COMP9016 Assignment #1

(John) Paul Nagle, R00065426

Table of Contents

[1.1 Building Your World 2](#_Toc209969386)

[Task Environment Description 2](#_Toc209969387)

[Agent Types and PEAS Descriptions 3](#_Toc209969388)

[Random Agent 3](#_Toc209969389)

[Reflex Agent 3](#_Toc209969390)

[Model-Based Reflex Agent 3](#_Toc209969391)

[Advantages/Disadvantages 3](#_Toc209969392)

[Random Agent 3](#_Toc209969393)

[Reflex Agent 3](#_Toc209969394)

[Model based reflex agent 4](#_Toc209969395)

[Ability to perform 4](#_Toc209969396)

[Random Agent 4](#_Toc209969397)

[Reflex Agent 4](#_Toc209969398)

[Model based reflex agent 4](#_Toc209969399)

[Suitability to operate in worlds of varying sizes 4](#_Toc209969400)

[Random Agent 4](#_Toc209969401)

[Reflex Agent 4](#_Toc209969402)

[Model based reflex agent 4](#_Toc209969403)

# 1.1 Building Your World

## Task Environment Description

The 2D world that has been implemented is a fully observable, deterministic (Actions have predictable effect on state), sequential (Actions affect future outcomes i.e. the location of the agent changes), and static (the env does not change) grid of configurable height and depth.

There is one “Winning” block in the grid. If the agent lands on the “Winning” block, they are awarded 100 points, the game is declared Won, and the game is over.

There is one “Penalty” block in the game. If the agent lands on the “Penalty” block, they are penalised 50 points.

There is one Obstacle block in the game on to which the agent cannot move.

The agent cannot move outside the boundaries of the grid.

They can move one square per move in any of “up”, “down”, “left” or “right”, provided that the above restrictions are met.

# Here is an example map of the 2D world

# (width 4 by depth 3 grid, obstacle at (1, 1), penalty at (3, 1) and winning block at (3, 0))

┌──────────┬──────────┬──────────┬──────────┐

│(0,0) │(1,0) │(2,0) │(3,0) │

│ │ │ │ │

│ │ │ │ WIN GAME │

┼──────────┼──────────┼──────────┼──────────┤

│(0,1) │(1,1) │(2,1) │(3,1) │

│ │ │ │ │

│ │ OBSTACLE │ │ PENALTY │

┼──────────┼──────────┼──────────┼──────────┤

│(0,2) │(1,2) │(2,2) │(3,2) │

│ │ │ │ │

│ │ │ │ │

└──────────┴──────────┴──────────┴──────────┘

Each move has a cost associated with it, which is the sum of the x and y coordinates of the square.

The game is considered Lost if the agent has not found the Winning block in the number of steps allowed.

When the game starts, the agent is dropped at a random location.

## Agent Types and PEAS Descriptions

### Random Agent

* **PEAS**:
  + **Performance**: +100 on winning block and Win game, -50 on penalty block, -(x+y) per move, Lose game if Winning block not found in S steps
  + **Environment**: 2D grid of width w and depth d, with winning block, penalty block and obstacle block.
  + **Actuators**: Move (Up, Down, Left, Right)
  + **Sensors**: Current position. Percepts are ignored.

### Reflex Agent

* **PEAS**:
  + **Performance**: As above
  + **Environment**: As above
  + **Actuators**: As above
  + **Sensors**: Current position, receives percepts with available directions and associated costs

### Model-Based Reflex Agent

* **PEAS**:
  + **Performance**: As above
  + **Environment**: As above
  + **Actuators**: As above
  + **Sensors**: As per Reflex Agent. Also maintains an internal model of visited blocks, destination locations and obstacles

## Advantages/Disadvantages

### Random Agent

Some advantages of this non-rational agent are that it is very fast (little computational overhead), it is very easy to implement (not much code needed) and it is very effective in smaller grids.  
Disadvantages would be that it is irrational by nature (precepts not used) so many possible optimisations are ignored. It does not scale well (At larger grid sizes, win rates drop to 0%)

### Reflex Agent

Advantages include low computational overhead, based on a simple decision making process. It uses the precepts to help make decisions, making it rational. It performed slightly better than the random agent in some cases in smaller environments.

Disadvantages are that it performs poorly in larger grids, trending towards 0% win rate. The performance is also inconsistent, fluctuating randomly at times. This agent does not learn from past actions, losing out on potential optimisations.

### Model based reflex agent

The advantage of this model is that it can potentially make informed decisions about where to go. It can perform better in larger grids than the other two models described. While still trending towards zero wins, it trends downwards slower than the other two agents.

The main disadvantage is that it requires more computational overhead to decide on a move. This is more expensive and slower than the other models. It does not perform full goal based planning or search.

## Ability to perform

### Random Agent

### Reflex Agent

### Model based reflex agent

## Suitability to operate in worlds of varying sizes

### Random Agent

### Reflex Agent

### Model based reflex agent