Simulation of Human Body Thermoregulation

Jana Hájková[*]

`hajkovaj@kiv.zcu.cz`

Martin Šašek[**]

`msasek@kfy.zcu.cz`

Pavel Herout[*]

`herout@kiv.zcu.cz`

Abstract: This paper describes aims and development of a project, which is created at the University of West Bohemia. As the result of the project a new exhibit, the system for simulation of human body thermoregulation, to the rising technical and science museum Technorama is coming up. The system is based on the physical based simulation kernel, its activity is visualized, the visualization serves for comfortable user interaction.

Key Words: Technorama, visualization, simulation, thermoregulation, OpenGL, human body, JOGL API.

1   Introduction

There was founded a new project at the University of West Bohemia in cooperation of the Department of Computer Science and Engineering, the Department of Physics and the New Technologies-Research Centre. Its aim is to create an exhibit into the new arising technical and science museum in Pilsen. The project deals with the system for simulation of human body thermoregulation. It should show the visitors of the museum how the human body reacts on the changes of the temperature of ambient air depending on other factors.

The whole system includes the simulation kernel and the visualization of the simulation. The kernel is based on the interconnection of biology and physics knowledges. The visualization will be joined to the kernel and will serve for the communication between the user (visitor) and the simulator.

Nowadays the system is in the first stage of development. For each single part of the project there are searched the best ways to solve the subproblems. We are checking possible tools that would be used. In process there is the investigation for potential speed of computation and for portability on the new supercomputer from the Westbohemian Supercomputing Centre. There are researched also file formats for human body model description to create and describe models for visualization. After the investigation all members of development team will be able to use necessary technologies.

In this paper you can find basic facts about the starting project. In Section 2 you can find some information about new rising technical and science museum Technorama, Section 3 speaks about the basic principles of the human body thermoregulation and models that are

---

* Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 22, 306 14 Plzeň

** Department of Physics, Faculty of Applied Sciences, University of West Bohemia

used. In Section 4 the mechanisms for visualization in Java using OpenGL are described. Section 5 concludes the paper and gives aims to the future.

2   Technorama

In Pilsen there is rising the largest technical exposition in the Czech Republic, the technical and science museum Technorama. Its residency will be situated into the district of the former building of the Škoda Works. The area intended for the exhibition hall is about 100×60 meters and according to expectations the museum should be opened in 2008.

The new rising Technorama is planned to be a new modern interactive type of museum. It should not be full of old dusty exhibitions, but the visitors will find here many interdisciplinary exhibitions not only for viewing, but also for experimenting with. The exhibitions will be interactive and they will have an educational function.

Moreover the exhibitions are expected to be completely changed after each three years of their existence. It means that each year about one third of exhibits will be replaced. Each year there could be seen something new. All visitors will come into their own - pupils, university students, experts or amateurs.

3   Simulation of Human Body Thermoregulation

Human body is a very interesting system which is able to regulate its temperature also if the air conditions surrounding are changing (sun, wind, rain, changing air humidity, etc.). Other factors that influence the thermoregulation are the quantity and quality of clothes and the performed activity. Important aspect is also the difference between the thermoregulation of man's and woman's body.

3.1   Physical Model

The mean temperature of human body is 36.7°C. The inner body thermoregulation keeps the body temperature in interval ± 2.5°C. Small divergences are caused by individual conditions of each single person. By overrunning the limiting values the thermoregulation break happens (overheating, collapse of the organism, undercooling, freeze). The thermoregulation is regulated by thermal sensors which detect the temperature of the environment and after detection of the temperature change they send signals to the brain to invoke necessary physiological processes.

Processes of thermoregulation are very sophisticated. That is why we have to simplify them. Of course, the model has to stay authentic and reliable enough. Many models for human body have been described [1]. The simplification of the model depends on the required function of the model.

The group of models that can be used for simulation of human body thermoregulation makes use of integral characteristic monitoring. This approach enables quite a high level of simplification. These models are more described in detail in [1].

Human body or its parts are replaced by geometrical models. The level of detail depends on the needs of simulation. The whole body can be described as one object or it can be divided into several parts (a body, arms, legs, fingers, etc.). For each model anatomical and geometrical properties are studied. The model can be a basic solid (a board, a cylinder, a sphere) which can have several layers. It is typical that models have from one to six layers. Some examples can be seen in Figure 3.1.
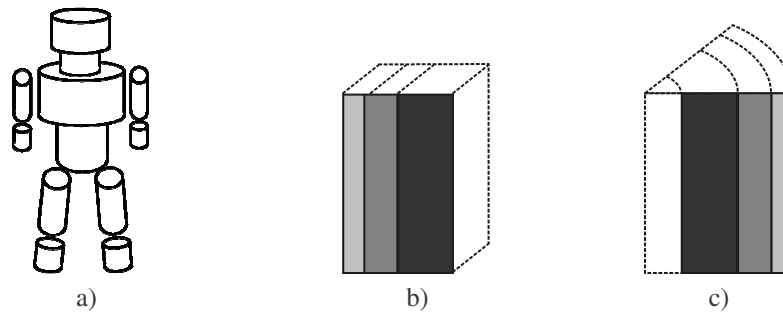
Figure 3.1 a) A substitutive geometrical model of human body; b) A board model of body or its parts (3 layers); c) A cylindrical model of body or its parts (more than 3 layers).

