

A tool for 3D orientation models generation

Pedro M. Chaves

Fernando A. L. Silva

Jan M. Corrêa

Carla D. Castanho

Department of Computer Science
University of Brasília

Abstract

This paper describes a solution to simplify the generation of 3D orientation models. Our main goal is to reduce the effort in generating such models, eliminating the need of previous programming and modeling knowledge by the user who will create them. To achieve so, we developed a tool for 3D orientation model construction. Our tool interprets an input file in a predefined format which describes the model to be build and generates the correspondent 3D orientation model. In addition, we also created a converter for converting files that represent maps into the file format used by our tool. The converter input maps may be images (BMP files) with floor plans or map drawings, and also maps created in a customized map editor.

Keywords:: 3D orientation models, Converter, Ogre, Tiled Map Editor, OpenCV, X3D

Author's Contact:

{pedro7x,donanferals}@gmail.com
carlacastanho@cic.unb.br
jancorrea@unb.br

1 Introduction

Human beings have an inherent need to know their place in space and the path to be followed in order to get from one point to another. Formerly, such need was supported by various orientation mechanisms, such as compasses, sun's movements, and cartographic maps. According to [Draper et al. 1965], orientation is the process of gathering and applying information for generating commands of maneuver to control a vehicle movements. In this case, the human being is the vehicle itself which controls its movements.

According to [Peterson and Gartner 1999], the old cartographic maps - printed on paper - can only represent a static and immutable world. The following issues are generally faced by users when dealing with 2D static maps:

- the need to memorize the map and its path when a portable version is not available;
- the amount of time spent to visually locate a desired spot in a printed map;
- the difficulty in understanding the map, mainly because it represents an aerial view;
- the difficulty in locating themselves in the map.

With the evolution of technology, maps representation mechanisms have been significantly improved. Nowadays, most of the above mentioned limitations have been overcome with the use of digital maps instead of the printed ones. [Fosse and Veiga 2004] explains that digital maps provide a new way of interaction because users can easily load them in mobile devices such as laptops, cell phones, and tablets. This feature eliminated the need of memorizing the entire path in advance. Also, other limitations of static maps could be minimized through the application of computational techniques for digital maps manipulation [Delazari 2004]. Digital maps became interactive, allowing users to apply map scaling (zoom in/out), insert symbols for representation in the map, perform place automated search, mark points of interest for further search, among other possibilities.

In addition, interactive maps that have predefined paths allow the automatic computation of distance and trajectory between two different locations. This support minimized the hassle users usually have trying to evaluate their own path in a map. Also, location

systems such as GPS (Global Positioning System), were combined with digital maps enabling to detect and update, in real time, the user's position on the map.

In spite of that, the use of 2D models limits the user's perspective to an aerial view of the place represented in the map. Even if the user knows his/her location in the map and which path must follow, this kind of representation may confuse him/her. It is likely he/she will have to first find a reference point before proceed.

To reduce this limitation, the representation of the information must be presented in another way. As for human beings, three-dimensional (3D) representations can be much easier to comprehend than two-dimensional ones. With the increase of computational resources, maps that were once represented only on paper, with the interpretation performed by the user as the only possible form of interaction, were transformed into 3D maps or models, with the possibility of user interaction through input devices such as keyboards, mouses, joysticks, and touchscreen displays.

A guidance system that uses such three-dimensional models is an abstract representation of a 3D world, and it can provide an interactive environment where users can navigate in a three-dimensional model. Here, we call this three-dimensional guidance system with search capabilities and path calculation as a 3D orientation model.

The most known 3D orientation model is the Google Earth [Google a] (also its derivatives Google Maps [Google b] and Google Street View [Google d]), that has many of the facilitators features mentioned above, such as search for places of interest and path calculation. This model as well as others used by GPS guidance systems typically installed in cars and smartphones, represent an outdoor environment. 3D orientation models that represent indoor environments are much less common. Chloe@University [Paternier et al. 2007] is one of these rare 3D orientation model of an indoor environment.

Building 3D orientation models is a time consuming task and requires a great level of experience and expertise. Many of today's software such as Autodesk 3ds Max [Autodesk a], Autodesk Maya [Autodesk b] and Blender [Blender], support the creation of 3D models, but do not provide ways for the implementation of navigation features present in a 3D orientation model, such as place search, path search, etc. Moreover, these programs often have complex interfaces that require prior understanding of various concepts.

From such perspective, in this work we developed a tool for the construction of 3D orientation models. Its main goal is to ease the process minimizing the effort to create such models. Although the tool was designed with focus on indoor environments, it can also be used to model outdoor environments.

More precisely, our tool interprets a file in a predefined format which contains the description of the map whose 3D orientation model will be generated. To facilitate the creation of this input file we also created a converter responsible for converting maps into the specific file format used by our tool. The converter is able to deal with input maps created in a customized map editor and also with images (BMP files) containing floor plans or map drawings.

The remaining of this paper is organized as follows. In Section 2 we briefly describe some related works. Section 3 presents the tool for 3D orientation model generation as well as the converter. Finally, in Section 4 we give some concluding remarks.

2 Related Work

After an extensive search in the literature it was not found a tool with the same purposes as the one we present here. In fact, there are well known tools for 3D models generation, as stated above,

however they do not provide ways for the implementation of 3D orientation models, which must include features related to navigation, path search, place search, among others.

Google SketchUp [Google c], for example, is a freeware and easy to use tool to generate 3D models, but does not generate 3D orientation models. The Chloe@University project on the other hand, is a 3D orientation model itself, but it only represents the specific environment of a building in the university where the tool was developed. Chloe@University is not a tool and thus it can not be used to construct the representation of other environments. Our tool aims the creation of 3D orientation models such as Chloe@University, in a way as simple as Google SketchUp does.

Other projects were also studied in order to identify relevant and specific features that could be incorporated in our tool. Among them, the following are highlighted:

- Distributed 3D Virtual Museum [Ciabatti et al. 1998]: this system creates a 3D representation of objects from a previous description in VRML (Virtual Reality Modeling Language) language. A text file to describe a 3D model is also employed in our tool.
- Google Earth [Google a]: is a 3D orientation model that represents the planet Earth. Among several possibilities, it allows place search and path search between two locations. These two features are considered essential in a 3D orientation model and also implemented in the tool presented in this work.
- MUVA (Museu de Artes Visuais El Pais) [de Artes El Pais Web]: this project implements a 3D model of a virtual museum environment. It allows the user to navigate and interact with the environment, thus enabling virtual visits in the exhibits of the museum. From this application we borrowed the idea of displaying a small two-dimensional map, together with the 3D environment interface in order to assist user location in the virtual museum.

3 Proposed Solution

The tool for 3D orientation model construction presented in this work is actually composed by two applications: one responsible for reading a file in a predefined format which represents a map and generating the correspondent 3D orientation model; and another for converting files that represent maps into the file format used by the first application. The purpose of the converter is to facilitate the creation of the desired map, as it eliminates the need of typing the input file in the predefined format. The converter accepts input files in the following formats: our tool predefined format, images (BMP files) with floor plans or map drawings, and also maps created in a customized map editor.

In this section, we first present the application responsible for the 3D orientation models generation followed by the description of the converter.

3.1 Application for the 3D orientation models construction

This application allows the user to create in a simplified manner, a 3D orientation model just by reading a file in a predefined format, called map.cfg, which describes all the elements present in the model. Figure 1 shows an example of a map.cfg file, which describes a model with only one floor that has three units of height, being this floor a plan of 30x30 units with five walls. The values [1, 26, 16] indicate the initial position of the avatar in the model. Figure 2 shows the corresponding model of this file generated by our tool.

