Time Window

An Interactive Exhibit to Explore Multi-Layer Historical Maps

Master Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Georgia-Anna Farmaki

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Wolfgang Prinz

Registration date: Apr 04th, 2006
Submission date: Oct 02nd, 2006

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, October 02nd, 2006

Contents

List of Figures

Abstract

Time Window is a system for interacting with maps of Regensburg, a medieval city in Germany. It features both temporal and spatial navigation, allowing the user to explore urban changes that have occurred at various places in the city and throughout history; the user can also retrieve further multimedia information about historical buildings and sites. *Time Window* will be installed as an interactive exhibit at the Regensburg Experience (REX) visitor centre.

The information about changes in a city can be seen in maps of the city from different time periods, and *Time Window* aims to support simultaneous spatio-temporal navigation of such map collections. After experimenting with different interaction techniques of spatio-temporal navigation, a "digging" metaphor was chosen, similar to an archaeological excavation. Analogous to how an archaeologist gradually uncovers past artifacts of a city by digging at specific sites, users in *Time Window* gradually reveal past maps of Regensburg at the parts of the touchscreen where they touch, just like digging through sand.

The thesis was developed at the Media Computing Group at RWTH Aachen University, in cooperation with Regensburg Experience GmbH, the Regensburg Land Surveying Office and the Department of Records and Preservation of Historical Monuments of Regensburg city.

Überblick

Time Window ist ein Interface um in Regensburg, einer deutschen, mittelalterlichen Stadt, mit Zeit und Raum zu interagieren. Das System bietet eine neuartige Methode zur Navigation durch die Geschichte der Stadt: Es ermöglicht die Erforschung der städtischen Entwicklung an einem bestimmten Ort durch die Jahrhunderte und die Betrachtung weiterer multimedialer Informationen über historische Gebäude und Stätten. Es wird im Regensburg Experience (REX) Besucherzentrum als ein interaktives Exponat zu sehen sein.

Während historische Karten aus verschiedenen Zeitperioden allgemeine Veränderungen einer Stadt repräsentieren, zielt *Time Window* auf die gleichzeitige räumlich-temporale Kartennavigation ab. Nach Experimenten mit verschiedenen Interaktionstechniken wurde letztendlich eine "Ausgrabungs"- Metapher gewählt, ähnlich wie in einer archäologischen Ausgrabungsstätte. Der Anwender in *Time Window* deckt dort, wo er den berührungsempfindlichen Bildschirm anfasst, nach und nach ältere Karten von Regensburg auf, als ob er mit dem Finger im Sand buddeln wurde - ähnlich wie ein Archäologe, der Schicht für Schicht nach Artefakten an bestimmten Stätten gräbt.

Diese Diplomarbeit wurde von der Media Computing Group, RWTH Aachen, mit Unterstützung der Regensburg Experience GmbH, des Amts für Vermessung und des Amts für Archiv und Denkmalpflege der Stadt Regensburg entwickelt.

Acknowledgements

I would very much like to thank Prof. Dr. Jan Borchers who gave me the motivation and the opportunity to pursue this interesting subject and who was always open to questions and made useful suggestions for the improvement of the interaction from a user point of view.

I would also like to thank Prof. Dr. Wolfgang Prinz for the helpful background literature that he provided me.

I would especially like to thank my supervisor Eric Lee, who gave his time to guide me, his useful advice and helped me throughout the development of the project. He was always open to questions and he reviewed the drafts of my thesis.

I particularly wish to acknowledge the generous cooperation I received from Dr. Julien Biere from Regensburg Experience and Mr. Lenz from the Regensburg Land Surveying Office. Dr. Julien Biere gave me directions and requirements about the exhibit in the weekly video conferences and useful material. Special thanks to Mr. Lenz Regensburg Land Surveying Office for providing me with maps and other multimedia material.

Many individuals were of tremendous help, notably my friend Nikos Gkoumas and my colleagues Eugen Yu and Rene Reiners, who reviewed parts of my thesis and gave me their suggestions. Appreciation goes also to all those individuals who assisted me in any way in the different stages of this work.

Thank you!

Chapter 1

Introduction

We live in cities that are in a constant state of metamorphosis, due to continuous growth, development, historical events and other reasons. New buildings are constructed or replace older ones, new roads and railways are built and the architectural style of all these buildings changes as well. During our lives we witness a small portion of these changes. But what can one say about what a city was like, as long as 1000 years ago? On a personal level, it is often magical and exciting to imagine the state of a city hundreds or even thousands of years ago, roaming back in time with the mind. It is even more fascinating to try to imagine what stood at a specific place, for example one's house or neighbourhood, 100, 200 or even 1000 years ago. From an academic point of view, the history of a specific place, for instance a famous or historical building, a temple, a church or a bridge, is of great importance for students of subjects like architecture or urban planning.

The face of a city changes throughout the centuries

Things one may wonder are: have these buildings developed, or have they remained the same? Has their architecture changed over the ages? How long have they existed and what was there before them? Do other buildings from a different historical era predate these buildings? Although one cannot travel back in time and live in the past, he has at his disposal historical documentation. The most obvious and visual form of documentation for cities are maps from different time periods, although some informa-

Maps of different time periods show the overall changes of a city tion from these maps may be missing. Maps of different time periods depict the overall changes that have occurred in a city or region in a visually clear and understandable way. But for specific places of interest, it can be difficult to pinpoint the same location on two different maps and visualize what changes have occurred to them, even if the maps are overlaid on top of each other.

Purpose is to explore how a city changed at specific places only

Maps are static, two-dimensional representations of the three-dimensional space of a city or a region. For the purpose of exploring the changing face of a city, a fourth dimension, namely the dimension of time, can be called into play. This is the time period to which the map corresponds. Our goal with Time Window is to design an interactive exhibit to navigate along the time axis of maps, moving from recent to older times at a specific spot, as well as to provide a way to visualize the changes which have occurred in a city. In that context, this thesis aims at exploring techniques of simultaneous "spatio-temporal" navigation and implementing a suitable method of interaction that provides an exciting and interesting experience. We believe this will lead to a better understanding of a city's history.

An archaeologist uncovers older archaeological layers of different times

To explore what once existed at a certain place and at a certain time, we used an approach analogous to the way that an archaeologist reveals the past of a city and discovers its history. As it happens at archaeological excavations, the artifacts from more recent cultures will lie above those from more ancient cultures. Therefore, an archaeologist, by digging at a specific site, gradually uncovers older archaeological layers, which correspond to different time periods, consisting of past buildings and artifacts.

"Digging" metaphor used similar to that of the archaeologist

We have used an interaction metaphor similar to that of an archaeological dig in order to visualize the changes that occurred at a specific spot over time. The deeper one digs, the older civilizations and periods one discovers.

Users dig through Regensburg's history on a touchscreen

Our system, Time Window, allows users to dig into the history (time) of Regensburg through the millenia, by interacting with a touchscreen which displays a map of Regensburg at present time. By moving their finger analogously to the motion of digging in sand, the users dig further and gradually reveal the underlying maps of Regensburg from older time periods (maps stacked on top of each other, moving from recent to older time periods, as depicted in Figure 1.1). However, any other movement of the finger than that of digging, allows the user to explore the map of the same time period. This "digging" metaphor enables users to simultaneously explore the space and time dimensions of Regensburg city as well as to explore the urban changes that have occurred in the city and at specific places.

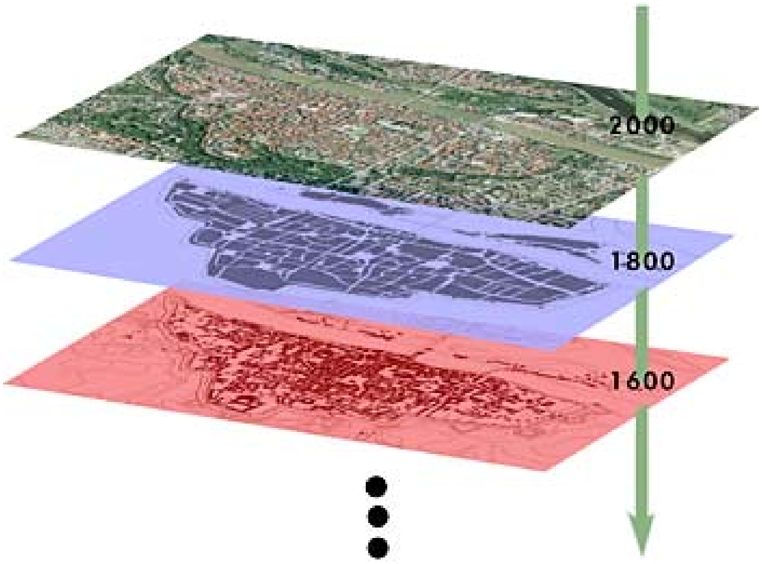

Figure 1.1: The maps of Regensburg from different time periods, stacked on top of each other, moving from recent to older time periods. These maps are later referred to as "time layers".

Additional information about historical places by tapping on diamonds

Furthermore, the system gives the opportunity to explore further multimedia information about historical buildings and places of interest. Every map of a particular time period has some special "hot spots" or landmarks, indicated by diamonds. The purpose of diamonds is to attract the attention of users and they convey that there is something interesting and valuable at that place to be discovered. When the user taps on one of them with his finger, an image or a short video clip of the building or the particular place zooms in.

1.1 The REX project

REX is a visitor centre for presenting Regensburg's history in an interactive way

The system was designed for the REX (Regensburg Experience) high-tech visitor centre, expected to open in fall 2006. REX aims at presenting Regensburg's history, present and future, in an interactive and interesting way, therefore adding an element of innovation to the medieval city. The presence of this technologically advanced museum adds to the prestige of the historical and architectural tradition of the town. REX includes a series of exhibits, providing an ideal blend of traditional and innovative media. A part of these exhibits are the interactive exhibits produced by the Media Computing Group[1] at RWTH Aachen University. Time Window is one of them. Other exhibits are REXPlorer[2] , REX Band[3] and Minnesang[4] .

Regensburg is a medieval city rich in history in Southern Germany

The scope of REX is not only limited to the inside of the museum space, at the renovated 1620 salt storehouse "Salzstadel" (see Figure 1.2), but extends to two other domains, the urban space of the city itself, with the REXplorer urban mobile game, and a web space, with an informative internet portal. Regensburg is a medieval city located in Bavaria, in Southern Germany. Regensburg has a rich history and culture, spanning over two millenia. It is unique in being Germany's most famous town for having a wide range of well-preserved buildings and monuments and is likely to be nominated for a UNESCO World Heritage Site in 2006. The city's history and architecture have been widely influenced by the existence of the Danube river which passes through the city.

1.2 Interactive exhibits and their role in learning

Interactive exhibits promote learning

Since our exhibit is going to be used in a museum setting, it

[1] http://media.informatik.rwth-aachen.de
[2] http://media.informatik.rwth-aachen.de/rexplorer.html
[3] http://www.erlebnismuseum-regensburg.de/im-salzstadel/erleben/rex-band/
[4] http://www.rex-regensburg.de/im-salzstadel/wissen/minnesang/

Figure 1.2: The old salt house "Salzstadel" in Regensburg, is the museum space of the REX visitor centre where Time Window will be located, as one of the exhibits.

should deliver an educational message, teach something to the visitors about the history of Regensburg and it should have a recreational purpose as well.

People learn best by doing

Based on the theory of constructivism (Dewey [1966] and Piaget [1973]) people learn best by doing, through activity. Knowledge and understanding is gradually built-up, step by step, through active involvement and the interaction with the physical and the social environment. Especially children, who are also included in our target group, develop cognitive structures through action, spontaneous activity and play (Piaget [1973]).

There are several references stated in Roussou [2004], about

Definitions of interactivity and its role in learning

the definition of interactivity and its role in learning, some of which are as follows. In the context of public exhibits, Adams and Moussouri [2002] define the interactive experience as that which can actively involve the visitor physically, intellectually, emotionally and socially. Ryan [2001] claims that an interactive medium opens its world after the user has made significant intellectual and emotional investment. Artists who have explored interactivity in their digital installations define the interactive experience as an active form of engagement (Rokeby [1998]). Barker [1994] considers interactivity in learning as a "necessary and fundamental mechanism for knowledge acquisition and the development of both cognitive and physical skills". Amthor [1992] arguments that people retain about 20% of what they hear, 40% of what they see and hear and 76% of what they see, hear and do.

Interactivity in museums promotes learning

Therefore, interactivity in the context of systems for public use in museums promotes learning. The challenge is to provide the ideal blend of educational and recreational value. Our exhibit is interactive as it has the characteristics of interactivity as defined above, and lets people get actively engaged by discovering history by themselves, and build their own individual experience based on their personal interests and preferences.

1.3 Thesis structure

The following overview presents the further structure of the thesis.

Chapter 2, Related work, describes systems relevant to Time Window. These are interactive exhibits about the spatial navigation of maps and systems for the spatio-temporal navigation based upon video material.

In Chapter 3, Requirements and scope, we will discuss two different requirements of the system, in terms of the functional and the technical aspects of our system.

Chapter 4, Design, analyzes the design goals and strategies we followed when designing the system. It includes a usage scenario and presents the different techniques that were explored for the temporal interaction of the multi-layer maps.

Chapter 5, REX Preview prototype, describes the technologies used for the system and the subset of features that were implemented for the first prototype presented at the REX Preview. In the end of the chapter, an intermediate evaluation is presented based on the visitors' feedback.

Chapter 6, Second prototype, deals with the software design of the fully functional system. It describes the Time Window classes and how the application was designed to include an arbitrary number of time layers and diamonds.

Chapter 7, Summary and future work, summarizes the contributions and explains what features need to exist in future versions of the system as well as possible ideas and suggestions for future research.

Chapter 2

Related work

There are two categories of systems that are relevant to Time Window. Firstly, we have systems or exhibits for navigating and interacting with maps and, of particular interest to us, the spatial navigation of maps. There are many existing systems for navigating maps collected from satellite images or aerial photographs, such as Google Earth[1] . However, only few of these systems, if any, support navigating through the time dimension in addition to the usual spatial dimensions. In this category of systems, we will examine the Swissarena interactive exhibit.

Few systems support temporal navigation of maps

Secondly, there are systems for simultaneously exploring time and space. Spatio-temporal navigation is a thoroughly researched area in digital art. These systems do not use maps as their source material, but they manipulate video sequences since these are good examples of spatio-temporal volumes (cf. Levin [2006]). The systems we will examine are the Khronos Projector, Asynchrony, Liquid Times and the Invisible Shape of Things Past.

Systems for spatio-temporal navigation of video sequences

2.1 Interactive exhibits about maps

This section examines interactive exhibits or systems for

Map navigation

[1] http://earth.google.com/

navigating through the spatial dimension of maps. The time dimension is not incorporated into these systems.

2.1.1 Swissarena

Swissarena (cf. Swi) is an interactive exhibit at the Swiss Transport Museum at Lucerne in Switzerland.

Swissarena is a walkable aerial photo of Switzerland

In this exhibit, an aerial photograph of Switzerland on a scale of 1:20,000 is displayed on a surface of a total of 200 square meters (see Figure 2.1). Visitors can walk or sit on the map and can make use of a large illuminated magnifying glass to view parts of the map in greater detail at the scale of individual houses and buildings. The map was composed from 7,800 individual shots and is the largest orthophotograph of Switzerland from the air.

Physical interface for map spatial navigation

Swissarena is a purely physical interface for navigating through the spatial dimension of the map and does not make use at all of technology. This is a major difference with Time Window, where technology plays the most important role in the exhibit and its realization, since all of the provided information is digital, and the input device of the touchscreen is based on technology as well.

Physical interaction by exploring a country "on foot"

The user can interact with the physical object of the magnifying lens, in order to view parts of the map at a higher resolution, providing a natural metaphor for the task of zooming. Users wander through the physical surface of the map, exploring it as one would explore a city "on foot". The eyelevel of an average adult standing over the photograph is equivalent to an airplane view at an altitude of about 36 kilometres above the earth, providing a real-world spatial analogy.

Only spatial navigation and no temporal

Another important difference of Swissarena with relevance to Time Window is its inability to support temporal navigation, as it focuses uniquely on the space dimension. The map of Switzerland is static and anchored in present time, and one cannot navigate through the past maps of Switzerland.

Figure 2.1: Visitors explore the map of Switzerland on a scale of 1:20,000 "on foot" at the Swissarena exhibit, walking or sitting on the map and wearing red slippers provided by the museum (cf. Swi). They can also use an illuminated magnifying lens in order to view parts of the map at a greater detail.

Recreational and educational value

The role of both interactive exhibits is recreational as well as educational. Both Swissarena and Time Window let visitors actively discover places, buildings, cities, and landscapes on their own in a playful manner. Swissarena lets people discover geographical information, whereas in Time Window, in addition to geography, the users can explore historical information about landmarks and uncover the urban changes of a city. Therefore, both exhibits offer a deeper understanding of the topic they address, geography and history respectively.

Map representation not constrained by device size

The physical representation of the map on the ground surface in the Swissarena exhibit is not constrained by the particular device space, such as the size of the available touchscreen, like in the case of the Time Window exhibit.

2.2 Spatio-temporal navigation

Time-space navigation of video

The following systems are examples of time-space navigation. These systems are primarily designed for film sequences. Video is often used because it is a readily available data source that contains both the space and the time dimension.

Topic in painting

The bold attempt to represent time in relation to space has already been a topic of painting and art, which are normally restricted to represent only space and are thus static in nature, as described by Jaschko [2003]. Time-space navigation has culminated in the work of the cubists and futurists of the last century, who have tried to represent a body at all possible stages of its movement or from many different perspectives at the same time.

Film is a space-time volume

The medium of film , as a form of art, integrates by its nature the representation of time along with space into a single, expressive medium. The space dimension is the place where the film takes place and the time dimension is the time span which the film represents. The usual way of visualizing video content consists of showing consecutively each image of the sequence.

Spatio-temporal navigation of film creates interesting visual effects

The simplest form of temporal navigation is to simply play the video sequence forwards and backwards linearly. Combining temporal navigation with spatial navigation, however, creates interesting aesthetic effects that are the topic of many contemporary interactive art projects. One example of temporal navigation is affecting only parts of the space through time, while the rest of the space remains unaffected, or navigating both along the time and the space axis. In the following sections we will examine some of the most important of these systems, the Khronos Projector, The Liquid Times Series, Asynchrony, and lastly the Invisible Shape of Things Past.

2.2.1 Khronos Projector

The *Khronos Projector* was created by Alvaro Cassinelli, with the support of Takahito Ito, Monica Bressaglia and Masatoshi Ishikawa at the University of Tokyo in 2005 (cf. Cassinelli [2005]).

In Khronos Projector, users touch a large tissue-based deformable screen that displays a video, and can send the parts of this video at the position where they touch forward or backwards in time. Users can also shake or curle the deformable screen, thus creating “islands of time” or “temporal waves”. The idea behind Khronos Projector is the transformation of a video sequence into a space-time sculpture that the users can “sculpt” with their own hands (see Figure 2.2). Users can interact with the screen with other ways as well, such as a light beamer. The program, which is extremely computationally expensive, was first prototyped in Matlab and then implemented in C++ using OpenGL.

Users send back in time the parts of the screen where they touch

Khronos Projector uses a tissue-based deformable projection screen made from a thin, elastic fabric. Its surface is then scanned in real-time by a vision chip, in order to extract its depth characteristics. The resulting image is in turn projected back to the screen using a standard LCD projector (see Figure 2.3).

Deformable screen, vision chip and projector

The purpose of Khronos Projector is aesthetic, intended for exploring the interesting, otherwise impossible, visual effects of spatio-temporal navigation. On the other hand, the purpose of Time Window is educational. Khronos Projector is video-oriented whereas Time Window’s aim is the exploration of maps. Furthermore, Khronos Projector experiments with a wide variety of different interaction metaphors for travelling in time e.g., painting with time, time ripples, time “punch”, time tunneling, chromatic time arrow, etc. The digging interaction metaphor used in Time Window is most suitable for the context of the application, which is map oriented.

Aesthetic purpose: creating interesting visual effects

Both projects use the sense of touch for the time navigation. Khronos Projector gives a stronger sense of touch as the user can really push the screen deeper -and therefore the time dimension- and receives tactile feedback. The deformable screen has the advantage over a touchscreen that it can sense multiple levels of pressure, thus measuring the amount of temporal pressure.

Sense of touch in both projects

At its initial state, before the user has started the interaction, the projection screen in Khronos Projector displays the first image frame of the video. This is similar to Time Window

Calm initial state in both projects

whose initial state displays a map of Regensburg at present day. A last common point with Time Window is that in Khronos Projector it is not possible to "push" deeper the whole image at once in order to send it forward or backwards in time. In the same way, in Time Window it is intentionally not easy to view overall changes of the city.

Figure 2.2: A "Space-Time" sculpture in Khronos Projector. The user , by touching a projection screen can send parts of the displayed video backwards or forwards in time (cf. Cassinelli [2005]). It is as if the user "sculpts" the video with his own hands. The resulting effects are visually interesting.

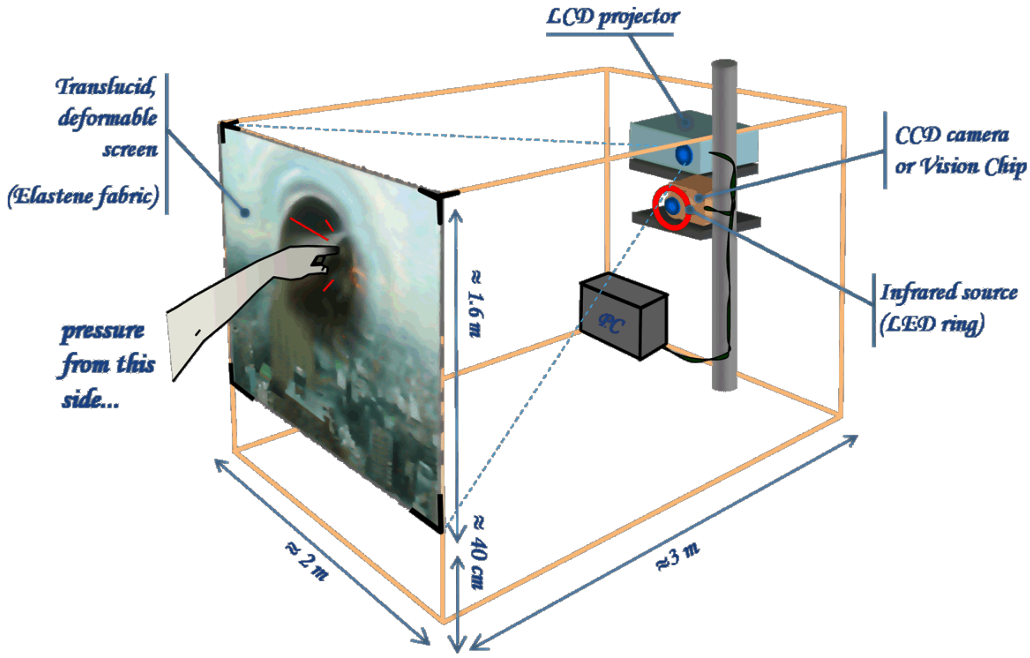

Figure 2.3: The installation setup of Khronos Projector. It consists of a tissue-based deformable screen, a vision chip or a camera for scanning the surface of the screen and extracting its depth characteristics, and a standard LCD projector for projecting the resulting image back to the screen (cf. Cassinelli [2005]).

2.2.2 Asynchrony

Asynchrony is a project created by James Seo in 2006 (Seo [2006]). It enables users to view simultaneously multiple points in time within a video sequence. The user, with the aid of the mouse, can specify regions within the video, for which he creates temporal displacements. These regions range from rectangular regions to amorphous areas that assign a different time-displacement to each pixel. The user can then shift these areas forward or backwards in time by simply clicking on them with the mouse. The resulting composite image, consisting of the time-shifted regions within the shared visual space, is rendered in real-time. The effects are similar to that of the Khronos Projector. Asynchrony was implemented in *"Processing"*[2] , an open source programming language and environment to program images, animations, and sound.

Simultaneous visualization of multiple points in time within a video

[2]http://www.processing.org/

Four modes to define regions

There are four different pre-defined modes to define regions within the rectangular frame of the video, named by Seo [2006]: “Slide”, “Smudge”, “Split”, and “Splotch”. Seo

Figure 2.4: A “time ripple” created with the Smudge mode in Asynchrony (cf. Seo [2006]). A video of the progress of a day and night in a city is used as the material. Time shifts forward at the point where the user presses with the mouse within the certain radius of the ripple.

[2006] describes the Smudge mode, amongst the four different modes: “Smudge: The user can make time ripple and shift forward in different areas using the mouse. Time is shifted within a radius; the amount of time shifted depends on the distance from the mouse location.” (see Figure 2.4).

Artistic purpose

The purpose of Asynchrony is mainly artistic, experimenting with different techniques of visualizing multiple points in time in a single, multi-time, multi-layer image.

Asynchrony for short-term changes, Time Window for long-term

Both Time Window and Asynchrony deal with exploring changes occurring in linear time. However, the main target of Time Window is to explore long-term changes of the time span of millennia that occur in a city. This is in contrast with Asynchrony where one can explore the progress of a day or a night within a city, and in general short events of the relatively restricted time span of a video, for example people dancing, a person moving his head, driving, etc. Time Window, though, uses a limited number of distinct time periods, therefore distinct points in time, while Asynchrony uses continuous time, and, in contrast, an unlimited number of time points. In Asynchrony time flows continuously, while in Time Window time is divided in distinct time periods.

Asynchrony implements a method of moving parts of the screen in time. Multiple modes are used to achieve the same. Out of these, the Smudge mode has most in common with Time Window, in that it affects only a small region of the screen and shifts this back in time, while the other parts remain unchanged. In the case of Time Window, this region is defined by the finger, whereas, in Asynchrony, the same is defined as the radius of a time ripple.

Both projects affect only small part of the screen

The effect of the "time ripple" is based on the natural phenomenon of a ripple in the water, where the circle with the bigger radius is equivalent to the oldest event, so time flows towards the biggest circle. In the same way in Asynchrony, when ones moves forward in time within the radius of the ripple, time flows faster at the location of the mouse with the smaller ripple circle, and slower towards the larger circles at a bigger distance from the mouse location. This metaphor of the "time ripple" gives an intuitive feel of moving forward into time. Time Window uses a completely different metaphor for moving backwards in time, the digging metaphor, which is more suitable for the context of our application of unravelling the past urban development phases of a city.

"Time ripple" a natural metaphor, like a water ripple

2.2.3 Liquid Times

The *Liquid Times Series* interactive art installation (Utterback [2001]) was created by Camille Utterback in 2001 - 2002. Two Liquid Time Series art exhibitions exist, one in Tokyo and one in New York City.

Two exhibitions in Tokyo and New York

Participants move back and forth in front of a projection screen, which displays a video clip. By moving closer to

Users send back in time parts of a projection screen by moving closer to it

the screen, the users send back in time the video clip at the area of the screen that is in front of them, creating a ´´time fragment" at that particular area. If they move away from the screen, the fragmented image returns slowly again to its still condition (see Figure 2.5). The videos with which the users interact were shot in sites in Tokyo and New York City respectively. Users can control the movements of pedestrians, cars, etc. However, the videos also contain static elements, such as street signs and trash cans, creating an unexpected contrast between the moving objects and people and the still objects (see Figure 2.6). The user inter-

Figure 2.5: The position of the body of the user determines fragments in time of a pre-recorded video clip (cf. Utterback [2001]). By moving towards the screen, the user sends the part of the image that is in front of him backwards in time. By moving away from the screen, the image returns at its still condition displaying the video clip.

acts with Liquid Times using his own body, which allows the creation of a space where multiple times and perspectives are visualized, although the body can only exist at one place and at one time.

Similarities of Liquid Times and Time Window

In both Liquid Times and Time Window, users affect only a part of the screen while the other parts remain unchanged, and they can push deeper the time dimension which is vertical to the two-dimensional space of the screen and which moves from more recent to older times. Furthermore, in both exhibits the screen returns at a state of stillness, when one stops interacting with it, in the case of Time Window, when one lifts his finger off the screen, and in the case

Figure 2.6: The resulting fragmented image in Liquid Times, taken from Utterback [2001]. Users can control the movement of pedestrians, cars, etc, but the image contains also static elements, creating an interesting contrast.

of Liquid Times, when one moves further away from the screen.

Time Window and Liquid Times however were designed with different purposes in mind. Time Window has an educational and recreational goal, while the purpose of Liquid Times is aesthetic, aiming at creating an embodied experience, that is an experience where the body of the user serves itself as the medium of interaction with the digital information. Another important difference is that, unlike what is happening in Time Window, in Liquid Times the user does not have at all contact with the screen, and interacts with it from a distance, without pressing it at all.

User interacts with screen from a distance

2.2.4 The Invisible Shape of Things Past

The *Invisible Shape of Things Past* was created by Joachim Sauter and Dirk Lüsebrink in 1995 (by ART+COM) (cf. Sauter and Lüsebrink [1995]). In this project, users can wander in the virtual space of a 3D model of the city of Berlin with a 3D headset or on a screen, and navigate through Berlin's urban development phases from year 1900 onwards till today, including the future outlook of Berlin. A time slider indicating the different years from year 1900 till

Users navigate through the space and time of a 3D model of Berlin

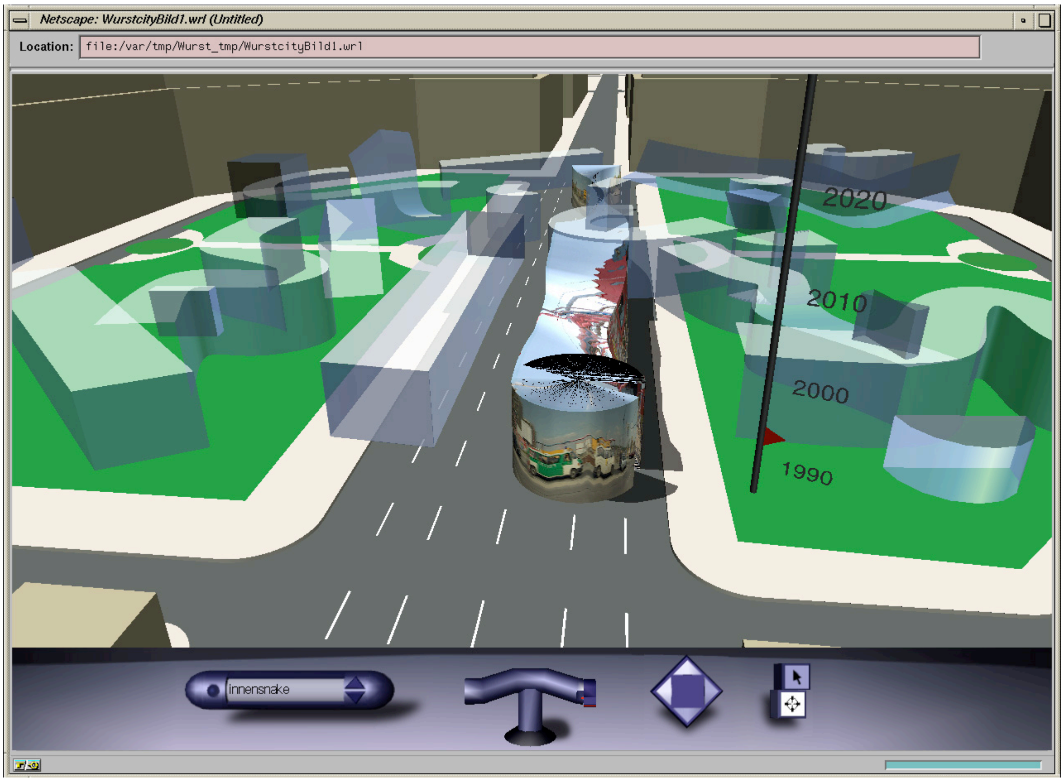

Figure 2.7: The interactive film 3D objects positioned in the virtual space at the place and time of their shot. The user can navigate through time with the aid of a time slider. The more transparent the film objects appear, the more distant they are from the particular time period in which we are. Image appears courtesy of ART+COM.

today serves for the navigation through time, and changes what the user sees in the 3D model according to the condition of Berlin at that particular time. Furthermore, users can interact with interactive 3D objects which represent film materials, positioned inside the virtual space at the place and time that these films were shot. For instance, if the user wanders in the old center of the city of Berlin at the year 1900 and stops at the Postdamer Platz, for example, he can view a video, in the form of a 3D object, of the Postdamer Platz that was shot at that time and at that place and interact with it. The more remote these 3D objects are from the particular time we are in, the more transparent they appear (see Figure 2.7).

Interactive 3D film objects

The interactive film 3D objects are created by positioning all the frames on a path based on the camera movement and perspective of the original film material (see Figure 2.8).

Figure 2.8: An interactive film 3D object in the "Invisible Shape of Things Past". The frames are positioned at the path of the camera based on its movement and perspective and create the interactive 3D object. Image appears courtesy of ART+COM.

The purpose of the Invisible Shape of Things Past is similar to that of Time Window, that is to allow users to explore the urban changes that have occurred in a city. In addition, users can obtain more historical information about specific buildings of interest, in the form of film material.

Both projects: explore urban changes of a city

However, each project uses its own metaphor to achieve this purpose. The Invisible Shape of Things Past, as a virtual reality application, allows users to navigate through a 3D model representation of Berlin. Time Window, on the other hand, uses a digging metaphor to give the user the feeling of digitally uncovering history like an archaeologist, but does not support the same feeling of presence inside the city. The Invisible Shape of Things Past uses a slider interface to navigate through time.

Virtual reality application, different metaphors

Another fundamental difference between the two projects is the way of interaction to navigate through time, as in the Invisible Shape of Things Past this is done with the aid of a time slider.

Time slider for time navigation

Both projects: further info about buildings in the form of video

Both projects provide further multimedia information about specific buildings of interest. This material, in the Invisible Shape of Things Past, is in the form of video, exactly like in Time Window, though Time Window also includes other types of material, images and sound. However, the Invisible Shape of Things Past does not display or zoom in the video but transforms it into an interactive 3D object.

Chapter 3

Requirements and scope

This chapter is dedicated to the requirements of our system. We begin with the functional requirements, which describe what exactly our system does, and continue with the technical requirements of the system.

Functional and technical requirements

3.1 Target group

The system is designed for public use at a museum space. The target group of Time Window are the tourists, who visit Regensburg attracted by its rich culture and history, and local residents who wish to learn more about REX. Out of the 1.6 million national and international visitors of the city per year, the REX visitor centre is expected to attract many thousands of visitors. These include organized groups, schools, individual visitors and a portion of the increased number of expected tourists with Regensburg becoming a UNESCO world heritage site in 2006.

Visitors of Regensburg

Museum visitors are people from all educational and cultural backgrounds, ages, abilities, and with varying experience in computer use. Also, two particular categories of museum visitors of differing ages, children and elderly people, are interesting target groups of our system. Children are active and learn best through playing and elderly people are usually not very familiar with technology. These

All backgrounds and ages

particular characteristics of these two target groups, impose special design goals for the exhibit, whose purpose is the immediate usability and ease of use for everyone.

First-time and one-time users

Museum visitors are typically first-time users as well as one-time users. Furthermore, they will use the exhibit for just a short duration of a few minutes, as they usually stay for just one day in the city where they have to see the sights and perhaps visit other museums throughout the day. Also, the short usage duration might be imposed by the museum itself, in order to give the opportunity to many visitors to experience the exhibit over the course of a day. This imposes a minimal learning curve of the system by the users.

3.2 Functional requirements

The functional requirements describe what our system does and what tasks it should support. The goals described in the following sections must be met.

3.2.1 Explore urban changes of a city

Explore how city developed over the course of a millenium

The purpose is to visualize the changes that have occurred in a city, specifically in Regensburg, throughout the time span of a millenium and see how the city has developed, moving from recent to older times. This exploration of the urban changes is focused at specific spots of the city the user is interested in. The range of the urban development phases starts from year 1100 AC till today, with a time interval of an average of 100 years. These phases are depicted in maps from these particular time periods. For instance, one scenario is to gradually reveal the history of a cathedral, by viewing the cathedral's state of development 200, 400 and 600 years ago, and what stood at its location before its creation.

3.2.2 Navigate along the time axis of maps

Maps containing a time dimension

Navigating along the time axis is related to the first requirement about exploring the development phases of a city, as these are clearly depicted in maps of different time periods. Therefore these maps, in addition to their space dimension, also contain a time dimension, which is simply the time period these maps represent. The purpose of our system is to navigate along the time axis of maps, moving from recent to older times, at a specific spot of the "space" dimension.

3.2.3 Additional information about specific places of interest

Images, videos or audios of historical buildings

The medieval city of Regensburg has a large number of well-preserved historical buildings and monuments spanning over two millenia of history. The system should provide the ability to view these buildings at a greater detail or additional information about them, in the form of video, audio, image or text material.

3.2.4 Historical accuracy

Maps and other material must be historically accurate

The maps of Regensburg that represent the city in different time periods must be historically accurate. Also, the same applies to any form of additional material for specific buildings, in the form of images, videos, audios or text.

The material that was used for the development of Time Window, in the form of maps of different time periods and additional images or videos about specific buildings, was provided by the Regensburg Land Surveying Office[1] .

[1]http://www.geodaten.bayern.de/bvv_web/va_52/home.html

3.3 Technical requirements

The technical requirements are related to the quality of the system as a software.

3.3.1 Responsive to user input

Navigating through time should happen in real-time

The system should respond in real-time to the actions of the user, by meeting the human-time deadlines for its particular tasks. The human-time deadlines for the perception of cause and effect is less than 0.1s (cf. Card et al. [1983]). For instance, the visualization of the user's actions, such as navigating through time, according to the user's input on the touchscreen, should take maximum 0.1s. If the system delays a lot more to respond than the human-time deadlines, the user will perform more actions repeatedly. When the system finally responds, he will not understand what happened and what was the result of his actions.

3.3.2 Extendable

Add or update material about maps and info about buildings

The maps of the city that are currently used in the application may change and additional maps of other time periods may be required. Therefore, the ability of the museum employees to add, update or change material about maps, images and videos about specific buildings easily, without an expert having to recompile or rewrite the application, is of great importance. Therefore, the extendability of the system to include an arbitrary number of maps and additional information about buildings is crucial.

3.3.3 Robust

Stability for many hours and recovery from unexpected situations

The system must run continuously for long hours during the week in the museum. Therefore, its stability is of great importance. Furthermore, it should be easy for the system to recover after unexpected situations that might occur in the museum, such as a power failure, so that the visitors can still use the system throughout the day.

Chapter 4

Design

4.1 Usage scenario

Michalis is a middle-aged architect. His interest for traditional architecture around the world is one of the reasons he enjoys visiting many countries, and it is this interest which has now led him to Regensburg to see samples of medieval architecture. Michalis is not very familiar with computers and even for his work he prefers to draw instead of using computer tools for architects.

