13th International Research/Expert Conference
"Trends in the Development of Machinery and Associated Technology"
TMT 2009, Hammamet, Tunisia, 16-21 October 2009

THE SIMULATION AND VISUALIZATION CONTROL APPLICATION

Pavel Pokorny
Thomas Bata University in Zlin, Faculty of Applied Informatics
nam. T.G Masaryka 5555, 760 01 Zlin
Czech Republic

Petr Mahdalicek
Thomas Bata University in Zlin, Faculty of Applied Informatics
nam. T.G Masaryka 5555, 760 01 Zlin
Czech Republic

ABSTRACT
The control system of the technological process can be the one or more layer structure. The one level structure is composed from the personal or industry control computer. This computer has implemented a technological card and trough this card this computer communicates with the controlled process. The modern control systems are composed from the more layer structure. For example, in the 2-layer structure can be the first layer programmable machine in the industry environment connected to the measured and controlled process. And the second layer can be personal computer with visualization and control software. Both phases are connected via the serial or wireless connection. This paper describes the simulation and visualization control application, which was developed on our faculty.
Keywords: simulation, visualization, application

1. INTRODUCTION
The modern control systems are composed from the more layer structure. The highest layer can be formed by personal computer with the visualization and control software.
We can use varied applications for the controlling and visualization. The commercial applications, like Intouch or ControlWeb are most used. 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 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
First, when we want to create application, we must select a programming language. Because we have many experiences with the C++ language, we selected them. The next step is 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. [3]
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. [1,4,5]

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. OpenGL 2.0 offers several stages that are fully programmable using GLSL. In situations, where DirectDraw cannot help us, we can use standard Win32 API function, like drawing text or simple 2D objects. [2]

3. THE CONCEPTION OF VISUALIZATION ENGINE

The basis of visualization application 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 encasements into the classes.

The connections between the classes are shown on the Figure 1. The classes are represented by blocks with the names. How is noted above, the basis is the engine. It is supported by the other parts of application. Below is the list of the most important classes with the short desription:

- Windows – it helps to create render window with any resolution.
- Camera – operations with the mouse, keyboard and control the look direction.
- Collidor – this class can detect the collisions of different objects.
- 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.).
- Texture – it is class, which can create and read raster image files.
- Timer – the class, which is used for the accurate timing.
- Shader – implements shaders and set uniform variables.
- Scene Manager – it administrates objects in the scene.

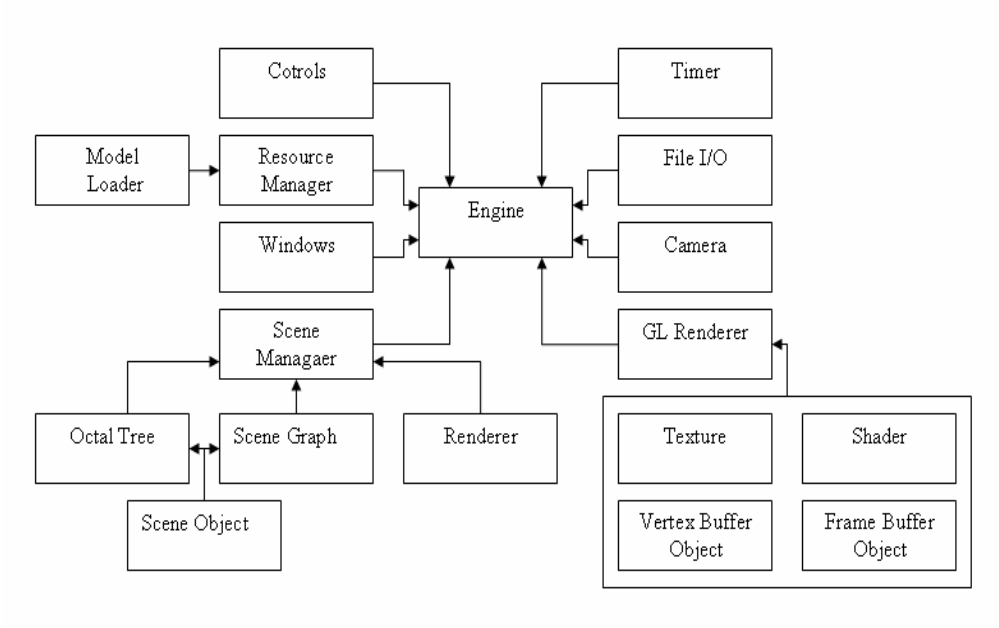

Figure 1. The Block scheme of classes the visualization application

4. THE VISUALIZATION APPLICATION

This described engine has the universal usage. To demonstrate its possibilities, we created the simple simulation visualization application. This application demonstrates the control of the high water level in the container.

