[Real Time Tactics]

Technical Design Document

Kieran Cooksley – C018249L

Contents

Project Introduction

The project is a technical prototype for a Real Time Tactics (RTT) game. The prototype will feature AI characters which can be selected by the player and controlled by giving in-game commands. The prototype will also include AI Enemies that can perceive and respond to visual & auditory stimuli and make behavioural decisions using Behaviour Trees, Blackboards, and Environment Query System (EQS).

Project Goals

The primary goal of the project is the implementation of tactical AI to develop a deeper and more advanced understanding of scripting Artificial Intelligence in Unreal Engine. This may include (but is not limited to) the implementation of Behaviour Trees and EQS to control AI navigation and behaviour and the AI Perception System to perceive and respond to stimuli.

Challenges and Risks

The primary challenge of the project currently is insufficient knowledge and understanding of scripting AI in Unreal Engine. Developing advanced tactical AI for this project will require extensive independent research and testing into the various AI systems and mechanics to create a sufficient AI that satisfies the brief criteria.

Due to the nature and quantity of research required for the success of the project, a lot of unknown variables exist. This means time and project management is fundamental here as more unplanned tasks and challenges may arise during research and development. Scheduling appropriate time for research and testing as well as additional contingency time if things don't go to plan is critical to ensure any bugs or challenges that arise can be resolved sufficiently.

Hardware Requirements

Below are the hardware requirements to run Unreal Engine 5 that is recommended on the Unreal Engine documentation, as well as the hardware specifications for Epic Games' systems as well as the University Lab PCs

	UE recommended*	**Epic Systems****	**University Lab PCs**
Operating System	Windows 10 64-bit	Windows 10 64-bit	Windows 10 64-bit
Processor	Quad Core, >2.5GHz	6 Core Xeon E5-2643 3.4GHz	8 Core i7-11700 2.5GHz
Memory	8GB	64GB	32GB
Graphics	DirectX 11/12 compatible	NVIDIA RTX 2080 super	NVIDIA RTX 3080

***Recommended hardware**

**** Performance Notes**

https://docs.unrealengine.com/5.0/en-US/hardware-and-software-specifications-for-unreal-engine/

Platforms

Target Platform

The prototype will be created for use on PC

Engine Summary & Specifications

Unreal Engine version 5.2 will be used during development of the project.

Refer to Hardware Requirements for a table documenting the recommended specifications to run the engine and prototype.

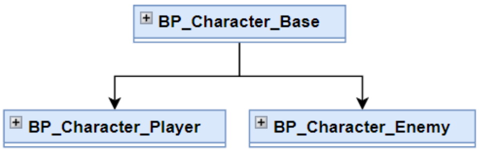
The 'Line of Sight' plugin has been used to implement the Line-of-Sight material and the corresponding Begin & End Overlap events for visual perception.