User profile

Inside a high-tech visitor centre about Regensburg's history and architecture, his attention is attracted by Time Window. On a large screen which is attached on a wall, an aerial photograph of Regensburg is displayed (see Figure 4.1). On the top of the screen, the title "Time Window" stands out and on the top of the map -but inside the screen- the label "Time Period 2006". The still image on the screen creates a calm impression which, in combination with the title implying about time travel, raises the curiosity of Michalis to find out what this exhibit is about. Michalis notices that there are no devices like a mouse or a keyboard, but just a screen.

A large screen displays a photo of Regensburg at present time

On a poster about the exhibit next to the screen it is written "Place your finger on the screen and move it around. See how the past maps of Regensburg are gradually revealed.". He touches the screen to see what will happen. Suddenly,

Hole in time appears by touching on the screen he sees that at the place where he touched a "hole" appears with a map of Regensburg in pink color. The label on the top of the screen also changes its name accordingly and indicates "Time Period 1650-1800". First the pink map is blended with the aerial photograph of Regensburg, and while he moves around with the finger on the screen it stays the same. But then he notices that if he continues rubbing around the same point or area continuously, the map becomes clearly visible without being blended with the map of present anymore. What he finds interesting is how the city changed at specific places. For example, he discovers the old walls of the city that now do not exist but justify the shape of the core city, or explores how the Danube river changed and how many different canals and bridges existed.

Figure 4.1: The touchscreen of Time Window attached on the wall, at the "Salzstadel" museum space in Regensburg. Before the interaction, it displays an aerial photograph of Regensburg at present time.

Next time layer in different color appears

Then he continues rubbing around the same point, and he sees a next hole with a map in orange color appearing. The label on the top of the map changes accordingly "Time Period 1450-1600". Like in the case of the previous map of the year 1800, the orange map is again blended with the content of the screen. But by "digging" further continuously around the same area, the content of the orange map becomes clear and he can examine what Regensburg looked like in the year 1600. Then he passes over the old stone bridge and he notices that a red diamond appears with the label "Stone bridge" under it. At first he wonders what the purpose of this diamond is and he is curious to see what it contains. After lifting his finger off the screen, he taps with it on the diamond to see what will happen. Suddenly a video of the stone bridge zooms in and plays, showing a 3D animation of what it was like in the year 1600. Finally, since he is not interested in seeing the whole video, he taps on it and zooms out. Fascinated by the diamonds, he starts digging in other parts of the screen in order to search for more. As he finally lifts the finger off the screen, he sees the hole slowly filling up again and the still aerial photograph of Regensburg at its initial state of calmness.

4.2 Design methodologies about interactive exhibits

Design goals for interactive exhibits

Due to the fact that Time Window is used by museum visitors, its interface is designed with specific design goals, which also apply to interactive exhibits in general. These design methodologies are inspired from the book of Borchers [2001], "A pattern approach to interaction design".

Inviting appearance

Attract the attention

Before the actual use of the exhibit, initially it is important to attract the attention and raise the curiosity of the visitor. This should be achieved without the exhibit being too noisy or visually active.

Still screen raising curiosity

Time Window shows a very simple screen and a calm impression of the map of Regensburg at present time, raising the curiosity of users and inviting them to explore the exhibit[1].

Ease of use

Minimal learning effort for novice users

Museum visitors are novice users and they will use the exhibit for just a few minutes, so they will not spend any time to learn how to use the system. So, the system should be intuitive and immediately usable by people.

Touchscreen direct form of input

The use of the finger to interact with a touchscreen is a form of input that the users do not have to learn and it is very intuitive.

Engaging, interactive, interesting, fun

Keep interest alive till educational message is delivered

While interacting with the system, the interest of the user should be kept alive. After the initial attraction, the user should be occupied until the educational message of the exhibit can be delivered.

Learning history in a fun and engaging way

By incrementally revealing knowledge to the user about the past time periods of Regensburg and more historical information about buildings in a playful manner, the user learns history in a fun and engaging way.

Clear message

System understandable

What the system does should be understandable and unambiguous. If this is not completely clear to the user, he will become confused and give up using the exhibit.

Clear how to move back in time

It should be understandable to the user how to move back in time, in which time period he is and how to view additional information by interacting with his finger.

Educational message

Teach visitors about Regensburg's history and historical buildings

An interactive exhibit has to fulfill a main purpose beyond plain entertainment and fun, which is to deliver an educational message to the visitors and to teach them some facts, concepts or ideas through the interaction.

[1] Design pattern "Simple Impression", as described in Borchers [2001], pages 120 - 121.

The educational message of Time Window is the history of Regensburg city and of its monuments and historical buildings.

Invisible hardware

The display of complex technology might scare off the users from using the exhibit, in case they think one would need expert skills to use the exhibit. Therefore only the technology absolutely necessary for the use of the exhibit should be shown. Also, "mouse-like" or "keyboard-like" input devices should be avoided in general[2].

Display of technology, such as keyboard or mouse, should be avoided

In our exhibit the only "technology" shown is the touchscreen and there is no input device, since the medium of input is the finger.

Touchscreen only technology displayed

4.3 Design features

The design features offer a solution to meet the functional requirements of the system, as described in the section 3.2—"Functional requirements".

4.3.1 Time layers to represent maps of different time periods

For the purpose of navigating through the time axis of maps as described in the functional requirements, the concept of "time layers" is introduced.

Concept of "time layers"

The urban development phases of Regensburg are represented in maps of twelve different time periods, spanning from year 1100 AC till today with a time interval of an average of 100 years. Some of the buildings, especially in the early development phases of Regensburg, are not depicted, because historical information is missing about their exis-

Maps of Regensburg of twelve time periods

[2]Design pattern "Invisible Hardware", as described in Borchers [2001], pages 141 - 143.

tence. Furthermore, the exact date of the creation of certain buildings is also not certain.

Time layers are maps of different time periods

Time layers are city maps of distinct time periods. For instance, a map of Regensburg of the time period 1600-1800 AC is a time layer, and a map of Regensburg of present time is another one. A map is a two-dimensional representation of the three-dimensional space of a city. Therefore, a time layer is a map that is associated with a time dimension, the time period that corresponds to the map. Time layers are snapshots of time of the evolution of a city (see Figure 4.2).

Colors to represent the different time periods

Although the above definition describes a time layer, the question is how one would know which buildings were created at which time period. This kind of information also had to be depicted. Therefore, different colors were used to represent the buildings that were created in different years. The maps of the different time periods then contain only the buildings that were created in these particular time periods, displayed in that color.

The time layers of different colors extracted from one single map

In fact, what was provided as the material for the application was one single map of the city core of Regensburg, containing all its buildings in different colors. From that single image all time layers of different colors, containing all the buildings created in these particular time periods were extracted.

Stacked on top of each other, moving from present to past

Time layers are stacked on the top of each other, starting from the topmost layer which represents Regensburg nowadays, and moving from the most recent to the older time layers.

4.3.2 Digging metaphor for temporal interaction

Archaeologist metaphor to discover urban changes at specific places

The time layers, as described above, are stacked on top of each other. The question is how to reach the deeper and older time layers, but at specific places only, when one sees only the present map of Regensburg. In other words, the purpose is to view how the city developed at specific places. The approach we chose is similar to how an archaeologist reveals past artifacts of a city and discovers its

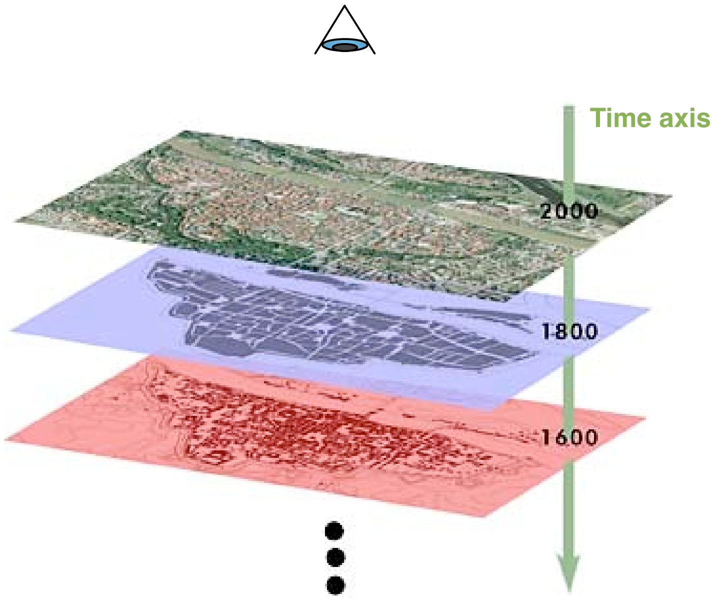

Figure 4.2: The maps of Regensburg from different time periods, stacked on top of each other, moving from recent to older time periods. These maps are referred to in our system as "time layers".

history. It is more interesting to make the users discover things on their own, than to reveal immediately the overall changes in the city, because then the mystery of exploring and revealing would be lost.

Reveal archaeological layers

In archaeology, artifacts from more recent times lie above those from more ancient times, constituting the archaeological layers. The archaeologist gradually digs only at specific sites of the city and not throughout the whole city all at once, to reveal the archaeological layers that correspond to particular eras. In the same way, the user digs on the screen with his finger at specific places in order to gradually reveal the past time layers at these places.

Return to the present when not digging

In addition to revealing the past time layers, the system provides the ability of returning in the present when one is not interested of digging anymore. Although this does not naturally happen in archaeology where the digged site cannot magically fill in, this feature was a design decision taken intentionally to allow other users to take over the exhibit after another person has used it[3]. Therefore, when the user lifts the finger off the screen, the digging slowly fades out and the screen returns at its state of stillness showing again the aerial photograph of Regensburg.

Focus on moving backwards in time

The purpose of Time Window is discovering the past by moving backwards in time, so the ability of moving forward in time was not in the scope of the system. If the user wishes to do so, he must start again digging from the top time layer once he has lifted his finger off the screen and returned to the present.

Explore specific places and not overall changes

Furthermore, the focus of Time Window is specific areas and not overall changes, which can be otherwise easily visualized. Therefore, the system was designed in such a way that it is difficult to see changes that have occurred for the whole city rather than a specific area without having revealed the whole map first. However, a slider on the screen next to the map could be used, where the user could click to see the whole city at a specific period, or have a "page" on the top of the screen which the user can drag in order to access the city map of the next time period.

Focus on exploring the past, although time consuming to reach last time layer

Lastly, it is time consuming until the user reaches the last, deepest time layer, since he has to traverse sequentially by digging all previous time layers first. However, exploring the past time layers starting from the present was of bigger importance than the time spent for reaching the last time layer. Having a slider at the side of the map where one can choose easily the time layer of one's choice could help.

4.3.3 Diamonds for additional information for specific places of interest

Audio-visual information about important historical places

Some of the buildings of Regensburg , for instance the "Goliathhaus", the "Salzstadel", or the stone bridge, are of great historical importance. These buildings or at least some parts of them, were created or modified at a specific time

[3] Design pattern "Easy Handover", as described in Borchers [2001], pages 117 - 119.

period. Some of these buildings even combine architectural styles from different eras. The visitor would like to view additional information about the historical landmarks of a particular time period. As the use of text was not generally preferred in the exhibit, the thought was to provide this kind of information audio-visually, with an image or a video providing a closer look of the building, or an audio describing a story about the building.

Diamonds to indicate that buildings contain information

There is a need to indicate that these specific historical buildings contain additional information. This is where the thought of using a diamond object for this indication was thought to be most suitable. A diamond at a certain place implies that there is something precious, a treasure to be discovered at that place. A comparison is the bottle with the genie that one opens causing the revealing and growing of its content. The idea of the bottle with the genie also gave the thought of zooming the image or the video, as if the diamond really contains the image. Once zoomed, the video can play, and the user can stop it at the time that he wishes to, or after the video has actually ended.

When diamond revealed, dug hole does not fill in

When a diamond has been revealed, the dug hole does not fill in when the user lifts the finger off the screen, but stays there so that the user can select the diamond in case he wishes to do so. The user opens the content of the diamond by tapping on it with his finger.

Diamonds are associated with a point in time

The diamonds belong to a particular time period, so they are linked with a point in time. Only when the particular time layer has been revealed is the diamond for that particular time layer visible.

Diamond labels

Though initially the use of text was not thought as an option in the system, it was later concluded based on visitor's feedback that the diamonds should contain labels, describing the buildings they contain. Also, a diamond can have a particular color, based on the time period to which it belongs or on aesthetic values. For example, red diamonds were chosen, as they are clearly visible with all maps and they are the most rare and precious in the world.

4.3.4 Touchscreen device

Benefits of touchscreens and use in public places

A touchscreen is one of the most direct forms of interaction, as there is no displacement between display and control. This is one of the reasons why touchscreens have already been successfully used in interactive systems in public spaces, such as ticketing machines, ATMs, and public information kiosks. They are very intuitive to use, especially for novice users. The user does not have to learn how to manipulate any input device in order to interact with them. Furthermore, one of their big advantages is their simplicity. For all these reasons, we chose to use a touchscreen for our exhibit. It is going to be used at a public space, and for museum visitors, who are typically novice users and one-time users, which makes it an appropriate form of input that the users do not have to learn. Also, for the digging metaphor, since one naturally digs with his finger in the sand, it would be the most intuitive to use the finger to imitate digging on the screen.

Detecting only one point in time and not multiple pressure levels

However, some of the disadvantages of touchscreens had to be taken into account for the design of the system. Most touchscreens can detect only one point at a time and they do not measure pressure, something which could be useful for allowing multiple users to interact with the application at a time, or for interacting with many fingers or with the whole hand. The measurement of multiple pressure levels would make the implementation of digging easier, as it would depend on really how deep you press the screen rather than just the location of the finger on the screen.

4.4 Interaction techniques for navigating through time

Mapping of 2D input of finger location to 3D visualization of digging

The purpose of the exhibit is to reveal the past time layers by pressing deeper with the finger on the touchscreen. However, most touchscreens are not pressure sensitive and measure only one single point at a time. Therefore, the task of digging depends only on the location of the finger, though one would wish to be able to sense the level of pressure on the screen. We have a mapping of the 2D input of the finger coordinates to a 3D visualization, containing the time dimension as well.

We tried out several interaction approaches for navigating through time based on the above restrictions of typical touchscreens, like for example holding the finger on the screen to reveal past time layers.

Different interaction techniques

4.4.1 Time slider

One approach, was a simple time slider that was used for navigating through time and was prototyped on paper (and with a video). On the screen a map of the city was shown, and on the side, a time slider. The time slider showed a medieval vehicle moving in between hills, whose valleys represented the different time periods. If the user tried to stop at some point in time other than the distinct time periods, by placing the vehicle at that point, the vehicle would roll because of gravity to its nearest valley. The incorporation of this physical analogy of the rolling vehicle in the time slider, was chosen so that one could choose only between particular time periods and not continuous time (see Figure 4.3).

Time slider with medieval vehicle

The idea of the time slider did not fulfill the purposes of the exhibit. First, it was very simple and not so interesting to use, and did not let people discover by themselves history. And, most importantly, it could not satisfy the need for exploring what existed at specific places of interest.

Shows overall changes and not changes at specific places

4.4.2 Expanding the temporal hole automatically

One idea that we explored in order to reveal the older time periods was the following: the longer the user holds his finger at a specific location on the screen, the deeper the user "digs" into time for that specific location only. The user sees the next less recent time period, as a small circle, at the point where he pressed his finger, which gradually grows till it reaches the size of the fingertip. The fingertip size is

The longer one holds finger still, the deeper he digs in time

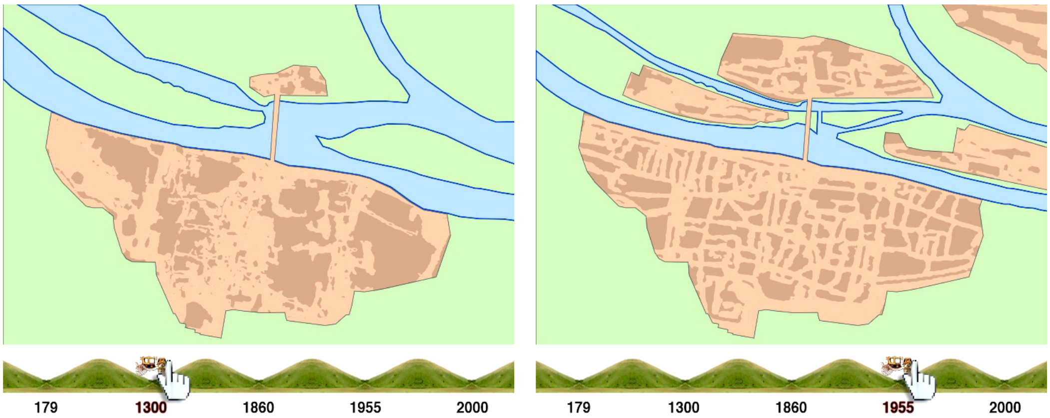

Figure 4.3: A time slider is used for the navigation through the time axis of maps. A medieval vehicle moves between the hills. The bottom of these hills represent the different time periods, and the medieval vehicle stops only at these points because of gravity. Here is an example of how the map of Regensburg was in year 1300 and in year 1955.

the "bottom" of that particular time period, which means that if one has reached it and continues to dig further, the next time period will be revealed. If the user moves to another part of the screen he remains at the same time period up to which he had dug, and does not dig any further (see Figure 4.4).

Drawbacks of approach

This approach suffers from several drawbacks considering the interaction which led to experimenting with different techniques.

Waiting till circle appears

The most fundamental problem is that the user has to wait till the circle appears, without performing any action during that time, since the only thing he does is just holding his finger still at a place. The speed with which the circle grows can be adjusted, but this does not improve the interaction. The circle should appear neither too fast, because then the user won't be able to see what is happening and realize the results of his actions, nor too slow, since then he will have to wait for some time.

Interaction is not smooth if user moves to another point

Another problem, less important -but still breaking the interaction- is that if the user moves to another point of the screen before the growing circle has reached the fingertip size, he sees a a small circle becoming suddenly a line

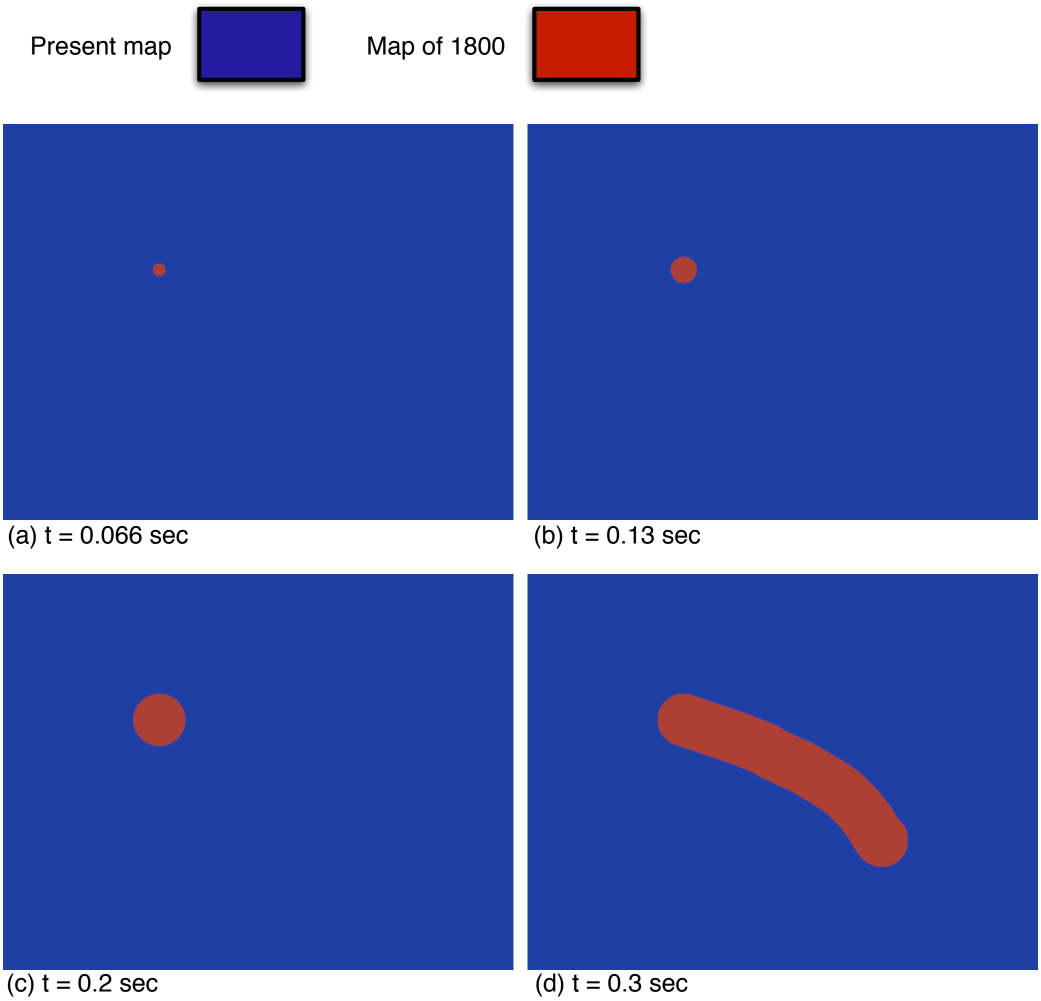

Figure 4.4: The "temporal hole" growing gradually up to the fingertip size, when the user holds his finger still at a specific place, as shown in (a), (b) and (c). In (d) the user moves with the finger to another point.

drawn with the "fingertip size", and the smoothness of this transition is broken (see Figure 4.5).

Due to all these problems, a digging metaphor was examined.

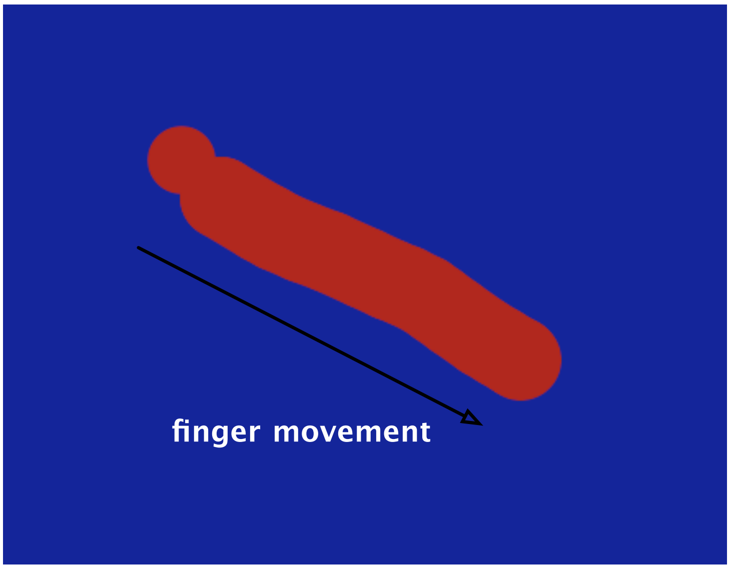

Figure 4.5: If the user moves to another point before the growing hole has reached the fingertip size, the transition from the small circle to the lines drawn with the fingertip size does not appear nice visually.

4.4.3 Temporal hole according to user movement

Used in the exhibit

This approach was the one that was actually used in the exhibit.

Digging with small continuous finger movements

In this approach, a drilling metaphor was examined in order to move backwards in the time dimension of maps. The user digs at a specific point with his finger, by making small continuous movements at that point resembling the movements one does when digging into sand, in order to reveal the next, deeper, time layers at this position.

Next time layer becomes gradually clearer till it is completely revealed

The user sees the next time layer appearing as a spot of the size of the fingertip, which is first blended with the more recent time layers, and as the user continues to dig, it becomes clearer until it is completely revealed and contains only that time layer. When the time layer has been thoroughly uncovered, the user reached the bottom of that time layer and if he continues to dig, then he will start revealing

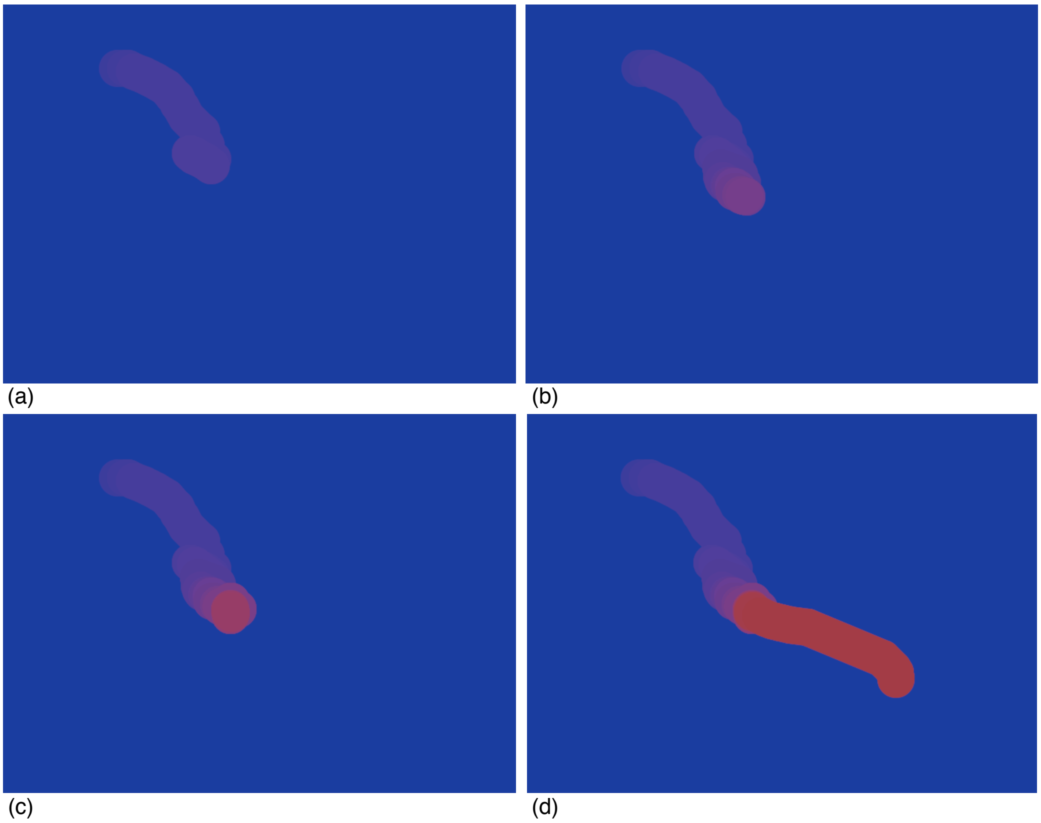

Figure 4.6: The next time layer of the year 1800 appears with the digging metaphor. First, it is blended with the present, but as the user continues to dig with small continuous movements around a certain place, the time layer becomes clear.

the next time layer in the same way, on top of what he has already dug. Again, if the user moves to another part of the map he stays at the same time layer up to which he had dug (see Figure 4.6).

Problems of previous approach overcome

With the use of this metaphor, the two problems of the previous technique are overcome. Since the user now performs the action of continuous movements to reveal the next time layer, he does not have to wait without performing any action. Secondly, the transition from the uncovering of one time layer to moving to other parts of the screen, appears smooth visually.

No feedback about depth of digging

However, this method has the disadvantage of lacking feedback about the depth of digging. Since the time layers

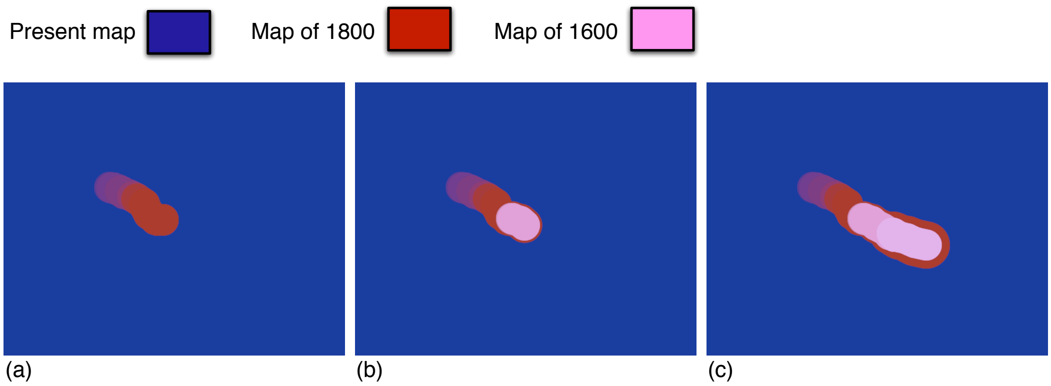

Figure 4.7: Two time layers gradually appearing with the "digging-depth feedback" method. The first time layer of 1800 appears as in the digging metaphor, as shown in (a), but the next time layer of the year 1600 appears with a margin around it, containing all previous time layers that have been already dug (in this case only of 1800), as shown in (b) and (c).

appear on the top of each other, the sense of how deep you have dug, in relation to the initial topmost time layer of the present, and the sense of how many time layers still remain to be explored, is missing. In order to provide this kind of digging-level-feedback, a last technique was explored, though finally it was not chosen.

4.4.4 Temporal hole with digging-depth feedback

Provide feedback about digging depth

This alternative technique was developed in order to provide feedback about how deep in time one has dug up to.

Next time layer appears inside previous ones

This approach has as follows: the next time layer does not appear on top of the previous one, but inside it. The previous time layers are like a margin surrounding the time layer in which we currently are. The deeper one digs, the more previous time layers will be added to this surrounding area. This is actually what happens in sand as well, where the deeper one digs the bigger the hole appears, containing all upper layers that you have already dug at the sides (see Figure 4.7).

Area around current time layer is redundant

However, the purpose of the user is to explore a specific place of a city back in time. The area around the current time layer is not of interest to him. This visual information is redundant and consumes a considerable part of the screen, especially for many time layers. That was the main reason why this technique was not chosen from amongst the others for further development. A possible solution for providing a digging-depth feedback is the previous approach of the drilling metaphor by using an alternative way of representing the depth of the digging. For example, one could have a slider on the side which could contain all the time periods and indicate in which one we are.

Chapter 5

REX Preview prototype

5.1 Software environment

Objective-C with use of Core Image, Quartz 2D and Core Video

The application was developed for a PowerBook with a 1.67 GHz PowerPC G4 Processor and Mac OS 10.4. The laptop was connected to a 40″ inch LCD screen with a SMART Board[1] touch screen overlay.

The program was written in Objective-C with the support of the following libraries: Core Image, Quartz 2D and Core Video.

5.1.1 Core Image

Image processing framework in Mac OS 10.4

Core Image is an image processing framework embedded in Mac OS 10.4. It enables developers to apply a large variety of filters and special effects on both videos and still images, without affecting the original images. It also provides support for custom filters. It takes advantage of the power of the GPU (Graphics Processing Unit) of today's graphics cards, which can process more than 6 billion pixels per second, instead of only the power of the CPU. This allows the real-time processing of high-resolution images. The Core Image application programming interface (API) is written

[1]http://www2.smarttech.com/st/en-US/Products/SMART+Boards/Overlays/

in the Objective-C language. This is the reason why the Time Window application was developed in Objective-C.

Core Image used for processing images

Core Image was used in Time Window for various tasks, for example blending images with a mask, blurring them, scaling them, etc. Also it was used in combination with Core Video for applying filters to video frames (see also sections 5.3.1—"Blending of two maps of different time periods", 5.3.2—"Zoom in/out of image or video" and 5.3.2—"Playing of video content").

5.1.2 Quartz 2D

2D drawing engine for Mac OS X

Quartz 2D is a vector-based two-dimensional drawing engine for all versions of Mac OS X. It can be used for drawing lines and shapes, path-based drawing, layer drawing, PDF document generation, etc.

Used to draw lines and circles to an image

Quartz 2D was used in Time Window to draw lines and circles to an image which was further processed by Core Image. It was also used for drawing the diamond images offscreen on the map images. For further details, see sections 5.3.1—"Drawing to the mask" and 6.4—"Design for arbitrary number of diamonds".

5.1.3 Core Video

Apply Core Image filters to video frames

Core Video is Mac OS Tiger's video processing framework. It enables developers to process individual video frames easily, for example with the use of Core Image filtering, without having to worry about hardware related or display synchronization issues.

Movie compressed in QuickTime

The video source in Time Window is a movie compressed in QuickTime, from which the individual frames are extracted. Core Image filters are applied to the frames and the video is displayed on the screen with Core Video.

The cooperation of all the above technologies is shown in Figure 5.1.

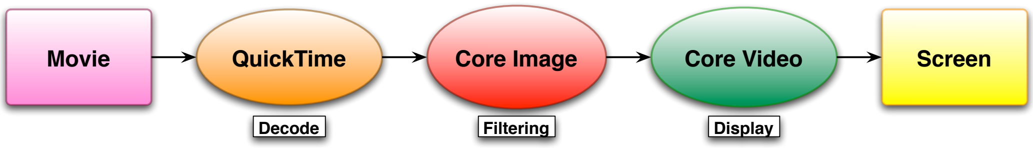

Figure 5.1: The cooperation of Core Image, QuickTime and Core Video. The movie is compressed by QuickTime. Core Image filters are applied to its individual video frames, which are then displayed with Core Video onto the screen.

5.2 The REX Preview

An initial version of the prototype of the system was presented at the REX Preview, an exhibition which took place at the Salzstadel museum space in Regensburg on 22 - 29 July 05. The aim of the REX Preview was to give a "taste" of all the interactive exhibits of REX, including REX Band, REXplorer and Minnesang.

Exhibition open to the public in July 2005

5.3 REX Preview features

For the REX Preview, a demo of the system was developed and tested for a week under real-world conditions. Many visitors, journalists, and school pupils interacted with the prototype, which had to run continuously for two weeks (see Figure 5.2).

Demo tested for a week

The application's functionality was provided for two time layers, a present map (an aerial photograph) of Regensburg, and a past map from 1600 AC. There was one diamond containing a video about the 860 year old stone bridge of the city "Steinerne Brücke".

Variety of visitors interacting with demo

The user could reveal the map of 1600 as follows: initially, the touchscreen showed the present map of Regensburg. The user, by moving his finger on the screen, could dig from the top time layer which shows the present, to the next time layer which represents the past map of 1600, and by doing so he could reveal it at the places where he touched. As soon as he lifted the finger off the screen, the dug hole

Two time layers and one diamond

Figure 5.2: The mayor of Regensburg Petra Betz, interacting with Time Window at the REX Preview, sending back in year 1600 the parts of the touchscreen where she touches. The picture was taken from the newspaper article of 23.07.2005 of the Donaupost[a] about the REX Preview.

[a]http://rex-regensburg.de/fileadmin/documents/10_Donaupost_23.07.05.pdf

would gradually fade out and the map would return to the present.

Zooming in of video upon diamond selection

The user could also uncover a diamond, which became visible by revealing the map of 1600 at the location where the diamond was hidden. If a diamond was uncovered, the dug hole would not fade out when the user lifted the finger off the screen, so that the user could tap on the diamond and select it. By selecting the diamond, an image would zoom in. Later, a video of the stone bridge was used instead of the still picture, which could zoom in and play its content. The interaction of the user with the application took place in real-time.

5.3.1 Hole in time for two time layers

Blending of two maps and drawing into mask

Two maps of Regensburg at different points in time represented the time layers. To create the effect of the hole in time, two components were needed, the blending of the two maps, and the drawing into a black and white mask image used for the blending.

Blending of two maps of different time periods

Two maps of Regensburg as CIImage objects

Two maps of Regensburg from 2000 and 1600 were used for the demo. These were represented as Core Image image (CIImage) objects. CIImage[2] objects are immutable, as the original images are not modified, and they can be processed with Core Image filters. In order to implement the revealing of the map of year 1600 below that of the present, as described above, the two maps had to be blended.

A blend mask filter used for the blending

A blend mask filter was used to reveal only a part of the older image that the user has "dug" up, the CIBlendWithMask[3] filter. The blending filter takes as input two images, one foreground image, the map of 1600, and one background image, the map of present, as well as a greyscale mask image. Then the filter blends the background and the foreground image, depending on the color of the mask. It chooses the background image (the present), at all places where there is black in the mask image, and the content of the foreground image (the past), wherever there is white, and a mixture of both images wherever there is grey. The filter returns the blended image as a CIImage object (see Figure 5.3).

Drawing of white lines into a black mask image

White lines are drawn in real-time in a black mask image, according to the position of the mouse, in order to show the content of the map of 1600 wherever the user touches.

Display of resulting image via OpenGL

The resulting blended image is drawn to the fullscreen window of the application via OpenGL. The resolution of the pictures that were used was chosen intentionally to be the same as the resolution of the particular screen, which was 1280x768.

[2] http://developer.apple.com/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_concepts/chapter_2_section_4.html

[3] http://developer.apple.com/documentation/GraphicsImaging/Reference/CoreImageFilterReference/Reference/reference.html

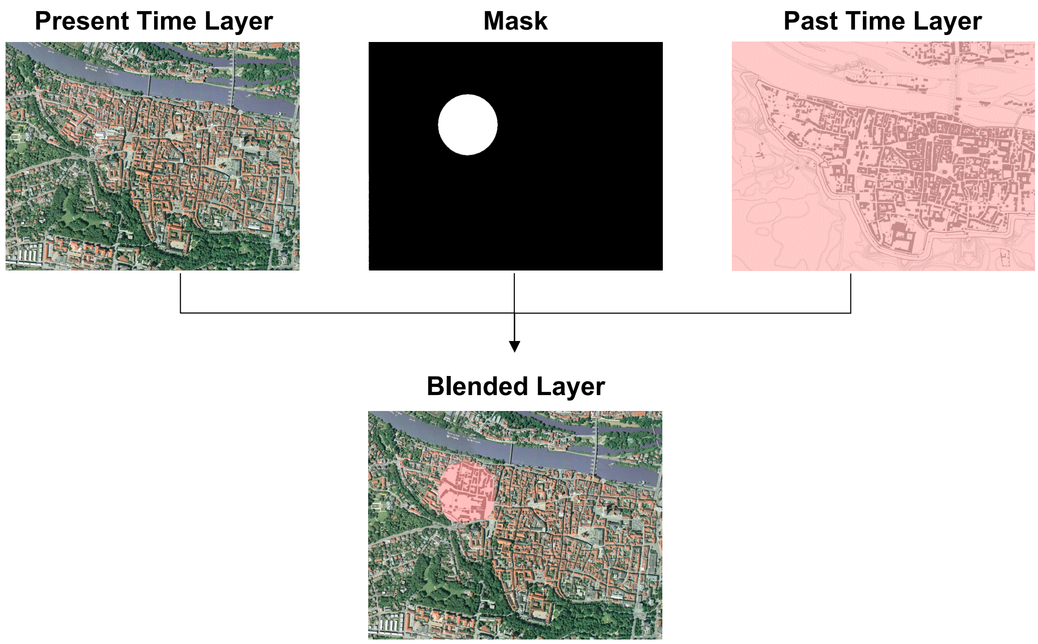

Figure 5.3: The CIBlendWithMaskFilter provided by Core Image, takes two images and a greyscale mask image as an input, and returns a mixture of the two images. At the places where there is black, the content of the background image is taken, and at the places where there is white, the content of the foreground image. Here the input images are a present map of Regensburg, and a map of 1600, and a black and white mask image and the resulting blended image is shown.

