Assignment 1

Sanchit Jain (160050043), Sneha Bhakare (150050040)

```
   ____                                      _____
  /      \    __    ___    ___              \       \ ___   ___   ___      ___
 /   \  / \  \\__   \ \__   // _  \  |       _/   \  / \   /    \_/ _  \
/     Y      \/ _  \/    /\  __/   |    |   \  |  /   |   \   |  \ __/|   | \/
\___|__     (___ /___  \\__  >    |___|_  /___/|___|  /___|   /\___  >_|
       \/        \/         \/   \/          \/           \/         \/   \/   \/
```
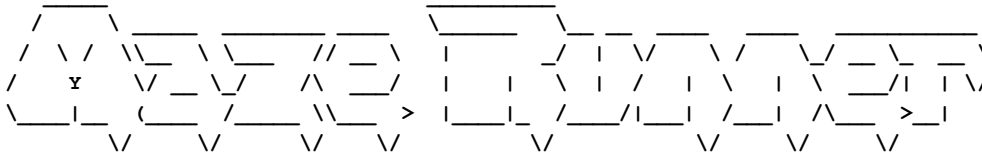
Game Design

Game Description : Maze Runner is an AR game with multiple players. Initially, hold the marker parallel to camera plane. A player can move the ball in the maze by tilting the marker. Gravity is the only force acting on ball which is affected by the orientation of marker. The fastest player with the least score(aka time) wins.
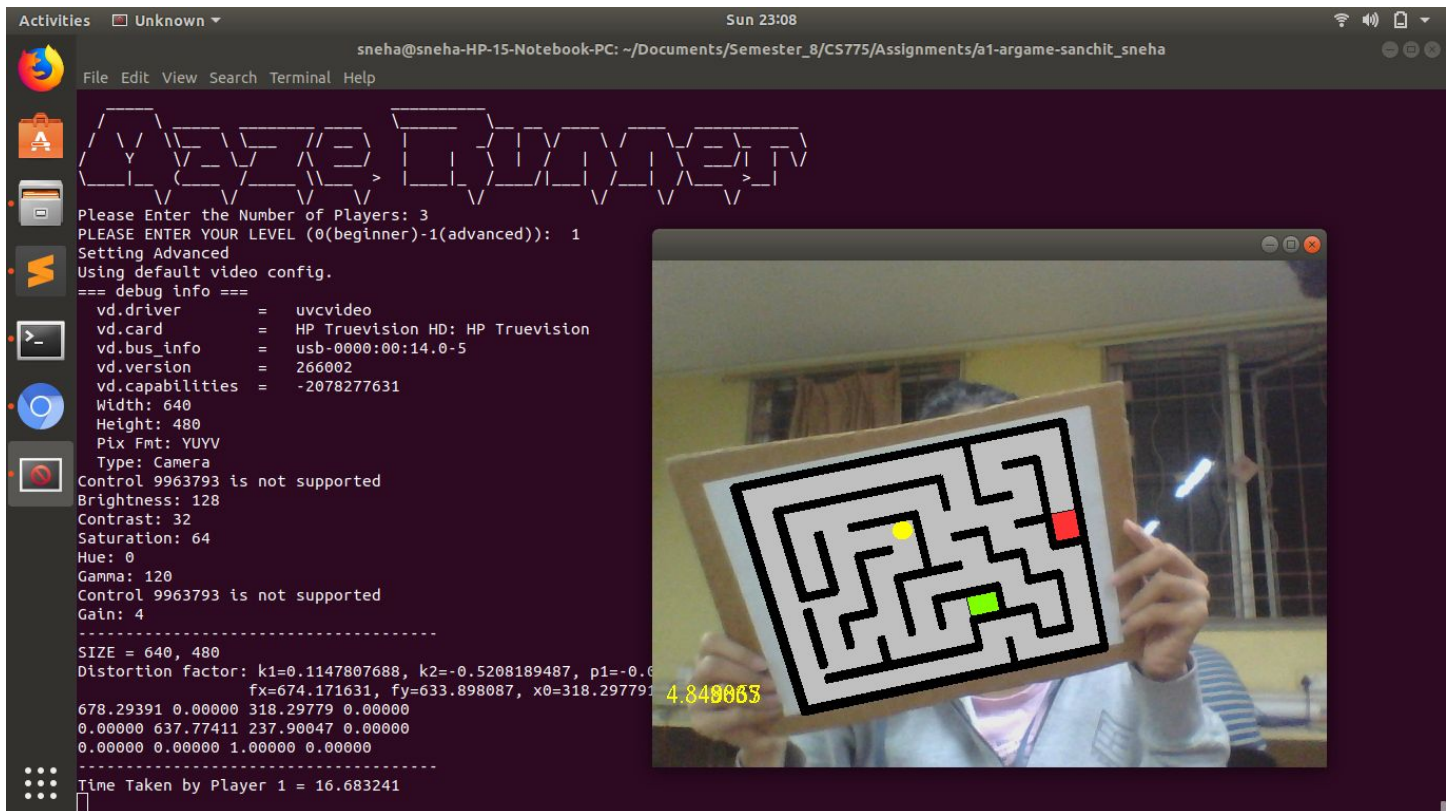
Actions:
Press 'r' to start the game for each player
Press escape to exit the game in between
Forward tilt -> movement in downward direction
Backward tilt -> movement in upward direction
Right tilt -> movement in right direction
Left tilt -> movement in left direction

Level: Beginner, Advanced. The end position is different for these two levels. The default level is beginner.

Goal: The game finishes when the ball reaches the end position (red).

Score: The time taken to move the ball from the start (green) to the end (red). It is displayed in realtime at the bottom left corner.

How to play:
1. Enter the number of players
2. Select the level of the game
3. Press 'r' to start session for a player
4. The real time score (which is also real time "BAZINGA") will be displayed at the bottom left corner
5. If a player wants to skip the session go to step 3
6. Reach the end (red) by tilting the marker, the final score will be printed to the terminal
7. Go to step 3 for the next player's session
8. After every player's session the winner is printed to the terminal.

Implementation Details

The simplelite example of ARToolKit is used as base code. The maze is constructed using rectangle class. The ball is made using circle class.

Rectangle Class: The constructor requires dimensions like length, breadth and translation parameters wrt to center of marker. It renders a cuboid according to these parameters. Creating a corresponding rectangle in Box2D is indicated by a boolean flag(int actually). For example, the base, the start and the end cuboids are absent in Box2D. They are rendered for better user experience. On the other hand, all the maze walls are created in physics world and hence collisions with ball is possible.

Circle Class: The constructor requires radius and translation parameters. It renders a sphere and creates a corresponding circle in Box2D according to these parameters.

Main Loop: For each frame, the orientation of marker is calculated and the world gravity is updated. The world is updated due to new gravity. The ball's position is checked if it has reached the end state. If yes, the score is recorded. If the current player is the last player the game ends.