

Final Project Proposal - The RAGE MAZE

Calvin Tran

The game will be a single-player game. The player will control a square which moves around a maze. The maze will have several moving obstacles that the player cannot touch. The goal of the player is to reach a certain area in the least amount of deaths. There will be several levels in the game, each one with increasing difficulty.

This program will incorporate many different elements learned in the course. Selection will be used to choose each level of the game. In addition, selection is used to check if the player has collided with a wall, obstacle, checkpoint or end point. While loops will be used to run each level, as the game runs while a certain variable is True. Each level will be stored in a procedure that will be called when a level is selected. Other procedures will be used for various purposes such as for the introduction page, the instructions page and the pause page. Graphics are used by drawing shapes and moving them. An integer variable will be used to keep track of the player's number of deaths in a level to see if they can unlock a 'star' which will be received if they complete a level with a certain number of deaths or lower. String variables will be used to store the map of each level.

I estimate the program to take about 1000 lines of code. I will need a separate class for the player, the checkpoints, the walls of the maze and the obstacles as well as a function for each level. Pygame will be used to create the game. Shapes will be used to draw in the map, the player, the checkpoint, the end point and the obstacles.

Pseudocode:

def Level 1():

- Draw map 1 using rectangles
- Draw in a checkpoint
- DeathCountOne=0
- Draw in moving squares that move at a constant rate
- Draw in player as a square on 'starting pad'
- While player presses 'up':
 - Move player up x units
- While player presses "down":
 - Move player down x units
- While player presses "left":
 - Move player left x units
- While player presses "right":
 - Move player right x units
- If player touches checkpoint:
 - Make subsequent respawns at checkpoint
 - Draw in ending square
- If player collides with a moving obstacle:
 - DeathCountOne+=1
 - Respawn player at last checkpoint

If player touches ending pad and checkpoint has been touched:

 If DeathCountOne <= score required to get star:

 levelOneStar = True

 Return back to level selection page

If the key 'p' is pressed:

 Display pause screen

def Level 2()

 Draw map 2 using rectangles

 Draw in a checkpoint

 DeathCountTwo=0

 Draw in moving squares that move at a constant rate

 Draw in 'ending' platform

 Draw in player as a square on 'starting pad'

 While player presses 'up':

 Move player up x units

 While player presses "down":

 Move player down x units

 While player presses "left":

 Move player left x units

 While player presses "right":

 Move player right x units

 If player touches checkpoint:

 Make subsequent respawns at checkpoint

 Draw in ending square

 If player collides with a moving obstacle:

 DeathCountTwo+=1

 Respawn player at last checkpoint

 If player touches ending pad and checkpoint has been touched:

 If DeathCountTwo <= score required to get star:

 levelTwoStar = True

 Return back to level selection page

 If the key 'p' is pressed:

 Display pause screen

def Level 3():

 Draw map 3 using rectangles

 Draw in a checkpoint

 DeathCountThree=0

 Draw in moving squares that move at a constant rate

 Draw in 'ending' platform

 Draw in player as a square on 'starting pad'

 While player presses 'up':

 Move player up x units

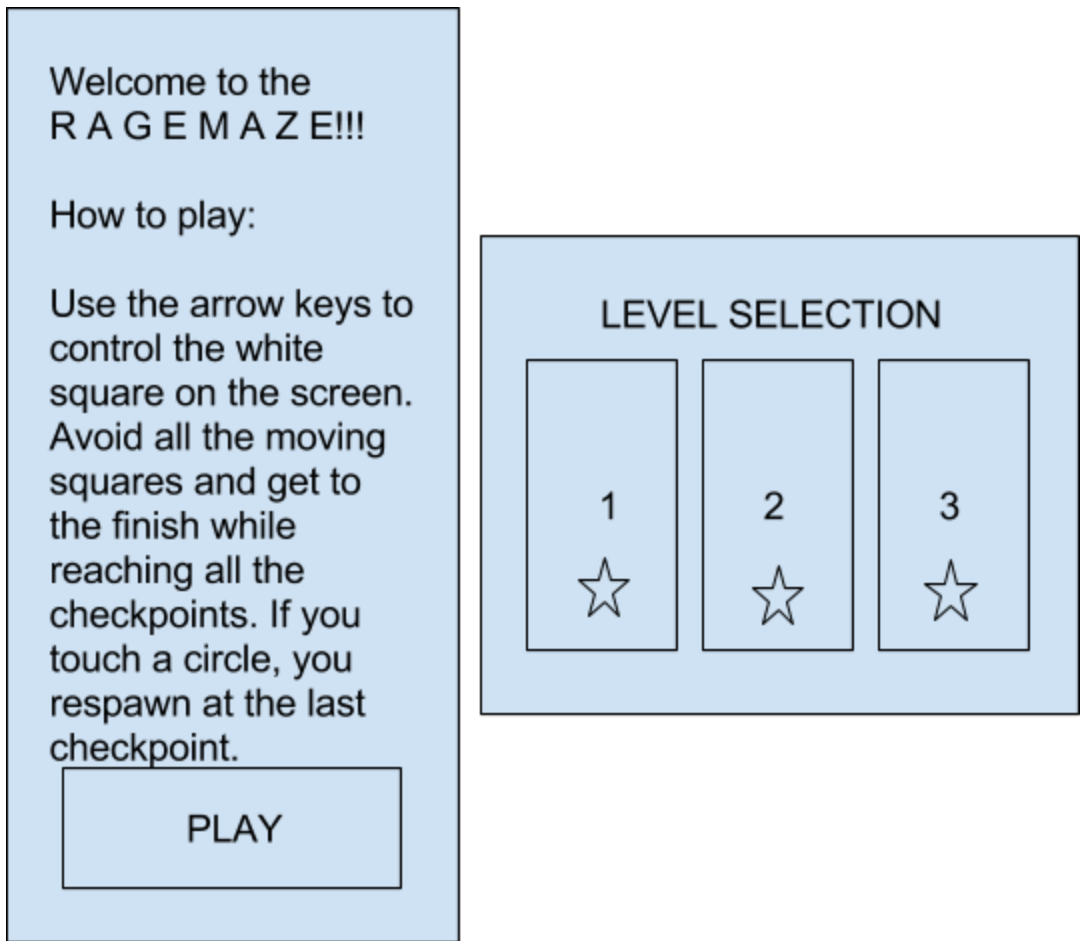
```
While player presses "down":
    Move player down x units
While player presses "left":
    Move player left x units
While player presses "right":
    Move player right x units
If player touches checkpoint:
    Make subsequent respawns at checkpoint
    Draw in ending square
If player collides with a moving obstacle:
    DeathCountThree+=1
    Respawn player at last checkpoint
If player touches ending pad and checkpoint has been touched:
    If DeathCountThree <= score required to get star:
        levelThreeStar = True
    Return back to level selection page
If the key 'p' is pressed:
    Display pause screen
```

Show starting page with instructions

If play button is clicked:

```
Undraw starting page
Draw level selection page, with three buttons for each level
Display grey stars on each level
If levelOneStar == True:
    Draw in yellow star to cover grey star for level 1
If levelTwoStar == True:
    Draw in yellow star to cover grey star for level 2
If levelThreeStar == True:
    Draw in yellow star to cover grey star for level 3
If Level 1 is clicked:
    Call Level 1()
If Level 2 is clicked:
    Call Level 2()
If Level 3 is clicked:
    Call Level 3()
```

Sketchboard:



SAMPLE LEVEL LAYOUT:

