

Interactive 3D models – From 3ds max to VRML

Sanjin Jeginovic

Electrical Engineering Faculty
University of Sarajevo
Bosnia and Herzegovina
sanjin.jeginovic@etf.unsa.ba

Abstract

Producing 3D interactive models is becoming a greater challenge every day. Choosing the right tool to handle the modelling process is essential if the final product is to be a VRML world, which can satisfy the user's desire for both interactivity and realism. **3ds max** offers a potentially excellent development environment for creating high quality 3D models. This paper discusses how the tools and techniques available within **3ds max** can be harnessed to produce complex interactive models, which are viewable with a VRML browser.

Keywords:

VRML, Virtual Reality, interactivity, **3ds max**, modelling, 3D models

1. Introduction

Chapters 1.1 and 1.2 set out the scope and motivation for this paper. Chapter 1.3 describes the working platform, testing platform and the software packages used in our projects. Chapter 2 provides a brief overview of **3ds max** modelling software and chapter 3 offers an overview of VRML language. Chapter 4 gives a short description of what interactivity in virtual worlds means and considers the user expectations of interactivity.

What **3ds max** offers for achieving interactivity in 3D models targeted for use in VRML and how we used this functionality in our projects, is brought together in chapter 4.1. When all objects are placed and scene is constructed (Chapter 4.1.), next significant step is to export the scene into VRML. Chapter 5 explains how this is done. Chapter 6 is a discussion of our experience

derived from our work on the two projects. Chapter 7 is the conclusion and contains our final words on the topic.

Finally, we end this paper with references to the technical literature we used and to some helpful links on World Wide Web.

1.1 Scope

As the processing power of the average computer grows, in seeming fulfilment of Moore's law, so do the expectations and requirements of the average user. The demands placed on Virtual Reality programmers can be described in one word, "More": More realism, more interaction. What was excellent yesterday is unacceptable today. For practitioners who model Virtual Reality worlds, the continual demands from users mean more hours spent optimising the code that describes these virtual environments. The modelling process is becoming more of a nightmare than an inspiring and creative challenge. So how do we shorten the modelling process and, at the same time, achieve the necessary level of interaction? One solution is to combine the use sophisticated tools, designed to support the creation of high quality models, with the techniques necessary for achieving interactivity. One such tool is **3ds max**. In this paper, we investigate the full scope of the **3ds max** package and we demonstrate how best the package can be exploited to achieve interactivity in 3D models targeted for use as VRML worlds.

To illustrate our findings, we will introduce two compelling examples:

- *Alien world* – A fictional 3D space containing an alien entity within.
- *Virtual Sarajevo* – A virtualisation of real urban space - in this case, part of the city of Sarajevo.

This project was inspired by the work of J.Zara [7, 8, 9]. *Virtual Sarajevo* was intended to be placed and viewed from CD-ROM media.

In this paper we will not discuss modelling techniques or techniques for code optimisation. This is out of the scope of our work and moreover detailed explanations can be found in the technical literature.

1.2 Motivation

When creating VRML worlds, the first choice a programmer must make is how to do the modelling part. There are three ways:

1. Visually create and edit the virtual world using specialized tools such as **3ds max**;
2. Create and edit VRML code with the help of a line editor (such as Notepad); or,
3. A hybrid of methods 1 and 2 which means switching back and forth from visually oriented tools to line editors.

Our goal was to achieve high quality interactive 3D models using the first method - depending only on the visual environment of the **3ds max** modelling software.

The objectives were:

- To minimize modelling time by using the visual environment for creating models. This approach would reduce the time spent in the coding phase of our projects. As a bonus, debugging time would be reduced to a minimum.
- To achieve necessary interaction in the created virtual environment through innovative use of the tools provided by the modelling software.
- To increase and extend the quality of models currently possible in a VRML world.