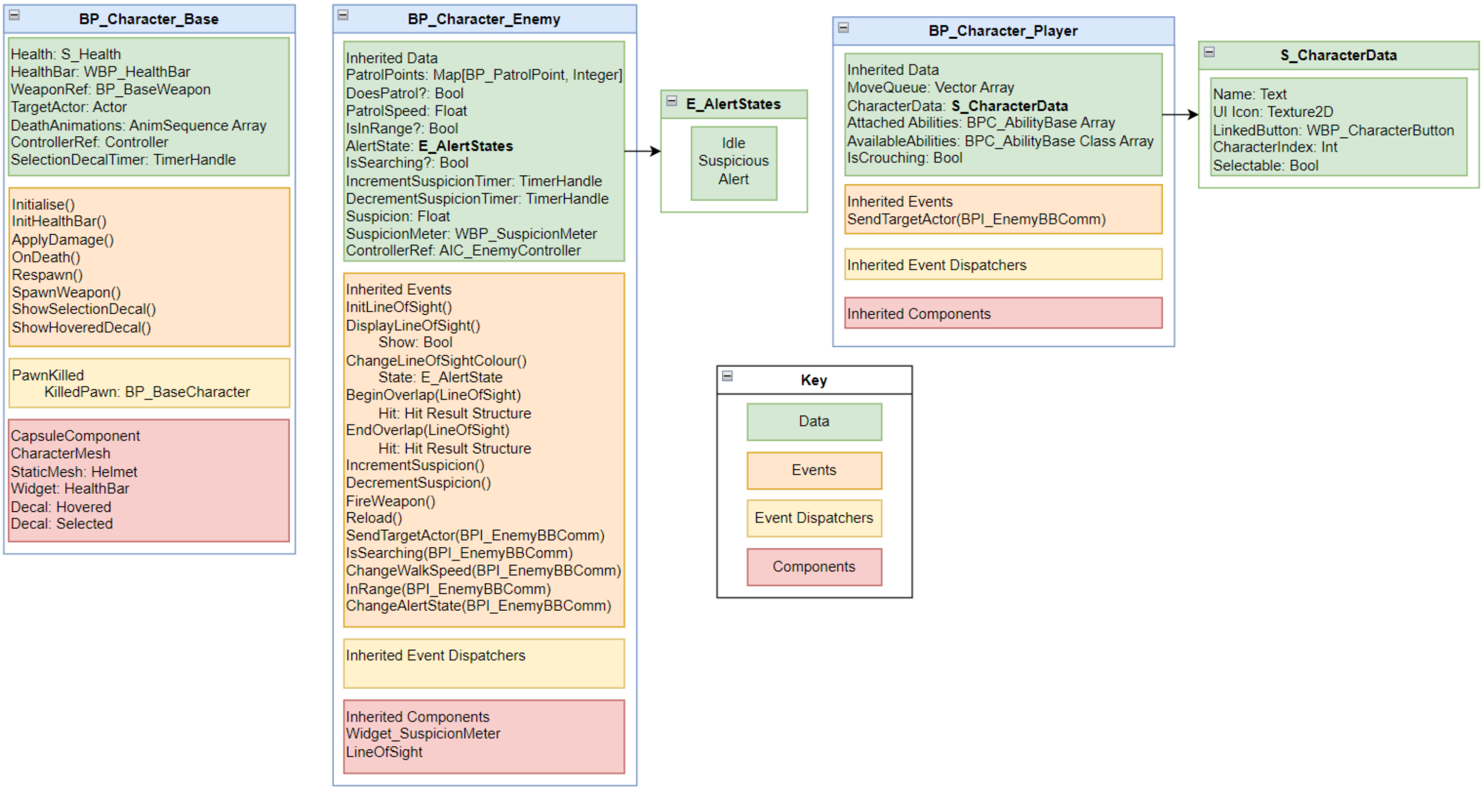
https://www.unrealengine.com/marketplace/en-US/product/line-of-sight

Systems and Diagrams

Character Class Diagram

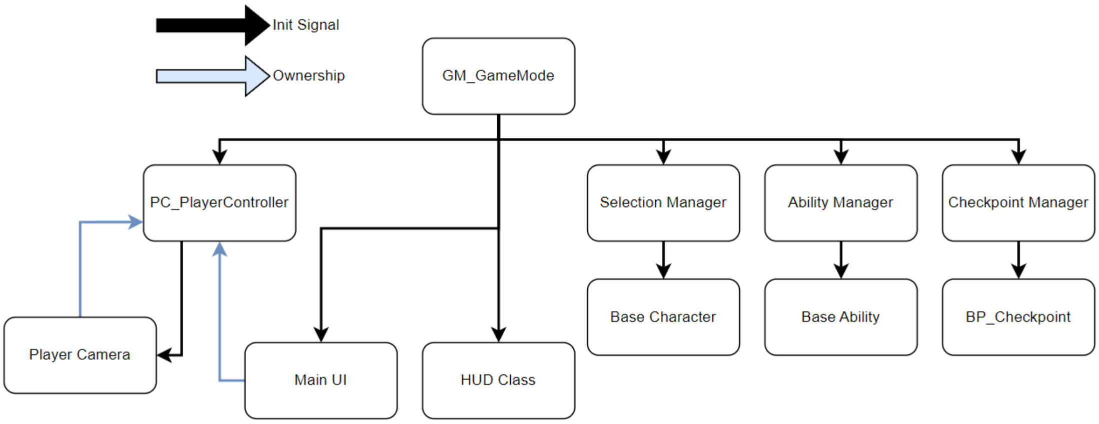
All characters in the game inherit from a Base Character class. Child classes are created for Player & Enemy Characters to store appropriate data and behaviour.