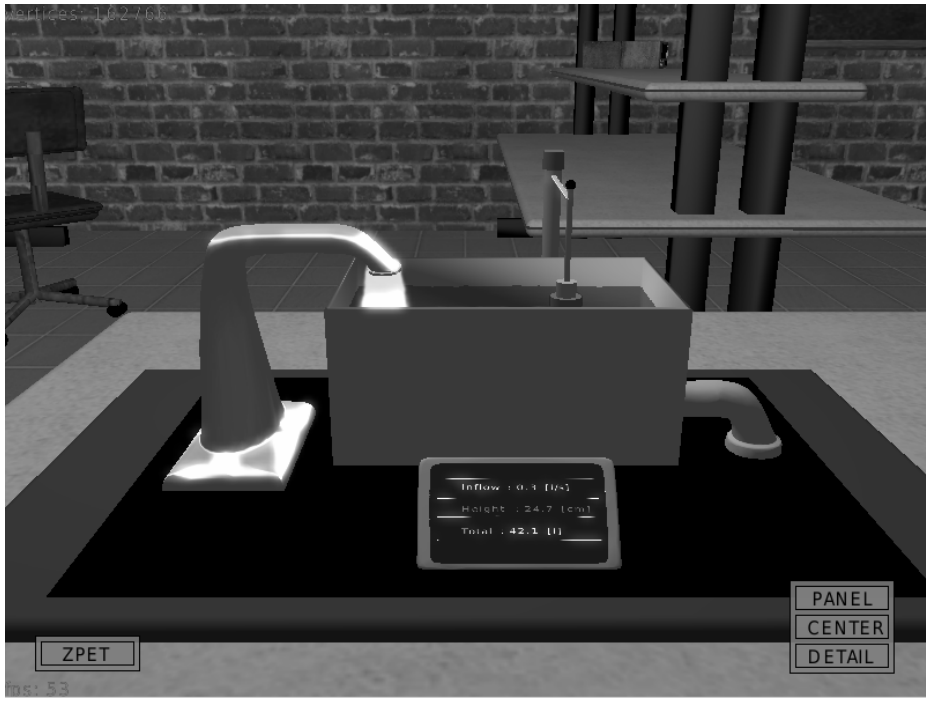

Figure 2. The screen of OpenGL Control application

The scene, where is visualized the control system of the high water level is shown on the figure 2. This application is designed to set three different views on 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 important control values – inflow, height of water and total flow of water in this system. This view is shown on the figure 3.

The second view “Center” represents the main view to the control process (Figure 2). And the third view “Detail” set the camera position above the controlled water container and its orientation is set to show more detailed inner of container to better see, which volume of water is here.

The control system is very simple. The water flows to the container and when is the container 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 inflow stops and run outflow.

It is written above – the control system is very simple. But, it is not hard to implement more difficult control algorithms. The main goal was to create visualization engine, which can be used in the simulations.

5. CONCLUSION

In this paper there is the short describe OpenGL API and its using for the creation of visualization application. OpenGL is a standard specification defining a cross-language cross-platform API for writing applications that produce 2D and 3D computer graphics. It is not difficult to implement OpenGL into Windows (or other platform) source code applications and it has low system

requirements to running on computer. These benefits give it the large sphere of usage in the many applications.

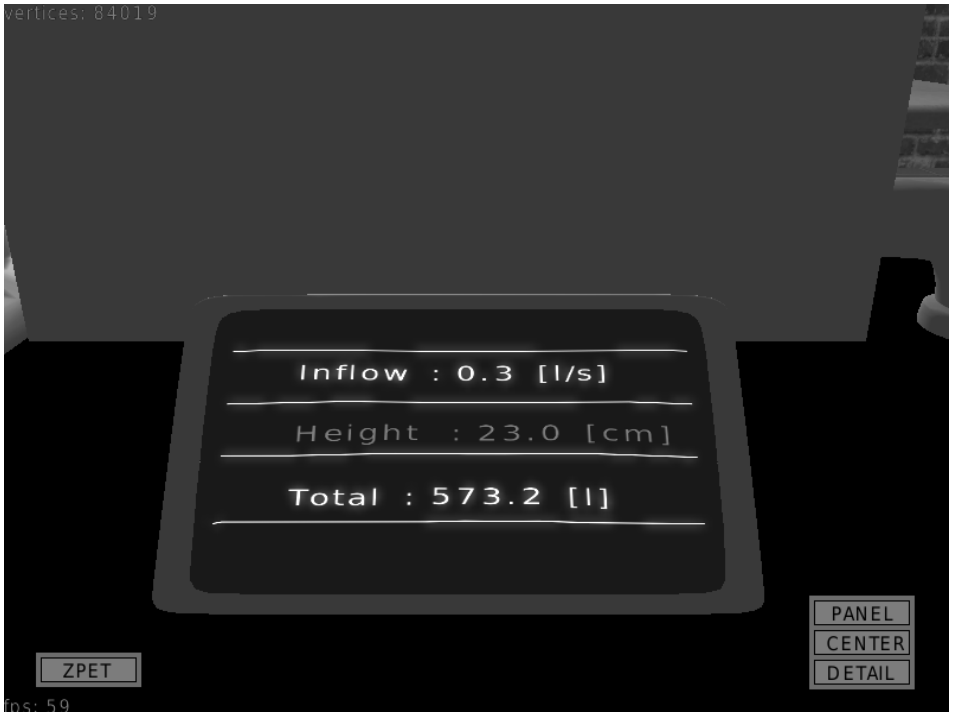

Figure 3. The control panel of OpenGL control application

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] OpenGL.org.: OpenGL - The Industry standard for High Pergormancce Graphics [online]. 1997-2009 [cit. 2009-06-10]. Available on: <http://www.opengl.org/>,

[2] Petzold Ch.: Programming Windows. 5th ed. Microsoft Press, 1998, 1478 p. ISBN 157231995X,

[3] Shreiber D., Woo M., et all: OpenGL Programming Guide: The Official Guide to Learning OpenGL. 6th ed., Addison-Wesley Professional, 2007, 928 p. ISBN 0321481003.

[4] Wikipedia Contributors: OpenGL [on line]. Wikipedia, The Free Encyclopedia, 2009. [cit. 2009-06-10]. Available on: < http://en.wikipedia.org/w/index.php?title=OpenGL>,

[5] Wirght R. S., Lipchak B., Haemel N.: OpenGL SuperBible: Comprehensive Tutorial and Reference. 4th ed., Addison-Wesley Professional, 2007, 1248 p. ISBN 0321498828.