

## C# with GoblinXNA library

I built a solar system including a sun, four planets (Earth, Mars, Mercury and Jupiter), a satellite (Moon) and four space debris fields (cylinder, cone, torus and box).

Here is a list of all files :

SolarSystem.sln

Program.cs

Game1.cs

Newton.dll

GoblinXNA.dll

Sun1.fbx

Sun2.fbx

Planet1.fbx

Planet2.fbx

Planet3.fbx

Planet4.fbx

Moon1.fbx

Moon2.fbx

My computer's OS type is Windows 7 32bit.

Instructions for how to use my Program:

When user first launches the interface, the camera position is at (0, 0, 4) where users can see the entire solar system, as show in figure 1.

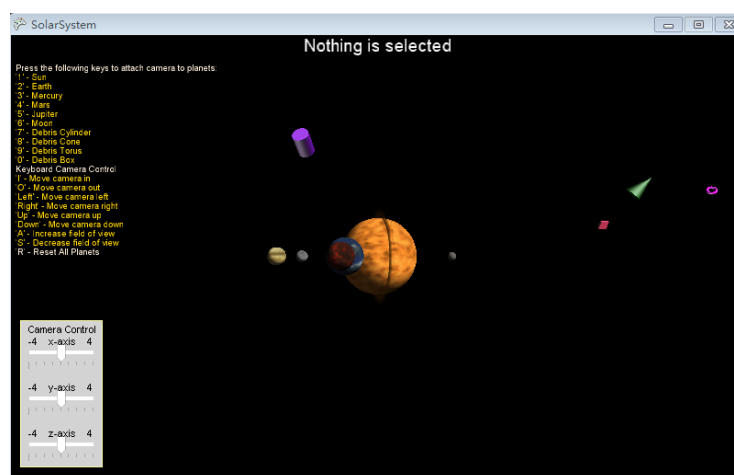


Figure 1: the solar system.

The camera control panel is at the bottom left corner where it does not bother users in viewing the solar system, as show in figure1. The background of this panel is light gray, which is distinguishable from the solar system with black background. This control panel is transparent. Users are able to see any objects fly behind it. This camera control panel has three sliders to control the

movements of the camera along the three axes. The range of these three sliders is from -4 to 4, and they all initially start from 0 where is at the middle of the panel, as shown in figure 2.

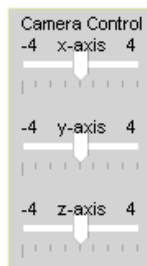


Figure 2: the camera control panel.

Users are able to move the camera right and left by adjusting the first slider.  
 Users are able to move the camera up and down by adjusting the second slider.  
 Users are able to move the camera in and out by adjusting the third slider.

A second transparent panel displays instructions of attaching camera to each planets and keyboard control is placed at the top left corner, as shown in figure 2. This panel does not have a background color. As a result of its transparency, users are able to see objects fly behind it. I set the font color to gold, which is easier to catch users' attention. As shown in figure 3, the instructions in the panel are clearly categorized to two sections including keyboard camera control and attachment control.

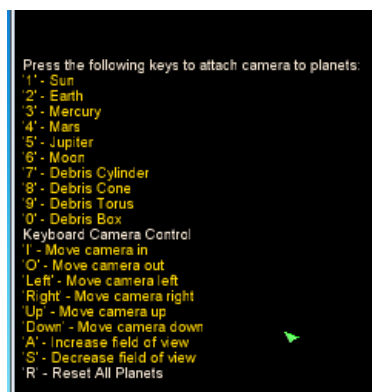


Figure 3: Panel for camera control and attachment.

#### Keyboard Camera Control:

Pressing keys "I" and "O" will move camera in and out (along z-axis).  
 Pressing keys "Left" and "Right" will move camera left and right (along x-axis).  
 Pressing keys "Up" and "Down" will move camera up and down (along y-axis).  
 Pressing keys "A" and "S" will increase and decrease camera's field of view.  
 Pressing key "R" will reset all planets and values.

#### Camera Attachments:

Pressing key "1" will attach the camera to the Sun, as shown in figure 4.  
 Pressing key "2" will attach the camera to the Earth, as shown in figure 5.

Pressing key “3” will attach the camera to the Mercury, as shown in figure 6.  
Pressing key “4” will attach the camera to the Mars, as shown in figure 7.  
Pressing key “5” will attach the camera to the Jupiter, as shown in figure 8.  
Pressing key “6” will attach the camera to the Moon, as shown in figure 9.  
Pressing key “7” will attach the camera to the cylinder debris, as shown in figure 10.  
Pressing key “8” will attach the camera to the cone debris, as shown in figure 11.  
Pressing key “9” will attach the camera to the torus debris, as shown in figure 12.  
Pressing key “10” will attach the camera to the box debris, as shown in figure 13.

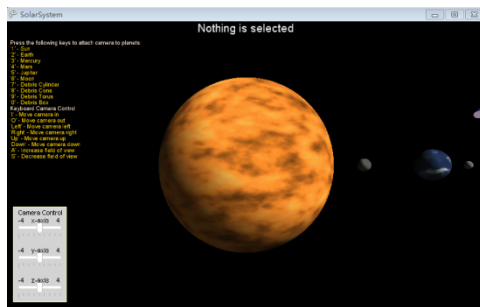


Figure 4: Attach the camera to the Sun.

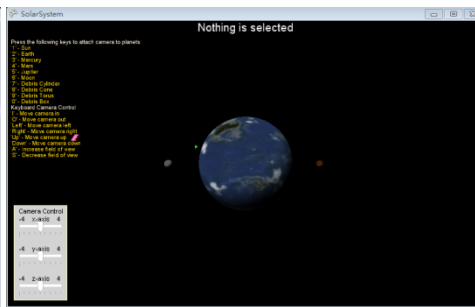


Figure 5: Attach the camera to the Earth.

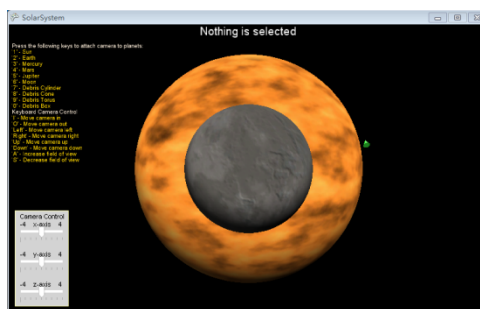


Figure 6: Attach the camera to the Mercury.

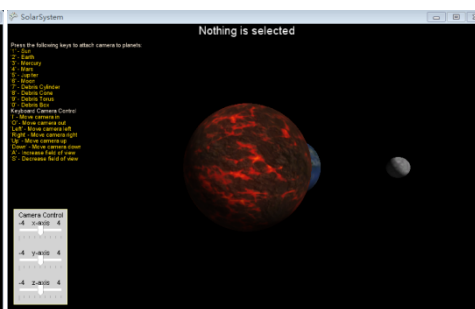


Figure 7: Attach the camera to the Mars.

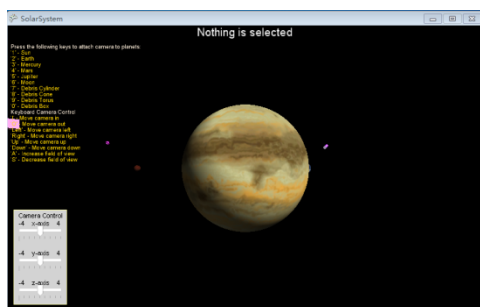


Figure 8: Attach the camera to the Jupiter.

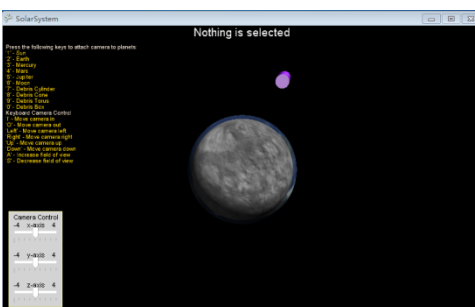


Figure 9: Attach the camera to the Moon.

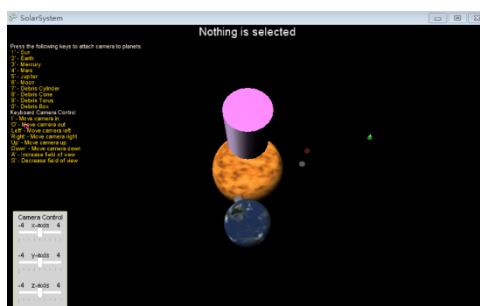


Figure 10: Attach the camera to the cylinder. Figure 11: Attach the camera to the cone.

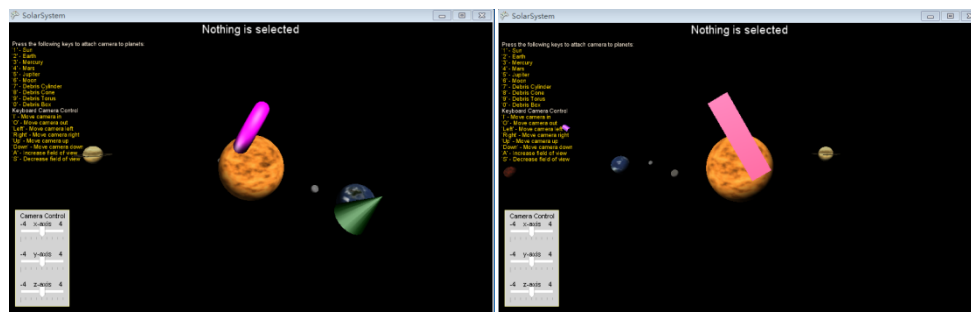


Figure 12: Attach the camera to the torus. Figure 13: Attach the camera to the box.