Drawing to the mask

Drawing of white lines to a black mask image, with a CGLayer

In order to blend the images of the present and the past, one has to draw continuously white lines in real-time into a black mask image. On mouse release, the dug hole must fade out. All operations related to the drawing inside the mask were implemented in a class called "BlendMask". The drawing functions were implemented in Quartz 2D. In order to simulate the digging on the screen, the line width was specified according to the size of the finger, and the lines were drawn with round edges. A CGLayer[4] object in Quartz 2D, which represents a layer for the drawing, was used, for increased performance of drawing. The mask image had to be redrawn when adding new lines, or when

[4]http://developer.apple.com/documentation/GraphicsImaging/Conceptual/drawingwithquartz2d/dq_layers/chapter_13_section_1.html

emptying the content of the mask.

The steps needed for the drawing are described as follows:

Steps for drawing

1. First a CGLayer (CGLayerRef) was created and was assigned to a graphics context (CGContextRef), which was our drawing destination.
2. The bounds of the mask image were specified to the size of the fullscreen window.
3. A black rectangle was drawn to the graphics context.
4. On mouse click, white circles were drawn at the location of the mouse and on mouse drag, white lines were drawn.
5. The lines were drawn in white color into the graphics context, by using an array of points storing the current and previous mouse location.
6. A CIImage object from the CGLayer to which we drew was returned, representing the mask image.

The dig fadeout on mouse release was created with the aid of a timer, with a time interval of 33 milliseconds, which continuously redrew the mask, to which a brightness filter was applied with gradually smaller brightness values.

Dig fade out with the use of a timer

Lastly, optionally the mask image was blurred with the use of a CIGaussianBlur filter, for smoother edges, resembling more a dug hole. However, applying the gaussian blur filter to the high-resolution (of 1280x768) mask image was too slow, due to the complexity of the Gaussian blur function, which performs a complex calculation for each pixel of the image.

Mask image blurred with a Gaussian blur

5.3.2 Diamond for displaying additional information

For this prototype only one diamond was supported. The location that was chosen, was that of the stone bridge

Support for one diamond

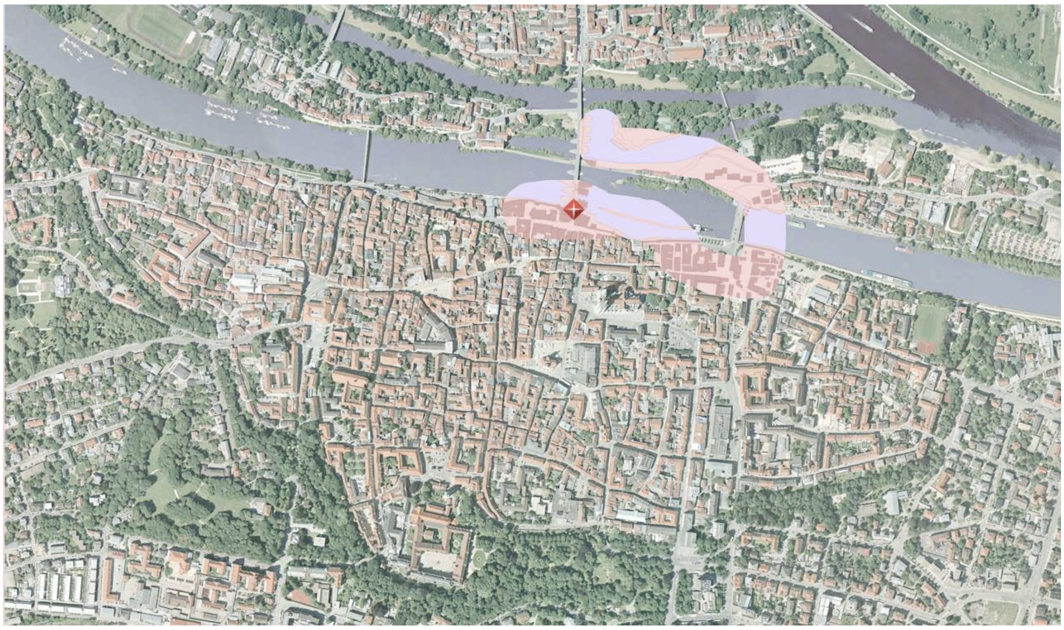

Figure 5.4: A diamond at the position of the stone bridge has been revealed.

"Steinerne Brücke" on the map. The coordinates of the diamond were hardcoded and integrated directly into the source map of year 1600 instead of being drawn by the Time Window application at runtime, as originally intended (see Figure 5.4).

Detection if diamond was revealed, zooming in/out and playing of video

The necessary actions for the selection of a diamond and the display of its content, are the following. First we had to detect if the mouse was near the diamond, or if the path of the user passed from it. Then the diamond would be revealed, and its content could be displayed, in the form of an image or a video, upon its selection. A filter for scaling images and timers was used for zooming in, playing of the video, and zooming out.

Detect if mouse is near diamond

Mouse position or formed path near diamond

To find if a diamond has been revealed, there are two cases: either the user clicks at a mouse point near the diamond, or the path that the user forms passes from the diamond (see diagram 5.5).

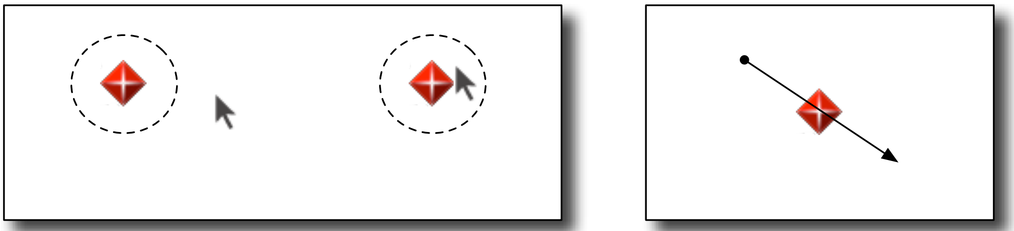

Figure 5.5: There are two cases for detecting if a diamond has been revealed or clicked. The first one, described by (a), is when the mouse is within a certain radius that contains the diamond, and the second one, described by (b), is when the path that the user forms passes from the diamond.

To detect if the mouse is near the diamond, the geometric function of computing the distance between two points was used[5] . The two points taken as input were the mouse location and the diamond location.

Distance between two points

The detection if the mouse path passes from the diamond point, was implemented by calculating the distance of a point and a line. The point was the diamond location and the line was the path formed on the screen by the user.

Distance of a point and a line

Zoom in/out of image or video

In the case that the mouse is near the diamond, as described in the previous section, the diamond is revealed. From there two options are available, either the user clicks at an area close to it (implemented with the same geometric function) and therefore displays its content, or the user clicks somewhere else and the dug map gradually fades out.

If diamond revealed, select it or click somewhere else

If the user selects the diamond, an image or a video of the building will zoom in to fill most of the screen. The image or the first frame of the video respectively is continuously scaled, with the use of a scale filter, and drawn on the top of the map image. A timer is used for the purpose of continuously scaling and drawing the image.

Upon selection, zoom in of image or video

[5]http://astronomy.swin.edu.au/ pbourke/geometry/pointline/

Video image continuously scaled and moves between initial and final position

The scaling is done with the use of the CIAffineTransform filter. The scaling factor gradually grows from 0.0f to 1.0f, from a very small size to the actual size of the picture, from the point where the timer starts, till the zooming out timer ends. Of special attention is the fact that the picture or the video, when it is at its actual size, it is displayed at a location where it does not exceed the borders of the map. This is safely done if the final image is displayed at the "detailed view center" location, a location around the middle of the map, where the image appears completely centered. The image though, must appear as if it is arising from the diamond, therefore the initial position of the small size image must be the diamond position. Then the image gradually moves between the initial diamond position to the final detailed view position where the actual size of the image or the video is displayed, by using a linear interpolation algorithm, as shown in the diagram 5.6).

Zooming out in the same way

Zooming out is implemented in the same way, but the scaling factor decreases to continously smaller values from 1.0f to 0.0f.

Playing of video content

Movie compressed by QuickTime and played with Core Video

Once zoomed in, the content of the video can be played (see Figure 5.7). If the user clicks on it while the video is playing, which means he is not interested anymore in viewing its content, the video stops playing and zooms out. The video source is compressed by QuickTime and played with Core Video, which continuously redraws frames of the it to the screen.

5.3.3 Central state machine

TimeWindowView main class of application

The core of the application is the class TimeWindowView, which is responsible for the cooperation of all the classes and the components described in the previous sections.

Fullscreen window

TimeWindowView is the main class of the application and it contains the fullscreen window of our application. The

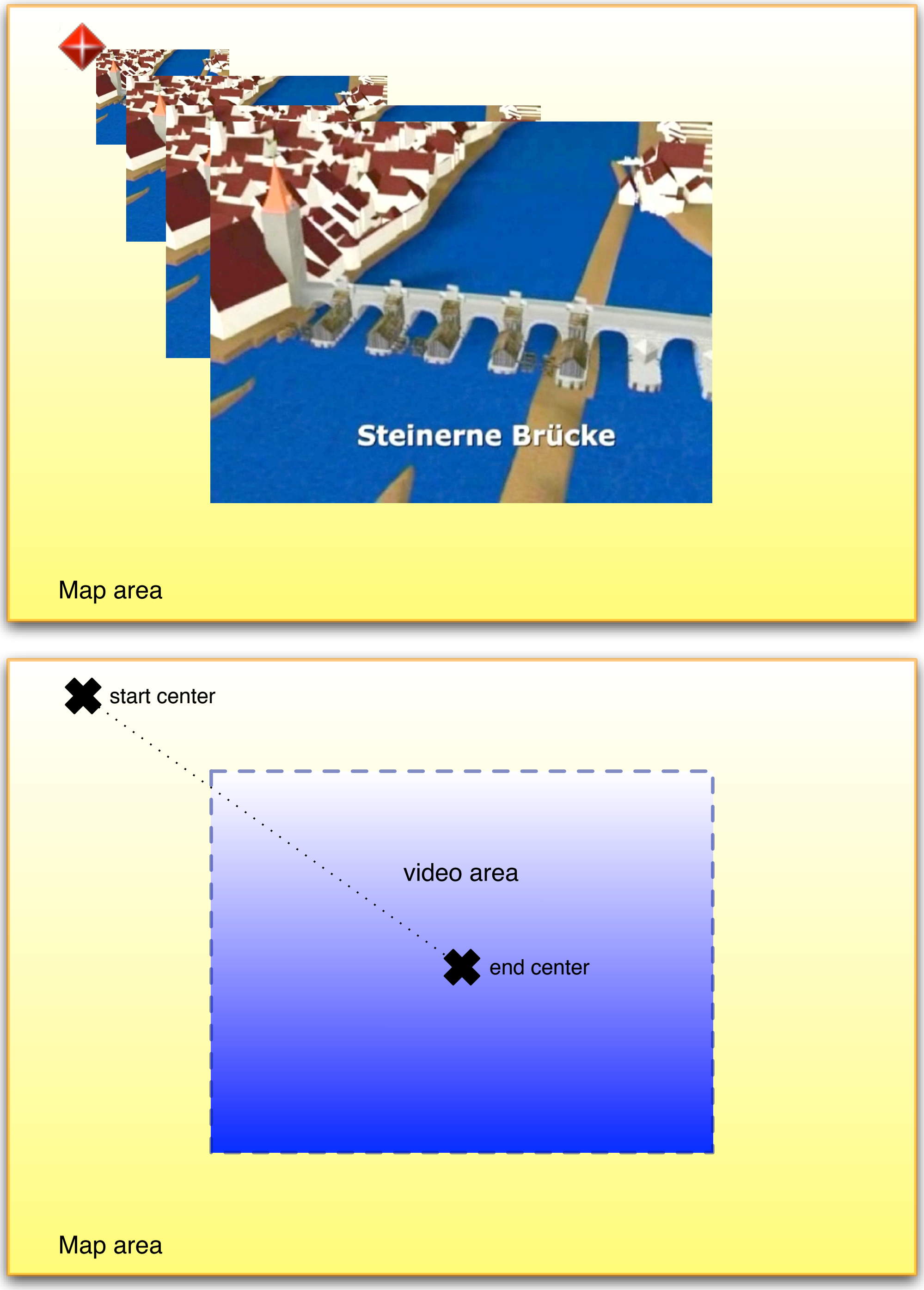

Figure 5.6: The video frame scales from a very small size to the actual one, positioned at the center of the map. The frame moves between the initial diamond position, to the final map center position, using a linear interpolation algorithm.

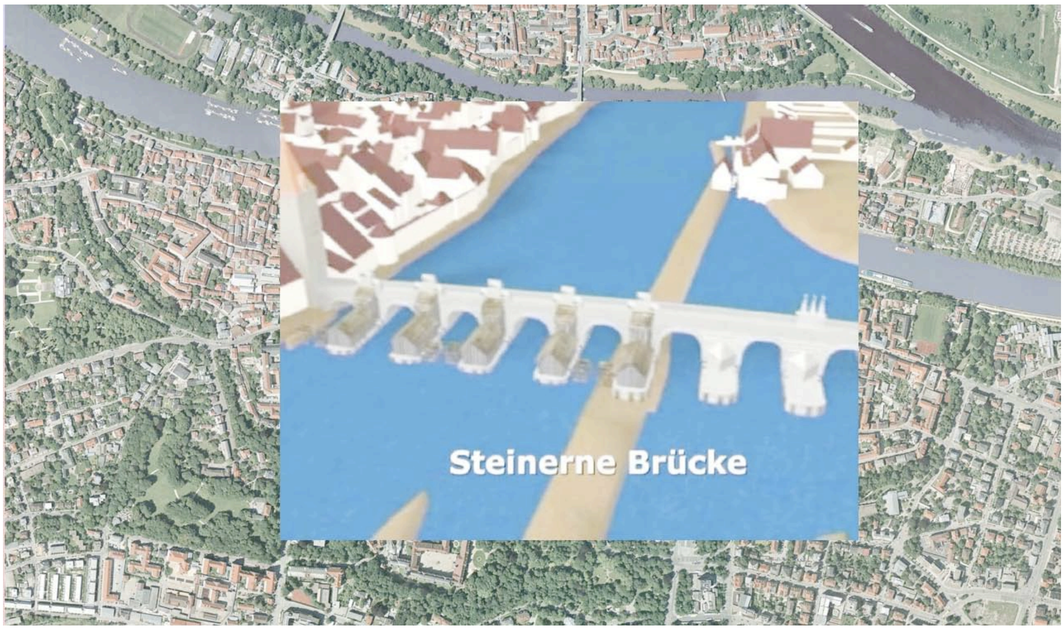

Figure 5.7: A video of the stone bridge plays upon selection of the diamond.

content of the images is drawn to the window and redrawn whenever necessary.

Deals with different mouse events

TimeWindowView deals with the different mouse events (mouse click, mouse release, mouse down) and performs the necessary actions according to them. For example, on mouse click, the hole in time appears, and on mouse release, the dug hole fades out.

State machine, switching between states like digging, diamond found, etc

TimeWindowView is in fact a state machine, switching among the different states the system can be in. These states are: digging, digging diamond found, zoom in, zoom view, zoom out, dig fade out. The transition through these states is described as follows:

Description of the transition between states

The initial screen displays the map of the present, therefore the initial state which corresponds to it is the state of "digging". If the user is at the digging mode and holds the mouse down, the hole in time appears. On mouse release, there are two options, either no diamond has been revealed, or the user has found one. In the first case, the system transitions to the state "dig fade out", which slowly fills in the hole, returning to the present and the initial state of "digging". In the case that the user has revealed a diamond, the system transitions to the "digging diamond found" mode. From there, he either clicks on the diamond, switching to "zoom in" state, followed by "zoom view" and "zoom out", if the states are not interrupted by a mouse click, or he clicks at an area far from the diamond, returning first to the "dig fade out" mode and then to the initial "digging" mode. If the user clicks with the mouse while being in "zoom view" mode and the video has not finished playing, the system switches to "zoom out" mode followed by the "digging" mode. Figure 5.8 illustrates a state transition network of the REX Preview prototype.

5.4 Intermediate evaluation

Time Window accepted with enthusiasm

The REX Preview was open to the public and attracted visitors of all ages amongst school pupils and journalists (see Figure 5.9). A number of press events took place during the preview week, including newspaper releases and some television coverage. The people who came to visit tried out and interacted with all the exhibits and gave their valuable feedback, which was collected and used for the further development of the system. The feedback of the people was very positive and full of enthusiasm for Time Window, and it topped all the expectations of the journalists and the press[6] . Therefore the REX Preview obtained great media echo and was reported not only by the regional press, TV and radio, but also by Germany's largest newspaper SZ (Süddeutsche Zeitung, Munich) and Antenne Bayern, Bavaria's radio with the largest audience[7] .

5.4.1 Visitors' profile and background

Visitors of all ages and backgrounds

The REX Preview attracted fourteen guided tours with 270 visitors[8] . The visitors were mostly German people of all

[6] http://rex-regensburg.de/service/presse/

[7] http://regensburg-experience.de/service/news-im-detail/artikel/3/17/

[8] http://www.it-speicher.de/home/5052-1,1,1.html

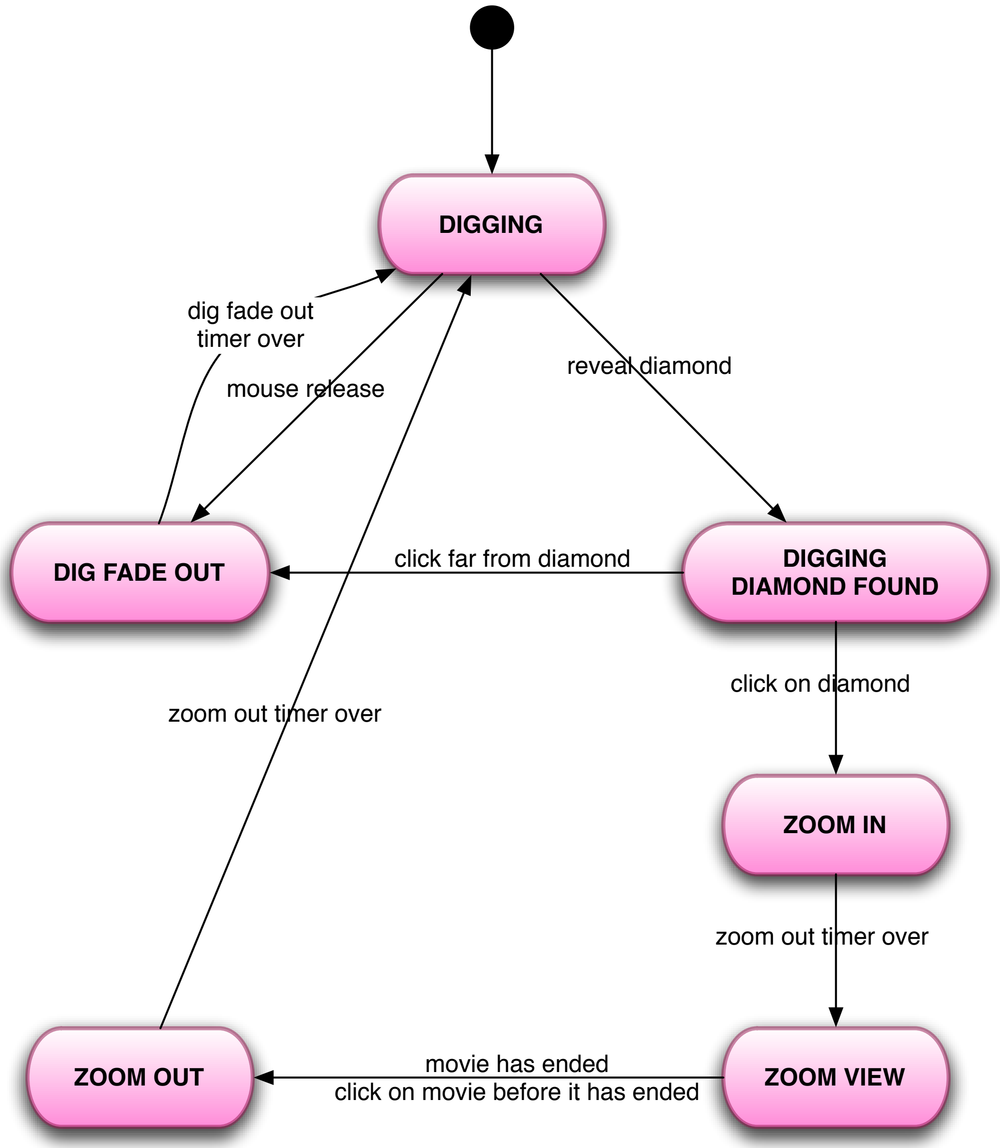

Figure 5.8: A state transition network of the REX Preview prototype.

ages, sexes and educational backgrounds.

