Spike: 6
Title: Spike, Navigation and Graph

Author: Hoang Bao Phuc Chau, 103523966

Goals / deliverables:
Expand the Task 5 navigation graph simulation to demonstrate the following:
• A game world that is divided into a larger number of navigation tiles, and corresponding larger navigation graph
structure.
• A path-planning system that can create paths for agents, based on the current dynamic environment, using cost-
based heuristic algorithms that accounts for at least six types of 'terrain' (i.e. nodes with different costs).
• Demonstrate multiple independent moving agent characters (at least four) that are able to each follow their own
independent paths.
• Demonstrate at least two different types of agents that navigate the world differently

Technologies, Tools, and Resources used:
List of information needed by someone trying to reproduce this work

- Pycharm
- ChatGPT
- VSCode

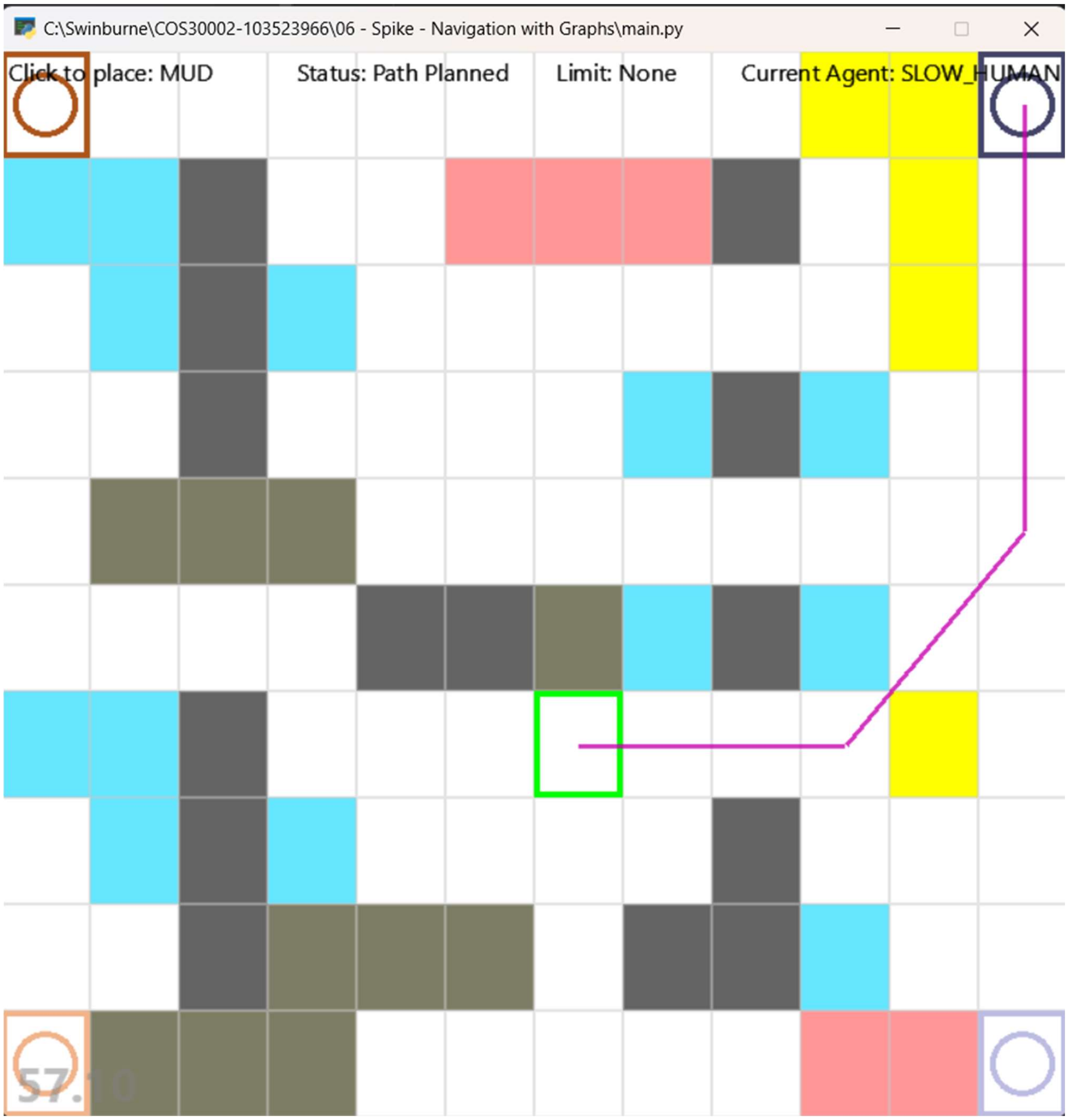
Tasks undertaken:

- In searches.py, delete all other search method except for Dikstra
- Define a class named Agent to represent an agent in the game. These following attributes are introduced:
  - ID and Name
  - Speed
  - Type
  - TraversalCost
  - Color
  - StartBox
  - Target Box
- In box_world.py, several modifications has been made:
  - Create 4 Agent objects, which are fastHuman, slowHuman, fastMonster, slowMonster. Where fastMonster looks for slowHuman and slowMonster looks for fastHuman
  - Introduce 2 more types of box, which are Swamp and Bridge
  - The navGraphs, paths, startMarkers, renderPaths and renderGraphs attributes are modified to used dictionaries, and agent ID is key.