Users are able to pick any objects in the system. When the Sun is selected, all planets and satellites will stop moving and pausing at their current poses. A red bounding box will be displayed around the Sun to indicate the selection, as shown in figure 14. Once the Sun is deselected, all planets and satellites will resume rotating around their axes and orbiting. When a planet or satellite is selected, it will stop rotating and pause at its current pose with a red bounding box. A new object control panel will be displayed at the bottom right corner, as shown in figure 15. This control panel is transparent as well and allows users to see moving objects behind it.

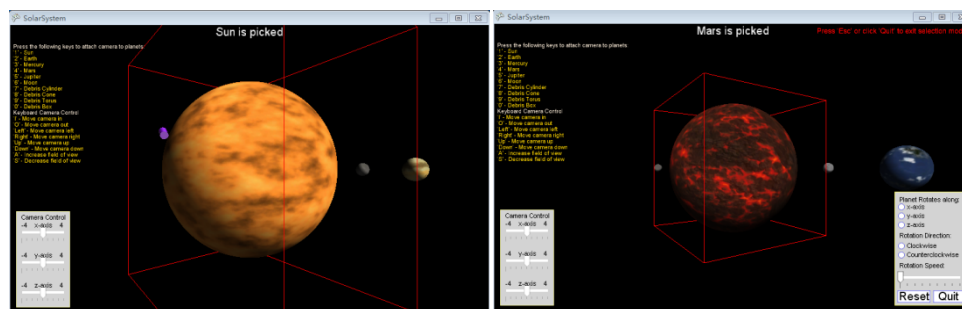


Figure 14: Selecting the Sun.

Figure 15: Selecting the Mars.

The third control panel is to control the selected planet or satellite. It has two sets of radio buttons and a slider. The first set of radio button is to select which axis to rotate (x-axis, y-axis or z-axis). The second set of radio button is to select which direction to rotate (forward or backward). The range of the slider is from 0 to 8, and it initially starts from 0 where is at the left of the panel, as shown in figure 16.

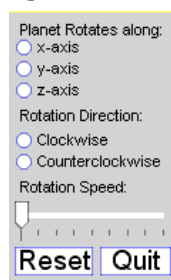


Figure 16: Control panel for the selected object.

Every time a user makes a new selection in the first set of radio buttons, the slider will be cleared (set to 0) where the bar will be placed to the starting point at the left. The system is able to remember the previous movements and states. Once a users adding a new rotation in an axis, the object will continue the rotations at other axes. As a result, the object is able to perform a set of rotations in my solar system.

This control panel also has two buttons. The first reset button is able to reset all changed objects back to their initial position with their initial movement setting. The second quit button is to quit the selection mode and come back to the moving solar system.

After entering the selection mode (select an object other than the Sun), the upper right corner will display a notification to remind users that they can quit the selection mode by clicking the “esc” key. As a result, my system gives users freedom to control the object and support multiple ways to quit the selection mode.

In selecting the four debris objects, the system will display a slightly different control panel at the bottom right corner, as shown in figure 17 and 18.

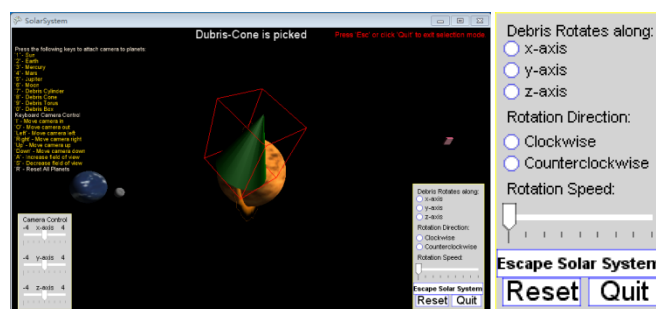


Figure 17: Selecting the debris (cylinder). Figure 18: Control panel for selected debris.

In addition to the existing functions, this new version of control panel has a new button “Escape Solar System.” The selected object will be scaled up and fly away from the solar system in observable speeds and movements. Again, users have two ways to reset the locations and movements of these debris objects by either pressing the “Reset” button or pressing the “esc” key.

All my models are from our CourseWorks site, and I have programmed all required functions and features. I also take into account the restricted computing power and memory spaces of computers. In order to make my solar system flow well, I did not use any loops in my implementation and I try my best to use simple methods to implement the functions.

Bugs of GoblinXNA that I found:

1. Unable to change the materials for the given objects. As a result, I cannot program additional features such as changing object color. I have no problems in changing colors for the debris objects.
2. The mouse picking is always activated even on the control panel. As a result, I have to unable the mouse picking when the third control panel is displayed.

Otherwise, the system will quit the selection mode once I click the control panel.

3. Newton.dll file need to be added in addition to the goblinXNA.dll. I thought goblinXNA.dll has included the Newton.dll library, but I was wrong. It takes me a whole afternoon to figure that out.