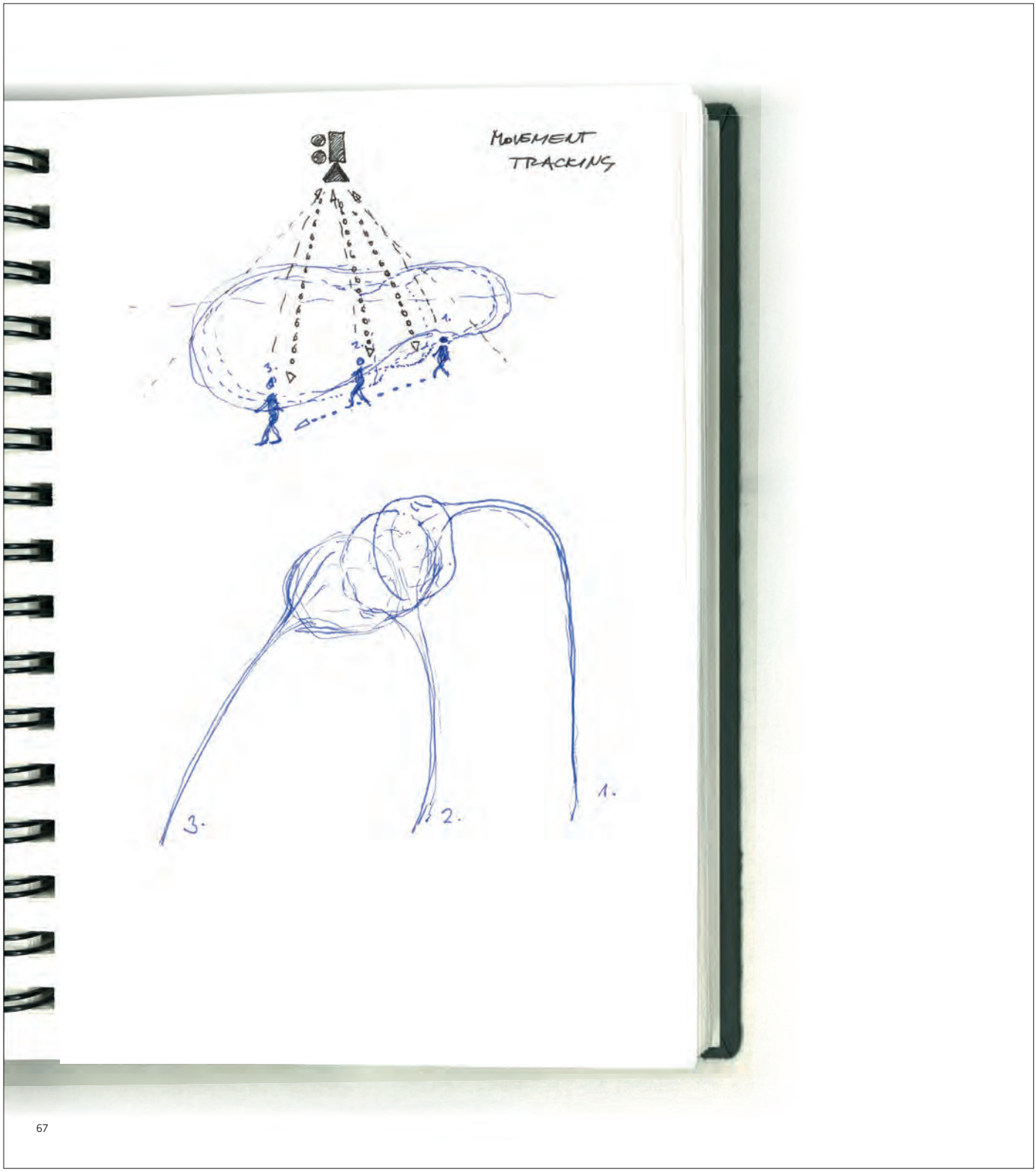
WHAT IF THERE ARE 2 OBJECTS?



- 1 FOR THE INTERACTION WITH THE SURROUNDING (PEOPLE, CARS, DOGS...) FORMS AND DEFORMS WITH REAL TIME MOVEMENT.
- 1 FOR THE WEB – DATABASE REPRESENTATION TO DEMONSTRATE THE INFORMATION STORED INSIDE FORMS AND DEFORMS ACCORDING TO THE "BROWSING" ACTION OF THE ACTUAL USER.

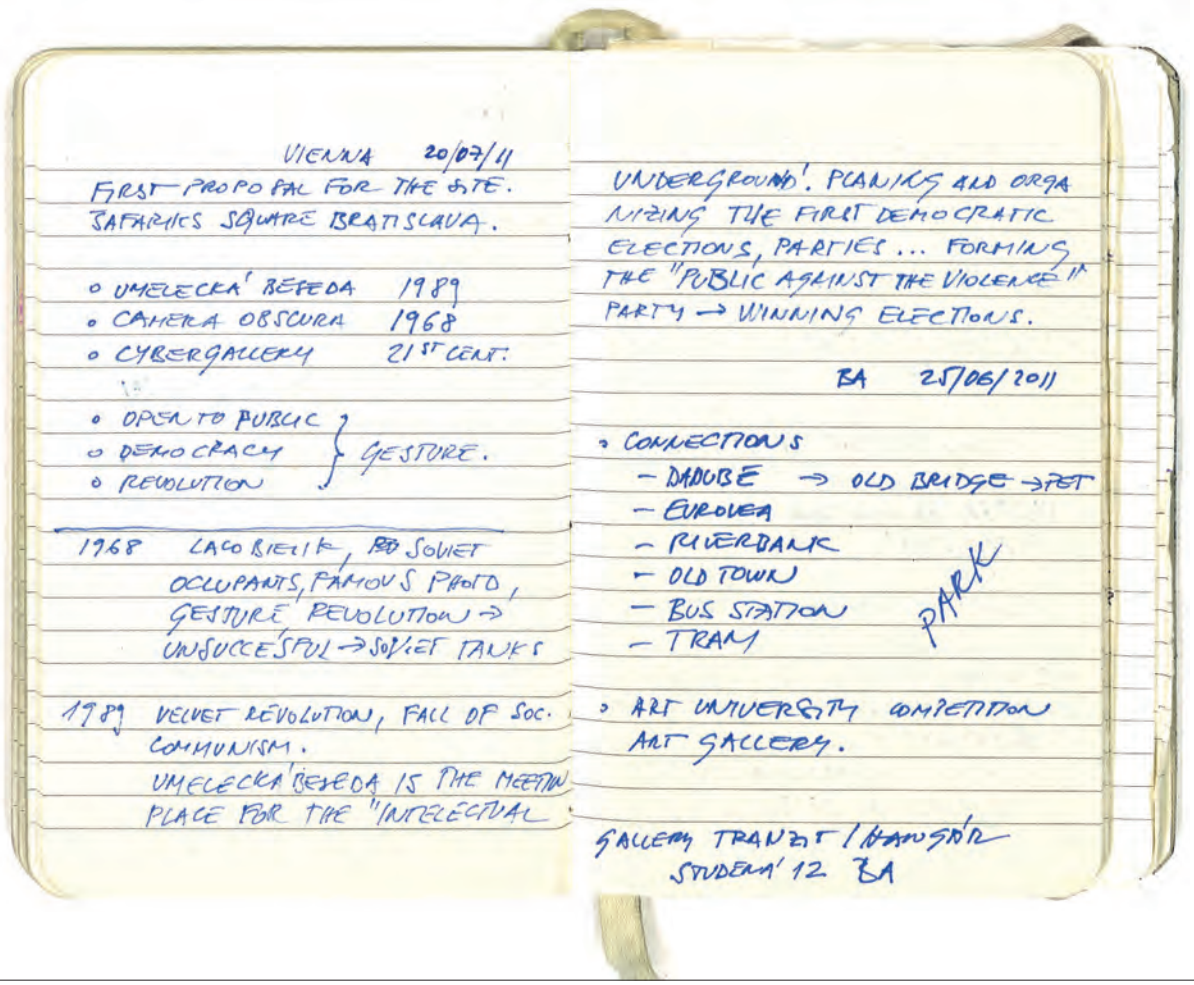
— BOTH EXISTING ON TOP OF EACH OTHER ....


DATABASE
WEB
ANALYSATOR
DATA ANALYSIS
SPACE CREATOR
DISPLAYING ART.
USERS
PEOPLE
CARS
DOGS
REALSPACE SURROUND.
SENSOR
TECHNOLOGY
INTERACTING BODY
DISPLAYING THE NODE. INTERFACE BETWEEN REAL & VIRTUAL
BEAST 1
BEAST 2
DATA ANALYSATOR
DATABASE "INFORM. STORED INSIDE"
DISPLAYED ART


MOVEMENT
TRACKING
1.
2.
3.
3.
2.
1.

Vienna, AT 20/07/2011

First proposal for the site. Šafárik square Bratislava.



Vienna, AT 28/06/2011

5. Session Alsop:

no place? :-)

25 sketches of no places

it's not about a site or building per se

Pohoda festival, SK 08/07/2011

Pattern recognition through repetition.

Cverna, Bratislava, SK 21/10/2011

Exploring space and informational

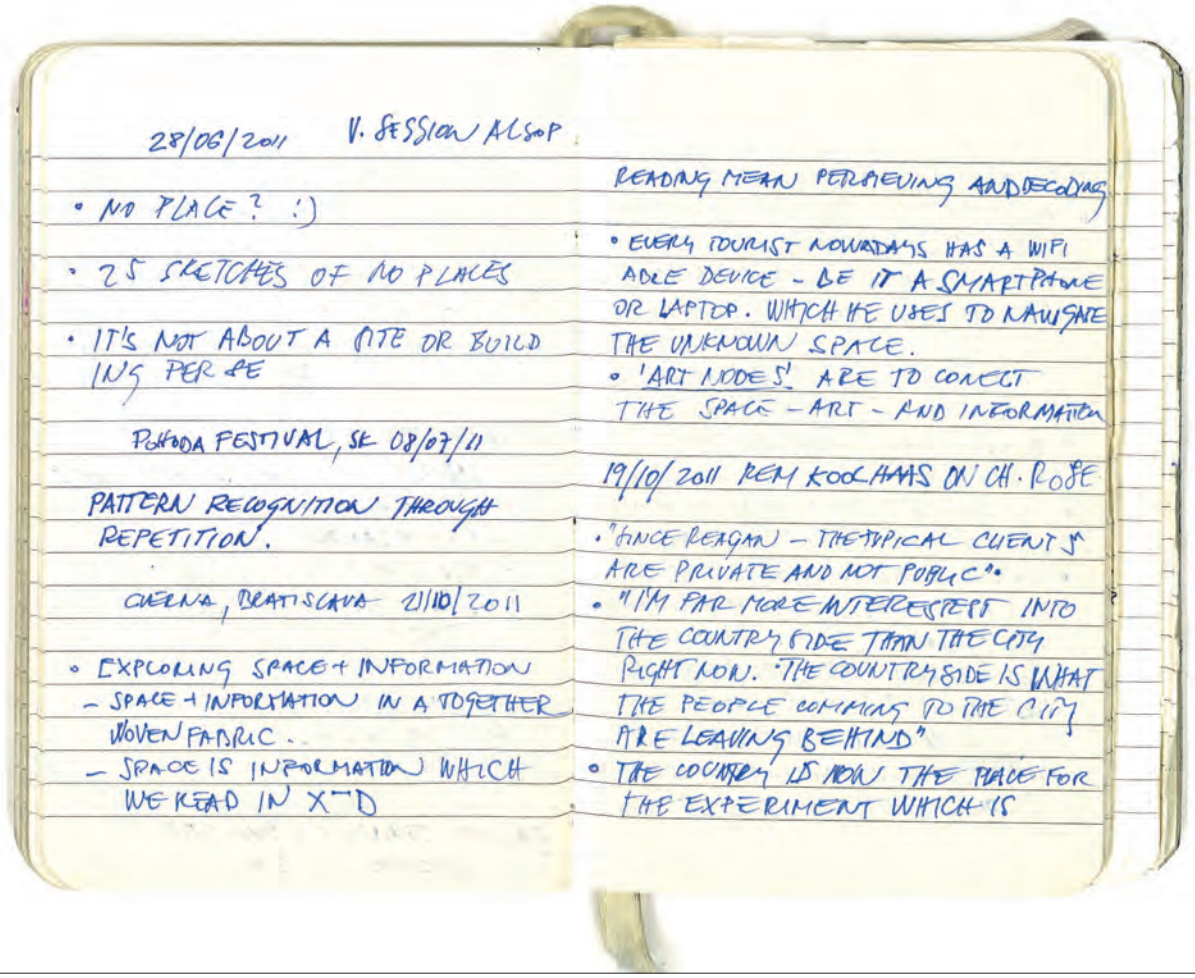
space + information in a together woven fabric

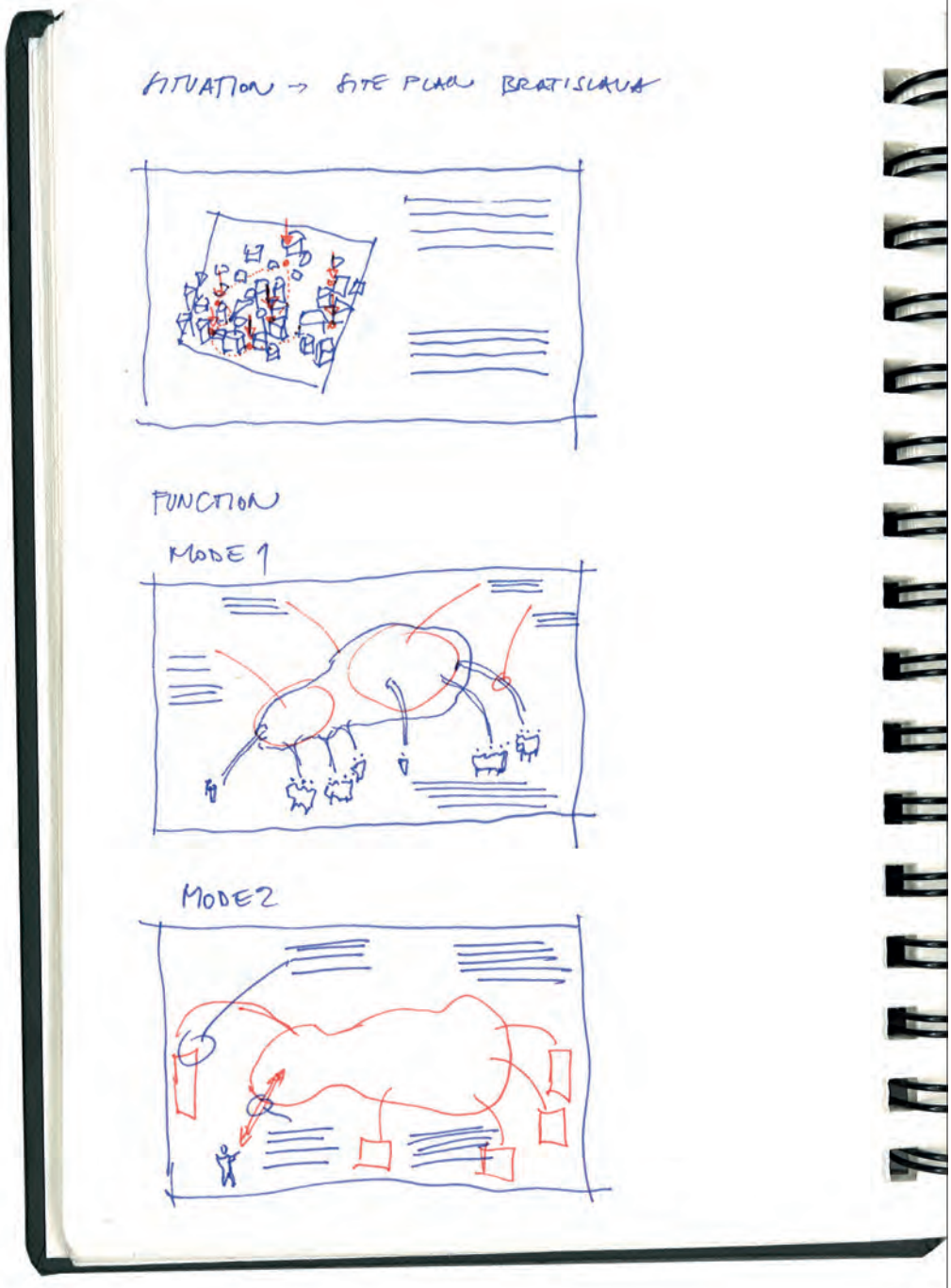
space is information which we read in XD.

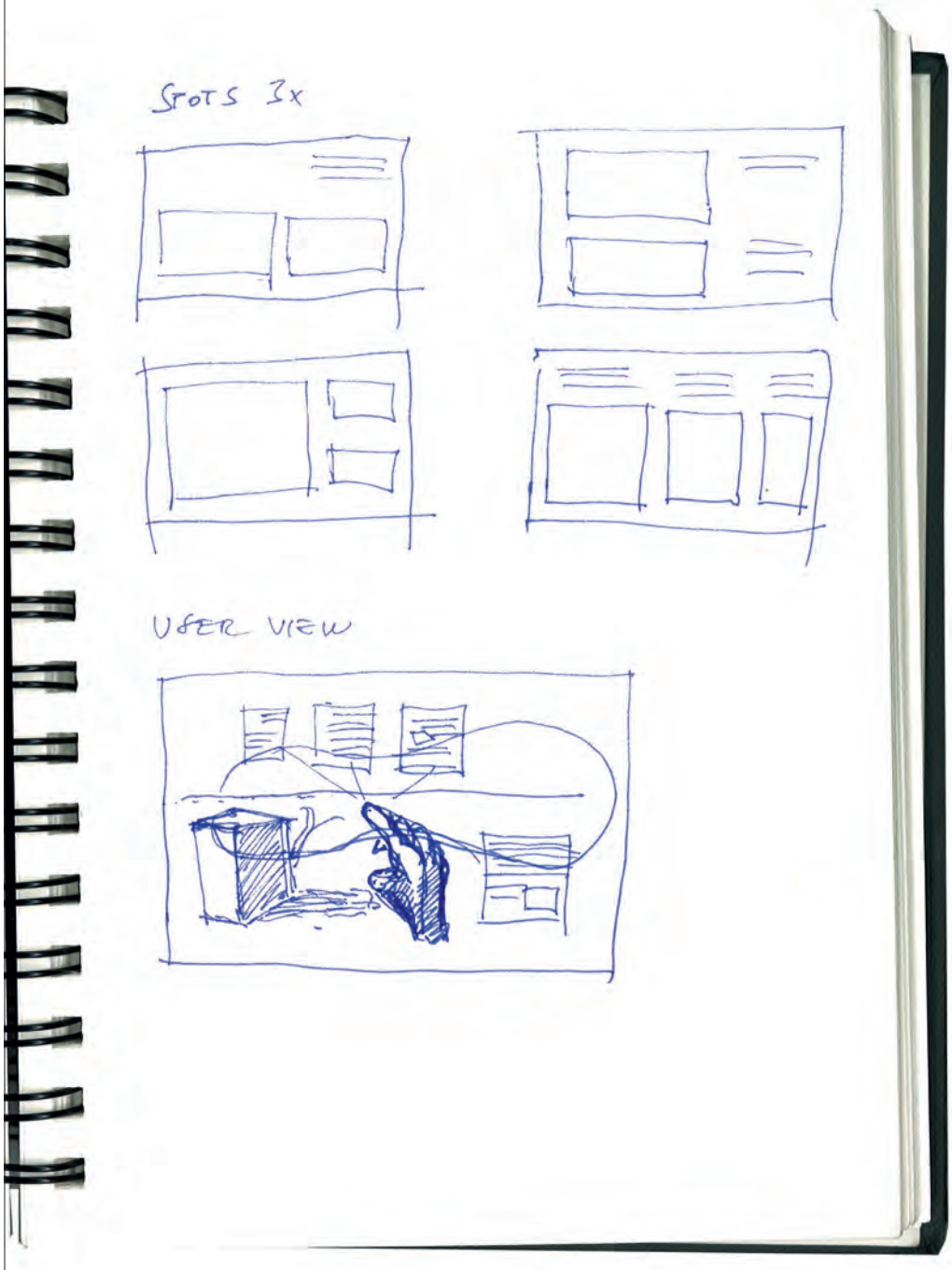
Reading means observing and decoding

Every tourist nowadays has a WIFI able device – be it a smartphone or a laptop. Which he uses to navigate the unknown space.

Art-nodes are to connect the space – art – information.




SITUATION → SITE PLAN BRATISLAVA
FUNCTION
MODE 1
MODE 2


SLOTS 3x
USER VIEW

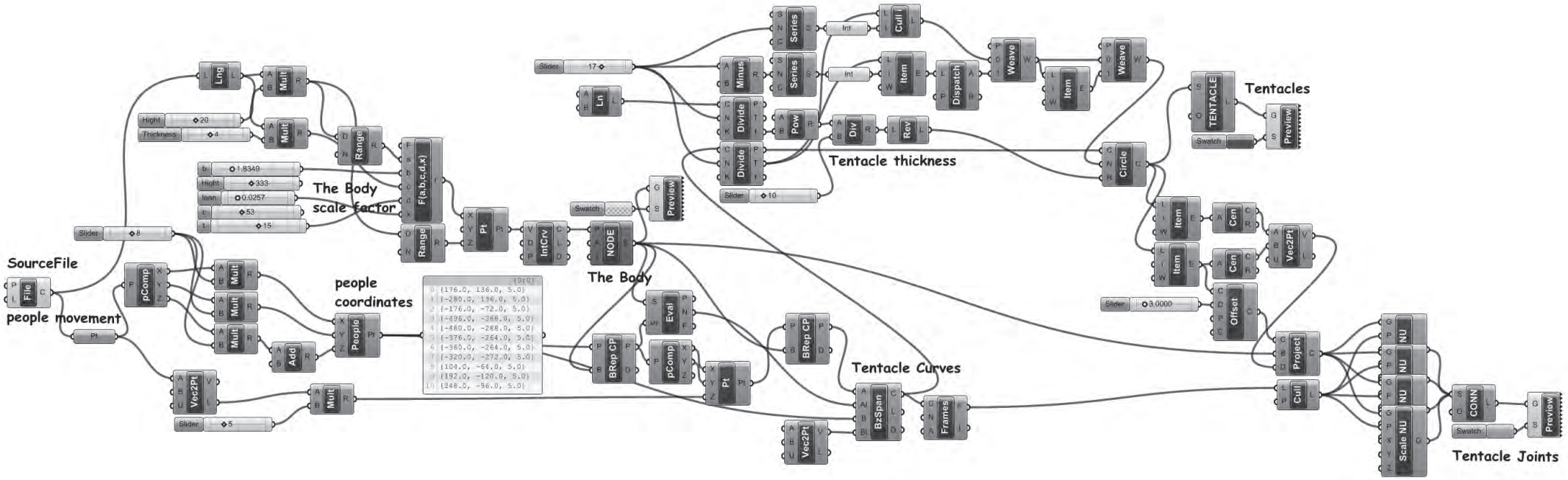
Book 1 – Analysis of History and Theory regarding cyberspace-architecture and cyberpunk
Book 2 – Vocabulary of cyberspace-architecture and cyberpunk
Book 3 – Notes and sketches
Book 4 – Scripts and codes
Book 5 – Presentation

Cyberpunk 門 Architecture

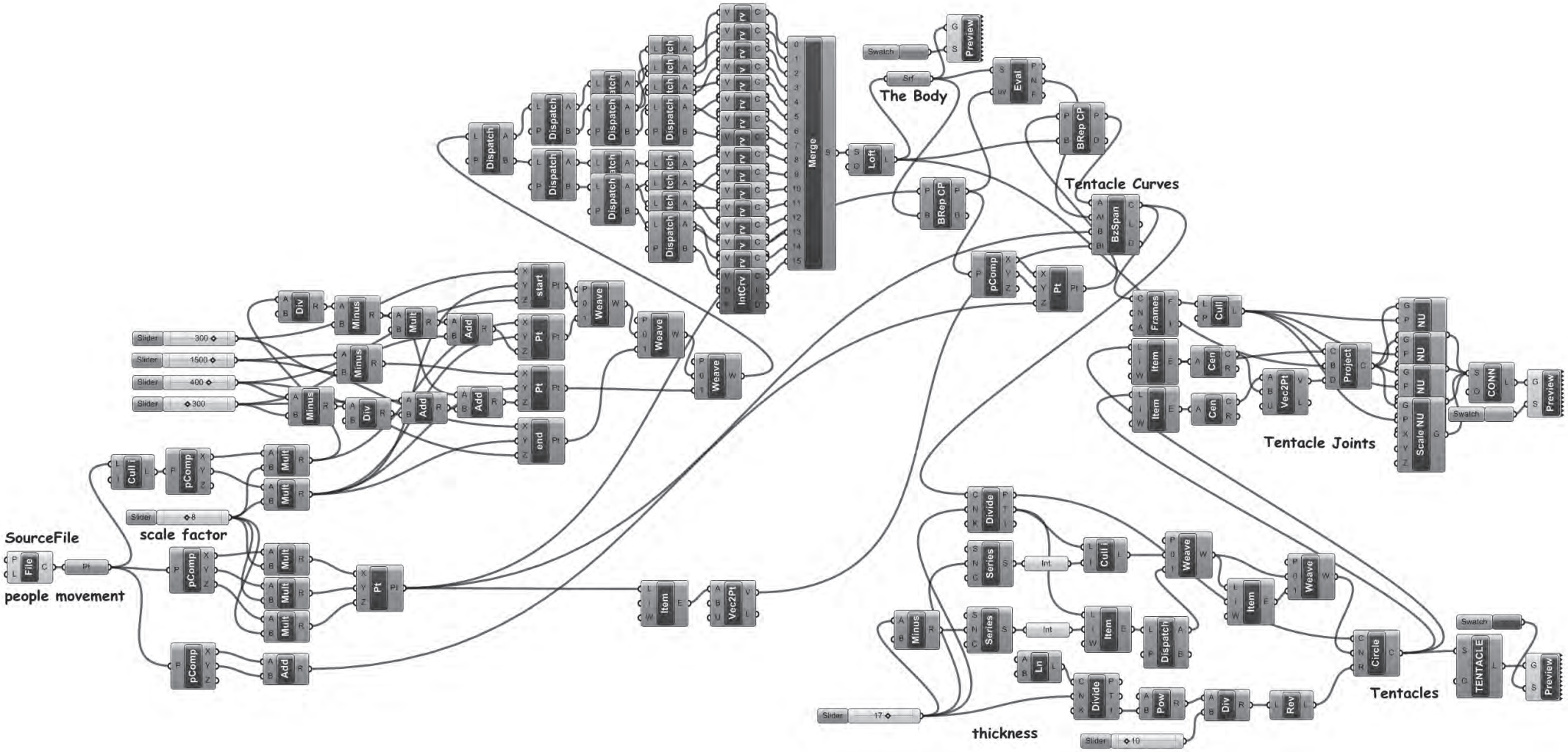
A humble attempt to give visual expression to new models of reality by Samo Bertok

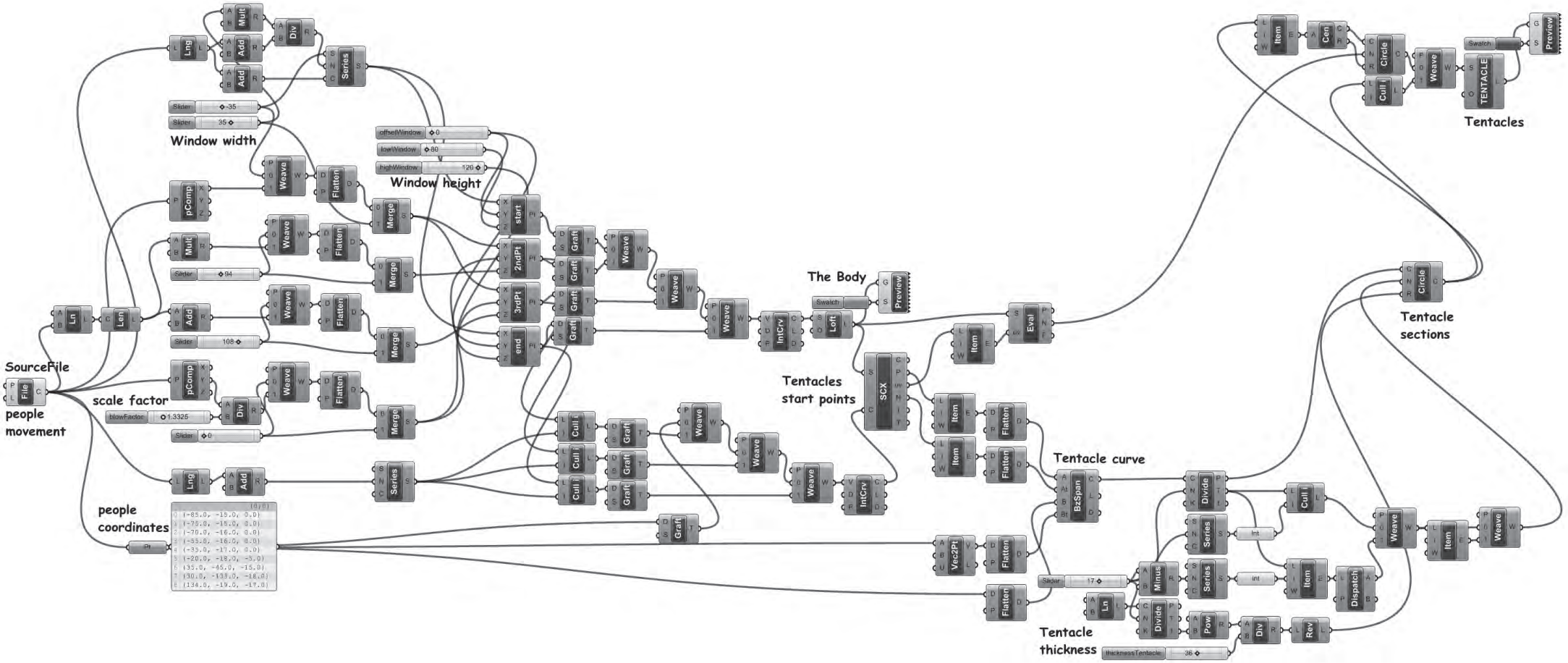
Book 4 – Scripts and codes

Grasshopper Script for the **Pin Type** Image Creator

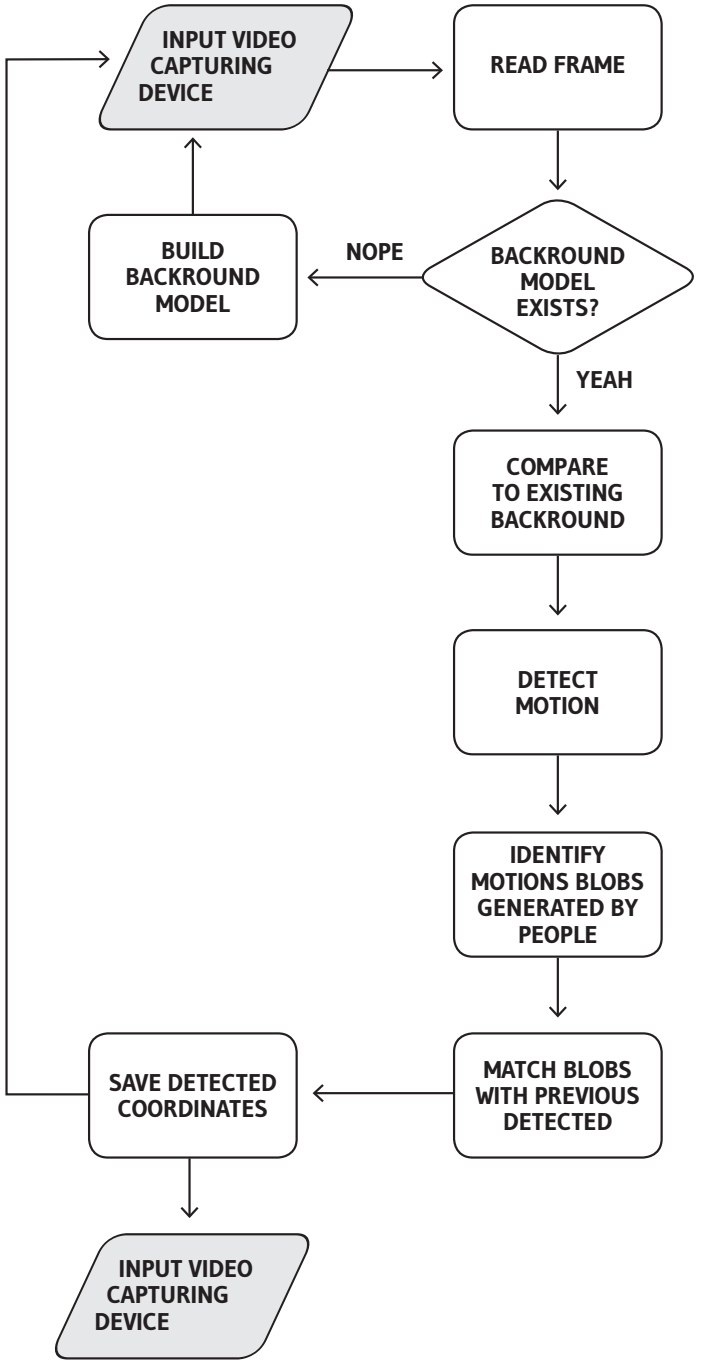


Grasshopper Script for the **Hammock Type** Image Creator




Window width
Window height
SourceFile
people movement
scale factor
people coordinates
The Body
Tentacles start points
Tentacle curve
Tentacle thickness
Tentacle sections
Tentacles

