Labyrinth

Project

By Michael Vacchina

Content

Project Overview……………………………………………………………………………………………………………………………………3

User Manual…………………………………………………………………………………………………………………………………………..4

Tech Manual…………………………………………………………………………………………………………………………………………12

Project Overview

Overall, I enjoyed this project. I did not use any external libraries besides gl, glut, and glm. The physics and collisions in my program were created by me. My implementation of the labyrinth game is object oriented utilizing inheritance when acceptable. I would not call my project completely finished as there are still bugs, some unimplemented functions, and some of the code is “ugly.” However, none of those affect the performance as seen by the user (except some bugs). My project does complete every required part listed in the labyrinth evaluation. I did also accomplish some of the extra credit that was available for this project. The ball does bounce of walls properly. The ball falls through holds. The menu has changeable settings. Lastly, there are two mazes that they user can play.

User Manual

Contents

Getting Started………………………………………………………………………………………………………………………………………..5

The Menu…………………………………………………………………………………………………………………………………..……………6

Controls…………………………………………………………………………………………………………………………………..……………10

Getting started

When a user first starts my program they will be presented with the screen as shown in figure 1.

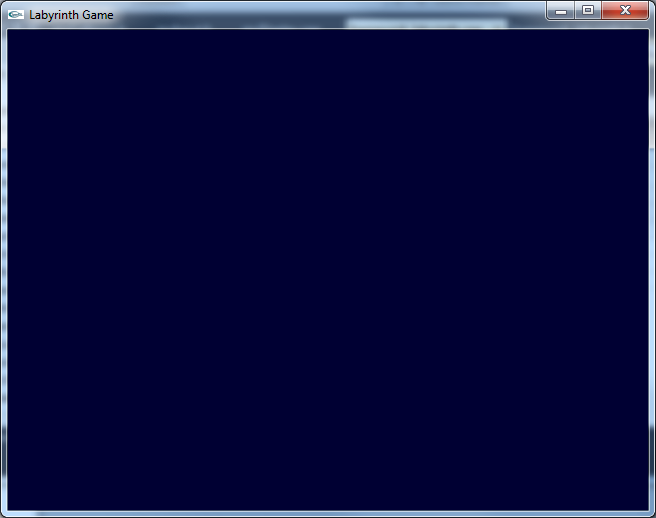


Figure 1

The user must right click within the window to be presented with the menu as shown in figure 2.

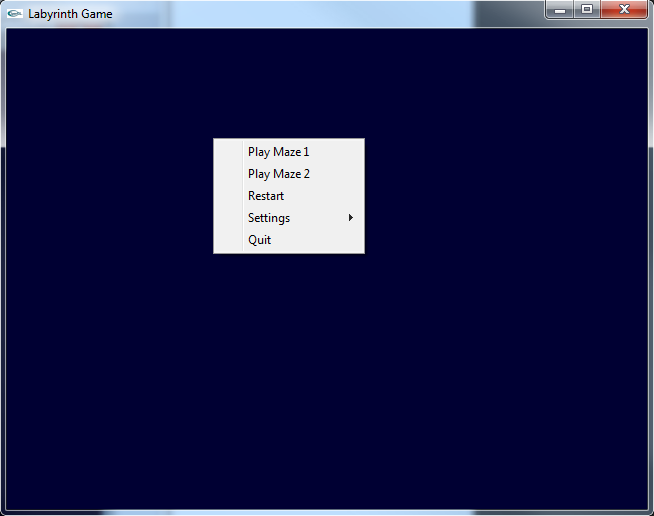


Figure 2

From the menu the user can pick which maze they want to play, restart a current attempt at playing (until they select play maze 1 or 2 this option does nothing), change settings, or quit the game. The goal of the game is to move the ball by using the controls (explained in section Controls) to move the ball from the starting circle to the goal circle without falling into the holes. At any point the user can select restart to set the maze orientation back to default and the ball back to the starting circle.

The Menu

Play Maze 1

This is the first maze the user can play. Each maze has a teal floor with holes, blue walls, a green starting circle, a red goal, and a yellow ball. This maze in particular has 5 holes where one is larger than the others, and 3 non-border walls. A picture of it is shown in figure 3.

