PROJECT REPORT:

MAZE RUNNER

PROJECT DESCRIPTION:

Labyrinths are regularly basic riddles for people; however, they present an incredible programming issue that we can settle utilizing the most limited way methods like Dijkstra's algorithm, Breadth First Algorithm, and Depth First Algorithm. Our goal is to create the shortest path which starts in the white and does not cross into the black boundaries.

However, we used Breadth First Search (BFS) to build our game.

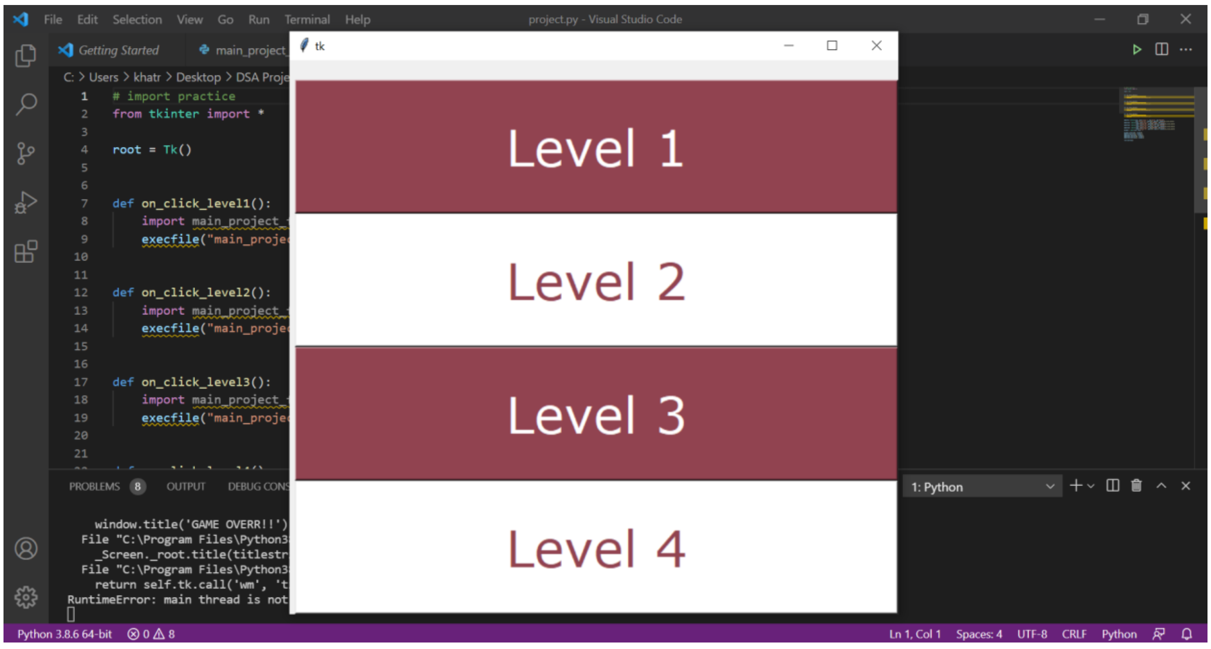
OBJECTIVES:

1. Implementation of DSA techniques using BFS and backtracking.
2. Using data structures like dictionaries, lists, tuples, and queues.

LIBRARIES USED:

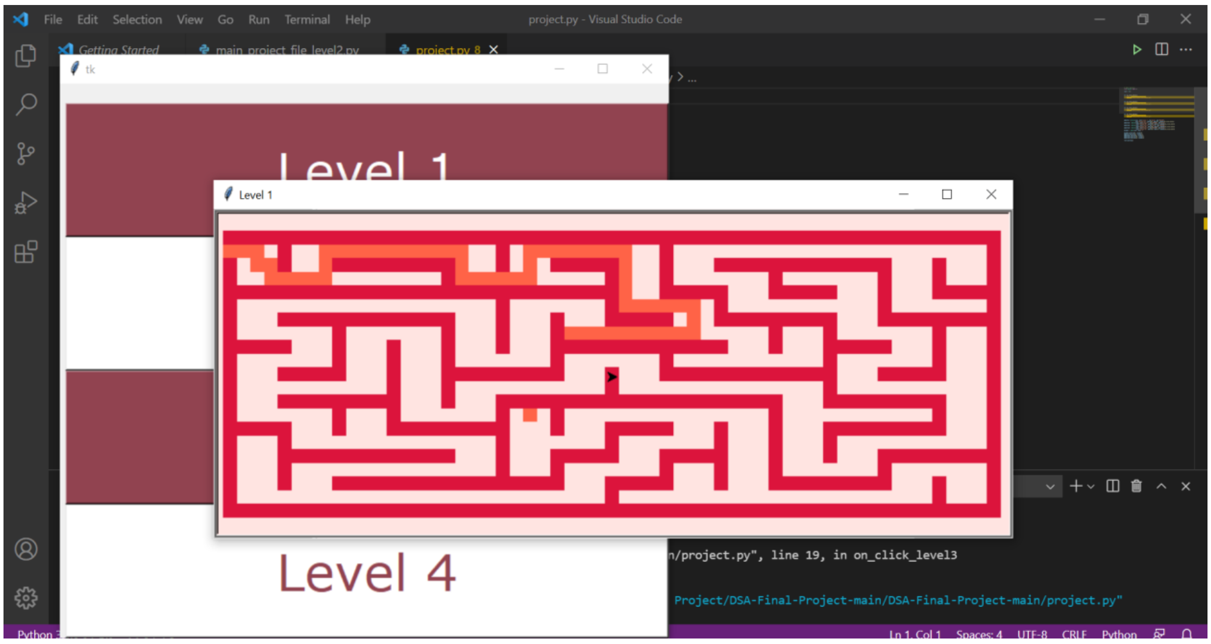
1. Tkinter
2. Turtle
3. Time
4. Threads

SCREENSHOTS OF OUTPUTS:

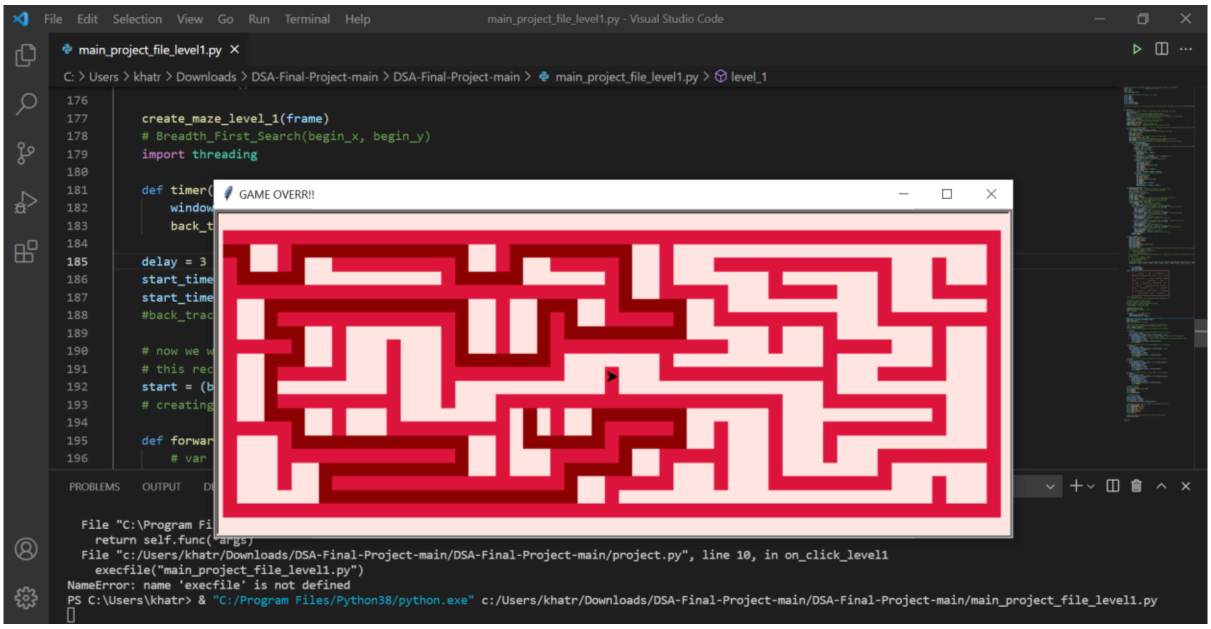


```
# import practice
from tkinter import *

root = Tk()


def on_click_level1():
    import main_project_
    execfile("main_proje


def on_click_level2():
    import main_project_
    execfile("main_proje


def on_click_level3():
    import main_project_
    execfile("main_proje
```

```
    window.title('GAME OVERR!!')
  File "C:\Program Files\Python3
    _Screen._root.title(titlestr
  File "C:\Program Files\Python3
    return self.tk.call('wm', 't
RuntimeError: main thread is not
```


File Edit Selection View Go Run Terminal Help
project.py - Visual Studio Code
Getting Started
main_project_file_level2.py
project.py 8
tk
Level 1
Level 1
Level 4
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python

```
n/project.py", line 19, in on_click_level3

Project/DSA-Final-Project-main/DSA-Final-Project-main/project.py"
```


File Edit Selection View Go Run Terminal Help
main_project_file_level1.py - Visual Studio Code
main_project_file_level1.py
C: > Users > khatr > Downloads > DSA-Final-Project-main > DSA-Final-Project-main > main_project_file_level1.py > level_1
GAME OVERR!!
PROBLEMS
OUTPUT

```
    create_maze_level_1(frame)
    # Breadth_First_Search(begin_x, begin_y)
    import threading

    def timer(
        window
        back_t

    delay = 3
    start_time
    start_time
    #back_trac

    # now we w
    # this rec
    start = (b
    # creating

    def forwar
        # var
```

```
  File "C:\Program Fi
    return self.func(*args)
  File "c:/Users/khatr/Downloads/DSA-Final-Project-main/DSA-Final-Project-main/project.py", line 10, in on_click_level1
    execfile("main_project_file_level1.py")
NameError: name 'execfile' is not defined
PS C:\Users\khatr> & "C:/Program Files/Python38/python.exe" c:/Users/khatr/Downloads/DSA-Final-Project-main/DSA-Final-Project-main/main_project_file_level1.py
```

REFERNCES:

1. https://www.makeschool.com/mediabook/oa/tutorials/trees-and-mazes/solving-the-maze/
2. https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/tutorial/
3. https://stackoverflow.com/questions/58688299/python-maze-bfs-shortest-path
4. https://www.codespeedy.com/call-a-function-after-some-interval-in-python/
5. https://zetcode.com/tkinter/snake/
6. https://www.codegrepper.com/code-examples/python/python+game+over+screen
7. https://www.includehelp.com/algorithms/backtracking-types-and-algorithms.aspx