1.3 Working environment

Hardware: For this project we used computers with the following characteristics:

- Pentium III class processor
Clock speed of 733 MHz
- 256 MB, 133 MHz SDRAM
- 20 GB ATA100 HDD
- ATI Rage 128-based graphic card with 32 MB RAM
- Creative Labs sound card

Software: The following software was used:

- Windows XP professional with Service Pack1 and patches to release 7.10.2003
- Cosmo Player version 2.1.1
- **3ds max 5**

2. 3ds max

3ds max is a integrated environment for creating professional-quality 3D models. Photo-realistic still images and film-quality animations can be created on a PC using these models. **3ds max** supports the construction of various kinds of 3D objects. Landscapes of *any* type can be created from these objects and characters. All of these objects can be arranged in settings and environments. These environments form the basis of the scenes. The characters and objects can be animated - setting them in motion, making them speak, sing and dance or kick and fight. Then - the whole virtual world can be captured as a film sequence.

3ds max is a rich modelling environment that supports a wide range of modelling techniques from low polygon modelling to modelling with compound objects to mesh modelling - all the way to NURBS modelling. Once our basic models have been constructed, we can assign different textures to improve the user's perception of realism. To complete the scene, different light sources (there are various kinds) can be added to illuminate the scene. Cameras can be added to capture other parts of the scene at different time intervals.

The scene can then be used in variety of ways - in animations or as a new Virtual World.

In our projects, the geometry was created first. Texture maps and materials were then assigned to this geometry. In the case of *Virtual Sarajevo*, materials for objects were acquired from digital pictures of actual objects. Finally, lights and cameras were added to construct a full scene. Figure 1 shows the constructed scene of *Alien World*, while figure 2 shows *Virtual Sarajevo*.

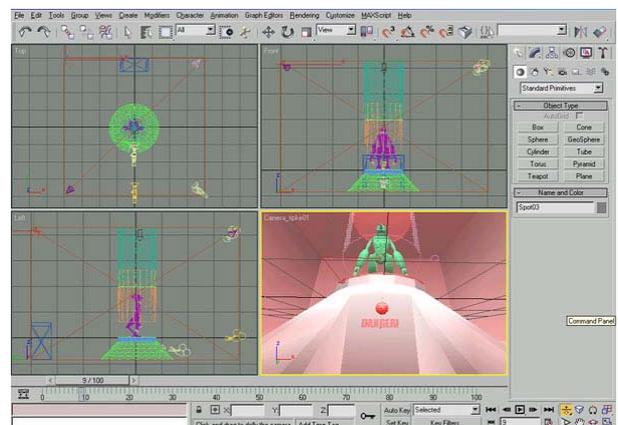


Figure 1 – *Alien World* scene



Figure 2 – *Virtual Sarajevo* scene

In *Alien World*, the alien is enclosed behind a glass cylinder, which is placed inside a room. Under the cylinder there is a little red button. The intended interaction is that, when a user presses this red button, the cylinder rises and the alien is animated by waving its hands. For construction of this scene, combination of mesh modelling and NURBS modelling was used with the application of different modifiers. For animation, a simple transform was applied.

In *Virtual Sarajevo*, more objects were used and this make the scene far more complex. Buildings were created using elementary shapes such as boxes, cylinders and spheres. Other virtual cities are handled the same way - for example, Prague [7]. For some objects, live photographs were used as textures. But we did encounter objects where the geometry made it difficult and impractical to apply photos as materials. In such cases, we tried to model the details on objects in order to achieve high level of realism.

3. VRML

VRML stands for Virtual Reality Modelling Language. VRML is an ASCII-based open, non-proprietary language. This means that it can be used by anyone without licensing. It has been officially adopted by International Standardization Organization (ISO). The current version of this language supports animation, spatial sound, collision detection and scripting. Virtual reality is defined by a VRML file which consists of a multi-tree of nodes. Each node is of a predefined type but note that new types of nodes can be created.

The *Virtual Reality* aspect of VRML is centred on the metaphor which it pursues: human space. That space is 3-dimensional and defines the ways we move in it, perceive it and interact with it. VRML accordingly includes many of the things that are required in making the virtual world: a way of describing the geometry which creates the objects and spaces we move around in - light, texture and sound. We can approach and view the

objects from different angles. We can hear the sound from different positions.

