

Post-WIMP User Interface Model for 3D Web Applications

Johnny Biström (21548C)
Alessandro Cogliati (64302H)
Kimmo Rouhiainen (55559D)

Post-WIMP User Interface Model for 3D Web Applications

Johnny Biström

HUT, Telecommunications Software and Multimedia Laboratory
johnny.bistrom@hut.fi

Alessandro Cogliati

HUT, Telecommunications Software and Multimedia Laboratory
alessandro.cogliati@hut.fi

Kimmo Rouhiainen

HUT, Telecommunications Software and Multimedia Laboratory
kimmo.rouhiainen@hut.fi

Abstract

Post-WIMP user interfaces (UI) have been proposed for application domains, in which traditional interaction methods are not suitable. One example of such domains are immersive 3D applications. In this paper, we identify and present the requirements for user interaction of a Web-based 3D application. We discuss a possible implementation technology and its limits. In addition, we propose the use of post-WIMP user interface concepts such as bi-manual interaction, interactive objects, and marking menus together with some ideas for navigation, selection and manipulation in 3D space. In order to demonstrate our proposals, a specific use case has been selected: Google Earth. This paper shows how the user interface of Google Earth can be extended and enhanced by applying post-WIMP interaction styles and evaluates the results with an usability discussion.

1 Introduction

Since the introduction of WIMP (windows, icons, menus, pointing) interfaces in early 80's, the range of commonly used applications, and the processing power of desktop computers, have increased enormously. However, we still use interfaces that have, at most, extended a typical WIMP interface with minor improvements. These old WIMP approaches do not seem to bring out the full potential of new applications and tools. Many applications are proposed to benefit from interfaces with new interaction methods, in the form of simplicity and increased user performance (Beaudouin-Lafon

& Lassen, 2000). These non-WIMP (Green & Jacob, 1991), or preferably post-WIMP, interfaces are still developing. Despite of some good studies, it is still not clear how these new interaction methods could be best used for some applications.

Applications concentrating on a three dimensional content are fundamentally different from the classical applications relying on the two dimensional desktop metaphor. They present many challenges from the interface designer's point of view. Most users have only 2D input and devices that should be mapped to a 3D environment. More degrees of freedom require also more efficient usability, since the number of tasks increases. Parallelism is also needed to enable the use of both hands at the same time.

The purpose of this paper is to address and present solutions for the problems of using post-WIMP interaction techniques to improve a 3D environment user interfaces. We present the requirements for an efficient post-WIMP UI and discuss the possible methods. We focus on the interaction of navigation, selection and manipulation of objects. We examine the special case of post-WIMP 3D interaction in the World Wide Web and discuss one possible solution, the extendible 3D (X3D) for the implementation. We use the popular Google Earth as a test case to present an unique post-WIMP UI for web-based 3D applications, and carry out an usability study to validate our results.

This paper is organized into the following sections. First, in section 2, we discuss the background for this paper, including the WIMP interfaces, X3D standard and related work. In section 3 we present the requirements for an efficient post-WIMP user interface. In section 4 we represent the current methods for interaction in 3D environment and discuss their advantages and disadvantages. Our use case study on enhanced, web-based, post-WIMP user interface for Google Earth is presented in section 5. Section 6 discusses the usability issues. In section 7 we present our conclusions and discuss the need for future work.

2 Background

2.1 WIMP-Interfaces

The WIMP model, together with the desktop metaphor, has been dominating the desktop computer user interface design since the early 80's after the Xerox PARC was born (van Dam, 1997). This new kind of interface was adapted and popularized by Macintosh and is still the dominant type of UI in modern desktop computers. WIMP interfaces have been improved with a set of new widgets during the years and even some post-WIMP features are included in the more recent applications and operating systems. Still the basic structure of a WIMP interface usually remains.

WIMP interfaces typically present the work space using a desktop metaphor (Cesar, 2005), where everything is presented in a two dimensional space with movable and overlapping windows. Application functionality can be accessed through different widgets like pull-down menus, dialogs, toolbars, palettes, radio buttons and icons. A

user gives his input to an application, normally, with a keyboard and a mouse, or some other 2D input devices. WIMP interfaces carry out the *computer-as-tool* paradigm, with the concept of direct manipulation of objects, as stated by Beaudouin-Lafon, 2004. The actions are serial in nature, so the input from the user and the output for the user are handled one at a time.

2.1.1 Limitations

Even though WIMP interfaces have been proven to be effective for simple document handling and other common office tasks, it has been argued by many (Beaudouin-Lafon, 2004; van Dam, 1997; Beaudouin-Lafon & Lassen, 2000) that WIMP interfaces can not cover all user needs today. New applications would benefit from other types of user interfaces. As the computing power has increased rapidly, new kinds of applications have emerged and old ones have gained more features. Desktop computers are not anymore our only computers used on a regular basis. Smartphones, PDAs and embedded systems present new constraints to the user interfaces, like small size and divergent input devices. Because of all the limitations of the WIMP-interfaces, new kinds of interaction techniques are emerging to provide solutions for the new challenges of the user interface design.