3.2 Visualization

Visualization part serves for communication between the user (a common visitor of Technorama) and the simulator. It should be a user friendly and well-arranged program that enables observation and controlling of behavior of human body and its energy produce during various temperature and weather conditions.

The visitor can observe, how the body responds to the changes. If it is cold, the body quakes for cold, it can come to chill, or chilblains, if the body is dressed to much, it get sweat and can chill. In the summer or in overheated rooms the body starts to perspire, it can come to overheating, even more to a total collapse of organism. Responses depend heavily on the air humidity and hardiness of the organism.

All these reactions of the body will be presented to the user by visualization of two characters (a man and a woman). The user will be able to dress them to make them practice some activities, to change the weather, temperature and many other factors influencing the thermoregulation. The visualization should be as realistic and attractive as possible, it should enable imagination of the real reactions of the body. For comfortable control of the simulation, we are projecting a well-arranged GUI with the potential level of detail in settings of the control (for someone, who wants to observe or for a specialists who needs to get exact information about the reaction of the body to the special condition).

4 OpenGL and Java

One of the most often used programming languages today is Java. The reasons why Java is so popular are especially Java's portability and operating system independency. In the world of computer graphics OpenGL is the most used tool for 3D visualization.

To interconnect Java and OpenGL we have to use some resource to join all OpenGL functions on the Java notation. There is a plenty of packages to ensure this. Some of them are object oriented (each object - vertex, triangle, light, texture, etc. - has its own instance; instances create necessary hierarchy), some are prepared in the same manner as the OpenGL is. We decided to use the JOGL (Java for OpenGL) API [2], which makes use of typical OpenGL approach.

Before the very beginning of using OpenGL we have to install all necessary libraries. There are two types of libraries - a platform-independent JAR file containing the Java classes of the library (`jogl.jar`) and a platform-dependent native library containing the associated JNI code which calls OpenGL (e.g. `jogl-natives-win32.jar`). Before programming of

the first OpenGL application in Java we have to save them and to set CLATHPATH correctly. Now we can start OpenGL in Java.

The basic example in OpenGL is to draw a triangle. Let's do it with Java. We need two classes - the first one to create drawing canvas, the other class for creating the main window.

The first class called `Jogl` serves as definition for drawing canvas. It has to implement `GLEventListener` and thus it includes four methods at minimum. The most important one is the display method that ensures drawing of defined objects. The source code of the `Jogl` class is listed below.

```
import javax.media.opengl.*;

public class Jogl implements GLEventListener
{
  private GL gl;
  public void init(GLAutoDrawable drawable)
  {
    this.gl = drawable.getGL();
  }
  public void display(GLAutoDrawable drawable)
  {
    gl.glClear(GL.GL_COLOR_BUFFER_BIT | GL.GL_DEPTH_BUFFER_BIT );
    gl.glBegin(GL.GL_TRIANGLES);
      gl.glColor3d(1.0, 0.0, 0.0);
      gl.glVertex3f(-0.8f, -0.8f, 0.0f);
      gl.glVertex3f(0.8f, -0.8f, 0.0f);
      gl.glVertex3f(0.0f, 0.8f, 0.0f);
      gl.glEnd();
  }
  public void reshape(GLAutoDrawable drawable, int x, int y, int width, int height) {}
  public void displayChanged(GLAutoDrawable drawable, boolean modeChanged, boolean deviceChanged) {}
}
```