Two categories of users: tourists and experts, including young and elderly people

Two categories of visitors with different goals interacted with Time Window. On the one hand there were citizens and tourists of Regensburg who wanted to learn about its history out of personal interest. On the other hand there were experts, such as city guides, journalists, museum cu-

Figure 5.9: Time Window is presented to school pupils at the REX Preview.

rators and marketing and tourist experts, with an interest in history, museums, and new media. Also, the REX Preview attracted young people and school pupils, including the students of the famous Domspatzen choir of Regensburg, as well as elderly people.

5.4.2 Evaluation environment

Informal, qualitative evaluation based on visitors' feedback

There was an informal, qualitative evaluation of the system based on the visitors' feedback during the REX Preview week. Users were observed while interacting with the exhibit and they were then asked for their opinions about their experiences and their suggestions for improvement. The application was tested under real-world conditions and in the natural environment of the museum. The evaluation had the advantage of including a wide range of people from different backgrounds, due to the nature of the preview. Also, the prototype was tested for a long period of time. The application had to run continuously during one whole week for many hours daily, and it proved to be stable during that time.

Visitors found exhibit interesting and educational

The main purpose of this evaluation was to see if the exhibit met its goal of being educative in an exciting and interesting way. The exhibit was very well received with enthusiasm and the feedback of the people was very positive. The users liked a lot the idea of uncovering the past map of 1600 under the current one, and were exploring how different spots changed, for instance the channels of the river or the walls of the city. Also, they were fascinated about discovering a diamond on the screen, and were surprised pleasantly by the zooming of a video once they clicked on the diamond. The visitors could understand what the system was doing and its purpose, and they found the idea of discovering things by themselves very interesting. Therefore, it was obvious that Time Window was at the right track.

Second iteration based on feedback

The visitors' feedback and suggestions for improvement led to a second iteration of the system and were taken into account for the further development of the second prototype.

Exhibit evaluated for two time layers and one diamond

However, only a subset of the functionality of the system was evaluated, as the exhibit at that early stage of development worked only for two time layers and one diamond. Therefore, the reaction of the users could not be tested for more than two time layers and diamonds. Also, possible problems for the many time layers could not be detected.

5.4.3 Visitors' comments and feedback

Enthusiasm reflected in comments

The enthusiasm of the users about Time Window and REX in general were reflected in their comments[9] . "I am anxiously looking forward to it!" "Your concept promises fascination!" "There is nowhere anything like this!"

Educational value

Many visitors noted the educational value of the exhibit about history: "Here, at last, history is told in an exciting way!" (a teacher).

[9]http://regensburg-experience.de/service/news-im-detail/artikel/1/17/

Attracting young people

Some thought that the exhibit would be appealing to young people too. A lady commented: "This would have even thrilled my son, who normally does not take interest in history".

Suggestions implemented in second prototype

The following suggestions for improvement and the feedback given by the visitors were implemented for the second prototype:

- Use drilling metaphor for going through time layers: the map is gradually revealed, as if one digs through sand, instead of appearing immediately.
- Labels on diamonds representing the buildings of interest.
- Improve the responsiveness of the time layers revealing themselves or of the video showing its content.

Improvements for future versions

The following are suggestions for future versions of the system:

- Wiping instead of using fingertips: digging with the whole hand or with more fingers instead of only one finger.
- Make color coding more clear: use meaningful colors for the representation of the time periods (see section 7.4—"Color coding").
- Add castle Prüfening and Walhalla temple, as they are important historical places.
- Two modes for the map, one with the city core only, and one with the surroundings, since the surroundings of Regensburg also include important historical places. Therefore, since the touchscreen cannot represent the whole area with the surroundings at high resolution one can give the possibility to switch between the historical centre and the surroundings of Regensburg (see section 7.4.1—"Include the surroundings of Regensburg").
- Pot of gold instead of diamond: replace diamond with other medieval or valuable objects.

- Multi-user interaction: provide the support for the simultaneous interaction of multiple users with the application (see section 7.3.4—"Multi-user support").
- Damages in World War II: indicate which buildings were destructed or bombed in World War II.
- Zoom into spot where user is digging: provide a higher-resolution view of the map.
- Links between buildings: show the relationship between buildings with common history.
- Create CD-version for home use.

Chapter 6

Second prototype

The prototype of the REX Preview was extended to include an arbitrary number of time layers and diamonds, thus providing the whole functionality of the system. In order for the application to be extendable and maintainable, the software structure was refactored to include additional classes such as "Diamond", "TimeLayer" and "TimeLayerBlend". These classes represent the concepts of diamonds, time layers, and blending, already introduced in 4—"Design" chapter.

Extension to include multiple time layers and diamonds

Also, some minor changes were made based on the visitors' feedback, like for instance the addition of labels for the diamonds and an indication of the time period in which one has dug to.

Small changes based on visitors' feedback

6.1 Map material preparation

A vector image of the historical centre of Regensburg was provided by the Regensburg Land Surveying Office, showing all its buildings painted in different colors, which represent the different time periods in which these buildings were built. From this single image, all different time layers, containing only the specifically colored buildings built during these time periods, were extracted (see Figure 6.1, 6.2 and 6.3).

Time layers extracted from one picture of Regensburg with buildings of different colors

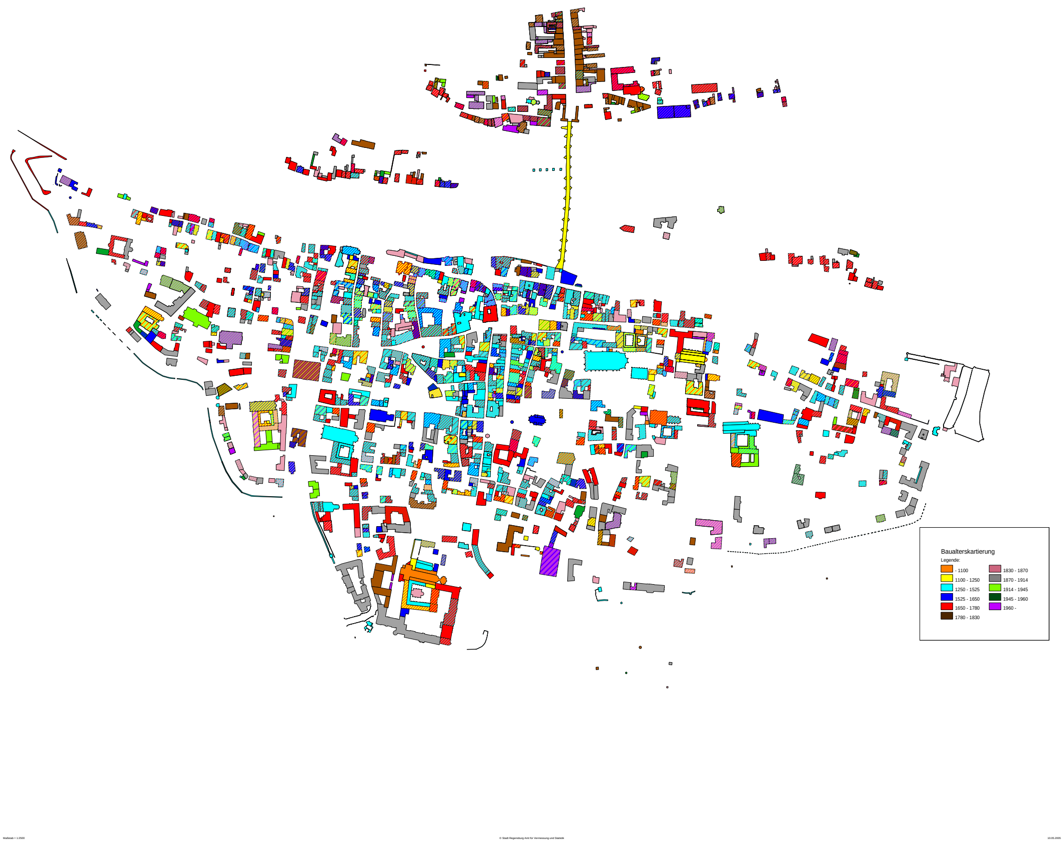

Figure 6.1: An image of the historical centre of Regensburg, containing all its buildings built in different time periods, represented by different colors. From this image all time layers were extracted.

6.2 Time Window classes

System architecture

In this section, the architecture of the system and its most important classes, including a UML diagram describing how these classes cooperate among them, are presented (see Figures 6.5).

Classes and their cooperation

An explanation of the different classes of the system, their attributes, their roles, as well as the interaction among them is presented.

Figure 6.2: A time layer extracted from the above image, which shows all buildings created in the time period 1100 - 1250 in yellow color.

6.2.1 TimeLayer

A time layer is characterized by its map image and its time dimension, which is the year that the map represents.

Map image and time dimension

There are additional attributes for extendability reasons, such as the label of the time layer, which is the title of the time period this layer represents, e.g., "Time Period 1400 - 1600". Also, a time layer stores an array of the diamonds it contains, to ensure the extendability of the system (see

TimeLayer additional attributes

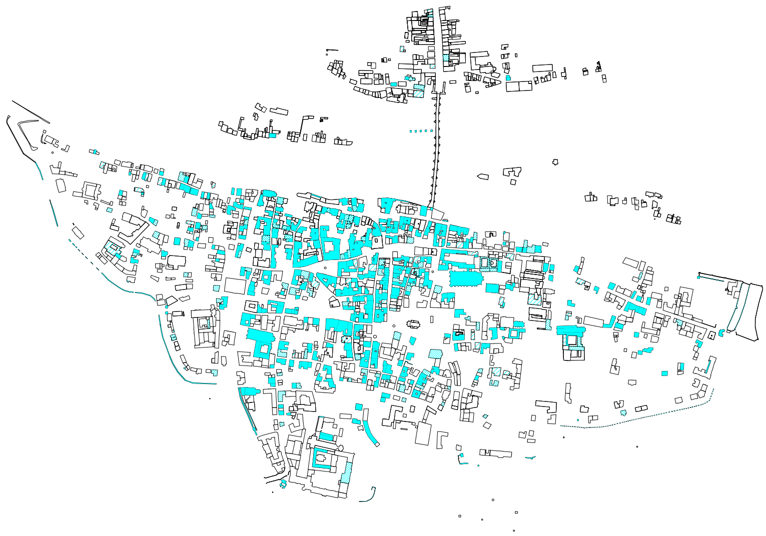

Figure 6.3: A second time layer representing the time period 1250 - 1525.

section 6.4—"Design for arbitrary number of diamonds"). Furthermore, an attribute representing the diamond image is stored in TimeLayer class, which is the display image for all the diamonds of that specific time layer.

Reasons for storing the diamond image in TimeLayer

There are two reasons for storing the diamond image inside TimeLayer class and not in Diamond class. Firstly, to provide the ability of having the same diamond picture for all the diamonds of the same time layer. This saves storing capacities and helps in the consistency of time layers. In this way every time layer will contain the same icon for all the diamonds so that they are representative of that particular time layer. Secondly, for having a different picture or color of a diamond for every time layer.

Label and array of diamonds

The label of the time layer can also have a specific color, according to the color of this time layer. Lastly, a time layer contains an array of diamonds. An upper limit of fifteen diamonds was chosen for performance reasons. The current diamond, which is the diamond that has been currently revealed or clicked, is also stored.

Operations to draw label

The operations to draw the label of the time layer are described as follows. Text is drawn with special fonts, and in the case of the time layer, with a specific color which is set with CGContextSetRGBFillColor, and is the same as the color of the current time layer period. Then, the text is displayed with CGContextShowTextAtPoint to the graphics context of the time layer image. Diamond labels are drawn in the same way.

6.2.2 Diamond

Attributes: position on map and label

A diamond is defined by its position on the map and its label, the title of the building of interest it represents. Initially, the image which represents a diamond, was stored as an attribute in the Diamond class. Later it was decided to include the diamond image as an attribute of TimeLayer class, as the same diamond image is used to represent all diamonds of every time layer. A diamond, specified by its position and its label, is drawn in TimeLayer class at the desired position.

Detection if mouse or path is near diamond: methods of Diamond class

The methods of detecting if the mouse or the path is near the diamond, implemented in the previous prototype in the main class TimeWindowView, became methods of the diamond class. Now these methods take as a parameter only the mouse position, and the last mouse position in case the user forms a path, as the diamond position is stored internally for every diamond.

TimeLayer performs many operations for diamonds

TimeLayer class performs many operations for the diamonds. First of all, it stores an array of diamonds. It prints, based on their position, all diamonds the time layer contains and their labels. This is done in the beginning of the program into the graphics context of the time layer image.

6.2.3 BlendMask

Draws circles and lines into mask

The BlendMask class is responsible for drawing circles and lines into the mask object, as well as redrawing and clearing the content of the mask, exactly like in the first prototype. In the first prototype however, only white lines are drawn, while in the second a grey value between black and white can be specified for the drawing of lines, used for creating the effect of gradually revealing the next time layer.

6.2.4 TimeLayerBlend

Blends current time layer with previous explored ones

TimeLayerBlends''s class responsibility is blending the current time layer with its previous most recent ones and performing the transition from one time layer to the next one. For this purpose it keeps an array of N time layers, starting from the present and moving towards the older ones. It also stores an array of N-1 blend mask objects, one for every time layer except the present, since it is the first one and does not need to be blended with the previous time layers.

Stores content of what has been explored

TimeLayerBlend stores the content of what we have already explored and blends it with the current time layer and the current blend mask object with the use of a CIBlendWithMask filter, as in the first prototype. An example of the result of the transition to the time layer of year 1400 is illustrated in Figure 6.4.

6.2.5 TimeWindowView

Main class: cooperation of classes and drawing result to screen

TimeWindowView is the core of the application and is responsible for the cooperation of all classes and drawing the resulting effect of the hole in time into the window of the application. It includes all actions and state transitions that were implemented for the REX Preview prototype, but in

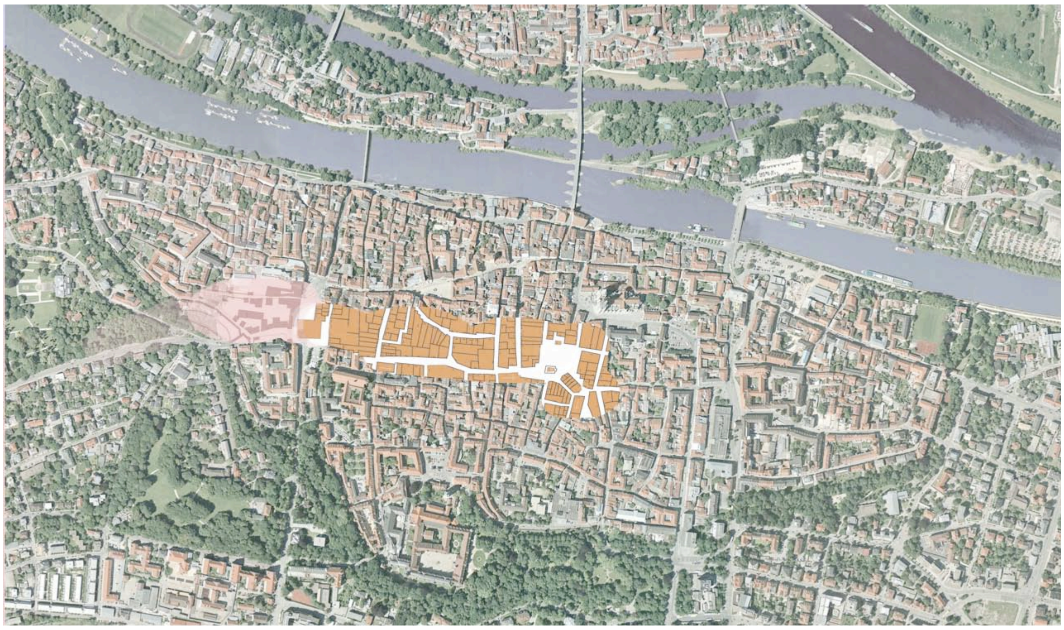

Figure 6.4: The dug hole after the user has entered the second deeper time layer of the year 1400, after having dug first the time layer of the year 1600.

addition to that, is responsible for the transition from one time layer to the next one as well as returning to the top time layer, based on the input of the user.

Creation of diamond, time layers, and time layer blend objects

At the beginning of the program, it creates all diamonds and time layers from their paths and with their labels, and one TimeLayerBlend object. It then adds the diamonds to the respective time layers, and the time layers to the TimeLayerBlend class.

TimeLayerBlend methods called when necessary

All blending operations are then performed by the TimeLayerBlend class and returned to TimeWindowView. Furthermore, all operations related to diamonds or drawing to the masks are implemented in class TimeLayerBlend and are called in TimeWindowView when necessary, based on the user mouse input.

Stores grid of points with clarity values

Also, TimeWindowView sets an array with a grid of points on the screen, storing the clarity value for each one of these points, which is used for the digging. Whenever the mouse passes from the same points, while the user is digging, the clarity value is increased and the current time layer becomes more clearly visible. If the bottom of that particular time layer is exceeded, the current time layer is increased. TimeWindowView resets the array with the grid points again to a minimum clarity value whenever a new time layer is entered.

The interactions between the different classes is illustrated in Figure 6.5.