One of the main disadvantages of WIMP interfaces is that use of a widget is individually easy to learn, but aggregating them creates complexity, as pointed out by van Dam (van Dam, 1997). The more the complexity of the application increases, the harder and more cumbersome the interface becomes to use. Manipulating the objects and accessing the application functions using Dialog boxes, pull-down menus and direct manipulation, is not necessarily the most effective and intuitive way to perform all the tasks.

The serialized nature of WIMP interfaces restrict the user and separates the user even more from the feeling of real time working, which is desirable for many applications and tasks. The lack of parallelism on input also restricts or even prevents from using bi-manual input, which has been found more efficient for some tasks (Odell *et al.*, 2004; Balakrishnan & Kurtenbach, 1999).

WIMP interfaces are traditionally using only 2D input devices, that work well with the 2D desktop metaphor, but are hard to map into three dimensions in an intuitive way. Improvement of the processing power and the graphic processors has encouraged software developers to build more 3D applications. Despite the growing amount of 3D applications, the input devices with multiple degrees of freedom are still rare, and most applications still try to cope with a WIMP-style user interface and two dimensional input devices.

2.2 X3D

X3D is the new web 3D standard from Web3D Consortium (Web3D-Consortium, 2005), a successor to VRML (Virtual Reality Modeling Language). X3D is a file for-

mat, that can be used to store and communicate 3D data between applications and to build interactive and real-time 3D content to be presented locally or over the web. According to X3D specifications (Web3D-Consortium, 2005) X3D extends VRML with the support for XML-encoding, new graphics features, better specification to make applications run similarly with different browsers, better event model and some other improved features.

X3D provides a couple of abstract sensor models to support the use of different types of input devices and recommendations for navigation. However according to Figueroa et al. (Figueroa *et al.* , 2005) these models still provide a quite limited support for more novel navigation techniques, highly interactive worlds, complex interaction techniques, and non-traditional input devices.

X3D supports pointing devices by providing a pointing device sensor with the following nodes (Web3D-Consortium, 2005), which enable an easy input device mapping for 3D environment:

- Cylinder Sensor
- Plane Sensor
- Sphere Sensor
- Touch Sensor

Even though the X3D specification does not meet all the needs that more complex post-WIMP interaction techniques would require, it is the most promising looking technology, and there have been efforts to extend X3D towards that goal. Figueroa et al. (Figueroa *et al.* , 2005) present such extensions to X3D and discuss some similar works.

2.3 Related Work

There is some basic research done on comparing WIMP and post-WIMP user interfaces, and the applications post-WIMP interfaces are suitable for. Research on 3D applications is mostly focused on Virtual Reality domain. The number of Geographic Information System (GIS) applications available for the normal desktop user is growing, but they have cumbersome WIMP user interfaces, that do not seem to offer as fluent and efficient interaction as would be desired.

Still, to our knowledge, the user interface needs of GIS and mapping systems have not been studied. Neither are the advantages of post-WIMP interaction methods with these applications.

The X3D specification (Web3D-Consortium, 2005) is the most promising to satisfy the need for post-WIMP interaction with web-based 3D applications, including GIS applications. It proposes a set of sensors to be used in 3D applications. These principles could also be adapted to GIS and mapping applications.

All X3D browsers have an interface, which supports navigation, manipulation, and switching between the modes. However, their interface is WIMP oriented and still quite cumbersome to use. It does not allow as efficient and fluent working, as we have been accustomed to with most modern applications. XJ3D¹, Contact², Octaga³ and Flux⁴ all use one mouse, or alternatively keyboard, for navigation and manipulation. Functionality of the mouse axis or the keyboard arrows is changed by a keyboard key or a mouse button. All browsers have two or more different navigation modes.

All these browsers use toolbars and menus for other tasks than the actual navigation, including the changing of the navigation mode. The actual navigation with the mouse is done according to the position of the mouse pointer, relative to the center of the view.

3 Requirements for Post-WIMP User Interfaces

User interfaces for computer applications are becoming more diverse. Mice, keyboards, windows, menus, and icons (the standard parts of traditional WIMP interfaces) are still prevalent, but non-traditional devices and interface components are proliferating rapidly. (Bowman *et al.* , 2001)

3D computer graphics is becoming more and more popular due to the increased availability of 3D hardware and software on all classes of computers. With this new technology, new problems have also been revealed. Many studies have demonstrated that 3D applications are significantly more difficult to design, implement and use than their 2D counterparts (Herndon *et al.* , 1994). Therefore, great care must be put into the design of user interfaces and interaction techniques for 3D applications. Simply adapting traditional WIMP interaction styles to three dimensions does not provide a complete solution to this problem.

The extensive range of possible interaction techniques can be overwhelming for the developer. None of the techniques can be identified as the "best": their performance is task and environment dependent (Bowman *et al.* , 2001). So, the most important principle for 3D user interface design is to choose interaction techniques based on the requirements of the application. Therefore, requirements of this paper are based on the use case: a Web-based desktop application representing a 3D model of our solar system. The scenario of the application can be divided into three phases:

- User is free to travel through the 3D model, for instance fly in the space.
- User could be interested to explore in detail a single planet. In this case he should be in the condition to select a planet and, focusing on it, and changing the modality of travel, for example zooming and rotating.

¹<http://xj3d.org>

²http://www.bitmanagement.com/products/bs_contact_vrml/6.2/

³<http://www.octaga.com/>

⁴<http://www.mediamachines.com/playerproductpage.html>

- When user is focusing on a planet, he has access to the metadata information, like map indicators etc.