Flowchart of the **Movement Analyzer** Script

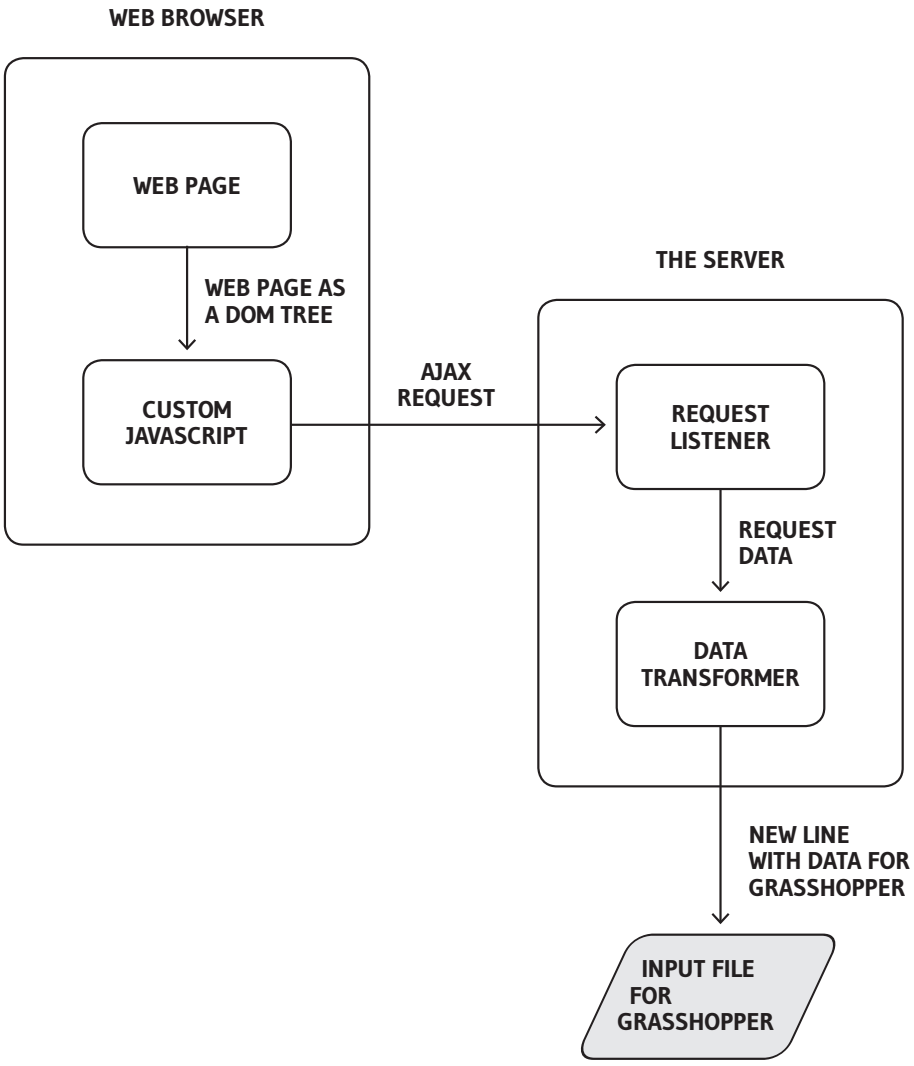


Movement Analyzer Script by Zdenka Šturcová

```
#include „stdafx.h“
using namespace std;
CVideoFilter::CVideoFilter(BYTE treshold)
{
  Graphics = new CVectorGraphics(CVideoToolkitInterface::vfRGB);
  SetTreshold(treshold);
  pFile = fopen („D:/example.txt“,“w“);
  LastFrame = new unsigned char[1];
  Background = new unsigned char[1];
  DifferenceBGR = new unsigned char[1];
  DifferenceChromaGR = new float[1];
  DifferenceHue = new int[1];
  Motion = new int[1];
  Mhi = new int[1];
  Hue = new int[1];
  ChromaGR = new float[1];
  ChromaMi = new float[1];
  ChromaVar = new float[1];
  ChromaMatch = new bool[1];
  ChromaGRprev = new float[1];
  Background = new unsigned char[1];
  PeaksX = new int[1];
  PeaksY = new int[1];
  marksH = new float[1];
  Door[0].x = 10;  Door[0].y = 205; Door[0].radius = 70;
  Door[1].x = 620; Door[1].y = 390; Door[1].radius = 70;
  Door[2].x = 640; Door[2].y = 50; Door[2].radius = 70;
  Door[3].x = 50; Door[3].y = 50; Door[3].radius = 70;
  InitPeople(10);
  PeopleCount = 0;
}
CVideoFilter::~CVideoFilter(void)
{
  delete Graphics;
  fclose(pFile);
  delete [] LastFrame;
  delete [] Background;
  delete [] DifferenceBGR;
  delete [] DifferenceChromaGR;
  delete [] DifferenceHue;
  delete [] Motion;
  delete [] Mhi;
  delete [] Hue;
  delete [] ChromaGR;
  delete [] ChromaMi;
  delete [] ChromaVar;
  delete [] ChromaMatch;
  delete [] ChromaGRprev;
  delete [] Background;
  delete [] PeaksX;
  delete [] PeaksY;
  DestroyPeople(10);
  delete [] marksH;
}
void
CVideoFilter::InitPeople(int num)
{
  Person = new People[num];
  for (int k=0; k<num; k++)
  {
    Person[k].time = 0;
    Person[k].maxtraj = 500;
    Person[k].trajectoryX = new float[Person[k].maxtraj];
    Person[k].trajectoryY = new float[Person[k].maxtraj];
    Person[k].idle = 0;
  }
  for (int k=0; k<50; k++)
    Colors[k] = CVectorGraphics::CColor((rand()%255)/255.0,
(rand()%255)/255.0, (rand()%255)/255.0);
}
void
CVideoFilter::DestroyPeople(int num)
{
  for (int k=0; k<num; k++)
  {
    delete []  Person[k].trajectoryX;
    delete []  Person[k].trajectoryY;
  }
  delete [] Person;
}
void
CVideoFilter::Activate(int width, int height)
{
  Graphics->SetSize(width, height);
  Width = width; Height = height;
  Width2 = width/2; Height2 = height/2;
  Width4 = width/4; Height4 = height/4;
  delete [] LastFrame; LastFrame = new unsigned
char[Width*Height*3];
  delete [] Background; Background = new unsigned
char[Width4*Height4*3];
  delete [] DifferenceBGR; DifferenceBGR = new unsigned
char[Width4*Height4*3];
  delete [] DifferenceChromaGR; DifferenceChromaGR = new
float[Width4*Height4*2];
  delete [] DifferenceHue; DifferenceHue = new int[Width4*Height4];
  delete [] Motion; Motion = new int[Width4*Height4];
  delete [] Mhi; Mhi = new int[Width4*Height4];
  delete [] Hue; Hue = new int[Width4*Height4];
  delete [] ChromaGR; ChromaGR = new float[Width4*Height4*2];
  delete [] ChromaMi; ChromaMi = new float[Width4*Height4*2];
  delete [] ChromaVar; ChromaVar = new float[Width4*Height4*2];
  delete [] ChromaMatch; ChromaMatch = new bool[Width4*Height4];
  delete [] ChromaGRprev; ChromaGRprev = new float[Width4*Height4*2];
  DestroyPeople(10);
  InitPeople(10);
  PeopleCount = 0;
  delete [] marksH; marksH = new float[5000];
  for (int k=0; k<5000; k++)
    marksH[k]=0;

  pFile = fopen („D:/myfile.txt“,“w“);
}
float
CVideoFilter::Dist2D(int x1, int y1, int x2, int y2)
{
  return sqrt((float)abs(x1-x2)*abs(x1-x2)+abs(y1-y2)*abs(y1-y2));
}
void
CVideoFilter::SetTreshold(BYTE treshold)
{
  Treshold = treshold;
}
void
CVideoFilter::MaxFromInterval(float * x, int from, int to, Extreme
&pex)
{
   pex.value = x[from];
   pex.index = from;
   for (int k=from; k<=to; k++)
     if (pex.value < x[k])
     {
       pex.value = x[k];
       pex.index = k;
     }
}
void
CVideoFilter::MinFromInterval(float * x, int from, int to, Extreme
&pex)
{
   pex.value = x[from];
   pex.index = from;
   for (int k=from; k<to; k++)
     if (pex.value > x[k])
     {
       pex.value = x[k];
       pex.index = k;
     }
}
float
CVideoFilter::SumInterval(float * x, int from, int to, float tresh)
{
  float sum = 0;
  for (int k=from; k<to; k++)
    if (x[k] > tresh) sum += x[k];
  return sum;
}
int
CVideoFilter::FindPeaks(int * peaks, float * a, int a_length, float
min_height)
{
  int peakcount = 0;
  int min_vzd = 30;
  bool allzeros = true;
  float * b = new float[a_length];
  // check if there is a non-zero value
  Extreme maxP;
  MaxFromInterval(a, 0, a_length, maxP);
  // clear everything that is less than 1/2 of biggest peak
  for (int k=0; k<a_length; k++)
    b[k] = (a[k] >= maxP.value*min_height) ? a[k] : 0;
  for (int k=0; k<a_length; k++)
            if (b[k]>0){ allzeros = false; k=a_length;}

  while (!allzeros)
  {
    Extreme maxP;
    MaxFromInterval(b, 0, a_length, maxP);
                peaks[peakcount] = maxP.index;
      peakcount++;
            int k = maxP.index;
            b[k] = 0;
            bool klesa = true;
            float klesa_prev = maxP.value;
            k--;
      while ((klesa) && (k>=0))
            {
                  if ((b[k] <= klesa_prev)||(maxP.index-k <
min_vzd))
                  {
                        klesa = true;
                        klesa_prev = b[k];
                        b[k] = 0;
                  }
        k--;
            }
            k = maxP.index;
            klesa = true;
            klesa_prev = maxP.value;
      k++;
            while ((klesa) && (k<=a_length-1))
            {
                  if ((b[k] <= klesa_prev)||(k-maxP.index <
min_vzd))
                  {
                        klesa = true;
                        klesa_prev = b[k];
                        b[k] = 0;
                  }
      k++;
            }
    allzeros = true;
    for (int k=0; k<a_length; k++)
    if (b[k]>0){ allzeros = false; k=a_length;}
  }
  delete [] b;
  return peakcount;
}
void
CVideoFilter::DrawGraph(float marginx, float marginy, float wid, float
hei, float * points, int point_len, float transp)
{
  marginx = marginx*Width;
  marginy = marginy*Height;
  wid = wid*Width;
  hei = hei*Height;

  Graphics->Rectangle(marginx, marginy, marginx+wid, marginy+hei,
      CVectorGraphics::CColor(1, 1, 1), 1.0,
CVectorGraphics::lsDefault, CVectorGraphics::CColor(1.0, 1.0, 1.0,
transp));

          Graphics->Rectangle(marginx, marginy, marginx+wid,
marginy+hei,
          CVectorGraphics::CColor(0, 0, 0), 1.0,
CVectorGraphics::lsDefault, CVectorGraphics::CColor(1.0, 1.0, 1.0,
0));
          // the graph is just one polygon, so we need to add first
and last point for correct visualisation
          double *pts = new double[(point_len+2)*2];
          pts[0] = marginx; pts[1] = hei+marginy;
          for (int k=0; k<point_len; k++)
          {
            pts[2*(k+1)] = (double)((float)k*wid/point_
len+marginx);
            pts[2*(k+1)+1] = (double)((1-points[k])*hei +
marginy);
          }
          pts[2] = marginx;
          pts[4] = marginx;
          pts[2*(point_len)] = wid+marginx;
          pts[2*(point_len+1)] = wid+marginx; pts[2*(point_
len+1)+1] = hei+marginy;
          // draw the graph line
          Graphics->Polygon(point_len+2, pts,
CVectorGraphics::CColor(), 1.0, CVectorGraphics::lsDefault,
CVectorGraphics::CColor(0.0,0.0,0.0,0.0));
  delete [] pts;
}
void
CVideoFilter::FindPeople()
{
  // calculating the rows and cols for motion
  float * rows = new float[Height4];
  float * cols = new float[Width4];
  for (int k=0; k < Height4; k++) rows[k] = 0;
  for (int k=0; k < Width4; k++) cols[k]=0;
  for (int k=0; k < Height4; k++)
    for (int j=0; j < Width4; j++)
    {
      rows[k] += ((ChromaMatch[k*Width4+j]) ||
(DifferenceBGR[3*(k*Width4+j)] > 70)) ? 1 : 0;
      cols[j] += ((ChromaMatch[k*Width4+j]) ||
(DifferenceBGR[3*(k*Width4+j)] > 70)) ? 1 : 0;
    }
  // median filter on rows/cols
  int median_width = 16;
  float * rowsmed = new float[Height4];
  float * colsmed = new float[Width4];
  for (int k=0; k < Height4; k++) rowsmed[k] = 0;
  for (int k=0; k < Width4; k++) colsmed[k]=0;
  for (int k=0; k < median_width/2; k++)
  {
    rowsmed[k] += rows[k]*median_width/2;
    colsmed[k] += cols[k]*median_width/2;
    rowsmed[Height4-k-1] += rows[Height4-k-1]*median_width/2;
    colsmed[Width4-k-1] += cols[Width4-k-1]*median_width/2;

  }
  for (int k=median_width/2; k < Height4-median_width/2; k++)
    for (int j=-median_width/2; j < median_width/2; j++)
      rowsmed[k] += rows[k+j];
  for (int k=median_width/2; k < Width4-median_width/2; k++)
    for (int j=-median_width/2; j < median_width/2; j++)
      colsmed[k] += cols[k+j];
  Extreme maxP;
  MaxFromInterval(rowsmed, 0, Height4, maxP);
  for (int k=0; k<Height4; k++)
    if ((maxP.value < median_width)|| (rowsmed[k] < median_width))
      rowsmed[k] = 0;
    else
      rowsmed[k] = rowsmed[k]/maxP.value;
  MaxFromInterval(colsmed, 0, Width4, maxP);
  for (int k=0; k<Width4;  k++)
    if ((maxP.value < median_width)|| (colsmed[k] < median_width))
      colsmed[k] = 0;
    else
      colsmed[k] = colsmed[k]/maxP.value;
  delete [] PeaksX; PeaksX = new int[Height4/10];
  int peakcount_row = FindPeaks(PeaksX, rowsmed, Height4, 0.1f);
  int * peaks_row = new int[peakcount_row];
  for (int k=0; k<peakcount_row; k++)
    peaks_row[k] = PeaksX[k];
  delete [] PeaksY; PeaksY = new int[Width4/10];
  int peakcount_col = FindPeaks(PeaksY, colsmed, Width4, 0.1f);
  int * peaks_col = new int[peakcount_col];
  for (int k=0; k< peakcount_col; k++)
    peaks_col[k] = PeaksY[k];
  float * peaks_sum = new float[peakcount_row*peakcount_col];
  int pers_radius = 10;

  for (int k=0; k<peakcount_row; k++)
    for (int j=0; j<peakcount_col; j++)
    {
      peaks_sum[k*peakcount_col+j]=0;
      int sum_count = 0;
      // count sums to find peaks
      for (int ki= max(peaks_row[k]-pers_radius,0); ki < min(peaks_
row[k]+pers_radius,Height4); ki++)
        for (int ji= max(peaks_col[j]-pers_radius,0); ji <
min(peaks_col[j]+pers_radius,Width4); ji++)
        {
          if ((ChromaMatch[ki*Width4+ji]) ||
(DifferenceBGR[3*(ki*Width4+ji)] > 70))
            peaks_sum[k*peakcount_col+j]++;
          sum_count++;
        }
      peaks_sum[k*peakcount_col+j] /= (float)sum_count;
    }
  Extreme maxSum;
  MaxFromInterval(peaks_sum, 0, peakcount_row*peakcount_col,
maxSum);
  while (maxSum.value > 0.05)
  {
    int max_row = peaks_row[maxSum.index/peakcount_col];
    int max_col = peaks_col[maxSum.index%peakcount_col];
    float * gradXFg = new float[pers_radius*pers_radius];
    float * gradYFg = new float[pers_radius*pers_radius];
    float * gradXBg = new float[pers_radius*pers_radius];
    float * gradYBg = new float[pers_radius*pers_radius];
    float * gradXDif = new float[pers_radius*pers_radius];
    float * gradYDif = new float[pers_radius*pers_radius];
    int kk = 0;
    for (int ki= max(max_row-pers_radius/2,0); ki < min(max_
row+pers_radius/2,Height4-1); ki++)
      for (int ji= max(max_col-pers_radius/2,0); ji < min(max_
col+pers_radius/2,Width4-1); ji++)
      {
        gradXBg[kk] = (float)((int)Background[3*(ki*Width4+ji)] -
(int)Background[3*(ki*Width4+ji+1)] +
                      (int)Background[3*(ki*Width4+ji)+1] - (int)
Background[3*(ki*Width4+ji+1)+1] +
                      (int)Background[3*(ki*Width4+ji)+2] - (int)
Background[3*(ki*Width4+ji+1)+2])/(255.f*3.f);
        gradYBg[kk] = (float)((int)Background[3*(ki*Width4+ji)] -
(int)Background[3*((ki+1)*Width4+ji+1)] +
                      (int)Background[3*(ki*Width4+ji)+1] - (int)
Background[3*((ki+1)*Width4+ji+1)+1] +
                      (int)Background[3*(ki*Width4+ji)+2] - (int)
Background[3*((ki+1)*Width4+ji+1)+2])/(255.f*3.f);
        gradXFg[kk] = (float)((int)Frame4[3*(ki*Width4+ji)] - (int)
Frame4[3*(ki*Width4+ji+1)] +
                      (int)Frame4[3*(ki*Width4+ji)+1] - (int)
Frame4[3*(ki*Width4+ji+1)+1] +
                      (int)Frame4[3*(ki*Width4+ji)+2] - (int)
Frame4[3*(ki*Width4+ji+1)+2])/(255.f*3.f);
        gradYFg[kk] = (float)((int)Frame4[3*(ki*Width4+ji)] - (int)
Frame4[3*((ki+1)*Width4+ji+1)] +
                      (int)Frame4[3*(ki*Width4+ji)+1] - (int)
Frame4[3*((ki+1)*Width4+ji+1)+1] +
                      (int)Frame4[3*(ki*Width4+ji)+2] - (int)
Frame4[3*((ki+1)*Width4+ji+1)+2])/(255.f*3.f);
        gradXDif[kk] = abs(gradXBg[kk]-gradXFg[kk]);
        gradYDif[kk] = abs(gradYBg[kk]-gradYFg[kk]);
        kk++;
      }
    float difGradients = SumInterval(gradXDif, 0, kk, 0) +
SumInterval(gradYDif, 0, kk, 0)/(float)kk;
    delete [] gradXBg;
    delete [] gradYBg;
    delete [] gradXFg;
    delete [] gradYFg;
    delete [] gradXDif;
    delete [] gradYDif;
    if (difGradients > 3)
      AssignBlob( 4*max_col, Height-4*max_row);
    peaks_sum[maxSum.index] = 0;
    for (int k=0; k<peakcount_row; k++)
```

Movement Analyzer Script by Zdenka Šturcová

```
      for (int j=0; j<peakcount_col; j++)
        if (Dist2D(peaks_row[k], peaks_col[j], max_row, max_col)<30)
          peaks_sum[k*peakcount_col+j] = 0;
    MaxFromInterval(peaks_sum, 0, peakcount_row*peakcount_col,
maxSum);
  }
  for (int k = 0; k < PeopleCount; k++)
  {
    if (Person[k].lastFrame < FrameNum)
    {
      Person[k].time++;
      Person[k].lastFrame = FrameNum;
      Person[k].trajectoryX[Person[k].time] = Person[k].
trajectoryX[Person[k].time-1];
      Person[k].trajectoryY[Person[k].time] = Person[k].
trajectoryY[Person[k].time-1];
      if (pFile!=NULL){
          fprintf (pFile,"{%d %d %d}\n",k, Person[k].
trajectoryX[Person[k].time], Person[k].trajectoryY[Person[k].time],
0);
      }
    }
    CheckWalkLength();
    }

  delete [] rows;
  delete [] cols;
  delete [] rowsmed;
  delete [] colsmed;
  delete [] peaks_row;
  delete [] peaks_col;
  delete [] peaks_sum;
}
void
CVideoFilter::AssignBlob(int x, int y)
{ bool BlobFound = false;
  int minVzd = max(Width, Height);
  int minPers = 0;
  for (int k = 0; k < PeopleCount; k++)
    if ((Dist2D(x, y, Person[k].x, Person[k].y) < min(minVzd,
(Height+Width)/4)))
    {
      minVzd = (int)(Dist2D(x, y, Person[k].x, Person[k].y));
      minPers = k;
      BlobFound = true;
    }
    if (BlobFound)
    {
      Person[minPers].x = x;
      Person[minPers].y = y;
      Person[minPers].time++;
      Person[minPers].lastFrame = FrameNum;
      Person[minPers].active = true;

      Person[minPers].trajectoryX[Person[minPers].time] = (float)x;
      Person[minPers].trajectoryY[Person[minPers].time] = (float)y;
      if (Person[minPers].time > 3)
      {
        Person[minPers].trajectoryX[Person[minPers].time-2] =
(Person[minPers].trajectoryX[Person[minPers].time-4]+
                                                Person[minPers].
trajectoryX[Person[minPers].time-3]+
                                                Person[minPers].
trajectoryX[Person[minPers].time-1]+x)/4;
        Person[minPers].trajectoryY[Person[minPers].time-2] =
(Person[minPers].trajectoryY[Person[minPers].time-4]+
                                                Person[minPers].
trajectoryY[Person[minPers].time-3]+
                                                Person[minPers].
trajectoryY[Person[minPers].time-1]+y)/4;
      }
      BlobFound = true;
      if (pFile!=NULL){
          fprintf (pFile,"%d %d %d %d\n",minPers, FrameNum, x, y);
      }
      CheckTrajectoryPattern(minPers);
    }
  for (int k = 0; k < 4; k++)
    if ((Dist2D(x, y, Door[k].x, Door[k].y) < Door[k].
radius)&&(!BlobFound))
    {
      Person[PeopleCount].x = x;
      Person[PeopleCount].y = y;
      Person[PeopleCount].time = 0;
      Person[PeopleCount].maxtraj = 500;
      Person[PeopleCount].trajectoryX = new
float[Person[PeopleCount].maxtraj];
      Person[PeopleCount].trajectoryY = new
float[Person[PeopleCount].maxtraj];
      Person[PeopleCount].trajectoryX[0] = (float)x;
      Person[PeopleCount].trajectoryY[0] = (float)y;
      Person[PeopleCount].lastFrame = FrameNum;
      Person[PeopleCount].active = true;
      PeopleCount++;
      BlobFound = true;
    }
}
void
CVideoFilter::CheckTrajectoryPattern(int p)
{ if (Person[p].time > 5*15)
  {
    bool idle = CheckIdle(p);
    if (idle)
    {}
    else
    {
      if ((Person[p].time - Person[p].lastPatternEnd) > 3*5*15)
        FindNewTrajectoryPattern(p);
    }
  }
}
bool
CVideoFilter::CheckIdle(int p)
{
  if (Person[p].time%15 == 0)
  {
    bool flatX = CheckFlat(Person[p].trajectoryX, Person[p].time -
10*15, Person[p].time);
    bool flatY = CheckFlat(Person[p].trajectoryY, Person[p].time -
10*15, Person[p].time);
    if ((flatX)&&(flatY))
      if (Person[p].idle == 0)
      {
        Person[p].idle = 10;
        return true;
      }
      else
      {
        Person[p].idle++;
        return true;
      }
    else
    {
      Person[p].idle = 0;
      return false;
    }
  }
  return false;
}
void
CVideoFilter::CheckWalkLength()
{
  for (int j=0; j<PeopleCount; j++)
    if (Person[j].time+10 > Person[j].maxtraj)
    {
      float * tmpWalkX = new float[Person[j].maxtraj];
      float * tmpWalkY = new float[Person[j].maxtraj];
      for (int k=0; k<=Person[j].time; k++)
      {
        tmpWalkX[k] = Person[j].trajectoryX[k];
        tmpWalkY[k] = Person[j].trajectoryY[k];
      }
      delete [] Person[j].trajectoryX;
      delete [] Person[j].trajectoryY;
      Person[j].maxtraj += 500;
      Person[j].trajectoryX = new float[Person[j].maxtraj];
      Person[j].trajectoryY = new float[Person[j].maxtraj];
      for (int k=0; k<=Person[j].time; k++)
      {
        Person[j].trajectoryX[k] = tmpWalkX[k];
        Person[j].trajectoryY[k] = tmpWalkY[k];
      }
      delete [] tmpWalkX;
      delete [] tmpWalkY;
    }
}
int
CVideoFilter::RGBtoH(int r, int g, int b)
{
  int rgb_min = (int)min(min(r, g), b);
  int rgb_max = (int)max(max(r, g), b);
  int val = rgb_max;
  int sat = 0;
  int hue = 0;
  if (val == 0)
  {
    hue = 0;
    sat = 0;
    return hue;
  }
  sat = 255*(rgb_max - rgb_min)/val;
  if (sat == 0) {
    hue = 0;
    return hue;
  }

  /* Compute hue */
  if (rgb_max == r)
    hue = 0 + 43*(g - b)/(rgb_max - rgb_min);
  else
    if (rgb_max == g)
      hue = 85 + 43*(b - r)/(rgb_max - rgb_min);
    else /* rgb_max == rgb.b */
      hue = 171 + 43*(r - g)/(rgb_max - rgb_min);
  return hue;
}
void
CVideoFilter::DrawWalk()
{ for (int j=0; j<PeopleCount; j++)
    if (Person[j].time>1)
    {
      double * drawW1 = new double[4*Person[j].time];
      float * drawVer = new float[Person[j].time];
      float * drawHor = new float[Person[j].time];
      for(int k = 0; k < Person[j].time; k++)
      {
        drawW1[2*k] = Person[j].trajectoryX[k];
        drawW1[2*k+1] = Person[j].trajectoryY[k];
        drawW1[4*Person[j].time - 2*k-2] = Person[j].trajectoryX[k];
        drawW1[4*Person[j].time - 2*k-1] = Person[j].trajectoryY[k];
        drawHor[k] = (float)Person[j].trajectoryX[k] / (float)Width;
        drawVer[k] = (float)(Height-Person[j].trajectoryY[k]) /
(float)Height;
      }
      Graphics->Polygon(2*Person[j].time, drawW1,
Colors[j], 0.6, CVectorGraphics::lsDefault,
CVectorGraphics::CColor(1.0,0.0,0.0,0.0));
      Graphics->Rectangle(Person[j].x-30, Person[j].y-30, Person[j].
x+30, Person[j].y+30,
                          Colors[j], 2.0, CVectorGraphics::lsDefault,
CVectorGraphics::CColor(0,0,0,0));
      delete [] drawW1;
      delete [] drawVer;
      delete [] drawHor;
    }
}
void
CVideoFilter::DrawBasicLayout(int frame)
{
  char textframe[128]; sprintf(textframe, „Frame#%d %d“, frame,
PeopleCount);
  Graphics->Text(textframe, 10, 10, CVectorGraphics::CColor(„red“),
12);
}
bool
CVideoFilter::Transform(LPBYTE bufferIn, int sizeIn, LPBYTE bufferOut,
int
    sizeOut, int frame)
{
  FrameNum = frame;
  // Compute chroma
  for (int k=0; k<(Height4*Width4); k++)
  {
    float ssum = (float)Frame4[3*k]+(float)Frame4[3*k+1]+(float)
Frame4[3*k+2];
    ChromaGR[2*k] = (float)Frame4[3*k+1]/(ssum);
    ChromaGR[2*k+1] = (float)Frame4[3*k+2]/(ssum);
    ssum = (float)Background[3*k]+(float)Background[3*k+1]+(float)
Background[3*k+2];
    ChromaGRprev[2*k] = (float)Background[3*k+1]/(ssum);
    ChromaGRprev[2*k+1] = (float)Background[3*k+2]/(ssum);
    if (frame==0)
    {
      DifferenceChromaGR[2*k] = 0; DifferenceChromaGR[2*k+1] = 0;
      ChromaMi[2*k] = ChromaGR[2*k]; ChromaMi[2*k+1] =
ChromaGR[2*k+1];
      ChromaVar[2*k] = 0.01f; ChromaVar[2*k+1] = 0.01f;
      Mhi[k] = 0;
    }
    else
    {
      float alpha = 0.999f;
      DifferenceChromaGR[2*k]   = abs(ChromaGRprev[2*k]-
ChromaGR[2*k]);
      DifferenceChromaGR[2*k+1] = abs(ChromaGRprev[2*k+1]-
ChromaGR[2*k+1]);
      // najdeme tie kde sa nieco zmenilo
      bool matchG = ((ChromaGR[2*k] - ChromaMi[2*k])*(ChromaGR[2*k]
- ChromaMi[2*k]) > ChromaVar[2*k]) ? true : false;
      bool matchR = ((ChromaGR[2*k+1] -
ChromaMi[2*k+1])*(ChromaGR[2*k+1] - ChromaMi[2*k+1]) >
ChromaVar[2*k+1])? true : false;
      ChromaMatch[k] = matchG||matchR;
      float oldMiG = ChromaMi[2*k];
      float oldMiR = ChromaMi[2*k+1];
      if ((matchG) && (matchR))
      {
        ChromaMi[2*k] = alpha*ChromaMi[2*k] +
(1-alpha)*(ChromaGR[2*k]);
        ChromaMi[2*k+1] = alpha*ChromaMi[2*k+1] +
(1-alpha)*(ChromaGR[2*k+1]);
      }
      ChromaVar[2*k] = alpha*(ChromaVar[2*k] + (ChromaMi[2*k]-
oldMiG)*(ChromaMi[2*k]-oldMiG)) + (1-alpha)*(ChromaGR[2*k]-
ChromaMi[2*k])*(ChromaGR[2*k]-ChromaMi[2*k]);
      ChromaVar[2*k+1] = alpha*(ChromaVar[2*k+1] + (ChromaMi[2*k+1]-
oldMiR)*(ChromaMi[2*k+1]-oldMiR)) + (1-alpha)*(ChromaGR[2*k+1]-
ChromaMi[2*k+1])*(ChromaGR[2*k+1]-ChromaMi[2*k+1]);
    }
  }
  for (int k=1; k<Height4-1; k++)
    for (int j=1; j<Width4-1; j++)
      if (ChromaMatch[k*Width4+j])
        if(ChromaMatch[(k-1)*Width4+j-1]||ChromaMatch[(k-
1)*Width4+j]||ChromaMatch[(k-1)*Width4+j+1]||
           ChromaMatch[k*Width4+j-1]||ChromaMatch[k*Width4+j+1]||
           ChromaMatch[(k+1)*Width4+j-1]||ChromaMatch[(k+1)*Width4+j
]||ChromaMatch[(k+1)*Width4+j+1])
        {}
        else
          ChromaMatch[k*Width4+j] = false;

  for (int k=0; k<(3*Height4*Width4); k++)
  {
    if (frame==0) DifferenceBGR[k] = 0;
    else
      DifferenceBGR[k] = abs(Background[k]-Frame4[k]);
  }
  for (int k=0; k<(Height4*Width4); k++)
  {
      DifferenceBGR[3*k] = (unsigned char)abs((double)
Background[3*k]*0.11+(double)Background[3*k+1]*0.56+(double)
Background[3*k+2]*0.3-(double)Frame4[3*k]*0.11-(double)
Frame4[3*k+1]*0.56-(double)Frame4[3*k+2]*0.3);
  }
  int h_new, h_bg;
  for (int k=0; k<(Height4*Width4); k++)
  {
    h_new = RGBtoH(Frame4[3*k], Frame4[3*k+1], Frame4[3*k+2]);
    h_bg = RGBtoH(Background[3*k], Background[3*k+1],
Background[3*k+2]);
     DifferenceHue[k] = (int)min(abs(h_new-h_bg), abs(h_new+h_bg-
255));
    Hue[k] = h_new;
  }
  for (int k=0; k<(Height4*Width4); k++)
  {
    Motion[k] = 0;
    Motion[k] += DifferenceBGR[3*k]/3;
    Motion[k] += DifferenceBGR[3*k+1]/3;
    Motion[k] += DifferenceBGR[3*k+2]/3;
    if (Motion[k] > 40)
    {if (Mhi[k]<250) Mhi[k]+=2;}
    else
    {if ((Mhi[k] > 0)&&(frame % 2 == 0)) Mhi[k]--;}
  }
  for (int k=0; k<Height4; k++)
    for (int j=0; j<Width4; j++)
    {
      if (frame == 0)
      {
        Background[3*(k*Width4+j)] = Frame4[3*(k*Width4+j)];
        Background[3*(k*Width4+j)+1] = Frame4[3*(k*Width4+j)+1];
        Background[3*(k*Width4+j)+2] = Frame4[3*(k*Width4+j)+2];
      }
      else
      {
        bool prazdno = true;
        for (int e=0; e<PeopleCount; e++)
          if (Dist2D(4*j, 4*k, Person[e].x, Height-Person[e].y) < 200)
            prazdno = false;
        if (prazdno)
        {
          Background[3*(k*Width4+j)] = (unsigned char)(0.95f*(float)
Background[3*(k*Width4+j)] + 0.05f*(float)Frame4[3*(k*Width4+j)]);
          Background[3*(k*Width4+j)+1] = (unsigned char)
(0.95f*(float)Background[3*(k*Width4+j)+1] + 0.05f*(float)
Frame4[3*(k*Width4+j)+1]);
          Background[3*(k*Width4+j)+2] = (unsigned char)
(0.95f*(float)Background[3*(k*Width4+j)+2] + 0.05f*(float)
Frame4[3*(k*Width4+j)+2]);
        }
        else
        {
          Background[3*(k*Width4+j)] = (unsigned char)(0.99999f*(float)
Background[3*(k*Width4+j)] + 0.00001f*(float)Frame4[3*(k*Width4+j)]);
          Background[3*(k*Width4+j)+1] = (unsigned char)
(0.99999f*(float)Background[3*(k*Width4+j)+1] + 0.00001f*(float)
Frame4[3*(k*Width4+j)+1]);
          Background[3*(k*Width4+j)+2] = (unsigned char)
(0.99999f*(float)Background[3*(k*Width4+j)+2] + 0.00001f*(float)
Frame4[3*(k*Width4+j)+2]);
        }
      }
    }
  Graphics->SetData(bufferOut);
  if (frame > 1)
    FindPeople();
  DrawBasicLayout(frame);

  for (int k=0; k<PeopleCount; k++)
  {
    char text1[128]; sprintf(text1, „%d“, (int)Person[k].time);
    Graphics->Text(text1, 100+k*40, 10,
CVectorGraphics::CColor(„green“), 12);
  }
  DrawWalk();

  Graphics->GetData(bufferOut);
    for (int k=0; k<(2*Height4*Width4); k++)
    {
      ChromaGRprev[k] = ChromaGR[k];
    }
  for (int k=0; k<Height4*Width4*3; k++)
    LastFrame[k] = Frame4[k];
  return true;
}
```



User is browsing a virtual gallery via web browser.
Goal: Capture user movement during his virtual visit of the gallery.
Prerequisites: User's web browser must be capable to run Javascript and store cookies (http://en.wikipedia.org/wiki/HTTP_cookie).

Realization

Web Browser

1. Web browsers Mozilla Firefox and Google Chrome are capable to run custom Javascript code in page.
2. Custom written Javascript code inspects content of currently viewed web page using Prototype Javascript library (http://www.prototypejs.org/) via DOM tree parsing (http://en.wikipedia.org/wiki/DOM_tree) and CSS navigation (http://en.wikipedia.org/wiki/Cascading_Style_Sheets).
3. When an user is about to visit gallery web portal for the first time, unique session ID is assigned to him and stored in a cookie:
```
sid = Math.round(new Date().getTime()/1000);
document.cookie="session_id="+sid;
```
4. If the currently viewed web page contains presentation of an artwork, following data is collected:
   - session ID
   - ID of artwork´s author
   - number of clicks the user already made during his visit of the gallery (stored in a HTTP cookie)
   - category of the artwork (one of 'Architecture and Design', 'Drawings', 'Film', 'Performance Art', 'Painting and Sculpture', 'Photography', 'Prints and Illustrated Books', 'Multiple', 'All').
5. Collected data is submitted via an AJAX request (http://en.wikipedia.org/wiki/Ajax_(programming)) to The Server:
```
new Ajax.Request('http://theserver:4567/add?
artist_id='+artist_id+'&dept_id='+dept_id+'&clicks='+clicks+'&session_id='+session_id());
```

The Server

1. Web server is written in Ruby programming language (http://en.wikipedia.org/wiki/Ruby_(programming_language)) using Sinatra web server engine (http://www.sinatrarb.com/).
2. Web server is listening to incoming request from the web:
```
options '/add' do
   browser = connected_browsers.index(params[:session_id])
   // process the rest of the data from request
end
```
3. On incoming AJAX request from our Javascript code, data is extracted from the request, data is transformed and written to output file for the Grasshopper visualization tool.