The second part of VRML, *Modelling Language*, describes the process of making a virtual world. While much of VRML is composed and optimised in text, VRML authors typically use other modelling programs for designing and creating 3D geometry in a graphical format. Later, these models, which will constitute a Virtual World, are translated to VRML and viewed using a VRML browser. The browser renders the VRML in real time and turns the code into perceivable space with which we can interact.

3ds max supports the modelling process in two ways:

1. with a modelling environment capable of producing high quality 3D models
2. with VRML helper objects

Helper objects are very important in VRML worlds. Without them, the user is still able to move around the virtual world but interaction is limited to looking at objects from different angles. Integration of sound, touch and other spatial perceptions can only be accomplished by using helper objects - which significantly extend interactivity.

4. Interactivity

Interactivity in Virtual Reality worlds traditionally means employment of *three* of our five senses:

- sight
- hearing
- touch

Combining targeted impressions on these senses produces perceptions of space and interaction with objects located in that space. For example, incremental enlargement of an object produces a perception of movement towards that object.

What a user expects to do in a virtual world is to move freely about, manipulate objects as one does in the real world and experience a spatial sense of sound. A spatial sense of sound means that the sound has a source fixed at a single point (a node) in the virtual environment. Moving towards that source increases the volume of sound and moving away decreases the volume. Also, the perception of the source of the sound is experienced relative to the user's position in the world. By shifting position in the virtual environment, the user's perception of the direction of the sound source changes.

User interaction in VRML world is gained by using standard predefined VRML sensors. Sensors can

be considered as special kind of nodes designed to react when properties of the Virtual Environment change or when a user operates a sensor in a predefined way.

4.1 VRML helper Objects in 3ds max

In **3ds max**, interaction in a VRML world is achieved by adding VRML helper objects. During export, the helper objects become VRML nodes. The following helper objects are predefined in **3ds max**:

Anchor - creates a link in a VRML file. The link is embedded in an object;

TouchSensor – (touch sensor) represents an area of sensitive space that, when touched by user, triggers event. There is a difference between Anchor and TouchSensor!

Sound and Audio clip – these two helpers allows the insertion of spatial or ambient sounds in a scene. Sound helper is always used in combination with Audio clip, while Audio clip can be combined with TouchSensor.

ProxSensor - (proximity sensor) represents an area of sensitive space that, when activated by approach of user, triggers an event;

TimeSensor – (time sensor) is used for adding time-based animation controls - such as the start and end frames for a particular object's animation;

NavInfo – (navigational information) provides navigational information of virtual space. This helper directly influences the way a user moves around the virtual world. Every scene should have this helper included;

Background - the Background helper allows the creation of a Sky and/or Ground backdrop for the virtual world. This produces very simple and plain “Earth and sky” perception. At the time of writing this paper, we are experimenting with using different types of objects and modifiers to produce effect of ground elevation. For *Virtual Sarajevo* projects, this is essential - Sarajevo is a city nested between several Olympic mountains;

Fog - enables specification of the colour and range of fog in a VRML world;

LOD - (Level of Detail) Lod helper allows for different levels of detail and complexity in an object depending on a user's distance from the object. When the user is closer to the object, the browser renders the object with a higher quality - producing more details. This helper can be used for optimisation of performance;

Billboard - permits creation of geometry that is camera-aligned in the VRML97 browser;

InLine - lets the user reference another VRML97 file. This second file would be included in the user's world when loaded into the VRML97 browser;

Detailed explanations of these helpers and their properties can be found in [1].