Based on this scenario, different aspects have to be considered for the requirements: Input/Output (I/O), navigation selection and manipulation.

3.1 I/O

Input/Output devices are an important component in building 3D user interfaces for virtual environment (VE) applications. Interaction designers must have a thorough understanding of the ergonomics, advantages and limitations of the devices used so they can find natural and intuitive mappings between interaction techniques and hardware. (Bowman *et al.* , 2001)

Related to our use case aural displays (audio output) and speech recognition systems (audio input) are not taken in account as possible tool for build a 3D user interfaces. As well as all those displays that usually are not used in desktop applications as haptic, tactile, and olfactory channels.

3.1.1 Output

We commonly use the term display to describe output. In the context of 3D user interfaces for VEs, visual displays can be roughly categorized into:

- *Fully immersive* displays that occlude the real world.
- And *semi-immersive* displays that allow the user to see both the physical and the virtual world.(Bowman *et al.* , 2001)

Since our use case is a desktop Web application, semi-immersive displays are required.

3.1.2 Input

Input devices can be roughly categorized based on the types of events they generate:

- *Discrete-input* devices generate one event at a time based on the user actions.
- *Continuous-input devices* generate a stream of events.
- *Hybrid-input devices* combine both discrete and continuous events to form single, more-flexible devices.(Bowman *et al.* , 2001)

Since our use case is a desktop Web application, but the goal of the paper is to create a post-WIMP UI that is more immersive and natural, hybrid-input devices are required.

3.2 Navigation

The task of navigation provides movement between distant locations. Navigation tasks can generally be classified into three categories.

- *Exploration* is navigation with no explicit target, where the user is free to investigate the environment.
- *Search* tasks involve moving to a particular target location.
- *Maneuvering* tasks are characterized by short-range, high-precision movement that is used to place the viewpoint at a more advantageous location for performing a particular task. (Bowman *et al.* , 2001)

For our use case application, the UI requires *exploration*, to allow user to navigate through different planets, and *maneuvering* ,to explore a specific planet.

3.3 Selection

Two categories of selection are choosing a direction (e.g., pointing) or choosing an item (Sherman & Craig, 2003). In our use case, the first technique is required, when user is navigating free in the space. Direction selection is useful especially as a directional indicator for travel control. The second technique is useful when user would like to focus and explore a single planet and has to select through a set of planets (items) in the environment.

3.4 Manipulation

One of the major benefits to use 3D interactive VE is the possibility to interact with the objects placed in such environment (Sherman & Craig, 2003). In our use case two kinds of manipulation occur after an object (a planet) has been selected:

- *Zooming and rotating*. This case can be re-conducted in the case of navigating in constrained environment (i.e., *maneuvering*), and has been already discussed in the chapter Navigation (3.2). Therefore, for the rest of the paper, this subject will be treated as navigation issue.
- *Control the metadata*: like map indicators etc. In this case new post-WIMP techniques are required to generate a more immersive application. Sherman & Craig (Sherman & Craig, 2003) list three different categories of performing manipulation: Direct user control that mimics real world interaction, Physical control that utilizes real devices and Virtual control that utilizes virtual devices.

In our use case, the last one seems to be the best, because can be easily implemented for desktop environments, without adding any extra input devices.

4 Comparing Solutions and Implementations

Designing 3D interfaces has not yet reached a maturity state. There are still no cohesive 3D interface paradigms similar to the WIMP paradigm in desktop user interfaces. It is only possible to consider general guidelines that are applicable from 2-D interface design: the requirements of simplicity, consistency, error prevention, and so on (Bowman *et al.* , 2001). It is for this reason that, in this paper, only a 3D interface of a specific use case is studied.

4.1 I/O

A distinction must be made between I/O devices and interaction techniques. I/O devices are just the physical tools used to implement various interaction techniques.

4.1.1 Output

There are a variety of visual displays with different characteristics and technologies. Since for our use case application a semi-immersive display is required, the most simple and least expensive solution is to utilize the so called monitor-based display or Fishtank. It is generally an extension of a simple desktop computer setup, and it can be used with many of the same interface devices as standard mouse, track ball, etc. For application in which the metaphor of peering through a window is appropriate (as in our use case), this solution can provide an effective experience (Sherman & Craig, 2003). Other semi-immersive display as projection and non-occlusive HMDs can be considered as possible alternatives.

4.1.2 Input

According to the requirements hybrid-input devices are needed, and since monitor-based display was chosen as output device, a possible solution could be to use those hybrid-input devices that mainly are already integrated with such output device. Standard mice can be an efficient solution, for example in bimanual interaction.

Bi-manual interfaces can according to the research work of Odell & al. (Odell *et al.* , 2004) improve intuitiveness by enabling more input methods and more sensory feedback. It can also improve efficiency by facilitating parallels inputs that reduces overall input time. Finally bi-manual interfaces can improve comfort by splitting workload and providing new body positions for the work. Research in bi-manual (or, two-handed) interfaces has shown that the presence of pointing devices in both hands can lead to more natural interaction, because allows the simultaneous specification of multiple parameters.

