Final Year Project Report

Full Unit - Project Plan

Playing Games and Solving Puzzles Using AI

Ong, Man Hei

A report submitted in part fulfilment of the degree of

BSc (Hons) in Computer Science

Supervisor: Dr Wahlstrom, Magnus

Department of Computer Science
Royal Holloway, University of London

October 10, 2024

Abstract

Human evolution, from the Stone Age to the Industrial Age to the current era of the Information Age. Humanity's technological advancements reflect our innate desire from mere survival to solve complex problems, yet to understand and explore the fundamental mechanisms that drive the natural world through scientific inquiry—groundbreaking technological growth that has driven many possibilities, such as the development of AI and gaming.

Development of Artificial Intelligence. Opening to innovative solutions for various problems with a level of intelligence, providing aids in areas such as research, and modern games and puzzles, enables us well to document complex tasks and puzzles with speed and accuracy. Puzzle and AI, Artificial Intelligence has improved over time. For example, the case of the AI chess engine. One of the world champions, Garry Kasparov [10]., played against an AI called Deep Blue during 1996-1997. Garry Kasparov resigned during the final match. During modern times, another chess AI engine called Stockfish [5] also defeated the current world champion, Magus Carlsen [2] with the highest rating of 2882 in history. Marking a significant milestone in artificial intelligence's capabilities.

This project aimed to deliver AI's ability to learn, adapt and optimize. Solving one of a classical puzzle game, "Unblock Me," also known as "Rush Hour," a popular sliding block puzzle game. The game requires the player to maneuver brown blocks that obstruct a designated red block through a 6 x 6 grid. The yellow blocks can be moved either vertically or horizontally depending on their orientation, while the red block can only move horizontally. The objective is to guide the red block to the exit using the fewest possible moves."

Python will be utilized to develop the puzzle game, supported by the Pygame module library. Pygame [8] is an open-source Python library designed for creating 2D games or multimedia applications and is based on the Simple Direct Media Layer (SDL) library [4]. Techniques that are used to tackle "Unblock me" Puzzles, the adaption of the search algorithm such as Depth-first search, Breadth-first search [9], Greedy Search, Dijkstra's Algorithm, and lastly A* Search. In theory, each of these search methods can find the shortest path. However, they differ in terms of time complexity and cost, which will be examined in detail.

