

**Subject: 15-112 Term Project Proposal**

**From: Richelle Guice**

## **Project: Tomb of the Mask**

### **Description**

In this game, the user will have to navigate through a randomized maze. However, while traversing there will be obstacles the player must be aware of and be able to pass.

Obstacles include spikes, missiles, and explosions. As the player is traversing, there will be coins and stars to collect to up their score. The objective of the game is to reach an exit door.

### **Competitive Analysis**

This game is similar to some other projects I have seen online. For example, Adhvik's Primerunner game. He also had what seemed like a randomized maze where a player navigates through and picks up things and also avoids obstacles. Furthermore, his project also implemented side-scrolling which will be a big component in my project.

Another example that I have seen which I feel is a little similar to mine, is Anesha's 'Survival of the Fittest' game. Every time the player plays, there is also a randomized maze that the user must traverse through. Here, the user has to pick up survival items to keep their health up or else they die. While a completely different function, it seems to be a very similar implication to mine where I have to put obstacles the user must avoid.

### **Structural Plan**

One function will be the appStarted that sets all the base values. One main function for generating the randomized maze. Another main function for sidescrolling with the player. For allowing the player to actually move, there will be multiple functions for: moving the playerPiece, seeing if the move is valid, and then drawing where the player currently is. There is obviously a keyPressed function for allowing the player to move as they wish. There will be another function for generating obstacles and another function for obstacle collision.

### **Algorithmic Plan**

The first trickiest part of my game is implementing a randomized maze that will account for obstacles and allow for my character to move through in a timely manner. This is not just a game where the object is to be able to simply navigate through a randomized maze and find an exit. Instead, it must be playable and fun and also have an "action" feel. The user moves in a "hard" direction. For example, if pressed down, it moves immediately to the nearest wall that is eligible. To tackle this problem I first set up the board as rows and cols. For now, I set a baseline number of columns (20) just for easier debugging purposes.

1. Board is set up as rows and cols
  2. Randomize by how many rows there will be a row of wall
    - a. Randomized a number between (2, 3)
    - b. For every row, there has to be at least one cell space to allow the character to move thus randomize an exitSpace(1, 3)
    - c. Randomize by how many cols the row of wall will reach while accounting for space for the character to move
  3. Right side of wall is just from start to randomized length of col
  4. Left side of wall is ^ + randomized exitSpace
  5. Issue of allowing player to be able to bounce off of walls: Using an imaginary player, set its column (iCol) always at the middle of board to play through the randomized maze of rows then to draw additional vertical walls
  6. Figure out where there is gap: wallCol → start of gap, gapSpace → app.exitSpace
  7. Three cases:
    - a. If iCol on gap → nothing, imaginary player can fall through
    - b. If gap is on the left of imaginary player → add block above (row-1) the left wall
    - c. if gap is on right of imaginary player → add block above right wall ONLY if in bounds
  8. At end row+=1 to keep iterating through entirety of board
- Second trickiest part of my game is implementing sidescrolling. Have not gotten there yet tho. !!


## Timeline Plan

Feature:	Date:
Randomized maze	4-26-2021
Character on-screen and movable	4-26-2021
Obstacles and obstacle collision	5-01-2021
Coins and stars and show score	5-01-2021
On-screen design improvement	5-01-2021
Sidescrolling	5-01-2021

## Version Control Plan

I copy all my files to google drive after every change.

## Folders


 TERM PROJECT

## Files

Hi Rubeen,  
Your MP1 is currently creating tons of the black cells. To prep for TP0, you should look into making the interesting cells that interact on meeting and deciding on an algorithm to randomly generate a maze. If you would rather handcode a maze, let me know so we can figure out a different complete component for your project.

Randomizing  
Randomly generate maze  
Randomize appearance/properties  
Custom appearance for maze  
Objectives to get to end state


**Randomized depth-first search**



Randomize appearance using both for search

The algorithm, also known as the "recursive backtracker" algorithm, is a randomized version of the depth-first search algorithm.

I frequently implement with a stack. This approach is one of the simplest ways to generate a maze using a computer. Consider the space for a maze being a large grid of cells. One simple (naïve) search will explore each cell until it reaches a random cell. One can continue this search as

 TP0

**Subject: 15-112 Term Project Proposal**  
**From: Rubeen Datta**

**Name**  
Rubeen Datta

**Description**  
In this game, the user will have to navigate through a randomized maze. However, when searching there will be obstacles that the player must be aware of and be able to pass. Obstacles include spikes, mines, and captives. As the player is searching, there will be notes and clues to collect along their path. The objective of the game is to reach an exit door.

**Conceptual Analysis**  
This game is similar to some other projects I have seen online. For example, Advent's Dungeon game, the user must search for a randomized maze where a player navigates through and picks up items and also avoids obstacles. Furthermore, the project also implemented side-scrolling which will be a key component in my project.

Another example that I have seen which I find is pretty similar to mine, is *Advent's Dungeon* of the *15-112* game. Every time the player plays, there is also a randomized maze that the user must navigate through. Here, the user has to pick up items/clues to help them reach up or near their exit. After a completely different function, it seems to be a very similar experience to mine where I have to get obstacles that user must avoid.

**Algorithm Plan**  
**Outline**  
**Algorithmic Plan**  
The first random part of my game is implementing a randomized maze that will account for obstacles and allow for my computer to move through in a timely manner. This is not a game where the user is to be able to simply navigate through a randomized maze and that's all. Instead, it must be created and fun and also have an "adventure" feel. To tackle this problem, I'll set up the board as rows and cols. For now, I set a baseline number of columns and for rows

 TP1 Design Proposal

## Module List

- I do not think I need external modules/hardware/technologies as of now

## Storyboard