Also Hinckley from University of Virginia in his PhD work (Hinckley, 1997) reports that a well-designed bi-manual interface allows performing simultaneously subcomponents of the same task.

4.2 Navigation

Based on the requirements the UI should allow two navigation models: exploration and maneuvering. In both the models Fly-through paradigm (most generic and common method to travel through a virtual space) can be used.

- *Exploration*. In this modality user can explore the 3D environment. The movement of the viewpoint from one location to another can be achieved in two interaction techniques:
 - *Steering*: This is the most popular travel metaphor and includes techniques such as gaze-directed steering (wherein the user's head orientation determines the direction of travel). In this situation, to prevent or reduce participant looseness, constraints can be applied to user travel. Buchholz et al. (Buchholz *et al.* , 2005) propose diverse techniques for "smart" constraints. For example limit camera control in a way that the resulting view is guaranteed to contain a significant amount of orientation-supporting information, and define a maintenance strategy. For instance, in our use case, the application implementation could always check that in the actual view there is always a planet, otherwise automatically moves back to previous view.
 - *Target-based travel*: The user specifies the destination, and the system handles the actual movement. This may take the form of "teleportation", the system may perform some transitional movement between the starting point and the destination (Sherman & Craig, 2003). This travel technique needs a selection paradigm as Item selection that is explained in the following chapter.
- *Maneuvering*. In this modality user can explore/manipulate a single planet. Basically two actions can be taken: Zooming and Rotating. Both these actions can be easily performed with one input device

4.3 Selection

Based on the requirements our interest is about direction selection and item selection.

- *Direction selection*. Selection of direction is required when user is in the exploration modality, as explained in chapter 4.2 in the exploration/steering section. There are different ways of selecting a direction, but considering our use case, only one can be relevant: the so called Gaze-direct Selection. Selection by gaze depends on the direction the user is looking, and taking in account that our application is based on the Fly-through paradigm, this can be considered one of the best solutions.

- *Item selection.* Item selection methods are basically all ways to select an item from an enumerated list, which can appear as individual object in the world, groups of objects, or even as locations. In our application, Item selection is used by the user when he wants to select and focus on a planet. There are different ways of selecting an item, but only one is significant for our use case: the so called Point to select in which a pointer indicates the object selected. A pointer can be fixed in the center of the view and select a planet that it is matching. This is a good solution because user doesn't need any other controls but only needs to use the same controls of navigation and move the view in a way that the planet, he would like to select, is in the central point. An alternative way is proposed by Khan et al. (Khan *et al.* , 2005): an automatic way to select object in the environment and than pass from exploration directly to maneuvering modality. The concept is simple: when a user is moving freely in the space and he arrives really close to one object, so close to be in a prefixed orbit, the travel modality change and the user remain focus on such object.

4.4 Manipulation

In this chapter we focus on post-WIMP techniques to interact with the metadata of the application. The method is based on Virtual Control paradigm, according to the requirements.

A number of novel interaction techniques have been developed over the past decade, but few have found their way into commercial products. Beaudouin-Lafon reports that Toolglasses and Marking menus for example have been shown to be significantly more efficient than traditional palettes and menus (Beaudouin-Lafon, 2004), yet they are not available in commercial toolkits.

Odell et al. (Odell *et al.* , 2004) argue that of all the six tested control methods, Grouped Hotkeys and Bi-manual Marking Menus were found to be the fastest. Both were significantly faster than Toolbars, and neither was significantly faster than the other.

According to research made by Balakrishnan & Kurtenbach (Balakrishnan & Kurtenbach, 1999), the use of the non- dominant hand must assist the dominant and perform tasks that do not require the same precision as the dominant. According to the research made the non-dominant hand should set the frame to the work done by the dominant hand. The hands should also have asymmetric scales of motion which means that the non-dominant hand could be used for rough positioning while the dominant does the fine positioning. There is also a precedence of the non-dominant hand which means that the non-dominant hand should start the work before the dominant hand.

5 Use Case Google Earth

The use case application chosen to illustrate the advantages of a post WIMP user interface is called "Google Earth"⁵. It is a separate 3D browser application which through an Internet connection uses a multi-terabyte database of satellite and airplane imagery of the world. The user can navigate all over the globe, find places that interests him, approach them and explore them in detail. The places are described by 3D models having the pictures as textures. Metadata can be attached to the mapping information and thus a GIS covering the whole globe can be organized.

Google Earth is not the only effort in this area. NASA has a similar separate application called "World Wind"⁶. In World Wind you can select the moon as another target to study. There is also an Open Source effort called "Planet Earth"⁷ under development. This project is described by Thorne & Weiley, 2003. The platform used is called 3map and it is based on open standards. To use it you need a VRML/X3D plug-in.

Google Earth was chosen as a use case because it is well known and well documented and illustrates the problems with navigation, selection and manipulation in a clear way. The WIMP user interface is implemented in an user friendly intuitive manner because the product is aimed at a large market. Google Earth is not implemented using X3D.

5.1 Google Earth UI

Google Earth has a WIMP user interface with combined keyboard and mouse control. The application has a menu bar, a search window, a place window, a layer window and a 3D viewer window as shown in figure 1. In this paper we focus on the 3D viewer but menu choices and layer selections are also considered as they are necessary for efficient use of the application.

