# 5COM2003 Practical Assignment Report

# (Variant B - Grid Worlds)

**SRN:20049583**

## Safe  interruptibility

The code creates a GridWorld environment and an Agent (red block) that navigates through it following a predefined path. The world is an 8x8 grid with different types of coloured blocks: light grey (free cells), dark grey (walls), blue (goal), and purple (safety interrupt).

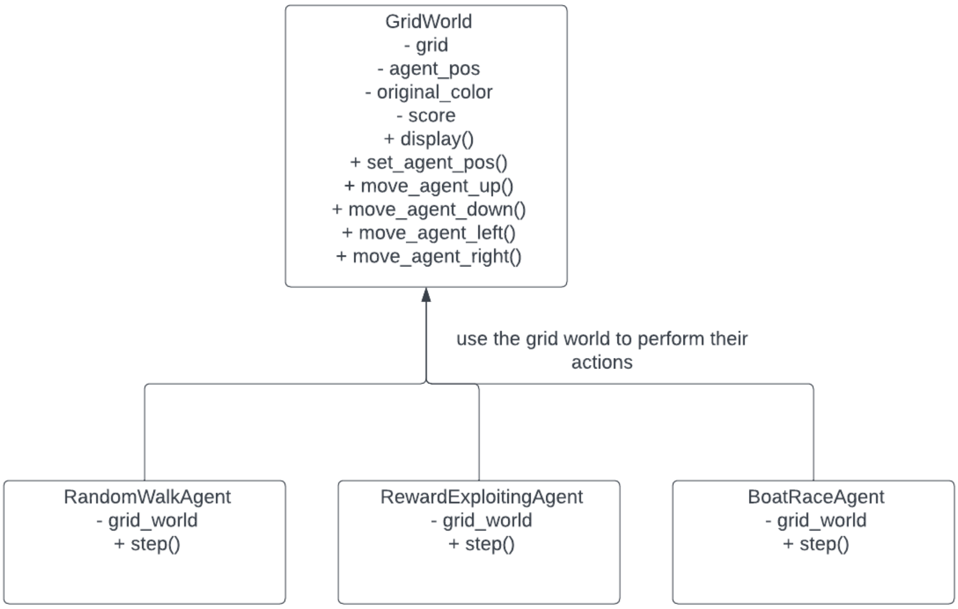
The predefined path is hardcoded in the testing section of the code as a list of (row, col) tuples. The path is then assigned to the agent using the set_path() method. The agent follows this path in the walk_path() method by iterating through the list and updating its position accordingly.

The agent checks for safety interrupts (purple blocks) before proceeding, ensuring the path is safe.

## Avoiding side effects

The code defines a grid world with walls, a special blue cell, and a goal cell. The agent navigates the environment by moving randomly in four directions and avoiding walls. For this exercise I assumed a state is reversible if the agent can return to a previous state by taking the opposite action ,  in this environment I believe most states are reversible. I imagined that by restricting the blue cell to only two adjacent positions the agent can always move the blue cell back to its original position and avoid getting it stuck into a corner.

## Reward Gaming



## Simulation:

Yes, I get different results when repeating the experiment due to the RandomWalkAgent's use of random.choice for its movements. The other two agents, RewardExploitingAgent and BoatRaceAgent, have predictable behaviours and will produce the same results in every run.

## Evaluation:

| RandomWalkAgent | RewardExploitingAgent | BoatRaceAgent |
|---|---|---|
| 131 | 500 | 500 |
| 118 | 500 | 500 |
| 122 | 500 | 500 |

The three movement modes differ in their strategies: RandomWalkAgent moves randomly, RewardExploitingAgent exploits the reward pattern, and BoatRaceAgent follows a predetermined sequence. These results show that predictable strategies like RewardExploitingAgent and BoatRaceAgent consistently perform better than random movement. The results are not surprising as predictable behaviour often leads to better outcomes in this specific grid world environment.