**Project Proposal**

**Project Description: 3D Maze game**

The project is a maze, except for the fact that you can also traverse up and down while you are in the maze. The player will be able to navigate through the maze by looking at the cross-sectional view from where the person is in the maze currently.

**Competitive Analysis**

On the internet, there were minimal projects about a playable 3d maze where you can traverse up and down. Most of it was just a maze in a 3d first person view. There was some algorithms and examples for generating a 3d maze on stack overflow, but there wasn’t any playable 3d maze online.

**Structural Plan**

The project will be separated into many different files, the main file that runs the code, the block class file for different types of blocks, the maze generation file with the maze object and the 3d maze object that inherit from it. There will be a file for all the windows, such as the 2d maze, the 3d maze, the help window, the impossible mode window, and the main window.

**Algorithmic Plan**

The most complex section of the project is to generate the 3d maze. I will use Eller’s algorithm as it is the fastest maze generating algorithm. I will start by first generating a 2D maze with the Eller’s algorithm. First, I will write in comments of all the major steps that I will need to take to generate the maze, then I will implement step as test it as I implement it. After being able to generate the 2D maze, I will modify some code so that it can also generate a 3D maze as well. I will make sure the mazes that was generated works by starting by generating a really small maze, like a 2 by 2 by 2 maze, print it out by printing each layer with the print2dList function, and l walk through it in my mind to make sure that it is possible to reach the finish position from the starting position.

**Timeline Plan**

11/12: Research maze generating algorithms

11/13: TP0

11/14: Able to generate 2D maze using Eller’s

11/16: Finish generating 3D maze and the graphics to show a playable 2D maze

11/17: Show a playable 3D maze and reach MVP

11/18: TP1

11/20: Finish Main page and all pages in storyboard

11/22: Draw 3D cube

11/23: TP2

11/23: Include fun animations to the game

11/26: Draw 3D maze

11/27: Write instructions/help/functions for the game

11/29: Final touches to the design and code

11/30: Create the video

12/01: TP2

**Version Control Plan**

I will back up my code using GitHub desktop, and I will store each version before I make any major changes. The code will also be uploaded to GitHub.

Graphical user interface

Description automatically generated with medium confidence

Showing where the code is stored on the computer. I will also save all the code from each TP0 and after any major successes

Graphical user interface, text, application, email

Description automatically generated

Showing all the past changes in the GitHub desktop app

Graphical user interface, application

Description automatically generated

Making commits of the files

A screenshot of a computer

Description automatically generated

Image of the code uploaded to GitHub. The code will be updated at the end of the day if I made any changes.

**Module List**

I will not use any modules until I reach MVP

**TP2 update**

As I finished the 3D maze generating algorithm earlier than what I expected, so I decided to add a few extra functions to the maze game, such and being able to change the size of the maze that was generated and toggle different switches such hide the walls or only able to view part of the make to make the game harder. For this, I created a myButton class with the CommandBar class that inherits from it in the ButtonClass file.