Annals of DAAAM for 2009 & Proceedings of the 20th International DAAAM Symposium, Volume 20, No. 1, ISSN 1726-9679
ISBN 978-3-901509-70-4, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2009
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
DAAAM World Symposium

THE VISUALIZATION CONTROL APPLICATIONS BASED ON OPENGL

POKORNY, P[avel] & MAHDALICEK, P[etr]

Abstract: *The modern control systems are composed from the more layer structure. For example, in the 2-layer structure can be the first layer based on programmable machine in the industry environment connected to the measured and controlled process. And the second layer can be based on the personal computer with the visualization and control software. This paper describes the two simulation and visualization control applications, which are running on the personal computer. These applications were developed on our faculty.*
Key words: *visualization, simulation, control, programming, OpenGL*

1. INTRODUCTION

Visualization is a technique for creating images, animations and diagrams to communicate a message. Visualization today has large sphere of usage - education, interactive multimedia, science, engineering, etc. The typical using of a visualization application is the field of the computer graphics.

The computer graphics is deeply wedded with the computer performance. With the computer performance growth, the graphics algorithms can be more complex and difficult. The results are better visual and more user friendly applications.

The one sphere of this, are visualization and control applications. These programs help us better manage and control of the real processes. There are existing some commercial applications, like Intouch or ControlWeb. These applications have some advantages (for example variability and increase engineering productivity), but also some disadvantages (for example they are not suitable for small control systems and they are very costly).

The other way is to create own application. This application has a benefit for the simple control processes, where we don't need the very large and complicated software.

2. THE BASE OF APPLICATION CREATION

Our plan was to create a universal 3D visualization engine, and to use this engine in some control programs to demonstrate its application field.

As the programming language was selected C++ language. This language has a universal usage and combines high-level and low level programming language features. The next step was to choose a good graphics library. In the C++ language are available many graphics libraries with the different possibilities. In the light of efficiency, quantity of functions, the OpenGL is available. (Schreiber et al., 2007)

OpenGL (Open Graphics Library) is a standard specification defining a cross-language cross-platform API for writing applications that produce 2D and 3D computer graphics. This interface consists of over 250 different functions. These functions can be used to draw complex three-dimensional scenes from simple primitives. OpenGL was developed by Silicon Graphics Inc. (SGI) in 1992 and is widely used in CAD, virtual reality, scientific visualization, information visualization, and flight simulation (***, 1997) (Wright et al., 2007).

OpenGL's basic operation is to accept primitives such as points, lines and polygons, and convert them into pixels. This is done by a graphics pipeline known as the OpenGL state machine. Most OpenGL commands either issue primitives to the graphics pipeline, or configure how the pipeline processes these primitives. Prior to the introduction of OpenGL 2.0, each stage of the pipeline performed a fixed function and was configurable only within tight limits. Where OpenGL cannot help us (for example drawing text), we can use standard API functions (Petzold, 1998).

3. THE CONCEPTION OF VISUALIZATION ENGINE

The basis of visualization applications is the 3D graphics engine. This engine is designed and created to easy use. It is not difficult to extend it with new functions (e.g. the physics algorithms or new render techniques). The engine is composed from many parts. All code is created by the object oriented conception – the all program code is encasement into the classes (Mahdalicek, 2007).

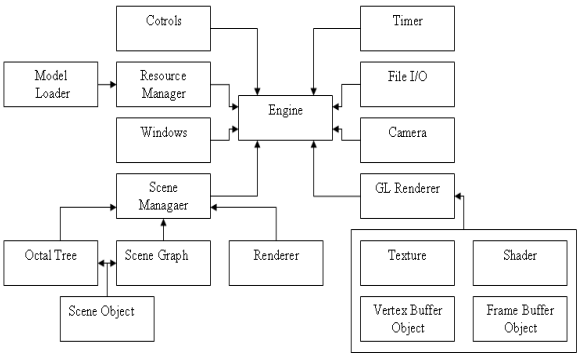

Fig. 1. The visualization engine: the block scheme of the classes

The connections between the classes are shown on the Figure 1. These classes are represented by blocks with the names. Below is the list of the most important classes with the short description:

- Windows – it helps to create render window with any resolution.
- Camera – operations with the mouse, keyboard and control the look direction.
- Controls – it represents the control algorithms.
- File I/O – it can read from the text files and write into them.
- GL Renderer – this is the renderer parts which uses OpenGL.
- Model Loader – it reads models from the ASE file.
- Resource Manager – this class manages the all imported objects (models, pictures, etc.).
- Timer – the class, which is used for the accurate timing.
- Scene Manager – it administrates objects in the scene.