After generating the corresponding 3D orientation model, the navigation begins in the free navigation mode, which can be switched by the user. The available modes are:

- Free navigation mode: in this mode, the user can move freely through the model, moving the avatar representing him in the

```
{GENERAL_OPTIONS}
SIZE RANGE_X, 30
SIZE RANGE_Y, 30
NUMBER_OF_FLOORS, 1, 3
INITIAL_FLOOR, 1
INITIAL_X, 26
INITIAL_Y, 16

{FLOOR, 1}
STRAIGHT_WALL, 5, 5, 5, 26, wall_texture
STRAIGHT_WALL, 5, 5, 26, 5, wall_texture
STRAIGHT_WALL, 5, 26, 26, 26, wall_texture
STRAIGHT_WALL, 26, 5, 26, 14, wall_texture
STRAIGHT_WALL, 26, 17, 26, 26, wall_texture
```

Figure 1: Example of a map.cfg



Figure 2: The 3D orientation model of the map.cfg in Figure 1

virtual environment (Figure 3).



Figure 3: Free navigation mode

- Free viewing mode: when using this mode, the user is allowed to move the camera without restrictions (through walls and floors) to quickly become familiar with the whole map (Figure 4).
- Search mode: in this mode the user can perform a search for places of interest in the model. If the search is successful it is possible to see an animation of the avatar through the shortest path from the point where the avatar is until the located place of interest (Figure 5).

In all ways we can take a print screen and also view a two-dimensional map of the model. This mini map represents part of the floor where the avatar is, and aid the user navigation.



Figure 4: Free viewing mode



Figure 5: Search mode

3.2 The converter

The need to create the converter came after we identify that the creation of the file (map.cfg), through a common text editor, required the user to have a prior knowledge of the file format. Although the application for the creation of 3D orientation model made the job much easier, it was crucial to facilitate the creation of the map.cfg as well.

Therefore, the converter was designed to deal with maps created in a map editor and also in drawing softwares such as KollerPaint [Koller], and Gimp [GIMP]. These map drawings (and floor plans) must be saved in the 24 bits RGB (Red, Green, Blue) bitmap (BMP) format.

As for the map editor option, an intuitive tool capable of generating an output file that could be easily converted into the map.cfg format was required. We elected the Tiled Map Editor [Tiled], an open source software that allows the construction and edition of two-dimensional maps using pictures (tiles), objects and images. The user can define all the elements and characteristics of the map using this tool, and save it to a .tmx file format, that is a CSV (Comma-separated values) text file. The overall purpose of this editor is to be easy to use, and also flexible enough to work with different game engines. Because of these features, we adopted this map editor in order to build the map in a more pleasant and natural way than typing the map file (map.cfg) through a common text editor. The Tiled Map Editor was customized with a specific tilesset and some basic configuration, to have intuitive images in order to make it more user friendly, as shown in Figure 6.

The initial interface screen of the converter is shown in Figure 7. Using the first option (Load and convert text file map) the user selects the name and format of the file to be read (the input file), as well as the name and format of the output file. Here, map.cfg and Tiled Map Editor .tmx formats can be used as either input or output files. As an alternative, we also decided to create the op-

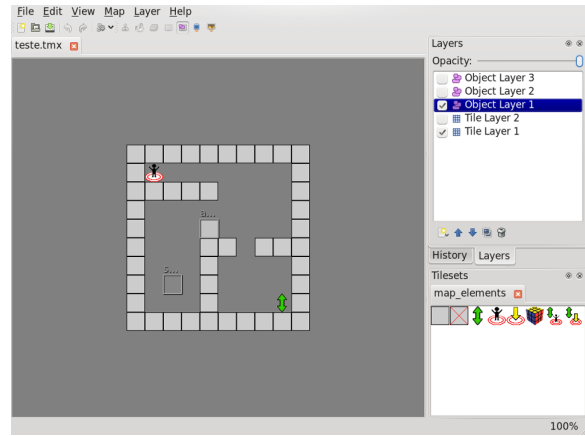


Figure 6: The customized Tiled Map Editor

tion to generate the map in X3D (Extensible 3D Graphics) format [Web3DConsortium], since it is an open standard and it can be used by any tool that can interpret it. So, this format is also available as an output format in the first option of the converter. In the second button (Load and convert image file map), the user can create a map file in the map.cfg format using images (BMP files) with floor plans or map drawings.