The second class called `Jogl_main` is used to create the main window. Let's look at the source code. In the constructor there are set parameters of the window, size, title, etc. It is also possible to use any other control features of GUI (buttons, textfields, etc.) to influence drawing at the drawing canvas. The instance of the canvas is also created in the constructor. The animator serves as a drawing loop for continuously redrawing of defined objects. The main window of the basic example you can see in Figure 4.1.

```
import javax.swing.JFrame;
import javax.media.opengl.GLCanvas;
import com.sun.opengl.utils.Animator;

public class Jogl_main extends JFrame
{
  Jogl jg = new Jogl();
  Animator animator = new Animator();
  public Jogl1() {
    GLCanvas canvas = new GLCanvas();
    canvas.addGLEventListener(jg);
    animator.add(canvas);
    this.getContentPane().add(canvas);

    this.setTitle("Triangle");
    this.setSize(330, 380);

    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setVisible(true);
    animator.start();
  }

  public static void main( String[] args ) {
    new Jogl_main();
  }
}
```
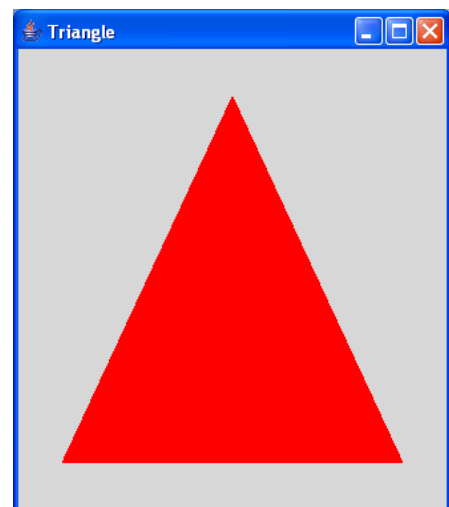

Figure 4.1 Main window - instance of the `Jogl_main` class.

It is surely possible to use remainder functions of OpenGL for materials, lights, texturing, cameras, settings of perspective, etc. These functions are described for example in [3]. We can also draw much more complicated objects and scenes.

For visualization of human body thermoregulation it will be necessary to prepare a model of human body. This can be done e.g. in 3D Max. The model can be exported into some of special formats and visualized. Some examples of our research you can see in Figures 4.2 a-b.