In *Alien world*, several sensors were used. TouchSensor (Figure 4), TimeSensor, NavInfo and FOG (Figure 3). Touch sensors were connected to the button that starts the animation of the rising cylinder and exit point of the world. On activation of the button, the cylinder animation commences, followed by animation of the alien. Activation of the sensor at the exiting point of the world enables the user to leave Alien World. The NavInfo node provided information to VRML browser on:

- What type of navigation to allow, in this case *FLY*. **3ds max** allows the NavInfo helper only one selection: walk, examine or fly. However, after exporting, the VRML file can be edited manually to include multiple types of navigation. In such an example, NavInfo would look like this:

```
DEF NavInfo01 NavigationInfo {
    avatarSize [0.25, 1.6, 0.75]
    headlight FALSE
    speed 3
    type ["EXAMINE", "FLY", "WALK"]
}
```

- If the user can turn on a *headlight*, set to FALSE in our projects, then Headlight will illuminate objects directly in front of the user. This option is useful when there is no other light included in the scene. Otherwise the option should be turned off.
- How far the user can see – this option was turned off and FOG helper was used instead
- How fast the user can move
- The size of the virtual user (the *avatar*). This is required to detect collisions. This also influences user perception of how big certain objects are eg., users expect when approaching a door, to see the upper third of the door - in reality, to see the door at eye-level.
- How close the user can get to something before colliding with it
- How high above the terrain the *avatar* is.
- How high the *avatar* can step. This is an efficient way of controlling where user can and cannot go. In *Virtual Sarajevo*, we use this information to prevent user from trying to reach unfinished parts of the town.



Figure 3 – Use of the NavInfo and Fog helpers

As we can see in Figure 3, the Fog helper was used to simulate atmospheric effects.

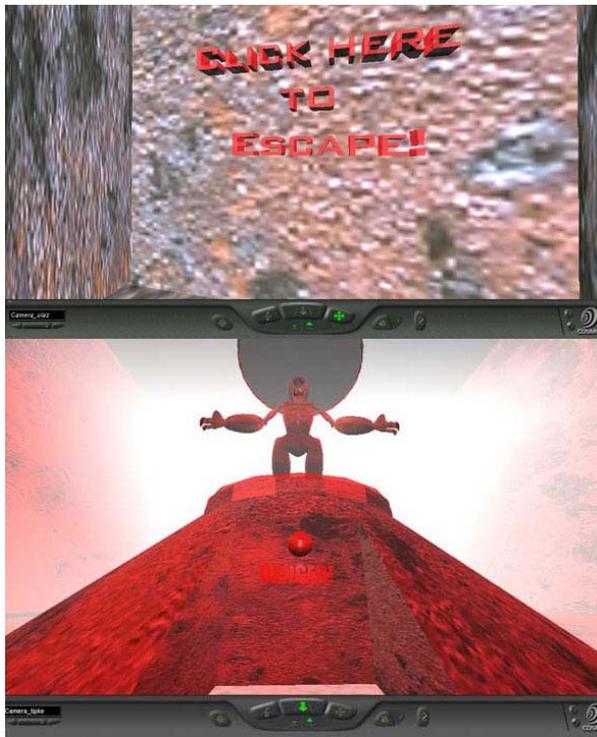


Figure 4 – Use of the TouchSensor helper

The TouchSensor was used instead of Anchor because it allows the starting of a sequence of animations. In this case, the rising of a cylinder and the animation of an alien.

Virtual Sarajevo uses the NavInfo and Background helpers. The result of using one colour background can be seen in Figures 5 and 6.

