Appendix A:

* PDD

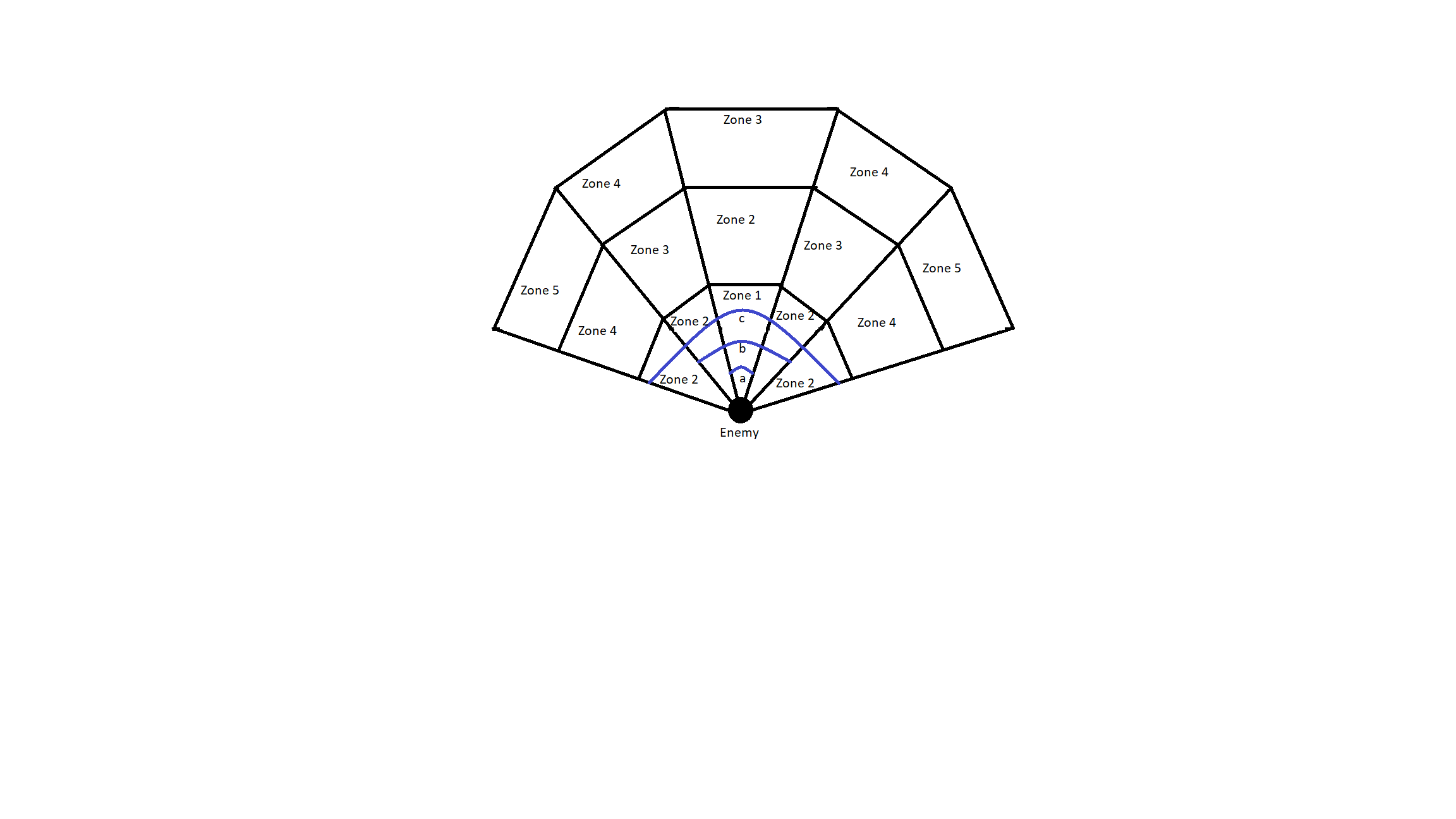
Appendix B:

REUSE SUMMARY

* Unity
* MS Paint
* Visual Studio
* Adobe Behaviour Tree Visual Editor
* Piskel

Appendix C:

MODELS/DIAGRAMS



*An example of an Enemy Vision Cone. 15 vision zones, split into 5 zone types. The higher the zone type, the slower the detection. 'a', 'b' and 'c' are the close, medium and wide angles respectively, displayed with the blue lines.*

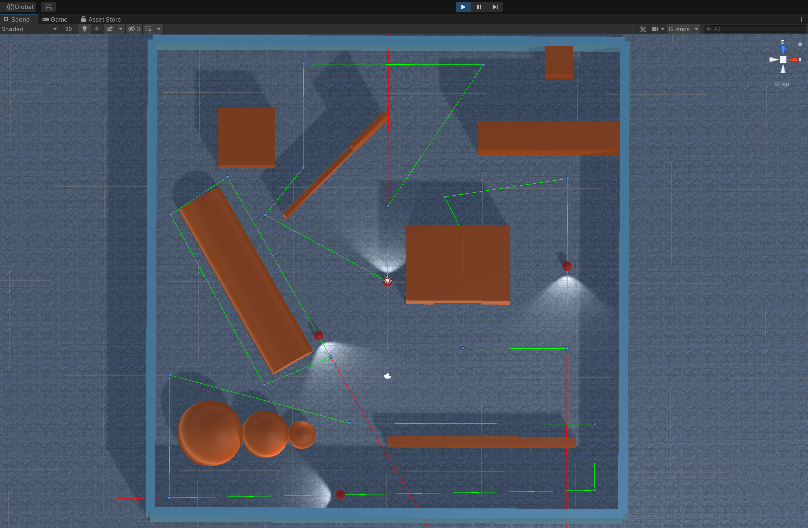
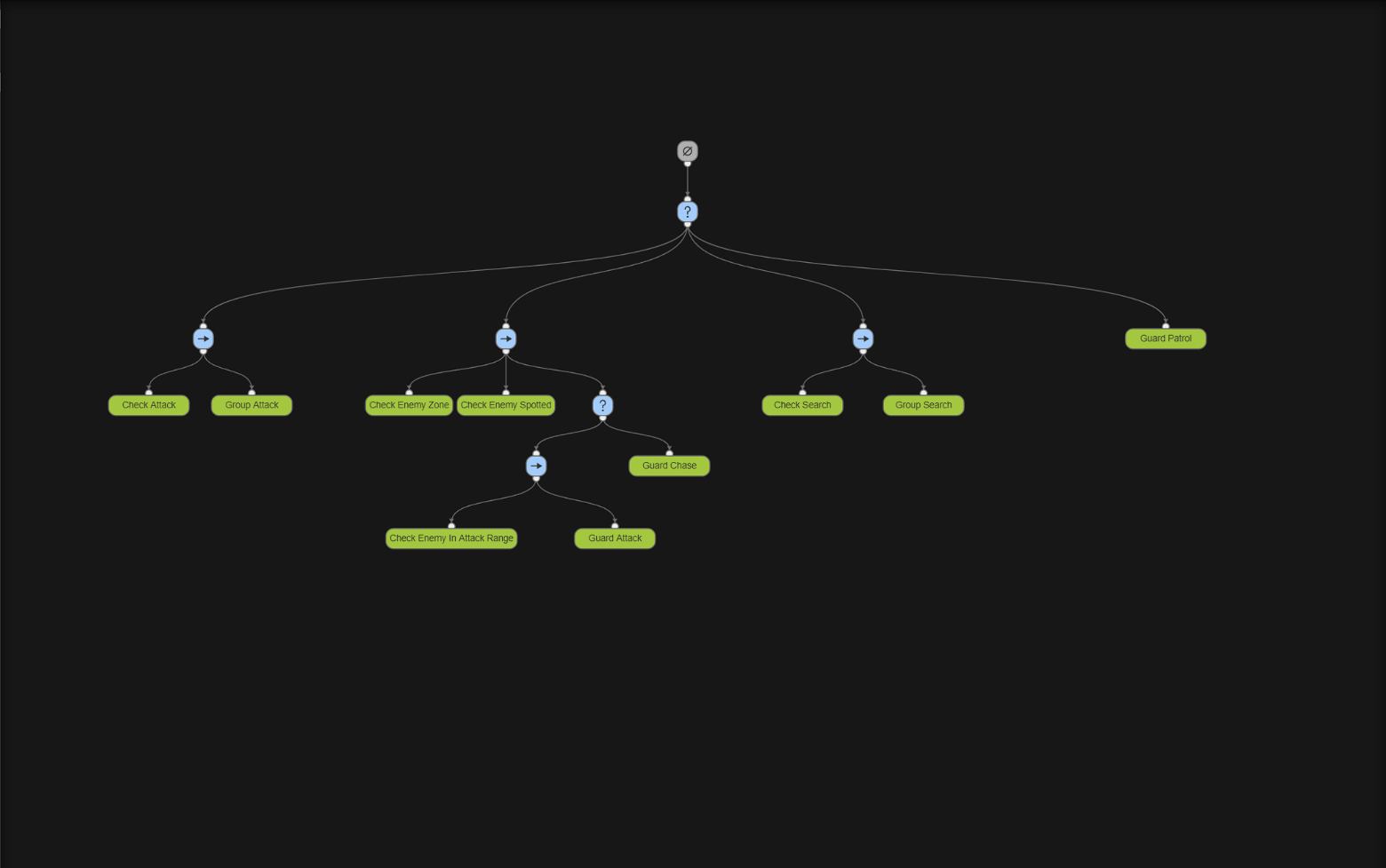