5   Conclusions

The project is in very early stage of development, separate parts are in the phase of projection and start to evolve. Our aim is to create the first version of the simulation system and later on to enlarge it. It is very probable that during the development we will find many other interesting usages or extending of the project. One of plenty occasions could be e.g. expansion into organisms others than humans (e.g. seals, arctic foxes), where a number of interesting and surprising mechanisms can be found.
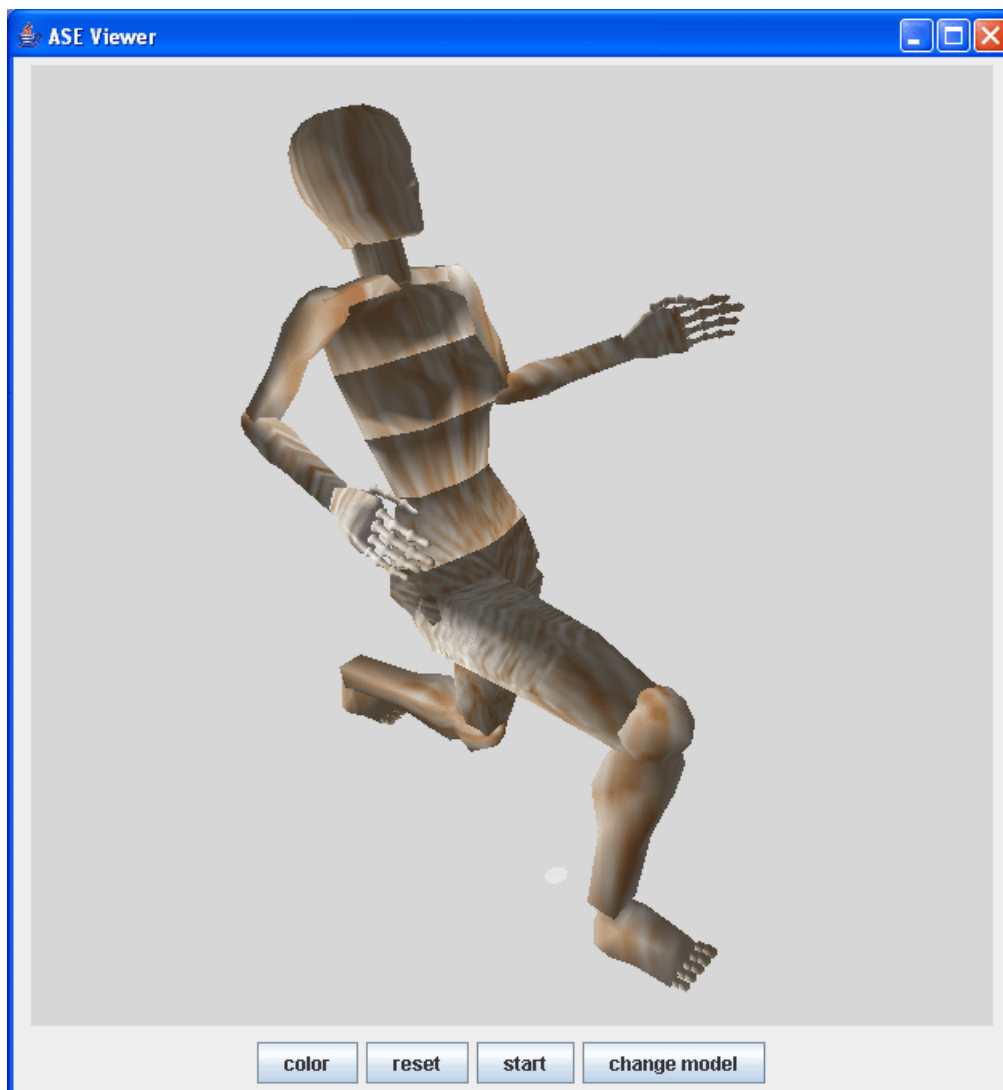
a)
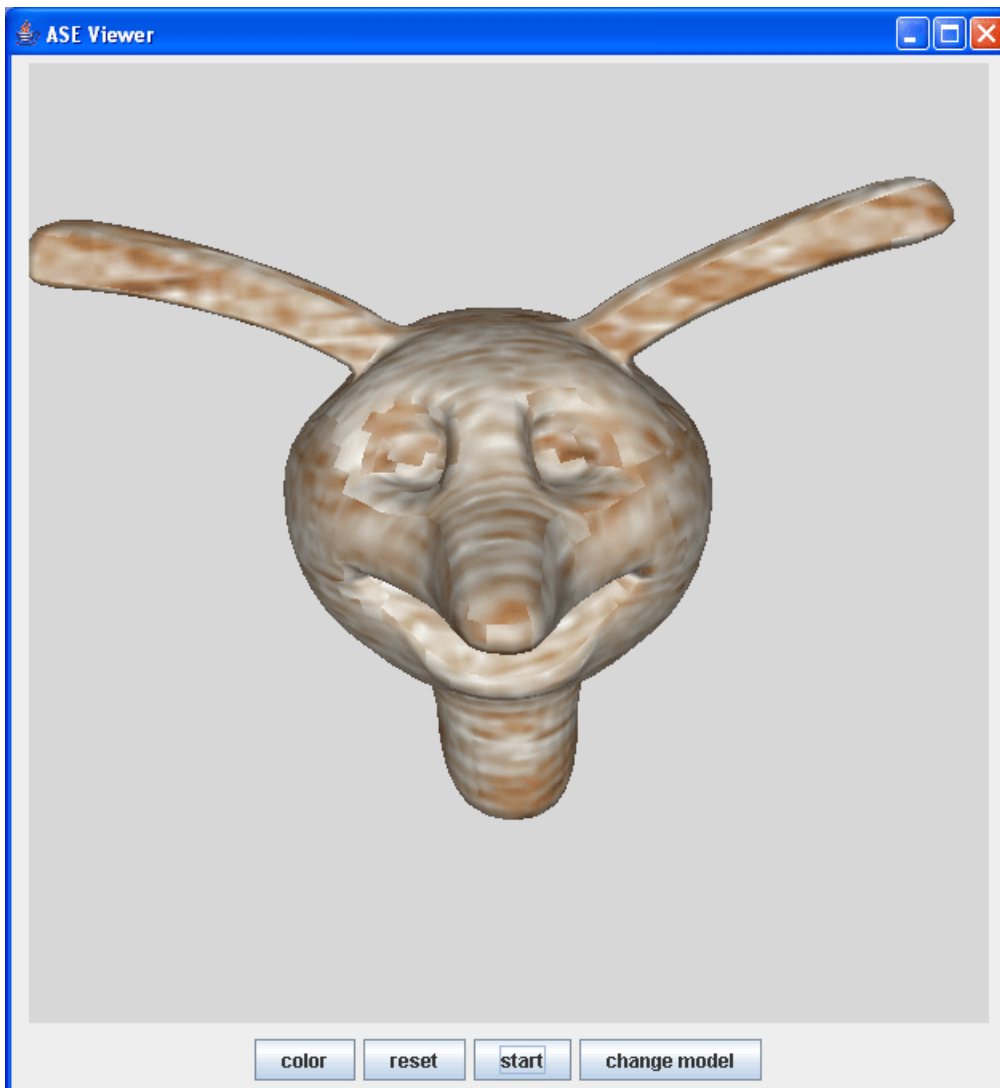
b)

Figure 4.2   Models exported from 3D Max and visualized in Java using OpenGL.

References

1. Josef Kuneš, Zdeněk Veselý, Milan Honner: *Tepelné bariéry*. Academia, Praha 2003, ISBN 80-200-1218-4

2. https://jogl.dev.java.net/

3. Dava Shreiner, Mason Woo, Jackie Neider, Tom Davis: *OpenGL Programming Guide*. 2004, Addison-Wesley, ISBN 0-321-17348-1