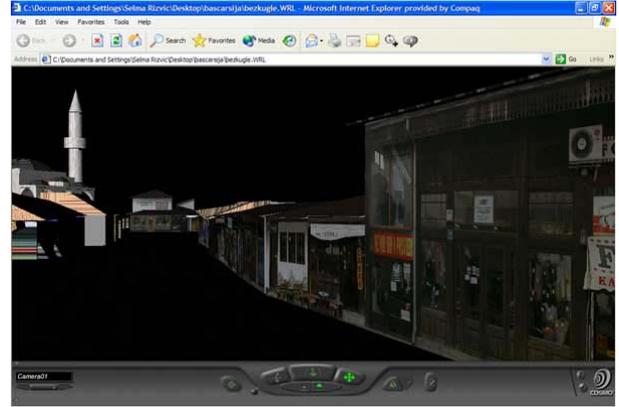


Figure 5 – Without the Background helper

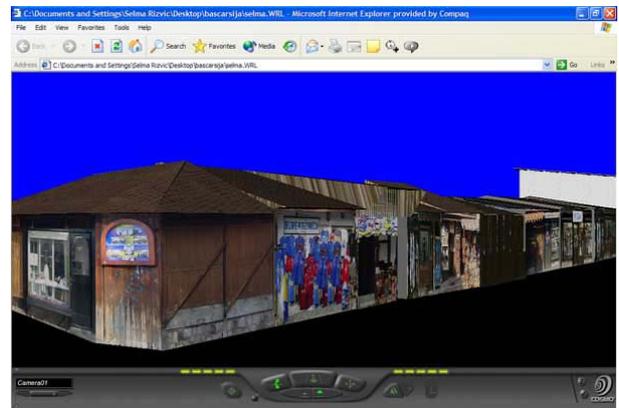


Figure 6 – With the Background helper

One of the purposes of the *Virtual Sarajevo* project is to provide information about well-known town areas and specific objects. The idea was that this information would be presented to the user via a sound file. The user could reach a specific object and, by clicking on it, start a sound file, which would provide information on that building/object. We have started by introducing short audio clips to a few objects and linking them to different touch sensors. For example, in the old town district (Stari Grad) of Sarajevo is a famous landmark (object) called the *Sebilj* (see Figure 7). The Sebilj was built in late 19 century as a public fountain for drinking water.

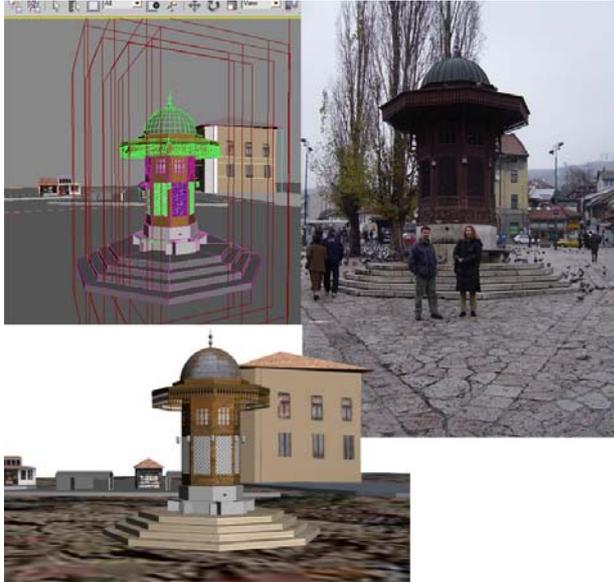


Figure 7 – Sebilj - from real image to model and finally to VRML

To further expand interactivity in *Virtual Sarajevo* we are introducing animation and spatial sound. The animation of a *Virtual Tram* is currently under construction. To this node we are planning to attach a sound node. We hope thereby to produce the spatial perception of tram movement.

5. Exporting 3d scenes from 3ds max to VRML

Once a scene is created and everything is in place, the scene is ready to be exported to VRML. **3ds max** has a built-in VRML exporter. In version 5, **3ds max** creates VRML files (.wrl) that are compatible with the VRML97 standard. When exporting to VRML, the user has the opportunity to configure a number of important parameters in the VRML97 Exporter dialogue box shown in Figure 8.

Here is explanation of the options and their use:

- *Normals* - Creates real normals for objects, potentially making VRML files larger. This option is used for exporting 3ds max geometry, which uses smoothing modifiers. When used, it significantly improves the visual quality of objects, but, as stated, makes VRML files *larger*. This option should be used in combination with the Primitives option for the best results.
- *Indentation* - Indents the VRML source code to make it easier to read.
- *Primitives* - Exports VRML primitives instead of **3ds max** primitives. VRML predefines certain types of objects - such as sphere for

example. When this option is turned on, **3ds max** tries to export these objects as VRML primitives. Using this option *reduces* size of the VRML file. In our projects, we used this option in combination with the normals option.