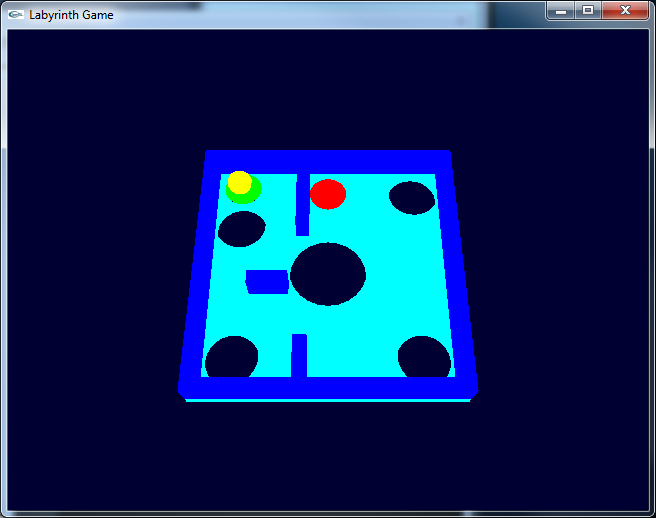


Figure 3

Play Maze 2

This is the second maze the user can play. Each maze has a teal floor with holes, blue walls, a green starting circle, a red goal, and a yellow ball. This maze in particular has 6 holes, and 4 non-border walls which are connected in pairs of 2. A picture of it is shown in figure 4.

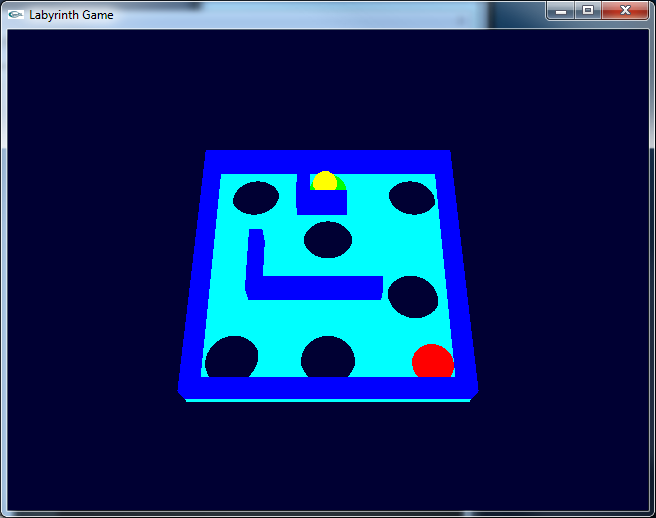


Figure 4

Restart

This menu option resets the current maze to the starting orientation and returns the ball to a resting state on the starting circle. This option or one of the Play Maze options must be chosen when the ball falls down a hole as the ball will not be replaced on its own.

Settings

This menu option shows a submenu where the user can alter settings in the game which will take effect immediately. The submenu is shown in figure 5. Each option in the submenu is explained further below. It is important to note that none of these options have a cap on their effect in the game. For example, that means that there is no maximum gravity. It will simply keep increasing if you keep clicking the button. It is not recommended to increase or decrease any of these settings more than 3 times from the default (An increase will undo a decrease and vice versa). Good performance of the program cannot be guaranteed under such circumstances.

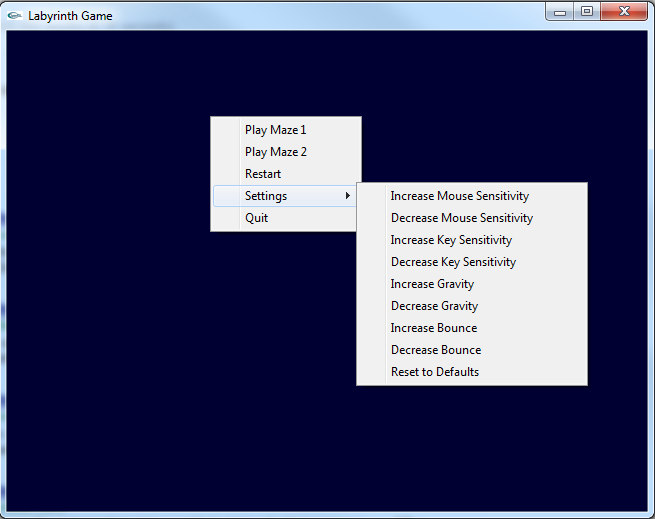


Figure 5

Increase Mouse Sensitivity

This option will increase the mouse sensitivity by a factor of 2. Basically, one would only need half of the movement to get the same affect.

Decrease Mouse Sensitivity

This option will decrease the mouse sensitivity by a factor of 2. Basically, one would only need twice the movement to get the same affect.

Increase Key Sensitivity

This option will increase the key sensitivity by a factor of 2. Basically, one would only need to hold down a key for half as long to get the same affect.

Decrease Key Sensitivity

This option will decrease the key sensitivity by a factor of 2. Basically, one would only need to hold down a key for twice as long to get the same affect.

Increase Gravity

This option will increase gravity by a factor of 2. Gravity starts out as -9.8 so selecting this option once will make the gravity -19.6.

Decrease Gravity

This option will decrease gravity by a factor of 2. Gravity starts out as -9.8 so selecting this option once will make the gravity -4.9.

Increase Bounce

This option will increase the bounciness of the ball by a factor of 1.25. When a ball bounce it will lose 50% of its velocity so after selecting this option once the ball will lose 37.5% of its velocity following a bounce. It is possible to increase this setting enough to cause the ball to gain velocity when bouncing. This is unwise and not recommended.

