PA Final: Aircraft

By Ronn Quijada and Sean Stevens

Project Overview

In a world where there is one gun seeking vengeance and a bunch of airplanes, the player must destroy as many airplanes as possible. Over time, you ask yourself “is all fair in love and war?” only to realize that the next wave of airplanes has arrived.

Build Instructions:

1) Open the terminal in the PA Final directory and write the following commands:
    a. mkdir build
    b. cd build
    c. cmake ..
    d. make
2) Move the assets folder into the build directory
3) Run the following command: ./Tutorial

Required Libraries:

- SDL
- OpenGL
- Bullet 2.87
- GLM
- Assimp 3.3.1
- Magick++

Controls:

- Use the mouse to aim the camera
- Press spacebar to fire gun
- Press ESC to quit program

Graphics Features

- Model Loading: assimp is used to load a .obj file from the assets folder. This object is rendered using OpenGL.
- Texture loading: assimp is used to load a .png file from the assets folder.
- Mipmap Textures: Textures are mipmapped so they retain detail at a distance.
- Skybox: Six planes and six textures are used to create a cube that represents the sky. Each face in the box is rendered first, so they are behind everything.
- Physically-based Rendering
- Ocean Waves: A shader moves the ocean water texture by offsetting the uv coordinates. Also, each vertex is offet in the y axis by a sine and cosine function, creating waves.
- Billboarding: A 2D plane that always faces the camera is used during explosions.
- Lighting
- Sound Effects: A sound system using SDL was created so the gun can create noise when firing bullets and when the planes explode.
- Raycast physics: The gun uses a raycast system to detect when a plane has been hit instead of physical projectiles because the bullets move too fast to consistently register collision.
- Collision Detection: A collision between each plane and the ocean can be detected so the game knows when the plane should explode.
- Pathfinding: The planes have a rudimentary pathfinding system that has them fly towards and fly around a point for an arbitrary period of time.
- GUI: IMGUI is used to create a GUI overlay on the screen. This gives information for the number of planes that were destroyed and it could be used for debugging purposes without having to refer back to the terminal.

Code Outline

Animator.h – Defines pathfinding protocols for airplanes.

Camera.h – Defines camera used in window

Light.h – Defines objects used to illuminate world

billboard.h – Defines objects that face towards the camera

engine.h – Defines game loop, initializes graphics and physics engine.

event.h – Helper class for SDL keyboard events and physics world events

framebuffer.h – Helper class for frame buffer

graphics.h – Helper class for OpenGL functions

graphics_headers.h – Class that contains GLM, OpenGL, and custom structs

gui.h – Helper class for GUI functionality

gun.h – Class for the player controls and rendering the gun model, derived from physObject

imconfig.h – Configures IMGUI

imgui.h – Header for IMGUI functions

imgui_impl_sdl_gl3.h – SDL implementation for IMGUI

imgui_internal.h – Header for IMGUI functions

kineObject.h – Defines kinematic objects, derives from physObject

loader.h – Loads textures and objects that are used to render objects

obj.h – Loads .obj files, used by loader.h

object.h – Base class for all objects that need to be rendered

ocean.h – Class that loads and renders the ocean, derived from object

physObject.h – Defines objects that have physics, derived from object

shader.h – Helper class for loading, compiling, and linking shaders to OpenGL

sound.h – Defines sound functions used in SDL

stb_rect_pack.h – Used by IMGUI

stb_textedit.h – Used by IMGUI

stb_truetype.h – Used by IMGUI

window.h – Helper class for SDL windows

world.h – Helper class that contains all objects that are rendered.