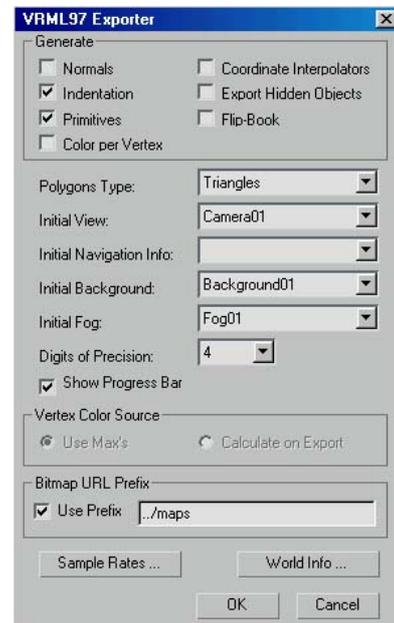


Figure 8 – The VRML97 Exporter dialog box

- *Color per Vertex* - Exports vertex colors for geometry. If this option is used, the Vertex Color Source section below can be used to tell the exporter whether to use max's vertex colors or to calculate colors on export, based on lighting and the object's materials. For our projects, we did not need this option because objects were mostly covered by different materials stored as images.
- *Coordinate Interpolators* - Enables export of animations that use coordinate interpolation, not just simple transforms. This option is mandatory with certain modifiers such as Morph. Using this option can result in very large VRML files, because the exporter has to calculate position of every vertex. Coordinate interpolation is incompatible with certain types of animation, such as animating a mesh that changes size between frames. The exporter gives warning, if this option was checked and an incompatible type of animation is detected.
- *Export Hidden Objects* - If this option is checked, the exporter will export hidden objects which will be visible in the VRML file.

This option is sometimes confused with the Renderable option in certain objects such as Splines. Splines are by default not rendered, and so are also hidden. If Splines are used, the Renderable option

should be turned on. But this has no connection with the Export Hidden Objects option in the Exporter dialog box.

- *Flip-Book* - Exports the scene to multiple files at a sampling rate set in the *Animation Sample Rates* dialogue box.
- *Polygons Type* - Determines how geometric faces are written out as VRML IndexedFaceSet nodes. In our projects we used *triangles*, which writes only triangular faces. This is default for this option.
- *Initial View* – Sets the entry camera for the scene and controls what first appears when the user opens the VRML file in the browser. For this option, at least one camera should be introduced into a scene. This is why we previously advised that any scene should have at least one camera. Adding cameras into scene does *not* noticeably increase size of VRML file. If more cameras need to be included in a scene, the user will be able to switch camera views in the browser. For the initial view, any type of camera can be used, not only target cameras. Both our projects use several cameras.
- *Initial Navigation Info* – Specifies the Initial Navigation Info helper object to use with the world when it is loaded in the browser. This is why we stated that every project should have NavInfo helper included.
- *Initial Background* - Select the initial Background helper object to use with the virtual world when it is loaded with the browser.
- *Initial Fog* - Select the initial Fog helper object to use with the virtual world when it is loaded with browser.
- *Digits of Precision* - Decimal points used for calculating dimensions.
- *Show Progress Bar* - View the progress of the process as the VRML file is created.
- *Vertex Color Source* - Source of colors for vertex colors. This option is available if the *Color per Vertex* option is used.
- *Bitmap URL prefix* - Specify the location for texture bitmaps assigned to objects. If this is not checked, then the bitmaps must be in the same location as the world file.
- *Samples Rates* - Specify sampling rates in frames per second. Higher numbers reduce the number of key values and the VRML file size.
- *World Info* - Some browsers can display this information. This does not affect the appearance or behavior of the world.

Detailed explanation of each of these properties can be found in [1, 6].

6. Discussion

Based on our work on *Alien World* and *Virtual Sarajevo* we have drawn the conclusions, which will be presented in the following text

Creating models using visual environment of **3ds max** does shorten the time of coding but only in cases where good model design was created beforehand. A general rule of thumb is that before to any coding begins, there should be an analysis time. The same applies here. Model should be first drawn on paper and only then modeled in modeling environment. Otherwise, too much time is spent in the *search for good design*.

The interactivity achieved in our created worlds was found to be satisfactory. Animation and spatial sound were extremely easy to create and functioned in the required way.

