

OpenGL Breakout Game Project

Aly Sewelam

September 5, 2024

Contents

1	Summary	3
2	File Structure	3

1 Summary

Classic game made in opengl following the learnopenGL tutorial

2 File Structure

- game contains primary data for game, including player, ball loading levels and textures etc
- collision contains logic code for collision detection The AABB - AABB is for detecting collisions between two rectangles while the AABB - Circle detects collision between a circle (our ball) and rectangles (our bricks)
- ball_{object} Defining the ball values needed radius, position and stuck which is used for detecting if ball is on the player platform or not
- game_{object} class for all objects that will be rendered in our game such as our player and bricks on the level (ball object is a child of it)
- sprite_{render} responsible for rendering sprites using the shader and texture files
 - contains the boilerplate code responsible for binding the VBOs to initiate rendering data
 - after initiation of rendering data, DrawSprite function can be used to draw our objects using the textures and shaders
- resource_{manager} responsible for loading our shaders and resources instead of hardcoding the values
- program the entry point for our game where the game object is initialised
- game_{level} responsible for loading levels from text files (.lvl extension) the level format is a matrix of numbers each number representing a type of brick each type has a different color (0 is no brick) a Gameobject vector is created using the init function which is then rendered by the game by calling the draw function IsCompleted() is responsible for checking if the level is won by checking if all bricks have been hit
- shader file for compiling shaders and loading them using the resource manager

- texture wrapper class for textures so they can be handled by resource manager
- `stb_imageimpl` library for loading images for textures