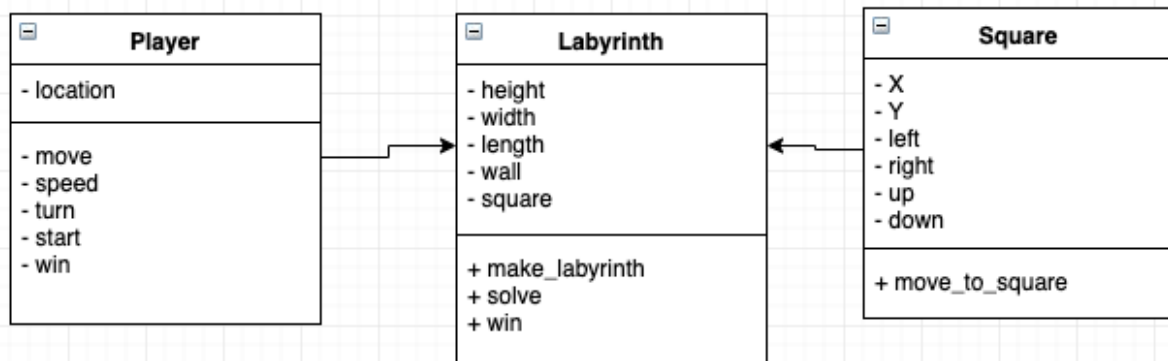# Labyrinth – Project plan

Since I have not completed round 5 or 6 yet in this course, I don't really know yet exactly how to create this program or how exactly I will make the game. So I will leave some of the questions empty since, I have not yet learned the information needed to answer the questions.

1. Matilda Stendahl, 1032747, Bioinformation technology, 2:nd year student, 23.2.2023
2. Creating a program that makes and draws a labyrinth. The player is a mouse, who is placed in the middle of the labyrinth and their goal is to guide the mouse out of the labyrinth. The labyrinth is in 3D and always has a way out. To make my game unique, I will possibly include monsters in the game so that the player has to find their way out as soon as possible, or make a timer in the game so the player can try to beat themselves by getting a better time each time. Difficulty level: Medium

3. In the program window the user will be able to see the maze in 3D, and possibly a "give up" button or a text that tells them what key to press to give up. The user does not write anything in the program, they only use their keyboard to move the mouse and the mousepad to look around. The program possibly prints something when the game starts and when the user wins the game.

   Real case: The user spawns in the middle of the labyrinth. They can after that look around and move around with the arrow keys on the keyboard to try to find the way out. If successful, the user finds the way out and wins the game, but if not, they can choose to give up and after that the game will lead the mouse out of the game by itself.

4. This is a bit unclear still, but my plan now is to have a class for the player, and functions in the class that for example make the player move and look around. In addition to this, I will implement a class for the labyrinth and for the squares.

Player

- location

- move
- speed
- turn
- start
- win

Labyrinth

- height
- width
- length
- wall
- square

+ make_labyrinth
+ solve
+ win

Square

- X
- Y
- left
- right
- up
- down

+ move_to_square

5. Probably lists and arrays.

6. I don't think I will need files.

7. I need algorithms to make my maze and solve the labyrinth. I will use some maze-solving algorithm, I may use the Prim's randomized algorithm, the Depth-first search algorithm, Dijkstra's algorithm or Shortest Path Finder.

8. I should for example test that the labyrinth is made correctly and has a solution and only one solution. I should also test things that are not allowed, for example going through a wall, and that you can not do them in the game. I can also test that when the player moves, it is in the correct square that it should have moved to, and that if the player solves the labyrinth, the player wins the game and that is they do not solve it, the game solves itself.

9. I will maybe use PyQt, but other than that, I am not sure yet what I need or if I need any.

10. I think that before the first checkpoint, I will work about 10h/week to get the base of the game done well. After the first checkpoint, I believe to have done the most work of the project, and will maybe use 5-10h/week to test is and finalize it.

11. Here are some sources I plan to use. I will add more if I find more.

   https://inventwithpython.com/bigbookpython/project45.html
   https://www.youtube.com/watch?v=5xyeWBxmqzc

https://medium.com/swlh/fun-with-python-1-maze-generator-931639b4fb7e

https://github.com/xFrednet/3D-Packman

https://stackoverflow.com/questions/39355587/speeding-up-dijkstras-algorithm-to-solve-a-3d-maze