CS480

PA 11: Labyrinth

12/07/15

Marco Rivas, Martin Revilla, Truman Chan

Overview

Extra Credit

- Music/Sound effects
- PS4 controller
- Top 10 Scoreboard
- Multiple levels

User Manual

To build and run the program:

cd build Then: *make*

The executable will be put in the “bin” directory.

cd bin Then: *./Labyrinth*

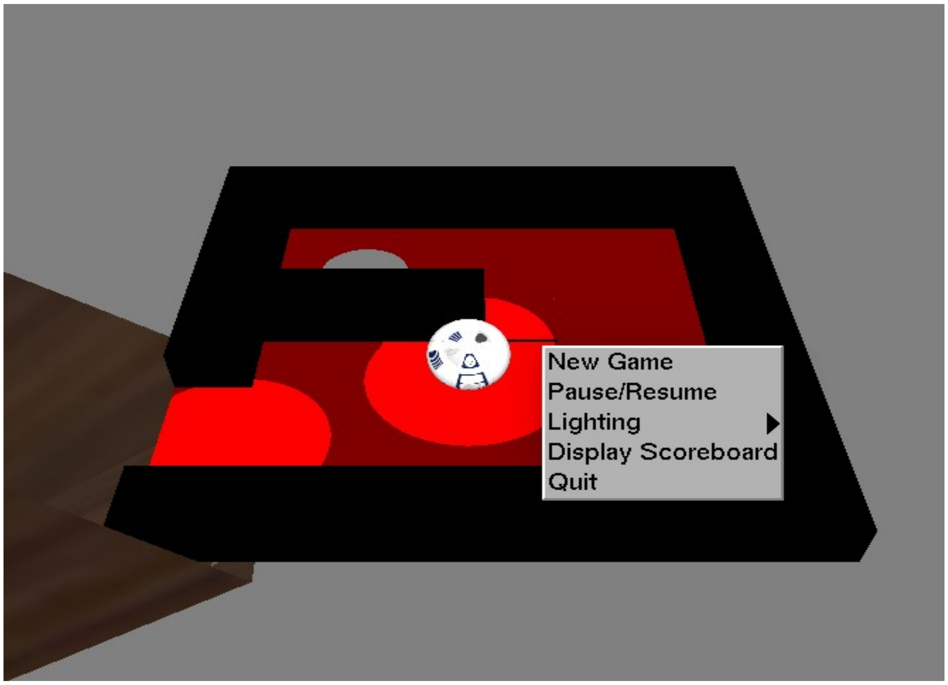
To clean (removing object files and executables):

cd build Then: *make clean*

Description

The purpose of this assignment is to produce a 3D simulation of the Labyrinth game that is as realistic to the actual game as possible, yet still being a playable and an enjoyable experience for the end use.

There are three levels to the game. The goal for each level is marked by a spotlight! Once a goal has been reached, a checkpoint for the furthest level is saved. Avoid the holes and get to the final goal with the best time! If the ball falls into a hole, the ball will respawn at the last checkpoint. Access the menu to do stuff!



Controls

Mouse

Mouse Input	Action
Hold left button and move	Tilt table
Right-Click	Open menu
Scroll Wheel up/down	Zoom in/Zoom out

Keyboard

Keyboard Input	Action
'x'	Play/Pause song
Spacebar	Play next song
Up Arrow Key	Move camera up
Down Arrow Key	Move camera down
Left Arrow Key	Move camera left
Right Arrow Key	Move camera right
Esc	Exit game

PS4 Controller

PS4 Input	Action
Left Analog Stick	Tilt table

Technical Manual

Classes

main.cpp
The main driver of the program. It calls on Graphics.cpp, GameEngine.cpp, MusicPlayer.cpp, GameController.cpp, and PhysicsEngine.cpp to create the game, and run it.

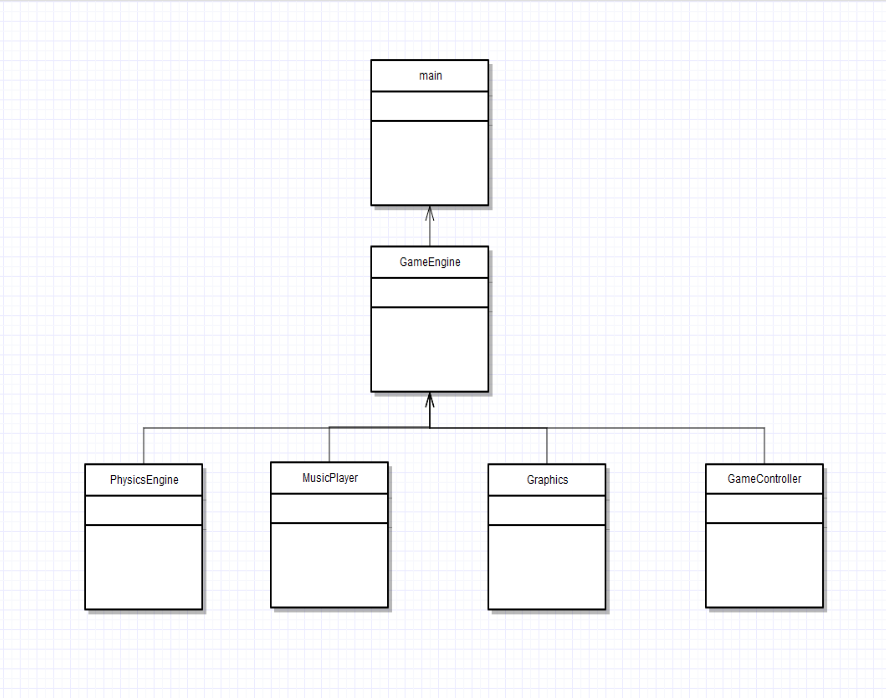
GameEngine.h
The logic of the game is handled in this class. It uses Graphics.h, PhysicsEngine.h, MusicPlayer.h, and GameController.h to properly update the state of the game.

Graphics.h
This class renders the labyrinth game including the maze, ball, and light. It's also responsible for the scoring text shown on the screen. Assimp is used to load the textures for the game.

PhysicsEngine.h
Collisions in the game, primarily how the puck interacts with the paddles and the walls of the table, are done in this class. The Bullet library was used to handle collisions between bodies and how they would react.

MusicPlayer.h
This class stores the songs and sounds used for the game. "#include <SFML/Audio.hpp>" was used to play the WAV/OGG files loaded.

GameController.h
This class handles PS4 controller compatibility. "#include <SFML/Window.hpp>" was used to access joystick axis for its use in the game.



Issues

Segfaults occurred after quiting game with menu interactions. Issue was due to newest version of VMWare Player, and was resolved by downgrading to version 6. Also had an issue with the ball bouncing off the walls, resolved by the lowering restitution of the sphere.

Future edition will be Star Wars themed. Texture for level 1 would be the Death Star, level 2 would be Hoth, and level 3 would be Endor. Once all three mazes are completed, an advertisement for Star Wars: Episode VII, The Force Awakens, would play coming out December 18, 2015. Tickets available for purchase at www.Fandago.com/TheForceAwakens.