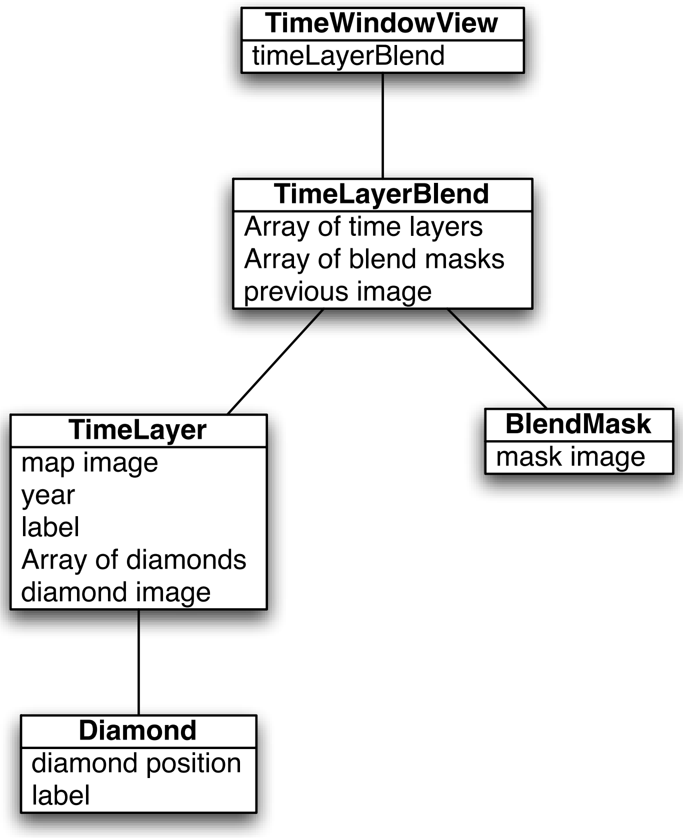

Figure 6.5: Simplified UML diagram: The classes of Time Window necessary for the blending and the cooperation between them. "TimeLayerBlend" class takes an array of time layers and of blend mask objects and blends them. "TimeLayer" stores a map image that represents it and the year, and an array of the diamonds that it contains.

6.3 Design for arbitrary number of time layers

The application has to work for an arbitrary number of time layers. The upper limit of time layers to be included was twelve, which is the total number of time periods of the history of Regensburg. Regardless of the special requirements for REX though, it would not make sense to include more time layers than this upper limit, due to performance and interaction reasons. Blending such a large number of time layers would reduce a lot the response time of the system, due to the blending filter bottleneck. Furthermore, the engagement of the interaction would be affected, as it would be time consuming till the user reaches the last time layer and perhaps would lose his interest to explore it.

Extend to arbitrary number of time layers

Stating that the application has to work with an arbitrary number of layers, means that the program has to perform the blending of the time layers and the transition from one time layer to the next one dynamically, independently of how many time layers there are. New time layer objects, defined by their map image and the time period they represent can be created, and they will be taken into account in the algorithms of the program for navigating through time.

Dynamic blending and transition of time layers

For this purpose all time layers are stored in an array, starting from the present and moving to the deeper ones (we assume we have N time layers). A class TimeLayerBlend is responsible for blending the time layers dynamically. Also, TimeLayerBlend stores an array of blend mask objects (N-1 blend masks) that are used for the blending of each time layer with all the previous ones.

TimeLayerBlend stores array of time layers and blend masks

Every time the user enters the next deeper time layer, the dug hole of the previous time layers that have been explored till now has to be stored in an image, called “previousImage”. The content of the “previousImage” based on the already dug time layers is stored only once, every time the user enters a new time layer, therefore reducing the complexity to that of a regular blend mask filter that blends two images. Then, the “previousImage” is taken as the background image in the blend mask filter, in the same

“Dug hole” of explored time layers stored in a variable way as in the REX Preview prototype, instead of the present map, and the foreground image is the current time layer we are in, extracted from the array. For the second time layer (the map of 1600), assuming the first one is the present, the "previousImage" is the present image (see Figure 6.6).

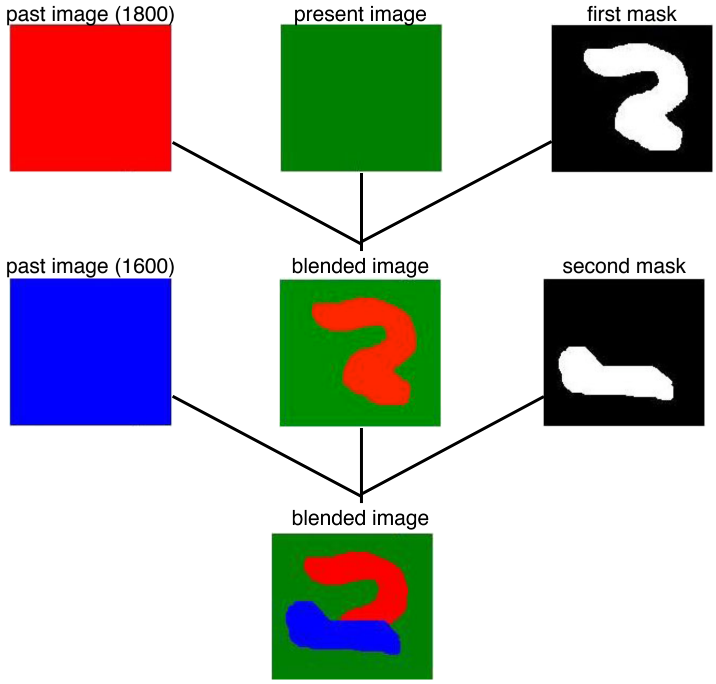

Figure 6.6: The blended image for an arbitrary number of time layers. The current time layer and the current blend mask are extracted from the array of the time layers and blend masks, and the previous content of what we explored till now is stored in a "previousImage" variable.

Current blend mask extracted from array of blend masks

The blend mask image used for the blending is the one that corresponds to the current time layer, extracted from the array of the blend mask objects.

Reasons for using array of blend masks

The reason an array of blend masks was used instead of one blend mask object, in which one could continuously draw, after having cleared the mask when entering a new time layer was, that the blended image for one second was not shown correctly. At the very second of entering the new time layer, when the content of the mask is not yet empty, the blended image appears as if it has been produced from the current time layer only. For example, if the map of 1600 was pink and the map of 1500 orange, the dug hole that would normally be shown in pink, is displayed in orange color for one second.

The only options for the transition from one time layer to another one are, either going from one to the next deeper one, thus increasing the current time layer, or returning to the topmost time layer on mouse release. In class TimeLayerBlend, an integer representing the current time layer is increased, to be used to retrieve the current time layer object from the array.

Transition from one time layer to the next

The threshold for entering a new time layer is the "bottom" of that particular time layer and is implemented in the main class. The user returns to the top time layer on mouse release.

Enter new time layer when bottom is reached

6.3.1 Grid with grey values

The motivation for the hole in time was to imitate the digging in sand. Each particle of sand has its own "height" value. This height value becomes lower, the more one presses around this point. The same happens with digging on the screen, where the assumption is made that each pixel has its own height value, just like the particles in sand. This height value is then mapped to a clarity value, since the deeper one digs, the more revealed and clearly visible is the time layer under the lower layers.

Each point has a "height" or clarity value

The clarity value corresponds to the grey value with which the lines are drawn in the mask object. Since lines drawn with a grey value blend the content of the time layers in the blend mask filter, the darker a grey value is, the more blended they appear, and the lighter, the less blended (see Figure 6.7). White is the maximum grey value, corresponding to 1.0f, leaving the current time layer completely visible and unblended with the rest of the time layers. Black is the minimum grey value, corresponding to 0.0f, where the time layer is not at all visible.

Clarity value corresponds to grey value of drawn lines to the mask

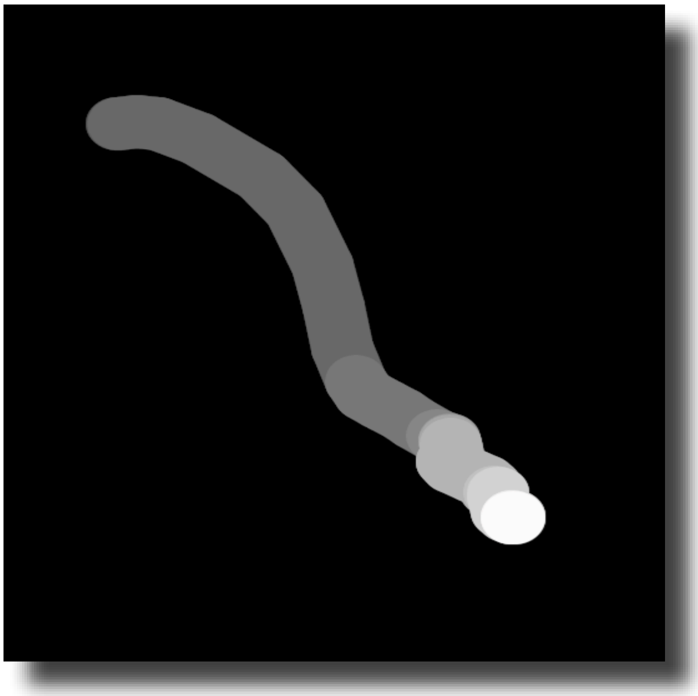

Figure 6.7: The lines in the mask drawn with a dark grey value, which means that the current time layer will appear blended with the content of the previous time layers.

Storing grey value for every block on the screen of size equal to fingertip

In order to implement the simulation of digging in sand as described above, a height or clarity value would have to be stored for each pixel on the screen in a grid. Then the height value of each of these pixels would be updated if the user passes from it with the mouse. However, based on the resolution of 1280x768 of the pictures, this would mean we would have to store and process in real-time the height value of around one million pixels, which is computationally expensive. Instead, since the finger size is what matters for the digging, a grid was made based on the finger size. Each block of the grid is equivalent to the tip of the finger, which is the line width used for the drawing of the masks. The resolution of the picture was divided with the chosen line width, therefore the size of the grid was approximated at 50x20. At first, all grey values for all the blocks are set to a minimum of 0.2f, which is equivalent to dark grey.

When the user passes again from each block with the mouse, so when the mouse is inside the the block, the clarity value of this block is increased (see Figure 6.8). Therefore, the grey value of the mask becomes gradually lighter, till it gets white, when the time layer is completely visible. The bottom of the time layer was chosen to be slightly higher that the white value, i.e., since white is 1.0f the bottom is 1.5f, so that the next time layer is not immediately shown before the previous one has been completely revealed (see Figure 6.9).

When mouse passes again from block, clarity value is increased

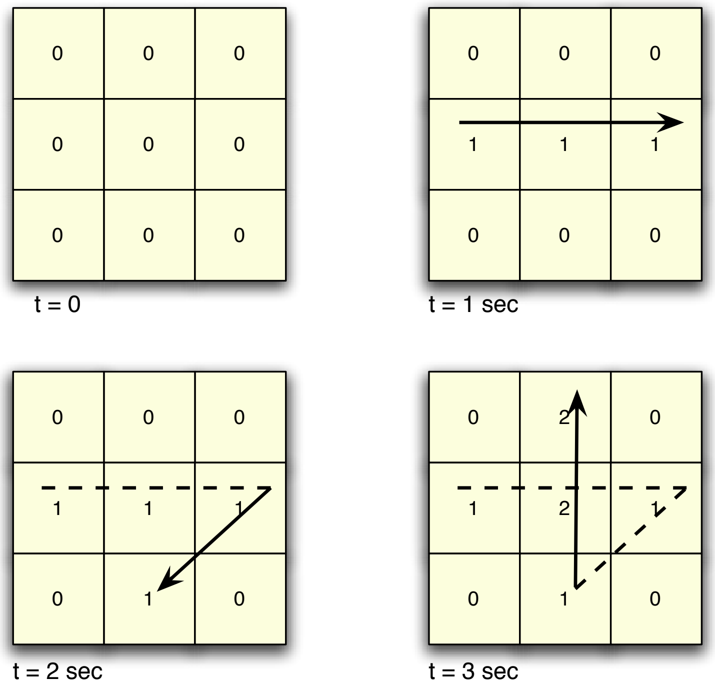

Figure 6.8: The grid of blocks storing the grey values of the mask. When the user passes again from each block, the grey value is increased.

6.4 Design for arbitrary number of diamonds

The application has to work for an arbitrary number of di-

Dynamic drawing of diamonds

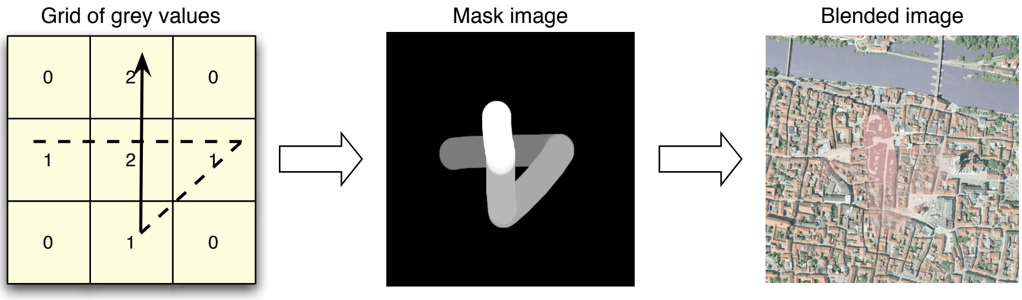

Figure 6.9: The increasing clarity values of the grid of blocks, is assigned to lighter grey values of the lines drawn in the mask. Therefore, the next time layer appears gradually more clearly visible and less blended with the previous time layers. White is the "bottom" of that time layer where it is completely clear.

amonds. A diamond is specified by its location in (x,y) coordinates and its label (and optionally its image) and the diamond has to be dynamically drawn at the specific location in the time layer it belongs to (see Figure 6.10).

Each time layer contains array of diamonds

Every time layer contains a specific number of diamonds. These diamonds are kept inside it in an array. In order for the diamond to be dynamically drawn inside the time layer, the area exceeding the contour of the diamond image is made transparent.

Diamonds drawn offscreen into time layer image

The diamonds are drawn offscreen at their specific locations into the time layer map image, in class TimeLayer. This does not affect the responsiveness of the system, as the drawing is done at the beginning of the application when the map images are loaded. The drawing is faster when it is done by the GPU, since it outperforms the CPU. Therefore the entire filtering and the drawing pipeline takes place on the GPU. CIImage and CGLayer are mechanisms provided by Core Image and Quartz 2D to accomplish fast offscreen drawing.

A CGLayer for the time layer image

TimeLayer class also contains a CGLayer for the time layer image, and a context reference (CGContextRef) in which to draw the specific time layer image.

Steps for drawing diamonds

The operations for drawing the diamonds into the time layer and returning the time layer image are:

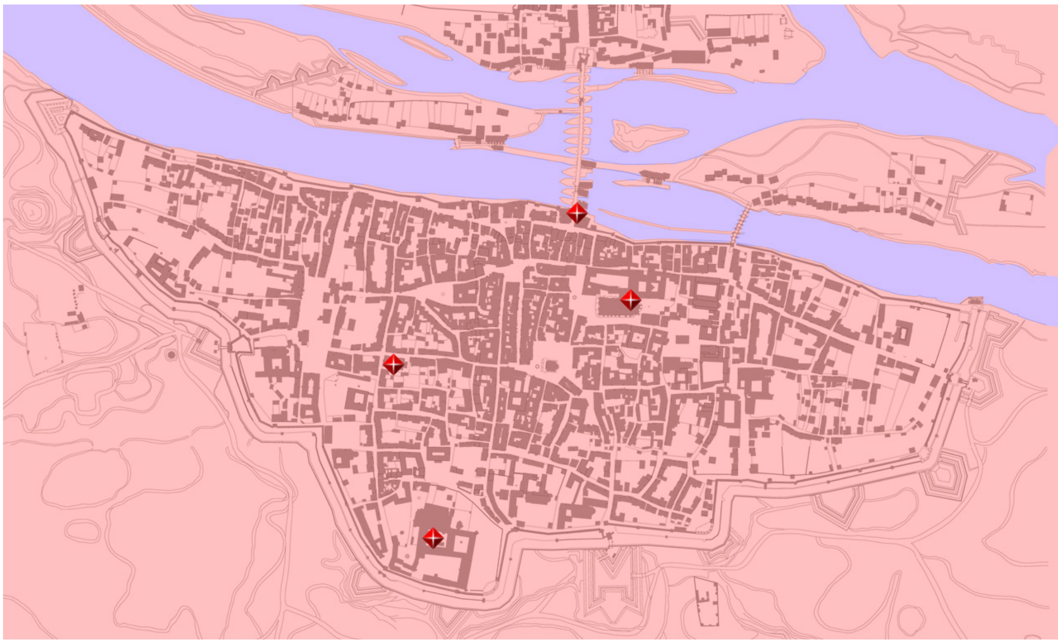

Figure 6.10: The diamonds are drawn dynamically at their specific location in the time layer image where they belong to, at the beginning of the program. Here the diamonds are drawn inside the map of 1600. Every time layer contains an array of diamonds and all diamond-related operations are called in it.

1. Specify a CGlayer and a CGContextRef for the time layer to be drawn (timeLayerImageLayer and timeLayerImageContext).

2. First draw the time layer, including its text. There is an option for either displaying the label of the time layer or not. The same applies for diamonds. In order to implement the drawing of the time layer image, specify a CGImageRef (timeLayerCGImage), which is a Quartz 2D image taken from the CIImage timelayerImage, and then draw this manually to our graphics context with CGContextDrawImage method. The CGContextDrawImage draws an image into a graphics context, by drawing a CGImageRef image at the position and size specified by a CGRect rectangle.

3. After the time layer image has been drawn to our graphics context, the diamonds array is drawn.

4. Then return the time layer image as a CIImage object after having drawn its text and its diamond.

Drawing array of diamonds into graphics context of time layer image

Inside the time layer class, all the diamonds it contains must be drawn into its time layer image. Therefore, the diamonds array must be drawn into the time layer image graphics context, which translates into drawing every diamond in the diamonds array.

Drawing Quartz 2D image of diamond into time layer image

The diamond then is drawn in the same way as the time layer image. We created a Quartz 2D image of the diamond, taken from its CIImage, and we make a rectangle starting from the diamond position and having the size of the diamond image. Then we draw this Quartz 2D image of the diamond manually to the time layer image graphics context with CGContextDrawImage. In this way, the diamond image is drawn into the graphics context on the top of the time layer image that has been already drawn.

Return a CIImage of time layer image

Then we can take the content of the time layer image as a CIImage object, from its graphics context.

Operations for diamonds implemented in TimeLayer

The detection if a diamond of a specific time layer has been revealed, has to work regardless of the number of the diamonds it contains. Therefore, the method for checking if the mouse is near the diamond in TimeLayer class, calls the checkMouseNearDiamond method of every diamond in the array. A diamond has been revealed in that time layer (stored in the variable currentDiamond) if at least one of the diamonds of its array has been revealed (with their own method checkMouseNearDiamond implemented in class Diamond).

Chapter 7

Summary and future work

Throughout the development of the REX project, Time Window will be an evolving product, rather than a finished one. The features that could not be implemented in the scope of this thesis, are mentioned in a later section, and would add to the overall experience provided by the exhibit to the visitors.

Further features

7.1 Summary and contributions

The goal of the system was to navigate along the time axis of maps and visualize the changing face of a city over time at specific places. Furthermore, as the exhibit's aim is to ultimately teach users about the history of Regensburg by entertaining them, it had to provide the possibility to view further information about historical buildings of interest.

Goal: visualize changes of city at specific places

At the early stages of the development, part of the work was to explore various interaction methods to support temporal navigation, such as a zoomable interface or a time slider. In order to explore the urban changes of a city, the interaction metaphor used was a digging metaphor, analogous to archaeological excavations that reveal past phases and artifacts of the city. This digging metaphor could best

Part of the work was exploring various interaction techniques be realized with the use of a touchscreen where the user could press on, to gradually reveal the past of Regensburg. The material used was maps of Regensburg of different time periods, referred to as "time layers". Also, the visualization of additional information about specific buildings was realized with the use of "diamonds", shown on the respective time layer at their specific position. One could click on the diamond items by tapping on them to zoom in an image or a video of the building.