The first limitation of the UI of Google Earth can be found in the principle of having only one object to study. If one would like to have a look at the moon or an other planet the interface does not support true 3D as it does not allow rotation of the viewer or viewing point. There is no need for an object selection mechanism as the same object always is selected.

There is no interaction with the metadata presented in the different layers. In the best case a information box appears as you click on the data. This information box shows the address of dining places and lodgings.

The manipulation of the view and moving on earth is very limited as only one movement is allowed at a time. With the mouse you can move forward and to the side at the same time but this is the only move that involves two axis. All other moves can only be done one at a time as shown in Table 1. Partly this is due to the fact that the

⁵<http://earth.google.com>

⁶<http://worldwind.arc.nasa.gov/>

⁷<http://www.planet-earth.org/>



Figure 1: Google Earth user interface.

mouse only can point at one button at a time and partly this is due to the fact that the keyboard only recognizes the use of one shortcut at a time. Combinations of keys and mouse only registers the first entry.

It is very clear that the Google Earth application would benefit from a post-WIMP user interface and in the next section we will present a solution for it.

5.2 Improving Google Earth With a Post-WIMP UI

The WIMP UI suffers from several disadvantages when it is used in 3D environments. Interacting in 3D worlds is a more complex task than the WIMP 2D interfaces can handle. The WIMP interface offers only a synthetic vision of the interaction opportunities that can be accessed. In a 3D world the interaction involves an exploratory approach and operations typical to real worlds claims (Celentano & Pittarello, 2004).

Considering the interaction the model Celentano & Pittarello (Celentano & Pittarello, 2004) use is applicable also to the use case here. According to this model the interaction objects in the 3D world can be divided into three basic types:

- *Interactive objects* on which the user operates directly. They change their state or appearance as a consequence of approach or interaction.

Action	Possible mouse actions	Alternative keyboard action
rotate around x- and y-axis	mouse movement + left mouse button or click on 3D viewer button	up, down, left, right arrow keys
rotation around z-axis	special buttons in 3D viewer	shift + left arrow and shift + right arrow
zoom in / zoom out	mouse movement + right mouse button or slider or buttons in 3D viewer	control + up arrow or control + down arrow
tilt view	slider or buttons in 3D viewer	PageUp or PageDown keys
using menu	mouse movement and click on pull-down menu	
select visible layer	mouse movement and click on selection box	

Table 1: Basic interaction schemes of Google Earth

- *Artifacts* that are mediators of the interaction and makes it evident to the user that an object has an interactive functionality.
- *Dynamic information objects*, which modify their state of appearance, e.g. hide or show up, as the user interacts with other objects in the scene. All these interaction objects are used to replace the user based layer selection procedure existing in Google Earth.

5.2.1 Input Devices and Actions

Two standard two button mice are proposed as the primary input devices, one for the dominant and one for the non-dominant hand. They should allow continuous, real-time, parallel input from the user to the application. The buttons are oriented so that the button for each hands forefinger is the primary button and the other button is the secondary button. The primary buttons are used to select, by a click and to activate, by holding down, the two-dimensional continuous control of the mice. The secondary button is used to activate marking menus for quick selection of actions and options.

5.2.2 Navigation in a 3D Space

The new Google Earth conceptual interface implementation, developed by the authors of this paper, is based on X3D and is called “Stellar Explorer” and it opens as an animated 3D stellar view of our solar system where the user is placed in the outskirts