Diagram showing guards on patrol paths indicated by green lines and small blue dots.

Behaviour Tree Diagram showing the Guard Behaviour Tree layout and classes. (The topmost node is the root node. The ‘?’ nodes are selector nodes. The ‘->’ nodes are sequence nodes. The green nodes are leaf nodes.)



Appendix D:

USE CASE REQUIREMENTS

|  |  |
| --- | --- |
| **Use Case:** Enemy Detection | **ID:** 1C |
| **Description:** An enemy will be able to detect the player slowly over time | |
| **Primary Actors:** Enemy | **Secondary Actors:** |
| **Preconditions:**   1. Player is in an undetected state | |
| **Main Flow:**   1. The use case will begin when the player enters the enemy`s field of view 2. A timer will start to check how long the player is in the enemy`s field of view 3. If the timer reaches a pre-determined endpoint, the player will be spotted 4. The timer should have a shorter pre-determined endpoint if the player is closer and more central in the enemy`s line of sight | |
| **Postconditions:**   1. The player is spotted by the enemy | |
| **Alternative Flows:** The player escapes the enemy`s field of view before the timer ends | |
| **Preconditions:**   1. The player is in the enemy`s field of view AND the timer has not ended | |
| **Alternative Flow:**   1. The player escapes the enemy`s field of view 2. The timer starts counting back down until 0 3. The player is not spotted and the timer reaches 0 | |
| **Postconditions:**   1. The enemy does not spot the player | |

|  |  |
| --- | --- |
| **Use Case:** Enemy Pathfinding | **ID:** 1B |
| **Description:**  The enemies will use a pathfinding algorithm to get to the player’s last known location. The enemies will then path find from that point outwards to try to locate the player. | |
| **Primary Actors:** Enemy | **Secondary Actors:** |
| **Preconditions:**   1. The player has been spotted by an enemy AND has since escaped | |
| **Main Flow:**   1. All of the enemies nearby, convene on the player`s last known location 2. The enemies use a pathfinding algorithm to traverse different parts and corners of the map | |
| **Postconditions:**   1. The enemies end their search after a short search and return to their pre-determined patrol paths | |
| **Alternative Flows:** N/A | |

