

**Computer Graphic Project**

**SUBMITTED BY:**

Shahrukh

(SP12-BCS-202)

Talha Abbas

(SP12-BCS-220)

**SUBMITTED TO:Ms. Saadia Maqbool**

**DATE: May27,2014**

**Idea**

The idea is about to make a maze game in *Processing.* Just like other maze games where the player has to find the right path through the maze starting from one end to the other. Different difficulties are made in different games to put a challenge for the player. Examples of some popular maze games are as follows:



**Description of Model**

As stated earlier that the game is developed under *Processing 3D* Environment. First the whole wire frame of the *maze* has been made. Then the color texture was given to boxes for realistic view. The game is in camera view; means the player will have a camera view through the game. The player can view the wire frame by hitting the “W” key from the keyboard. Moreover the difficulties set for the player are:

* Hitting the walls will result in restart from the *Start* point.
* The player will have a time period of 60 seconds to reach to the end point otherwise the game will be ended.

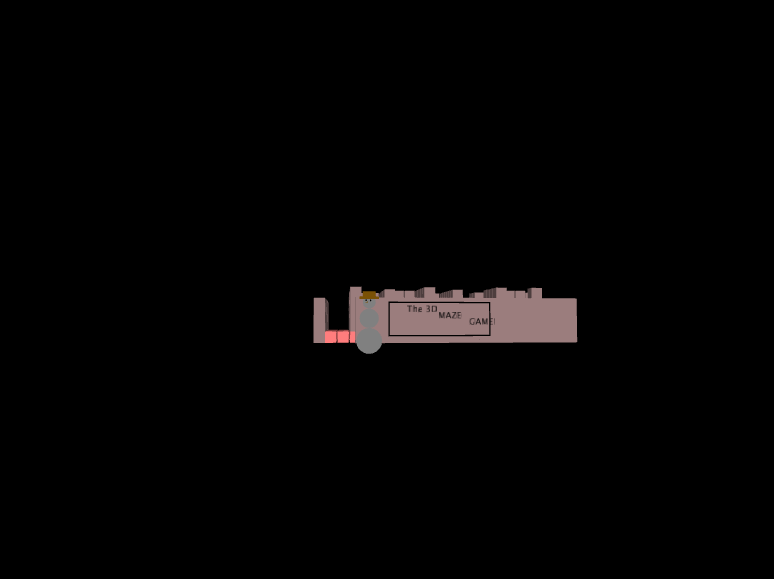
**User interaction**

Player can move along the path by pressing keys to move left, right, forward & backward respectively. To change the direction, the player will have to move the mouse in the respective direction.

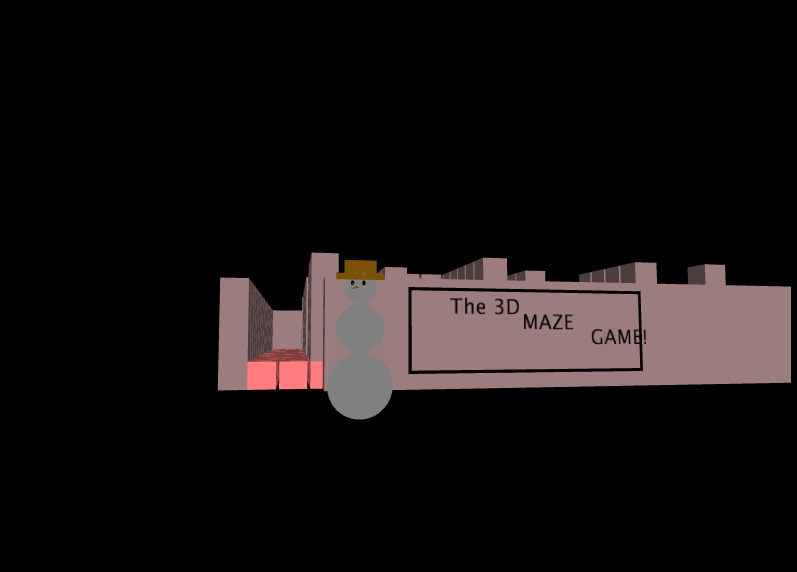
When user presses ‘w’ key wire frame of the scene is displayed. They keyboard’s ‘l’ key can be used to change the lighting effects.