Data Analyzer Script by Peter Vojtek

```
// ==UserScript==
// @name           Saddie's Thesis Script
// @namespace      http://petervojtek.eu/saddie.user.js
// @description    Moma Gallery Visualizer
// @include        *
// @match http://*.moma.org/*
// ==/UserScript==
/*  Prototype JavaScript framework, version 1.7
 *  (c) 2005-2010 Sam Stephenson
 *
 *  Prototype is freely distributable under the terms of an MIT-style license.
 *  For details, see the Prototype web site: http://www.prototypejs.org/
 *
 *--------------------------------------------------------------------------*/
var Prototype = {
  Version: '1.7',
  Browser: (function(){
    var ua = navigator.userAgent;
    var isOpera = Object.prototype.toString.call(window.opera) == '[object Opera]';
    return {
      IE:             !!window.attachEvent && !isOpera,
      Opera:          isOpera,
      WebKit:         ua.indexOf('AppleWebKit/') > -1,
      Gecko:          ua.indexOf('Gecko') > -1 && ua.indexOf('KHTML') === -1,
      MobileSafari:   /Apple.*Mobile/.test(ua)
    }
  })(),
  BrowserFeatures: {
    XPath: !!document.evaluate,
    SelectorsAPI: !!document.querySelector,
    ElementExtensions: (function() {
      var constructor = window.Element || window.HTMLElement;
      return !!(constructor && constructor.prototype);
    })(),
    SpecificElementExtensions: (function() {
      if (typeof window.HTMLDivElement !== 'undefined')
        return true;
      var div = document.createElement('div'),
          form = document.createElement('form'),
          isSupported = false;
      if (div['__proto__'] && (div['__proto__'] !== form['__proto__'])) {
        isSupported = true;
      }
      div = form = null;
      return isSupported;
    })()
  },
  ScriptFragment: '<script[^>]*>([\\S\\s]*?)<\/script>',
  JSONFilter: /^\/\*-secure-([\s\S]*)\*\/\s*$/,
  emptyFunction: function() { },
  K: function(x) { return x }
};
if (Prototype.Browser.MobileSafari)
  Prototype.BrowserFeatures.SpecificElementExtensions = false;
var Abstract = { };
var Try = {
  these: function() {
    var returnValue;
    for (var i = 0, length = arguments.length; i < length; i++) {
      var lambda = arguments[i];
      try {
        returnValue = lambda();
        break;
      } catch (e) { }
    }
    return returnValue;
  }
};
/* Based on Alex Arnell's inheritance implementation. */
var Class = (function() {
  var IS_DONTENUM_BUGGY = (function(){
    for (var p in { toString: 1 }) {
      if (p === 'toString') return false;
    }
    return true;
  })();
  function subclass() {};
  function create() {
    var parent = null, properties = $A(arguments);
    if (Object.isFunction(properties[0]))
      parent = properties.shift();
    function klass() {
      this.initialize.apply(this, arguments);
    }
    Object.extend(klass, Class.Methods);
    klass.superclass = parent;
    klass.subclasses = [];
    if (parent) {
      subclass.prototype = parent.prototype;
      klass.prototype = new subclass;
      parent.subclasses.push(klass);
    }
    for (var i = 0, length = properties.length; i < length; i++)
      klass.addMethods(properties[i]);
    if (!klass.prototype.initialize)
      klass.prototype.initialize = Prototype.emptyFunction;
    klass.prototype.constructor = klass;
    return klass;
  }
  function addMethods(source) {
    var ancestor   = this.superclass && this.superclass.prototype,
        properties = Object.keys(source);
    if (IS_DONTENUM_BUGGY) {
      if (source.toString != Object.prototype.toString)
        properties.push("toString");
      if (source.valueOf != Object.prototype.valueOf)
        properties.push("valueOf");
    }
    for (var i = 0, length = properties.length; i < length; i++) {
      var property = properties[i], value = source[property];
      if (ancestor && Object.isFunction(value) &&
          value.argumentNames()[0] == "$super") {
        var method = value;
        value = (function(m) {
          return function() { return ancestor[m].apply(this, arguments); };
        })(property).wrap(method);
        value.valueOf = method.valueOf.bind(method);
        value.toString = method.toString.bind(method);
      }
      this.prototype[property] = value;
    }
    return this;
  }
  return {
    create: create,
    Methods: {
      addMethods: addMethods
    }
  };
})();
(function() {
  var _toString = Object.prototype.toString,
      NULL_TYPE = 'Null',
      UNDEFINED_TYPE = 'Undefined',
      BOOLEAN_TYPE = 'Boolean',
      NUMBER_TYPE = 'Number',
      STRING_TYPE = 'String',
      OBJECT_TYPE = 'Object',
      FUNCTION_CLASS = '[object Function]',
      BOOLEAN_CLASS = '[object Boolean]',
      NUMBER_CLASS = '[object Number]',
      STRING_CLASS = '[object String]',
      ARRAY_CLASS = '[object Array]',
      DATE_CLASS = '[object Date]',
      NATIVE_JSON_STRINGIFY_SUPPORT = window.JSON &&
        typeof JSON.stringify === 'function' &&
        JSON.stringify(0) === '0' &&
        typeof JSON.stringify(Prototype.K) === 'undefined';
  function Type(o) {
    switch(o) {
      case null: return NULL_TYPE;
      case (void 0): return UNDEFINED_TYPE;
    }
    var type = typeof o;
    switch(type) {
      case 'boolean': return BOOLEAN_TYPE;
      case 'number':  return NUMBER_TYPE;
      case 'string':  return STRING_TYPE;
    }
    return OBJECT_TYPE;
  }
  function extend(destination, source) {
    for (var property in source)
      destination[property] = source[property];
    return destination;
  }
  function inspect(object) {
    try {
      if (isUndefined(object)) return 'undefined';
      if (object === null) return 'null';
      return object.inspect ? object.inspect() : String(object);
    } catch (e) {
      if (e instanceof RangeError) return '...';
      throw e;
    }
  }
  function toJSON(value) {
    return Str('', { '': value }, []);
  }
  function Str(key, holder, stack) {
    var value = holder[key],
        type = typeof value;
    if (Type(value) === OBJECT_TYPE && typeof value.toJSON === 'function') {
      value = value.toJSON(key);
    }
    var _class = _toString.call(value);
    switch (_class) {
      case NUMBER_CLASS:
      case BOOLEAN_CLASS:
      case STRING_CLASS:
        value = value.valueOf();
    }
    switch (value) {
      case null: return 'null';
      case true: return 'true';
      case false: return 'false';
    }
    type = typeof value;
    switch (type) {
      case 'string':
        return value.inspect(true);
      case 'number':
        return isFinite(value) ? String(value) : 'null';
      case 'object':
        for (var i = 0, length = stack.length; i < length; i++) {
          if (stack[i] === value) { throw new TypeError(); }
        }
        stack.push(value);
        var partial = [];
        if (_class === ARRAY_CLASS) {
          for (var i = 0, length = value.length; i < length; i++) {
            var str = Str(i, value, stack);
            partial.push(typeof str === 'undefined' ? 'null' : str);
          }
          partial = '[' + partial.join(',') + ']';
        } else {
          var keys = Object.keys(value);
          for (var i = 0, length = keys.length; i < length; i++) {
            var key = keys[i], str = Str(key, value, stack);
            if (typeof str !== "undefined") {
              partial.push(key.inspect(true)+ ':' + str);
            }
          }
          partial = '{' + partial.join(',') + '}';
        }
        stack.pop();
        return partial;
    }
  }
  function stringify(object) {
    return JSON.stringify(object);
  }
  function toQueryString(object) {
    return $H(object).toQueryString();
  }
  function toHTML(object) {
    return object && object.toHTML ? object.toHTML() : String.interpret(object);
  }
  function keys(object) {
    if (Type(object) !== OBJECT_TYPE) { throw new TypeError(); }
    var results = [];
    for (var property in object) {
      if (object.hasOwnProperty(property)) {
        results.push(property);
      }
    }
    return results;
  }
  function values(object) {
    var results = [];
    for (var property in object)
      results.push(object[property]);
    return results;
  }
  function clone(object) {
    return extend({ }, object);
  }
  function isElement(object) {
    return !!(object && object.nodeType == 1);
  }
  function isArray(object) {
    return _toString.call(object) === ARRAY_CLASS;
  }
  var hasNativeIsArray = (typeof Array.isArray == 'function')
    && Array.isArray([]) && !Array.isArray({});
  if (hasNativeIsArray) {
    isArray = Array.isArray;
  }
  function isHash(object) {
    return object instanceof Hash;
  }
  function isFunction(object) {
    return _toString.call(object) === FUNCTION_CLASS;
  }
  function isString(object) {
    return _toString.call(object) === STRING_CLASS;
  }
  function isNumber(object) {
    return _toString.call(object) === NUMBER_CLASS;
  }
  function isDate(object) {
    return _toString.call(object) === DATE_CLASS;
  }
  function isUndefined(object) {
    return typeof object === "undefined";
  }
  extend(Object, {
    extend:        extend,
    inspect:       inspect,
    toJSON:        NATIVE_JSON_STRINGIFY_SUPPORT ? stringify : toJSON,
    toQueryString: toQueryString,
    toHTML:        toHTML,
    keys:          Object.keys || keys,
    values:        values,
    clone:         clone,
    isElement:     isElement,
    isArray:       isArray,
    isHash:        isHash,
    isFunction:    isFunction,
    isString:      isString,
    isNumber:      isNumber,
    isDate:        isDate,
    isUndefined:   isUndefined
  });
})();
Object.extend(Function.prototype, (function() {
  var slice = Array.prototype.slice;
  function update(array, args) {
    var arrayLength = array.length, length = args.length;
    while (length--) array[arrayLength + length] = args[length];
    return array;
  }
  function merge(array, args) {
    array = slice.call(array, 0);
    return update(array, args);
  }
  function argumentNames() {
    var names = this.toString().match(/^[\s\(]*function[^(]*\(([^)]*)\)/)[1]
      .replace(/\/\/.*?[\r\n]|\/\*(?:.|[\r\n])*?\*\//g, '')
      .replace(/\s+/g, '').split(',');
    return names.length == 1 && !names[0] ? [] : names;
  }
  function bind(context) {
    if (arguments.length < 2 && Object.isUndefined(arguments[0])) return this;
    var __method = this, args = slice.call(arguments, 1);
    return function() {
      var a = merge(args, arguments);
      return __method.apply(context, a);
    }
  }
  function bindAsEventListener(context) {
    var __method = this, args = slice.call(arguments, 1);
    return function(event) {
      var a = update([event || window.event], args);
      return __method.apply(context, a);
    }
  }
  function curry() {
    if (!arguments.length) return this;
    var __method = this, args = slice.call(arguments, 0);
    return function() {
      var a = merge(args, arguments);
      return __method.apply(this, a);
    }
  }
  function delay(timeout) {
    var __method = this, args = slice.call(arguments, 1);
    timeout = timeout * 1000;
    return window.setTimeout(function() {
      return __method.apply(__method, args);
    }, timeout);
  }
  function defer() {
    var args = update([0.01], arguments);
    return this.delay.apply(this, args);
  }
  function wrap(wrapper) {
    var __method = this;
    return function() {
      var a = update([__method.bind(this)], arguments);
      return wrapper.apply(this, a);
    }
  }
  function methodize() {
    if (this._methodized) return this._methodized;
    var __method = this;
    return this._methodized = function() {
      var a = update([this], arguments);
      return __method.apply(null, a);
    };
  }
  return {
    argumentNames:       argumentNames,
    bind:                bind,
    bindAsEventListener: bindAsEventListener,
    curry:               curry,
    delay:               delay,
    defer:               defer,
    wrap:                wrap,
    methodize:           methodize
  }
})());
(function(proto) {
  function toISOString() {
    return this.getUTCFullYear() + '-' +
      (this.getUTCMonth() + 1).toPaddedString(2) + '-' +
      this.getUTCDate().toPaddedString(2) + 'T' +
      this.getUTCHours().toPaddedString(2) + ':' +
      this.getUTCMinutes().toPaddedString(2) + ':' +
      this.getUTCSeconds().toPaddedString(2) + 'Z';
  }
  function toJSON() {
    return this.toISOString();
  }
  if (!proto.toISOString) proto.toISOString = toISOString;
  if (!proto.toJSON) proto.toJSON = toJSON;
})(Date.prototype);
RegExp.prototype.match = RegExp.prototype.test;
RegExp.escape = function(str) {
  return String(str).replace(/([.*+?^=!:${}()|[\]\/\\])/g, '\\$1');
};
var PeriodicalExecuter = Class.create({
  initialize: function(callback, frequency) {
    this.callback = callback;
    this.frequency = frequency;
    this.currentlyExecuting = false;
    this.registerCallback();
  },
  registerCallback: function() {
    this.timer = setInterval(this.onTimerEvent.bind(this), this.frequency * 1000);
  },
  execute: function() {
    this.callback(this);
  },
  stop: function() {
    if (!this.timer) return;
    clearInterval(this.timer);
    this.timer = null;
  },
  onTimerEvent: function() {
    if (!this.currentlyExecuting) {
      try {
        this.currentlyExecuting = true;
        this.execute();
        this.currentlyExecuting = false;
      } catch(e) {
        this.currentlyExecuting = false;
        throw e;
      }
    }
  }
});
Object.extend(String, {
  interpret: function(value) {
    return value == null ? '' : String(value);
  },
  specialChar: {
    '\b': '\\b',
    '\t': '\\t',
    '\n': '\\n',
    '\f': '\\f',
    '\r': '\\r',
    '\\': '\\\\'
  }
});
Object.extend(String.prototype, (function() {
  var NATIVE_JSON_PARSE_SUPPORT = window.JSON &&
    typeof JSON.parse === 'function' &&
    JSON.parse('{"test": true}').test;
  function prepareReplacement(replacement) {
    if (Object.isFunction(replacement))
      return replacement;
    var template = new Template(replacement);
    return function(match) { return template.evaluate(match) };
  }
  function gsub(pattern, replacement) {
    var result = '', source = this, match;
    replacement = prepareReplacement(replacement);
    if (Object.isString(pattern))
      pattern = RegExp.escape(pattern);
    if (!(pattern.length || pattern.source)) {
      replacement = replacement('');
      return replacement + source.split('').join(replacement) + replacement;
    }
    while (source.length > 0) {
      if (match = source.match(pattern)) {
        result += source.slice(0, match.index);
        result += String.interpret(replacement(match));
        source  = source.slice(match.index + match[0].length);
      } else {
        result += source, source = '';
      }
    }
    return result;
  }
  function sub(pattern, replacement, count) {
    replacement = prepareReplacement(replacement);
    count = Object.isUndefined(count) ? 1 : count;
    return this.gsub(pattern, function(match) {
      if (--count < 0) return match[0];
      return replacement(match);
    });
  }
  function scan(pattern, iterator) {
    this.gsub(pattern, iterator);
    return String(this);
  }
  function truncate(length, truncation) {
    length = length || 30;
    truncation = Object.isUndefined(truncation) ? '...' : truncation;
    return this.length > length ?
```

```
      this.slice(0, length - truncation.
length) + truncation : String(this);
  }
  function strip() {
    return this.replace(/^\s+/, ,‘).
replace(/\s+$/, ,‘);
  }
  function stripTags() {
    return this.replace(/<\w+(\s+(„[^“]*“|‘[
^’]*’|[^>])+)?>|<\/\w+>/gi, ,‘);
  }
  function stripScripts() {
    return this.replace(new
RegExp(Prototype.ScriptFragment, ,img‘),
,‘);
  }
  function extractScripts() {
    var matchAll = new RegExp(Prototype.
ScriptFragment, ,img‘),
        matchOne = new RegExp(Prototype.
ScriptFragment, ,im‘);
    return (this.match(matchAll) || []).
map(function(scriptTag) {
      return (scriptTag.match(matchOne) ||
[,‘, ,‘])[1];
    });
  }
  function evalScripts() {
    return this.extractScripts().
map(function(script) { return eval(script)
});
  }
  function escapeHTML() {
    return this.replace(/&/g,‘&amp;‘).
replace(/<g,‘&lt;‘).replace(/>/g,‘&gt;‘);
  }
  function unescapeHTML() {
    return this.stripTags().
replace(/&lt;/g,‘<‘).replace(/&gt;/g,‘>‘).
replace(/&amp;/g,‘&‘);
  }
  function toQueryParams(separator) {
    var match = this.strip().match(/([^?#]*)
(#.*)?$/);
    if (!match) return { };
    return match[1].split(separator || ,&‘).
inject({ }, function(hash, pair) {
      if ((pair = pair.split(,=‘))[0]) {
        var key = decodeURIComponent(pair.
shift()),
            value = pair.length > 1 ? pair.
join(,=‘) : pair[0];
        if (value != undefined) value =
decodeURIComponent(value);
        if (key in hash) {
          if (!Object.isArray(hash[key]))
hash[key] = [hash[key]];
          hash[key].push(value);
        }
        else hash[key] = value;
      }
      return hash;
    });
  }
  function toArray() {
    return this.split(,‘);
  }
  function succ() {
    return this.slice(0, this.length - 1) +
      String.fromCharCode(this.
charCodeAt(this.length - 1) + 1);
  }
  function times(count) {
    return count < 1 ? ,‘ : new Array(count
+ 1).join(this);
  }
  function camelize() {
    return this.replace(/-+(.)?/g,
function(match, chr) {
      return chr ? chr.toUpperCase() : ,‘;
    });
  }
  function capitalize() {
    return this.charAt(0).toUpperCase() +
this.substring(1).toLowerCase();
  }
  function underscore() {
    return this.replace(/::/g, ,/‘)
               .replace(/([A-Z]+)([A-Z]
[a-z])/g, ,$1_$2‘)
               .replace(/([a-z\d])([A-Z])/g,
,$1_$2‘)
               .replace(/-/g, ,_‘)
               .toLowerCase();
  }
  function dasherize() {
    return this.replace(/_/g, ,-‘);
  }
  function inspect(useDoubleQuotes) {
    var escapedString = this.replace(/[\
x00-\x1f\\]/g, function(character) {
      if (character in String.specialChar) {
        return String.
specialChar[character];
      }
      return ,\\u00‘ + character.
charCodeAt().toPaddedString(2, 16);
    });
    if (useDoubleQuotes) return ,“‘ +
escapedString.replace(/“/g, ,\\“‘) + ,“‘;
    return „‘“ + escapedString.replace(/‘/g,
,\\\‘‘) + „‘“;
  }
  function unfilterJSON(filter) {
    return this.replace(filter || Prototype.
JSONFilter, ,$1‘);
  }
  function isJSON() {
    var str = this;
    if (str.blank()) return false;
    str = str.replace(/\\(?:[„\\\/
bfnrt]|u[0-9a-fA-F]{4})/g, ,@‘);
    str = str.replace(/“[^“\\\n\
r]*“|true|false|null|-?\d+(?:\.\d*)?(?:[eE]
[+\-]?\d+)?/g, ,]‘);
    str = str.replace(/(?:^|:|,)(?:\s*\
[)+/g, ,‘);
    return (/^[\],:{}\s]*$/).test(str);
  }
  function evalJSON(sanitize) {
    var json = this.unfilterJSON(),
        cx = /[\u0000\u00ad\u0600-\u0604\
u070f\u17b4\u17b5\u200c-\u200f\u2028-\u202f\
u2060-\u206f\ufeff\uff0-\uffff]/g;
    if (cx.test(json)) {
      json = json.replace(cx, function (a) {
        return ,\\u‘ + (,0000‘ +
a.charCodeAt(0).toString(16)).slice(-4);
      });
    }
    try {
      if (!sanitize || json.isJSON()) return
eval(,(, + json + ,)‘);
    } catch (e) { }
    throw new SyntaxError(,Badly formed JSON
string: , + this.inspect());
  }
  function parseJSON() {
    var json = this.unfilterJSON();
    return JSON.parse(json);
  }
  function include(pattern) {
    return this.indexOf(pattern) > -1;
  }
  function startsWith(pattern) {
    return this.lastIndexOf(pattern, 0) ===
0;
  }
  function endsWith(pattern) {
    var d = this.length - pattern.length;
    return d >= 0 && this.indexOf(pattern,
d) === d;
  }
  function empty() {
    return this == ,‘;
  }
  function blank() {
    return /^\s*$/.test(this);
  }
  function interpolate(object, pattern) {
    return new Template(this, pattern).
evaluate(object);
  }
  return {
    gsub:           gsub,
    sub:            sub,
    scan:           scan,
    truncate:       truncate,
    strip:          String.prototype.trim
|| strip,
    stripTags:      stripTags,
    stripScripts:   stripScripts,
    extractScripts: extractScripts,
    evalScripts:    evalScripts,
    escapeHTML:     escapeHTML,
    unescapeHTML:   unescapeHTML,
    toQueryParams:  toQueryParams,
    parseQuery:     toQueryParams,
    toArray:        toArray,
    succ:           succ,
    times:          times,
    camelize:       camelize,
    capitalize:     capitalize,
    underscore:     underscore,
    dasherize:      dasherize,
    inspect:        inspect,
    unfilterJSON:   unfilterJSON,
    isJSON:         isJSON,
    evalJSON:       NATIVE_JSON_PARSE_
SUPPORT ? parseJSON : evalJSON,
    include:        include,
    startsWith:     startsWith,
    endsWith:       endsWith,
    empty:          empty,
    blank:          blank,
    interpolate:    interpolate
  };
})());
var Template = Class.create({
  initialize: function(template, pattern) {
    this.template = template.toString();
    this.pattern = pattern || Template.
Pattern;
  },
  evaluate: function(object) {
    if (object && Object.isFunction(object.
toTemplateReplacements))
      object = object.
toTemplateReplacements();
    return this.template.gsub(this.pattern,
function(match) {
      if (object == null) return (match[1]
+ ,‘);
      var before = match[1] || ,‘;
      if (before == ,\\‘) return match[2];
      var ctx = object, expr = match[3],
          pattern = /^([^.[]+|\
[((?:.*?[^\\])?)\])(\.|\[|$)/;
      match = pattern.exec(expr);
      if (match == null) return before;
      while (match != null) {
        var comp = match[1].startsWith(,[,)
? match[2].replace(/\\\\]/g, ,]‘) :
match[1];
        ctx = ctx[comp];
        if (null == ctx || ,‘ == match[3])
break;
        expr = expr.substring(,[, ==
match[3] ? match[1].length : match[0].
length);
        match = pattern.exec(expr);
      }
      return before + String.interpret(ctx);
    });
  }
});
Template.Pattern = /(^|.|\r|\n)(#\
{(.*?)\})/;
var $break = { };
var Enumerable = (function() {
  function each(iterator, context) {
    var index = 0;
    try {
      this._each(function(value) {
        iterator.call(context, value,
index++);
      });
    } catch (e) {
      if (e != $break) throw e;
    }
    return this;
  }
  function eachSlice(number, iterator,
context) {
    var index = -number, slices = [], array
= this.toArray();
    if (number < 1) return array;
    while ((index += number) < array.length)
      slices.push(array.slice(index,
index+number));
    return slices.collect(iterator,
context);
  }
  function all(iterator, context) {
    iterator = iterator || Prototype.K;
    var result = true;
    this.each(function(value, index) {
      result = result && !!iterator.
call(context, value, index);
      if (!result) throw $break;
    });
    return result;
  }
  function any(iterator, context) {
    iterator = iterator || Prototype.K;
    var result = false;
    this.each(function(value, index) {
      if (result = !!iterator.call(context,
value, index))
        throw $break;
    });
    return result;
  }
  function collect(iterator, context) {
    iterator = iterator || Prototype.K;
    var results = [];
    this.each(function(value, index) {
      results.push(iterator.call(context,
value, index));
    });
    return results;
  }
  function detect(iterator, context) {
    var result;
    this.each(function(value, index) {
      if (iterator.call(context, value,
index)) {
        result = value;
        throw $break;
      }
    });
    return result;
  }
  function findAll(iterator, context) {
    var results = [];
    this.each(function(value, index) {
      if (iterator.call(context, value,
index))
        results.push(value);
    });
    return results;
  }
  function grep(filter, iterator, context) {
    iterator = iterator || Prototype.K;
    var results = [];
    if (Object.isString(filter))
      filter = new RegExp(RegExp.
escape(filter));
    this.each(function(value, index) {
      if (filter.match(value))
        results.push(iterator.call(context,
value, index));
    });
    return results;
  }
  function include(object) {
    if (Object.isFunction(this.indexOf))
      if (this.indexOf(object) != -1) return
true;
    var found = false;
    this.each(function(value) {
      if (value == object) {
        found = true;
        throw $break;
      }
    });
    return found;
  }
  function inGroupsOf(number, fillWith) {
    fillWith = Object.isUndefined(fillWith) ?
null : fillWith;
    return this.eachSlice(number,
function(slice) {
      while(slice.length < number) slice.
push(fillWith);
      return slice;
    });
  }
  function inject(memo, iterator, context) {
    this.each(function(value, index) {
      memo = iterator.call(context, memo,
value, index);
    });
    return memo;
  }
  function invoke(method) {
    var args = $A(arguments).slice(1);
    return this.map(function(value) {
      return value[method].apply(value,
args);
    });
  }
  function max(iterator, context) {
    iterator = iterator || Prototype.K;
    var result;
    this.each(function(value, index) {
      value = iterator.call(context, value,
index);
      if (result == null || value >= result)
        result = value;
    });
    return result;
  }
  function min(iterator, context) {
    iterator = iterator || Prototype.K;
    var result;
    this.each(function(value, index) {
      value = iterator.call(context, value,
index);
      if (result == null || value < result)
        result = value;
    });
    return result;
  }
  function partition(iterator, context) {
    iterator = iterator || Prototype.K;
    var trues = [], falses = [];
    this.each(function(value, index) {
      (iterator.call(context, value, index)
?
        trues : falses).push(value);
    });
    return [trues, falses];
  }
  function pluck(property) {
    var results = [];
    this.each(function(value) {
      results.push(value[property]);
    });
    return results;
  }
  function reject(iterator, context) {
    var results = [];
    this.each(function(value, index) {
      if (!iterator.call(context, value,
index))
        results.push(value);
    });
    return results;
  }
  function sortBy(iterator, context) {
    return this.map(function(value, index) {
      return {
        value: value,
        criteria: iterator.call(context,
value, index)
      };
    }).sort(function(left, right) {
      var a = left.criteria, b = right.
criteria;
      return a < b ? -1 : a > b ? 1 : 0;
    }).pluck(,value‘);
  }
  function toArray() {
    return this.map();
  }
  function zip() {
    var iterator = Prototype.K, args =
$A(arguments);
    if (Object.isFunction(args.last()))
      iterator = args.pop();
    var collections = [this].concat(args).
map($A);
    return this.map(function(value, index) {
      return iterator(collections.
pluck(index));
    });
  }
  function size() {
    return this.toArray().length;
  }
  function inspect() {
    return ,#<Enumerable:‘ + this.toArray().
inspect() + ,>‘;
  }
  return {
    each:       each,
    eachSlice:  eachSlice,
    all:        all,
    every:      all,
    any:        any,
    some:       any,
    collect:    collect,
    map:        collect,
    detect:     detect,
    findAll:    findAll,
    select:     findAll,
    filter:     findAll,
    grep:       grep,
    include:    include,
    member:     include,
    inGroupsOf: inGroupsOf,
    inject:     inject,
    invoke:     invoke,
    max:        max,
    min:        min,
    partition:  partition,
    pluck:      pluck,
    reject:     reject,
    sortBy:     sortBy,
    toArray:    toArray,
    entries:    toArray,
    zip:        zip,
    size:       size,
    inspect:    inspect,
    find:       detect
  };
})();
function $A(iterable) {
  if (!iterable) return [];
  if (,toArray‘ in Object(iterable)) return
iterable.toArray();
  var length = iterable.length || 0, results
= new Array(length);
  while (length--) results[length] =
iterable[length];
  return results;
}
function $w(string) {
  if (!Object.isString(string)) return [];
  string = string.strip();
  return string ? string.split(/\s+/) : [];
}
Array.from = $A;
(function() {
  var arrayProto = Array.prototype,
      slice = arrayProto.slice,
      _each = arrayProto.forEach; // use
native browser JS 1.6 implementation if
available
  function each(iterator, context) {
    for (var i = 0, length = this.length >>>
0; i < length; i++) {
      if (i in this) iterator.call(context,
this[i], i, this);
    }
  }
  if (!_each) _each = each;
  function clear() {
    this.length = 0;
    return this;
  }
  function first() {
    return this[0];
  }
  function last() {
    return this[this.length - 1];
  }
  function compact() {
    return this.select(function(value) {
      return value != null;
    });
  }
  function flatten() {
    return this.inject([], function(array,
value) {
      if (Object.isArray(value))
        return array.concat(value.flatten());
      array.push(value);
      return array;
    });
  }
  function without() {
    var values = slice.call(arguments, 0);
    return this.select(function(value) {
      return !values.include(value);
    });
  }
  function reverse(inline) {
    return (inline === false ? this.
toArray() : this)._reverse();
  }
  function uniq(sorted) {
    return this.inject([], function(array,
value, index) {
      if (0 == index || (sorted ? array.
last() != value : !array.include(value)))
        array.push(value);
      return array;
    });
  }
  function intersect(array) {
    return this.uniq().findAll(function(item)
{
      return array.detect(function(value) {
return item === value });
    });
  }
  function clone() {
    return slice.call(this, 0);
  }
  function size() {
    return this.length;
  }
  function inspect() {
    return ,[, + this.map(Object.inspect).
join(,, ,) + ,]‘;
  }
  function indexOf(item, i) {
    i || (i = 0);
    var length = this.length;
    if (i < 0) i = length + i;
    for (; i < length; i++)
      if (this[i] === item) return i;
    return -1;
  }
  function lastIndexOf(item, i) {
    i = isNaN(i) ? this.length : (i < 0 ?
this.length + i : i) + 1;
    var n = this.slice(0, i).reverse().
indexOf(item);
    return (n < 0) ? n : i - n - 1;
  }
  function concat() {
    var array = slice.call(this, 0), item;
    for (var i = 0, length = arguments.
length; i < length; i++) {
```

Data Analyzer Script by Peter Vojtek

```
      item = arguments[i];
      if (Object.isArray(item) && !(,callee‘
in item)) {
        for (var j = 0, arrayLength = item.
length; j < arrayLength; j++)
          array.push(item[j]);
      } else {
        array.push(item);
      }
    }
    return array;
  }
  Object.extend(arrayProto, Enumerable);
  if (!arrayProto._reverse)
    arrayProto._reverse = arrayProto.
reverse;
  Object.extend(arrayProto, {
    _each:     _each,
    clear:     clear,
    first:     first,
    last:      last,
    compact:   compact,
    flatten:   flatten,
    without:   without,
    reverse:   reverse,
    uniq:      uniq,
    intersect: intersect,
    clone:     clone,
    toArray:   clone,
    size:      size,
    inspect:   inspect
  });
  var CONCAT_ARGUMENTS_BUGGY = (function() {
    return [].concat(arguments)[0][0] !== 1;
  })(1,2)
  if (CONCAT_ARGUMENTS_BUGGY) arrayProto.
concat = concat;
  if (!arrayProto.indexOf) arrayProto.
indexOf = indexOf;
  if (!arrayProto.lastIndexOf) arrayProto.
lastIndexOf = lastIndexOf;
})();
function $H(object) {
  return new Hash(object);
};
var Hash = Class.create(Enumerable,
(function() {
  function initialize(object) {
    this._object = Object.isHash(object) ?
object.toObject() : Object.clone(object);
  }
  function _each(iterator) {
    for (var key in this._object) {
      var value = this._object[key], pair =
[key, value];
      pair.key = key;
      pair.value = value;
      iterator(pair);
    }
  }
  function set(key, value) {
    return this._object[key] = value;
  }
  function get(key) {
    if (this._object[key] !== Object.
prototype[key])
      return this._object[key];
  }
  function unset(key) {
    var value = this._object[key];
    delete this._object[key];
    return value;
  }
  function toObject() {
    return Object.clone(this._object);
  }
  function keys() {
    return this.pluck(,key‘);
  }
  function values() {
    return this.pluck(,value‘);
  }
  function index(value) {
    var match = this.detect(function(pair) {
      return pair.value === value;
    });
    return match && match.key;
  }
  function merge(object) {
    return this.clone().update(object);
  }
  function update(object) {
    return new Hash(object).inject(this,
function(result, pair) {
      result.set(pair.key, pair.value);
      return result;
    });
  }
  function toQueryPair(key, value) {
    if (Object.isUndefined(value)) return
key;
    return key + ‚=‘ +
encodeURIComponent(String.interpret(value));
  }
  function toQueryString() {
    return this.inject([], function(results,
pair) {
      var key = encodeURIComponent(pair.
key), values = pair.value;
      if (values && typeof values ==
‚object‘) {
        if (Object.isArray(values)) {
          var queryValues = [];
          for (var i = 0, len = values.
length, value; i < len; i++) {
            value = values[i];
            queryValues.
push(toQueryPair(key, value));
          }
          return results.
concat(queryValues);
        }
      } else results.push(toQueryPair(key,
values));
      return results;
    }).join(,&‘);
  }
  function inspect() {
    return ‚#<Hash:{‚ + this.
map(function(pair) {
      return pair.map(Object.inspect).
join(‚: ‚);
    }).join(‚, ‚) + ‚}>‘;
  }
  function clone() {
    return new Hash(this);
  }
  return {
    initialize:             initialize,
    _each:                  _each,
    set:                    set,
    get:                    get,
    unset:                  unset,
    toObject:               toObject,
    toTemplateReplacements: toObject,
    keys:                   keys,
    values:                 values,
    index:                  index,
    merge:                  merge,
    update:                 update,
    toQueryString:          toQueryString,
    inspect:                inspect,
    toJSON:                 toObject,
    clone:                  clone
  };
})());
Hash.from = $H;
Object.extend(Number.prototype, (function()
{
  function toColorPart() {
    return this.toPaddedString(2, 16);
  }
  function succ() {
    return this + 1;
  }
  function times(iterator, context) {
    $R(0, this, true).each(iterator,
context);
    return this;
  }
  function toPaddedString(length, radix) {
    var string = this.toString(radix || 10);
    return ‚0‘.times(length - string.length)
+ string;
  }
  function abs() {
    return Math.abs(this);
  }
  function round() {
    return Math.round(this);
  }
  function ceil() {
    return Math.ceil(this);
  }
  function floor() {
    return Math.floor(this);
  }
  return {
    toColorPart:    toColorPart,
    succ:           succ,
    times:          times,
    toPaddedString: toPaddedString,
    abs:            abs,
    round:          round,
    ceil:           ceil,
    floor:          floor
  };
})());
function $R(start, end, exclusive) {
  return new ObjectRange(start, end,
exclusive);
}
var ObjectRange = Class.create(Enumerable,
(function() {
  function initialize(start, end, exclusive)
{
    this.start = start;
    this.end = end;
    this.exclusive = exclusive;
  }
  function _each(iterator) {
    var value = this.start;
    while (this.include(value)) {
      iterator(value);
      value = value.succ();
    }
  }
  function include(value) {
    if (value < this.start)
      return false;
    if (this.exclusive)
      return value < this.end;
    return value <= this.end;
  }
  return {
    initialize: initialize,
    _each:      _each,
    include:    include
  };
})());
var Ajax = {
  getTransport: function() {
    return Try.these(
      function() {return new
XMLHttpRequest()},
      function() {return new
ActiveXObject(,Msxml2.XMLHTTP‘)},
      function() {return new
ActiveXObject(,Microsoft.XMLHTTP‘)}
    ) || false;
  },
  activeRequestCount: 0
};
Ajax.Responders = {
  responders: [],
  _each: function(iterator) {
    this.responders._each(iterator);
  },
  register: function(responder) {
    if (!this.include(responder))
      this.responders.push(responder);
  },
  unregister: function(responder) {
    this.responders = this.responders.
without(responder);
  },
  dispatch: function(callback, request,
transport, json) {
    this.each(function(responder) {
      if (Object.
isFunction(responder[callback])) {
        try {
          responder[callback].
apply(responder, [request, transport,
json]);
        } catch (e) { }
      }
    });
  }
};
Object.extend(Ajax.Responders, Enumerable);
Ajax.Responders.register({
  onCreate:   function() { Ajax.
activeRequestCount++ },
  onComplete: function() { Ajax.
activeRequestCount-- }
});
Ajax.Base = Class.create({
  initialize: function(options) {
    this.options = {
      method:       ‚post‘,
      asynchronous: true,
      contentType:  ‚application/x-www-form-
urlencoded‘,
      encoding:     ‚UTF-8‘,
      parameters:   ‚‘,
      evalJSON:     true,
      evalJS:       true
    };
    Object.extend(this.options, options ||
{ });
    this.options.method = this.options.
method.toLowerCase();
    if (Object.isHash(this.options.
parameters))
      this.options.parameters = this.
options.parameters.toObject();
  }
});
Ajax.Request = Class.create(Ajax.Base, {
  _complete: false,
  initialize: function($super, url, options)
{
    $super(options);
    this.transport = Ajax.getTransport();
    this.request(url);
  },
  request: function(url) {
    this.url = url;
    this.method = this.options.method;
    var params = Object.isString(this.
options.parameters) ?
          this.options.parameters :
          Object.toQueryString(this.options.
parameters);
    if (!![,get‘, ‚post‘].include(this.
method)) {
      params += (params ? ‚&‘ : ‚‘) + „_
method=“ + this.method;
      this.method = ‚post‘;
    }
    if (params && this.method === ‚get‘) {
      this.url += (this.url.include(‚?‘) ?
‚&‘ : ‚?‘) + params;
    }
    this.parameters = params.
toQueryParams();
    try {
      var response = new Ajax.
Response(this);
      if (this.options.onCreate) this.
options.onCreate(response);
      Ajax.Responders.dispatch(‚onCreate‘,
this, response);
      this.transport.open(this.method.
toUpperCase(), this.url,
        this.options.asynchronous);
      if (this.options.asynchronous) this.
respondToReadyState.bind(this).defer(1);
      this.transport.onreadystatechange =
this.onStateChange.bind(this);
      this.setRequestHeaders();
      this.body = this.method == ‚post‘ ?
(this.options.postBody || params) : null;
      this.transport.send(this.body);
      /* Force Firefox to handle ready state
4 for synchronous requests */
      if (!this.options.asynchronous &&
this.transport.overrideMimeType)
        this.onStateChange();
    }
    catch (e) {
      this.dispatchException(e);
    }
  },
  onStateChange: function() {
    var readyState = this.transport.
readyState;
    if (readyState > 1 && !((readyState ==
4) && this._complete))
      this.respondToReadyState(this.
transport.readyState);
  },
  setRequestHeaders: function() {
    var headers = {
      ‚X-Requested-With‘: ‚XMLHttpRequest‘,
      ‚X-Prototype-Version‘: Prototype.
Version,
      ‚Accept‘: ‚text/javascript, text/html,
application/xml, text/xml, */*‘
    };
    if (this.method == ‚post‘) {
      headers[‚Content-type‘] = this.
options.contentType +
        (this.options.encoding ? ‚;
charset=‘ + this.options.encoding : ‚‘);
      /* Force „Connection: close“ for older
Mozilla browsers to work
       * around a bug where XMLHttpRequest
sends an incorrect
       * Content-length header. See Mozilla
Bugzilla #246651.
       */
      if (this.transport.overrideMimeType &&
          (navigator.userAgent.match(/
Gecko\/(\d{4})/) || [0,2005])[1] < 2005)
        headers[‚Connection‘] = ‚close‘;
    }
    if (typeof this.options.requestHeaders
== ‚object‘) {
      var extras = this.options.
requestHeaders;
      if (Object.isFunction(extras.push))
        for (var i = 0, length = extras.
length; i < length; i += 2)
          headers[extras[i]] = extras[i+1];
      else
        $H(extras).each(function(pair) {
headers[pair.key] = pair.value });
    }
    for (var name in headers)
      this.transport.setRequestHeader(name,
headers[name]);
  },
  success: function() {
    var status = this.getStatus();
    return !status || (status >= 200 &&
status < 300) || status == 304;
  },
  getStatus: function() {
    try {
      if (this.transport.status === 1223)
return 204;
      return this.transport.status || 0;
    } catch (e) { return 0 }
  },
  respondToReadyState: function(readyState)
{
    var state = Ajax.Request.
Events[readyState], response = new Ajax.
Response(this);
    if (state == ‚Complete‘) {
      try {
        this._complete = true;
        (this.options[‚on‘ + response.
status]
         || this.options[‚on‘ + (this.
success() ? ‚Success‘ : ‚Failure‘)]
         || Prototype.emptyFunction)
(response, response.headerJSON);
      } catch (e) {
        this.dispatchException(e);
      }
      var contentType = response.
getHeader(‚Content-type‘);
      if (this.options.evalJS == ‚force‘
          || (this.options.evalJS && this.
isSameOrigin() && contentType
          && contentType.match(/^\
s*(text|application)\/(x-)?(java|ecma)
script(;.*)?\s*$/i)))
        this.evalResponse();
    }
    try {
      (this.options[‚on‘ + state] ||
Prototype.emptyFunction)(response, response.
headerJSON);
      Ajax.Responders.dispatch(‚on‘ + state,
this, response, response.headerJSON);
    } catch (e) {
      this.dispatchException(e);
    }
    if (state == ‚Complete‘) {
      this.transport.onreadystatechange =
Prototype.emptyFunction;
    }
  },
  isSameOrigin: function() {
    var m = this.url.match(/^\s*https?:\/\/
[^\/]*/);
    return !m || (m[0] ==
‚#{protocol}//#{domain}#{port}‘.
interpolate({
      protocol: location.protocol,
      domain: document.domain,
      port: location.port ? ‚:‘ + location.
port : ‚‘
    }));
  },
  getHeader: function(name) {
    try {
      return this.transport.
getResponseHeader(name) || null;
    } catch (e) { return null; }
  },
  evalResponse: function() {
    try {
      return eval((this.transport.
responseText || ‚‘).unfilterJSON());
    } catch (e) {
      this.dispatchException(e);
    }
  },
  dispatchException: function(exception) {
    (this.options.onException || Prototype.
emptyFunction)(this, exception);
    Ajax.Responders.dispatch(‚onException‘,
this, exception);
  }
});
Ajax.Request.Events =
  [‚Uninitialized‘, ‚Loading‘, ‚Loaded‘,
‚Interactive‘, ‚Complete‘];
Ajax.Response = Class.create({
  initialize: function(request){
    this.request = request;
    var transport  = this.transport  =
request.transport,
        readyState = this.readyState =
transport.readyState;
    if ((readyState > 2 && !Prototype.
Browser.IE) || readyState == 4) {
      this.status       = this.getStatus();
      this.statusText   = this.
getStatusText();
      this.responseText = String.
interpret(transport.responseText);
      this.headerJSON   = this._
getHeaderJSON();
    }
    if (readyState == 4) {
      var xml = transport.responseXML;
      this.responseXML  = Object.
isUndefined(xml) ? null : xml;
      this.responseJSON = this._
getResponseJSON();
    }
  },
  status:      0,
  statusText: ‚‘,
  getStatus: Ajax.Request.prototype.
getStatus,
  getStatusText: function() {
    try {
      return this.transport.statusText ||
‚‘;
    } catch (e) { return ‚ ‘ }
  },
  getHeader: Ajax.Request.prototype.
getHeader,
  getAllHeaders: function() {
    try {
      return this.getAllResponseHeaders();
    } catch (e) { return null }
  },
  getResponseHeader: function(name) {
    return this.transport.
getResponseHeader(name);
  },
  getAllResponseHeaders: function() {
    return this.transport.
getAllResponseHeaders();
  },
  _getHeaderJSON: function() {
    var json = this.getHeader(‚X-JSON‘);
    if (!json) return null;
    json = decodeURIComponent(escape(json));
    try {
      return json.evalJSON(this.request.
options.sanitizeJSON ||
        !this.request.isSameOrigin());
    } catch (e) {
      this.request.dispatchException(e);
    }
  },
  _getResponseJSON: function() {
    var options = this.request.options;
    if (!options.evalJSON || (options.
evalJSON != ‚force‘ &&
        !(this.getHeader(‚Content-type‘) ||
‚‘).include(‚application/json‘)) ||
        this.responseText.blank())
          return null;
    try {
      return this.responseText.
evalJSON(options.sanitizeJSON ||
        !this.request.isSameOrigin());
    } catch (e) {
      this.request.dispatchException(e);
    }
  }
});
Ajax.Updater = Class.create(Ajax.Request, {
  initialize: function($super, container,
url, options) {
    this.container = {
      success: (container.success ||
container),
      failure: (container.failure ||
(container.success ? null : container))
    };
    options = Object.clone(options);
    var onComplete = options.onComplete;
    options.onComplete = (function(response,
json) {
      this.updateContent(response.
responseText);
      if (Object.isFunction(onComplete))
onComplete(response, json);
    }).bind(this);
    $super(url, options);
  },
  updateContent: function(responseText) {
    var receiver = this.container[this.
success() ? ‚success‘ : ‚failure‘],
        options = this.options;
    if (!options.evalScripts) responseText =
```

```
responseText.stripScripts();
    if (receiver = $(receiver)) {
      if (options.insertion) {
        if (Object.isString(options.
insertion)) {
          var insertion = { };
insertion[options.insertion] = responseText;
          receiver.insert(insertion);
        }
        else options.insertion(receiver,
responseText);
      }
      else receiver.update(responseText);
    }
  }
});
Ajax.PeriodicalUpdater = Class.create(Ajax.
Base, {
  initialize: function($super, container,
url, options) {
    $super(options);
    this.onComplete = this.options.
onComplete;
    this.frequency = (this.options.frequency
|| 2);
    this.decay = (this.options.decay || 1);
    this.updater = { };
    this.container = container;
    this.url = url;
    this.start();
  },
  start: function() {
    this.options.onComplete = this.
updateComplete.bind(this);
    this.onTimerEvent();
  },
  stop: function() {
    this.updater.options.onComplete =
undefined;
    clearTimeout(this.timer);
    (this.onComplete || Prototype.
emptyFunction).apply(this, arguments);
  },
  updateComplete: function(response) {
    if (this.options.decay) {
      this.decay = (response.responseText ==
this.lastText ?
        this.decay * this.options.decay :
1);
      this.lastText = response.responseText;
    }
    this.timer = this.onTimerEvent.
bind(this).delay(this.decay * this.
frequency);
  },
  onTimerEvent: function() {
    this.updater = new Ajax.Updater(this.
container, this.url, this.options);
  }
});
function $(element) {
  if (arguments.length > 1) {
    for (var i = 0, elements = [], length =
arguments.length; i < length; i++)
      elements.push($(arguments[i]));
    return elements;
  }
  if (Object.isString(element))
    element = document.
getElementById(element);
  return Element.extend(element);
}
if (Prototype.BrowserFeatures.XPath) {
  document._getElementsByXPath =
function(expression, parentElement) {
    var results = [];
    var query = document.
evaluate(expression, $(parentElement) ||
document,
      null, XPathResult.ORDERED_NODE_
SNAPSHOT_TYPE, null);
    for (var i = 0, length = query.
snapshotLength; i < length; i++)
      results.push(Element.extend(query.
snapshotItem(i)));
    return results;
  };
}
/*--------------------------------------------
--------------------------------*/
if (!Node) var Node = { };
if (!Node.ELEMENT_NODE) {
  Object.extend(Node, {
    ELEMENT_NODE: 1,
    ATTRIBUTE_NODE: 2,
    TEXT_NODE: 3,
    CDATA_SECTION_NODE: 4,
    ENTITY_REFERENCE_NODE: 5,
    ENTITY_NODE: 6,
    PROCESSING_INSTRUCTION_NODE: 7,
    COMMENT_NODE: 8,
    DOCUMENT_NODE: 9,
    DOCUMENT_TYPE_NODE: 10,
    DOCUMENT_FRAGMENT_NODE: 11,
    NOTATION_NODE: 12
  });
}
(function(global) {
  function shouldUseCache(tagName,
attributes) {
    if (tagName === ‚select‘) return false;
    if (‚type‘ in attributes) return false;
    return true;
  }
  var HAS_EXTENDED_CREATE_ELEMENT_SYNTAX =
(function(){
    try {
      var el = document.
createElement(‚<input name="x">‘);
      return el.tagName.toLowerCase() ===
‚input‘ && el.name === ‚x‘;
    }
    catch(err) {
      return false;
    }
  })();
  var element = global.Element;
  global.Element = function(tagName,
attributes) {
    attributes = attributes || { };
    tagName = tagName.toLowerCase();
    var cache = Element.cache;
    if (HAS_EXTENDED_CREATE_ELEMENT_SYNTAX
&& attributes.name) {
      tagName = ‚<‘ + tagName + ‚ name="‘ +
attributes.name + ‚">‘;
      delete attributes.name;
      return Element.
writeAttribute(document.
createElement(tagName), attributes);
    }
    if (!cache[tagName]) cache[tagName]
= Element.extend(document.
createElement(tagName));
    var node = shouldUseCache(tagName,
attributes) ?
      cache[tagName].cloneNode(false) :
document.createElement(tagName);
    return Element.writeAttribute(node,
attributes);
  };
  Object.extend(global.Element, element ||
{ });
  if (element) global.Element.prototype =
element.prototype;
})(this);
Element.idCounter = 1;
Element.cache = { };
Element._purgeElement = function(element) {
  var uid = element._prototypeUID;
  if (uid) {
    Element.stopObserving(element);
    element._prototypeUID = void 0;
    delete Element.Storage[uid];
  }
}
Element.Methods = {
  visible: function(element) {
    return $(element).style.display !=
‚none‘;
  },
  toggle: function(element) {
    element = $(element);
    Element[Element.visible(element) ?
‚hide‘ : ‚show‘](element);
    return element;
  },
  hide: function(element) {
    element = $(element);
    element.style.display = ‚none‘;
    return element;
  },
  show: function(element) {
    element = $(element);
    element.style.display = ‚‘;
    return element;
  },
  remove: function(element) {
    element = $(element);
    element.parentNode.removeChild(element);
    return element;
  },
  update: (function(){
    var SELECT_ELEMENT_INNERHTML_BUGGY =
(function(){
      var el = document.
createElement(‚select‘),
          isBuggy = true;
      el.innerHTML = ‚<option
value=\‘test\‘>test</option>‘;
      if (el.options && el.options[0]) {
        isBuggy = el.options[0].nodeName.
toUpperCase() !== ‚OPTION‘;
      }
      el = null;
      return isBuggy;
    })();
    var TABLE_ELEMENT_INNERHTML_BUGGY =
(function(){
      try {
        var el = document.
createElement(‚table‘);
        if (el && el.tBodies) {
          el.innerHTML =
‚<tbody><tr><td>test</td></tr></tbody>‘;
          var isBuggy = typeof el.tBodies[0]
== ‚undefined‘;
          el = null;
          return isBuggy;
        }
      } catch (e) {
        return true;
      }
    })();
    var LINK_ELEMENT_INNERHTML_BUGGY =
(function() {
      try {
        var el = document.
createElement(‚div‘);
        el.innerHTML = ‚<link>‘;
        var isBuggy = (el.childNodes.length
=== 0);
        el = null;
        return isBuggy;
      } catch(e) {
        return true;
      }
    })();
    var ANY_INNERHTML_BUGGY = SELECT_
ELEMENT_INNERHTML_BUGGY ||
      TABLE_ELEMENT_INNERHTML_BUGGY || LINK_
ELEMENT_INNERHTML_BUGGY;
    var SCRIPT_ELEMENT_REJECTS_TEXTNODE_
APPENDING = (function () {
      var s = document.
createElement(‚script‘),
          isBuggy = false;
      try {
        s.appendChild(document.
createTextNode(‚‘));
        isBuggy = !s.firstChild ||
          s.firstChild && s.firstChild.
nodeType !== 3;
      } catch (e) {
        isBuggy = true;
      }
      s = null;
      return isBuggy;
    })();
    function update(element, content) {
      element = $(element);
      var purgeElement = Element._
purgeElement;
      var descendants = element.
getElementsByTagName(‚*‘),
       i = descendants.length;
      while (i--)
purgeElement(descendants[i]);
      if (content && content.toElement)
        content = content.toElement();
      if (Object.isElement(content))
        return element.update().
insert(content);
      content = Object.toHTML(content);
      var tagName = element.tagName.
toUpperCase();
      if (tagName === ‚SCRIPT‘ && SCRIPT_
ELEMENT_REJECTS_TEXTNODE_APPENDING) {
        element.text = content;
        return element;
      }
      if (ANY_INNERHTML_BUGGY) {
        if (tagName in Element._
insertionTranslations.tags) {
          while (element.firstChild) {
            element.removeChild(element.
firstChild);
          }
          Element._
getContentFromAnonymousElement(tagName,
content.stripScripts())
            .each(function(node) {
              element.appendChild(node)
            });
        } else if (LINK_ELEMENT_INNERHTML_
BUGGY && Object.isString(content) &&
content.indexOf(‚<link‘) > -1) {
          while (element.firstChild) {
            element.removeChild(element.
firstChild);
          }
          var nodes = Element._
getContentFromAnonymousElement(tagName,
content.stripScripts(), true);
          nodes.each(function(node) {
element.appendChild(node) });
        }
        else {
          element.innerHTML = content.
stripScripts();
        }
      }
      else {
        element.innerHTML = content.
stripScripts();
      }
      content.evalScripts.bind(content).
defer();
      return element;
    }
    return update;
  })(),
  replace: function(element, content) {
    element = $(element);
    if (content && content.toElement)
content = content.toElement();
    else if (!Object.isElement(content)) {
      content = Object.toHTML(content);
      var range = element.ownerDocument.
createRange();
      range.selectNode(element);
      content.evalScripts.bind(content).
defer();
      content = range.
createContextualFragment(content.
stripScripts());
    }
    element.parentNode.replaceChild(content,
element);
    return element;
  },
  insert: function(element, insertions) {
    element = $(element);
    if (Object.isString(insertions) ||
Object.isNumber(insertions) ||
        Object.isElement(insertions) ||
(insertions && (insertions.toElement ||
insertions.toHTML)))
          insertions = {bottom:insertions};
    var content, insert, tagName,
childNodes;
    for (var position in insertions) {
      content  = insertions[position];
      position = position.toLowerCase();
      insert = Element._
insertionTranslations[position];
      if (content && content.toElement)
content = content.toElement();
      if (Object.isElement(content)) {
        insert(element, content);
        continue;
      }
      content = Object.toHTML(content);
      tagName = ((position == ‚before‘ ||
position == ‚after‘)
        ? element.parentNode : element).
tagName.toUpperCase();
      childNodes = Element._
getContentFromAnonymousElement(tagName,
content.stripScripts());
      if (position == ‚top‘ || position ==
‚after‘) childNodes.reverse();
      childNodes.each(insert.
curry(element));
      content.evalScripts.bind(content).
defer();
    }
    return element;
  },
  wrap: function(element, wrapper,
attributes) {
    element = $(element);
    if (Object.isElement(wrapper))
      $(wrapper).writeAttribute(attributes
|| { });
    else if (Object.isString(wrapper))
wrapper = new Element(wrapper, attributes);
    else wrapper = new Element(‚div‘,
wrapper);
    if (element.parentNode)
      element.parentNode.
replaceChild(wrapper, element);
    wrapper.appendChild(element);
    return wrapper;
  },
  inspect: function(element) {
    element = $(element);
    var result = ‚<‘ + element.tagName.
toLowerCase();
    $H({‚id‘: ‚id‘, ‚className‘: ‚class‘}).
each(function(pair) {
      var property = pair.first(),
          attribute = pair.last(),
          value = (element[property] || ‚‘).
toString();
      if (value) result += ‚ ‘ + attribute +
‚=‘ + value.inspect(true);
    });
    return result + ‚>‘;
  },
  recursivelyCollect: function(element,
property, maximumLength) {
    element = $(element);
    maximumLength = maximumLength || -1;
    var elements = [];
    while (element = element[property]) {
      if (element.nodeType == 1)
        elements.push(Element.
extend(element));
      if (elements.length == maximumLength)
        break;
    }
    return elements;
  },
  ancestors: function(element) {
    return Element.
recursivelyCollect(element, ‚parentNode‘);
  },
  descendants: function(element) {
    return Element.select(element, „*“);
  },
  firstDescendant: function(element) {
    element = $(element).firstChild;
    while (element && element.nodeType != 1)
element = element.nextSibling;
    return $(element);
  },
  immediateDescendants: function(element) {
    var results = [], child = $(element).
firstChild;
    while (child) {
      if (child.nodeType === 1) {
        results.push(Element.extend(child));
      }
      child = child.nextSibling;
    }
    return results;
  },
  previousSiblings: function(element,
maximumLength) {
    return Element.
recursivelyCollect(element,
‚previousSibling‘);
  },
  nextSiblings: function(element) {
    return Element.
recursivelyCollect(element, ‚nextSibling‘);
  },
  siblings: function(element) {
    element = $(element);
    return Element.
previousSiblings(element).reverse()
      .concat(Element.
nextSiblings(element));
  },
  match: function(element, selector) {
    element = $(element);
    if (Object.isString(selector))
      return Prototype.Selector.
match(element, selector);
    return selector.match(element);
  },
  up: function(element, expression, index) {
    element = $(element);
    if (arguments.length == 1) return
$(element.parentNode);
    var ancestors = Element.
ancestors(element);
    return Object.isNumber(expression) ?
ancestors[expression] :
      Prototype.Selector.find(ancestors,
expression, index);
  },
  down: function(element, expression, index)
{
    element = $(element);
    if (arguments.length == 1) return
Element.firstDescendant(element);
    return Object.isNumber(expression) ?
Element.descendants(element)[expression] :
      Element.select(element, expression)
[index || 0];
  },
  previous: function(element, expression,
index) {
    element = $(element);
    if (Object.isNumber(expression)) index =
expression, expression = false;
    if (!Object.isNumber(index)) index = 0;
    if (expression) {
      return Prototype.Selector.find(element.
previousSiblings(), expression, index);
    } else {
      return element.
recursivelyCollect(„previousSibling“, index
+ 1)[index];
    }
  },
  next: function(element, expression, index)
{
    element = $(element);
    if (Object.isNumber(expression)) index =
expression, expression = false;
    if (!Object.isNumber(index)) index = 0;
    if (expression) {
      return Prototype.Selector.find(element.
nextSiblings(), expression, index);
    } else {
      var maximumLength = Object.
isNumber(index) ? index + 1 : 1;
      return element.
recursivelyCollect(„nextSibling“, index + 1)
[index];
    }
  },
  select: function(element) {
    element = $(element);
    var expressions = Array.prototype.slice.
call(arguments, 1).join(‚, ‚);
    return Prototype.Selector.
select(expressions, element);
  },
  adjacent: function(element) {
    element = $(element);
    var expressions = Array.prototype.slice.
call(arguments, 1).join(‚, ‚);
    return Prototype.Selector.
select(expressions, element.parentNode).
without(element);
  },
  identify: function(element) {
    element = $(element);
    var id = Element.readAttribute(element,
‚id‘);
    if (id) return id;
    do { id = ‚anonymous_element_‘ +
Element.idCounter++ } while ($(id));
    Element.writeAttribute(element, ‚id‘,
id);
    return id;
  },
  readAttribute: function(element, name) {
    element = $(element);
    if (Prototype.Browser.IE) {
      var t = Element._
attributeTranslations.read;
      if (t.values[name]) return
t.values[name](element, name);
      if (t.names[name]) name =
t.names[name];
      if (name.include(‚:‘)) {
        return (!element.attributes ||
!element.attributes[name]) ? null :
          element.attributes[name].value;
      }
    }
    return element.getAttribute(name);
  },
  writeAttribute: function(element, name,
value) {
    element = $(element);
    var attributes = { }, t = Element._
attributeTranslations.write;
    if (typeof name == ‚object‘) attributes
= name;
    else attributes[name] = Object.
isUndefined(value) ? true : value;
    for (var attr in attributes) {
      name = t.names[attr] || attr;
      value = attributes[attr];
      if (t.values[attr]) name =
t.values[attr](element, value);
      if (value === false || value === null)
        element.removeAttribute(name);
      else if (value === true)
        element.setAttribute(name, name);
      else element.setAttribute(name,
value);
    }
    return element;
  },
  getHeight: function(element) {
    return Element.getDimensions(element).
height;
  },
  getWidth: function(element) {
    return Element.getDimensions(element).
```

