Labyrinth Guide: Graduate Student 680

Liesl Wigand, Khoa Tran

November 25, 2013

1 Overview

Labyrinth is a non-branching path that the user must traverse through to reach the center of a board. In this game, Labyrinth is a hybrid between labyrinth and a maze game. Instead of traversing through a non-branching path to the center, the user is to move a ball through a non-branching path from one end of the board to the opposite side of the board.

To move the ball from one end of the board to the other end, the user can tilt the board respect to the center of the board to roll the ball through the maze. In addition, the use must dodge the holes along the path.

1.1 Features

Labyrinth Requirements:

1. Board, ball, and knobs
2. Texture on board and ball
3. Maze on board
4. Holes/winning position marked
5. Basic physics
6. Multiple lights
7. Movable viewpoint (camera)
8. Display score
9. Menus (start, quit, pause)
10. Colors and textures
11. Controls to tilt the board
12. Clock (for scoring)
13. Multiple balls
14. Multiple levels

Extra Credits:

1. Sky box
2. Mouse controls
3. History of top 5 scores

2 User Manual

1. **Software Requirements.** If any of the libraries are missing, please run the commands or follow the instructions listed below each set of libraries.

   (a) cmake, GLM, glew, glut
   sudo apt-get install freeglut3-dev freeglut3 libglew1.6-dev
   sudo apt-get install libglm-dev

   (b) devil
   sudo apt-get install libtiff-dev libdevil-dev

   (c) Assimp 3.0, and Bullet.
   For Assimp, go to: *http://clanlib.org/assimp-linux.html*
   For Bullet physics, go to: *http://bulletphysics.org/mediawiki-1.5.8/index.php/Installation*

2. **Compile and running Labyrinth**

   (a) Run cmake and make in build folder
   cd build
   cmake ..
   make

   (b) Execute program in bin folder
   cd ..
   cd bin
   ./pa

3. **Game Controls** The controls in this game allows the user to tilt the board respect to the board center by keyboard or mouse.
   Keyboard:

   (a) **a or A:** Tilt the board left.

   (b) **d or D:** Tilt the board right.

   (c) **w or W:** Tilt the board up.

   (d) **s or S:** Tilt the board down.

   Mouse: Moving the mouse cursor in a respective direction will emulate the keyboard press.

   (a) **m or M:** Activate mouse control

   (b) **n or N:** Deactivate mouse control

4. **Perspective** Playing the game can be done in 2nd person or 3rd person point of view. **2nd person point of view** has the camera focused to the center of the board. To move the camera around, the user can control the camera distance, height, and rotation.

   - **arrow key up and arrow key down:** Change the height between the camera and board center
   - **arrow key left and arrow key right:** Change the rotation between the camera and board center
   - **Page Up and Page Down:** Change the distance between the camera and board center

   3rd person point of view follows the ball from a distance with part of the maze viewable. In order to enter this mode, press:

   - **o or O:** Activate 3rd person point of view.
   - **p or P:** Deactivate 3rd person point of view and return back to 2nd person point of view.

3 Technical Manual

There were a number of issues with this project. The movement of the board was especially difficult. Moving the board in the physics world quickly caused, the ball to fly through the board or off of it. Rotations were set as quaternions then switched to Euler, and this caused some issues with the speed of rotation.

There are certain texture images which are loaded correctly and others which are not. There did not seem to be anything consistent between these: they varied in size, color, and format. The sky box loaded perfectly in Blender, but when loaded in OpenGL the top and bottom faces were swapped.

Resetting objects in the physics world continues to cause problems. If objects are not all reset at once, and the world updated to reflect their new locations, then the physics world can behave as though the objects remain in their old positions for several steps. There was some confusion about whether the position of these objects should be set in the RigidBody, or the MotionState, and whether after this is set the object needs to be removed and added back into the PhysicsWorld.