**Screenshots**

1. **Start Up Look**



1. **Startup closer snap shot.**

****

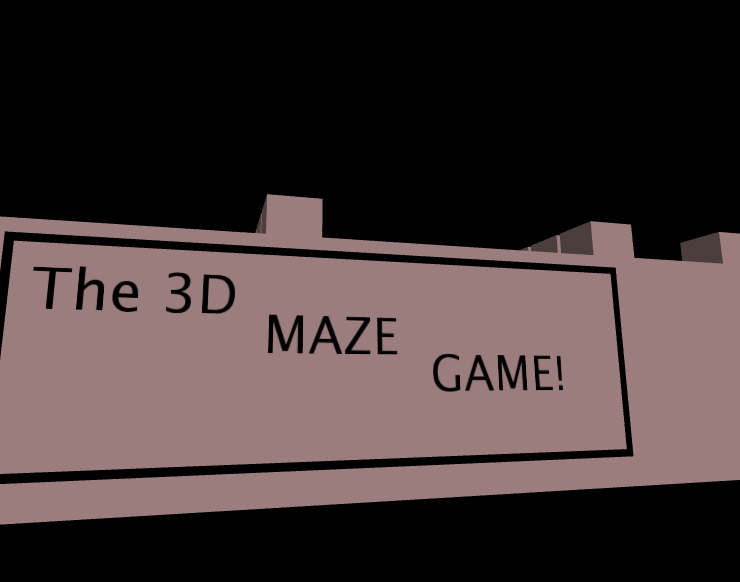
1. **Starting location of game (no lights).**

****

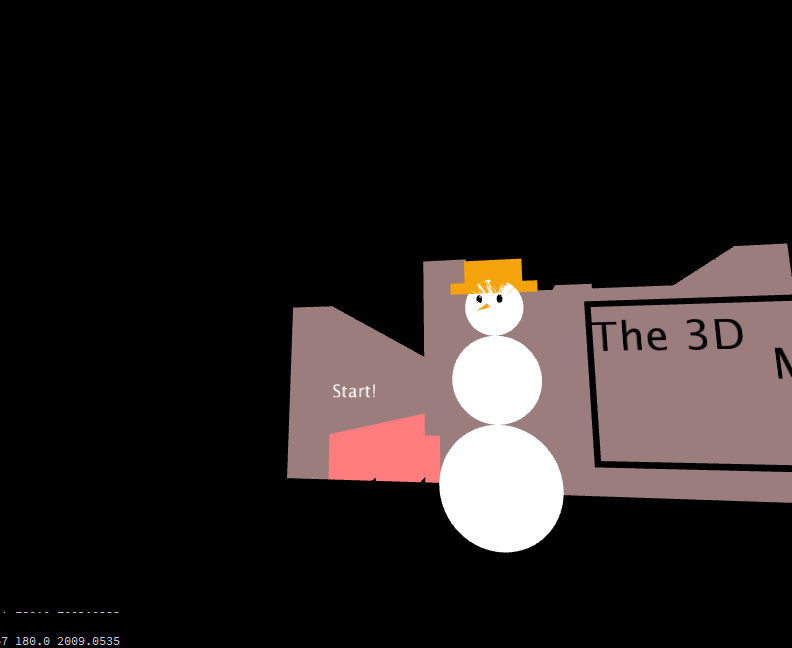
1. **Starting location of game (lights).**