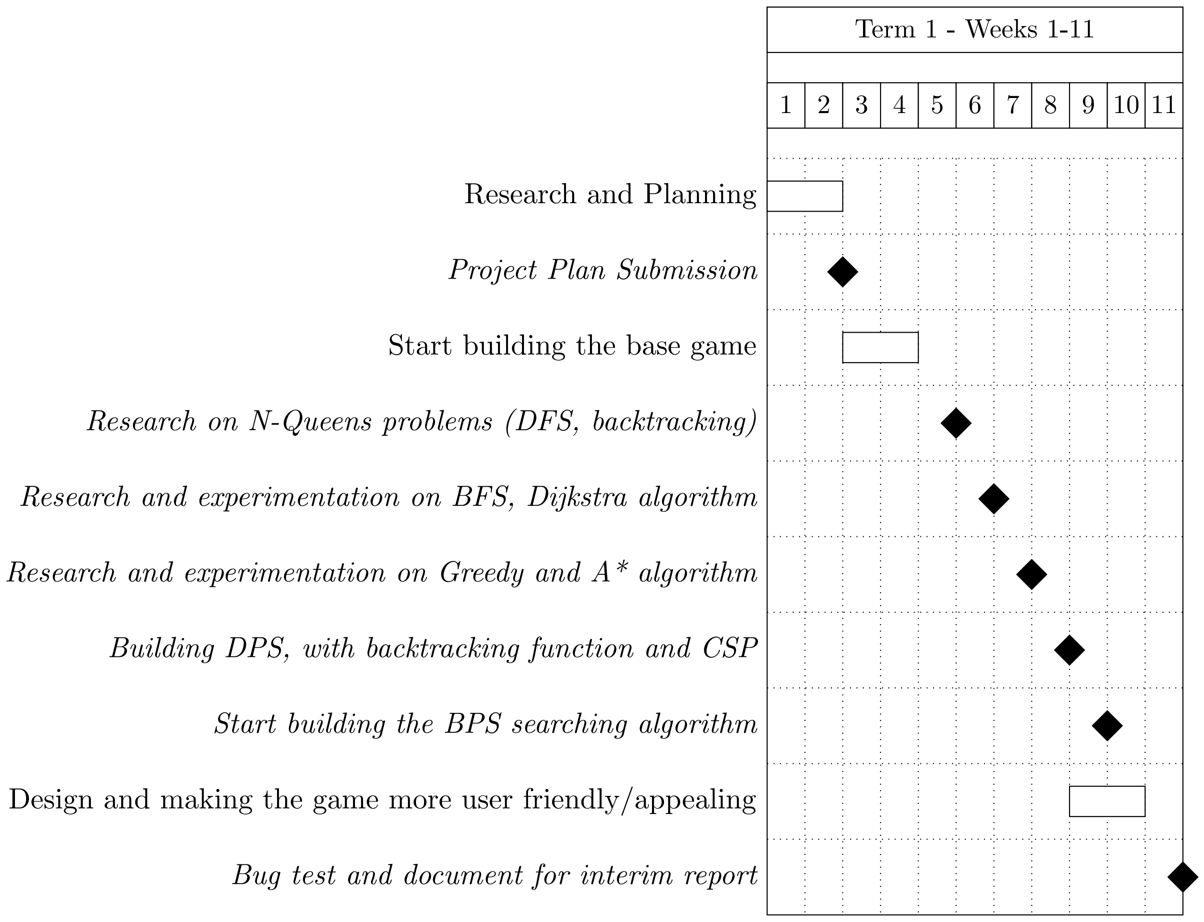
Theoretically, Constraint Satisfaction Problems (CSP) [1] along with backtracking can enhance the effectiveness of Depth-First Search (DFS) [6], as CSP make sure well-defined Variables, Domains, and Constraints known as ¡ "V, D, C" ¿. In this case, the variable would be the total number of blocks inside the 6 x 6 grid; the domain would be grid size or the block position; the constraints would be how the blocks move within the grid. Backtracking is enabled whenever a DFS search gets stuck in deep paths or loops, and backtracking is used to reverse the process or node. Allowing to explore different nodes or paths.

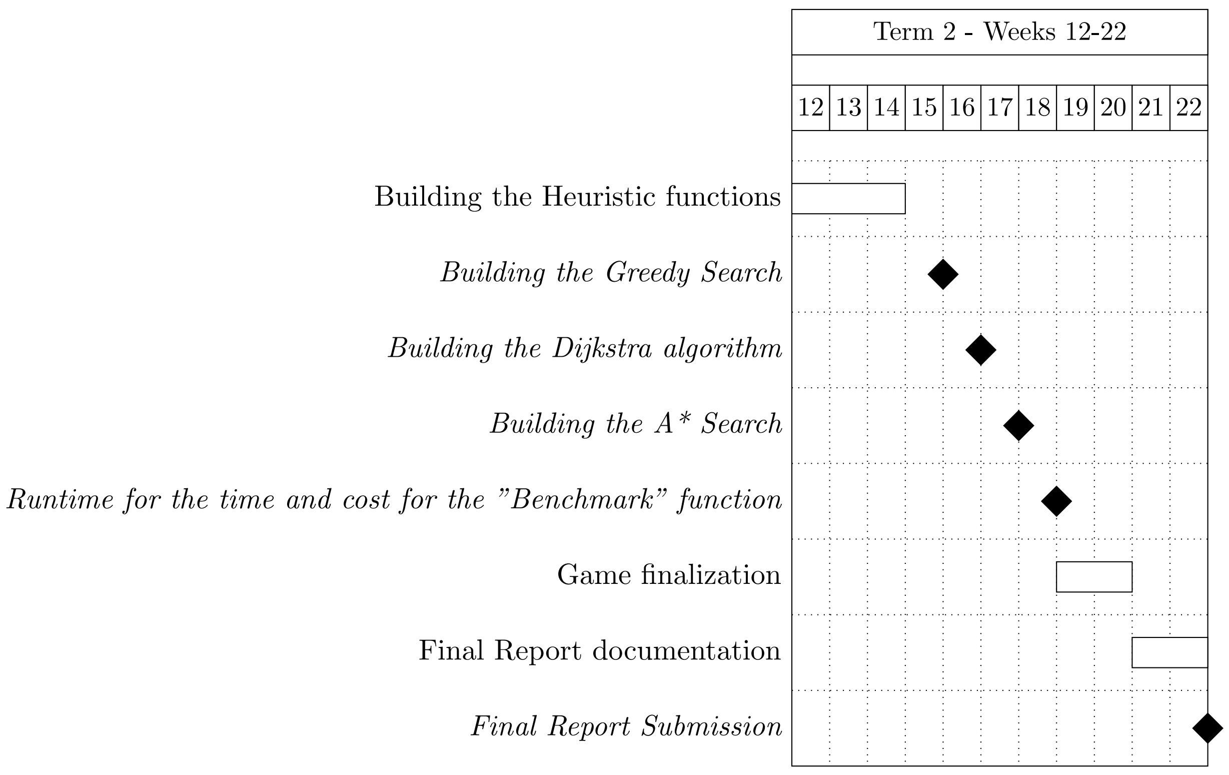
Heuristic functions [3] play a crucial role in guiding Greedy search, and A* search. The Greedy search process relies solely on the heuristic function. The A* search, on the other hand, combines the heuristic function with the actual cost of the path taken. A* ensuring the optimal solutions or paths are found.

