CA417 Assignment 2

Overview

Three distinct pieces of interactivity were added to the scene in this part – Mouse controls for moving objects, mouse controls for selecting objects and keyboard controls.

The interactivity of the Scene

Keyboard controls – TV

This is the simplest example of interactivity – the program waits on keyboard presses and changes the texture on the TV screen based on input.

The keys are mapped to resemble a remote control – keys 1,2 and 3 offer 3 different "channels" on the TV, 9 gives us a "no signal" blue screen and 0 switches the television off.

Mouse controls (Picking objects) – The Lamp

This piece of interactivity is used primarily for the lamp – by clicking on the lamp the user can toggle between the "on" and "off" states. This class uses the functionality that is provided by the PickMouseBehavior class, where each interactive shape is given a name through a call to setUserData() and can then be queried about its identity when a mouse click occurs.

This class is also used to recognise the chair, which has its own special piece of functionality.

Mouse controls (Moving objects) – The Chair

This is the most advanced portion of the program - it records mouse movement and transforms the object accordingly. It operates on the chair and relies on drag-and-drop movement.

When the user selects the chair and drags the mouse the camera movement that is normally associated with that action is disabled for the duration of the mouse press.

The chair travels along with the mouse within a bounded area and can be dropped at any time. Behind the scenes, the position of the viewer, the position of the mouse and a plane are used to map the mouse coordinates onto one of the 3-D surfaces (the plane). The program is using the floor as its base plane and the chair can only travel along the X and Z axes.

Issues

Because the program maps the points to the floor, sometimes the floor gets selected instead of the object that was meant to be picked. This is most evident in the case of the chair, which sometimes proves to be difficult to "select" (The beast area for selection seems to be the back of the chair when the camera is looking downwards).

I’ve looked for a solution to this problem and attempted to use the PickRay class in combination with the plane intersection equation, but despite having the point of origin and an idea of what the user’s POV should be, I couldn’t map the direction of the PickRay to properly face the user.

If that were successful, I could select objects that are closest to the viewer along the PickRay vector – when we select objects we tend to click “behind them”, effectively clicking on the floor/wall behind the object. A line between the user’s POV and the place where the click was registered on the floor would therefore go through the object that the user wanted to select.

Fixes introduced since Assignment 1

In Assignment 1, the chair’s textured appearance did not interact with the lighting properly. This has now been fixed.

References

Springer and Frank Klawonn, useful Java 3D demos - http://public.rz.fh-wolfenbuettel.de/~klawonn/computergraphics/e-buch/java3d.html

Java3d.org, Of Mice and Men, point/plane intersections. The equation to calculate the point of intersection was taken from this example - http://www.java3d.org/miceandmen.html

Java2s.com, Keyboard controls - http://www.java2s.com/Code/Java/3D/Rotatewhenanykeyispressed.htm

Game of Thrones – a promotional image was used for one of the “channels” on the TV

The Shawshank Redemption – an image from the movie was used for one of the “channels” on the TV

PlayStation 3 home screen– a screenshot from the game console was used for one of the “channels” on the TV

NASA’s photograph of Mars - http://o.canada.com/2012/07/09/this-photo-will-make-you-feel-like-youre-on-mars/