//////////////////////////////////////////////////////////////////////////////////////////////////////////

**TEMPLATE**

|  |  |
| --- | --- |
| **Use Case:** | **ID:** |
| **Test Number:** | |
| **Objective:** | |
| **Set up:** | |
| **Expected Results:** | |
| **Test:** | |
| **Test Record:** | |
| **Date:** | **Tester:** |
| **Result:** | |

**TEMPLATE**

//////////////////////////////////////////////////////////////////////////////////////////////////////////

|  |  |
| --- | --- |
| **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 enemy’s five vision zones. The player will start outside 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:**  After spending 1 second in zone 1, the player should be spotted, and the enemy`s spotlight should go red.  After spending 1.5 seconds in zone 2, the player should be spotted, and the enemy`s spotlight should go magenta.  After spending 2 seconds in zone 3, the player should be spotted, and the enemy`s spotlight should go yellow.  After spending 3 seconds in zone 4, the player should be spotted, and the enemy`s spotlight should go green.  After spending 5 seconds in zone 5, the player should be spotted, 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, 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, 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, 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, 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, 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 succeed in independent pathfinding to search for the player. | |
| **Set up:**  The enemies should be going along their patrol paths, as usual, 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 pathfind to search for the player. | |
| **Expected Results:**  The enemies should all congregate at the player’s last known location. They should then spend 20 seconds searching a pre-determined position on the map using the pathfinding algorithm to traverse to that location. After 20 seconds, they should then pathfind to another location on the map and search there for 15 seconds. After 40 total seconds of searching(accounting for 5 seconds at the start to get to the player’s last known location), 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, hide, and be outside 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 four 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 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 four different actions within the game and use them at the correct time. | |
| **Expected Results:**  The enemies should patrol when the player has yet to be 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. They will chase him if they have spotted him but are too far away. If they all lose sight of him after having previously seen him. They will enter a search pattern. They will search two 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 1 enemy and be subsequently spotted by the enemy.  The player will wait until all the guards have traversed to the player’s location.  The player will then slowly walk around the map. The tester will ensure that all 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 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 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 the current state of the guard. | |
| **Set up:**  The player will play the game normally, going through all four states for each enemy and checking if the UI correctly depicts 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 guard has completely spotted the player.  The player will then hide, 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 | |

|  |  |
| --- | --- |
| **Use Case:** UI Depicting Enemy Detection Level | **ID:** 2B |
| **Test Number:** 5 | |
| **Objective:**  To test whether the UI correctly depicts how close the enemy is to detecting the player. | |
| **Set up:**  The test version of the game will have only one enemy on the map, and the enemy will not be able to move or look around. This will make it easier to check the detection level. | |
| **Expected Results:**  The enemy`s detection bar should not appear above their head until the player walks into the enemy`s vision zone.  The enemy will then have a detection bar above their head, which should slowly start filling up at the same rate the enemy would detect the player.  Once the enemy`s detection bar is full and the enemy detects the player, the detection bar should disappear, and the Alerted UI element should appear instead. | |
| **Test:**  The player will start outside of the enemy`s vision cone.  The player will then walk into the vision cone, and the tester should ensure the detection bar appears above the enemy and starts to fill up.  The player will stay in the zone until the bar is half full and then step out of the zone. The tester should ensure that the bar starts to decrease slowly, and the bar should disappear once empty.  The player will then re-enter the enemy`s vision zone and stay in the zone until the bar is full. The tester should observe that the bar disappears and the Alerted UI element appears above the enemy`s head. | |
| **Test Record:**  The test was largely successful. However, there was one minor pitfall. When the player leaves the enemy’s vision zone, the bar will initially jump down before slowly decreasing. This is due to the way the bar`s detection amount is coded. This is discussed further in the Results section. | |
| **Date:** 11/05/23 | **Tester:** Tayyab Hussain |
| **Result:** Passed, but a slight adjustment is needed to make the feature more comprehensive | |

|  |  |
| --- | --- |
| **Use Case:** Smoke Bomb Mechanic | **ID:** 3Ai |
| **Test Number:** 6 | |
| **Objective:**  To test whether the smoke bomb deploys correctly, obstructs the view of the Enemies, stops the enemies from moving when inside the smoke and is correctly detected by an enemy without the player’s presence. | |
| **Set up:**  To test the smoke bomb, a test version of the game will be set up with only the player and a test enemy who cannot move. This is so that the tester can verify whether the enemy can see the player through the smoke and verify the basics, such as the deployment of the smoke and the visuals.  Then the normal game will be played to check the enemies` reaction to the smoke. | |
| **Expected Results:**  When the player hits the ‘G’ key on the keyboard, the smoke bomb canister will spawn in front of them and drop to the ground.  After a 1-second delay, the smoke should be instantiated, and the canister should be culled.  The enemies will not be able to see the player if the smoke obstructs them.  The enemies will not move if they are caught in the smoke.  The enemies will search for the player if they cannot find them after the smoke clears.  If a smoke bomb appears, but the player has yet to be spotted by the enemies, the enemies should move towards the smoke but not go in the smoke and search for the player after the smoke dissipates. | |
| **Test:**  The player will start in the test version and be spotted by the test enemy.  The player will then press ‘G’ and drop the smoke bomb.  The tester will ensure the canister will behave as a physics object and drop to the ground and have a 1-second delay before being culled from the level.  The tester will also observe the smoke being instantiated at the location of the now-culled canister.  As the smoke billows, the tester will observe the enemy and ensure that the enemy can no longer see the player when the smoke obstructs the player.  Once the smoke clears, the tester will ensure that the enemy can now see and spot the player.  Then the normal version of the game will be played.  The player will be spotted by all of the enemies and drop the smoke.  The tester will ensure that the enemies do not move while in the smoke.  The player will leave the map, and the tester will ensure that the enemies start searching for the player and complete a full search pattern before returning to their patrol paths.  The player will then drop smoke in front of one of the enemies on the level without being spotted.  The tester will ensure that the enemy spots the smoke even without seeing the player and moves all of the enemies to the location of the smoke.  The tester will ensure that when the smoke dissipates that the enemies enter a searching pattern again. | |
| **Test Record:**  The tests related to the smoke itself passed. This includes the smoke being released from the canister after a delay and the canister being culled. The smoke also blocks the enemy’s vision, and the enemies cannot spot the player through the smoke, which works as intended.  However, the remaining parts of the test all failed. This was predicted, as they were not implemented successfully into the build. The enemies did not spot the smoke, they only saw it as an obstacle in their way, and therefore they did not stop when inside the smoke or move to the smoke when they spotted it without seeing the player. This will be explored further in the Results section. | |
| **Date:** 20/05/23 | **Tester:** Tayyab Hussain |
| **Result:** Failed | |