Data Analyzer Script by Peter Vojtek

```
width;
  },
  classNames: function(element) {
    return new Element.ClassNames(element);
  },
  hasClassName: function(element, className)
{
    if (!(element = $(element))) return;
    var elementClassName = element.
className;
    return (elementClassName.length > 0 &&
(elementClassName == className ||
      new RegExp(„(^|\\s)“ + className +
„(\\s|$)“).test(elementClassName)));
  },
  addClassName: function(element, className)
{
    if (!(element = $(element))) return;
    if (!Element.hasClassName(element,
className))
      element.className += (element.
className ? ‚ ‚ : ‚‘) + className;
    return element;
  },
  removeClassName: function(element,
className) {
    if (!(element = $(element))) return;
    element.className = element.className.
replace(
      new RegExp(„(^|\\s+)“ + className +
„(\\s+|$)“), ‚ ‚).strip();
    return element;
  },
  toggleClassName: function(element,
className) {
    if (!(element = $(element))) return;
    return Element[Element.
hasClassName(element, className) ?
      ‚removeClassName‘ : ‚addClassName‘]
(element, className);
  },
  cleanWhitespace: function(element) {
    element = $(element);
    var node = element.firstChild;
    while (node) {
      var nextNode = node.nextSibling;
      if (node.nodeType == 3 && !/\S/.
test(node.nodeValue))
        element.removeChild(node);
      node = nextNode;
    }
    return element;
  },
  empty: function(element) {
    return $(element).innerHTML.blank();
  },
  descendantOf: function(element, ancestor)
{
    element = $(element), ancestor =
$(ancestor);
    if (element.compareDocumentPosition)
      return (element.
compareDocumentPosition(ancestor) & 8) ===
8;
    if (ancestor.contains)
      return ancestor.contains(element) &&
ancestor !== element;
    while (element = element.parentNode)
      if (element == ancestor) return true;
    return false;
  },
  scrollTo: function(element) {
    element = $(element);
    var pos = Element.
cumulativeOffset(element);
    window.scrollTo(pos[0], pos[1]);
    return element;
  },
  getStyle: function(element, style) {
    element = $(element);
    style = style == ‚float‘ ? ‚cssFloat‘ :
style.camelize();
    var value = element.style[style];
    if (!value || value == ‚auto‘) {
      var css = document.defaultView.
getComputedStyle(element, null);
      value = css ? css[style] : null;
    }
    if (style == ‚opacity‘) return value ?
parseFloat(value) : 1.0;
    return value == ‚auto‘ ? null : value;
  },
  getOpacity: function(element) {
    return $(element).getStyle(‚opacity‘);
  },
  setStyle: function(element, styles) {
    element = $(element);
    var elementStyle = element.style, match;
    if (Object.isString(styles)) {
      element.style.cssText += ‚;‘ + styles;
      return styles.include(‚opacity‘) ?
        element.setOpacity(styles.match(/
opacity:\s*(\d?\.?\d*)/)[1]) : element;
    }
    for (var property in styles)
      if (property == ‚opacity‘) element.
setOpacity(styles[property]);
      else
        elementStyle[(property == ‚float‘ ||
property == ‚cssFloat‘) ?
            (Object.isUndefined(elementStyle.
styleFloat) ? ‚cssFloat‘ : ‚styleFloat‘) :
             property] = styles[property];
    return element;
  },
  setOpacity: function(element, value) {
    element = $(element);
    element.style.opacity = (value == 1 ||
value === ‚‘) ? ‚‘ :
      (value < 0.00001) ? 0 : value;
    return element;
  },
  makePositioned: function(element) {
    element = $(element);
    var pos = Element.getStyle(element,
‚position‘);
    if (pos == ‚static‘ || !pos) {
      element._madePositioned = true;
      element.style.position = ‚relative‘;
      if (Prototype.Browser.Opera) {
        element.style.top = 0;
        element.style.left = 0;
      }
    }
    return element;
  },
  undoPositioned: function(element) {
    element = $(element);
    if (element._madePositioned) {
      element._madePositioned = undefined;
      element.style.position =
        element.style.top =
        element.style.left =
        element.style.bottom =
        element.style.right = ‚;‘;
    }
    return element;
  },
  makeClipping: function(element) {
    element = $(element);
    if (element._overflow) return element;
    element._overflow = Element.
getStyle(element, ‚overflow‘) || ‚auto‘;
    if (element._overflow !== ‚hidden‘)
      element.style.overflow = ‚hidden‘;
    return element;
  },
  undoClipping: function(element) {
    element = $(element);
    if (!element._overflow) return element;
    element.style.overflow = element._overflow
== ‚auto‘ ? ‚‘ : element._overflow;
    element._overflow = null;
    return element;
  },
  clonePosition: function(element, source) {
    var options = Object.extend({
      setLeft:    true,
      setTop:     true,
      setWidth:   true,
      setHeight:  true,
      offsetTop:  0,
      offsetLeft: 0
    }, arguments[2] || { });
    source = $(source);
    var p = Element.viewportOffset(source),
delta = [0, 0], parent = null;
    element = $(element);
    if (Element.getStyle(element,
‚position‘) == ‚absolute‘) {
      parent = Element.
getOffsetParent(element);
      delta = Element.viewportOffset(parent);
    }
    if (parent == document.body) {
      delta[0] -= document.body.offsetLeft;
      delta[1] -= document.body.offsetTop;
    }
    if (options.setLeft)   element.style.
left  = (p[0] - delta[0] + options.
offsetLeft) + ‚px‘;
    if (options.setTop)    element.style.
top   = (p[1] - delta[1] + options.offsetTop)
+ ‚px‘;
    if (options.setWidth)  element.style.
width = source.offsetWidth + ‚px‘;
    if (options.setHeight) element.style.
height = source.offsetHeight + ‚px‘;
    return element;
  }
};
Object.extend(Element.Methods, {
  getElementsBySelector: Element.Methods.
select,
  childElements: Element.Methods.
immediateDescendants
});
Element._attributeTranslations = {
  write: {
    names: {
      className: ‚class‘,
      htmlFor:   ‚for‘
    },
    values: { }
  }
};
if (Prototype.Browser.Opera) {
  Element.Methods.getStyle = Element.
Methods.getStyle.wrap(
    function(proceed, element, style) {
      switch (style) {
        case ‚height‘: case ‚width‘:
          if (!Element.visible(element))
return null;
          var dim =
parseInt(proceed(element, style), 10);
          if (dim !== element[‚offset‘ +
style.capitalize()])
            return dim + ‚px‘;
          var properties;
          if (style === ‚height‘) {
            properties = [‚border-top-
width‘, ‚padding-top‘,
              ‚padding-bottom‘, ‚border-
bottom-width‘];
          }
          else {
            properties = [‚border-left-
width‘, ‚padding-left‘,
              ‚padding-right‘, ‚border-right-
width‘];
          }
          return properties.inject(dim,
function(memo, property) {
            var val = proceed(element,
property);
            return val === null ? memo :
memo - parseInt(val, 10);
          }) + ‚px‘;
        default: return proceed(element,
style);
      }
    }
  );
  Element.Methods.readAttribute = Element.
Methods.readAttribute.wrap(
    function(proceed, element, attribute) {
      if (attribute === ‚title‘) return
element.title;
      return proceed(element, attribute);
    }
  );
}
else if (Prototype.Browser.IE) {
  Element.Methods.getStyle =
function(element, style) {
    element = $(element);
    style = (style == ‚float‘ || style
== ‚cssFloat‘) ? ‚styleFloat‘ : style.
camelize();
    var value = element.style[style];
    if (!value && element.currentStyle)
value = element.currentStyle[style];
    if (style == ‚opacity‘) {
      if (value = (element.getStyle(‚filter‘)
|| ‚‘).match(/alpha\(opacity=(.*)\)/))
        if (value[1]) return
parseFloat(value[1]) / 100;
      return 1.0;
    }
    if (value == ‚auto‘) {
      if ((style == ‚width‘ || style ==
‚height‘) && (element.getStyle(‚display‘) !=
‚none‘))
        return element[‚offset‘ + style.
capitalize()] + ‚px‘;
      return null;
    }
    return value;
  };
  Element.Methods.setOpacity =
function(element, value) {
    function stripAlpha(filter){
      return filter.replace(/alpha\([^\)]*\)/
gi,‘‘);
    }
    element = $(element);
    var currentStyle = element.currentStyle;
    if ((currentStyle && !currentStyle.
hasLayout) ||
        (!currentStyle && element.style.zoom
== ‚normal‘))
      element.style.zoom = 1;
    var filter = element.getStyle(‚filter‘),
style = element.style;
    if (value == 1 || value === ‚‘) {
      (filter = stripAlpha(filter)) ?
        style.filter = filter : style.
removeAttribute(‚filter‘);
      return element;
    } else if (value < 0.00001) value = 0;
    style.filter = stripAlpha(filter) +
      ‚alpha(opacity=‘ + (value * 100) +
‚)‘;
    return element;
  };
  Element._attributeTranslations =
(function(){
    var classProp = ‚className‘,
        forProp = ‚for‘,
        el = document.createElement(‚div‘);
    el.setAttribute(classProp, ‚x‘);
    if (el.className !== ‚x‘) {
      el.setAttribute(‚class‘, ‚x‘);
      if (el.className === ‚x‘) {
        classProp = ‚class‘;
      }
    }
    el = null;
    el = document.createElement(‚label‘);
    el.setAttribute(forProp, ‚x‘);
    if (el.htmlFor !== ‚x‘) {
      el.setAttribute(‚htmlFor‘, ‚x‘);
      if (el.htmlFor === ‚x‘) {
        forProp = ‚htmlFor‘;
      }
    }
    el = null;
    return {
      read: {
        names: {
          ‚class‘:      classProp,
          ‚className‘:  classProp,
          ‚for‘:        forProp,
          ‚htmlFor‘:    forProp
        },
        values: {
          _getAttr: function(element,
attribute) {
            return element.
getAttribute(attribute);
          },
          _getAttr2: function(element,
attribute) {
            return element.
getAttribute(attribute, 2);
          },
          _getAttrNode: function(element,
attribute) {
            var node = element.
getAttributeNode(attribute);
            return node ? node.value : „“;
          },
          _getEv: (function(){
            var el = document.
createElement(‚div‘), f;
            el.onclick = Prototype.
emptyFunction;
            var value =
el.getAttribute(‚onclick‘);
            if (String(value).indexOf(‚{‚)
> -1) {
              f = function(element,
attribute) {
                attribute = element.
getAttribute(attribute);
                if (!attribute) return null;
                attribute = attribute.
toString();
                attribute = attribute.
split(‚{‚)[1];
                attribute = attribute.
split(‚}‘)[0];
                return attribute.strip();
              };
            }
            else if (value === ‚‘) {
              f = function(element,
attribute) {
                attribute = element.
getAttribute(attribute);
                if (!attribute) return null;
                return attribute.strip();
              };
            }
            el = null;
            return f;
          })(),
          _flag: function(element, attribute)
{
            return $(element).
hasAttribute(attribute) ? attribute : null;
          },
          style: function(element) {
            return element.style.cssText.
toLowerCase();
          },
          title: function(element) {
            return element.title;
          }
        }
      }
    };
  })();
  Element._attributeTranslations.write = {
    names: Object.extend({
      cellpadding: ‚cellPadding‘,
      cellspacing: ‚cellSpacing‘
    }, Element._attributeTranslations.read.
names),
    values: {
      checked: function(element, value) {
        element.checked = !!value;
      },
      style: function(element, value) {
        element.style.cssText = value ?
value : ‚‘;
      }
    }
  };
  Element._attributeTranslations.has = {};
  $w(‚colSpan rowSpan vAlign dateTime
accessKey tabIndex , +
      ‚encType maxLength readOnly longDesc
frameBorder‘).each(function(attr) {
    Element._attributeTranslations.write.
names[attr.toLowerCase()] = attr;
    Element._attributeTranslations.has[attr.
toLowerCase()] = attr;
  });
  (function(v) {
    Object.extend(v, {
      href:        v._getAttr2,
      src:         v._getAttr2,
      type:        v._getAttr,
      action:      v._getAttrNode,
      disabled:    v._flag,
      checked:     v._flag,
      readonly:    v._flag,
      multiple:    v._flag,
      onload:      v._getEv,
      onunload:    v._getEv,
      onclick:     v._getEv,
      ondblclick:  v._getEv,
      onmousedown: v._getEv,
      onmouseup:   v._getEv,
      onmouseover: v._getEv,
      onmousemove: v._getEv,
      onmouseout:  v._getEv,
      onfocus:     v._getEv,
      onblur:      v._getEv,
      onkeypress:  v._getEv,
      onkeydown:   v._getEv,
      onkeyup:     v._getEv,
      onsubmit:    v._getEv,
      onreset:     v._getEv,
      onselect:    v._getEv,
      onchange:    v._getEv
    });
  })(Element._attributeTranslations.read.
values);
  if (Prototype.BrowserFeatures.
ElementExtensions) {
    (function() {
      function _descendants(element) {
        var nodes = element.
getElementsByTagName(‚*‘), results = [];
        for (var i = 0, node; node =
nodes[i]; i++)
          if (node.tagName !== „!“) //
Filter out comment nodes.
            results.push(node);
        return results;
      }
      Element.Methods.down =
function(element, expression, index) {
        element = $(element);
        if (arguments.length == 1) return
element.firstDescendant();
        return Object.isNumber(expression) ?
_descendants(element)[expression] :
          Element.select(element,
expression)[index || 0];
      }
    })();
  }
}
else if (Prototype.Browser.Gecko && /
rv:1\.8\.0/.test(navigator.userAgent)) {
  Element.Methods.setOpacity =
function(element, value) {
    element = $(element);
    element.style.opacity = (value == 1) ?
0.999999 :
      (value === ‚‘) ? ‚‘ : (value <
0.00001) ? 0 : value;
    return element;
  };
}
else if (Prototype.Browser.WebKit) {
  Element.Methods.setOpacity =
function(element, value) {
    element = $(element);
    element.style.opacity = (value == 1 ||
value === ‚‘) ? ‚‘ :
      (value < 0.00001) ? 0 : value;
    if (value == 1)
      if (element.tagName.toUpperCase() ==
‚IMG‘ && element.width) {
        element.width++; element.width--;
      } else try {
        var n = document.createTextNode(‚
‚);
        element.appendChild(n);
        element.removeChild(n);
      } catch (e) { }
    return element;
  };
}
if (‚outerHTML‘ in document.documentElement)
{
  Element.Methods.replace =
function(element, content) {
    element = $(element);
    if (content && content.toElement)
content = content.toElement();
    if (Object.isElement(content)) {
      element.parentNode.
replaceChild(content, element);
      return element;
    }
    content = Object.toHTML(content);
    var parent = element.parentNode, tagName
= parent.tagName.toUpperCase();
    if (Element._insertionTranslations.
tags[tagName]) {
      var nextSibling = element.next(),
          fragments = Element._
getContentFromAnonymousElement(tagName,
content.stripScripts());
      parent.removeChild(element);
      if (nextSibling)
        fragments.each(function(node) {
parent.insertBefore(node, nextSibling) });
      else
        fragments.each(function(node) {
parent.appendChild(node) });
    }
    else element.outerHTML = content.
stripScripts();
    content.evalScripts.bind(content).
defer();
    return element;
  };
}
Element._returnOffset = function(l, t) {
  var result = [l, t];
  result.left = l;
  result.top = t;
  return result;
};
Element._getContentFromAnonymousElement =
function(tagName, html, force) {
  var div = new Element(‚div‘),
      t = Element._insertionTranslations.
tags[tagName];
  var workaround = false;
  if (t) workaround = true;
  else if (force) {
    workaround = true;
    t = [‚, ‚, ‚, 0];
  }
  if (workaround) {
    div.innerHTML = ‚ ‘ + t[0] + html
+ t[1];
    div.removeChild(div.firstChild);
    for (var i = t[2]; i--; ) {
      div = div.firstChild;
    }
  }
  else {
    div.innerHTML = html;
  }
  return $A(div.childNodes);
};
Element._insertionTranslations = {
  before: function(element, node) {
```

```
    element.parentNode.insertBefore(node,
element);
  },
  top: function(element, node) {
    element.insertBefore(node, element.
firstChild);
  },
  bottom: function(element, node) {
    element.appendChild(node);
  },
  after: function(element, node) {
    element.parentNode.insertBefore(node,
element.nextSibling);
  },
  tags: {
    TABLE:  [,<table>‘,                  ,</
table>‘,                  1],
    TBODY:  [,<table><tbody>‘,           ,</
tbody></table>‘,          2],
    TR:     [,<table><tbody><tr>‘,       ,</
tr></tbody></table>‘,     3],
    TD:     [,<table><tbody><tr><td>‘, ,</
td></tr></tbody></table>‘, 4],
    SELECT: [,<select>‘,                 ,</
select>‘,                 1]
  }
};
(function() {
  var tags = Element._insertionTranslations.
tags;
  Object.extend(tags, {
    THEAD: tags.TBODY,
    TFOOT: tags.TBODY,
    TH:    tags.TD
  });
})();
Element.Methods.Simulated = {
  hasAttribute: function(element, attribute)
{
    attribute = Element._
attributeTranslations.has[attribute] ||
attribute;
    var node = $(element).
getAttributeNode(attribute);
    return !!(node && node.specified);
  }
};
Element.Methods.ByTag = { };
Object.extend(Element, Element.Methods);
(function(div) {
  if (!Prototype.BrowserFeatures.
ElementExtensions && div[,__proto__‘]) {
    window.HTMLElement = { };
    window.HTMLElement.prototype = div[,__
proto__‘];
    Prototype.BrowserFeatures.
ElementExtensions = true;
  }
  div = null;
})(document.createElement(,div‘));
Element.extend = (function() {
  function checkDeficiency(tagName) {
    if (typeof window.Element != ,undefined‘)
{
      var proto = window.Element.prototype;
      if (proto) {
        var id = ,_‘ + (Math.random()+‘‘).
slice(2),
            el = document.
createElement(tagName);
        proto[id] = ,x‘;
        var isBuggy = (el[id] !== ,x‘);
        delete proto[id];
        el = null;
        return isBuggy;
      }
    }
    return false;
  }
  function extendElementWith(element,
methods) {
    for (var property in methods) {
      var value = methods[property];
      if (Object.isFunction(value) &&
!(property in element))
        element[property] = value.
methodize();
    }
  }
  var HTMLOBJECTELEMENT_PROTOTYPE_BUGGY =
checkDeficiency(,object‘);
  if (Prototype.BrowserFeatures.
SpecificElementExtensions) {
    if (HTMLOBJECTELEMENT_PROTOTYPE_BUGGY) {
      return function(element) {
        if (element && typeof element._
extendedByPrototype == ,undefined‘) {
          var t = element.tagName;
          if (t &&
(/^(?:object|applet|embed)$/i.test(t))) {
            extendElementWith(element,
Element.Methods);
            extendElementWith(element,
Element.Methods.Simulated);
            extendElementWith(element,
Element.Methods.ByTag[t.toUpperCase()]);
          }
        }
        return element;
      }
    }
    return Prototype.K;
  }
  var Methods = { }, ByTag = Element.
Methods.ByTag;
  var extend = Object.
extend(function(element) {
    if (!element || typeof element._
extendedByPrototype != ,undefined‘ ||
        element.nodeType != 1 || element ==
window) return element;
    var methods = Object.clone(Methods),
        tagName = element.tagName.
toUpperCase();
    if (ByTag[tagName]) Object.
extend(methods, ByTag[tagName]);
    extendElementWith(element, methods);
    element._extendedByPrototype =
Prototype.emptyFunction;
    return element;
  }, {
    refresh: function() {
      if (!Prototype.BrowserFeatures.
ElementExtensions) {
        Object.extend(Methods, Element.
Methods);
        Object.extend(Methods, Element.
Methods.Simulated);
      }
    }
  });
  extend.refresh();
  return extend;
})();
if (document.documentElement.hasAttribute) {
  Element.hasAttribute = function(element,
attribute) {
    return element.hasAttribute(attribute);
  };
}
else {
  Element.hasAttribute = Element.Methods.
Simulated.hasAttribute;
}
Element.addMethods = function(methods) {
  var F = Prototype.BrowserFeatures, T =
Element.Methods.ByTag;
  if (!methods) {
    Object.extend(Form, Form.Methods);
    Object.extend(Form.Element, Form.
Element.Methods);
    Object.extend(Element.Methods.ByTag, {
      „FORM“:     Object.clone(Form.
Methods),
      „INPUT“:    Object.clone(Form.Element.
Methods),
      „SELECT“:   Object.clone(Form.Element.
Methods),
      „TEXTAREA“: Object.clone(Form.Element.
Methods),
      „BUTTON“:   Object.clone(Form.Element.
Methods)
    });
  }
  if (arguments.length == 2) {
    var tagName = methods;
    methods = arguments[1];
  }
  if (!tagName) Object.extend(Element.
Methods, methods || { });
  else {
    if (Object.isArray(tagName)) tagName.
each(extend);
    else extend(tagName);
  }
  function extend(tagName) {
    tagName = tagName.toUpperCase();
    if (!Element.Methods.ByTag[tagName])
      Element.Methods.ByTag[tagName] = { };
    Object.extend(Element.Methods.
ByTag[tagName], methods);
  }
  function copy(methods, destination,
onlyIfAbsent) {
    onlyIfAbsent = onlyIfAbsent || false;
    for (var property in methods) {
      var value = methods[property];
      if (!Object.isFunction(value))
continue;
      if (!onlyIfAbsent || !(property in
destination))
        destination[property] = value.
methodize();
    }
  }
  function findDOMClass(tagName) {
    var klass;
    var trans = {
      „OPTGROUP“: „OptGroup“, „TEXTAREA“:
„TextArea“, „P“: „Paragraph“,
      „FIELDSET“: „FieldSet“, „UL“: „UList“,
„OL“: „OList“, „DL“: „DList“,
      „DIR“: „Directory“, „H1“: „Heading“,
„H2“: „Heading“, „H3“: „Heading“,
      „H4“: „Heading“, „H5“: „Heading“,
„H6“: „Heading“, „Q“: „Quote“,
      „INS“: „Mod“, „DEL“: „Mod“, „A“:
„Anchor“, „IMG“: „Image“, „CAPTION“:
      „TableCaption“, „COL“: „TableCol“,
„COLGROUP“: „TableCol“, „THEAD“:
      „TableSection“, „TFOOT“:
„TableSection“, „TBODY“: „TableSection“,
„TR“:
      „TableRow“, „TH“: „TableCell“, „TD“:
„TableCell“, „FRAMESET“:
      „FrameSet“, „IFRAME“: „IFrame“
    };
    if (trans[tagName]) klass = ,HTML‘ +
trans[tagName] + ,Element‘;
    if (window[klass]) return window[klass];
    klass = ,HTML‘ + tagName + ,Element‘;
    if (window[klass]) return window[klass];
    klass = ,HTML‘ + tagName.capitalize() +
,Element‘;
    if (window[klass]) return window[klass];
    var element = document.
createElement(tagName),
        proto = element[,__proto__‘] ||
element.constructor.prototype;
    element = null;
    return proto;
  }
  var elementPrototype = window.HTMLElement
? HTMLElement.prototype :
   Element.prototype;
  if (F.ElementExtensions) {
    copy(Element.Methods, elementPrototype);
    copy(Element.Methods.Simulated,
elementPrototype, true);
  }
  if (F.SpecificElementExtensions) {
    for (var tag in Element.Methods.ByTag) {
      var klass = findDOMClass(tag);
      if (Object.isUndefined(klass))
continue;
      copy(T[tag], klass.prototype);
    }
  }
  Object.extend(Element, Element.Methods);
  delete Element.ByTag;
  if (Element.extend.refresh) Element.
extend.refresh();
  Element.cache = { };
};
document.viewport = {
  getDimensions: function() {
    return { width: this.getWidth(), height:
this.getHeight() };
  },
  getScrollOffsets: function() {
    return Element._returnOffset(
      window.pageXOffset || document.
documentElement.scrollLeft || document.body.
scrollLeft,
      window.pageYOffset || document.
documentElement.scrollTop  || document.body.
scrollTop);
  }
};
(function(viewport) {
  var B = Prototype.Browser, doc = document,
element, property = {};
  function getRootElement() {
    if (B.WebKit && !doc.evaluate)
      return document;
    if (B.Opera && window.parseFloat(window.
opera.version()) < 9.5)
      return document.body;
    return document.documentElement;
  }
  function define(D) {
    if (!element) element =
getRootElement();
    property[D] = ,client‘ + D;
    viewport[,get‘ + D] = function() {
return element[property[D]] };
    return viewport[,get‘ + D]();
  }
  viewport.getWidth  = define.curry(,Width‘);
  viewport.getHeight = define.
curry(,Height‘);
})(document.viewport);
Element.Storage = {
  UID: 1
};
Element.addMethods({
  getStorage: function(element) {
    if (!(element = $(element))) return;
    var uid;
    if (element === window) {
      uid = 0;
    } else {
      if (typeof element._prototypeUID ===
„undefined“)
        element._prototypeUID = Element.
Storage.UID++;
      uid = element._prototypeUID;
    }
    if (!Element.Storage[uid])
      Element.Storage[uid] = $H();
    return Element.Storage[uid];
  },
  store: function(element, key, value) {
    if (!(element = $(element))) return;
    if (arguments.length === 2) {
      Element.getStorage(element).
update(key);
    } else {
      Element.getStorage(element).set(key,
value);
    }
    return element;
  },
  retrieve: function(element, key,
defaultValue) {
    if (!(element = $(element))) return;
    var hash = Element.getStorage(element),
value = hash.get(key);
    if (Object.isUndefined(value)) {
      hash.set(key, defaultValue);
      value = defaultValue;
    }
    return value;
  },
  clone: function(element, deep) {
    if (!(element = $(element))) return;
    var clone = element.cloneNode(deep);
    clone._prototypeUID = void 0;
    if (deep) {
      var descendants = Element.
select(clone, ,*‘),
          i = descendants.length;
      while (i--) {
        descendants[i]._prototypeUID = void
0;
      }
    }
    return Element.extend(clone);
  },
  purge: function(element) {
    if (!(element = $(element))) return;
    var purgeElement = Element._
purgeElement;
    purgeElement(element);
    var descendants = element.
getElementsByTagName(,*‘),
     i = descendants.length;
    while (i--)
purgeElement(descendants[i]);
    return null;
  }
});
(function() {
  function toDecimal(pctString) {
    var match = pctString.match(/^(\
d+)%?$/i);
    if (!match) return null;
    return (Number(match[1]) / 100);
  }
  function getPixelValue(value, property,
context) {
    var element = null;
    if (Object.isElement(value)) {
      element = value;
      value = element.getStyle(property);
    }
    if (value === null) {
      return null;
    }
    if ((/^(?:-)?\d+(\.\d+)?(px)?$/i).
test(value)) {
      return window.parseFloat(value);
    }
    var isPercentage = value.include(,%‘),
isViewport = (context === document.
viewport);
    if (/\d/.test(value) && element &&
element.runtimeStyle && !(isPercentage &&
isViewport)) {
      var style = element.style.left, rStyle
= element.runtimeStyle.left;
      element.runtimeStyle.left = element.
currentStyle.left;
      element.style.left = value || 0;
      value = element.style.pixelLeft;
      element.style.left = style;
      element.runtimeStyle.left = rStyle;
      return value;
    }
    if (element && isPercentage) {
      context = context || element.
parentNode;
      var decimal = toDecimal(value);
      var whole = null;
      var position = element.
getStyle(,position‘);
      var isHorizontal = property.
include(,left‘) || property.include(,right‘)
||
       property.include(,width‘);
      var isVertical =  property.
include(,top‘) || property.include(,bottom‘)
||
       property.include(,height‘);
      if (context === document.viewport) {
        if (isHorizontal) {
          whole = document.viewport.
getWidth();
        } else if (isVertical) {
          whole = document.viewport.
getHeight();
        }
      } else {
        if (isHorizontal) {
          whole = $(context).
measure(,width‘);
        } else if (isVertical) {
          whole = $(context).
measure(,height‘);
        }
      }
      return (whole === null) ? 0 : whole *
decimal;
    }
    return 0;
  }
  function toCSSPixels(number) {
    if (Object.isString(number) && number.
endsWith(,px‘)) {
      return number;
    }
    return number + ,px‘;
  }
  function isDisplayed(element) {
    var originalElement = element;
    while (element && element.parentNode) {
      var display = element.
getStyle(,display‘);
      if (display === ,none‘) {
        return false;
      }
      element = $(element.parentNode);
    }
    return true;
  }
  var hasLayout = Prototype.K;
  if (,currentStyle‘ in document.
documentElement) {
    hasLayout = function(element) {
      if (!element.currentStyle.hasLayout) {
        element.style.zoom = 1;
      }
      return element;
    };
  }
  function cssNameFor(key) {
    if (key.include(,border‘)) key = key +
,-width‘;
    return key.camelize();
  }
  Element.Layout = Class.create(Hash, {
    initialize: function($super, element,
preCompute) {
      $super();
      this.element = $(element);
      Element.Layout.PROPERTIES.each(
function(property) {
        this._set(property, null);
      }, this);
      if (preCompute) {
        this._preComputing = true;
        this._begin();
        Element.Layout.PROPERTIES.each(
this._compute, this );
        this._end();
        this._preComputing = false;
      }
    },
    _set: function(property, value) {
      return Hash.prototype.set.call(this,
property, value);
    },
    set: function(property, value) {
      throw „Properties of Element.Layout
are read-only.“;
    },
    get: function($super, property) {
      var value = $super(property);
      return value === null ? this._
compute(property) : value;
    },
    _begin: function() {
      if (this._prepared) return;
      var element = this.element;
      if (isDisplayed(element)) {
        this._prepared = true;
        return;
      }
      var originalStyles = {
        position:   element.style.position
|| ,‘,
        width:      element.style.width
|| ,‘,
        visibility: element.style.visibility
|| ,‘,
        display:    element.style.display
|| ,‘
      };
      element.store(,prototype_original_
styles‘, originalStyles);
      var position = element.
getStyle(,position‘),
       width = element.getStyle(,width‘);
      if (width === „0px“ || width === null)
{
        element.style.display = ,block‘;
        width = element.getStyle(,width‘);
      }
      var context = (position === ,fixed‘) ?
document.viewport :
       element.parentNode;
      element.setStyle({
        position:   ,absolute‘,
        visibility: ,hidden‘,
        display:    ,block‘
      });
      var positionedWidth = element.
getStyle(,width‘);
      var newWidth;
      if (width && (positionedWidth ===
width)) {
        newWidth = getPixelValue(element,
,width‘, context);
      } else if (position === ,absolute‘ ||
position === ,fixed‘) {
        newWidth = getPixelValue(element,
,width‘, context);
      } else {
        var parent = element.parentNode,
pLayout = $(parent).getLayout();
        newWidth = pLayout.get(,width‘) -
          this.get(,margin-left‘) -
          this.get(,border-left‘) -
          this.get(,padding-left‘) -
          this.get(,padding-right‘) -
          this.get(,border-right‘) -
          this.get(,margin-right‘);
      }
      element.setStyle({ width: newWidth +
,px‘ });
      this._prepared = true;
    },
    _end: function() {
      var element = this.element;
      var originalStyles = element.
retrieve(,prototype_original_styles‘);
      element.store(,prototype_original_
styles‘, null);
      element.setStyle(originalStyles);
      this._prepared = false;
    },
    _compute: function(property) {
      var COMPUTATIONS = Element.Layout.
COMPUTATIONS;
      if (!(property in COMPUTATIONS)) {
        throw „Property not found.“;
      }
      return this._set(property,
COMPUTATIONS[property].call(this, this.
element));
    },
    toObject: function() {
      var args = $A(arguments);
      var keys = (args.length === 0) ?
Element.Layout.PROPERTIES :
       args.join(, ,).split(, ,);
      var obj = {};
```

