Single-Agent and Multi-Agent Search in Maze Games
Project Plan

Natalia Kenrick
Supervisior - Farid Shahandeh

September/October 2022

Contents

1 Abstract

The aim of the project is to implement general search algorithms to single and multi agent environments in order to find the optimal route through a maze and to complete some goal. By designing and implementing agents with different goals, each will need to find efficient ways through the maze in order to complete their goal.

This project is intriguing because it shows how we can use autonomous agents to complete some task without user input. This is especially interesting in a multi-Agent scenario where agents can communicate and help each other to reach their goal. Agents are also an efficient option when we need to scale our application, especially when compared to using objects who do not have autonomy over their actions. A very simple example of intelligent agent use, specifically a software agent, is to use them as an extension of the user. Agents are able to scan the internet and retrieve information that are of known interest to the user, without getting distracted or tired.

An agent is something that can perceive its environment through sensors, and can complete some autonomous action through actuators in said environment. The environments the agents will be working in are accessible, deterministic, static and discrete. Given these environment properties, agents are able to perceive the whole environment, all moves are known, the environment will remain unchanged after any action, and there will be finite amount of actions that our agents can perform.

An agent is considered intelligent if it has the following characteristics: reactivity, pro-activeness and social ability. In this project, the use of intelligent agents will be visible by the way they are able to react to their environment and proactively pick the most optimal next step for their goal. Later, the social ability of communicating/interacting with different agents will be displayed.

In order to implement the search algorithms to our maze, we will model the maze as a search problem. By using simple search algorithms such as depth-first search and breadth-first search, we can find a specific location in a maze. Later, we can introduce a cost function to our search to find the shortest distance. We can also apply more complex search algorithms, such as the A* algorithm, in order to better inform our agent of its surroundings. By combining different search algorithms, we can produce the most optimal outcome for our goal.

2 Goals/Objectives

The main goal of this project is to show how different search algorithms work, and how the combination of different algorithms applied to our agents can give us an efficient and optimal outcome.

During the first term, one of the main objectives is to find a way to display a search algorithm in a way that visually makes sense. Another key objective is to instantiate an agent into the maze environment and have it behave correctly according to the algorithm that should be running. From there, the objective is to implement more algorithms to show the different way the agent moves around the maze. Once the visualisation is working, introducing new algorithms/functions should be simple.

By the second term, focus should be switched more onto the agents. As agents will behave in different ways we need to make sure they are implemented correctly, especially when in a MultiAgent scenario. Writing the evaluation function for the agents will most likely prove quite difficult therefore making sure that it is done carefully and properly is a main objective.

During this term I also hope to describe in detail what agents are being used as well as how and why they are being implemented to visualise an algorithm.

By the end of this project I hope to showcase how the use of intelligent agents can be used to automate tasks by displaying how their decision making movements occur in a game-like scenario.

3 Timeline

Term	Goal	Start Week	Date	End Week
1	Write code for maze generation	Over Summer		
	Reading on agents			
	Write intro/aims/objectives of report	2	26/09/22	3
	Reading on search algorithms	2	26/09/22	3
	Reading on graph visualisation/implementation in python	3	3/10/22	4
	Write theory on search algorithms and agents	4	10/10/22	9
	Implement DFS and BFS	4	10/10/22	7
	Add cost function & implement uniform cost search	7	31/10/22	9
	Write summary of completed work in report	10	21/11/22	11
2	Write rationale for report	1	9/01/23	2
	Implement A* algorithm	1	9/01/23	2
	Implement greedy search	2	16/01/23	4
	Write literature review & theory	2	16/01/23	6
	Add additional agents	4	30/01/23	7
	Write contents and knowledge as well as technical decision making	6	13/02/23	7
	Reading about and adversarial Minmax search and Expectimax	7	20/02/23	7
	Add adversarial Minmax search to agents by writing evaluation function	7	20/02/23	8
	Implement alpha-beta pruning	8	27/02/23	9
	Implement Expectimax agent	9	6/03/23	10
	Write professional issues and Critical analysis and discussion	7	20/02/23	11