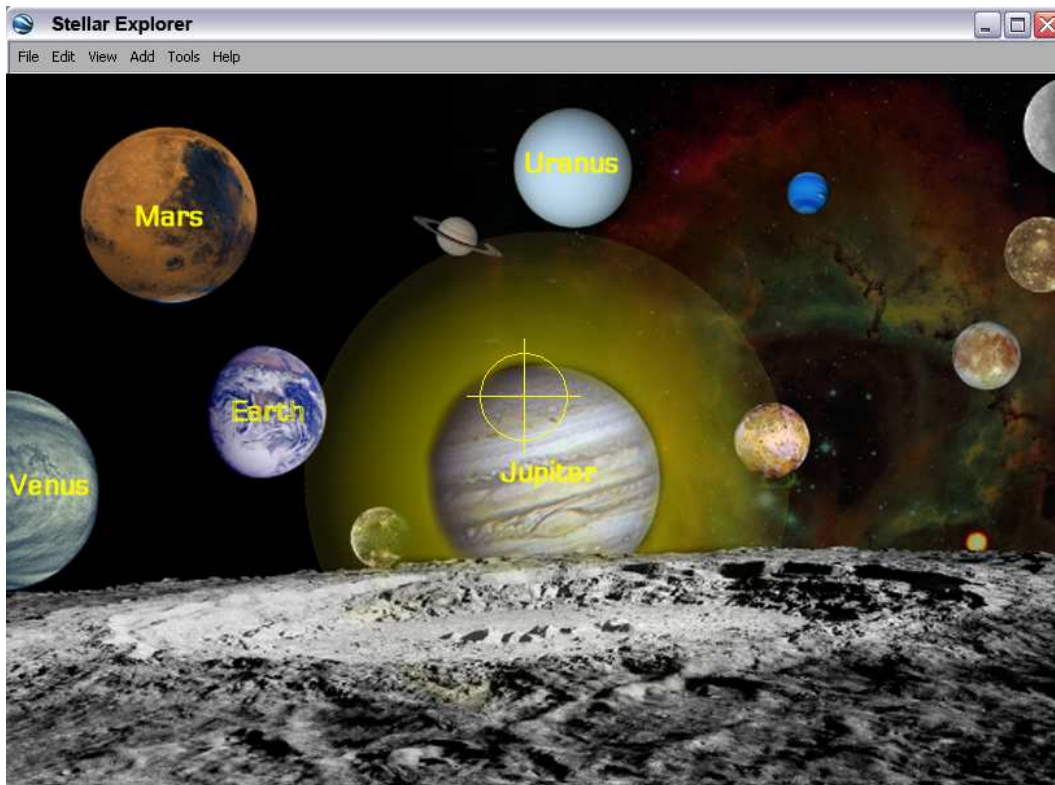


Figure 2: Stellar Explorer user interface: Fly mode

above our solar system and the sun is in the center as shown in figure 2 . The starting view is copied with permission from NASA's open picture archive. The planets are enlarged and all placed in the visible area to assist the navigation. This requires that the user is able to fly freely around the 3D space. In this navigation mode the mouse for the dominant hand acts as a 2D rudder for a spacecraft and the mouse for the non-dominant hand acts as the throttle while it moves in the vertical direction. When it moves in the horizontal direction side engines move the spacecraft sideways in the x-axis direction. The controls work as long as the primary mice buttons are pushed. A release of the rudder makes the spacecraft go straight forward and a release of the throttle slowly stops the spacecraft gradually. The user's task is then to identify the planets and to select which object to examine.

5.2.3 Selection of an Object

As the spacecraft approaches objects they appear as dynamic information objects and facts about them appear in as text areas. The spacecraft has a hair-cross in the center of the display and as objects approach it they show they are interactive objects by enlightening a corona around them. This shows that the focus is now set on the object.

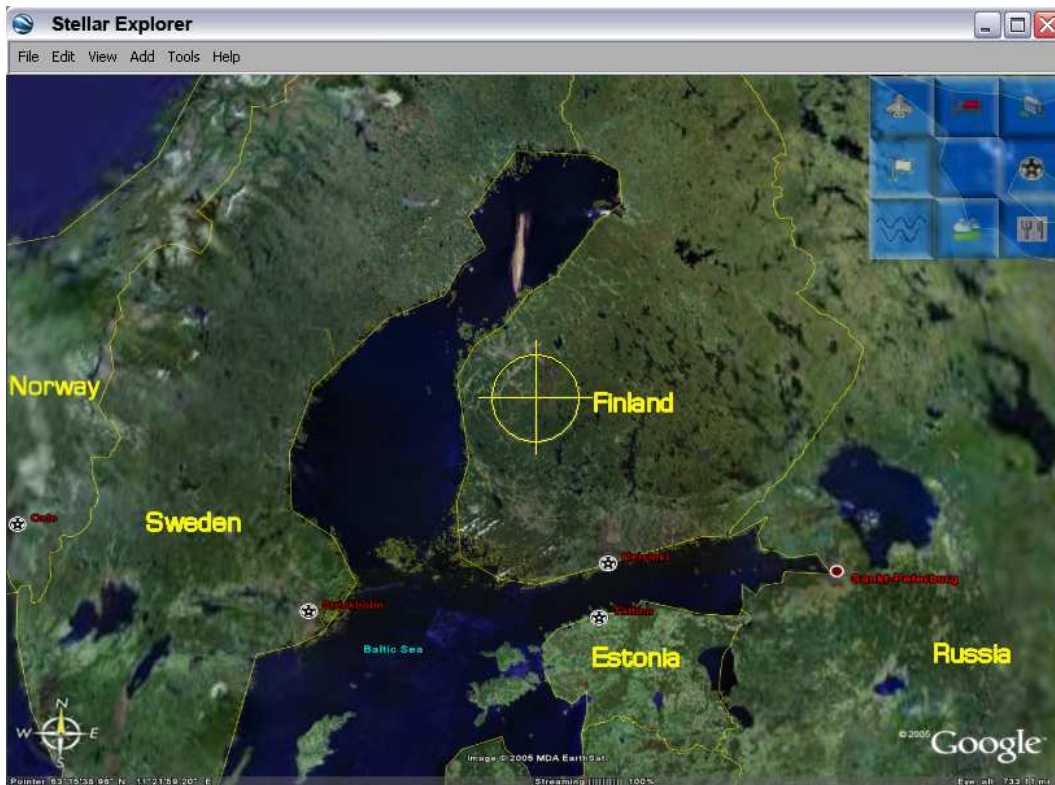


Figure 3: Stellar Explorer user interface: Manipulation mode with a marking menu

As the object has focus the user action on it can be selected from the marking menu of the secondary mouse button of the dominant hand. As no pointer is active on the screen the menu appears on the dominant hand side of the display. The non-dominant hands secondary mouse button is used for the adjusting of lightning properties as the viewing conditions as very different on the light side and dark side of the objects.