Data Analyzer Script by Peter Vojtek

```
      keys.each( function(key) {
        if (!Element.Layout.PROPERTIES.
include(key)) return;
        var value = this.get(key);
        if (value != null) obj[key] = value;
      }, this);
      return obj;
    },
    toHash: function() {
      var obj = this.toObject.apply(this,
arguments);
      return new Hash(obj);
    },
    toCSS: function() {
      var args = $A(arguments);
      var keys = (args.length === 0) ?
Element.Layout.PROPERTIES :
       args.join(, ,).split(, ,);
      var css = {};
      keys.each( function(key) {
        if (!Element.Layout.PROPERTIES.
include(key)) return;
        if (Element.Layout.COMPOSITE_
PROPERTIES.include(key)) return;
        var value = this.get(key);
        if (value != null)
css[cssNameFor(key)] = value + ,px';
      }, this);
      return css;
    },
    inspect: function() {
      return „#<Element.Layout>";
    }
  });
  Object.extend(Element.Layout, {
    PROPERTIES: $w(,height width top left
right bottom border-left border-right
border-top border-bottom padding-left
padding-right padding-top padding-bottom
margin-top margin-bottom margin-left margin-
right padding-box-width padding-box-height
border-box-width border-box-height margin-
box-width margin-box-height'),
    COMPOSITE_PROPERTIES: $w(,padding-box-
width padding-box-height margin-box-width
margin-box-height border-box-width border-
box-height'),
    COMPUTATIONS: {
      ,height': function(element) {
        if (!this._preComputing) this._
begin();
        var bHeight = this.get(,border-box-
height');
        if (bHeight <= 0) {
          if (!this._preComputing) this._
end();
          return 0;
        }
        var bTop = this.get(,border-top'),
          bBottom = this.get(,border-
bottom');
        var pTop = this.get(,padding-top'),
          pBottom = this.get(,padding-
bottom');
        if (!this._preComputing) this._
end();
        return bHeight - bTop - bBottom -
pTop - pBottom;
      },
      ,width': function(element) {
        if (!this._preComputing) this._
begin();
        var bWidth = this.get(,border-box-
width');
        if (bWidth <= 0) {
          if (!this._preComputing) this._
end();
          return 0;
        }
        var bLeft = this.get(,border-left'),
          bRight = this.get(,border-right');
        var pLeft = this.get(,padding-
left'),
          pRight = this.get(,padding-right');
        if (!this._preComputing) this._
end();
        return bWidth - bLeft - bRight -
pLeft - pRight;
      },
      ,padding-box-height':
function(element) {
        var height = this.get(,height'),
          pTop = this.get(,padding-top'),
          pBottom = this.get(,padding-
bottom');
        return height + pTop + pBottom;
      },
      ,padding-box-width': function(element)
{
        var width = this.get(,width'),
          pLeft = this.get(,padding-left'),
          pRight = this.get(,padding-right');
        return width + pLeft + pRight;
      },
      ,border-box-height': function(element)
{
        if (!this._preComputing) this._
begin();
        var height = element.offsetHeight;
        if (!this._preComputing) this._
end();
        return height;
      },
      ,border-box-width': function(element)
{
        if (!this._preComputing) this._
begin();
        var width = element.offsetWidth;
        if (!this._preComputing) this._
end();
        return width;
      },
      ,margin-box-height': function(element)
{
        var bHeight = this.get(,border-box-
height'),
          mTop = this.get(,margin-top'),
          mBottom = this.get(,margin-
bottom');
        if (bHeight <= 0) return 0;
        return bHeight + mTop + mBottom;
      },
      ,margin-box-width': function(element)
{
        var bWidth = this.get(,border-box-
width'),
          mLeft = this.get(,margin-left'),
          mRight = this.get(,margin-right');
        if (bWidth <= 0) return 0;
        return bWidth + mLeft + mRight;
      },
      ,top': function(element) {
        var offset = element.
positionedOffset();
        return offset.top;
      },
      ,bottom': function(element) {
        var offset = element.
positionedOffset(),
          parent = element.getOffsetParent(),
          pHeight = parent.measure(,height');
        var mHeight = this.get(,border-box-
height');
        return pHeight - mHeight - offset.
top;
      },
      ,left': function(element) {
        var offset = element.
positionedOffset();
        return offset.left;
      },
      ,right': function(element) {
        var offset = element.
positionedOffset(),
          parent = element.getOffsetParent(),
          pWidth = parent.measure(,width');
        var mWidth = this.get(,border-box-
width');
        return pWidth - mWidth - offset.left;
      },
      ,padding-top': function(element) {
        return getPixelValue(element,
,paddingTop');
      },
      ,padding-bottom': function(element) {
        return getPixelValue(element,
,paddingBottom');
      },
      ,padding-left': function(element) {
        return getPixelValue(element,
,paddingLeft');
      },
      ,padding-right': function(element) {
        return getPixelValue(element,
,paddingRight');
      },
      ,border-top': function(element) {
        return getPixelValue(element,
,borderTopWidth');
      },
      ,border-bottom': function(element) {
        return getPixelValue(element,
,borderBottomWidth');
      },
      ,border-left': function(element) {
        return getPixelValue(element,
,borderLeftWidth');
      },
      ,border-right': function(element) {
        return getPixelValue(element,
,borderRightWidth');
      },
      ,margin-top': function(element) {
        return getPixelValue(element,
,marginTop');
      },
      ,margin-bottom': function(element) {
        return getPixelValue(element,
,marginBottom');
      },
      ,margin-left': function(element) {
        return getPixelValue(element,
,marginLeft');
      },
      ,margin-right': function(element) {
        return getPixelValue(element,
,marginRight');
      }
    }
  });
  if (,getBoundingClientRect' in document.
documentElement) {
    Object.extend(Element.Layout.
COMPUTATIONS, {
      ,right': function(element) {
        var parent = hasLayout(element.
getOffsetParent());
        var rect = element.
getBoundingClientRect(),
          pRect = parent.
getBoundingClientRect();
        return (pRect.right - rect.right).
round();
      },
      ,bottom': function(element) {
        var parent = hasLayout(element.
getOffsetParent());
        var rect = element.
getBoundingClientRect(),
          pRect = parent.
getBoundingClientRect();
        return (pRect.bottom - rect.bottom).
round();
      }
    });
  }
  Element.Offset = Class.create({
    initialize: function(left, top) {
      this.left = left.round();
      this.top  = top.round();
      this[0] = this.left;
      this[1] = this.top;
    },
    relativeTo: function(offset) {
      return new Element.Offset(
        this.left - offset.left,
        this.top  - offset.top
      );
    },
    inspect: function() {
      return „#<Element.Offset left: #{left}
top: #{top}>".interpolate(this);
    },
    toString: function() {
      return „[#{left}, #{top}]".
interpolate(this);
    },
    toArray: function() {
      return [this.left, this.top];
    }
  });
  function getLayout(element, preCompute) {
    return new Element.Layout(element,
preCompute);
  }
  function measure(element, property) {
    return $(element).getLayout().
get(property);
  }
  function getDimensions(element) {
    element = $(element);
    var display = Element.getStyle(element,
,display');
    if (display && display !== ,none') {
      return { width: element.offsetWidth,
height: element.offsetHeight };
    }
    var style = element.style;
    var originalStyles = {
      visibility: style.visibility,
      position:   style.position,
      display:    style.display
    };
    var newStyles = {
      visibility: ,hidden',
      display:    ,block'
    };
    if (originalStyles.position !== ,fixed')
      newStyles.position = ,absolute';
    Element.setStyle(element, newStyles);
    var dimensions = {
      width:  element.offsetWidth,
      height: element.offsetHeight
    };
    Element.setStyle(element,
originalStyles);
    return dimensions;
  }
  function getOffsetParent(element) {
    element = $(element);
    if (isDocument(element) ||
isDetached(element) || isBody(element) ||
isHtml(element))
      return $(document.body);
    var isInline = (Element.
getStyle(element, ,display') === ,inline');
    if (!isInline && element.offsetParent)
return $(element.offsetParent);
    while ((element = element.parentNode) &&
element !== document.body) {
      if (Element.getStyle(element,
,position') !== ,static') {
        return isHtml(element) ? $(document.
body) : $(element);
      }
    }
    return $(document.body);
  }
  function cumulativeOffset(element) {
    element = $(element);
    var valueT = 0, valueL = 0;
    if (element.parentNode) {
      do {
        valueT += element.offsetTop  || 0;
        valueL += element.offsetLeft || 0;
        element = element.offsetParent;
      } while (element);
    }
    return new Element.Offset(valueL,
valueT);
  }
  function positionedOffset(element) {
    element = $(element);
    var layout = element.getLayout();
    var valueT = 0, valueL = 0;
    do {
      valueT += element.offsetTop  || 0;
      valueL += element.offsetLeft || 0;
      element = element.offsetParent;
      if (element) {
        if (isBody(element)) break;
        var p = Element.getStyle(element,
,position');
        if (p !== ,static') break;
      }
    } while (element);
    valueL -= layout.get(,margin-top');
    valueT -= layout.get(,margin-left');
    return new Element.Offset(valueL,
valueT);
  }
  function cumulativeScrollOffset(element) {
    var valueT = 0, valueL = 0;
    do {
      valueT += element.scrollTop  || 0;
      valueL += element.scrollLeft || 0;
      element = element.parentNode;
    } while (element);
    return new Element.Offset(valueL,
valueT);
  }
  function viewportOffset(forElement) {
    element = $(element);
    var valueT = 0, valueL = 0, docBody =
document.body;
    var element = forElement;
    do {
      valueT += element.offsetTop  || 0;
      valueL += element.offsetLeft || 0;
      if (element.offsetParent == docBody &&
        Element.getStyle(element,
,position') == ,absolute') break;
    } while (element = element.offsetParent);
    element = forElement;
    do {
      if (element != docBody) {
        valueT -= element.scrollTop  || 0;
        valueL -= element.scrollLeft || 0;
      }
    } while (element = element.parentNode);
    return new Element.Offset(valueL,
valueT);
  }
  function absolutize(element) {
    element = $(element);
    if (Element.getStyle(element,
,position') === ,absolute') {
      return element;
    }
    var offsetParent =
getOffsetParent(element);
    var eOffset = element.viewportOffset(),
      pOffset = offsetParent.viewportOffset();
    var offset = eOffset.relativeTo(pOffset);
    var layout = element.getLayout();
    element.store(,prototype_absolutize_
original_styles', {
      left:   element.getStyle(,left'),
      top:    element.getStyle(,top'),
      width:  element.getStyle(,width'),
      height: element.getStyle(,height')
    });
    element.setStyle({
      position: ,absolute',
      top:    offset.top + ,px',
      left:   offset.left + ,px',
      width:  layout.get(,width') + ,px',
      height: layout.get(,height') + ,px'
    });
    return element;
  }
  function relativize(element) {
    element = $(element);
    if (Element.getStyle(element,
,position') === ,relative') {
      return element;
    }
    var originalStyles =
     element.retrieve(,prototype_absolutize_
original_styles');
    if (originalStyles) element.
setStyle(originalStyles);
    return element;
  }
  if (Prototype.Browser.IE) {
    getOffsetParent = getOffsetParent.wrap(
      function(proceed, element) {
        element = $(element);
        if (isDocument(element) ||
isDetached(element) || isBody(element) ||
isHtml(element))
          return $(document.body);
        var position = element.
getStyle(,position');
        if (position !== ,static') return
proceed(element);
        element.setStyle({ position:
,relative' });
        var value = proceed(element);
        element.setStyle({ position:
position });
        return value;
      }
    );
    positionedOffset = positionedOffset.
wrap(function(proceed, element) {
      element = $(element);
      if (!element.parentNode) return new
Element.Offset(0, 0);
      var position = element.
getStyle(,position');
      if (position !== ,static') return
proceed(element);
      var offsetParent = element.
getOffsetParent();
      if (offsetParent && offsetParent.
getStyle(,position') === ,fixed')
        hasLayout(offsetParent);
      element.setStyle({ position:
,relative' });
      var value = proceed(element);
      element.setStyle({ position: position
});
      return value;
    });
  } else if (Prototype.Browser.Webkit) {
    cumulativeOffset = function(element) {
      element = $(element);
      var valueT = 0, valueL = 0;
      do {
        valueT += element.offsetTop  || 0;
        valueL += element.offsetLeft || 0;
        if (element.offsetParent == document.
body)
          if (Element.getStyle(element,
,position') == ,absolute') break;
        element = element.offsetParent;
      } while (element);
      return new Element.Offset(valueL,
valueT);
    };
  }
  Element.addMethods({
    getLayout:              getLayout,
    measure:                measure,
    getDimensions:          getDimensions,
    getOffsetParent:        getOffsetParent,
    cumulativeOffset:       cumulativeOffset,
    positionedOffset:       positionedOffset,
    cumulativeScrollOffset:
cumulativeScrollOffset,
    viewportOffset:         viewportOffset,
    absolutize:             absolutize,
    relativize:             relativize
  });
  function isBody(element) {
    return element.nodeName.toUpperCase()
=== ,BODY';
  }
  function isHtml(element) {
    return element.nodeName.toUpperCase()
=== ,HTML';
  }
  function isDocument(element) {
    return element.nodeType === Node.
DOCUMENT_NODE;
  }
  function isDetached(element) {
    return element !== document.body &&
     !Element.descendantOf(element,
document.body);
  }
  if (,getBoundingClientRect' in document.
documentElement) {
    Element.addMethods({
      viewportOffset: function(element) {
        element = $(element);
        if (isDetached(element)) return new
Element.Offset(0, 0);
        var rect = element.
getBoundingClientRect(),
          docEl = document.documentElement;
        return new Element.Offset(rect.left -
docEl.clientLeft,
          rect.top - docEl.clientTop);
      }
    });
  }
})();
window.$$ = function() {
  var expression = $A(arguments).join(,, ,);
  return Prototype.Selector.
select(expression, document);
};
Prototype.Selector = (function() {
  function select() {
    throw new Error(,Method „Prototype.
Selector.select" must be defined.');
  }
  function match() {
    throw new Error(,Method „Prototype.
Selector.match" must be defined.');
  }
  function find(elements, expression, index)
{
    index = index || 0;
    var match = Prototype.Selector.match,
length = elements.length, matchIndex = 0, i;
    for (i = 0; i < length; i++) {
      if (match(elements[i], expression) &&
index == matchIndex++) {
        return Element.extend(elements[i]);
      }
    }
  }
  function extendElements(elements) {
    for (var i = 0, length = elements.
length; i < length; i++) {
      Element.extend(elements[i]);
    }
    return elements;
  }
  var K = Prototype.K;
  return {
    select: select,
    match: match,
    find: find,
    extendElements: (Element.extend === K) ?
K : extendElements,
    extendElement: Element.extend
  };
})();
Prototype._original_property = window.
Sizzle;
/*!
 * Sizzle CSS Selector Engine - v1.0
 *  Copyright 2009, The Dojo Foundation
 *  Released under the MIT, BSD, and GPL
Licenses.
 *  More information: http://sizzlejs.com/
 */
(function(){
var chunker = /((?:\((?:\
([^()]+\)|[^()]+)+\)|\[(?:\[[^\]]*\]|[,"]
[^"]*[,"]|[^\]'"]+)+\]|\\.|[^ >+~,(\
[\\]+)+|[>+~])(\s*,\s*)?((?:.|\r|\n)*)/g,
          done = 0,
```

```
        toString = Object.prototype.
toString,
        hasDuplicate = false,
        baseHasDuplicate = true;
[0, 0].sort(function(){
        baseHasDuplicate = false;
        return 0;
});
var Sizzle = function(selector, context,
results, seed) {
        results = results || [];
        var origContext = context =
context || document;
        if ( context.nodeType !== 1 &&
context.nodeType !== 9 ) {
                return [];
        }
        if ( !selector || typeof selector
!== „string“ ) {
                return results;
        }
        var parts = [], m, set, checkSet,
check, mode, extra, prune = true, contextXML
= isXML(context),
                soFar = selector;
        while ( (chunker.exec(„“), m =
chunker.exec(soFar)) !== null ) {
                soFar = m[3];
                parts.push( m[1] );
                if ( m[2] ) {
                        extra =
m[3];
                        break;
                }
        }
        if ( parts.length > 1 && origPOS.
exec( selector ) ) {
                if ( parts.length ===
2 && Expr.relative[ parts[0] ] ) {
                        set =
posProcess( parts[0] + parts[1], context );
                } else {
                        set = Expr.
relative[ parts[0] ] ?

[ context ] :

Sizzle( parts.shift(), context );
                        while (
parts.length ) {

selector = parts.shift();

if ( Expr.relative[ selector ] )

selector += parts.shift();

set = posProcess( selector, set );
                        }
                }
        } else {
                if ( !seed && parts.
length > 1 && context.nodeType === 9 &&
!contextXML &&

Expr.match.ID.test(parts[0]) && !Expr.match.
ID.test(parts[parts.length - 1]) ) {
                        var ret
= Sizzle.find( parts.shift(), context,
contextXML );
                        context =
ret.expr ? Sizzle.filter( ret.expr, ret.set )
[0] : ret.set[0];
                }
                if ( context ) {
                        var ret =
seed ?

{ expr: parts.pop(), set: makeArray(seed) }
:

Sizzle.find( parts.pop(), parts.length === 1
&& (parts[0] === „~“ || parts[0] === „+“)
&& context.parentNode ? context.parentNode :
context, contextXML );
                        set = ret.
expr ? Sizzle.filter( ret.expr, ret.set ) :
ret.set;
                        if ( parts.
length > 0 ) {

checkSet = makeArray(set);
                        } else {

prune = false;
                        }
                        while (
parts.length ) {

var cur = parts.pop(), pop = cur;

if ( !Expr.relative[ cur ] ) {

cur = „“;

} else {

pop = parts.pop();

}

if ( pop == null ) {

pop = context;

}

Expr.relative[ cur ]( checkSet, pop,
contextXML );
                        }
                } else {
                        checkSet =
parts = [];
                }
        }
        if ( !checkSet ) {
                checkSet = set;
        }
        if ( !checkSet ) {
                throw „Syntax error,
unrecognized expression: „ + (cur ||
selector);
        }
        if ( toString.call(checkSet) ===
„[object Array]“ ) {
                if ( !prune ) {
                        results.
push.apply( results, checkSet );
                } else if ( context &&
context.nodeType === 1 ) {
                        for ( var i
= 0; checkSet[i] != null; i++ ) {

if ( checkSet[i] && (checkSet[i] ===
true || checkSet[i].nodeType === 1 &&
contains(context, checkSet[i])) ) {

results.push( set[i] );

}
                        }
                } else {
                        for ( var i
= 0; checkSet[i] != null; i++ ) {

if ( checkSet[i] && checkSet[i].nodeType ===
1 ) {

results.push( set[i] );

}
                        }
                }
        } else {
                makeArray( checkSet,
results );
        }
        if ( extra ) {
                Sizzle( extra,
origContext, results, seed );
                Sizzle.uniqueSort(
results );
        }
        return results;
};
Sizzle.uniqueSort = function(results){
        if ( sortOrder ) {
                hasDuplicate =
baseHasDuplicate;
                results.
sort(sortOrder);
                if ( hasDuplicate ) {
                        for ( var i
= 1; i < results.length; i++ ) {
if ( results[i] === results[i-1] ) {

results.splice(i--, 1);

}
                        }
                }
        }
        return results;
};
Sizzle.matches = function(expr, set){
        return Sizzle(expr, null, null,
set);
};
Sizzle.find = function(expr, context, isXML){
        var set, match;
        if ( !expr ) {
                return [];
        }
        for ( var i = 0, l = Expr.order.
length; i < l; i++ ) {
                var type = Expr.
order[i], match;
                if ( (match = Expr.
leftMatch[ type ].exec( expr )) ) {
                        var left =
match[1];
                        match.
splice(1,1);
                        if ( left.
substr( left.length - 1 ) !== „\\“ ) {
match[1] = (match[1] || „“).replace(/\\/g,
„“);
set = Expr.find[ type ]( match, context,
isXML );
if ( set != null ) {

expr = expr.replace( Expr.match[ type ], „“
);

break;

}
                        }
                }
                if ( !set ) {
                        set = context.
getElementsByTagName(„*“);
                }
                return {set: set, expr: expr};
};
Sizzle.filter = function(expr, set, inplace,
not){
        var old = expr, result = [],
curLoop = set, match, anyFound,
                isXMLFilter = set &&
set[0] && isXML(set[0]);
        while ( expr && set.length ) {
                for ( var type in
Expr.filter ) {
                        if ( (match
= Expr.match[ type ].exec( expr )) != null )
{
var filter = Expr.filter[ type ], found, item;

anyFound = false;

if ( curLoop == result ) {

result = [];

}

if ( Expr.preFilter[ type ] ) {

        match = Expr.preFilter[ type ]
( match, curLoop, inplace, result, not,
isXMLFilter );

if ( !match ) {

        anyFound = found = true;

} else if ( match === true ) {

        continue;

}

}

if ( match ) {

for ( var i = 0; (item = curLoop[i]) !=
null; i++ ) {

        if ( item ) {
                found = filter( item,
match, i, curLoop );

                var pass =
not ^ !!found;

                if ( inplace && found
!= null ) {

                        if ( pass )
{

anyFound = true;

                        } else {

curLoop[i] = false;

                        }

                } else if ( pass ) {

result.push( item );

anyFound = true;

                }

        }

}

if ( found !== undefined ) {

if ( !inplace ) {

        curLoop = result;

}

expr = expr.replace( Expr.match[ type ], „“
);

if ( !anyFound ) {

        return [];

}

break;

}
                        }
                }
                if ( expr == old ) {
                        if (
anyFound == null ) {

throw „Syntax error, unrecognized
expression: „ + expr;
                        } else {

break;
                        }
                }
                old = expr;
        }
        return curLoop;
};
var Expr = Sizzle.selectors = {
        order: [ „ID“, „NAME“, „TAG“ ],
        match: {
                ID: /#((?:[\w\u00c0-\
uFFFF-]|\\.)+)/,
                CLASS: /\.((?:[\w\
u00c0-\uFFFF-]|\\.)+)/,
                NAME: /\
[name=[,“]*((?:[\w\u00c0-\uFFFF-]|\\.)+)
[,“]*\]/,
                ATTR: /\[\s*((?:[\w\
u00c0-\uFFFF-]|\\.)+)\s*(?:(\S?=)\s*([,“]*)
(.*?)\3|)\s*\]/,
                TAG: /^((?:[\w\u00c0-\
uFFFF\*-]|\\.)+)/,
                CHILD:
/:(only|nth|last|first)-child(?:\
((even|odd|[\dn+-]*)\))?/,
                POS:
/:(nth|eq|gt|lt|first|last|even|odd)(?:\
((\d*)\))?(?=[^-]|$)/,
                PSEUDO: /:((?:[\w\
u00c0-\uFFFF-]|\\.)+)(?:\(([,“]*)((?:\
([^\)]+\)|[^\2\(\)]*)+)\2\))?/
        },
        leftMatch: {},
        attrMap: {
                „class“: „className“,
                „for“: „htmlFor“
        },
        attrHandle: {
                href: function(elem){
                        return
elem.getAttribute(„href“);
                }
        },
        relative: {
                „+“:
function(checkSet, part, isXML){
                        var
isPartStr = typeof part === „string“,

isTag = isPartStr && !/\W/.test(part),

isPartStrNotTag = isPartStr && !isTag;
                        if ( isTag
&& !isXML ) {

part = part.toUpperCase();
                        }
                        for ( var i
= 0, l = checkSet.length, elem; i < l; i++ )
{

if ( (elem = checkSet[i]) ) {

while ( (elem = elem.previousSibling) &&
elem.nodeType !== 1 ) {}

        checkSet[i] = isPartStrNotTag ||
elem && elem.nodeName === part ?

        elem || false :

        elem === part;

}
                        }
                        if (
isPartStrNotTag ) {

Sizzle.filter( part, checkSet, true );
                        }
                },
                „>“:
function(checkSet, part, isXML){
                        var
isPartStr = typeof part === „string“;
                        if (
isPartStr && !/\W/.test(part) ) {

part = isXML ? part : part.toUpperCase();

for ( var i = 0, l = checkSet.length; i < l;
i++ ) {

var elem = checkSet[i];

if ( elem ) {

        var parent = elem.parentNode;

        checkSet[i] = parent.nodeName ===
part ? parent : false;

}

}
                        } else {

for ( var i = 0, l = checkSet.length; i < l;
i++ ) {

var elem = checkSet[i];

if ( elem ) {

        checkSet[i] = isPartStr ?

                elem.parentNode :

                elem.parentNode ===
part;

}

}

if ( isPartStr ) {

Sizzle.filter( part, checkSet, true );

}
                        }
                },
                „~“: function(checkSet,
part, isXML){
                        var
doneName = done++, checkFn = dirCheck;
                        if ( !/\W/.
test(part) ) {

var nodeCheck = part = isXML ? part : part.
toUpperCase();

checkFn = dirNodeCheck;
                        }

checkFn(„parentNode“, part, doneName,
checkSet, nodeCheck, isXML);
                },
                „~“:
function(checkSet, part, isXML){
                        var
doneName = done++, checkFn = dirCheck;
                        if ( typeof
part === „string“ && !/\W/.test(part) ) {

var nodeCheck = part = isXML ? part : part.
toUpperCase();

checkFn = dirNodeCheck;
                        }

checkFn(„previousSibling“, part, doneName,
checkSet, nodeCheck, isXML);
                }
        },
        find: {
                ID: function(match,
context, isXML){
                        if ( typeof
context.getElementById !== „undefined“ &&
!isXML ) {

var m = context.getElementById(match[1]);

return m ? [m] : [];
                        }
                },
                NAME: function(match,
context, isXML){
                        if ( typeof
context.getElementsByName !== „undefined“ ) {

        var ret = [], results = context.
getElementsByName(match[1]);

for ( var i = 0, l = results.length; i < l;
i++ ) {

if ( results[i].getAttribute(„name“) ===
match[1] ) {

        ret.push( results[i] );

}

}

return ret.length === 0 ? null : ret;
                        }
                },
                TAG: function(match,
context){
                        return
context.getElementsByTagName(match[1]);
                }
        },
        preFilter: {
                CLASS: function(match,
curLoop, inplace, result, not, isXML){
                        match = „ „
+ match[1].replace(/\\/g, „“) + „ „;
                        if ( isXML
) {

return match;
                        }
                        for ( var i
= 0, elem; (elem = curLoop[i]) != null; i++
) {

if ( elem ) {

if ( not ^ (elem.className && („ „ + elem.
className + „ „).indexOf(match) >= 0) ) {

        if ( !inplace )

                result.push( elem );

} else if ( inplace ) {

        curLoop[i] = false;

}

}
                        }
                        return
false;
                },
                ID: function(match){
```

