Justin Shi

**Procedurally Generated Multi-Level Maze/Puzzle Game with Horror/Survival Elements**

Advisor: Holly Rushmeier

**Introduction:**

For this project, I aim to create a procedurally generated maze game with multiple levels; that is, multiple procedurally generated mazes will exist at any time, drawing inspiration from science fiction notions of multiple realities. The mazes will include procedurally generated puzzles and riddles that cannot be solved in any one level alone, but require information or solutions found in different levels, which the player will be able to move between in some fashion. Thus, to progress, the player must be able to keep track of multiple levels and puzzles at the same time. I also aim to develop a method for ensuring that at any state of the game, some maze-level will be solvable; there should never be a point where the procedurally generated puzzles completely block each other. Finally, some elements typical of horror and survival games will be added, including setting, ambience, sound effects, and an enemy character with a simple player-seeking AI, in an attempt to shake the focus of the player trying to solve the puzzles.

**Deliverables:**

The main deliverable for this project will be the packaged executable for a three-dimensional maze game created in the Unity game engine. The game will be played from the first-person perspective, with movement and actions performed with the keyboard and camera-panning performed with the mouse. Additional deliverables will be the game's source code and any assets created and used in the game, such as graphics or audio. If time allows, entries in a blog detailing the development process and thoughts behind certain design ideas will also accompany the aforementioned details.

**Ideas:**

**Maze:** The primary idea is to implement an algorithm that will randomly generate an undirected graph that will represent the layout of the maze, with some node being selected as the player's starting
position and another node representing the goal location the player must get to. The algorithm must be
tuned such that the graph is of an acceptable complexity, such that the start and goal nodes are not too
close to each other and the maze contains enough twists and turns to make solving it not trivial. One
option that might be taken is to make the maze graphs trees, such that there is only one path between the
start and goal nodes; however, a graph with cycles may introduce desired complexity.

**Puzzles:** The next component is to introduce roadblocks within the mazes that block the player
from progressing to the goal nodes; these roadblocks can only be removed by performing actions in
different levels, which the player will be able to move between in some fashion. For example, there may
exist at some point 5 different mazes at the same time, with the 1st requiring a switch flipped in the 5th
and the 4th requiring a switch flipped in the 3rd. We can possibly increase the complexity and
intertwinedness of these mazes by chaining these requirements: perhaps the switch on the 2nd level to
open a path to a solution on the 1st level can only be reached upon hitting a switch to the 4th level. The
core behind this will be an algorithm that analyzes all live mazes and ensures that, at any given moment,
there is some path of actions for the player to ultimately reach the goal node of some puzzle. Beyond this,
alternative forms of puzzles beyond switches may deserve some time for thought.

**Threat:** Finally, I envision the game taking place within a certain dark atmosphere, with a limited
draw distance and enemy entities that are slowly introduced. The AI for these enemy characters will be a
simple patrol routine, where they randomly move between nodes in the maze, until the player moves
close enough to them, in which case they will begin searching for the player. The player will be given no
tools to combat the enemy; instead, the enemy must be avoided, or run away from if encountered. The
sense of defenselessness and dark setting (sounds, textures, enemy appearance) are intended to create
something of a distraction for the player from the puzzles, such that they will have to keep the maze and
the enemy in mind at the same time.

**Timeline:**

The first step in this project will be setting up a general 3-dimensional environment in Unity,
similar to one the game will occur in, with a character the player controls from the first-person perspective
using the mouse and keyboard, as well as designing the algorithm for random graph generation, which I aim to complete within 2 to 3 weeks.

Next, the graphs generated by this algorithm must be represented in Unity and used to populate the environment with corridors that will form the mazes. This process will have to ensure that nodes and edges, when translated to corridors, are properly spaced out, and code must be added to ensure proper interaction between the character and the walls (no clipping or physics errors). At this point, the player will spawn in the start node-area and the level will complete upon reaching the goal node-area. I hope to have this maze setup complete by the end of October.

The next component will be keeping track of the existence of multiple mazes at the same time, allowing the player to move between them, and keeping track of the player's position in each maze. Here, we will implement an algorithm that checks live mazes and places roadblocks and switches across them in a way that ensures progression is always possible, using some form of path-finding algorithm; I aim to complete this within another 2 to 3 weeks.

The final components, which I prioritize lowest and will only move to if time permitting, will be the development of an enemy character and a simple player-hunting AI, and extra cosmetic components, such as more detailed textures for walls, limited draw distances/light, and sound/music.