OpenGl Assignment

Computer Graphics

Semester 2 – 2019

Tristan Fazio

15492391

Main.cpp

Purpose

The main controller of the program. This file contains the initial settings setup, the input processing for keyboard and mouse, and event handling. The main game loop also exists within this function, iterating every frame and drawing the objects as they are needed.

Functions

void framebuffer_size_callback(*GLFWwindow* window*, int *width*, int *height*);

Handles the resizing of the opengl window when it is dragged.

void processInput(*GLFWwindow *window*);

Listens for user input via the keyboard. Has a series of if statements assigned to it that trigger when certain keys are pressed. Specifically, WASD for moving, E for pick up Sven, F for pickup torch, R for restart, and P for projection switch.

GLFWwindow* **initWindow(int *width*, int *height*);**

Initialises the necessary GFLW variables and enables GLAD.

void mouse_callback(*GLFWwindow* window*, double *xpos*, double *ypos*);

Listens for mouse input and adjusts the camera angle accordingly.

void scroll_callback(*GLFWwindow* window*, double *xoffset*, double *yoffset*);

Listens for the mouse scroll input, but currently has no other functionality

void restart();

Replaces all objects back at their starting positions and resets the game.

void toggleProj();

Toggles between the Orthographic and Perspective based projections available within openGl

void toggle_sven_distance();

Checks to see if the threshold for being close enough to Sven to allow pick up has been reached, if so, toggles on the setting so that Sven is now picked up by that player.

void toggle_torch_distance();

Check to see if the threshold for being close enough to the Torch to allow pick up has been reached, if so, toggles on the setting and the torch is now in the players right hand.

Animations

Since this is the main game loop function, there is no animation available.
However it does control movement of the camera by multiplying the direction vectors by the camera speed * frame rate.

Ground.cpp

Purpose

Displays a series of cubes in a grid to tile and represent the ground. Is covered with a dirt texture

Functions

void drawGround(unsigned int texture, glm::mat4 view, glm::mat4 projection)

Using a double for loop the contained sub is tiled throughout a 50x50 grid.

Animations

No animations

Sven.cpp

Purpose

Displays the dog character named Sven. He is made up of a torso, 4 legs, a snout and 2 ears.
A dog fur text is applied.
He is the objective of the game and needs to be picked up and rescued from watersheep.

Functions

void drawSven(unsigned int texture, glm::mat4 view, glm::mat4 projection,glm::vec3 cameraPos,bool svenEquipped)

Draws sven in the middle of the forest of the game world.

Animations

Sven attaches to the player and follows them when being picked up.
Currently now other animations are implemented for Sven.

Torch.cpp

Purpose

A small eye-catching torch to light the way for the player.
The torch is an aid for the player and will attach to the player when acquired.

Functions

void drawTorch(unsigned int tex_torch, glm::mat4 view, glm::mat4 projection, glm::vec3 cameraPos, bool torchEquipped)

Draws the torch at a specified location in the game world, in this case, in front of the wall.
It also textures and colours the torch to look wooden.
The lighting functionality was unfortunately not working and was stripped to ensure the project still built.

Animations

To be eye catching to the player, the torch rotates on the spot and bobs up and down on the spot. This is done by using the sin() function with the glfwGetTime()) function to generate a movement arc.

Trees.cpp

Purpose

Draws several trees around the world space to give the illusion of being surrounded by a forest.

Each tree has been specifically placed to provide a pleasing shape.

Each tree consists of two cubes, the trunk, textured with a brown coloured bark texture.
And the leaves, textured with a dark green coloured leaves texture.

Functions

void drawTrees(unsigned int tex_leaves, unsigned int tex_wood, glm::mat4 view, glm::mat4 projection)

A simple drawing function placing each tree at a specified location. Does not have any extra functionality.

Animations

No animations are provided as the trees are static.

Wall.cpp

Purpose

The wall class exists to draw an obstacle for the player. This class consists of 3 components.
The fence post cubes, that have been stretched to be a thinner and taller than the rest of the wall, marking an entry to the forest.

The wall itself, that is placed by for loops spanning to the edge of the map.

And a sliding door obstacle, designed to give the player an obstacle to avoid to get through the gap.

Functions

void drawWall(unsigned int texture, glm::mat4 view, glm::mat4 projection)

Draws the wall components and applies the sliding animation to the sliding door component of the wall

Animations

The door is made to bounce in a similar fashion to the torch. A sin() is applied to the time variable, and constrained within the limits of the y axis. It then gets animated a long said axis, allowing the player a short time window into which they can step through.

Watersheep.cpp

Purpose

Watersheep is the antagonist of the game. He has captured Sven. Watersheep has been strategically placed through the trees, so that he can be watching as the player approaches.
Intended design is for watersheep to rotate and watch the player, and then be activated and chase the player once sven is picked up.
However neither function is currently working, as watersheep ends up teleporting into the wall.

Functions

void drawWatersheep(unsigned int texture, glm::mat4 view, glm::mat4 projection, glm::vec3 cameraPos,float cameraSpeed, bool svenEquipped)

Draws watersheep and handles the animating of him towards the player once an activation occurs.

Animations

On activation watersheep is supposed to start chasing the player.
However there is a bug with the algorithm, the distance isn't being calculated properly between the two vectors, and watersheep is instead teleporting to inside the wall object.

Stb_image.c

Created by Sean Barret, this class handles the loading of different image file types into the textureloader.

https://github.com/nothings/stb/blob/master/deprecated/stb_image.c

Textureloader.cpp

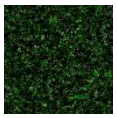
This class was designed using code from learnopengl.com.

It was designed to take a file path and then create a texture out of the supplied file, returning a pointer for the program to use.

Textures

All textures were sourced from the free section of Turbosquid.com

https://www.turbosquid.com/FullPreview/Index.cfm/ID/342917



https://www.turbosquid.com/FullPreview/Index.cfm/ID/171720

https://www.turbosquid.com/FullPreview/Index.cfm/ID/351955

https://www.turbosquid.com/FullPreview/Index.cfm/ID/659888

https://www.turbosquid.com/FullPreview/Index.cfm/ID/253873

https://www.turbosquid.com/FullPreview/Index.cfm/ID/370779