Map 2D input of finger location to 3D navigation through time

For the realization of the digging metaphor, the finger location on the touchscreen was taken as an input. However, as typical touchscreens detect only one point and not different pressure levels, the problem was how to implement the three-dimensional navigation through time, based on a two-dimensional input. Different interaction techniques were examined to overcome this problem. One of them, for instance, was holding the finger still at a place in order for the temporal hole to gradually grow. Finally this approach was not chosen, due to the waiting time of users, and the "digging metaphor" was chosen for further development, where the user imitates digging through sand to reveal the past time periods.

Digging metaphor implementation

The touchscreen does not measure multiple pressure levels to assign these to different depths of digging. Therefore, the finger position was used for the digging, based on the approach that the more times one passes from the same points, the more he digs at these points. However, on a pixel-level basis, such a calculation would be very expensive computationally, so the screen was divided in a grid of blocks with size equal to the fingertip. Furthermore, these images had to be blended. The content of the hole that one had already dug had to be stored and taken as input for the further revealing of the next time layer. All time layers and masks used for the blending were stored in two arrays. However, the blending filter is itself a bottleneck to the performance of the application. Also, blurring the images to create a smooth effect of digging was too slow, and another faster way to achieve the same was needed.

REX Preview prototype

The first prototype was developed for the REX Preview, and the functionality was provided for two time layers and one diamond. Visitors tried out the exhibit for one week and their valuable feedback reflected we were headed in the right direction. Minor changes were made before the second prototype was built.

Second prototype for multiple time layers and diamonds

The second prototype was designed for an arbitrary number of time layers and diamonds. The digging metaphor was refined. The system was not evaluated for an arbitrary number of time layers, something which is advisable for the further development of the system.

7.2 Future evaluation for many time layers

First prototype evaluated informally at the REX Preview

The first prototype of the system was evaluated informally with the REX Preview. The exhibit was in general well received and met its goals of being understandable and delivering an educational message about the history of Regensburg. Minor adjustments were made for the development of the second prototype.

Second prototype has to be evaluated for many time layers

The system has to be evaluated for an arbitrary number of time layers. The goal is to find out how users perceive digging as a metaphor and if diamonds can be easily found and selected but also to test if the transition from one time layer to the next one was effective and understandable by the users. Ultimately, the evaluation has the purpose of testing if the users enjoyed exploring the urban changes of a city and discovering information about historical buildings, and if they learned something about the history of Regensburg.

Planned in autumn 2006

A second evaluation is planned for autumn 2006, but is beyond the scope of this thesis.

7.3 Extensions to the application

Additional features

The overall experience provided by the exhibit can be enhanced with the following additional features.

7.3.1 Auditive feedback

Adding sound will improve the experience

Our experiences in life are multimodal; they make use of all the senses and provide us with knowledge and understanding. The use of multiple senses enhances the overall experience provided by the exhibit. Currently the visual and tactile channels are used. Adding sound in the exhibit will improve the experience a lot and make it more realistic, natural, and enjoyable.

Sound resembling digging in sand

Firstly, the videos or audios describing the buildings of interest upon the selection of diamonds could also feature sound. Another idea is to add sound that resembles the one of digging through sand, making the digging metaphor more realistic. Furthermore, we can also apply audio feedback for each operation of the diamonds to indicate if a diamond was found or selected and to zoom in and out a video or an image. For example, if a diamond is found, there could be a click sound followed by a gleam sound.

Careful about the use of sound at public places

However, since the exhibit is in a public environment, attention must be paid about using sound, so that it does not become a nuisance.

7.3.2 Enhance the digging metaphor

Typical touchscreens detect only one point in time

Due to mainly technological restrictions, for instance the fact that touchscreens cannot measure pressure and can only detect one point at a time, the digging metaphor is not as natural as in reality.

More realistic digging metaphor

The goal of the following features is to make the digging metaphor more realistic, analogous to real digging in sand.

Digging in reality, not only one point

Using the whole hand for digging

When digging in sand, the hand does not contact the surface only at one point. One can use more than one fingers or the whole hand for digging. However, conventional touchscreens detect only one point at a time, not making possible the above scenario. A multi-touch screen, as it can detect multiple points at a time, could be used to give the opportunity to dig at more than one points at a time.

Deformable screen

No matter how visually nice and realistic the digging on a touchscreen may appear, it is only two-dimensional and one does not really push the screen deeper in its vertical time dimension. Also, a touchscreen cannot measure multiple levels of finger pressure that would determine how deep the user digs.

Digging on a touchscreen is 2D

The use of a deformable screen like that of the Khronos Projector system, described in section 2.2.1—"Khronos Projector", could improve the feeling of really pushing into the third dimension of time and provide a better sense of touch and tactile feedback.

Deformable screen could improve 3D digging and sense of touch

A technology could be a tissue-based deformable screen in combination with a camera or a vision ship to detect the deformation of the screen and a projector to project back the resulting image onto the deformable screen in real-time.

Deformable screen with a camera and a projector

Digging-depth feedback

Another useful feature for better perceiving digging metaphor is to provide feedback of how deep one has dug, which is not given in the current version of the prototype.

Digging-depth feedback

On the one hand, the two-dimensional effect of the hole in time on the flat touchscreen has to appear three-dimensional, and on the other hand, there has to be a kind of feedback for how deep one has dug in relation to the initial time layers.

3D effect of hole in time

The hole in time, in fact, does not appear really like a hole in the screen, since one time layer replaces the previous one on the top of another and looks completely flat. In order to

Shadows or 3D effects can be used

give a three-dimensional sense of depth for the hole in time, shadows and 3D effects can be used (see Figure 7.1). Also, the viscosity of sand as a material can be represented with these shadows, by showing how the sand flows back into the hole.

Figure 7.1: Shadows or 3D effects can be used to improve the sense of three-dimensional digging and give feedback about how deep one has dug (image by David Holman).

Margin with previously explored time layers

When one digs a hole in sand, the radius of the hole gradually grows and includes, at its sides, all the previous layers of sand one has already dug. This allows the user to know how deep he has dug relative to the initial topmost sand layer. In the same way, one time layer could appear inside the previous already explored time layers (see also section 4.4.4—"Temporal hole with digging-depth feedback"). Alternatively, the digging depth could be shown as a curve with the current time period at the bottom, placed at the side of the map.

7.3.3 Time period indicator

Colored time period title does not fulfill our purpose

When a user looks at a time layer, he would probably like to know which time period the current layer is referring to. In the current prototype, there is a title shown on the top of the map, indicating the time period in which users are, drawn with a specific color. This title does not give the general overview with the total number of time periods, and to which time period users are, relative to the topmost time layer.

Colored time slider

An alternative to using titles is the use of a time slider which represents the different time periods showed with their specific colors, as depicted in Figure 7.2. However, the disadvantage of this indicator is the lack of the sense of how deep one has dug.

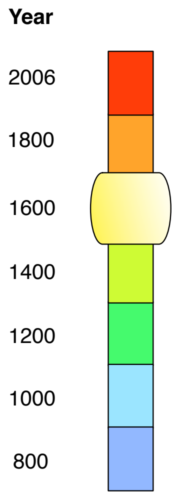

Figure 7.2: A time slider, which represents the different time periods of the development of Regensburg depicted with their specific colors. Here the year 1600 is chosen.

Curve with time periods

Another possible solution which gives the depth of digging, would be a curve representing the time periods. The present is at the top, and the time period which the users currently explore is shown at the bottom of the curve. The curve could be shown at the side of the map on the screen, or alternatively on a different screen, placed vertically to the touchscreen. However, this method also loses the general overview with all the range of colors given at the time slider.

“Tool-tips like” labels

Another option could be a small label appearing on the map at the position of the finger as the user digs, or a “tool-tips like” label. We can provide an additional device such as a pen to inquire for this information.

7.3.4 Multi-user support

Single-user application

The current prototype Time Window can be used only by a single person at a time. As long as a user interacts with the exhibit he occupies it and no other person can use it till he has finished. Sometimes other people will have to wait in a queue to use the system, depending on the visitors' throughput of the museum[1].

Advantage of wide screen

However, our exhibit carries the advantage of having a wide screen, leaving enough space for at least two more users to interact with the application at the same time.

At REX Preview, people tried to interact simultaneously

At the REX Preview, it was observed that during the interaction of a user with the system, other people would tend to touch the screen and try to interact with the application as well.

Multi-user support important

Because of all these reasons, it is crucial for future versions of the system to provide the opportunity for collocated users to interact with the system at the same time. Allowing multiple users to interact with the system introduces a variety of new problems that do not exist in the single user application and which need to be addressed, namely the interfering of the tasks of users and the time indicators for multiple users.

Interfering of tasks of different users

The first problem is the interfering of the tasks of different users. If, for example, two users select two diamonds at the same time and their images or videos zoom, they would zoom on the top of each other at the same place of the screen. One possible solution is to reduce the final size of the images or the videos, so that they do not consume considerable part of the screen and make it unlikely that they zoom on top of each other, but then less detail of the images would be visible, something which is not desirable. Other solutions are zooming the images at different parts of

[1]Design pattern "Cooperative Experience", as described in Borchers [2001], pages 113 - 116.

the screen, or using separate small displays next to the map where the zoomed images/videos can be displayed. Furthermore, if two videos are playing at the same time, the audio signals of both videos will interfere with each other. Then a strategy to solve the problem would be giving priorities to the users and waiting till the video of the high-priority user is over before the other one can play.

Time indicator for multiple users a problem

The second problem is the time indicator for multiple users, since then multiple time sliders or multiple time curves, one for each user, would be needed. Small labels representing the year of the time layer could be shown instead, at the finger position. For instance, a small number indicating the year of the time layer could appear at the position of the finger while the user digs. Multi touch screen is required for multiple users to interact with the system.

Move from wall-like to table-like exhibit

Another idea for the collocated-user application is to move from the wall-like exhibit to a table-like exhibit, by using a multi-touchscreen table (see Figure 7.3).

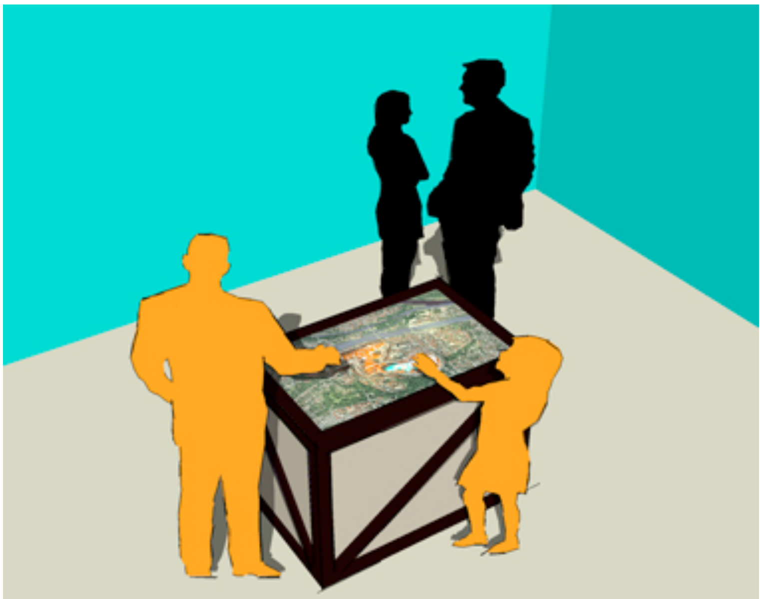

Figure 7.3: An idea is to move from the current wall-like exhibit to a table-like exhibit, using a multi-touchscreen table (image by David Holman).

The interaction with many fingers allows for other tasks as

Interaction with many fingers also for other tasks

well, except the navigation in time (e.g., use of two or more fingers in order to zoom in a map). Jeff Han from New York University introduces a technique for multi-touch sensing[2] , enabling users to interact with multiple fingers at a time to perform a variety of tasks.

7.3.5 Enhance performance

Application slower for many time layers

The high-resolution images representing the time layers, as well as the blend mask objects are stored in arrays, which consumes a lot of memory. Furthermore, for arbitrary number of layers the system has to keep track of what has already been explored. This makes the application slower the more time layers it includes. Lastly, the blending of two high-resolution images is itself a bottleneck to the performance.

Pixel-scale choice perhaps faster

A solution for future versions of the system would be performing a pixel-scale choice of the time layer to be displayed, and only at the parts where the user presses his finger. This would enhance the performance as in this case it would not be necessary that the whole images are stored and blended.

Blurring is expensive

Another source of computational bottleneck is applying a Gaussian blur to the whole high-resolution image, in order to create the effect of smooth edges of the time layers. The solution to that is using a Gaussian gradient that is painted to the mask instead of the lines.

7.3.6 Transfer to other domains

Digging metaphor used to explore other kind of layers

The digging metaphor or the concept of revealing deeper layers under the more recent ones can be used in order to explore media with multiple layers in fields of interest such as medicine, geodata analysis and art.

Example: human anatomy

For example, in medicine one could explore human

[2]http://mrl.nyu.edu/~jhan/ftirtouch/index.html

anatomy in a fun way, by revealing the different layers with the different systems of the human body, for example the skeletal, the muscular, neural, visceral, etc. When one presses his finger on a touchscreen with the human body, he gradually reveals the rest of the layers with all the different organs at the places where he touches. The purpose would be to provide users with understanding of the different functions of the body.

Or one could also do geodata analysis, by exploring geological or topographical layers. Furthermore, the same approach could be applied to explore other time-based media, like video. For example, one could have a video of the Regensburg city. Lastly the digging metaphor could be used for artistic and aesthetic purposes, for example, having different images on top of each other and explore them, etc.

Geodata analysis or artistic purposes

7.3.7 Zooming

The application at its current stage of development does not allow zooming at a specific spot of the map to get a higher resolution view of that place. Desirable features of the system that would add to the overall user experience are zooming in to a desired depth, zooming out in order to return to the initial overview of the city, as well as panning, since the screen may not suffice to fill the whole map.

Zooming not supported

The interaction without zooming is one-handed. Since the one hand is used for the navigation through time, it would be non-intuitive and confusing to use the same hand to navigate through the space of the city. Therefore, the suggested interaction in order to support the task of zooming is one that makes use of both hands in an effective and facilitating way. One possible way of interaction is using the dominant hand for digging through time and exploring how the city changed and the non-dominant hand for the task of zooming and panning.

Two-handed interaction for zooming

7.4 Suggestions from visitors' feedback

Suggestions based on visitors' feedback

The following suggestions for improvement are based on the comments and the feedback that visitors gave during the REX Preview.

Color coding

Linear order of colors

The colors of the time periods used in the application were randomly chosen. However, colors should show the age of a layer and thus afford a linear order.

Gradually decrease time layer-color

For future versions, a natural mapping would have to be defined between the range of the different time periods, moving from recent to older ones and the scale of colors they represent. The colors must be similar to each other so that one has the impression of gradually increasing/decreasing the color, as one increases/decreases the time period.

Possible ideas for selection of colors

A possible idea could be that the present is colorful and as we move towards the past it becomes more on a black-and-white scale. Another idea would be the use of red for "new" and blue for "old" and move in the color scale from red to blue, or alternatively clear colors for new ages and washed out colors for old ones.

7.4.1 Include the surroundings of Regensburg

Switch between historical centre of Regensburg and surroundings

The surroundings of Regensburg also contain some interesting historical buildings, castles, such as the Walhalla temple. However, the size of the touchscreen is not big enough to represent the entire area of Regensburg and its surroundings in the high-resolution chosen for the project (1280x768). Furthermore, a zoom mode is not provided. Therefore, we could have a two-mode content to switch between a lower-resolution map including the surroundings of Regensburg and a higher-resolution view of the historical centre.

7.4.2 Diamond icon replacement

We use diamonds to indicate the fact of discovering something precious. Some visitors find this object not suitable. To stress the time period to which an item belongs, we can use objects that are representative for that time period, for example a treasure chest or a pot of gold.

Use other object instead of diamond

One could also have medieval objects particular for the city of Regensburg of the time period we speak of. Another idea is having the same building of interest, at a smaller scale, glowing so that the user understands that it contains some interesting information.

Medieval objects or buildings at smaller scale

Bibliography

REX visitor centre. URL http://www.rex-regensburg.de/.

Swissarena. URL http://www.verkehrshaus.ch/de/museum/seilbahn-tourismus/swissarena.php.

M. Adams and T. Moussouri. The interactive experience: Linking research and practice, keynote presentation. In *Proceedings of International Conference on Interactive Learning in Museums of Art and Design*, Victoria and Albert Museum, London, 2002.

G.R. Amthor. Multimedia in education: An introduction. *Int. Business Mag*, 1992.

P. Barker. Designing interactive learning. In *Design and Production of Multimedia and Simulation-based Learning Material*, pages 1–30. T. de Jong and L. Sarti, eds. Kluwer Academic, Dordrecht, The Netherlands, 1994.

Jan Borchers. *A Pattern Approach to Interaction Design*. John Wiley and Sons, Ltd, 2001. ISBN 0-471-49828-9.

Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1983. ISBN 0898592437.

A. Cassinelli, T. Ito, and M. Ishikawa. Khronos Projector. *Emerging Technologies, SIGGRAPH, Los Angeles*, 2005.

Alvaro Cassinelli. The Khronos Projector, 2005. URL http://www.k2.t.u-tokyo.ac.jp/members/alvaro/Khronos/.

J. Dewey. Democracy and education. *Free Press*, 1966.

Apple Computer Inc. Core Image, 2005a. URL http://developer.apple.com/macosx/coreimage.html.

Apple Computer Inc. Core Video, 2005b. URL http://developer.apple.com/documentation/GraphicsImaging/Conceptual/CoreVideo/CVProg_Intro/chapter_1_section_1.html.

Apple Computer Inc. Quartz 2D, 2005c. URL http://developer.apple.com/documentation/GraphicsImaging/Conceptual/drawingwithquartz2d/dq_intro/chapter_1_section_1.html.

Susanne Jaschko. Space-time correlations focused in film objects and interactive video. *Future Cinema: The Cinematic Imaginary after Film, MIT Press*, pages 340–345, 2003.

Golan Levin. An Informal Catalogue of Slit-Scan Video Artworks, 2006. URL http://www.flong.com/writings/lists/list_slit_scan.html.

Jean Piaget. *To Understand is to Invent: The Future of Education*. Grossman, New York, 1973.

David Rokeby. The construction of experience: interface as content. In *Digital illusion: entertaining the future with high technology*, pages 27–47. ACM Press/Addison-Wesley Publishing Co., 1998.

Maria Roussou. Learning by doing and learning through play: an exploration of interactivity in virtual environments for children. *Computers in Entertainment (CIE), ACM Press*, 2(1):10–10, 2004. ISSN 1544-3574. doi: http://doi.acm.org/10.1145/973801.973818.

M.L. Ryan. *Narrative as Virtual Reality: Immersion and Interactivity in Literature and Electronic Media*. Johns Hopkins University Press, Baltimore, MD, USA, 2001.

Joachim Sauter and Dirk Lüsebrink. The Invisible Shape of Things Past, 1995. URL http://www.artcom.de/index.php?option=com_acprojects&page=6&id=26&Itemid=144&details=0&lang=en.

James Seo. Asynchrony, 2006. URL http://www.lossless.net/projects/asynchrony/.

Camille Utterback. Liquid Times Series, 2001. URL http://www.camilleutterback.com/liquidtime.html.

Index

Typeset October 11, 2006