Artificial Intelligence for Games – Project Research Workbook

This workbook will help you focus your research for your project.  
Once you have answered these questions, use this information in your Technical Design Document.

|  |
| --- |
| **Briefly describe the application, game or simulation you are researching.**  **(This is your initial idea to focus your research. The application described in your design documents or your final build may end up being different from this description)** |
| My application Will be a Stealth game, the player Traverses Throughout a Dungeon full of goblins to steal back his loot/gold  Goblins will charge at you and attack the player if they are found. There will be multiple chests the player needs to loot.  Once all the loot/gold is collected head to the ladder and escape.  Stretch goals goblins fleeing on low health, traps, More Enemies |

|  |
| --- |
| **What pathfinding algorithm will you implement.**  **List some (at least one) online or class resources that will assist you in implementing this algorithm.**  **Examples may include YouTube videos, blogs, textbooks, or class resources** |
| Dijkstra is the pathfinding algorithm I'm implementing  Videos on SharePoint, YouTube videos explaining how to implement, AIE game ai videos that explain pathfinding.  <https://www.youtube.com/watch?v=pVfj6mxhdMw>  https://www.youtube.com/watch?v=LDCa4U-ZbJ0 |

|  |
| --- |
| **Describe how this pathfinding algorithm will be used by the agents in your application.**  **For example,**   * **Will the player-controlled character find a path to the last click?** * **Will AI agents follow a target?** * **Will the destination be updated dynamically (if so, at what interval), or only when the agent reaches its current destination**   **Reflect on (and describe here) how these choices influence the design and development of your application.** |
| * Player will be controlled by keyboard behaviour. * The goblins will be using Seek, FollowPath, PathFinding, Wonder and Flee * Goblin 1 will be wondering around the map. * Goblin 2 will be following a particular path as a guard * When you are seen by a goblin they will all seek the last known location then make a choice go back to what there were doing or continue to seek your last known location |

|  |
| --- |
| **What AI strategies could be used. (For example, state-machines, blackboards, decision trees, etc)**  **Pick two strategies and list a brief description of how they work, along with some resources that could assist you in implementing these strategies. (Examples may include YouTube videos, blogs, textbooks, or class resources)** |
| **BlackBoards**   * When the player is spotted their position is posted on the blackboard. * The ai will change its behaviour based on the blackboard updating the players position.   **Youtube link**: <https://www.youtube.com/watch?v=-93hYX-yG98> - What is BLACKBOARD SYSTEM?  **Decision trees**   * Used to predict the outcome of an event given attributes * “A flowchart like structure in which each internal node represents a test on an attribute “ * A branch represents an outcome of the test and each leaf node represents a class label (decision taken after computing all attributes)  **Youtube link:** https://www.youtube.com/watch?v=aVf3awPrVPE - Behaviour Trees in Unity using C# |

|  |
| --- |
| **What is a Technical Design Document, and why is it useful?**  **List the major topics or points of discussion (at least 5) commonly included in technical design documents.** |
| A tdd is useful because it helps the audience of my game understand the background of what's happening e.g the behaviours of my ai but shows how to play my game.   * Title page – Showcases the name of the product * Quick summery – the start of the tdd should be a quick statement explaining whats in the product * Entities – Shows and explains what entities do in the product * Flow chart – Explains how the player paths through the product e.g through each game states |

|  |
| --- |
| **List the libraries, APIs, algorithms, or assets (i.e., any pre-existing component) that you will use, or might consider using, when implementing your application.**  **For each component, include a brief statement listing it’s functionality, suitability, and technical impact on the project.** |
| **Raylib -** This game library is used to help in the programming by having pre made functions.  **Assets -** The functionality of the assets is to give the game a medieval look.  This links provides where I got the following from:  <https://opengameart.org/content/dungeon-tileset>   * Ladder * Map * Player   <https://opengameart.org/content/animated-orc-goblin-16bit>  - Goblin |

|  |
| --- |
| **What are the licensing arrangements or restrictions for the pre-existing components you have identified?**  **(Identify the licence of each component)** |
| Public domain licence - https://creativecommons.org/publicdomain/zero/1.0/   * No Restrictions   Gpl.20 (Some Rights Reserved) - http://www.gnu.org/licenses/old-licenses/gpl-2.0.html   * Some required to give Attribution by giving you a link to the original source. * Some were free until you wanted to make money off it. |

|  |
| --- |
| **Identify a possible audience for your application, game or simulation.**  **What platform(s) are you targeting?** |
| The audience I'm going for is a 10-20 age group who enjoys a medieval dungeon feel.  The platform I'm targeting is a pc build only. |

|  |
| --- |
| **Describe the real-world environment you will be simulating.**  **Describe any real-world aspects that are present in the game world, and how their real-world behaviour will influence your simulation.**  **For example, does your program have animals that will head towards water when thirsty? Or will you implement prison guards that head towards the source of a noise?** |
| My goblins in my game have a feature where they will head towards the player when spotted and will “catch” the player if they stay to long in the radius of the goblins. The radius also lights up red over time to indicate they are being caught. |

|  |
| --- |
| **Try to describe the complexity of your project.**  **For example,**   * **How complex are the AI behaviours you have chosen?** * **How many classes are needed to simulate all your entities?** * **How deep is the inheritance hierarchy?** * **Which algorithms are the most complex or difficult to implement?** |
| The complex varies in the behaviours, the least complex is the wonderer, they just pick a spot inside their radius and moves towards it, the 2 most complex is the patrol and seek. Using Dijkstra's Algorithm, I am getting a list of nodes around the map and adding them to a list and moving to the closest node each time, the patrol is to a yellow tile and back, the seek is to the players position known by a blackboard method.  This was the most complex system and the most difficult to implement  I have 4 entities, goblins, the player, chests and a ladder,  The inheritance hierarchy only goes down a couple of connections, between the entity class and the main game state. |

|  |
| --- |
| **Identify any tools that may help you in your implementation.**  **For example, graphical applications for asset development, debuggers or IDEs.** |
| Tilemap – used to create the background image  Paint.net - used to create the coloured tiles in the background. |

|  |
| --- |
| **List any other additional information that may aid in the design of your project.** |
| All assets I used had the license – public domain |