****

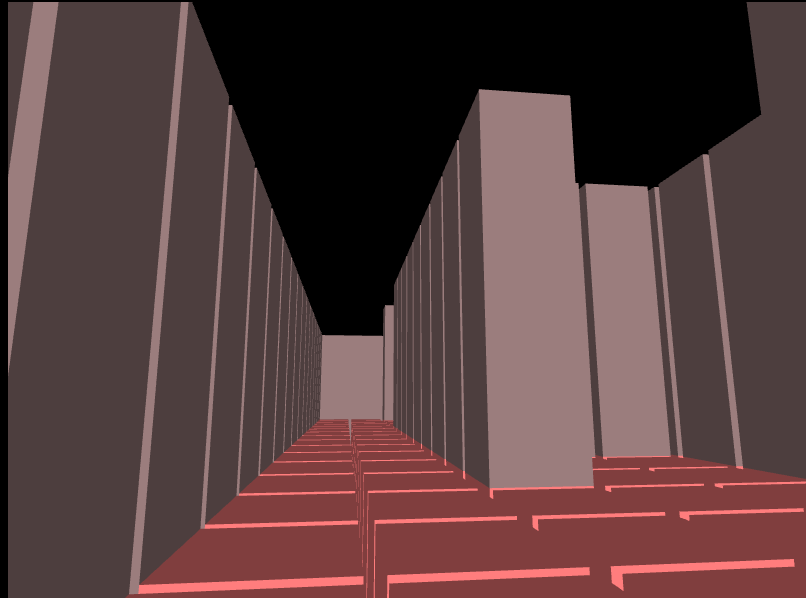
1. **Game Title**

****

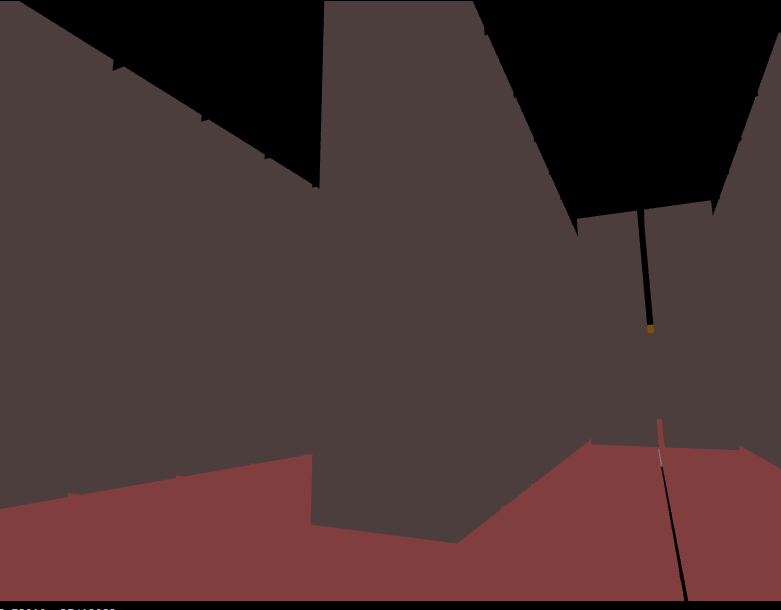
1. **Startup Snow man.**

****

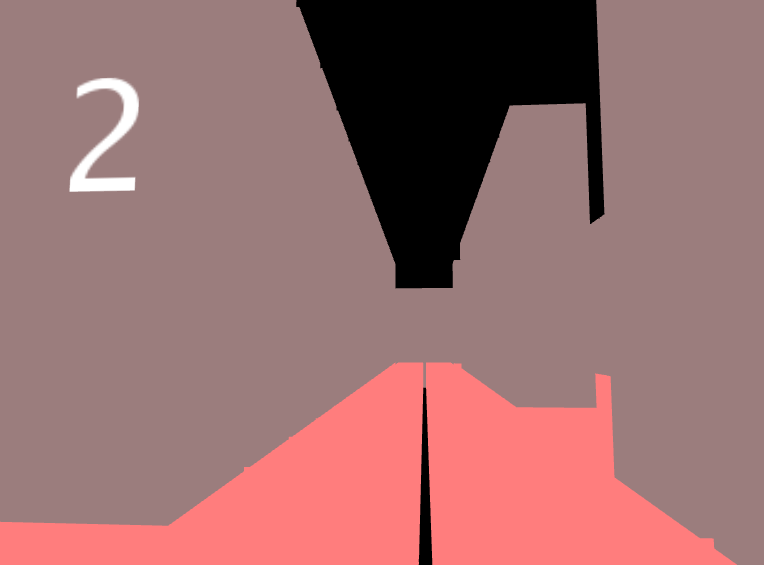
1. **Maze Lane 1.**

****

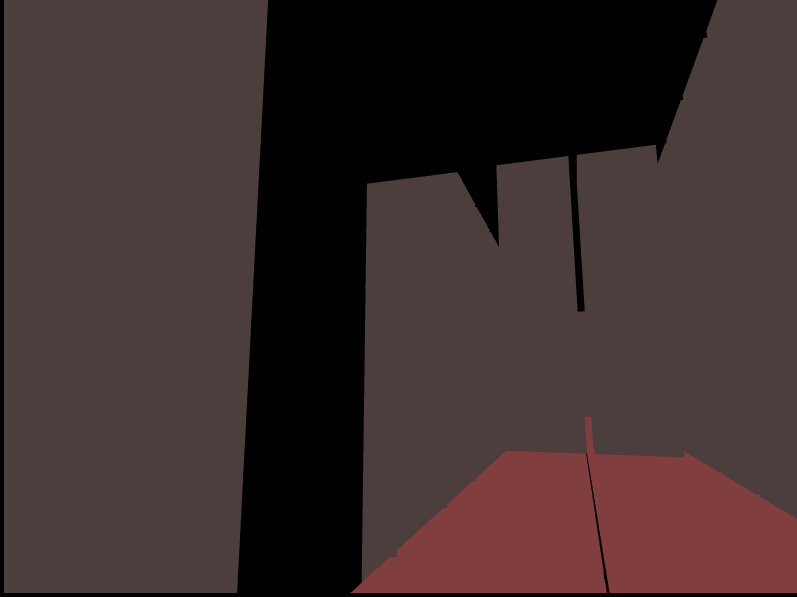
1. **Maze Lane 2 and 3.**

****

1. **Maze Lane 4 and 5.**

****

1. **Maze Home.**

****

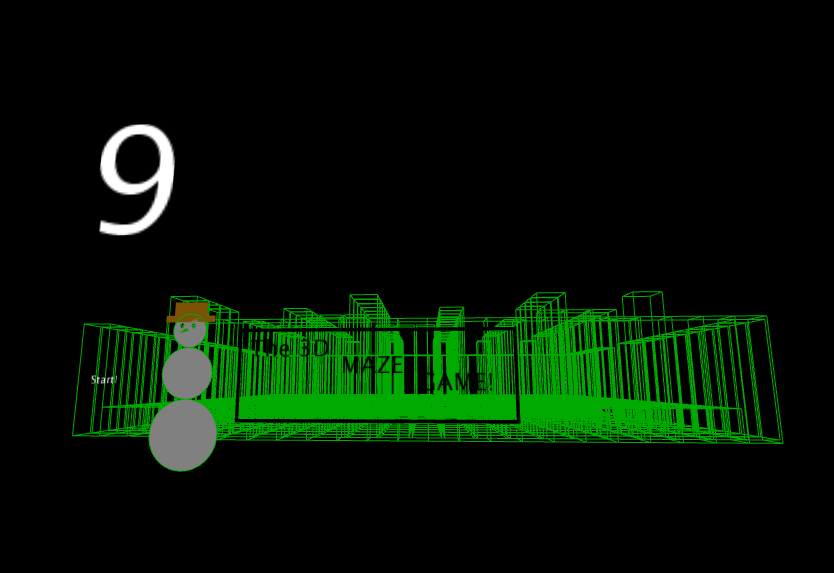
1. **Game Over Screen**

****

**12 You Win Screen.**

****

**13 Wire Frame of the game.**

****

**Activity Diagram**

**Start**

**Time Up**

Yes

No

**Wall hit**

Yes

No

**End**

**Algorithm**

1. All global variables are defined at start.
2. The setup() function is called by default, in which all variables are initialized and screen settings are defined.
3. Then draw() function is called which redraws the scene at 60fps. It also contains all the functions which are used for collision detection, timers, movement, updating the view, drawing walls of the maze which can be filled or wireframed.
4. decreZ() function gradually moves the camera from remote coordinates to start of the frame by decrementing in Z coordinates.
5. checkDest() function determines where the player has reached at destination by comparing coordinates.
6. collisionDetection() detects collisions with walls, and whenever collision occurs the camera moves to the start of the maze.
7. lifeUpdate() function decrements the lives of the player whenever collision occurs; there are only one life.
8. drawWall1() function is used to draw left boundary wall of the maze using nested for loop and boxes.
9. drawWall2() function is used to draw all inner walls of the maze, path of the maze by leaving gap of two boxes and the right out wall.
10. drawWall3() and drawWall4() draws the back and front walls respectively.
11. drawFloor() simply draws the floor on which the camera moves.
12. All wire\_wall() functions and wire\_floor() draw the wire frame of the whole scene.

# Bibliography

Processing. (2010, August 5). *Processing*. Retrieved May 26, 2014, from Processing: https://www.processing.org/examples/

Processing. (2012, September 26). *Processing*. Retrieved May 26, 2014, from Processing: https://www.processing.org/tutorials/

Processing. (2013, February 16). *Processing*. Retrieved May 26, 2014, from Processing: https://www.processing.org/reference/