[redacted]

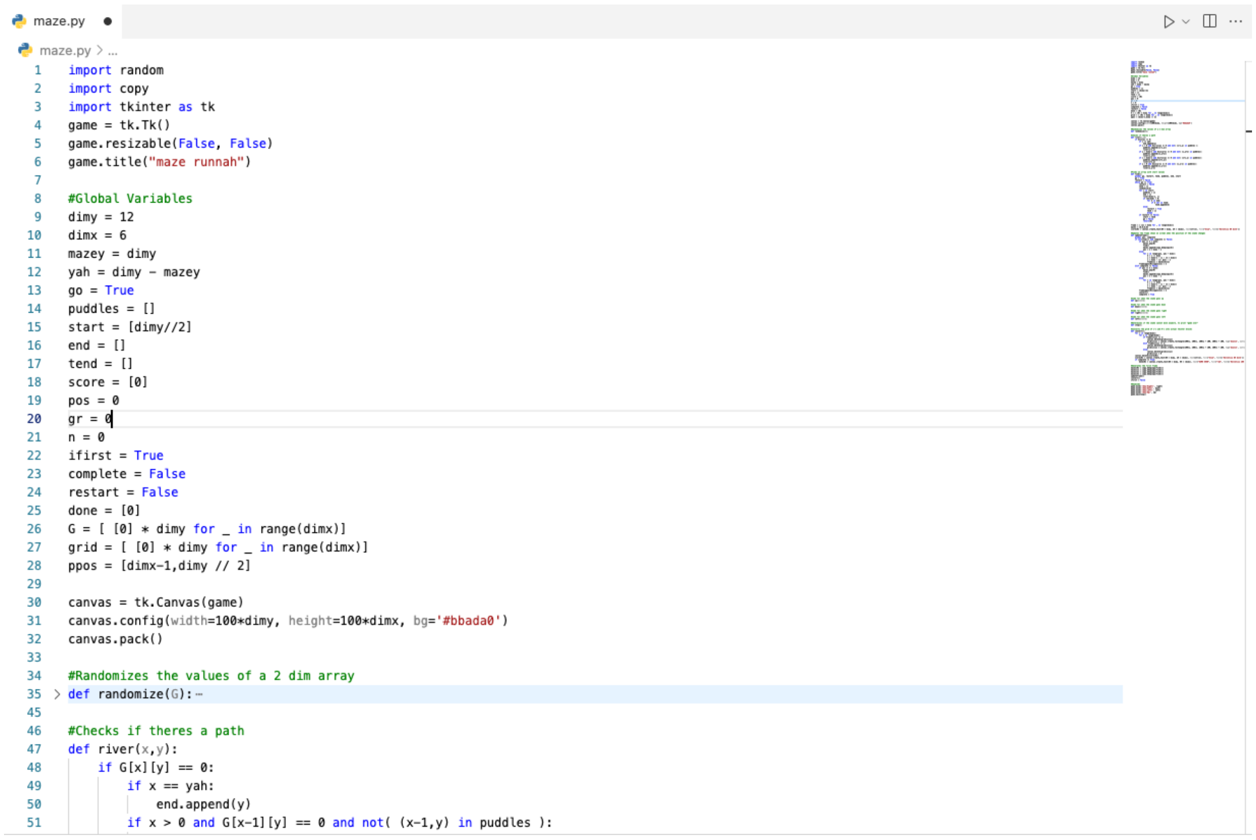
Video:
https://drive.google.com/file/d/1rK472mEc-bGvkZ8zSXiJazXIGtzMsVVk/view?usp=sharing

Purpose This is a maze snake game, where you play as a green snake traversing a maze. This maze is randomly and infinitely generated (this was the hardest part ofthe project) as you go up, and although it's random it's programmed so that there is always a path for the snake to go up (never impossible). The score is centered on the top and increases every time the snake moves up. If the snake cannot move anywhere, a “Game Over” screen pops up.

Strings: In the maze game, we display a score at the top, showing how many times the snake has gone up. We use the n variable to display the score, which goes up by 1 every time the snake goes up. To display it using tkinter, we have to convert n into a **string** using the **str** function.