Initialisation Communication Diagram

This Communication Diagram displays how the core components of the game are initialised.



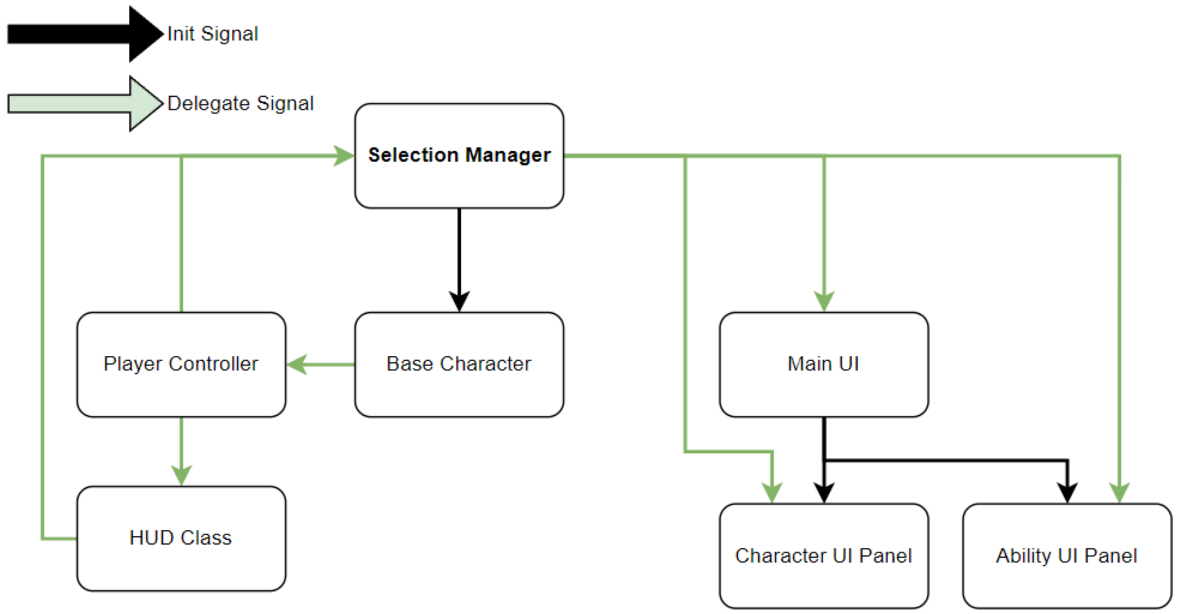
System 1 - Character Selection

This system deals with how characters are selected in the world.

