Tommy Inouye and Alfred Tan
ti2181 and avt2123
Computer Graphics
Professor Zheng
Final project

Purpose:

Our project will be a 3D rendering of the game Flappy Bird. The rules of the game are that by the click of a mouse, a bird object will move up. When a click or button is not pressed, the bird object will be moving down. The bird is continuously moving forward and there are an infinite number of pipes with varying sizes along the way. The objective of the game is to keep moving forward and avoid the pipes. If the bird collides with a pipe, the game is over and the game can be reset.

How to Run:

To run the program, basically just call "ant" at the root directory and everything will compile and run for you. When run, it is by default run without any models applied to it. Below are how to toggle between different modes.

Sources used

(1) Slick-Util
(2) LWJGL

Our approach

(1) Implement the basic game template
(2) Use slick-util library to add templates
(3) Add lighting and shading

(1) Implement the basic game template

We implemented the basic game engine. The idea was to randomly generate the pipes locations and to render them the more you progress. The collisions were detected by the locations of the edges between objects and any overlaps were automatically found and the game would automatically be reset. To start, all you need to do is click. Clicking makes the bird sprite jump and the idea is to avoid all the obstacles. Additionally, the bird rotates depending on the bird's movement so that it illustrates a falling and rising motion. We utilized drawing primitives, transformations, and textures.

(2) Use slick-util library to add templates

Once we got the basic game down, we added textures to every object and added sound effects for all types of actions. As a result, we were able to render the main sprite, the pipes, and all the necessary sound effects.

(3) Add lighting and shading

Before we even started, the first thing we decided to do was to move the vertex and fragment shaders to their own files. We then created a function that will open these files and return the string of it's context. Since this is a very common and relatively feasible function, we just looked

it up from StackOverflow at this link:http://stackoverflow.com/questions/16027229/reading-from-a-text-file-and-storing-in-a-string. Then we just added very simple shaders to our objects.

Contributions by each teammate:

Tommy Inouye: I implemented task 1 and half of task 3. I implemented most of the game engine and did all the collision management. I also fine tuned many of the game dynamics to ensure consistency and feasibility. I also added all the shaders.
Alfred Tan: I implemented task 2 and half of task 3. I looked at all available resources and decided to utilize the Stick-Util for the scene. I heavily used their texture and sound functions to overlay the objects that were made. I also implemented all the textures and sound effects. I also added the lighting to the scene.

Results:

We were successfully able to accomplish our goal. We were able to make a simple version of the game and made it three-dimensional. You can use the arrow key to rotate the view.

Additional instructions:

1) Right click on the mouse- Start game and to make the fish “jump
2) Left arrow key- Rotates the view leftward
3) Right arrow key- Rotates the view rightward

Resources

http://www.kilobolt.com/zombie-bird-tutorial-flappy-bird-remake.html
http://slick.ninjacave.com/slick-util/
http://goharsha.com/lwjgl-tutorial-series/