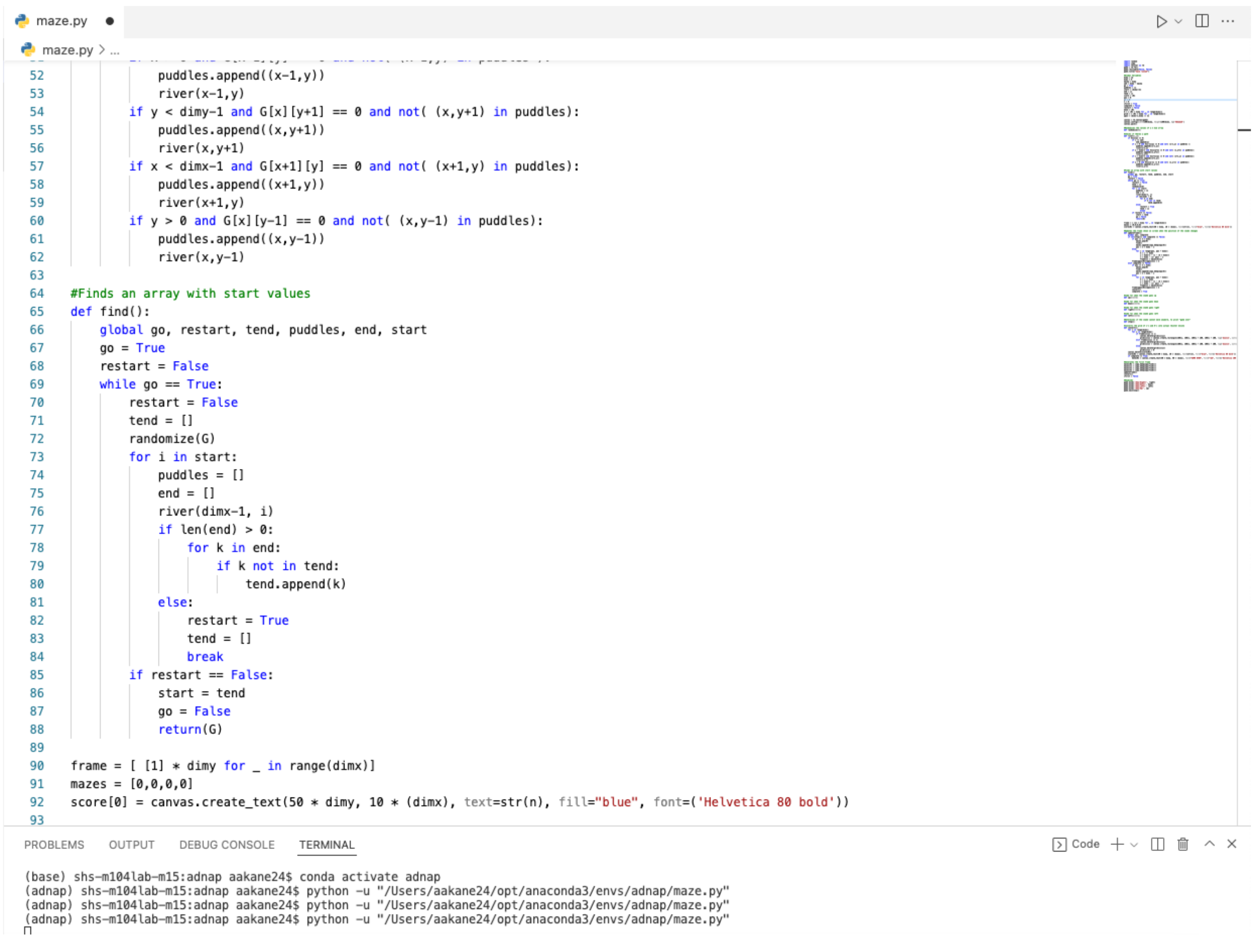
Events: The program can handle keyboard events such as up, right, down, and left key. For each of these events there is an event handler which makes the snake go in the respective direction.

Functions: Many functions are used, the most notable one being the *river* function. When generating the infinite maze, we first use the randomize function to generate a random array of 1’s and 0’s (1 represents a barrier, 0 an open space). The river function checks if there's a path from up to down for the snake to go through. In the find function, lots of possible arrays are generated, and the river function is used to check for a path. We have to check for a path many times since we go through lots of possible arrays, which is why the river function is so useful.