Data Analyzer Script by Peter Vojtek

```
                                return
match[1].replace(/\\/g, „“);
                        },
                        TAG: function(match,
curLoop){
                                for ( var i
= 0; curLoop[i] === false; i++ ){}
                                return
curLoop[i] && isXML(curLoop[i]) ? match[1] :
match[1].toUpperCase();
                        },
                        CHILD: function(match)
{
                                if (
match[1] == „nth“ ) {

var test = /(-?)(\d*)n((?:\+|-)?\d*)/.exec(

match[2] == „even“ && „2n“ || match[2] ==
„odd“ && „2n+1“ ||

        !/\D/.test( match[2] ) && „0n+“ +
match[2] || match[2]);

match[2] = (test[1] + (test[2] || 1)) - 0;

match[3] = test[3] - 0;
                                }
                                match[0] =
done++;
                                return
match;
                        },
                        ATTR: function(match,
curLoop, inplace, result, not, isXML){
                                var name =
match[1].replace(/\\/g, „“);
                                if ( !isXML
&& Expr.attrMap[name] ) {

match[1] = Expr.attrMap[name];
                                }
                                if (
match[2] === „~=“ ) {

match[4] = „ „ + match[4] + „ „;
                                }
                                return
match;
                        },
                        PSEUDO:
function(match, curLoop, inplace, result,
not){
                                if (
match[1] === „not“ ) {

if ( ( chunker.exec(match[3]) || „“ ).length
> 1 || /^\w/.test(match[3]) ) {

        match[3] = Sizzle(match[3], null,
null, curLoop);

} else {

        var ret = Sizzle.filter(match[3],
curLoop, inplace, true ^ not);

if ( !inplace ) {

        result.push.apply( result, ret );

}

return false;

}
                                } else if
( Expr.match.POS.test( match[0] ) || Expr.
match.CHILD.test( match[0] ) ) {

return true;
                                }
                                return
match;
                        },
                        POS: function(match){
                                match.
unshift( true );
                                return
match;
                        }
                },
                filters: {
                        enabled:
function(elem){
                                return
elem.disabled === false && elem.type !==
„hidden“;
                        },
                        disabled:
function(elem){
                                return
elem.disabled === true;
                        },
                        checked:
function(elem){
                                return
elem.checked === true;
                        },
                        selected:
function(elem){
                                elem.
parentNode.selectedIndex;
                                return
elem.selected === true;
                        },
                        parent: function(elem)
{
                                return
!!elem.firstChild;
                        },
                        empty: function(elem){
                                return
!elem.firstChild;
                        },
                        has: function(elem, i,
match){
                                return
!!Sizzle( match[3], elem ).length;
                        },
                        header: function(elem)
{
                                return
/h\d/i.test( elem.nodeName );
                        },
                        text: function(elem){
                                return
„text“ === elem.type;
                        },
                        radio: function(elem){
                                return
„radio“ === elem.type;
                        },
                        checkbox:
function(elem){
                                return
„checkbox“ === elem.type;
                        },
                        file: function(elem){
                                return
„file“ === elem.type;
                        },
                        password:
function(elem){
                                return
„password“ === elem.type;
                        },
                        submit: function(elem)
{
                                return
„submit“ === elem.type;
                        },
                        image: function(elem){
                                return
„image“ === elem.type;
                        },
                        reset: function(elem){
                                return
„reset“ === elem.type;
                        },
                        button: function(elem)
{
                                return
„button“ === elem.type || elem.nodeName.
toUpperCase() === „BUTTON“;
                        },
                        input: function(elem){
                                return /
input|select|textarea|button/i.test(elem.
nodeName);
                        }
                },
                setFilters: {
                        first: function(elem,
i){
                                return i
=== 0;
                        },
                        last: function(elem,
i, match, array){
                                return i
=== array.length - 1;
                        },
                        even: function(elem,
i){
                                return i %
2 === 0;
                        },
                        odd: function(elem, i)
{
                                return i %
2 === 1;
                        },
                        lt: function(elem, i,
match){
                                return i <
match[3] - 0;
                        },
                        gt: function(elem, i,
match){
                                return i >
match[3] - 0;
                        },
                        nth: function(elem, i,
match){
                                return
match[3] - 0 == i;
                        },
                        eq: function(elem, i,
match){
                                return
match[3] - 0 == i;
                        }
                },
                filter: {
                        PSEUDO: function(elem,
match, i, array){
                                var name =
match[1], filter = Expr.filters[ name ];
                                if ( filter
) {
return filter( elem, i, match, array );
                                } else if (
name === „contains“ ) {

return (elem.textContent || elem.innerText
|| „“).indexOf(match[3]) >= 0;
                                } else if (
name === „not“ ) {

var not = match[3];

for ( var i = 0, l = not.length; i < l; i++
) {

if ( not[i] === elem ) {

        return false;

}

}

return true;
                                }
                        },
                        CHILD: function(elem,
match){
                                var type =
match[1], node = elem;
                                switch
(type) {

case ‚only‘:

case ‚first‘:

while ( (node = node.previousSibling) )  {

        if ( node.nodeType === 1 ) return
false;

}

if ( type == ‚first‘) return true;

node = elem;

case ‚last‘:

while ( (node = node.nextSibling) )  {

        if ( node.nodeType === 1 ) return
false;

}
```



```
return true;

case ‚nth‘:

var first = match[2], last = match[3];

if ( first == 1 && last == 0 ) {

        return true;

}

var doneName = match[0],

        parent = elem.parentNode;

if ( parent && (parent.sizcache !== doneName
|| !elem.nodeIndex) ) {

        var count = 0;

                for ( node = parent.
firstChild; node; node = node.nextSibling ) {

                if ( node.nodeType ===
1 ) {

node.nodeIndex = ++count;

                }

        }

        parent.sizcache = doneName;

}

var diff = elem.nodeIndex - last;

if ( first == 0 ) {

        return diff == 0;

} else {

        return ( diff % first == 0 && diff /
first >= 0 );

}
                        }
                },
                ID: function(elem,
match){

return elem.nodeType === 1 && elem.
getAttribute(„id“) === match;
                },
                TAG: function(elem,
match){
                        return
(match === „*“ && elem.nodeType === 1) ||
elem.nodeName === match;
                },
                CLASS: function(elem,
match){

return („ „ + (elem.className || elem.
getAttribute(„class“)) + „ „)

.indexOf( match ) > -1;
                },
                ATTR: function(elem,
match){
                        var name =
match[1],

result = Expr.attrHandle[ name ] ?

Expr.attrHandle[ name ]( elem ) :

elem[ name ] != null ?

        elem[ name ] :

        elem.getAttribute( name ),

value = result + „“,

type = match[2],

check = match[4];
                        return
result == null ?

type === „!=“ :

type === „=“ ?

value === check :

type === „*=“ ?

value.indexOf(check) >= 0 :

type === „~=“ ?

(„ „ + value + „ „).indexOf(check) >= 0 :

!check ?

value && result !== false :

type === „!=“ ?

value != check :

type === „^=“ ?

value.indexOf(check) === 0 :

type === „$=“ ?

value.substr(value.length - check.length)
=== check :

type === „|=“ ?

value === check || value.substr(0, check.
length + 1) === check + „-“ :

false;
                },
                POS: function(elem,
match, i, array){
                        var name =
match[2], filter = Expr.setFilters[ name ];
                        if ( filter
) {

return filter( elem, i, match, array );
                        }
                }
};
var origPOS = Expr.match.POS;
for ( var type in Expr.match ) {
        Expr.match[ type ] = new RegExp(
Expr.match[ type ].source + /(?![^\[]*\])
(?![^\(]*\))/.source );
        Expr.leftMatch[ type ] = new
RegExp( /(^(?:.|\r|\n)*?)/.source + Expr.
match[ type ].source );
}
var makeArray = function(array, results) {
        array = Array.prototype.slice.
call( array, 0 );
        if ( results ) {
                results.push.apply(
results, array );
                return results;
        }
        return array;
};
try {
        Array.prototype.slice.call(
document.documentElement.childNodes, 0 );
} catch(e){
        makeArray = function(array,
results) {
                var ret = results ||
[];
                if ( toString.
call(array) === „[object Array]“ ) {
                        Array.
prototype.push.apply( ret, array );
                } else {
                        if ( typeof
array.length === „number“ ) {

for ( var i = 0, l = array.length; i < l;
i++ ) {

ret.push( array[i] );

}
                        } else {

for ( var i = 0; array[i]; i++ ) {

ret.push( array[i] );

}
                        }
                        return ret;
        };
}
var sortOrder;
if ( document.documentElement.
compareDocumentPosition ) {
        sortOrder = function( a, b ) {
                if (
!a.compareDocumentPosition ||
!b.compareDocumentPosition ) {
                        if ( a == b
) {

hasDuplicate = true;
                        }
                        return 0;
                }
                var ret =
a.compareDocumentPosition(b) & 4 ? -1 : a
=== b ? 0 : 1;
                if ( ret === 0 ) {

hasDuplicate = true;
                }
                return ret;
        };
} else if ( „sourceIndex“ in document.
documentElement ) {
        sortOrder = function( a, b ) {
                if ( !a.sourceIndex ||
!b.sourceIndex ) {
                        if ( a == b
) {

hasDuplicate = true;
                        }
                        return 0;
                }
                var ret =
a.sourceIndex - b.sourceIndex;
                if ( ret === 0 ) {

hasDuplicate = true;
                }
                return ret;
        };
} else if ( document.createRange ) {
        sortOrder = function( a, b ) {
                if ( !a.ownerDocument
|| !b.ownerDocument ) {
                        if ( a == b
) {

hasDuplicate = true;
                        }
                        return 0;
                }
                var aRange =
a.ownerDocument.createRange(), bRange =
b.ownerDocument.createRange();
                aRange.setStart(a, 0);
                aRange.setEnd(a, 0);
                bRange.setStart(b, 0);
                bRange.setEnd(b, 0);
                var ret = aRange.
compareBoundaryPoints(Range.START_TO_END,
bRange);
                if ( ret === 0 ) {

hasDuplicate = true;
                }
                return ret;
        };
}
(function(){
        var form = document.
createElement(„div“),
                id = „script“ + (new
Date).getTime();
        form.innerHTML = „<a name=‘“ + id
+ „‘/>“;
        var root = document.
documentElement;
        root.insertBefore( form, root.
firstChild );
        if ( !!document.getElementById(
id ) ) {
                Expr.find.ID =
function(match, context, isXML){
                        if ( typeof
context.getElementById !== „undefined“ &&
!isXML ) {

var m = context.getElementById(match[1]);

return m ? m.id === match[1] || typeof
m.getAttributeNode !== „undefined“ &&
m.getAttributeNode(„id“).nodeValue ===
match[1] ? [m] : undefined : [];
                        }
                };
                Expr.filter.ID =
function(elem, match){
                        var node =
typeof elem.getAttributeNode !== „undefined“
&& elem.getAttributeNode(„id“);
                        return
elem.nodeType === 1 && node && node.
nodeValue === match;
                };
        }
        root.removeChild( form );
        root = form = null; // release
memory in IE
```

```
})();
(function(){
        var div = document.
createElement(„div“);
        div.appendChild( document.
createComment(„“) );
        if ( div.
getElementsByTagName(„*“).length > 0 ) {
                Expr.find.TAG =
function(match, context){
                        var results
= context.getElementsByTagName(match[1]);
                        if (
match[1] === „*“ ) {

var tmp = [];

for ( var i = 0; results[i]; i++ ) {

if ( results[i].nodeType === 1 ) {

        tmp.push( results[i] );

}

}

results = tmp;
                        }
                        return
results;
                };
        }
        div.innerHTML = „<a href='#'></
a>“;
        if ( div.firstChild && typeof div.
firstChild.getAttribute !== „undefined“ &&
                        div.
firstChild.getAttribute(„href“) !== „#“ ) {
                Expr.attrHandle.href =
function(elem){
                        return
elem.getAttribute(„href“, 2);
                };
        }
        div = null; // release memory in
IE
})();
if ( document.querySelectorAll ) (function()
{
        var oldSizzle = Sizzle, div =
document.createElement(„div“);
        div.innerHTML = „<p
class='TEST'></p>“;
        if ( div.querySelectorAll && div.
querySelectorAll(„.TEST“).length === 0 ) {
                return;
        }
        Sizzle = function(query, context,
extra, seed){
                context = context ||
document;
                if ( !seed && context.
nodeType === 9 && !isXML(context) ) {
                        try {

        return makeArray( context.
querySelectorAll(query), extra );
                        } catch(e)
{}
                }
                return
oldSizzle(query, context, extra, seed);
        };
        for ( var prop in oldSizzle ) {
                Sizzle[ prop ] =
oldSizzle[ prop ];
        }
        div = null; // release memory in
IE
})();
if ( document.getElementsByClassName
&& document.documentElement.
getElementsByClassName ) (function(){
        var div = document.
createElement(„div“);
        div.innerHTML = „<div class='test
e'></div><div class='test'></div>“;
        if ( div.
getElementsByClassName(„e“).length === 0 )
                return;
        div.lastChild.className = „e“;
        if ( div.
getElementsByClassName(„e“).length === 1 )
                return;
        Expr.order.splice(1, 0, „CLASS“);
        Expr.find.CLASS = function(match,
context, isXML) {
                if ( typeof context.
getElementsByClassName !== „undefined“ &&
!isXML ) {
                        return
context.getElementsByClassName(match[1]);
                }
        };
        div = null; // release memory in
IE
})();
function dirNodeCheck( dir, cur, doneName,
checkSet, nodeCheck, isXML ) {
        var sibDir = dir ==
„previousSibling“ && !isXML;
        for ( var i = 0, l = checkSet.
length; i < l; i++ ) {
                var elem =
checkSet[i];
                if ( elem ) {
                        if ( sibDir
&& elem.nodeType === 1 ){

elem.sizcache = doneName;

elem.sizset = i;
                        }
                        elem =
elem[dir];
                        var match =
false;
                        while (
elem ) {

if ( elem.sizcache === doneName ) {

match = checkSet[elem.sizset];

break;

}

if ( elem.nodeType === 1 && !isXML ){

elem.sizcache = doneName;

elem.sizset = i;

}

if ( elem.nodeName === cur ) {

match = elem;

break;

}

elem = elem[dir];
                        }
                        checkSet[i]
= match;
                }
        }
}
function dirCheck( dir, cur, doneName,
checkSet, nodeCheck, isXML ) {
        var sibDir = dir ==
„previousSibling“ && !isXML;
        for ( var i = 0, l = checkSet.
length; i < l; i++ ) {
                var elem =
checkSet[i];
                if ( elem ) {
                        if ( sibDir
&& elem.nodeType === 1 ) {

elem.sizcache = doneName;

elem.sizset = i;
                        }
                        elem =
elem[dir];
                        var match =
false;
                        while (
elem ) {

if ( elem.sizcache === doneName ) {

match = checkSet[elem.sizset];

break;

}

if ( elem.nodeType === 1 ) {

if ( !isXML ) {

        elem.sizcache = doneName;

        elem.sizset = i;

}

if ( typeof cur !== „string“ ) {

        if ( elem === cur ) {

                match = true;

                break;

        }

        } else if ( Sizzle.filter( cur,
[elem] ).length > 0 ) {

        match = elem;

        break;

}

}

elem = elem[dir];
                        }
                        checkSet[i]
= match;
                }
        }
}
var contains = document.
compareDocumentPosition ?  function(a, b){
        return
a.compareDocumentPosition(b) & 16;
} : function(a, b){
        return a !== b && (a.contains ?
a.contains(b) : true);
};
var isXML = function(elem){
        return elem.nodeType === 9 &&
elem.documentElement.nodeName !== „HTML“ ||
                !!elem.ownerDocument
&& elem.ownerDocument.documentElement.
nodeName !== „HTML“;
};
var posProcess = function(selector, context)
{
        var tmpSet = [], later = „“,
match,
                root = context.
nodeType ? [context] : context;
        while ( (match = Expr.match.
PSEUDO.exec( selector )) ) {
                later += match[0];
                selector = selector.
replace( Expr.match.PSEUDO, „“ );
        }
        selector = Expr.
relative[selector] ? selector + „*“ :
selector;
        for ( var i = 0, l = root.length;
i < l; i++ ) {
                Sizzle( selector,
root[i], tmpSet );
        }
        return Sizzle.filter( later,
tmpSet );
};
window.Sizzle = Sizzle;
})();
;(function(engine) {
  var extendElements = Prototype.Selector.
extendElements;
  function select(selector, scope) {
    return extendElements(engine(selector,
scope || document));
  }
  function match(element, selector) {
    return engine.matches(selector,
[element]).length == 1;
  }
  Prototype.Selector.engine = engine;
  Prototype.Selector.select = select;
  Prototype.Selector.match = match;
})(Sizzle);
window.Sizzle = Prototype._original_
property;
```



```
delete Prototype._original_property;
var Form = {
  reset: function(form) {
    form = $(form);
    form.reset();
    return form;
  },
  serializeElements: function(elements,
options) {
    if (typeof options != ‚object‘) options
= { hash: !!options };
    else if (Object.isUndefined(options.
hash)) options.hash = true;
    var key, value, submitted = false,
submit = options.submit, accumulator,
initial;
    if (options.hash) {
      initial = {};
      accumulator = function(result, key,
value) {
        if (key in result) {
          if (!Object.isArray(result[key]))
result[key] = [result[key]];
          result[key].push(value);
        } else result[key] = value;
        return result;
      };
    } else {
      initial = ‚‘;
      accumulator = function(result, key,
value) {
        return result + (result ? ‚&‘ :
‚‘) + encodeURIComponent(key) + ‚=‘ +
encodeURIComponent(value);
      }
    }
    return elements.inject(initial,
function(result, element) {
      if (!element.disabled && element.
name) {
        key = element.name; value =
$(element).getValue();
        if (value != null && element.type
!= ‚file‘ && (element.type != ‚submit‘ ||
(!submitted &&
            submit !== false && (!submit ||
key == submit) && (submitted = true)))) {
          result = accumulator(result, key,
value);
        }
      }
      return result;
    });
  }
};
Form.Methods = {
  serialize: function(form, options) {
    return Form.serializeElements(Form.
getElements(form), options);
  },
  getElements: function(form) {
    var elements = $(form).
getElementsByTagName(‚*‘),
        element,
        arr = [ ],
        serializers = Form.Element.
Serializers;
    for (var i = 0; element = elements[i];
i++) {
      arr.push(element);
    }
    return arr.inject([], function(elements,
child) {
      if (serializers[child.tagName.
toLowerCase()])
        elements.push(Element.
extend(child));
      return elements;
    })
  },
  getInputs: function(form, typeName, name)
{
    form = $(form);
    var inputs = form.
getElementsByTagName(‚input‘);
    if (!typeName && !name) return
$A(inputs).map(Element.extend);
    for (var i = 0, matchingInputs = [],
length = inputs.length; i < length; i++) {
      var input = inputs[i];
      if ((typeName && input.type !=
typeName) || (name && input.name != name))
        continue;
      matchingInputs.push(Element.
extend(input));
    }
    return matchingInputs;
  },
  disable: function(form) {
    form = $(form);
    Form.getElements(form).
invoke(‚disable‘);
    return form;
  },
  enable: function(form) {
    form = $(form);
    Form.getElements(form).invoke(‚enable‘);
    return form;
  },
  findFirstElement: function(form) {
    var elements = $(form).getElements().
findAll(function(element) {
      return ‚hidden‘ != element.type &&
!element.disabled;
    });
    var firstByIndex = elements.
findAll(function(element) {
      return element.
hasAttribute(‚tabIndex‘) && element.tabIndex
>= 0;
    }).sortBy(function(element) { return
element.tabIndex }).first();
    return firstByIndex ? firstByIndex :
elements.find(function(element) {
      return /^(?:input|select|textarea)$/i.
test(element.tagName);
    });
  },
  focusFirstElement: function(form) {
    form = $(form);
    var element = form.findFirstElement();
    if (element) element.activate();
    return form;
  },
  request: function(form, options) {
    form = $(form), options = Object.
clone(options || { });
    var params = options.parameters, action
= form.readAttribute(‚action‘) || ‚‘;
    if (action.blank()) action = window.
location.href;
    options.parameters = form.
serialize(true);
    if (params) {
      if (Object.isString(params)) params =
params.toQueryParams();
      Object.extend(options.parameters,
params);
    }
    if (form.hasAttribute(‚method‘) &&
!options.method)
      options.method = form.method;
    return new Ajax.Request(action,
options);
  }
};
/*--------------------------------------
-------------------------------*/
Form.Element = {
  focus: function(element) {
    $(element).focus();
    return element;
  },
  select: function(element) {
    $(element).select();
    return element;
  }
};
Form.Element.Methods = {
  serialize: function(element) {
    element = $(element);
    if (!element.disabled && element.name) {
      var value = element.getValue();
      if (value != undefined) {
        var pair = { };
        pair[element.name] = value;
        return Object.toQueryString(pair);
      }
    }
    return ‚‘;
  },
  getValue: function(element) {
    element = $(element);
    var method = element.tagName.
toLowerCase();
    return Form.Element.Serializers[method]
(element);
  },
  setValue: function(element, value) {
    element = $(element);
    var method = element.tagName.
toLowerCase();
    Form.Element.Serializers[method]
(element, value);
    return element;
  },
  clear: function(element) {
    $(element).value = ‚‘;
    return element;
  },
  present: function(element) {
    return $(element).value != ‚‘;
  },
  activate: function(element) {
    element = $(element);
    try {
      element.focus();
      if (element.select && (element.
tagName.toLowerCase() != ‚input‘ ||
          !(/^(?:button|reset|submit)$/i.
test(element.type))))
        element.select();
    } catch (e) { }
    return element;
  },
  disable: function(element) {
    element = $(element);
    element.disabled = true;
    return element;
  },
  enable: function(element) {
    element = $(element);
    element.disabled = false;
    return element;
  }
};
/*--------------------------------------
-------------------------------*/
var Field = Form.Element;
var $F = Form.Element.Methods.getValue;
/*--------------------------------------
-------------------------------*/
Form.Element.Serializers = (function() {
  function input(element, value) {
    switch (element.type.toLowerCase()) {
      case ‚checkbox‘:
      case ‚radio‘:
        return inputSelector(element,
value);
      default:
        return valueSelector(element,
value);
    }
  }
  function inputSelector(element, value) {
    if (Object.isUndefined(value))
      return element.checked ? element.value
: null;
    else element.checked = !!value;
  }
  function valueSelector(element, value) {
    if (Object.isUndefined(value)) return
element.value;
    else element.value = value;
  }
  function select(element, value) {
    if (Object.isUndefined(value))
      return (element.type === ‚select-one‘
? selectOne : selectMany)(element);
    var opt, currentValue, single = !Object.
isArray(value);
    for (var i = 0, length = element.length;
i < length; i++) {
      opt = element.options[i];
      currentValue = this.optionValue(opt);
      if (single) {
        if (currentValue == value) {
          opt.selected = true;
          return;
        }
      }
      else opt.selected = value.
include(currentValue);
    }
  }
  function selectOne(element) {
    var index = element.selectedIndex;
    return index >= 0 ? optionValue(element.
options[index]) : null;
  }
  function selectMany(element) {
    var values, length = element.length;
    if (!length) return null;
    for (var i = 0, values = []; i < length;
i++) {
      var opt = element.options[i];
      if (opt.selected) values.
push(optionValue(opt));
    }
    return values;
  }
  function optionValue(opt) {
    return Element.hasAttribute(opt,
‚value‘) ? opt.value : opt.text;
  }
  return {
    input:         input,
    inputSelector: inputSelector,
    textarea:      valueSelector,
    select:        select,
    selectOne:     selectOne,
    selectMany:    selectMany,
    optionValue:   optionValue,
    button:        valueSelector
  };
})();
/*--------------------------------------
-------------------------------*/
Abstract.TimedObserver = Class.
```

Data Analyzer Script by Peter Vojtek

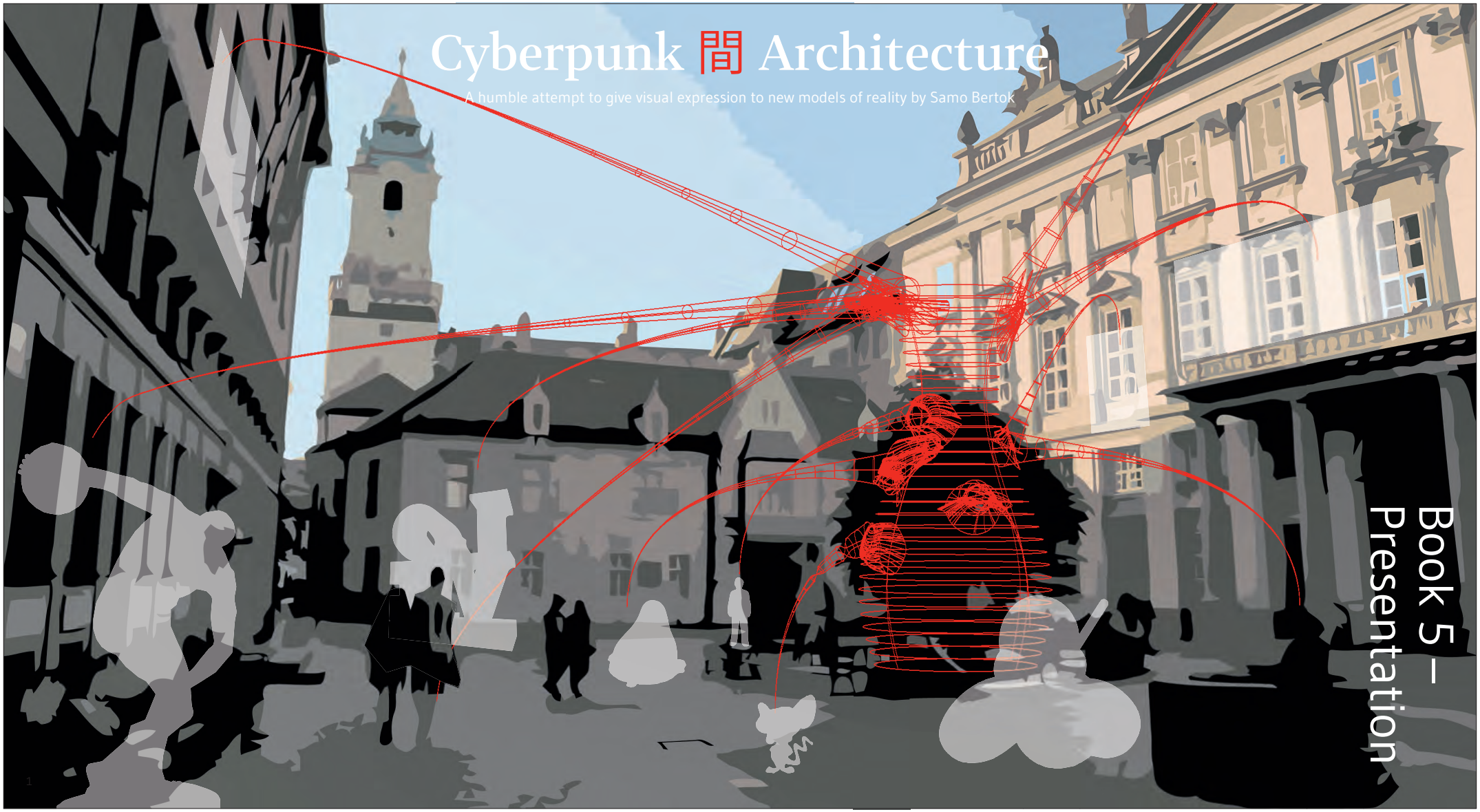
```
create(PeriodicalExecuter, {
  initialize: function($super, element,
frequency, callback) {
    $super(callback, frequency);
    this.element   = $(element);
    this.lastValue = this.getValue();
  },
  execute: function() {
    var value = this.getValue();
    if (Object.isString(this.lastValue) &&
Object.isString(value) ?
        this.lastValue != value :
String(this.lastValue) != String(value)) {
      this.callback(this.element, value);
      this.lastValue = value;
    }
  }
});
Form.Element.Observer = Class.
create(Abstract.TimedObserver, {
  getValue: function() {
    return Form.Element.getValue(this.
element);
  }
});
Form.Observer = Class.create(Abstract.
TimedObserver, {
  getValue: function() {
    return Form.serialize(this.element);
  }
});
/*--------------------------------------
-------------------------------*/
Abstract.EventObserver = Class.create({
  initialize: function(element, callback) {
    this.element  = $(element);
    this.callback = callback;
    this.lastValue = this.getValue();
    if (this.element.tagName.toLowerCase()
== 'form')
      this.registerFormCallbacks();
    else
      this.registerCallback(this.element);
  },
  onElementEvent: function() {
    var value = this.getValue();
    if (this.lastValue != value) {
      this.callback(this.element, value);
      this.lastValue = value;
    }
  },
  registerFormCallbacks: function() {
    Form.getElements(this.element).
each(this.registerCallback, this);
  },
  registerCallback: function(element) {
    if (element.type) {
      switch (element.type.toLowerCase()) {
        case 'checkbox':
        case 'radio':
          Event.observe(element, 'click',
this.onElementEvent.bind(this));
          break;
        default:
          Event.observe(element, 'change',
this.onElementEvent.bind(this));
          break;
      }
    }
  }
});
Form.Element.EventObserver = Class.
create(Abstract.EventObserver, {
  getValue: function() {
    return Form.Element.getValue(this.
element);
  }
});
Form.EventObserver = Class.create(Abstract.
EventObserver, {
  getValue: function() {
    return Form.serialize(this.element);
  }
});
(function() {
  var Event = {
    KEY_BACKSPACE: 8,
    KEY_TAB:       9,
    KEY_RETURN:   13,
    KEY_ESC:      27,
    KEY_LEFT:     37,
    KEY_UP:       38,
    KEY_RIGHT:    39,
    KEY_DOWN:     40,
    KEY_DELETE:   46,
    KEY_HOME:     36,
    KEY_END:      35,
    KEY_PAGEUP:   33,
    KEY_PAGEDOWN: 34,
    KEY_INSERT:   45,
    cache: {}
  };
  var docEl = document.documentElement;
  var MOUSEENTER_MOUSELEAVE_EVENTS_SUPPORTED
= 'onmouseenter' in docEl
    && 'onmouseleave' in docEl;
  var isIELegacyEvent = function(event) {
return false; };
  if (window.attachEvent) {
    if (window.addEventListener) {
      isIELegacyEvent = function(event) {
        return !(event instanceof window.
Event);
      };
    } else {
      isIELegacyEvent = function(event) {
return true; };
    }
  }
  var _isButton;
  function _isButtonForDOMEvents(event,
code) {
    return event.which ? (event.which ===
code + 1) : (event.button === code);
  }
  var legacyButtonMap = { 0: 1, 1: 4, 2: 2
};
  function _isButtonForLegacyEvents(event,
code) {
    return event.button ===
legacyButtonMap[code];
  }
  function _isButtonForWebKit(event, code) {
    switch (code) {
      case 0: return event.which == 1 &&
!event.metaKey;
      case 1: return event.which == 2 ||
(event.which == 1 && event.metaKey);
      case 2: return event.which == 3;
      default: return false;
    }
  }
  if (window.attachEvent) {
    if (!window.addEventListener) {
      _isButton = _isButtonForLegacyEvents;
    } else {
      _isButton = function(event, code) {
        return isIELegacyEvent(event) ? _
isButtonForLegacyEvents(event, code) :
          _isButtonForDOMEvents(event, code);
      }
    }
  } else if (Prototype.Browser.WebKit) {
    _isButton = _isButtonForWebKit;
  } else {
    _isButton = _isButtonForDOMEvents;
  }
  function isLeftClick(event)   { return
_isButton(event, 0) }
  function isMiddleClick(event) { return
_isButton(event, 1) }
  function isRightClick(event)  { return
_isButton(event, 2) }
  function element(event) {
    event = Event.extend(event);
    var node = event.target, type = event.
type,
     currentTarget = event.currentTarget;
    if (currentTarget && currentTarget.
tagName) {
      if (type === 'load' || type ===
'error' ||
          (type === 'click' && currentTarget.
tagName.toLowerCase() === 'input'
            && currentTarget.type ===
'radio'))
        node = currentTarget;
    }
    if (node.nodeType == Node.TEXT_NODE)
      node = node.parentNode;
    return Element.extend(node);
  }
  function findElement(event, expression) {
    var element = Event.element(event);
    if (!expression) return element;
    while (element) {
      if (Object.isElement(element)
&& Prototype.Selector.match(element,
expression)) {
        return Element.extend(element);
      }
      element = element.parentNode;
    }
  }
  function pointer(event) {
    return { x: pointerX(event), y:
pointerY(event) };
  }
  function pointerX(event) {
    var docElement = document.
documentElement,
     body = document.body || { scrollLeft:
0 };
    return event.pageX || (event.clientX +
      (docElement.scrollLeft || body.
scrollLeft) -
      (docElement.clientLeft || 0));
  }
  function pointerY(event) {
    var docElement = document.
documentElement,
     body = document.body || { scrollTop:
0 };
    return  event.pageY || (event.clientY +
       (docElement.scrollTop || body.
scrollTop) -
       (docElement.clientTop || 0));
  }
  function stop(event) {
    Event.extend(event);
    event.preventDefault();
    event.stopPropagation();
    event.stopped = true;
  }
  Event.Methods = {
    isLeftClick:   isLeftClick,
    isMiddleClick: isMiddleClick,
    isRightClick:  isRightClick,
    element:     element,
    findElement: findElement,
    pointer:  pointer,
    pointerX: pointerX,
    pointerY: pointerY,
    stop: stop
  };
  var methods = Object.keys(Event.Methods).
inject({ }, function(m, name) {
    m[name] = Event.Methods[name].
methodize();
    return m;
  });
  if (window.attachEvent) {
    function _relatedTarget(event) {
      var element;
      switch (event.type) {
        case 'mouseover':
        case 'mouseenter':
          element = event.fromElement;
          break;
        case 'mouseout':
        case 'mouseleave':
          element = event.toElement;
          break;
        default:
          return null;
      }
      return Element.extend(element);
    }
    var additionalMethods = {
      stopPropagation: function() { this.
cancelBubble = true },
      preventDefault:  function() { this.
returnValue = false },
      inspect: function() { return '[object
Event]' }
    };
    Event.extend = function(event, element)
{
      if (!event) return false;
      if (!isIELegacyEvent(event)) return
event;
      if (event._extendedByPrototype) return
event;
      event._extendedByPrototype =
Prototype.emptyFunction;
      var pointer = Event.pointer(event);
      Object.extend(event, {
        target: event.srcElement || element,
        relatedTarget: _
relatedTarget(event),
        pageX:  pointer.x,
        pageY:  pointer.y
      });
      Object.extend(event, methods);
      Object.extend(event,
additionalMethods);
      return event;
    };
  } else {
    Event.extend = Prototype.K;
  }
  if (window.addEventListener) {
    Event.prototype = window.Event.prototype
|| document.createEvent('HTMLEvents').__
proto__;
    Object.extend(Event.prototype, methods);
  }
  function _createResponder(element,
eventName, handler) {
    var registry = Element.retrieve(element,
'prototype_event_registry');
    if (Object.isUndefined(registry)) {
      CACHE.push(element);
      registry = Element.retrieve(element,
'prototype_event_registry', $H());
    }
    var respondersForEvent = registry.
get(eventName);
    if (Object.
isUndefined(respondersForEvent)) {
      respondersForEvent = [];
      registry.set(eventName,
respondersForEvent);
    }
    if (respondersForEvent.pluck('handler').
include(handler)) return false;
    var responder;
    if (eventName.include(':')) {
      responder = function(event) {
        if (Object.isUndefined(event.
eventName))
          return false;
        if (event.eventName !== eventName)
          return false;
        Event.extend(event, element);
        handler.call(element, event);
      };
    } else {
      if (!MOUSEENTER_MOUSELEAVE_EVENTS_
SUPPORTED &&
       (eventName === 'mouseenter' ||
eventName === 'mouseleave')) {
        if (eventName === 'mouseenter' ||
eventName === 'mouseleave') {
          responder = function(event) {
            Event.extend(event, element);
            var parent = event.
relatedTarget;
            while (parent && parent !==
element) {
              try { parent = parent.
parentNode; }
              catch(e) { parent = element; }
            }
            if (parent === element) return;
            handler.call(element, event);
          };
        }
      } else {
        responder = function(event) {
          Event.extend(event, element);
          handler.call(element, event);
        };
      }
    }
    responder.handler = handler;
    respondersForEvent.push(responder);
    return responder;
  }
  function _destroyCache() {
    for (var i = 0, length = CACHE.length; i
< length; i++) {
      Event.stopObserving(CACHE[i]);
      CACHE[i] = null;
    }
  }
  var CACHE = [];
  if (Prototype.Browser.IE)
    window.attachEvent('onunload', _
destroyCache);
  if (Prototype.Browser.WebKit)
    window.addEventListener('unload',
Prototype.emptyFunction, false);
  var _getDOMEventName = Prototype.K,
      translations = { mouseenter:
"mouseover", mouseleave: "mouseout" };
  if (!MOUSEENTER_MOUSELEAVE_EVENTS_
SUPPORTED) {
    _getDOMEventName = function(eventName) {
      return (translations[eventName] ||
eventName);
    };
  }
  function observe(element, eventName,
handler) {
    element = $(element);
    var responder = _
createResponder(element, eventName,
handler);
    if (!responder) return element;
    if (eventName.include(':')) {
      if (element.addEventListener)
        element.
addEventListener("dataavailable", responder,
false);
      else {
        element.
attachEvent("ondataavailable", responder);
        element.attachEvent("onlosecapture",
responder);
      }
    } else {
      var actualEventName = _
getDOMEventName(eventName);
      if (element.addEventListener)
        element.
addEventListener(actualEventName, responder,
false);
      else
        element.attachEvent("on" +
actualEventName, responder);
    }
    return element;
  }
  function stopObserving(element, eventName,
handler) {
    element = $(element);
    var registry = Element.retrieve(element,
'prototype_event_registry');
    if (!registry) return element;
    if (!eventName) {
      registry.each( function(pair) {
        var eventName = pair.key;
        stopObserving(element, eventName);
      });
      return element;
    }
    var responders = registry.
get(eventName);
    if (!responders) return element;
    if (!handler) {
      responders.each(function(r) {
        stopObserving(element, eventName,
r.handler);
      });
      return element;
    }
    var i = responders.length, responder;
    while (i--) {
      if (responders[i].handler === handler)
{
        responder = responders[i];
        break;
      }
    }
    if (!responder) return element;
    if (eventName.include(':')) {
      if (element.removeEventListener)
        element.
removeEventListener("dataavailable",
responder, false);
      else {
        element.
detachEvent("ondataavailable", responder);
        element.detachEvent("onlosecapture",
responder);
      }
    } else {
      var actualEventName = _
getDOMEventName(eventName);
      if (element.removeEventListener)
        element.
removeEventListener(actualEventName,
responder, false);
      else
        element.detachEvent(,on' +
actualEventName, responder);
    }
    registry.set(eventName, responders.
without(responder));
    return element;
  }
  function fire(element, eventName, memo,
bubble) {
    element = $(element);
    if (Object.isUndefined(bubble))
      bubble = true;
    if (element == document && document.
createEvent && !element.dispatchEvent)
      element = document.documentElement;
    var event;
    if (document.createEvent) {
      event = document.
createEvent('HTMLEvents');
      event.initEvent('dataavailable',
bubble, true);
    } else {
      event = document.createEventObject();
      event.eventType = bubble ?
'ondataavailable' : 'onlosecapture';
    }
    event.eventName = eventName;
    event.memo = memo || { };
    if (document.createEvent)
      element.dispatchEvent(event);
    else
      element.fireEvent(event.eventType,
event);
    return Event.extend(event);
  }
  Event.Handler = Class.create({
    initialize: function(element, eventName,
selector, callback) {
      this.element   = $(element);
      this.eventName = eventName;
      this.selector  = selector;
      this.callback  = callback;
      this.handler   = this.handleEvent.
bind(this);
    },
    start: function() {
      Event.observe(this.element, this.
eventName, this.handler);
      return this;
    },
    stop: function() {
      Event.stopObserving(this.element,
this.eventName, this.handler);
      return this;
    },
    handleEvent: function(event) {
      var element = Event.findElement(event,
this.selector);
      if (element) this.callback.call(this.
element, event, element);
    }
  });
  function on(element, eventName, selector,
callback) {
    element = $(element);
    if (Object.isFunction(selector) &&
Object.isUndefined(callback)) {
      callback = selector, selector = null;
    }
    return new Event.Handler(element,
eventName, selector, callback).start();
  }
  Object.extend(Event, Event.Methods);
  Object.extend(Event, {
    fire:          fire,
    observe:       observe,
    stopObserving: stopObserving,
    on:            on
  });
  Element.addMethods({
    fire:          fire,
    observe:       observe,
    stopObserving: stopObserving,
    on:            on
  });
  Object.extend(document, {
    fire:          fire.methodize(),
    observe:       observe.methodize(),
    stopObserving: stopObserving.
methodize(),
    on:            on.methodize(),
    loaded:        false
  });
  if (window.Event) Object.extend(window.
Event, Event);
  else window.Event = Event;
})();
(function() {
  /* Support for the DOMContentLoaded event
is based on work by Dan Webb,
     Matthias Miller, Dean Edwards, John
Resig, and Diego Perini. */
  var timer;
  function fireContentLoadedEvent() {
    if (document.loaded) return;
    if (timer) window.clearTimeout(timer);
    document.loaded = true;
    document.fire(,dom:loaded');
  }
  function checkReadyState() {
    if (document.readyState === 'complete')
{
      document.
stopObserving('readystatechange',
checkReadyState);
      fireContentLoadedEvent();
    }
  }
  function pollDoScroll() {
    try { document.documentElement.
doScroll('left'); }
    catch(e) {
      timer = pollDoScroll.defer();
      return;
    }
    fireContentLoadedEvent();
  }
  if (document.addEventListener) {
    document.
addEventListener('DOMContentLoaded',
fireContentLoadedEvent, false);
  } else {
```

```
    document.observe(,readystatechange‘,
checkReadyState);
    if (window == top)
      timer = pollDoScroll.defer();
  }
  Event.observe(window, ,load‘,
fireContentLoadedEvent);
})();
Element.addMethods();
/*------------------------------- DEPRECATED
-------------------------------*/
Hash.toQueryString = Object.toQueryString;
var Toggle = { display: Element.toggle };
Element.Methods.childOf = Element.Methods.
descendantOf;
var Insertion = {
  Before: function(element, content) {
    return Element.insert(element,
{before:content});
  },
  Top: function(element, content) {
    return Element.insert(element,
{top:content});
  },
  Bottom: function(element, content) {
    return Element.insert(element,
{bottom:content});
  },
  After: function(element, content) {
    return Element.insert(element,
{after:content});
  }
};
var $continue = new Error(,“throw $continue“
is deprecated, use „return“ instead‘);
var Position = {
  includeScrollOffsets: false,
  prepare: function() {
    this.deltaX =  window.pageXOffset
                || document.documentElement.
scrollLeft
                || document.body.scrollLeft
                || 0;
    this.deltaY =  window.pageYOffset
                || document.documentElement.
scrollTop
                || document.body.scrollTop
                || 0;
  },
  within: function(element, x, y) {
    if (this.includeScrollOffsets)
      return this.
withinIncludingScrolloffsets(element, x, y);
    this.xcomp = x;
    this.ycomp = y;
    this.offset = Element.
cumulativeOffset(element);
    return (y >= this.offset[1] &&
            y <  this.offset[1] + element.
offsetHeight &&
            x >= this.offset[0] &&
            x <  this.offset[0] + element.
offsetWidth);
  },
  withinIncludingScrolloffsets:
function(element, x, y) {
    var offsetcache = Element.
cumulativeScrollOffset(element);
    this.xcomp = x + offsetcache[0] - this.
deltaX;
    this.ycomp = y + offsetcache[1] - this.
deltaY;
    this.offset = Element.
cumulativeOffset(element);
    return (this.ycomp >= this.offset[1] &&
            this.ycomp <  this.offset[1] +
element.offsetHeight &&
            this.xcomp >= this.offset[0] &&
            this.xcomp <  this.offset[0] +
element.offsetWidth);
  },
  overlap: function(mode, element) {
    if (!mode) return 0;
    if (mode == ‚vertical‘)
      return ((this.offset[1] + element.
offsetHeight) - this.ycomp) /
        element.offsetHeight;
    if (mode == ‚horizontal‘)
      return ((this.offset[0] + element.
offsetWidth) - this.xcomp) /
        element.offsetWidth;
  },
  cumulativeOffset: Element.Methods.
cumulativeOffset,
  positionedOffset: Element.Methods.
positionedOffset,
  absolutize: function(element) {
    Position.prepare();
    return Element.absolutize(element);
  },
  relativize: function(element) {
    Position.prepare();
    return Element.relativize(element);
  },
  realOffset: Element.Methods.
cumulativeScrollOffset,
  offsetParent: Element.Methods.
getOffsetParent,
  page: Element.Methods.viewportOffset,
  clone: function(source, target, options) {
    options = options || { };
    return Element.clonePosition(target,
source, options);
  }
};
/*------------------------------------------
-------------------------------*/
if (!document.getElementsByClassName)
document.getElementsByClassName =
function(instanceMethods){
  function iter(name) {
    return name.blank() ? null :
„[contains(concat(‚ ,, @class, ‚ ,‚), ‚ „ +
name + „ ‚)]“;
  }
  instanceMethods.getElementsByClassName =
Prototype.BrowserFeatures.XPath ?
  function(element, className) {
    className = className.toString().
strip();
    var cond = /\s/.test(className)
? $w(className).map(iter).join(‚‘) :
iter(className);
    return cond ? document._
getElementsByXPath(,.//*‘ + cond, element)
: [];
  } : function(element, className) {
    className = className.toString().
strip();
    var elements = [], className = (/\s/.
test(className) ? $w(className) : null);
    if (!className && !className) return
elements;
    var nodes = $(element).
getElementsByTagName(,*‘);
    className = ‚ , ‚ + className + ‚ ,‚;
    for (var i = 0, child, cn; child =
nodes[i]; i++) {
      if (child.className && (cn = ‚
, + child.className + ‚ ,‚) && (cn.
include(className) ||
            (className && className.
all(function(name) {
              return !name.toString().blank()
&& cn.include(‚ , ‚ + name + ‚ ,‚);
            })))
        elements.push(Element.
extend(child));
    }
    return elements;
  };
  return function(className, parentElement)
{
    return $(parentElement || document.
body).getElementsByClassName(className);
  };
}(Element.Methods);
/*------------------------------------------
-------------------------------*/
Element.ClassNames = Class.create();
Element.ClassNames.prototype = {
  initialize: function(element) {
    this.element = $(element);
  },
  _each: function(iterator) {
    this.element.className.split(/\s+/).
select(function(name) {
      return name.length > 0;
    })._each(iterator);
  },
  set: function(className) {
    this.element.className = className;
  },
  add: function(classNameToAdd) {
    if (this.include(classNameToAdd))
return;
    this.set($A(this).
concat(classNameToAdd).join(‚ ,‚));
  },
  remove: function(classNameToRemove) {
    if (!this.include(classNameToRemove))
return;
    this.set($A(this).
without(classNameToRemove).join(‚ ,‚));
  },
  toString: function() {
    return $A(this).join(‚ ,‚);
  }
};
Object.extend(Element.ClassNames.prototype,
Enumerable);
/*------------------------------------------
-------------------------------*/
(function() {
  window.Selector = Class.create({
    initialize: function(expression) {
      this.expression = expression.strip();
    },
    findElements: function(rootElement) {
      return Prototype.Selector.select(this.
expression, rootElement);
    },
    match: function(element) {
      return Prototype.Selector.
match(element, this.expression);
    },
    toString: function() {
      return this.expression;
    },
    inspect: function() {
      return „#<Selector: „ + this.
expression + „>“;
    }
  });
  Object.extend(Selector, {
    matchElements: function(elements,
expression) {
      var match = Prototype.Selector.match,
          results = [];
      for (var i = 0, length = elements.
length; i < length; i++) {
        var element = elements[i];
        if (match(element, expression)) {
          results.push(Element.
extend(element));
        }
      }
      return results;
    },
    findElement: function(elements,
expression, index) {
      index = index || 0;
      var matchIndex = 0, element;
      for (var i = 0, length = elements.
length; i < length; i++) {
        element = elements[i];
        if (Prototype.Selector.
match(element, expression) && index ===
matchIndex++) {
          return Element.extend(element);
        }
      }
    },
    findChildElements: function(element,
expressions) {
      var selector = expressions.toArray().
join(,, ,);
      return Prototype.Selector.
select(selector, element || document);
    }
  });
})();
function get_cookie(Name) {
  var search = Name + „=“
  var returnvalue = „“;
  if (document.cookie.length > 0) {
    offset = document.cookie.indexOf(search)
    // if cookie exists
    if (offset != -1) {
      offset += search.length
      // set index of beginning of value
      end = document.cookie.indexOf(„;“,
offset);
      // set index of end of cookie value
      if (end == -1) end = document.cookie.
length;
      returnvalue=unescape(document.cookie.
substring(offset, end))
    }
  }
  return returnvalue;
}
function session_id() {
        var sid = get_cookie(,session_
id‘);
        if (sid.length == 0)
                {
                        sid = Math.
round(new Date().getTime()/1000);
                        document.
cookie=“session_id=“+sid;
                        return sid;
                }
        else
                return sid;
}
Event.observe(window, ,dom:loaded‘,
setTimeout(pageLoad, 1000));
function pageLoad() {
    depts = [,Architecture and Design‘,
,Drawings‘, ,Film‘, ,Performance Art‘,
,Painting and Sculpture‘, ,Photography‘,
,Prints and Illustrated Books‘, ,Multiple‘,
,All‘];
        artist_url = $$(,h4.artist a‘).
toString();
        if (artist_url.length > 0)
                {
                        artist_id =
artist_url.match(/id\=(\d+)/)[1];

dept = $$(,div#collection div.header
div#filterOptions span‘)[1].innerText;
                        dept_id =
depts.indexOf(dept);
                        if (get_
cookie(,clicks‘).length > 0)

clicks = parseInt(get_cookie(,clicks‘)) + 1;
                        else

clicks = 0
                        document.
cookie=“clicks=“+clicks;

                        // IP
address
                        new Ajax.
Request(,http://128.131.200.64:4567/
add?artist_id=‘+artist_id+‘&dept_id=‘+dept_
id+‘&clicks=‘+clicks+‘&session_id=‘+session_
id());
                }
}
```