In summary, this project's goal is to create a working graphical interface sliding puzzle game called Rush Hour using Python, create an AI model using different search methods, for example, DFS, BFS, Greedy Search, Dijkstra's Algorithm, A* search. that can solve the puzzle, and different results, time per cost, should be documented using Matplotlib known as a "benchmark". "Benchmark" would be used in various levels generated to determine which algorithm is the best.

Timeline

My plan will be focused on research and planning on term 1 and building the rest of the games in term 2. The remaining time will be used to fix bugs and errors.




Term 2 - Weeks 12-22
12
13
14
15
16
17
18
19
20
21
22
Building the Heuristic functions
Building the Greedy Search
Building the Dijkstra algorithm
Building the A Search*
Runtime for the time and cost for the "Benchmark" function
Game finalization
Final Report documentation
Final Report Submission

Motivation or goal on certain topic

1. **Start building the base game**
   The development of the base game for "Unblock Me" should begin with the creation of a fixed 6x6 grid, utilizing an array would be used for the creation of the grid. Each block should have its own set of orientation, color, and position data. Additionally, the game should have mechanics for determining valid moves, such as collision detection, and implement rules where only the red block can exit through the designated position at [5][2], which is the exit point. Lastly, blocks should be able to be dragged by the players.

2. **Research on N-Queens problems[7]**
   Researching the N-Queens problem can further understand the algorithm Depth-First search, the practical test can be experimental towards the Backtracking also with CSP.

3. **Research and practical experimentation on the algorithms**
   Research and practical tests on the algorithms, the practical test could be a simple maze solver using an array, it helps oneself to further understand how the algorithm works practically.

4. **Building DFS, with backtracking function and CSP**
   Building and implementing the DFS Search function according to the "Unblock Me", game logic should also be implemented into CSP.

5. **Design and making the game more user friendly/appealing**
   Designing the game, making the game look more appealing. For instance, the user should be able to understand each button. Dragging the blocks compared to moving them using arrow keys is better. All levels will be pre-generated for now.

6. **Building the Heuristic functions**
   A* Search and Greedy Search are heuristic-based algorithms used for finding the shortest path in graph traversal or path-finding problems. A* = f(n)=g(n)+h(n), and Greedy Search = f(n)=h(n). G(n) is the cost of the path from the start node to node, h(n) is the heuristic cost estimate of the cost from, and lastly, f(n) is the estimated total cost.

7. **Runtime for the time and cost for the "Benchmark" function**
   Benchmarks explained the performance of different algorithms (A*, DFS, BFS, Dijkstra's, Greedy) in terms of time and computational cost. Allowing keeping track of which algorithms are the best in solving different levels.

8. **Game finalization**
   It involves wrapping up all core features, polishing the gameplay experience, conducting thorough testing, and preparing the game the final report.

Risk and Mitigation

In every project, facing risks is inevitable. In this section, each risk comes with mitigation on different topics that will be suggested and discussed based on personal choice.

1. **Material confusion without the start of practical experiment**

   A common challenge is unfamiliarity with certain material within a project. Errors could be made if one lacks a proper understanding of key concepts. For instance, unclear about the definition of certain algorithms, and creating a non-optimal heuristic function, can lead to inaccuracies calculating in the search time. The suggested solution would be investing more time to conduct research, testing, optimization, and iteration.