5.2.4 Manipulation

Having selected the object the functionality of the user interface changes from controlling of a spacecraft to manipulating or maneuvering a 3D object as shown in figure 3. This figure is based on the same view Google Earth has when it is started. The center of the display is set to the center of the object. The dominant hands mouse controls the rotation of the object around x- and y-axis if the primary mouse button is pushed. The non-dominant hands mouse controls the movement on the z-axis which is equal to zooming if it is moved vertically if the primary mouse button is pushed. If this mouse is moved horizontally it rotates the object around its z-axis. If the secondary button is pushed the vertical movement controls the tilting angle of the viewer so that the center of the display can be moved from the ground up to the horizon.

As the viewer approaches the object, metadata gradually becomes visible as dynamic information objects. The dominant hands secondary mouse button is used for marking menus to select categories of metadata to be shown. This view also has a hair-cross in the center of the display. As objects approach it they get focused, which is indicated by the corona and the functionality of the dominant hands secondary mouse buttons changes so that the object again can be selected for manipulation. As the object is selected, the functionality of the mice and their buttons can be adjusted so that they are as intuitive as possible.

When the user wants to change the object observed he just zooms out of the one selected and navigates to the next. This applies to both manipulation and navigation modes.

5.2.5 Summary

The user interface for "Stellar Explorer" has no need for normal menus, icons, windows and pointing devices in the way they are used in WIMP-based applications. In every situation of the 3D exploration the controls, menus and selections are based on the needs of the user and the interface is strongly intuitive. The distance to the object is essential for the level of detail and the position of the object for its selection. Interactive objects, artifacts and dynamic information objects are used to identify which objects are interactive at a certain distance. The status of the interface and the interaction possible at every stage is thus clearly indicated to the user.

6 Usability Discussion

The usability of the new post-WIMP interface has yet not been user tested, as the implementation is only conceptual at this state, but some observations made about the usability of the solution are discussed here. Usability is considered a measure of the quality of the product and it generally deals with how *efficient*, how *effective* and how *satisfying* the product is considered by its users (Nielsen & Mack, 1994). When the product is a user interface some especially important factors are: intuitiveness, magnitude of task to learn, ease of retention of knowledge, speed of performance, error incidence, ease of error recovery, customizability (Dumas & Redish, 1999).

The most significant usability differences in this case are described in table 2.

Usability consideration	Google Earth	Stellar Explorer
Intuitiveness	Typical WIMP interface is intuitive.	Post-WIMP interface must be learned to be intuitive but if the controls are designed right it is intuitive from the start.
Magnitude of task to learn	WIMP interface is easy to learn as it has guiding information on the screen.	Post-WIMP interface has no guiding information on the screen which makes it more difficult to learn.
Ease of retention	Easy, but as most applications have WIMP it can sometimes cause confusion.	Easy, when once learned.
Speed of performance	The use of one mouse and buttons for functionality is not very efficient.	Parallel bi-manual control with the use of two mice is very efficient. Marking menus are also fast.
Error incidence	Higher, as interface needs correct positioning of mouse to choose function.	Lower, as positioning to select function is not needed.
Ease of error recovery	Slower, as repositioning of mouse is needed.	Faster, as no repositioning is needed and rewinding of control corrects error.
Customizability	Can be programmed if needed.	Can be programmed if needed.

Table 2: Usability differences between Google Earth and Stellar Explorer

The user interface designed for Stellar Explorer was developed from Google Earth and it is based on experiences of several researchers in the Post-WIMP area. In the design we have tried to avoid the most common usability problems. Nevertheless, some problems are hard to overcome in the Post-WIMP design. Also the use of bi-manual control brings up some problems that do not exist in the one mouse environment.

The first of the problems is that there are no visible components (buttons, menus, icons or pointer) on the screen to support or lead the user in his interaction. The user must before he starts understand the use of the two mice and the functions of the mouse buttons. In the design however, all possible measures were taken to insure that the interface is intuitive and should not require any memorizing. All movements of the mouse correspond to actions in real life. The primary mouse button corresponds to taking grip of something. The secondary mouse button offers marking menus to select

actions or options including help.

The second problem is that the managing of two 2D controls requires good manual skills and precision also from the non-dominant hand and the ability to observe, process and control things in parallel which generally is not needed in desktop computing. Driving a car or playing a computer game requires this and quickly develops the skill to observe and act in parallel. Generally most users need some practice to master this.

The third problem observed is that moving freely in a 3D environment is not that intuitive for human beings as we commonly are locked to a plane as we walk or drive a car. Few computer users have experience of flying an airplane, a chopper or a spacecraft. The correcting of navigational errors is however easy and intuitive as the natural reflex of rewinding of the control compensates the faulty action.

Switching from one mode to another, as an object is selected, might be confusing to some users though it is very practical and efficient as the need to navigate changes. The non-dominant hand's vertical movement results in linear x-axis movement of the user in free 3D navigation mode and changes to rotation of the y-axis of the object as an object is selected. This is however probably considered quite logical by most users and some users will probably not even notice the change.

The performance of the application is expected to be better than in the original application as different commands and actions can be given in parallel compared to the original application studied. Also the proportional 2D mouse controls which can produce continuous logarithmic output are very efficient compared to using the keyboard or mouse pointed buttons for the same action.

7 Conclusions

The goal of this study was to design a better user interface for 3D Web applications than the WIMP-based 3D browsers available today can offer. Thorough studies of the current research efforts and results on different Post-WIMP-solutions lead to the combination of the most efficient and promising technologies.

