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) |
| Top-down 2D stealth game, as described by the second brief, including features such as pathfinding and cone of vision. The game will be rendered using a tile map, with SDL for rendering. |

|  |
| --- |
| 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 |
| I will implement A\* pathfinding.  <https://www.youtube.com/watch?v=icZj67PTFhc>  Video by Javidx9 about A\* pathfinding in C++ using a grid of nodes |

|  |
| --- |
| 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. |
| AI agents will follow the player once he has been spotted.  The destination will be updated every time the player moves from one grid cell to another, so the AI agents will follow the last grid cell that the player was on. This will be faster than updating the algorithm every frame. |

|  |
| --- |
| 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) |
| A state machine could be used to control the AI agents as they either patrol, search, or seek the player. There are little enough states that a simple switch statement in each agent will be sufficient in order to implement this.  A blackboard could be used to let the other agents know when one agent spots the player. Any agent with a line of sight to the player can push the player position to the blackboard for the other agents to pathfind towards.  Resources:   * Class presentation slides about state machines and blackboards |

|  |
| --- |
| 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 details the specifics of how certain parts of an application will be implemented. It is important to help get a solid idea of how the program is going to come together, and for communicating the chosen methods of implementation to other team members.  Commonly included in technical design documents:   * Specifics of any external APIs, libraries or tools to be used. * Overview of what the final product should look like * Overview of how systems are supposed to interact with each other * Details about what algorithms will be used to implement certain non-trivial systems. * Details about technical goals for the final product, such as whether it is going to target low-power devices or not. |

|  |
| --- |
| 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. |
| * SDL (<https://libsdl.org>). I will use this for simple, cross platform 2D rendering and input. * Tiled (<https://mapeditor.org/>), along with my binary exporter for it (<https://gist.github.com/veridisquot/374a246b95c53886af60248e1861dd67>). Tiled will allow the level to be created more easily. The binary exporter will make it easier to load in the level from the file, without having to parse Json or XML. * 16x16 Dungeon Tileset (<https://0x72.itch.io/16x16-dungeon-tileset>) |

|  |
| --- |
| What are the licensing arrangements or restrictions for the pre-existing components you have identified?  (Identify the licence of each component) |
| SDL is distributed under the zlib license.  Tiled is distributed under either the Apache, BSD or GPL licenses.  The 16x16 Dungeon Tileset pack is distributed under the CC-0 license. |

|  |
| --- |
| Identify a possible audience for your application, game or simulation.  What platform(s) are you targeting? |
| The game will target Windows, though it will be possible to build it on any other desktop platform due to the cross-platform nature of SDL. |

|  |
| --- |
| 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? |
| The game will be set in a small dungeon, where the player must escape by collecting a key and unlocking a door without being detected by the guards.  The guards will use A\* pathfinding and wander around until they spot the player. When one of them spots the player, they will begin to pursue the player until they either catch him or he goes out of sight. The agent will also broadcast a message to all the other guards about the location of the player when he gets spotted. If the player goes out of sight, all the guards will path find towards the last known location of the player and continue their wander state. |

|  |
| --- |
| 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? |
| Two classes will be needed for entities, one for the player and one for the guards. Inheritance won’t be necessary, since the number of entity types is small, and they only have two common attributes, which will be the position and a sprite to render.  A class will be used to represent the map. It will be responsible for loading and drawing the map. A reference to an instance of this class will be passed to the enemies and the player so that they can do pathfinding and collision detection against it. |

|  |
| --- |
| Identify any tools that may help you in your implementation.  For example, graphical applications for asset development, debuggers or IDEs. |
| I will be using Premake5 as a build system to make the process of building and linking SDL easier and I will use Visual Studio for to build and debug the application. GIMP may be required if any edits to sprites is needed to make them more suitable for the game.  The “hexdump” utility will be used in order to debug the contents of any binary files that might be needed, such as the files generated by my Tiled exporter. |

|  |
| --- |
| List any other additional information that may aid in the design of your project. |
|  |