3.1 Gantt Chart

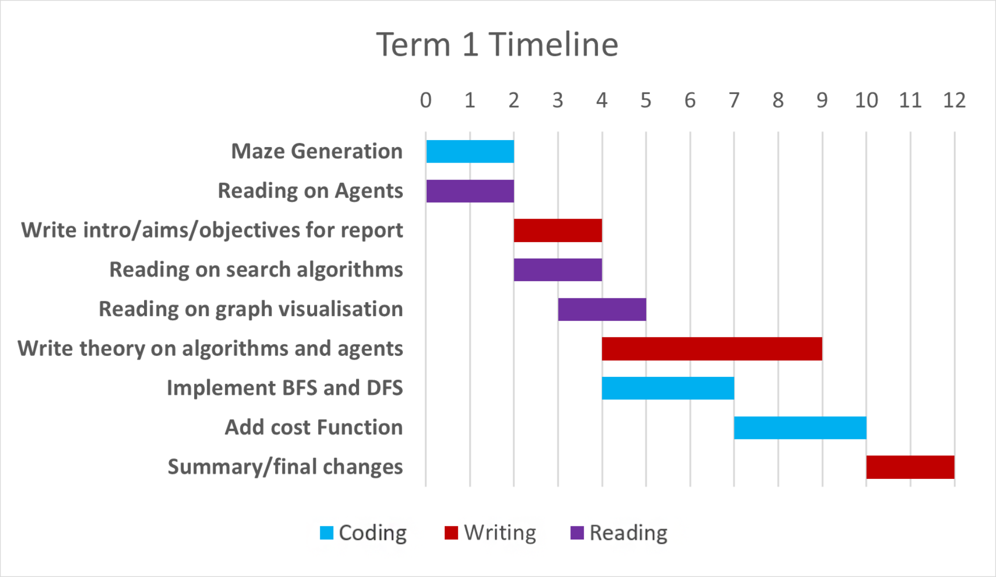

Figure 1: Gantt chart for term 1 (week 0-2 represents time over summer)

4 Risk Assessment

One risk that is likely to occur is finding the complexity of an algorithm more difficult then expected and therefore not advancing as quickly as planned. During this project I will have to write and integrate many search algorithms. To mitigate this I will ensure to use my time elsewhere such as switching focus to writing more of the report or reading for the project to get back on track. I will also ensure to read through how the algorithm works in detail so that it can be written correctly

A risk that may occur if finding the combination of algorithm + agent + visualisation challenging. During this project I will have to combine several methods/files in order to get them to work together. To mitigate this I will ensure to do reading on applying functions to agent programs as well as read over the documentation for PyGame to ensure I know how to apply the visualisation properly.

Another risk that may occur is not being able to visualise some of the algorithms properly. This is important as failure to visualise a search algorithm defeats one of the main purposes of this project. This could be mitigated by using a different method of visualising it from the others, or ensuring that the algorithm has been implemented correctly before trying to visualise it.

One more risk that could occur is having agents that do not work properly together. When implementing more that one agent the code is sure to become more complex. This could cause conflicts between multi - agents that weren't supposed to happen. To mitigate this the best option would be to ensure that the agent function is behaving correctly and that the evaluation function is being implemented properly.

5 Bibliography

Michael Wooldridge. 2009 (reprinted 2012). 2nd edition. An Introduction to MultiAgent Systems. Useful for understanding what and agent is and how it can be defined.

Stuart Russell, Peter Norvig. 2010. Artificial Intelligence: A Modern Approach, 3rd edition. useful background reading on agents and search algorithms.

Nick Jennings, Micheal Wooldridge. 1996. IEE Review. http://www.cs.ox.ac.uk/people/michael.wooldridge/pubs/review96.pdf. Useful explanation of software agents and their use.