Decrease Bounce

This option will decrease the bounciness of the ball by a factor of 1.25. When a ball bounce it will lose 50% of its velocity so after selecting this option once the ball will lose 60% of its velocity following a bounce.

Reset to Defaults

This option will return all settings to their default values.

Controls

Displaying the Menu

Right clicking in the screen brings up the menu. This causes the display to be paused but the ball will keep on moving. In this state, the collision detection does not work so the ball will move through walls and over holes. This is an undesired affect so it is not recommended to right click when the ball is moving.

Rotating the Maze

The maze can be rotated by 3 different controls. It can be rotated by the arrow keys. It can also be rotated by the aswd keys. Lastly it can be rotated by holding down the left mouse button and moving the mouse. The key controls can be used together without problems to get the maze to rotate quicker. However, the key controls cannot be used in unison with the mouse control. To do so is unwise and not recommended.

Quitting

In addition to being able to quit from the menu, the user can also quit by pressing ESC.

Tech Manual

Contents

Physics…………………………………………………………………………………………………………………………………………………..13

Collision………………………………………………………………………………………………………………………………………………...13

Known Bugs…………………………………………………………………………………………………………………………………………..13

General Issues……………………………………………………………………………………………………………………………………….14

General Comments………………………………………………………………………………………………………………………………..14

Physics

For the physics, I would rotation the standard gravity vector by the inverse of the rotation for the maze. I would then use the x and z values of the result as the acceleration of the ball. Physics would be applied to the ball before the ball was rotated with the maze. The movement that the physics would apply to the ball was restricted to the xz plane. The ball does not spin at all; it simply slides over the floor of the maze.

Collision

For the collision, I would start off by checking if the ball has entered the goal or holes. If it entered a hole, I attempted to apply a falling bounce to the ball if it would pass through the floor of the maze if it fell straight down. I would then add downward acceleration to the ball and let it fall until the user changed the maze or restarted. I had the walls of the maze sorted into two vectors one by the x values of the walls and the other by the z values. I would then go through the sorted vectors checking if the ball with its radius was in the wall. This is my collision detection. It is the case that since the ball is a sphere this collision detection could have false positives when the ball is near a corner of a wall. After the collision detection, I would determine the segment(s) of the wall that the ball hit and calculate their normal. I then moved the ball out of the wall based on that normal and bounced it by determining its new velocity using that normal. In an attempt to remove a bug that occurs when the ball rolls down a wall, I would set normal forces on the ball (the normal of the segments it hit). These normal forces would be observed in the physics and acceleration would not be applied against that normal force. However this did not fully work as intended as the normal force would not be reset on the next collision detection step and the ball would move into the wall. This causes a strong friction like force when the ball rolls down a wall.

Known Bugs

* The right click menu causes they system to “pause” but the delta time does not pause meaning the next dt will be quite large and the ball can pass through walls and over holes because of it.
  + I was unable to think of a quick solution to this in the allotted time as the mouse callback does not catch the right click that causes the menu to come up.
* Mouse controls and key controls used as the same time have weird effects.
* Since there are no caps on the settings that the user can change, it is possible to increase the balls speed enough that collisions will not be properly detected.
* Friction like effect on the ball rolling down a wall.
  + This bug was discussed in the collision section.
* Ball loses velocity when entering a hole and gains a specific velocity to stop it from passing through the floor.
  + Makes it unrealistic.

General Issues

Some of the code is not fully completed as there simply was not enough time. The only function specifically was the destructor for mvObjects as it was causing a segmentation fault. It was not causing a large or continuous memory leak so it had a low priority. It is not done because there simply was not enough time.

The fall physics are a bit dodgy as I did not actually implement collision detection with the floor.

I do not like the menus as they are not what I think they should be, but I was unable to determine how to put text on the screen in order to do a menu I would be more comfortable with.

The obj loader is incredibly simple. It only load vertices and faces and it expects each face to be a triangle.

Nothing really happens when you reach the goal. The console just spits out that you in the goal until you get out of the goal.

It would have been nice to display text on the screen and show the time to have a metric for score and consequently a score board of the top scores.

I’m not sure I much like the references to the objects in framework, physics, and collision, but I was unable to think of the better way at the time.

If the ball manages to get inside a while by moving fast enough or a large time step, no collision response is formed. Generally, I do not think this should happen so long as the computer is not terrible and the gravity not increased too much.

General Comments

I do not think I would do much differently. There was not enough time to do things more complex than I already had done. It would have been nice to have proper menus for the project, but the popup menu is okay. If I had more time I would try to improve the physics to actually work on forces that way the friction like effect on the ball rolling along a wall would not happen. I would also have liked to do proper falling of the ball, but I think that would require 3D collision detection as opposed to the 2D collision detection that I implemented.