maze.py

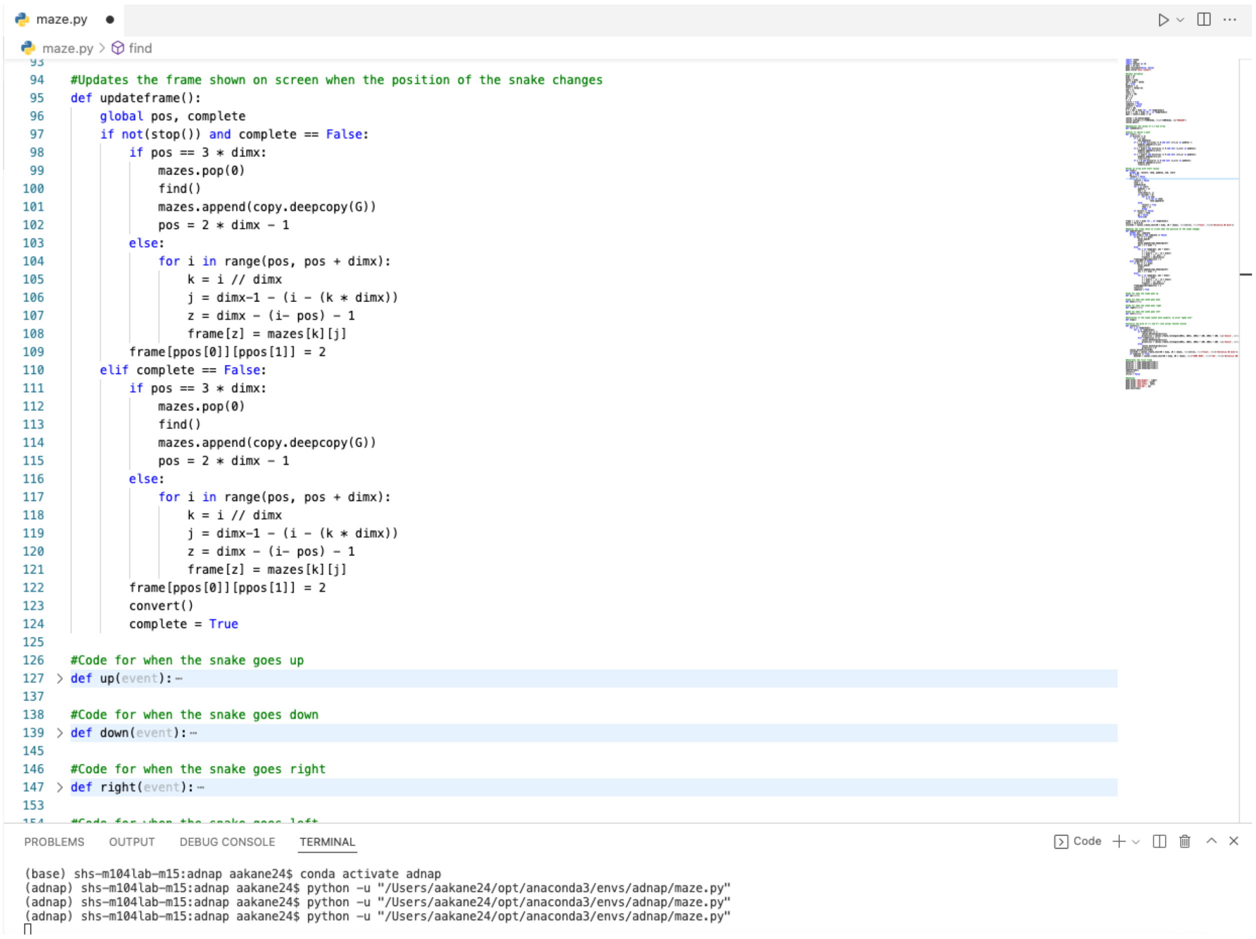
maze.py > ...

```python
import random
import copy
import tkinter as tk
game = tk.Tk()
game.resizable(False, False)
game.title("maze runnah")

#Global Variables
dimy = 12
dimx = 6
mazey = dimy
yah = dimy - mazey
go = True
puddles = []
start = [dimy//2]
end = []
tend = []
score = [0]
pos = 0
gr = 0
n = 0
ifirst = True
complete = False
restart = False
done = [0]
G = [ [0] * dimy for _ in range(dimx)]
grid = [ [0] * dimy for _ in range(dimx)]
ppos = [dimx-1,dimy // 2]

canvas = tk.Canvas(game)
canvas.config(width=100*dimy, height=100*dimx, bg='#bbada0')
canvas.pack()

#Randomizes the values of a 2 dim array
def randomize(G):...

#Checks if theres a path
def river(x,y):
    if G[x][y] == 0:
        if x == yah:
            end.append(y)
        if x > 0 and G[x-1][y] == 0 and not( (x-1,y) in puddles ):
```