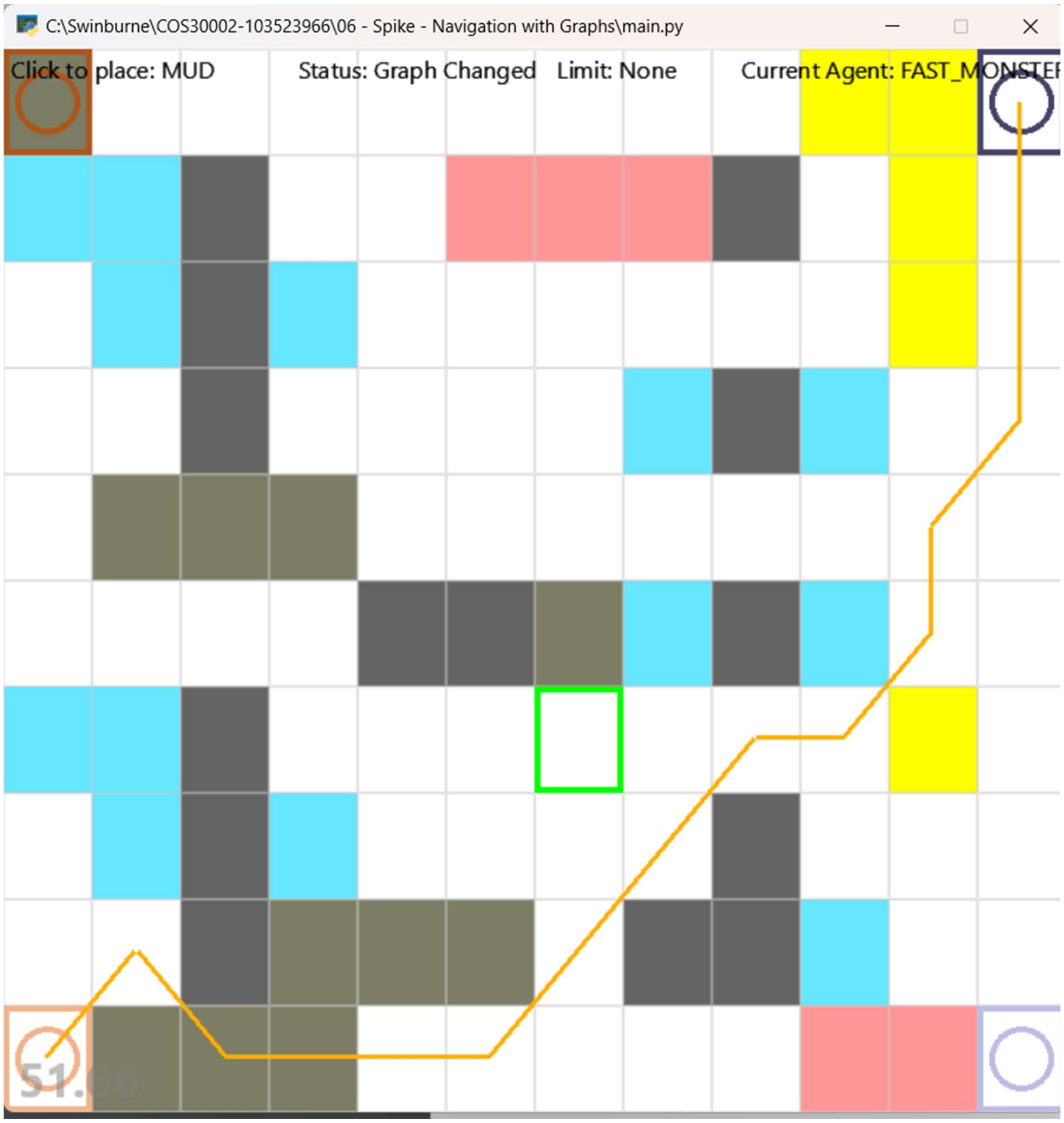
  - The methods of resetNavGraph is modified to reset the corresponding graph for each agent in the list
  - The planPath method has a new parameter of agent, which find the path for that agent individually
  - The setStart method also has a new paramater of agent, which set the startBox for that agent.
- In game.py, several modifications has been made:
  - Use only Dijkstra search method
  - Implement the mechanism where user and press Z, X, C, V to change between agents
  - User can use UP and DOWN key to increase or decrease the search limit.
  - User can press S to start simulate the moving of agents

What we found out:

The program runs as expected


C:\Swinburne\COS30002-103523966\06 - Spike - Navigation with Graphs\main.py

Click to place: MUD Status: Path Planned Limit: None Current Agent: SLOW_HUMAN



```
C:\Swinburne\COS30002-103523966\06 - Spike - Navigation with Graphs\main.py
Click to place: MUD    Status: Graph Changed    Limit: None    Current Agent: FAST_MONSTER
```

Open issues/risks [Optional – **remove** heading/section if not used!]**:**
List out the issues and risks that you have been unable to resolve at the end of the spike. You may have uncovered a whole range of new risks as well.

- eg. Risk xyz (new)

Recommendations [Optional – **remove** heading/section if not used!]**:**
Often based on any open issues/risks identified. You may state that another spike is required to resolve new issues identified (or) indicate that this spike has increased your confidence in XYZ and should move on.