Working with sensors that are linked together, (see above in examples of *ToucheSensor* and animation and *TouchSensor* and *Sound* node) showed that the automation of creating ROUTES in VRML functions in the required ways. This is very useful when working with scenes containing many different sensor nodes.

The export utility that is integrated in **3ds max** showed satisfactory results. All primitives were exported with high-quality results. Some results were achieved by exporting extended primitives with slightly bigger file sizes. Compound objects (Boolean type and Loft type) were also exported without problems and the size of exported files was shown to be directly dependent on the objects that constructed the scene, and much less from the operation used. NURBS objects were also exported faultlessly, and exhibited high-quality result. Objects that cannot be exported are: Particle Systems and Space Warps. These objects would have been excellent for producing different weather conditions. All exported objects maintained most of their modifiers. Although modifiers increase quality of the final VRML world, they should be used sparingly, since they considerably *increase* the size of final VRML file (in some cases we observed this increase to be over 100 times!). For some applications even better export results might be needed. In this case, several third-party export utilities have been developed for specific need. These utilities come in the ready to use form of plug-ins.

The approach we have adopted produces high quality 3d interactive models. The VRML files obtained through this approach, however, tend to be *large*. If this approach is to be used for Virtual Environments viewed via such medias as the internet, then optimization methods are required. These would include controlling the number of polygons in the scenes, using simple transforms for animation, optimizing material maps and so on. Our method, however is excellent for promotional

materials that can be distributed via media such as CD-ROM and DVD.

Although it proved sufficient for the needs of our projects, **3ds max** only provides a few predefined sensors. This problem could probably be circumvented by use of the MaxScript - a scripting language integrated in **3ds max**.

Probably the greatest advantage of this approach is that VRML worlds can be created without any great expertise in VRML. The only prerequisite is knowing how to model in the **3ds max** environment. **3ds max** takes care of the rest - but even so, optimization of the code is sometimes necessary and unavoidable.

7. Conclusion

The era of programming by writing code in a code editor is long gone. Demands on programmers are pushing them towards rapid code development. Shorter life cycle development and easier maintenance of developed systems can be achieved through different approaches ranging from computer-aided development to rapid prototyping and model development.

Demands from captivated users are driving virtual reality modellers to explore the modelling environments that can support high quality modelling and that possess integrated tools for achieving interactions in Virtual Reality worlds. Choosing a working environment is not an easy task. The chosen software tool may shorten the production period but, at worst, may considerably prolong it. If the modelling environment is too complex, the programmer will require training and practise time before becoming productive - time that we simply do not have.

3ds max is a *premium* modelling tool. Most people engaged in modelling have encountered the package at one point or another. This software is able to produce high quality and complex models and - what is more - produce them rapidly. It has a built-in VRML exporter that handles all elements of the constructed scene. As we have seen in this paper, it has built-in objects and helpers, which, if used alone or combined with one another, create the necessary perceptions of interactions. We can conclude that **3ds max** presents an excellent environment for creating 3d interactive models, which can be explored in VRML.

References

- [1] Discrete **3ds max** user reference
- [2] Discrete **3ds max** tutorial reference
- [3] *3ds max 4 Bible*, Kelly L. Murdock, Hungry Minds Inc., 2001;
- [4] *Mastering 3ds max 4*, Cat Woods, Alexander Bichalo, Chris Murry, Sybex publishing house, 2002;
- [5] *Web Publishing Unleashed*, William Robert Stanek, Sams.net Publishing, 1996;
- [6] *3DStudioMAX To VRML* Exporter Tutorial, <http://www.dform.com/inquiry/tutorials/3dsmax/>
- [7] Official page of J. Zara, <http://cs.felk.cvut.cz/webis/en/people/zara.html>
- [8] *Virtual City of Prague*: <http://www.cgg.cvut.cz/vsp/>
- [9] *On the complexity of web based presentations of large urban scenes*, J. Zara, Graz, EWV'02;
- [10] Data Acquisition for Virtual Old Prague, P. Bouchner, CESC 2002;
- [11] 3D Graphics and VRML2, L. Lemay, J. Couch, K. Murdock, Sams.net Publishing, 1996;