According to several studies a combination of bimanual interaction realized with two mice gives a good maneuvering capability in 3D environments if the differences between the dominant and non-dominant hand is recognized and taken into consideration. A special problem in the maneuvering is that there are several different modes the user can enter as flying, walking and inspecting involved in the interaction process. The change between these modes and the mapping of the controls must be handled so that the user has the most intuitive and best control for his ongoing task.

There are also other Post-WIMP features that have proven to be very efficient such as mapped hotkeys, toolglasses, marking menus and bimanual marking menus. From these solutions we chose the marking menus as an efficient and intuitive way to give the commands needed in the interaction.

A special problem is that as there is no mouse-pointer in these applications other techniques to focus and select must be applied. Objects must indicate that they are

interactive, by highlighting themselves, as they are approached or as they are needed. The use of a hair-cross in the center of the display is a proven way to assist focusing and selection of objects.

Our contribution is that we have combined the technologies and constructed a new user interface model for 3D Web applications which also is compatible with the X3D-standard and its design principles. To test the model we have applied it on the existing Google Earth application.

Our research shows that applying these principles on the Google Earth application improves the usability significantly. There might however be some usability problems in areas as memorizing and learning the user interface. Some users may experience that the precision and parallelism needed in the control is demanding. The changes of user mode might not be intuitive to all. That is why we propose extensive usability testing to verify if these are relevant problems.

REFERENCES

- Balakrishnan Ravin & Kurtenbach Gordon. 1999. Exploring bimanual camera control and object manipulation in 3D graphics interfaces. *Pages 56–62 of: CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM Press.
- Beaudouin-Lafon Michel. 2004. Designing interaction, not interfaces. *Pages 15–22 of: AVI '04: Proceedings of the working conference on Advanced visual interfaces.* New York, NY, USA: ACM Press.
- Beaudouin-Lafon Michel & Lassen Henry Michael. 2000. The architecture and implementation of CPN2000, a post-WIMP graphical application. *Pages 181–190 of: UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology.* New York, NY, USA: ACM Press.
- Bowman Doug A., Kruijff Ernst, jr. Joseph J. LaViola & Poupyrev Ivan. 2001. An Introduction to 3-D User Interface Design. *Presence*, **10**(1), 96–108.
- Buchholz Henrik, Bohnet Johannes & Dollner Jurgen. 2005. Smart and Physically-Based Navigation in 3D Geovirtual Environments. *Pages 629–635 of: IV '05: Proceedings of the Ninth International Conference on Information Visualisation (IV'05).* Washington, DC, USA: IEEE Computer Society.
- Celentano Augusto & Pittarello Fabio. 2004. Observing and adapting user behavior in navigational 3D interfaces. *Pages 275–282 of: AVI '04: Proceedings of the working conference on Advanced visual interfaces.* New York, NY, USA: ACM Press.
- Cesar Pablo. 2005. Tools for Adaptive and Post-WIMP User Interfaces. *New Directions on Human Computer Interaction.*

- Dumas Joseph S & Redish Janice C. 1999. *A Practical Guide to Usability Testing*. Intellect Books.
- Figuroa Pablo, Medina Omer, Jiménez Roger, Martínez José & Albarracín Camilo. 2005. Extensions for interactivity and retargeting in X3D. *Pages 103–110 of: Web3D '05: Proceedings of the tenth international conference on 3D Web technology*. New York, NY, USA: ACM Press.
- Green Mark & Jacob Robert. 1991. SIGGRAPH '90 Workshop report: software architectures and metaphors for non-WIMP user interfaces. *SIGGRAPH Comput. Graph.*, **25**(3), 229–235.
- Herndon Kenneth P., van Dam Andries & Gleicher Michael. 1994. The challenges of 3D interaction: a CHI '94 workshop. *SIGCHI Bull.*, **26**(4), 36–43.
- Hinckley Ken. 1997. *Haptic issues for virtual manipulation*. Ph.D. thesis, Charlottesville, VA, USA.
- Khan Azam, Komalo Ben, Stam Jos, Fitzmaurice George & Kurtenbach Gordon. 2005. HoverCam: interactive 3D navigation for proximal object inspection. *Pages 73–80 of: SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*. New York, NY, USA: ACM Press.
- Nielsen Jakob & Mack Robert L. (eds). 1994. *Usability inspection methods*. New York, NY, USA: John Wiley & Sons, Inc.
- Odell Daniel L., Davis Richard C., Smith Andrew & Wright Paul K. 2004. Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. *Pages 17–24 of: GI '04: Proceedings of the 2004 conference on Graphics interface*. School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society.
- Sherman William R. & Craig Alan B. 2003. *Understanding Virtual Reality*. Morgan Kaufmann publishers.
- Thorne Chris & Weiley Viveka. 2003. Earth's avatar: the Web augmented virtual earth. *Pages 1–1 of: GRAPH '03: Proceedings of the SIGGRAPH 2003 conference on Web graphics*. New York, NY, USA: ACM Press.
- van Dam Andries. 1997. Post-WIMP user interfaces. *Commun. ACM*, **40**(2), 63–67.
- Web3D-Consortium. 2005. *X3D Specifications*. available at <http://www.web3d.org>. referenced at 11/2005.