Figure 7: Initial interface screen of the converter

For conversion of maps based on images (.bmp), it is necessary that the user loads the images that compose the entire map (one per floor), as well as the size of map and the initial position of the avatar. As for the types that are not based on images (map.cfg and .tmx files) only the file that describes the map is loaded into the converter.

The user can, for example, create a drawing of a map (a file .bmp), use the converter to generate the map in the format used by the Tiled Map Editor (.tmx), make adjustments on the map using the editor, again use the converter to generate the file map.cfg from the edited map and finally use it to construct the 3D orientation model with our tool. Figure 8 shows the interaction of the converter with the application for the 3D orientation models construction.

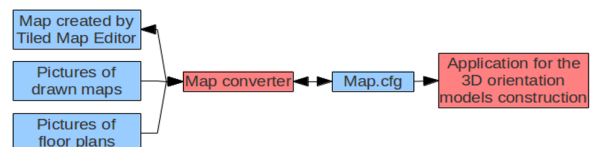


Figure 8: Interaction of the the converter with the application for the 3D orientation model construction

As the converter also generates, as an output, the map in the X3D format, it is possible to visualize the map using an X3D interpreter embedded in a web browser like Firefox that has extensions supporting the interpretation of files in X3D format. Figure 9 shows a 3D map generated in the X3D format.

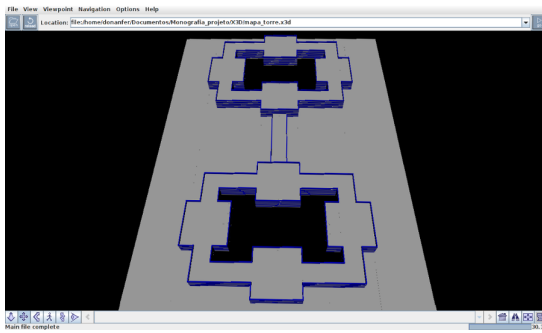


Figure 9: A 3D model of a file in the X3D format generated by the converter

The converter was implemented using the same idea of GCC (GNU Compiler Collection), which supports multiple input formats, uses them to generate a single intermediate format (independent of the input format), and finally uses this intermediate format to generate one of several output formats available. Based on this idea, the converter has a similar structure as it takes several input map formats: map.cfg, Tiled Map Editor .tmx, and the images with floors plan and map drawings (.bmp). From these formats, the converter creates a structure that contains the data of the map which is used to generate one of the output formats available: map.cfg, Tiled Map Editor .tmx and map in X3D (Extensible 3D Graphics) format. Figure 10 illustrates the architecture of the converter.

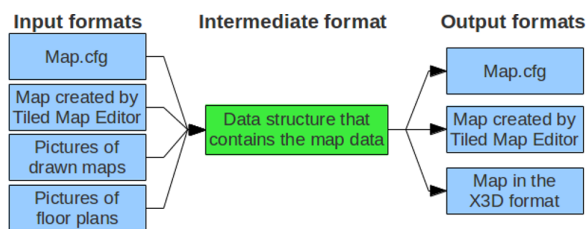


Figure 10: High level architecture of the converter

Reading the map.cfg and map.tmx files is done using a parser that uses regular expressions to search for specific keywords in the file. In the case of the maps on bitmap images (floor plans or map drawings), each image represents one floor, each floor image is read pixel by pixel, and the color of the pixel determines the contents of the coordinate that pixel represents. The colors are pre-set, for example, the black color indicates that the coordinate has a wall, the red color indicates that the coordinate has a vertical path, and so on. The difference between processing map drawings and floor plans is that before processing the images of the latter, the program uses the open source library OpenCV [Bradski 2000] to apply homogenization (blur) and edge detection filters in images in order to leave only the walls of the maps contained in the images. After that, the same processing of the map drawing is performed.

4 Conclusion

In this work we presented a tool to facilitate the hard, specialized and time consuming task of generating 3D orientation models. The tool does not require any programming or modeling knowledge. It interprets an input file in a predefined format which describes the model to be built. In addition, we developed a converter application capable of converting maps in different formats into the input format required by our tool.