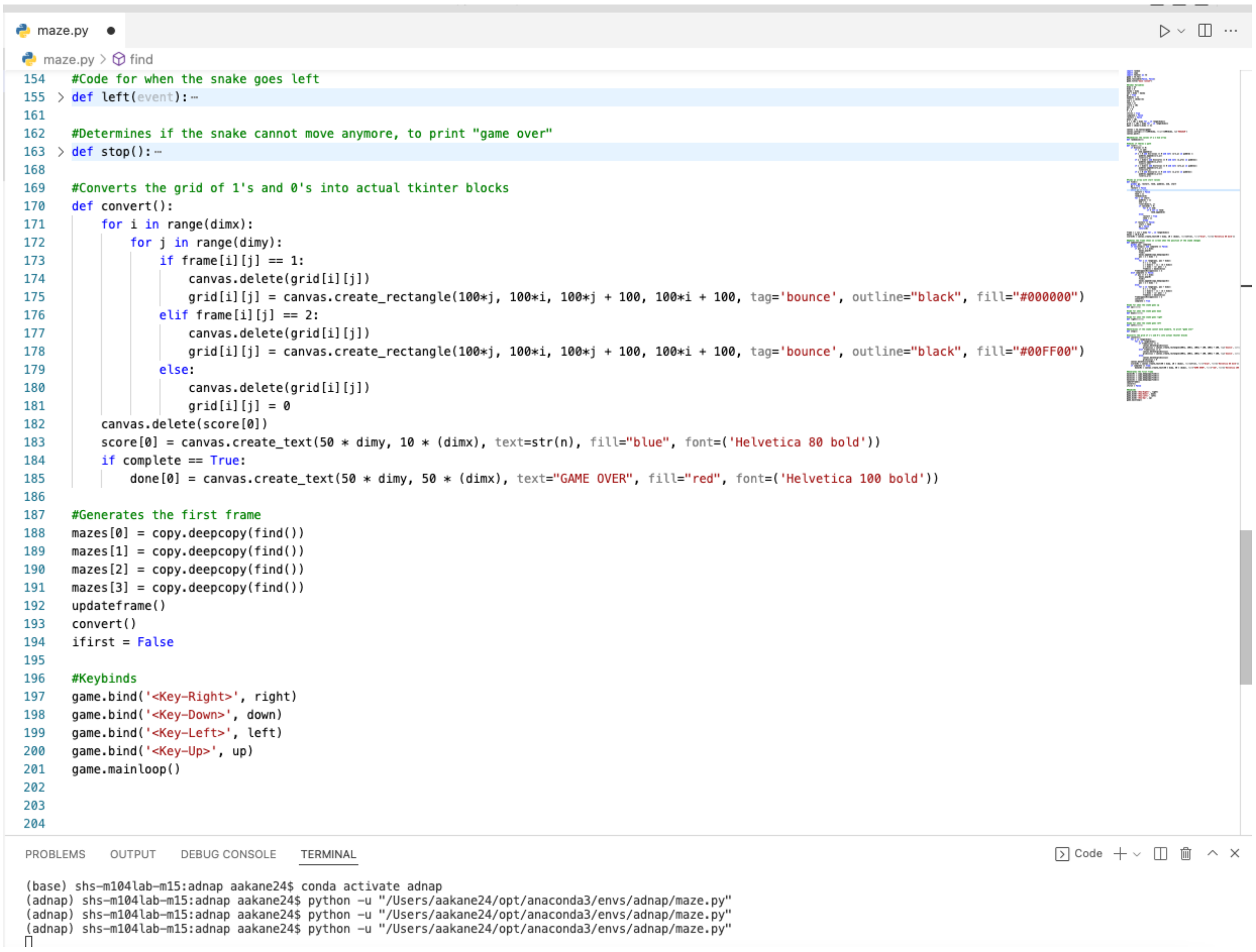
```python
            puddles.append((x-1,y))
            river(x-1,y)
        if y < dimy-1 and G[x][y+1] == 0 and not( (x,y+1) in puddles):
            puddles.append((x,y+1))
            river(x,y+1)
        if x < dimx-1 and G[x+1][y] == 0 and not( (x+1,y) in puddles):
            puddles.append((x+1,y))
            river(x+1,y)
        if y > 0 and G[x][y-1] == 0 and not( (x,y-1) in puddles):
            puddles.append((x,y-1))
            river(x,y-1)

#Finds an array with start values
def find():
    global go, restart, tend, puddles, end, start
    go = True
    restart = False
    while go == True:
        restart = False
        tend = []
        randomize(G)
        for i in start:
            puddles = []
            end = []
            river(dimx-1, i)
            if len(end) > 0:
                for k in end:
                    if k not in tend:
                        tend.append(k)
            else:
                restart = True
                tend = []
                break
        if restart == False:
            start = tend
            go = False
            return(G)

frame = [ [1] * dimy for _ in range(dimx)]
mazes = [0,0,0,0]
score[0] = canvas.create_text(50 * dimy, 10 * (dimx), text=str(n), fill="blue", font=('Helvetica 80 bold'))

