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) |
| I will be researching and creating will be a game like hide and seek (stealth, survival genre) where the player will simply move around, avoiding the AI enemy that is roaming and will need to survive for as long as possible. There will be items to interact such as a health station that will restore the players health up to a random amount. |

|  |
| --- |
| 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 |
| one of the pathfinding algorithms that i will be looking to implement into this research/project will be A\* algorithm for the enemy AI.  sources for A\*:  - AIE lecture material - A\*  - A\* Pathfinding in Unity - A\* (by Code Monkey on Youtube): <https://www.youtube.com/watch?v=alU04hvz6L4>  - Game Programming algorithms and techniques - Pathfinding (by Sanjay Madhav, pg: 180 -> 200): <https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Game%20Programming%20Algorithms%20and%20Techniques_%20A%20Platform-Agnostic%20Approach%20%5BMadhav%202013-12-29%5D.pdf> |
| 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. |
| This pathfinding algorithm will be used on the AI enemy in the game to calculate the fastest route to chase/pursue the player to attack and defeat them, when within line of sight or within a certain distance. The algorithm will be updated after the AI is at the L.K.P (last known position) but if the enemy cannot sense the player or see, so the algorithm will be updated when AI is at the set location. The state machine will be used since there are (currently) 3 different states for the AI (two for the health station) such as:  - enemy: roam, chase, attack  - Health station: empty, full |

|  |
| --- |
| 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) |
| As mentioned earlier, state-machines will definitely be used to change the behaviour of the AI. decision trees could possibly be used or even blackboards.  **state-machines:**  State-machines is a technique used to define what actions/instructions should follow when the required state of the object (such as if health is at 0, do this...) has been met, should only have a few states, but if there is more required, use another technique.  references:  - AIE lecture - finite state-machines  - Game Programming algorithms and techniques - Pathfinding (by Sanjay Madhav, pg: 192 -> 197): <https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Game%20Programming%20Algorithms%20and%20Techniques_%20A%20Platform-Agnostic%20Approach%20%5BMadhav%202013-12-29%5D.pdf>  - Implementing finite state (by Joonyee Chuah): <https://www.youtube.com/watch?v=pxaIyNbcPrA>  **decision trees:**  Decision tree is a technique used to make decisions while not affected by interruptible states. This will be useful and needed for the AI to make a decision based on the current state it’s in.  references:  - AIE lecture - decision trees |

|  |
| --- |
| 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 technical design document (TDD) is a written document used by the development team to describe in detail either the entire or parts of the design/s such as how a program will behave/think, a function will perform, etc. what it contains includes:  - The signature of an interface, with all including data types/structures required.  - Detailed class model with all the methods, attributes dependencies and the associations.  - The specific algorithm/s that is used for a component.  - Physical data models that include the attributes and types of each entity/data type.  So in other words, the product requirements are defined, the technical requirements are defined, the technical design is described, followed with the implementation of the solution and the testing of the solution. |

|  |
| --- |
| 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 library will be used to assist in creating the game application, as well as using its mathematical and image functions. This is an essential component for this project for it to work.  Kenny asset store will be used for the assets/images in the project to look more appealing. |

|  |
| --- |
| What are the licensing arrangements or restrictions for the pre-existing components you have identified?  (Identify the licence of each component) |
| * Raylib licensing: This is a closed source software that is safe for my uses as it restricts uses to personal and commercial without sharing the code. * Kenny allows free access to some asset materials but are not responsible for any issues regarding its products. |

|  |
| --- |
| Identify a possible audience for your application, game or simulation.  What platform(s) are you targeting? |
| The targeted audience for this game is for people who are into setting a record to see how long they will survive; this is the main target audience as this is meant to be something simple to make but still fun. It can only be played on the desktop so far, it is simpler to code the keys into an input. |

|  |
| --- |
| 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? |
| There is the aspect of being heard or seen that will affect the state-machine for the AI enemy as well as the health station if it is full or not. There are also objects to assist the player in hiding/running from the AI which will affect the A\* pathfinding algorithm. There will also be an area of view and area of hearing for the AI which mostly plays into the state-machine. If there are neutral NPC's later on down the line, if they are run into, they will move away and maybe periodically if idle, move elsewhere. |

|  |
| --- |
| 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 AI behaviour should be moderately simple to make and perform while the A\* pathfinding algorithm should not take too long to create. There are the state-machines (two classes (.h, .cpp)) to worry about followed by the algorithm (class (.h, .cpp)), then there is the player(class), enemy(class), Game (class - should be general console run functions such as init, update, draw, shutdown), then the actual main.cpp |

|  |
| --- |
| Identify any tools that may help you in your implementation.  For example, graphical applications for asset development, debuggers or IDEs. |
| -Visual studio intellisense.  - Kenny – free game asset store (includes Raylib assets) |

|  |
| --- |
| List any other additional information that may aid in the design of your project. |
| Could possible add items that have and effect on the player for them to pickup such as “speed boost” for example.  Possible NPC’s. |