2. **Underestimation of workload**

   Underestimating a certain topic might significantly impact the time that is allocated for a given task. If a task is overrunning then the given time which was mentioned in the timeline graph, it might affect the quality and the quantity of the later work. Therefore, other than good planning of the whole workflow, knowing how to adapt is important as well, early delivered tasks can use the spare time to work on the upcoming tasks as well.

3. **Integration Issues between Game Logic and AI**

   Creating a game with bugs and messy documentation is not optimal, the base game should be tested regularly and well-structured in the early development.

4. **Game-play performance and performance in general**

   Creating a game with a "hint" or solver button requires computing and memory power, if DFS is chosen for the "hint" algorithm for the player to use, might take a long time for the player to get the final solution, affecting the game-play. A* in theory could guarantee optimal and fast solutions, but a good heuristic should be made or chosen else the time could be affected negatively. Game-play performance and performance in general.

5. **User experience(installation)**

   A messy folder might affect the user to download and evaluate the game. Therefore, README.md and requirement.txt are included in the folder with all the instructions that are required for the user to run the game.

6. **User experience(Gameplay)**

   When considering the gameplay and level design, the absence of cohesive elements can significantly impact the player's experience. For example, in Unblock Me, the use of assorted colors for the blocks can create a visually overwhelming and chaotic effect. To enhance clarity and appeal, only using two different primary colors for the block to drag and move, making the game more easy to play and aesthetically appealing.

7. Data Loss

Data loss in a project can be a huge drawback to the ongoing progress. Hardware failure (personal laptop), software corruption, human error, etc., could be the reason leading to the incident of data loss. Therefore, any code or research would be suggested to be uploaded to the designated online and offline databases, For instance, GitHub/Lab, OneDrive or even an external hard drive.

Acronyms

DFS - Depth First Search

BFS - Breadth First Search

CSP - Constraint Satisfaction Problem

SDL - Simple DirectMedia Layer

Glossary

Python - A high-level, general-purpose programming language.

PyGame - A Python module which is designed for writing video games.

GitLab - Web-based tools for software development

NumPy - A Python module which is designed for scientific computing.

Matplotlib - A Python module which is designed for graph plotting.

Bibliography

[1] S. C. Brailsford, C. N. Potts, and B. M. Smith. Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119(3):557–581, Dec 1999.
—Value to the project: Understanding whats CSP consists of, optimal variables, domain and constraints for the ”Unblock Me” game.

[2] Chess.com. Magnus carlsen — top chess player. *Chess.com*, 2014.

[3] X. Cui and H. Shi. Direction oriented pathfinding in video games. *International Journal of Artificial Intelligence Applications*, 2(4):1–11, Oct 2011.
—Value to the project: Theory for Search Algorithms like Dijkstra's algorithm and A* algorithm.

[4] S. Lantinga. Simple directmedia layer - homepage.

[5] S. Maharaj, N. Polson, and A. Turk. Chess ai: Competing paradigms for machine intelligence. *Entropy*, 24(4):550, Apr 2022. ”For end users, our work implies that Stockfish may currently be the better tool for studying deep puzzles. To confirm this hypothesis, a chess studies dataset ought to be created and used as a benchmark for modern chess engines.”.

[6] S. Makki and G. Havas. Distributed algorithms for depth-first search. *Information Processing Letters*, 60(1):7–12, Oct 1996.
—Value to the project: Theory of DFS, their time complexity and Pseudocode are well written.

[7] K. Mathew. Experimental comparison of uninformed and heuristic ai algorithms for n puzzle and 8 queen puzzle solution. *International Journal of Digital Information and Wireless Communications*, 4(1):143–154, 2014.
—Value to the project: Theory for the N-Queens problem additional solution using DFS AND BFS to solve it.

[8] Pygame. About - pygame wiki, 2019. ”Pygame is a set of Python modules designed for writing games. It is written on top of the excellent SDL library.”
—Value to the project: Pygame is the main engine for the creation of ”Unblock Me” game.

[9] D. Scarpazza, O. Villa, and F. Petrini. Efficient breadth-first search on the cell/be processor. *IEEE Transactions on Parallel and Distributed Systems*, 19(10):1381–1395, Oct 2008.

[10] C. Team. Kasparov vs. deep blue — the match that changed history. *Chess.com*, Oct 2018.