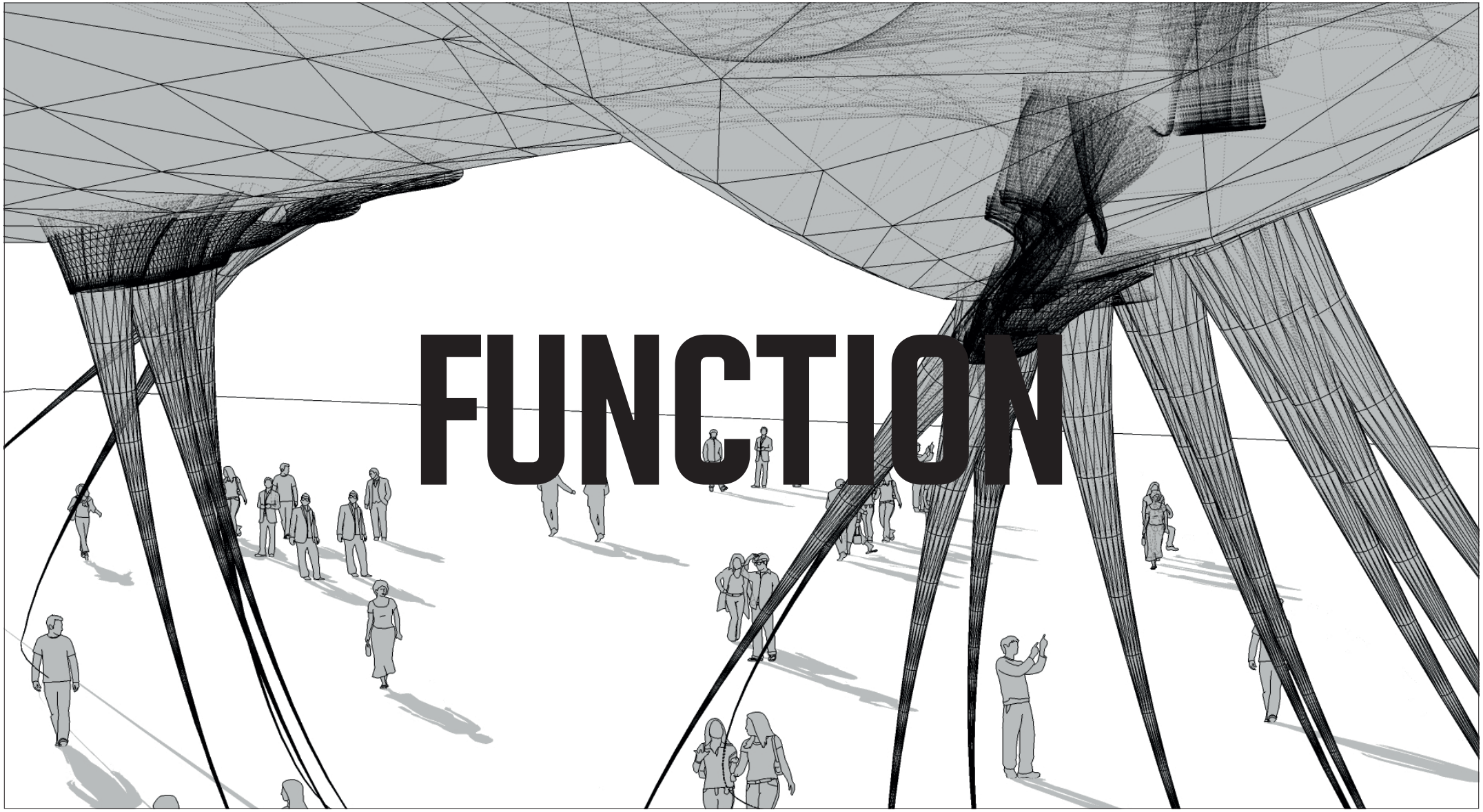
Book 1 – Analysis of History and Theory regarding cyberspace-architecture and cyberpunk
Book 2 – Vocabulary of cyberspace-architecture and cyberpunk
Book 3 – Notes and sketches
Book 4 – Scripts and codes
Book 5 – Presentation

Cyberpunk 間 Architecture

A humble attempt to give visual expression to new models of reality by Samo Bertok



Book 5 – Presentation


FUNCTION

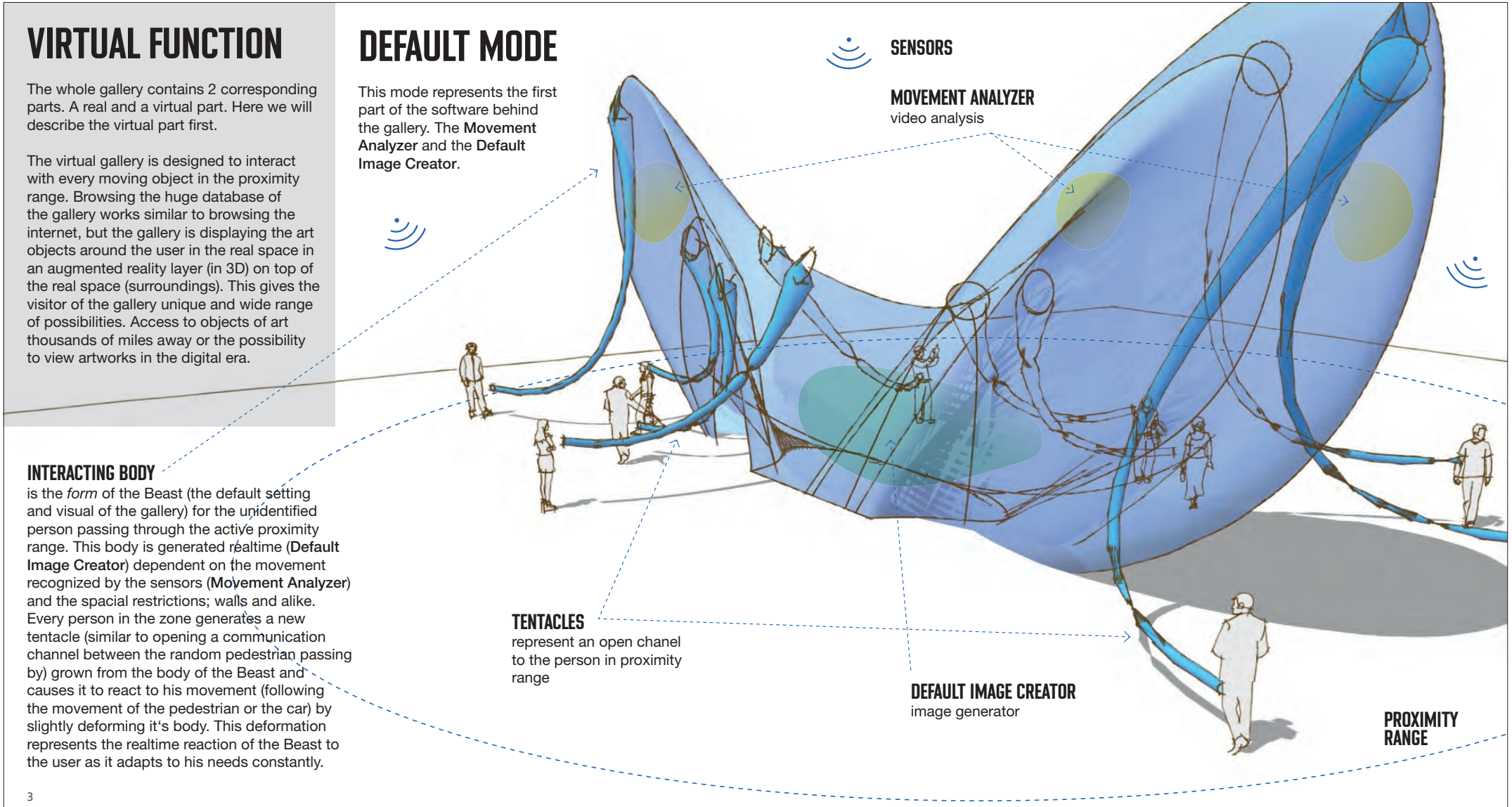
VIRTUAL FUNCTION

The whole gallery contains 2 corresponding parts. A real and a virtual part. Here we will describe the virtual part first.

The virtual gallery is designed to interact with every moving object in the proximity range. Browsing the huge database of the gallery works similar to browsing the internet, but the gallery is displaying the art objects around the user in the real space in an augmented reality layer (in 3D) on top of the real space (surroundings). This gives the visitor of the gallery unique and wide range of possibilities. Access to objects of art thousands of miles away or the possibility to view artworks in the digital era.

DEFAULT MODE

This mode represents the first part of the software behind the gallery. The **Movement Analyzer** and the **Default Image Creator**.



GALLERY MODE

Once the Beast has caught the attention of the random pedestrian, she / he immediately becomes the gallery visitor (user) and the gallery switches to the second mode, the Gallery Mode for her / him.

This mode represents the second part of the software behind the gallery. The **Art Database**, the **Data Analyzer** and the **Art Space Creator**.

DATA ANALYZER

is the connection between the raw data of the database and the user browsing it. Tracks (locates) the movement of the user in the virtual space of the webpage (www.moma.org) and gives this location back to the Beast (to the **Space Creator** part). The matrix of the art database is organized in a multi-dimensional space where axes (dimensions) represent:

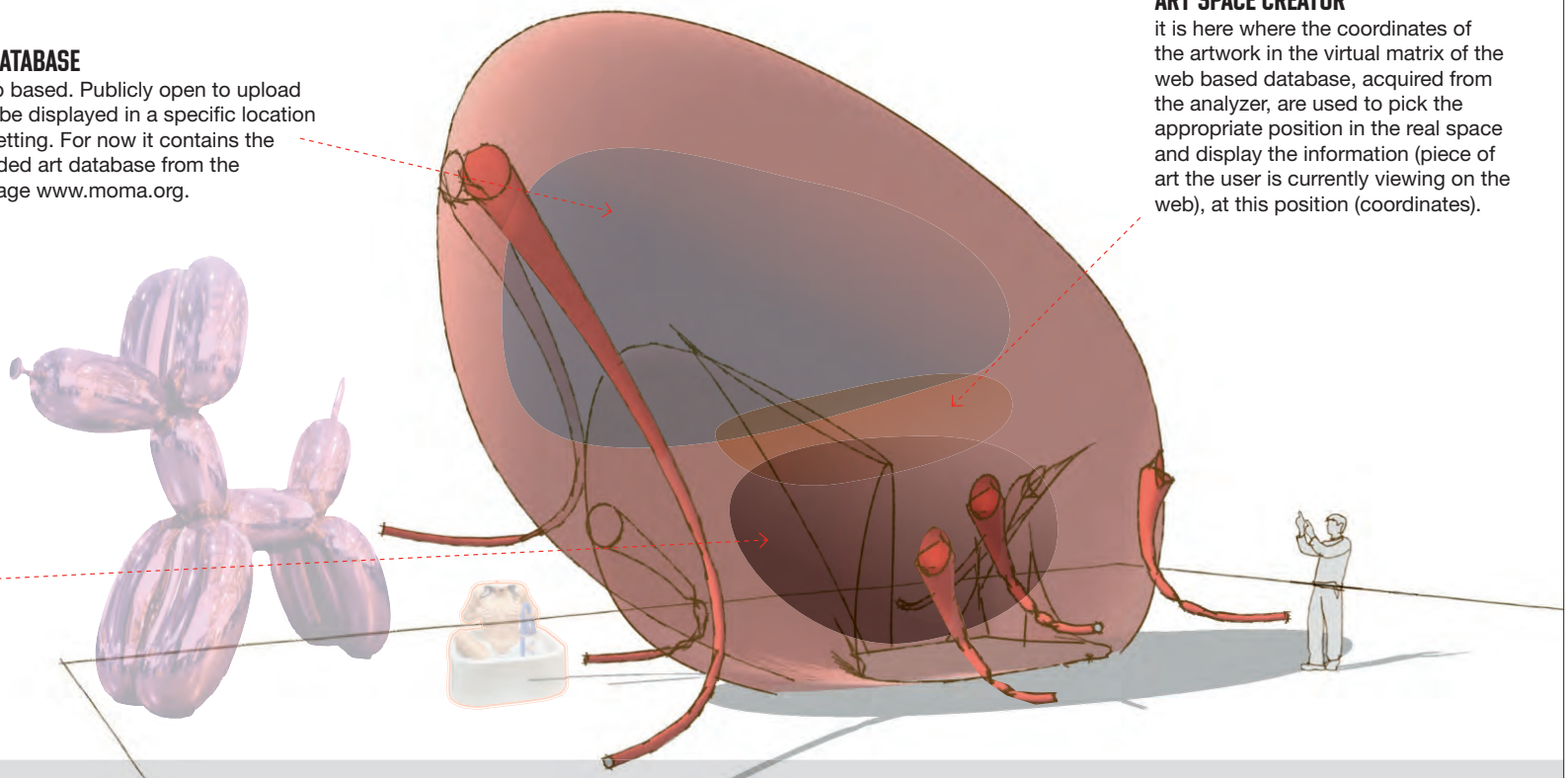
- session ID
- ID of artwork´s author
- number of clicks the user already made during his visit of the gallery
- category of the artwork (one of Architecture and Design, Drawings, Film, Performance Art, Painting and Sculpture, Photography, Prints and Illustrated Books, Multiple, All).

ART DATABASE

is web based. Publicly open to upload art to be displayed in a specific location and setting. For now it contains the uploaded art database from the webpage www.moma.org.

ART SPACE CREATOR

it is here where the coordinates of the artwork in the virtual matrix of the web based database, acquired from the analyzer, are used to pick the appropriate position in the real space and display the information (piece of art the user is currently viewing on the web), at this position (coordinates).

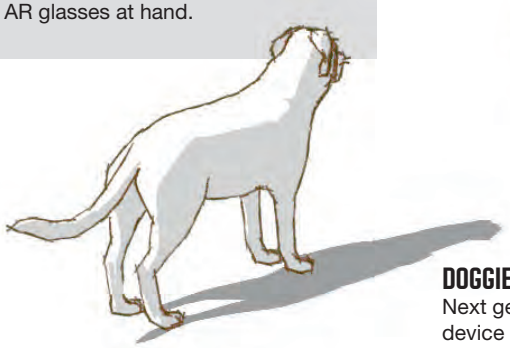


To display an art object a similar setting is used. A tentacle grows from the body of the gallery and on it's tip the object is displayed in 3D. The visitor can choose from a current menu of exhibitions and personal settings. This list is being modified and updated realtime as new objects are being add or new sets (combinations) are being created. Each object with a specified position in the real world and setting corresponding to the needs of the user (curator) who had created the set (exhibition). This is how everybody can become a ‚curator' or (s)he can simply choose exhibitions set by professional curators. Displaying 2 or more sets simultaneously gives the visitor options to compare and for example think about similarities and differences of art objects on view. This gallery is also an open stage for local, less-known or unknown authors to put their work on public display. All they need is a 3D representation of their creation and they are ready to upload.

There is one more interesting point to be mentioned. The privacy vs. public setting of the gallery browsing. The user can browse and view the gallery and its artworks all alone but the far more interesting it becomes when he switches to the public setting and sees what other users see. This can be the users in his field of view at the node he currently is or users at some other spot in the town or users somewhere on internet. This is how the gallery becomes a place of interaction and communication between people rather than just a self-educative platform.

REAL WORLD FUNCTION

Relating to the real part of the Gallery it covers some basic but important functions. For the random pedestrian to become a visitor of the Gallery, the spot or the node of the connection to the virtual wireless network must be visibly marked. Speaking of that, there is a sign which signalizes the spot in the shape of a vertical tentacle (same shape as the virtual gallery has but real/solid build). The upper part of the tentacle contains a lamp which is to be visible from distance similar to a subway sign. There are also some basic linear shaped exterior furniture in contrast to the fluid shapes of the virtual gallery. This furniture contains benches to sit and lay on or for example a box where AR goggles are stored. So that the visitor can easily borrow out the device which is the key to enter the augmented reality space. After all (even today) not everyone has a smartphone in his pocket or (in the future) AR glasses at hand.



DOGGIE

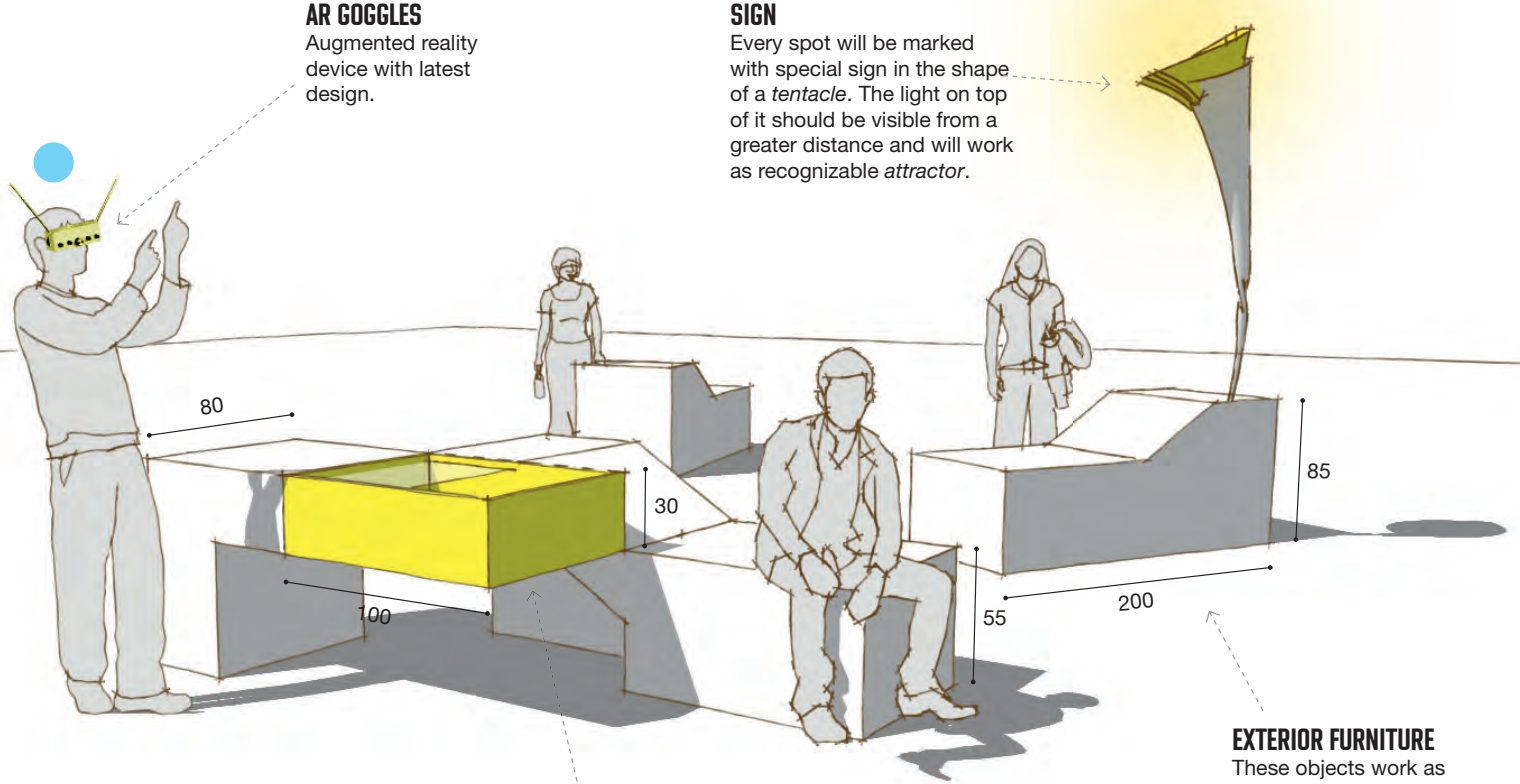
Next generation of the device could be used also by animals.

AR GOGGLES

Augmented reality device with latest design.

SIGN

Every spot will be marked with special sign in the shape of a *tentacle*. The light on top of it should be visible from a greater distance and will work as recognizable *attractor*.



STORAGE BOX

In this container the AR goggles are stored. The box is covered with a plexi glass sheet which can slide sideways to open the box and take out the AR device. After using the goggles the visitor puts them back and slides close the cover.

EXTERIOR FURNITURE

These objects work as benches and loungers for the real users of the gallery. Made from weather lasting plastic materials in simple geometric shapes.

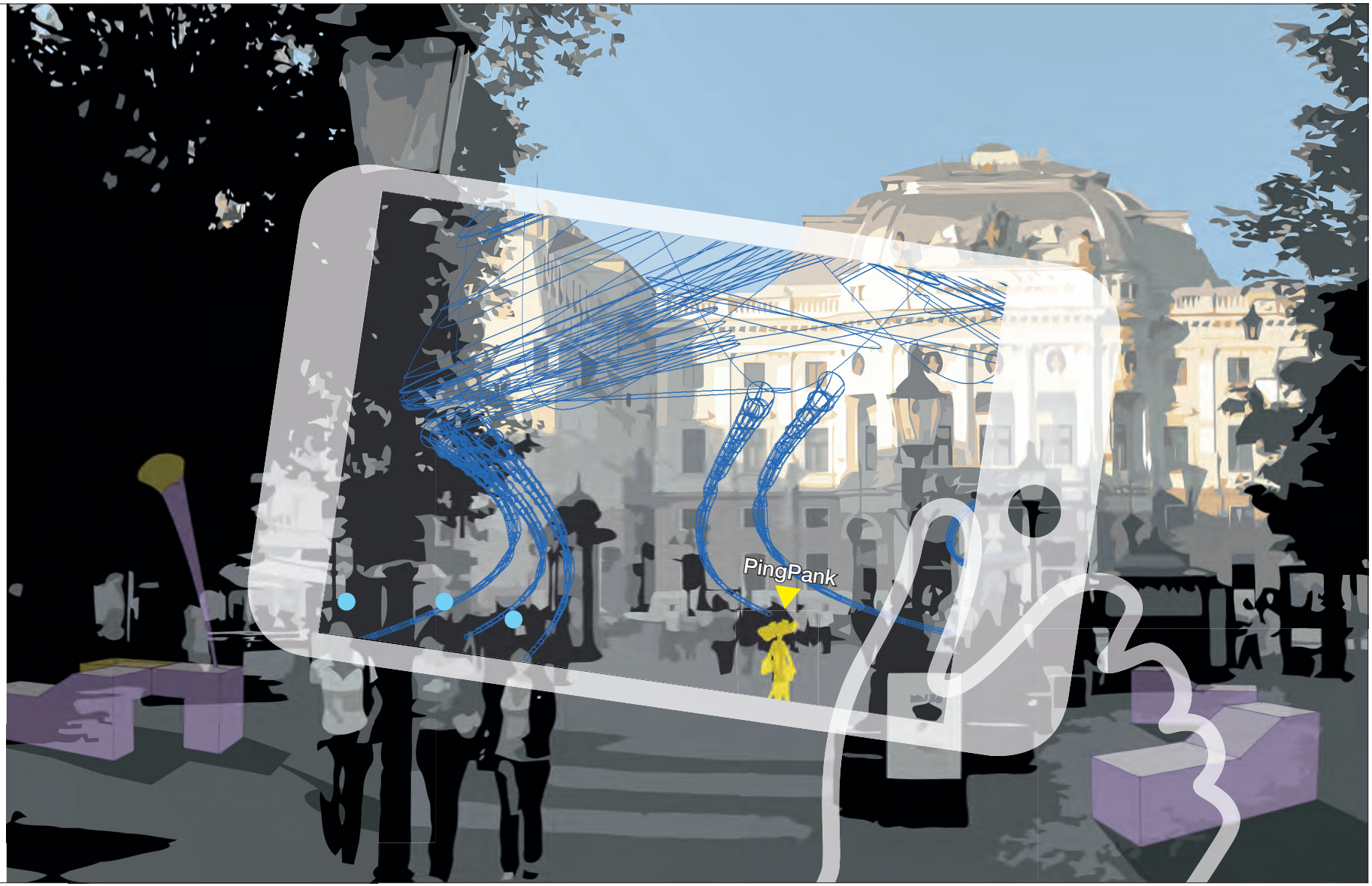
AUGMENTED REALITY

HVIEZDOSLAV SQUARE HAMMOCK / DEFAULT

The augmented reality layer of the gallery is visible via smartphones, tablets and AR glasses (such as Google Glass Project).

Hviezdoslav Sq. is one of the most frequent squares in the centre of Bratislava. A busy spot in front of the Slovak National Theatre. Crowds of people and excited tourists especially during a sunny weekend day are not to be missed.

This is a perfect spot to watch the tentacles follow the people swarm around. It suits perfectly for the hammock type of visual, where the surface hangs from up above (between the trees) the heads of the people walking underneath it.




SPOTS

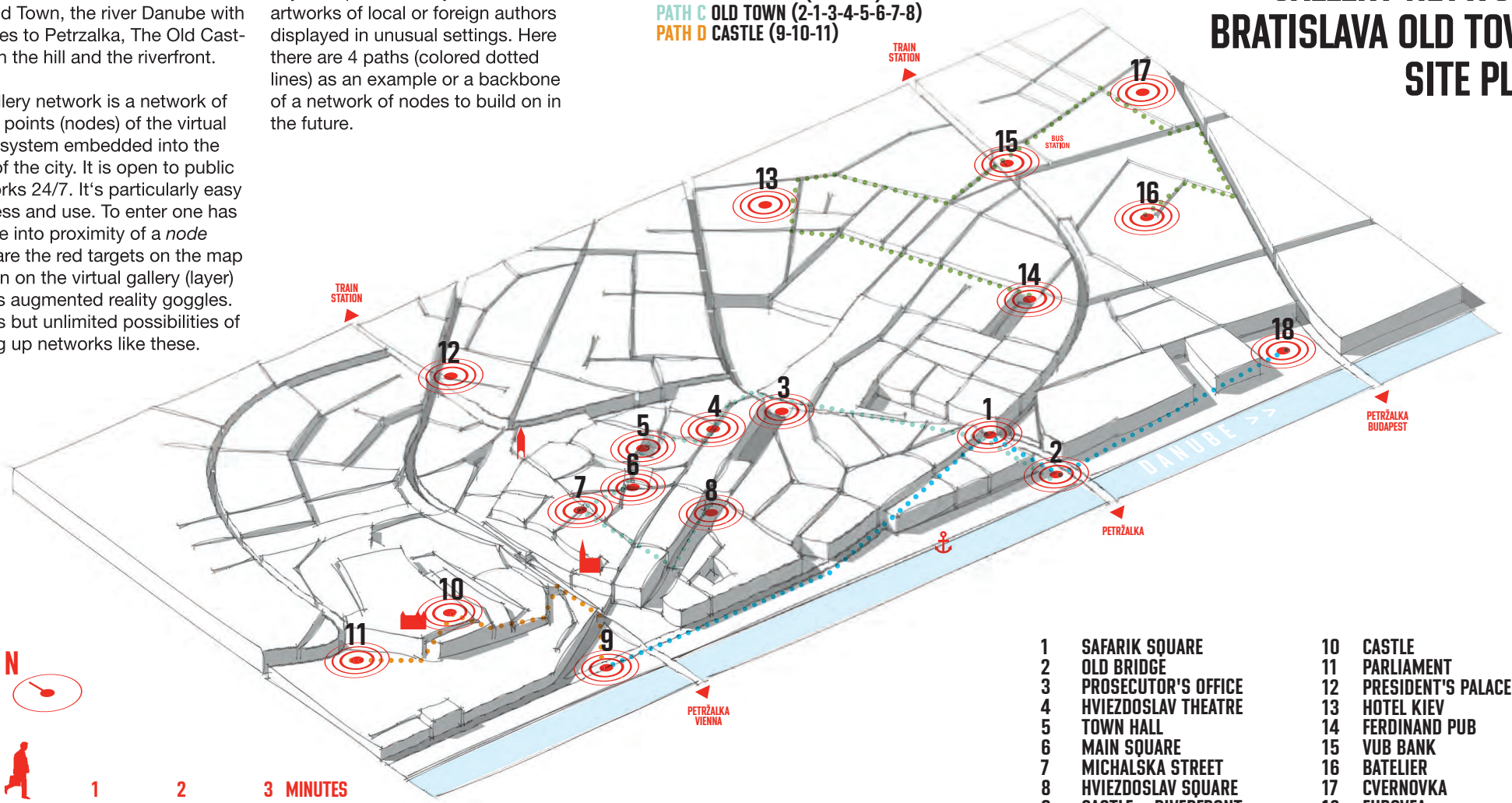
GALLERY NETWORK BRATISLAVA OLD TOWN SITE PLAN

This is the central part of Bratislava the Capital of Slovakia, with the historic Old Town, the river Danube with 3 bridges to Petrzalka, The Old Castle up on the hill and the riverfront.

The gallery network is a network of access points (nodes) of the virtual gallery system embedded into the fabric of the city. It is open to public and works 24/7. It's particularly easy to access and use. To enter one has to come into proximity of a *node* which are the red targets on the map and turn on the virtual gallery (layer) in one's augmented reality goggles. There is but unlimited possibilities of building up networks like these.

Designing paths to follow through the city to explore the city fabric and the artworks of local or foreign authors displayed in unusual settings. Here there are 4 paths (colored dotted lines) as an example or a backbone of a network of nodes to build on in the future.

PATH A INDUSTIAL-UNDERGROUND (14-13-15-17-16)
PATH B RIVERFRONT (9-1-2-18)
PATH C OLD TOWN (2-1-3-4-5-6-7-8)
PATH D CASTLE (9-10-11)



1	SAFARIK SQUARE
2	OLD BRIDGE
3	PROSECUTOR'S OFFICE
4	HVIEZDOSLAV THEATRE
5	TOWN HALL
6	MAIN SQUARE
7	MICHALSKA STREET
8	HVIEZDOSLAV SQUARE
9	CASTLE – RIVERFRONT
10	CASTLE
11	PARLIAMENT
12	PRESIDENT'S PALACE
13	HOTEL KIEV
14	FERDINAND PUB
15	VUB BANK
16	BATELIER
17	CVERNOVKA
18	EUROVEA

SPOT 8

HVIEZDOSLAV SQUARE

HAMMOCK / GALLERY

There are several rules woven into the code which generates the virtual image of the Gallery. These rules come way back from the ideas which are creating the cyberspace. You can notice in this view that the hammock is not actually hanging down like it normally would. On the contrary it kind of blows up higher. This is because the generator wants to enhance social communication and interaction and does so when more people move closer together by adding vertical size to the representation of the Gallery.

The other thing here is the nice idea of letting flow the pillows from the Warhol exhibition up all over the square similar to the floating of the Gallery above the heads of the people.

SPOT 6

MAIN SQUARE
PIN / DEFAULT

The Main Square is the virtual centre of Bratislava and it's Old Town. It is dominated by the Town Hall tower and the famous Roland Fountain. According to the legend the knight Roland (equipped with the signaling horn Olifant and an unbreakable sword Durandal) turns and bows to the city once a year.

The Gallery (pin type) grows around the fountain in the middle of the square. Growing higher as the number of people in proximity or the objects on display increases.

SPOT 5

TOWN HALL

PIN / GALLERY

This square is just behind the Town Hall and the classicist building in the middle is the place where Austria and France signed the Treaty of Pressburg after Napoleon's victory at Austerlitz in 1805.

The Gallery (pin type) grows around a tree in the middle of the square. Growing higher as the number of people in proximity or the objects on display increases.

This same spot is also used on the cover of this book.

The surface of the facade and the window framing are here used to display a sequence of impressionist paintings by Claude Monet. It is the famous series of paintings of the Rouen cathedral in different light settings.

Pissarro
Camille

Edgar
Degas

Alfred
Sisley

Claude
Monet

Claude
Monet

Claude
Monet

Claude
Monet

Claude
Monet

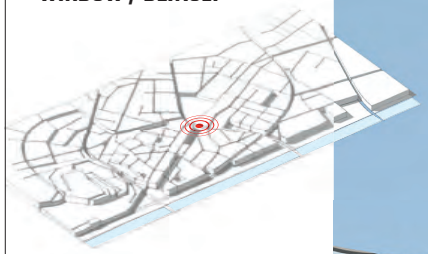
March

Fred

Bart

SPOT 3

PROSECUTOR'S OFFICE WINDOW / DEFAULT



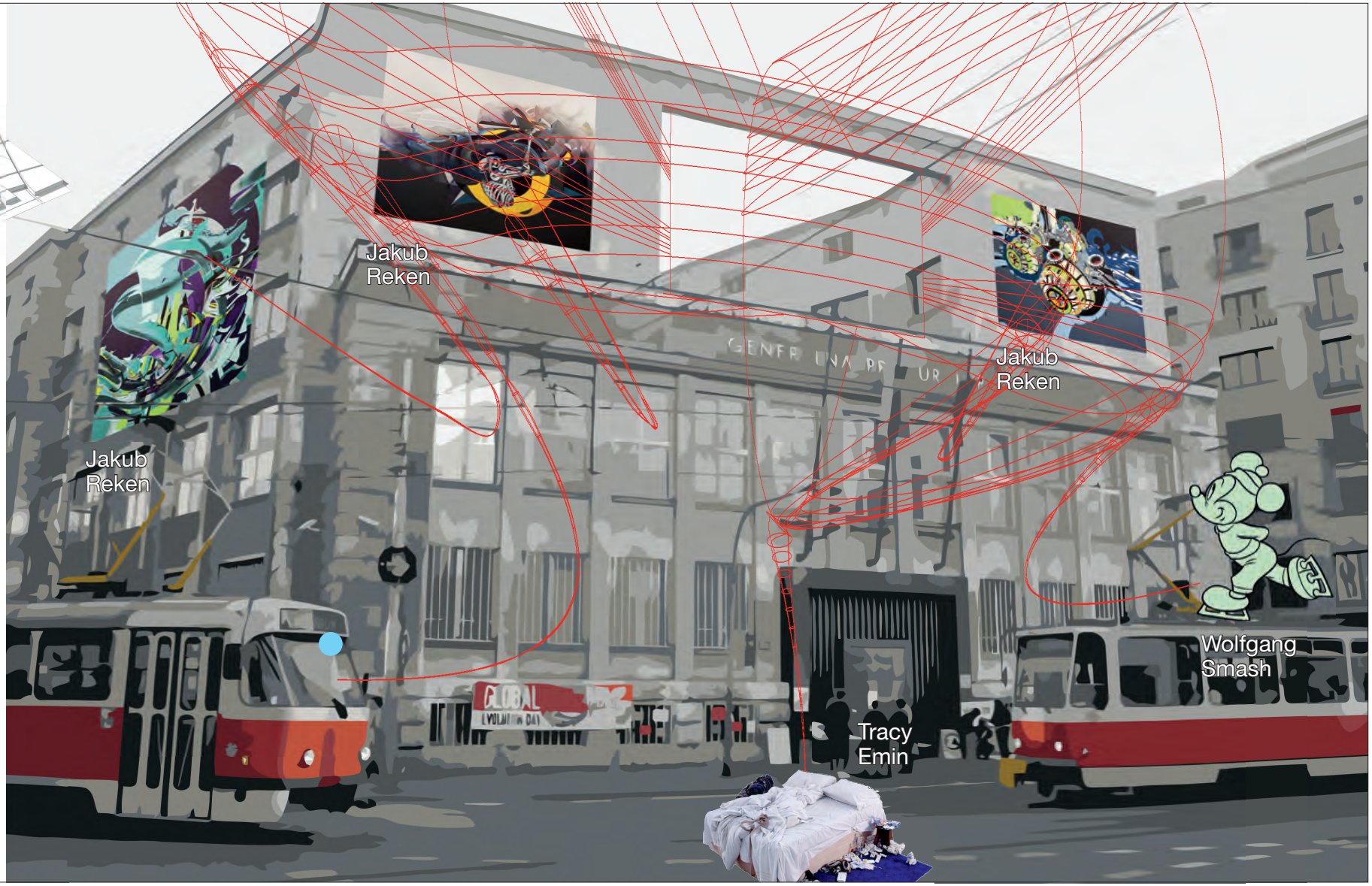
This is a visually appealing example of the gallery generating type called the window. On the Prosecutor's office building there is a strange opening (once there was a statue in there) and the body of the Gallery is growing from it. You can also see how the tracking of cars in the proximity range works here.