|  |  |
| --- | --- |
| **Use Case:** Hiding Mechanic | **ID:** 3Aii |
| **Test Number:** 7 | |
| **Objective:**  To test whether the player can be spotted when by the enemy when the player is inside a bush | |
| **Set up:**  A test version of the game will be set up. This will include the player, the bush and a test enemy. | |
| **Expected Results:**  When the player stands next to but not in the bush, the enemy should be able to spot the player, and the enemy`s UI symbols should reflect this.  The enemy should lose sight of the player when the player enters the bush.  The enemy should then move to where the player was last spotted.  The enemy should enter a search pattern as usual if the player cannot be found. | |
| **Test:**  The player will begin next to the bush and let the enemy spot them.  Once the enemy has spotted the player, they will move into the bush.  The tester will ensure that the UI elements above the enemy`s head indicate that the enemy has lost sight of the player.  The tester should ensure the enemy moves towards the player’s last known location but cannot spot the player through the bush.  The player will remain in the bush, and when the enemy is also in the bush, the tester will ensure that the enemy has spotted the player. | |
| **Test Record:** Expected Results Observed | |
| **Date:** 21/05/23 | **Tester:** Tayyab Hussain |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case:**  Limiting Use of Smoke Bomb Mechanic | **ID:** 3Bi |
| **Test Number:** 8 | |
| **Objective:**  To test whether the smoke bombs have a finite use and ensure they cannot be used repeatedly in quick succession. | |
| **Set up:**  A test version of the game will be set up with just the player and no enemies. | |
| **Expected Results:**  When the player presses the button for the smoke bomb, a smoke bomb will deploy.  There will then be a delay before the player can drop another smoke bomb.  The player also should only be able to drop a specified number of smoke bombs. | |
| **Test:**  The player will drop the first smoke bomb.  After the first smoke bomb is dropped, the player will continue to press the button to drop another smoke bomb.  The tester should ensure that no more smoke bombs are dropped until the specified cooldown timer for the smoke bomb has ended.  Once the timer has ended, the player will be able to drop another smoke bomb.  They will continue to press the smoke bomb button continuously.  The tester will ensure that when the specified number of smoke bombs have been dropped, no more smoke bombs are dropped by the player, despite them still pressing the smoke bomb button. | |
| **Test Record:** Expected Results Observed | |
| **Date:** 22/05/23 | **Tester:** Tayyab Hussain |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case:**  Limiting Use of Hiding Mechanic | **ID:** 3Bii |
| **Test Number:** 9 | |
| **Objective:**  To test whether the player can rely solely on the game's hiding mechanic. | |
| **Set up:**  The normal game will begin | |
| **Expected Results:**  The player will be able to hide in several locations on the outskirts of the map.  Once the player gets closer to the centre of the map, the number of hiding locations will reduce significantly. | |
| **Test:**  The player will begin by going to each hiding area on the map’s outskirts.  The player will then go closer to the level’s main part and utilise all the hiding areas.  The tester will ensure that there are significantly fewer places for the player to hide in the main area of the level as opposed to the outskirts. | |
| **Test Record:** Expected Results Observed | |
| **Date:** 22/05/23 | **Tester:** Tayyab Hussain |
| **Result:** Passed | |

|  |  |
| --- | --- |
| **Use Case:** Unique Traversal | **ID:** 4a |
| **Test Number:** 10 | |
| **Objective:**  To test whether the player can successfully use the zipline as intended | |
| **Set up:**  The player will start positioned within 1 meter of the start of the zipline | |
| **Expected Results:**  The player will be transported down the zipline at a constant speed when they press the correct button on the keyboard.  The player will not be able to move around while on the zipline  The player will, however, be able to look around when on the zipline  When the player lands on the other side, they should be able to move freely again  The player should not be able o use the zipline backwards(going uphill) | |
| **Test:**  The player will first press the zipline button on the keyboard  When travelling down the zipline, the player will then try to use every movement control in the game, such as walking, sprinting, jumping and crouching  Before the player reaches the end of the zipline, they will check that moving the mouse still allows them to look around wherever they want while on the zipline  When the player lands, they should recheck every control and ensure that they can now move freely  Lastly, The player will ensure that they cannot use the zipline back to where they came from | |
| **Test Record:** Expected Results Observed. Despite this, the zipline does not look very aesthetically pleasing at this point. The rope on the zipline does not connect properly, and the line is not straight, so the player phases through the line when getting closer to the bottom of the zipline. The zipline hook point could also do with a texture applied along with the ropes. However, functionally, the feature works fine. | |
| **Date:** 06/06/23 | **Tester:** Tayyab Hussain |
| **Result:** Passed. More polish is required. | |