4. THE TEMPERATURE CONTROL

To demonstrate the possibilities of the above described engine, we created two simple full 3D simulation visualization applications. The first application is the simulation of the temperature control. This model is composed from some parts – temperature sensor, heating valve, cooling valve, graphics display and graph. This system is shown on the figure 2.

There is implemented the two-step controller with the penalization (there are two positions only – heating or cooling), but it is not difficult to implement more complex controllers.

We can communicate with this application via mouse or keyboard. We can set some important control parameters – sampling period, required value, critical value and penalization of the controller. Any time we can restart the control loop and export data from the controller into the text file.

Fig. 2. The temperature control visualization

The graph visualizes and records the temperature history (the requested value, and the real controlled value in dependence on the time). The example of this archiving is shown on the figure 3.

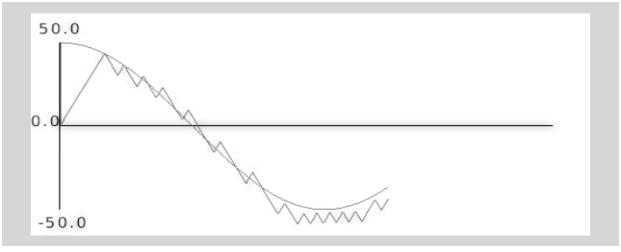

Fig. 3. The graphics visualization of the temperature control

5. THE HIGH WATER LEVEL CONTROL

The second visualization application demonstrates the control of the high water level in the container. This scene is shown on the figure 2. The base of application is designed to set three different views on this scene. First view is named "Panel" and when we select it, the camera in the scene is zoomed to the control panel. There are displayed the most important control values – inflow, height of water and total flow of water in this system.

The second view "Center" represents the main view to the control process (shown on the figure 4). And the third view "Detail" sets the camera position above the controlled water container and its orientation is set to show more detailed inner of the container to better see, which volume of water is here.

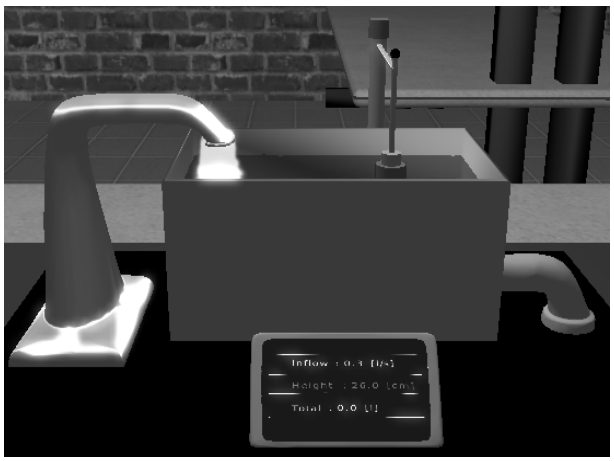

Fig. 4. The high water level control visualization

The control system is very simple. The water flows to the container and when the container is full, the inflow is stopped. After it, it is running the outflow so long as the container is empty. When the container is empty, it is running the inflow and all process is running again. In this simulation we can set the sampling period and water height in the container. This height specifies when the inflow stops and outflow runs.

6. CONCLUSION

In this paper there is the short describe of the two visualization control applications, which were developed on our faculty. These applications are based on the visualization engine. This engine uses cross-language and cross-platform API OpenGL. These independencies are benefits, which give it the large sphere of usage in the many applications.

The first application simulates the temperature control. Here we have temperature sensor and main goal is to keep the required value (the temperature). To gain it, we can use heating or cooling valve. The second application simulates the high water level control in the container, where the water flow in and flow out.

The created applications demonstrate the power of this engine and show the additional usage. As next, we are planning to use this visualization engine to control some real models in our laboratory.

7. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education of the Czech Republic in the range of research projects No. MSM 7088352102.

8. REFERENCES

Mahdalicek, P. (2007). *Using OpenGL for the simulation and visualization of the control processes*, Bachelor thesis, Thomas Bata University in Zlin, Zlin

Petzold, Ch. (1998). *Programming Windows*, Microsoft Press, ISBN 157231995X, San Francisco

Schreiber, D.; Woo, M., Neider, J. & Davis, T. (2007). *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, Addison-Wesley Professional, ISBN 0321481003, Chicago

Wright, R. S.; Lipchak, B. & Haemel, N. (2007). *OpenGL Superbible: Comprehensive Tutorial and Reference*, Addison-Wesley Professional, ISBN 0321498828, Chicago

*** (1997) http://www.opengl.org – OpenGL–The Industry Standard for High Performance Graphics, *Accesed on:2009-06-14*