```

```
(base) shs-m104lab-m15:adnap aakane24$ conda activate adnap
(adnap) shs-m104lab-m15:adnap aakane24$ python -u "/Users/aakane24/opt/anaconda3/envs/adnap/maze.py"
(adnap) shs-m104lab-m15:adnap aakane24$ python -u "/Users/aakane24/opt/anaconda3/envs/adnap/maze.py"
(adnap) shs-m104lab-m15:adnap aakane24$ python -u "/Users/aakane24/opt/anaconda3/envs/adnap/maze.py"
```



```python
#Updates the frame shown on screen when the position of the snake changes
def updateframe():
    global pos, complete
    if not(stop()) and complete == False:
        if pos == 3 * dimx:
            mazes.pop(0)
            find()
            mazes.append(copy.deepcopy(G))
            pos = 2 * dimx - 1
        else:
            for i in range(pos, pos + dimx):
                k = i // dimx
                j = dimx-1 - (i - (k * dimx))
                z = dimx - (i- pos) - 1
                frame[z] = mazes[k][j]
        frame[ppos[0]][ppos[1]] = 2
    elif complete == False:
        if pos == 3 * dimx:
            mazes.pop(0)
            find()
            mazes.append(copy.deepcopy(G))
            pos = 2 * dimx - 1
        else:
            for i in range(pos, pos + dimx):
                k = i // dimx
                j = dimx-1 - (i - (k * dimx))
                z = dimx - (i- pos) - 1
                frame[z] = mazes[k][j]
        frame[ppos[0]][ppos[1]] = 2
        convert()
        complete = True

#Code for when the snake goes up
def up(event):⋯

#Code for when the snake goes down
def down(event):⋯

#Code for when the snake goes right
def right(event):⋯

#Code for when the snake goes left
```

```
(base) shs-m104lab-m15:adnap aakane24$ conda activate adnap
(adnap) shs-m104lab-m15:adnap aakane24$ python -u "/Users/aakane24/opt/anaconda3/envs/adnap/maze.py"
(adnap) shs-m104lab-m15:adnap aakane24$ python -u "/Users/aakane24/opt/anaconda3/envs/adnap/maze.py"
(adnap) shs-m104lab-m15:adnap aakane24$ python -u "/Users/aakane24/opt/anaconda3/envs/adnap/maze.py"
```



```python
#Code for when the snake goes left
def left(event):⋯

#Determines if the snake cannot move anymore, to print "game over"
def stop():⋯

#Converts the grid of 1's and 0's into actual tkinter blocks
def convert():
    for i in range(dimx):
        for j in range(dimy):
            if frame[i][j] == 1:
                canvas.delete(grid[i][j])
                grid[i][j] = canvas.create_rectangle(100*j, 100*i, 100*j + 100, 100*i + 100, tag='bounce', outline="black", fill="#000000")
            elif frame[i][j] == 2:
                canvas.delete(grid[i][j])
                grid[i][j] = canvas.create_rectangle(100*j, 100*i, 100*j + 100, 100*i + 100, tag='bounce', outline="black", fill="#00FF00")
            else:
                canvas.delete(grid[i][j])
                grid[i][j] = 0
    canvas.delete(score[0])
    score[0] = canvas.create_text(50 * dimy, 10 * (dimx), text=str(n), fill="blue", font=('Helvetica 80 bold'))
    if complete == True:
        done[0] = canvas.create_text(50 * dimy, 50 * (dimx), text="GAME OVER", fill="red", font=('Helvetica 100 bold'))

#Generates the first frame
mazes[0] = copy.deepcopy(find())
mazes[1] = copy.deepcopy(find())
mazes[2] = copy.deepcopy(find())
mazes[3] = copy.deepcopy(find())
updateframe()
convert()
ifirst = False

#Keybinds
game.bind('<Key-Right>', right)
game.bind('<Key-Down>', down)
game.bind('<Key-Left>', left)
game.bind('<Key-Up>', up)
game.mainloop()
```

```
(base) shs-m104lab-m15:adnap aakane24$ conda activate adnap
(adnap) shs-m104lab-m15:adnap aakane24$ python -u "/Users/aakane24/opt/anaconda3/envs/adnap/maze.py"
(adnap) shs-m104lab-m15:adnap aakane24$ python -u "/Users/aakane24/opt/anaconda3/envs/adnap/maze.py"
(adnap) shs-m104lab-m15:adnap aakane24$ python -u "/Users/aakane24/opt/anaconda3/envs/adnap/maze.py"
```

Code: https://drive.google.com/file/d/1oI3Zoz6_IEdXIRESiiVhDsyj762Qm6yo/view?usp=sharing