This work have also demonstrate that it is possible to build such a tool using only open technology, such as C++ language, Tiled Map Editor, OpenCV, Qt and the graphics engine OGRE. This fact shows us that the cost to create 3D orientation models can be significantly reduced.

Among several possibilities for future works we consider the following: (i) improve the visual aesthetics of the graphics generated

by our tool; (ii) allow the creation of curved walls; (iii) add a shortcut button on the converter interface to directly run the application that builds the 3D orientation model from the file generated in the converter, and (iv) perform evaluation tests with users who have no knowledge of modeling and programming to evaluate if the tool and the converter accomplish its purposes.

References

- AUTODESK. *Autodesk 3ds Max*. Internet: <http://usa.autodesk.com/3ds-max>. Accessed in july 2012.
- AUTODESK. *Autodesk Maya*. Internet: <http://usa.autodesk.com/maya>. Accessed in july 2012.
- BLENDER. *Blender*. Internet: <http://www.blender.org>. Accessed in july 2012.
- BRADSKI, G. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- CIABATTI, E., CIGNONI, P., MONTANI, C., AND SCOPIGNO, R. 1998. *Towards a distributed 3D virtual museum*. In *AVI, ACM Press*, T. Catarci, M. F. Costabile, G. Santucci, and L. Tarantino, Eds., 264–266.
- DE ARTES EL PAIS WEB, M. V. *Museo Virtual de Artes El Pais*. Internet: <http://muva.elpais.com.uy/flash/muva.htm?&lang=en>. Accessed in july 2012.
- DELAZARI, L. S. 2004. *Modelagem e implementação de um Atlas Eletrônico Interativo utilizando métodos de visualização cartográfica*. PhD thesis, Departamento de Engenharia de Transportes - Escola Politécnica da Universidade de São Paulo, São Paulo.
- DRAPER, C. S., WRIGLEY, W., HOAG, D. G., BATTIN, R. H., MILLER, J. E., KOSO, D. A., HOPKINS, A. L., AND VELDE, W. E. V. 1965. *Apollo Guidance and Navigation*, vol. I – Space Navigation Guidance and Control. Instrumentation Laboratory of the Massachusetts Institute of Technology.
- FOSSE, J. M., AND VEIGA, L. A. K. 2004. *Modelo Tridimensional Interativo do Campus Centro Politécnico da UFPR em VRML*. Universidade Federal do Paraná, Curso de Pós-Graduação em Ciências Geodésicas, Paraná.
- GIMP. *Gimp*. Internet: <http://www.gimp.org>. Accessed in july 2012.
- GOOGLE. *Google Earth*. Internet: <http://earth.google.com>. Accessed in july 2012.
- GOOGLE. *Google Maps*. Internet: <http://maps.google.com.br>. Accessed in july 2012.
- GOOGLE. *Google SketchUp*. Internet: <http://sketchup.google.com>. Accessed in july 2012.
- GOOGLE. *Google Street View*. Internet: <http://maps.google.com.br/intl/pt-BR/help/maps/streetview>. Accessed in july 2012.
- KOLLER, M. *KolourPaint*. Internet: <http://kolourpaint.sourceforge.net>. Accessed in july 2012.
- PETERNIER, A., RIGHETTI, X., HOPMANN, M., THALMANN, D., PAPAGIANNAKIS, G., DAVY, P., LIM, M., MAGNENAT-THALMANN, N., BARSOCCI, P., KIRYKOU, A., AND REPETTO, M. 2007. *Chloe@University: An indoor, HMD-based mobile mixed reality guide*.
- PETERSON, M. P., AND GARTNER, G. 1999. *Multimedia Cartography*, first ed. Springer-Verlag, Berlin, 31–40.
- TILED. *Tiled Map Editor*. Internet: <http://www.mapeditor.org>. Accessed in july 2012.
- WEB3DCONSORTIUM. *X3D*. Internet: <http://http://www.web3d.org/realtime-3d/specification/version/V3.3>. Accessed in july 2012.