SPOT 3

PROSECUTOR'S OFFICE WINDOW / GALLERY

On the corner of Jesenského and Štúrova Str. is the Prosecutor's office. There's a tram driving around this corner. On this example you can see how the Gallery would interact with the tram driver. It also shows how young promising artists (in this case Slovak young artist Jakub Reken, born 1986) can use this platform to display their work to the public just like the already established ones.

Objects do not have to be necessarily glued to a fix surface. As shown here they can be put even on top of a moving tram.



SPOT 1

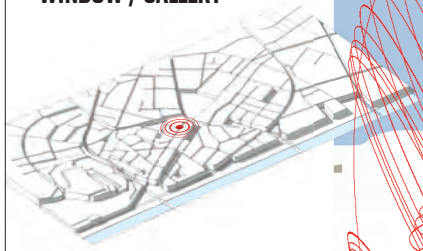
ŠAFÁRIK SQUARE
PIN / GALLERY

There is a small park with a fountain on Šafárik square. This is a perfect spot if you want to relax and take an after lunch break. Sit down on one of the benches and surround yourself with some of the top modern art sculptures.

Here is for example Alexander Calder's sculpture exhibition. Note that the object on display can vary in scale in comparison to the originals.

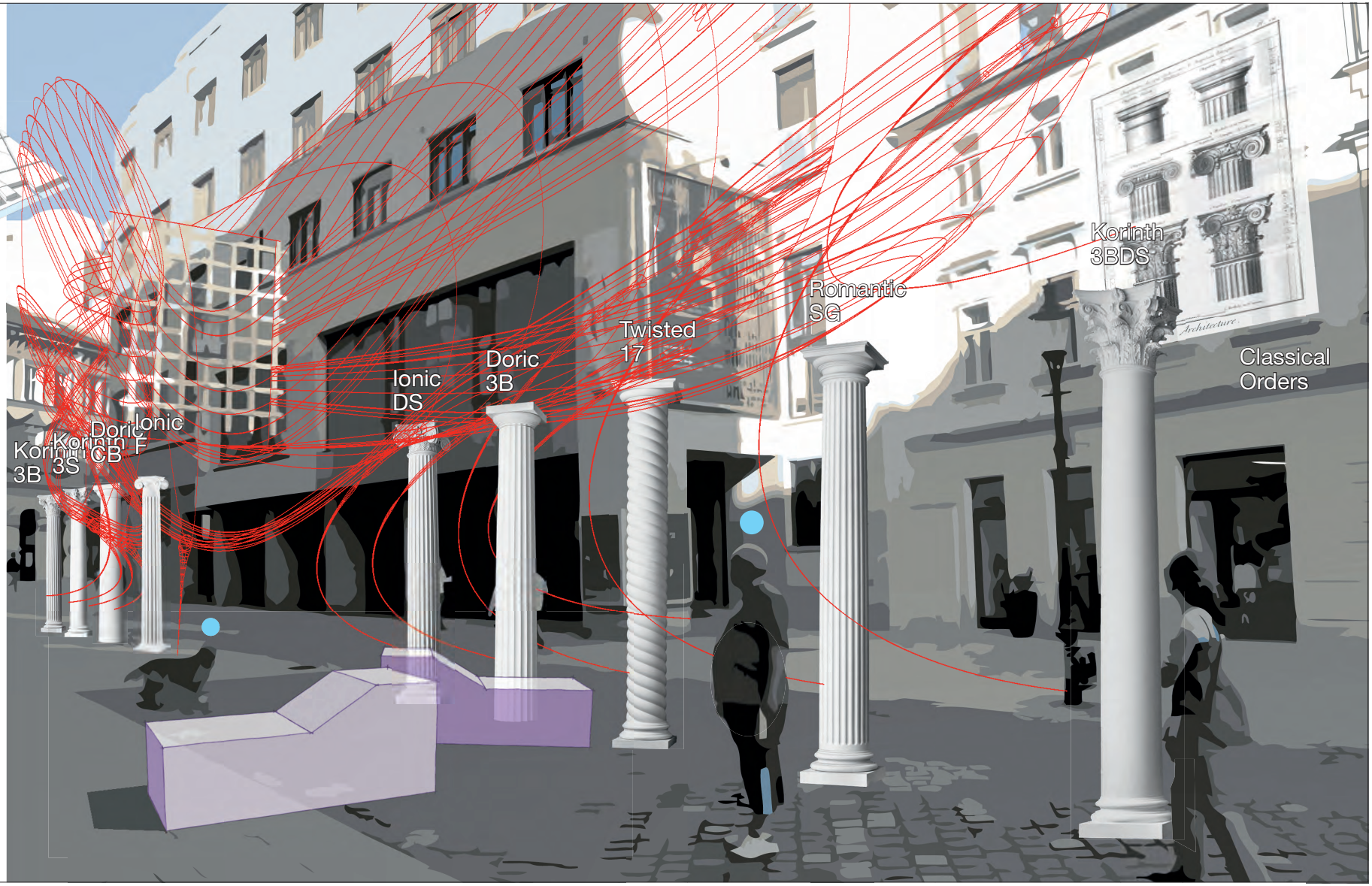
SPOT 4

HVIEZDOSLAV THEATRE WINDOW / GALLERY



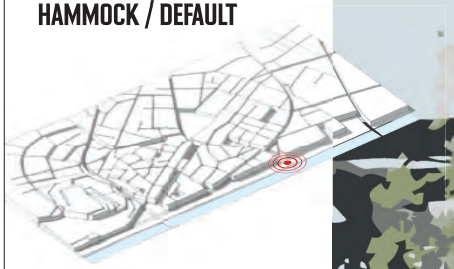
In the old centre of Bratislava there are a few historic pawed streets with a friendly human scale like this example. The building in the middle is the Hviezdoslav theatre. Once an important theatre scene of Czechoslovakia is nowadays struggling to survive. There is an iron lattice of a drop gate hanging in the air in front of the theatre, symbolizing the entrance to the old town of Bratislava. This is where the Body of the Gallery is growing from.

The virtual Gallery itself can be used as an educative platform just like Wikipedia on the Internet. Here you can see it in action, showing a lecture about greek architectural orders. Imagine walking around and through rows of examples of ancient greek columns. This gives a new dimension of learning of history for sure.



SPOT 2

OLD BRIDGE
HAMMOCK / DEFAULT

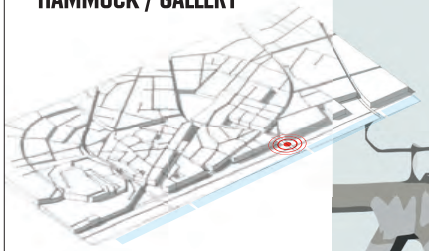


The Old Bridge is the oldest standing bridge in Bratislava. It connects the Old Town with the residential district of Petrzalka. The riverfront of the river Danube is being revitalized as part of a considerable development effort of the town and the private sector. It is here where on sunny days the people stroll along the river. This is the view to the west up the river.

Again an ideal spot for the Gallery hanging down from the bridge structure which is currently being reconstructed. It could be seen from far away as a living parasitic organism underneath the bridge.

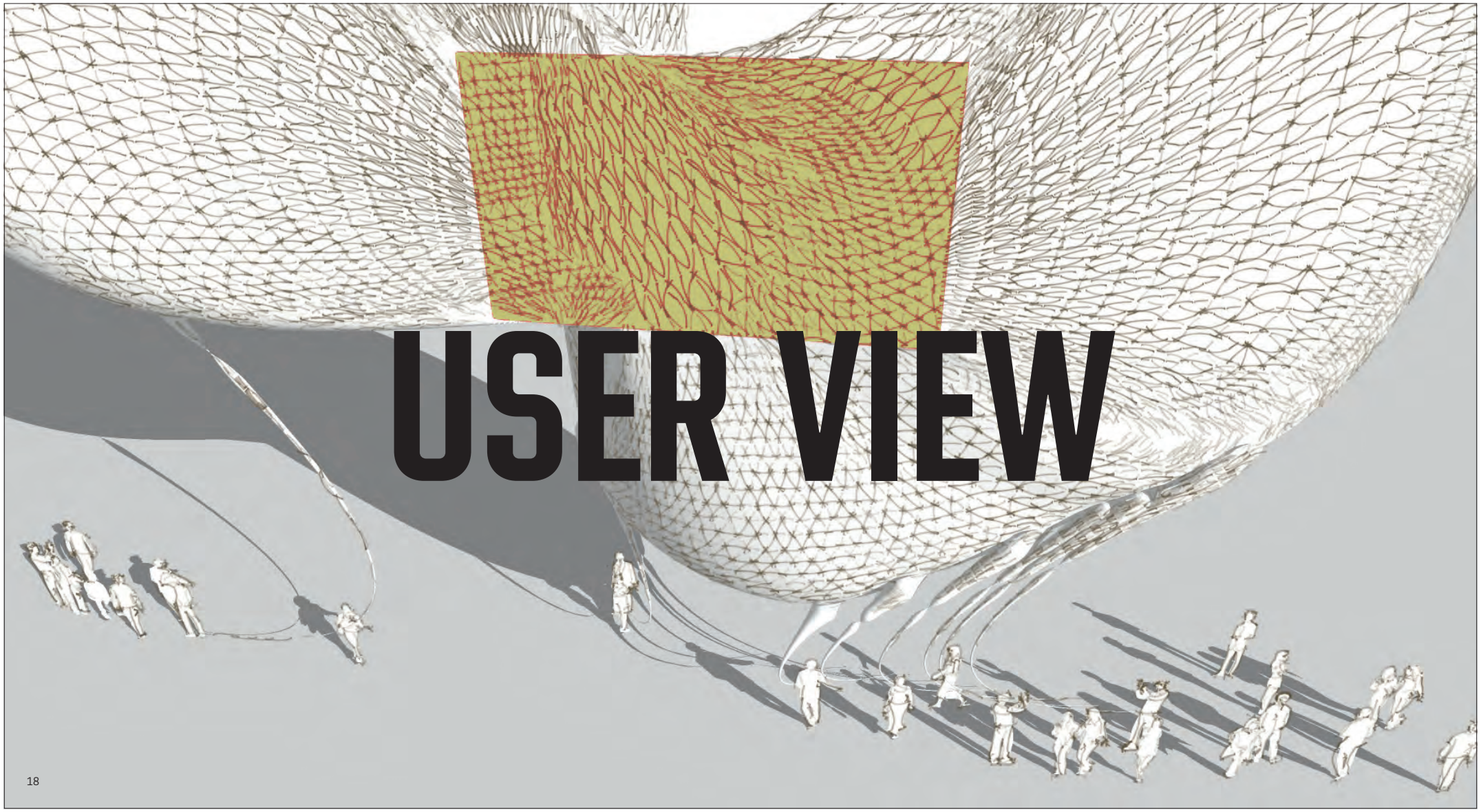
SPOT 2

OLD BRIDGE
HAMMOCK / GALLERY



This is the view of the Old Bridge to the east down the river. In the back the harbor is visible.

The name of this exhibition could be The Sculpture Walk through Time. Here the visitors can compare works of sculptors from different ages and places. This kind of sets could hardly be possible with the real art objects as it would be too expensive or logistically impossible to gather such a collection physically to one place.


USER VIEW

USER VIEW

DEFAULT MODE

Approaching the gallery in default mode. It is reaching out to every moving object in proximity and is also communicating with users on other spots and *web* based users. Those are here shown through their digital representation *avatars*.

Avatar symbols

- Real user
- Web user
- City user (connected from a different spot in the city)

Type 3 (Window)
The Beast is *growing* from a real *window type* object, in this case a billboard.

SAFARIK SQUARE
SPOT # 1 – TYPE 3 (WINDOW)

USER VIEW

DEFAULT MODE

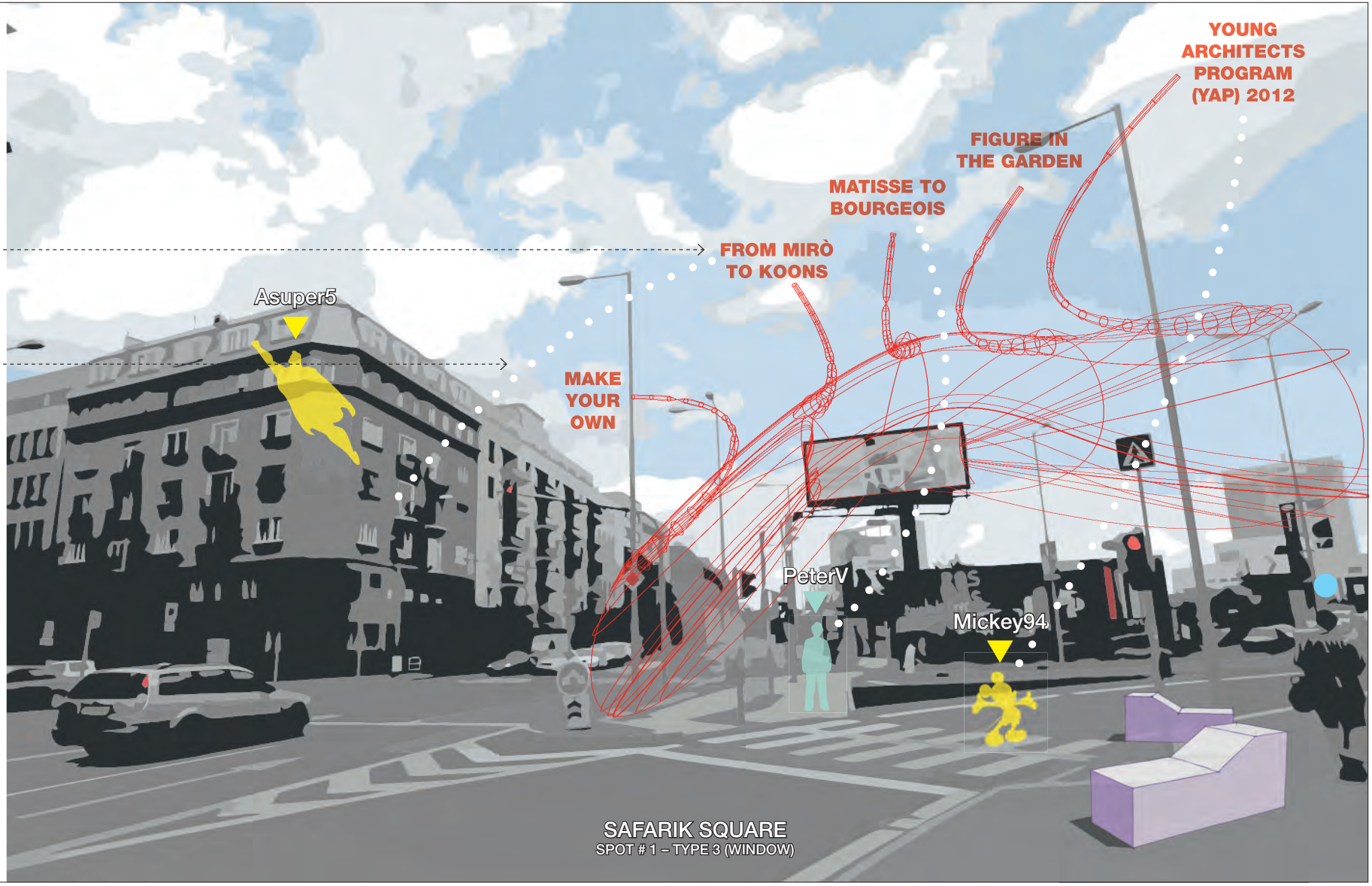
Entering the foyer of the gallery. Other users are still visible.

USER VIEW

GALLERY MODE

Foyer of the gallery showing all the gallery options and exhibitions to choose from. There is a link to the art database too, for creating one's own exhibition.

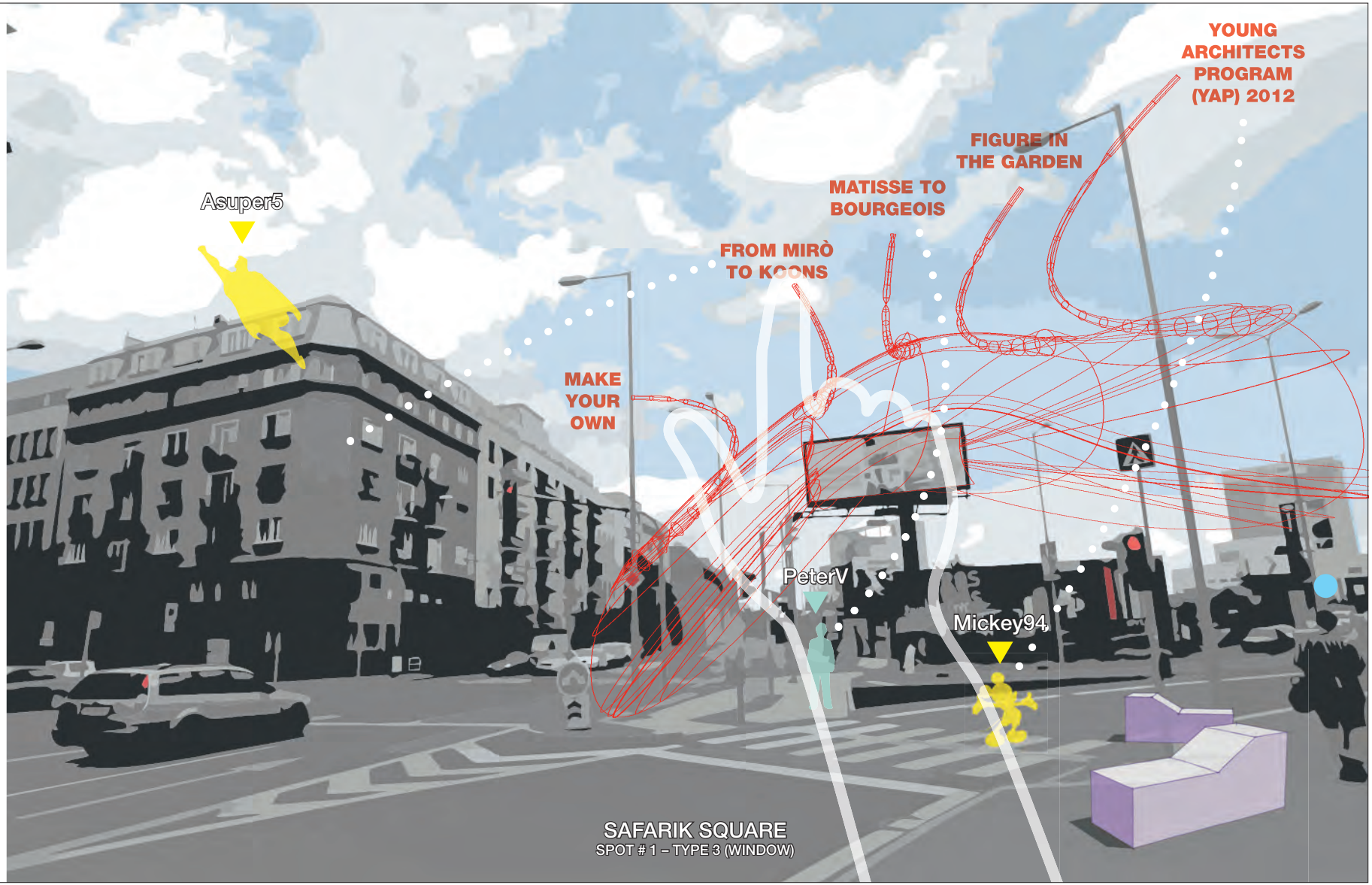
Other users are still visible and the dotted links represent their position in the gallery, which exhibition they are currently viewing.



USER VIEW

GALLERY MODE

Picking the desired exhibition.



USER VIEW
GALLERY MODE

Link back to the foyer.

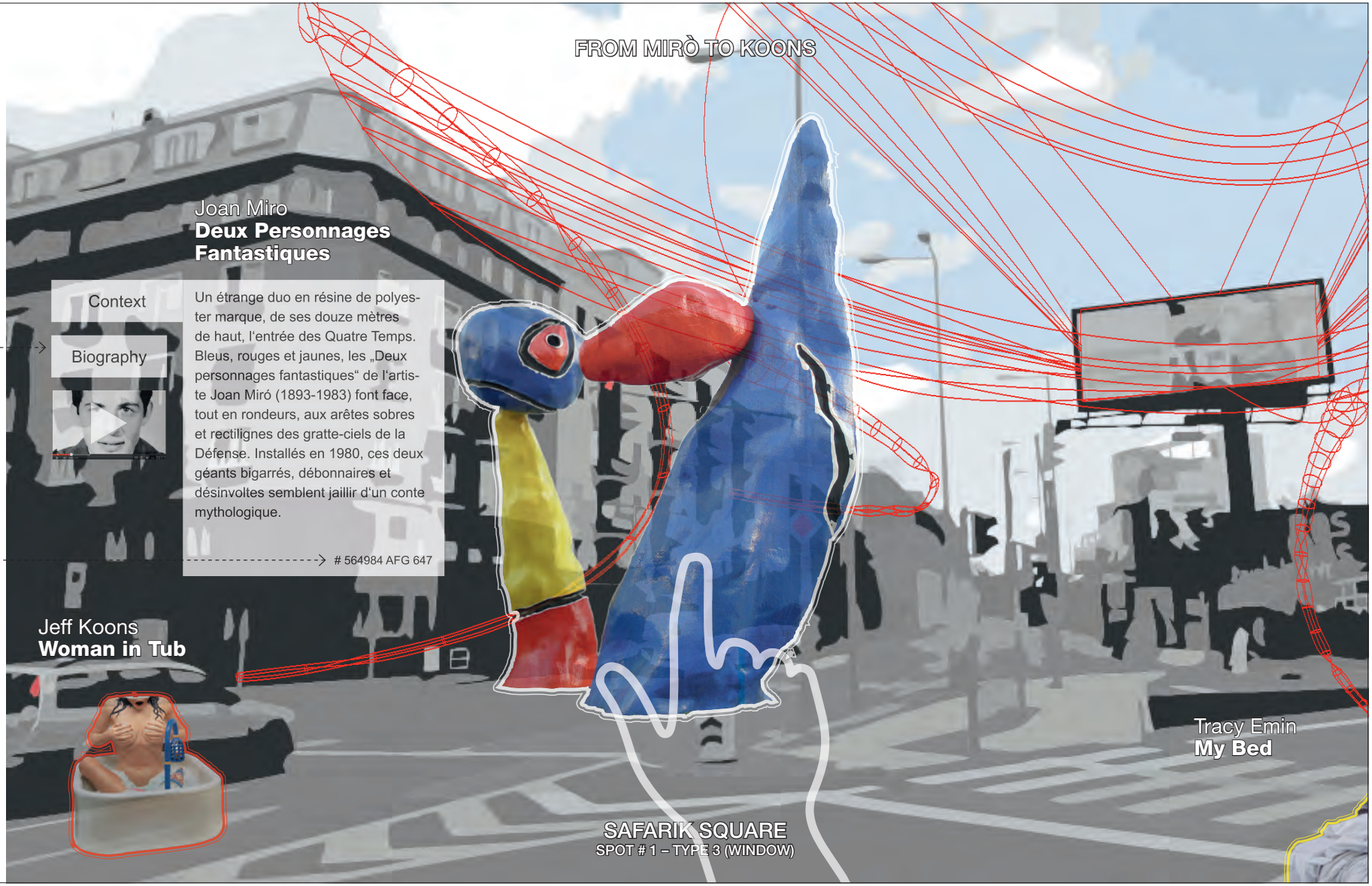
Here the tentacles (connections) represent the links from the gallery to the objects on display.

Every object of art has an author, a title and is highlighted with a red outline once selected it turns white.

USER VIEW
GALLERY MODE

Approaching an object once selected, it's highlight turns white and more content and related info pops out. Texts, video, music, links..

At the bottom of the information box there is a reference number which marks the position of the object in the matrix of the art database. This position has a corresponding position in the real world and the object is displayed on it.



USER VIEW
DEFAULT MODE

Approaching the gallery in default mode and entering the foyer of the gallery.

Type 1 (Hammock)
Growing from the trees in the alley.

USER VIEW

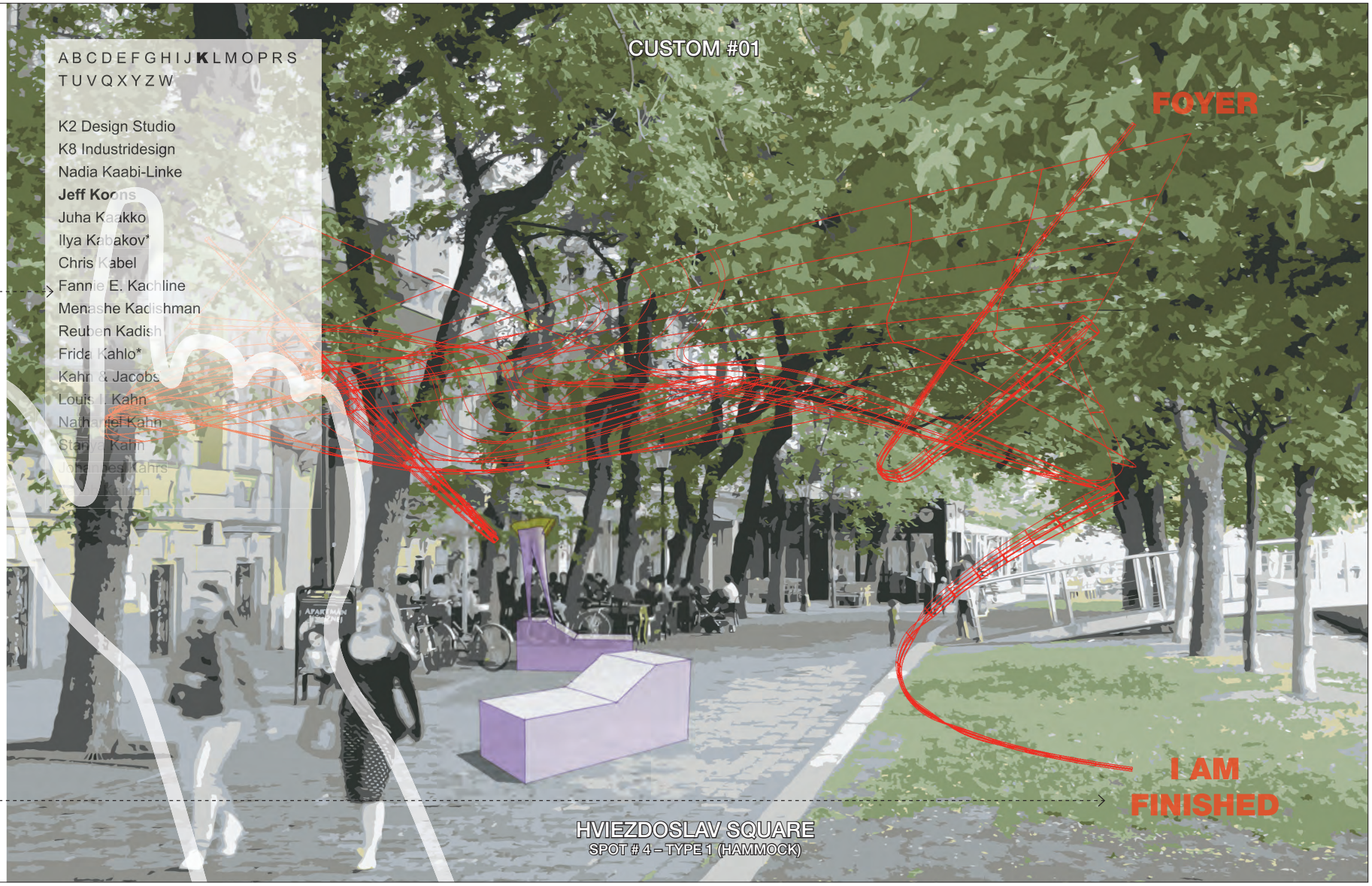
GALLERY MODE

This will show how to design one's own exhibition.

USER VIEW
GALLERY MODE

This will be a private exhibition so there is nobody inside yet. On the left are the authors (database) in alphabetical order which you can choose from.

On the right there is a link back to the foyer of the gallery and on the bottom is a finish and upload button.



USER VIEW
GALLERY MODE

Pictures, video files, audio files.

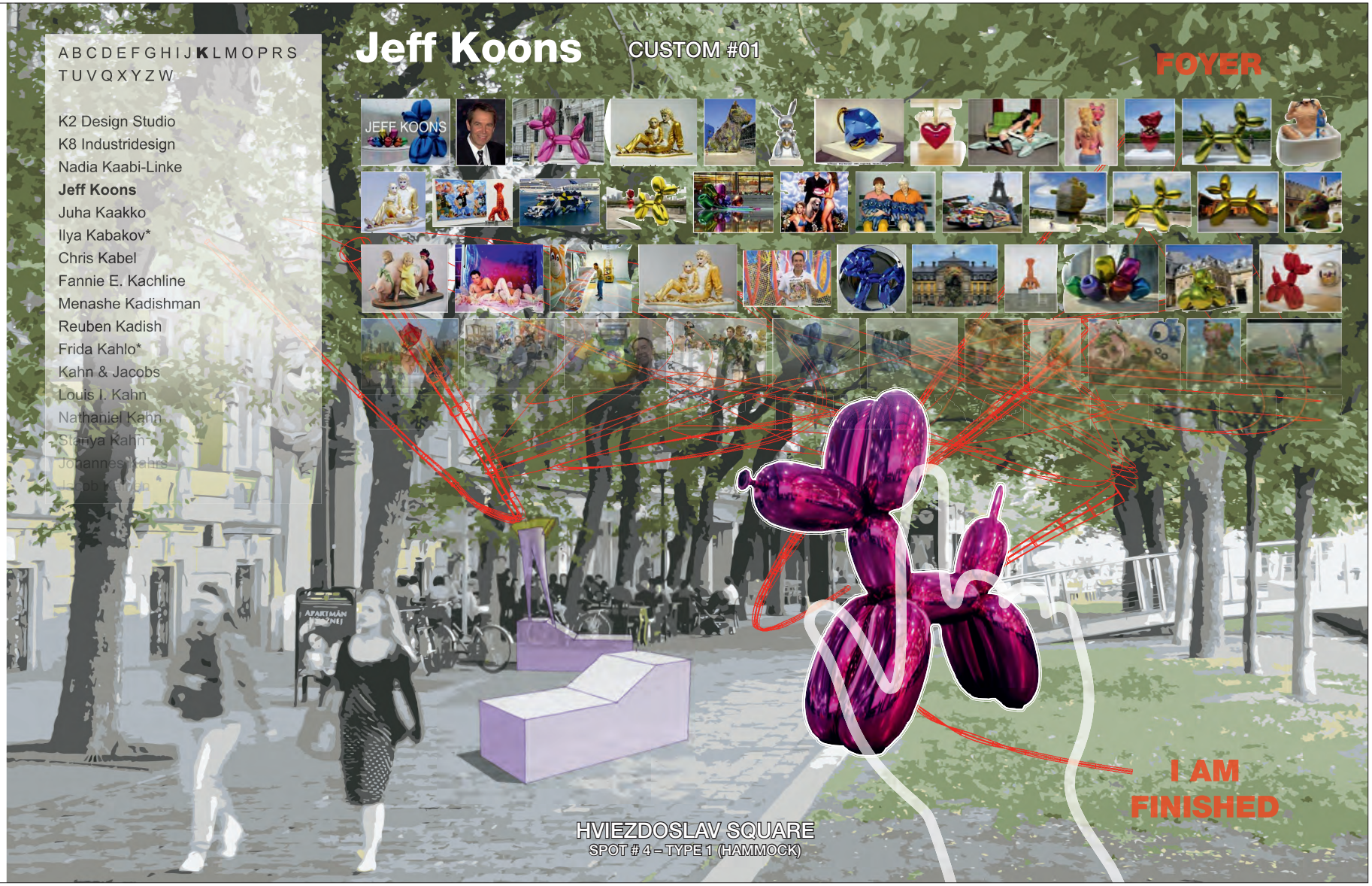
Choosing The Red Balloon dog sculpture.

USER VIEW
GALLERY MODE

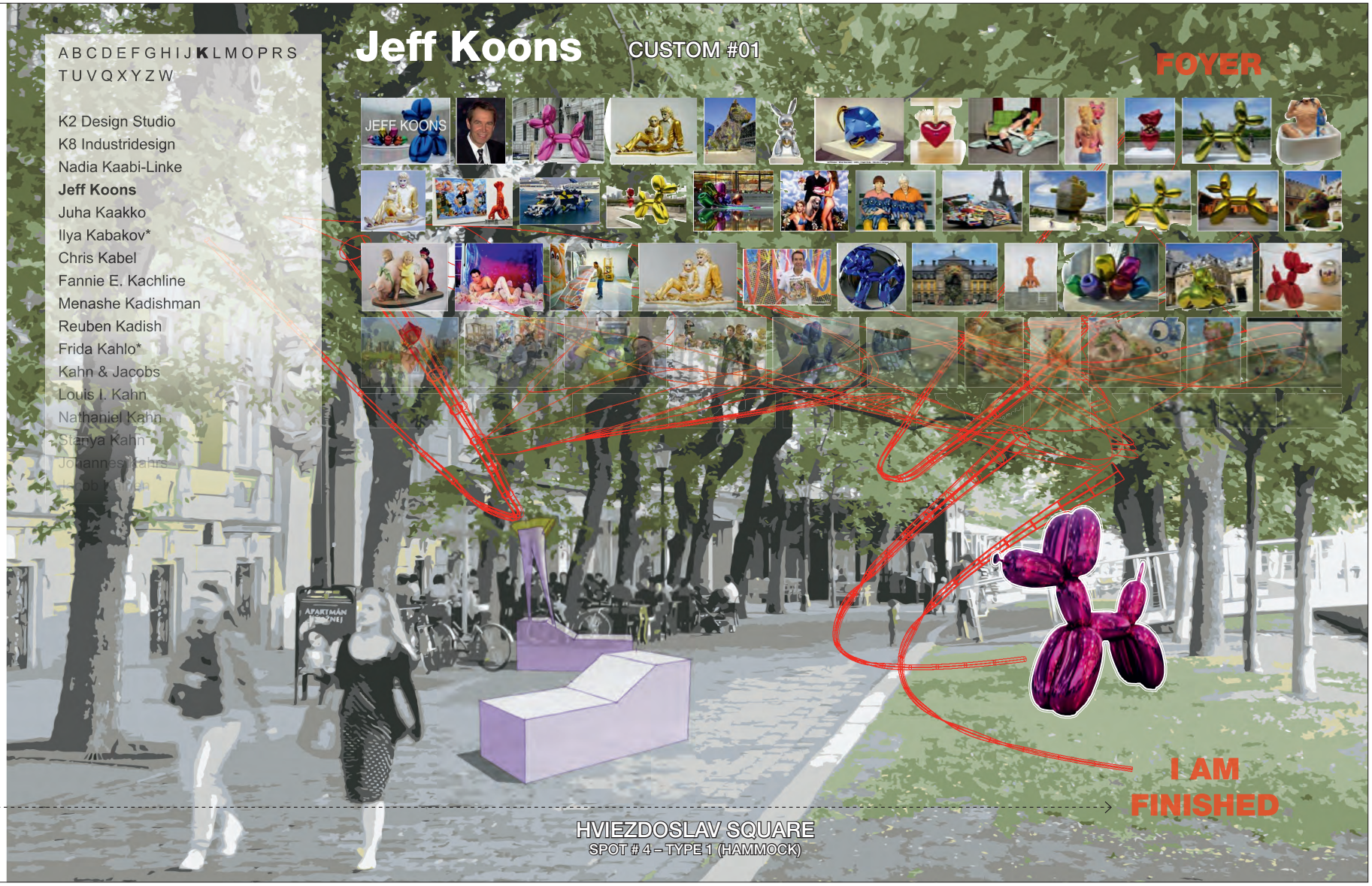
Selected piece of art content pops out.

USER VIEW
GALLERY MODE

Drag and drop it to your desired location.



USER VIEW
GALLERY MODE



Now you can add more objects to the set or by pressing *I am finished* end the session and upload the exhibition for other users (if public) to view.

Book 1 – Analysis of History and Theory regarding cyberspace-architecture and cyberpunk
Book 2 – Vocabulary of cyberspace-architecture and cyberpunk
Book 3 – Notes and sketches
Book 4 – Scripts and codes
Book 5 – Presentation

Cyberpunk 間 Architecture

A humble attempt to give visual expression to new models of reality by Samo Bertok

thesis by
samo bertok / 0026052 / saddie
vienna university of technology

my identities
saddie@seznam.cz
saddiesta@gmail.com
pifko sADDIE / sADDIE_iNSITA
icq 63999844
skype saddiesta
fb samo saddie
gmailchat saddie
msn saddie

identities for social networking research on fb
ronald fergusson / HubBigEnd / ronaldfergusson@seznam.cz
zuzana hanakova / aceFace / zuzahanakova@seznam.cz
rigo johansson / Jimmy Carlyle / rigojohansson@seznam.cz
gustva polke / Odile Richard / gustavpolke@seznam.cz
hildo robertson / Bobby Chombo / hildorobertson@seznam.cz

total 205 pages

Ha ha, this is how science fiction was imagined in 2008.

Maroš Krivý, Café Kiasma, Helsinki, 30th November 2008