|  |  |
| --- | --- |
| **Use Case:**  Enemy Behaviour Tree | **ID:** 1A |
| **Description:**  The enemy’s behaviour will be dictated by a behaviour tree containing 4 states. The 4 states include Patrolling, Chasing, Attacking, Searching | |
| **Primary Actors:** Enemy | **Secondary Actors:** |
| **Preconditions:**   1. There are instances of guards using the behaviour tree within the game | |
| **Main Flow:**   1. If the player has not been spotted, the guard will patrol a pre-determined route 2. When a guard spots the player, all of the guards will chase the player 3. When the guards are within range, they will attack the player 4. If the player escapes, the guards will search the area 5. If the player is found again, they will chase and attack 6. If the player is not found again, they will go back to their patrol paths | |
| **Postconditions:**   1. The states will reset back to what they were before the player was spotted | |

|  |  |
| --- | --- |
| **Use Case:** UI Depicting Enemy States | **ID:** 2A |
| **Description:**  The enemies will have a small UI element/sprite above their heads depicting their current state. The 4 states the enemy can be in include, Patrolling, Chasing, Attacking and Searching. These will be split into 3 groups. One for Patrolling which will have no UI element. One for Searching will have a UI element/ Finally, one for Chasing and Attacking, which will have the same UI element. | |
| **Primary Actors:** Enemy | **Secondary Actors:** |
| **Preconditions:**   1. The game will be running | |
| **Main Flow:**   1. The guards will have no UI element above their heads when patrolling 2. The guards will have a small exclamation mark icon above their heads when the player has been detected and the guards are either chasing or attacking 3. The guards will have a small question mark icon above their heads when the player has been lost and they are searching | |
| **Postconditions:**   1. The guards will return to having no icons above their heads if the player has not been found and they return to patrolling | |
| **Alternative Flows:** N/A | |

Appendix E:

USE CASE TESTING

|  |  |
| --- | --- |
| **Use Case:** Enemy Detection | **ID:** 1C |
| **Test Number:** 1 | |
| **Objective:**  To test whether a player will be detected by the enemy and be detected at different speeds based on where they are in the enemy’s field of view | |
| **Set up:**  The player will take turns standing in the 5 different vision zones the enemy has. The player will start outside of the enemy’s field of view AND ensure the timer has not started. The player will then move to a vision zone and test how long it takes for the player to be spotted. The enemy’s spotlight will change to a different colour based on which zone the player has been spotted in, making it easy to visualise this test. | |
| **Expected Results:**  The player should be spotted after spending 1 second in zone 1 AND the enemy`s spotlight should go red.  The player should be spotted after spending 1.5 seconds in zone 2 AND the enemy`s spotlight should go magenta.  The player should be spotted after spending 2 seconds in zone 3 AND the enemy`s spotlight should go yellow.  The player should be spotted after spending 3 seconds in zone 4 AND the enemy`s spotlight should go green.  The player should be spotted after spending 5 seconds in zone 5 AND the enemy`s spotlight should go blue. | |
| **Test:**  The player will enter zone 1, check how long it takes for the enemy`s spotlight to go red and then leave the enemy`s field of view and wait for the timer to reset.  The player will enter zone 2, check how long it takes for the enemy`s spotlight to go magenta and then leave the enemy`s field of view and wait for the timer to reset.  The player will enter zone 3, check how long it takes for the enemy`s spotlight to go yellow and then leave the enemy`s field of view and wait for the timer to reset.  The player will enter zone 4, check how long it takes for the enemy`s spotlight to go green and then leave the enemy`s field of view and wait for the timer to reset.  The player will enter zone 5, check how long it takes for the enemy`s spotlight to go blue and then leave the enemy`s field of view and wait for the timer to reset. | |
| **Test Record:** Expected results observed | |
| **Date:** 23rd March 2023 | **Tester:** Tayyab Hussain |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case:** Enemy Pathfinding | **ID:** 1B |
| **Test Number:** 2 | |
| **Objective:**  To test whether the enemies are successful in independent pathfinding in order to search for the player. | |
| **Set up:**  The enemies should be going along their patrol paths as normal, to begin with. After they spot the player, they should path find to the player’s location. Once the player has escaped and the enemies can no longer see the player, they should path find to search for the player. | |
| **Expected Results:**  The enemies should all congregate at the player’s last known location. They should then each spend 15 seconds searching a pre-determined position on the map using the pathfinding algorithm to traverse to that location. After 15 seconds they should then path find to another location on the map and search there for 15 seconds. After 30 total seconds of searching, they should path find back to their patrol paths and continue patrolling. | |
| **Test:**  The player will start in an undetected state and check that the enemies are following their patrol paths.  The player will then enter an enemy’s vision zone and be spotted.  The player will then run and hide and be completely outside of any enemy vision zone until the enemies have completed both searches and returned to their patrol paths. | |
| **Test Record:** Expected Results Observed | |
| **Date:** 22/04/23 | **Tester:** Tayyab Hussain |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case:** Enemy Behaviour Tree | **ID:** 1A |
| **Test Number:** 3 | |
| **Objective:**  To test whether the behaviour tree can successfully implement the 4 relevant actions defined in the requirements specification and do so at the correct time based on the current game state. | |
| **Set up:**  The player will begin the game outside of the map to test the enemy patrol paths. The player will then be placed into the map and play in a way that will result in the enemy AI having to use all of the 4 different actions within the game and use them at the correct time. | |
| **Expected Results:**  The enemies should patrol when the player has not yet been spotted.  If a guard spots the player, all the other guards should be alerted and should pathfind to the player’s location. If the guards are close enough to the player they should attack. If they have spotted him but are too far away, they will chase him. If they all lose sight of him after having previously seen him. They will enter a search pattern. They will search 2 randomly assigned locations on the map until they have been searching for 40 seconds. Then they should return to their patrol paths. If they find the player whilst searching, they should return to either chase or attack the player based on the distance between them and the player. | |
| **Test:**  The player will start in an undetected state and ensure that the enemies all follow their patrol paths.  The player will then appear in the vision cone of one enemy and be subsequently spotted by the enemy.  The player will wait there until all of the guards have traversed to the player’s location.  The player will then slowly walk around the map. The tester will ensure that all of the guards attack the player when they are close and chase the player when they are far away.  The player will then go into hiding.  The tester will ensure that all guards enter their search pattern.  During the searching, the player will then attempt o be spotted again by the guards to ensure they switch states from searching to either attacking or chasing.  The player will then go back into hiding.  The tester will ensure once again that the guards enter their search pattern.  The tester will then observe if the guards will return to their patrol paths after 40 seconds. | |
| **Test Record:** Expected Results observed | |
| **Date:** 05/05/23 | **Tester:** Tayyab Hussain |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case:** UI Depicting Enemy States | **ID:** 2A |
| **Test Number:** 4 | |
| **Objective:**  To test whether the UI elements depicting the enemy`s state are working correctly based on which state the guard is currently in. | |
| **Set up:**  The player will play the game as normal ensuring to go through all 4 states for each enemy and checking if the UI is correctly depicting each guard’s state. | |
| **Expected Results:**  The enemies should not have a sprite above their heads when patrolling.  The enemies should have an exclamation mark sprite over their heads when the player has been spotted.  The enemies should have a question mark sprite above their heads when in a search pattern.  The enemies should return to having no UI element above their heads when they return to a patrolling state after the search is complete. | |
| **Test:**  The player will start the game outside of the map. The tester will ensure the guards have no UI elements above their heads while patrolling.  The player will then be put inside the map and try to be detected by a guard.  The tester should only see an exclamation mark over the guard’s head when the player has been completely spotted by the guard.  The player will then go into hiding and the guards should enter a search pattern.  During this search pattern, the tester should check that the guards no longer display an exclamation mark sprite over their heads and now display a question mark sprite.  The player will attempt to be spotted again by a guard while the search is ongoing to check if the UI elements can change back from a question mark sprite to an exclamation mark sprite.  The player will then leave the map entirely and the tester will observe if the guards all change their UI elements to a question mark sprite while in a search pattern.  Once the search pattern has concluded, the tester will observe if the guards have returned to patrolling and no longer have any UI element above their heads. | |
| **Test Record:**  Expected Results observed | |
| **Date:** 08/05/23 | **Tester:** Tayyab Hussain |
| **Result:** Passed | |