Communication Diagram

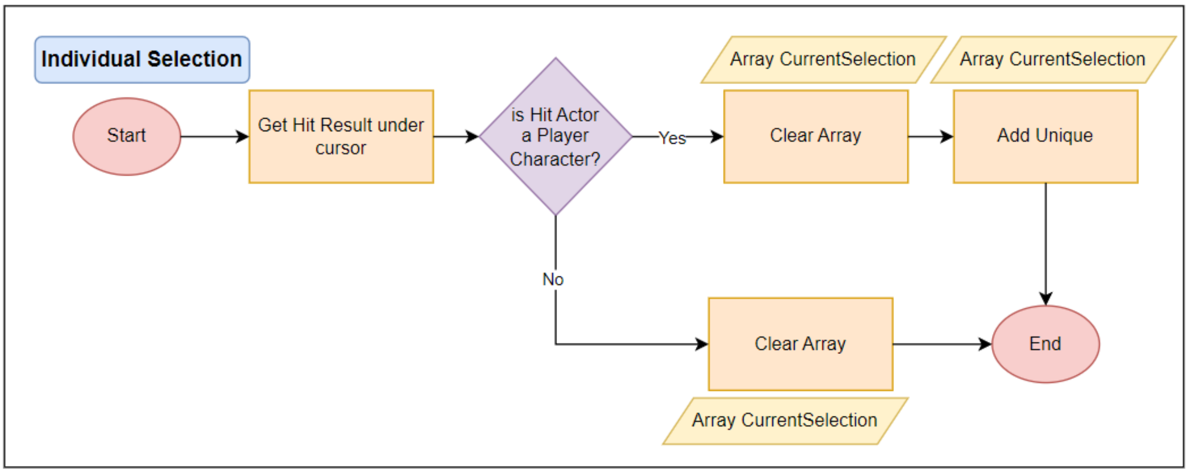
Selecting characters requires communication between several blueprints, visualised below.

The Selection Manager stores much of the System logic, updating the UI and receiving communication from the Player Controller and HUD Class to execute functions.

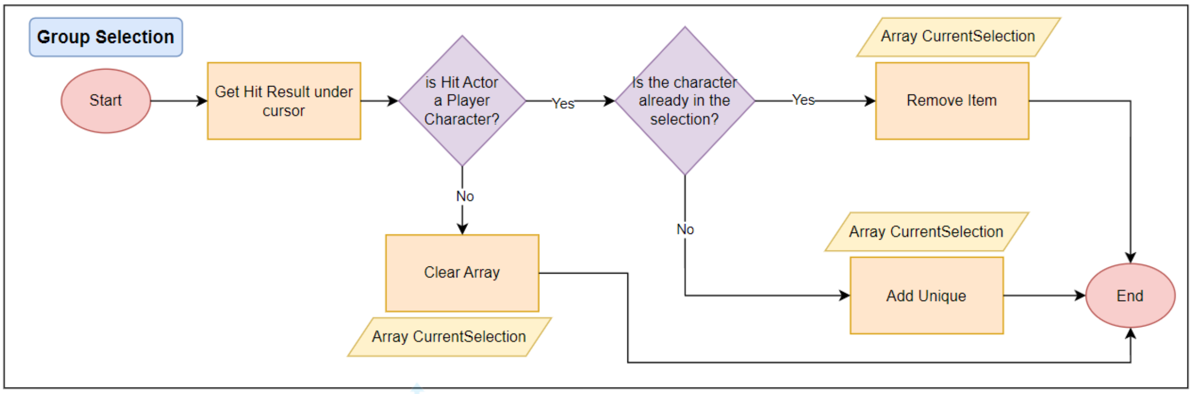


Flowcharts

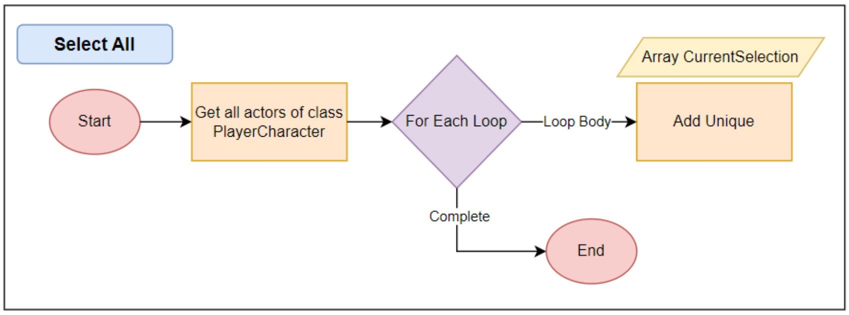
Individual Selection is the standard selection method, allowing the player to select one unit at a time, deselecting any currently selected units first.



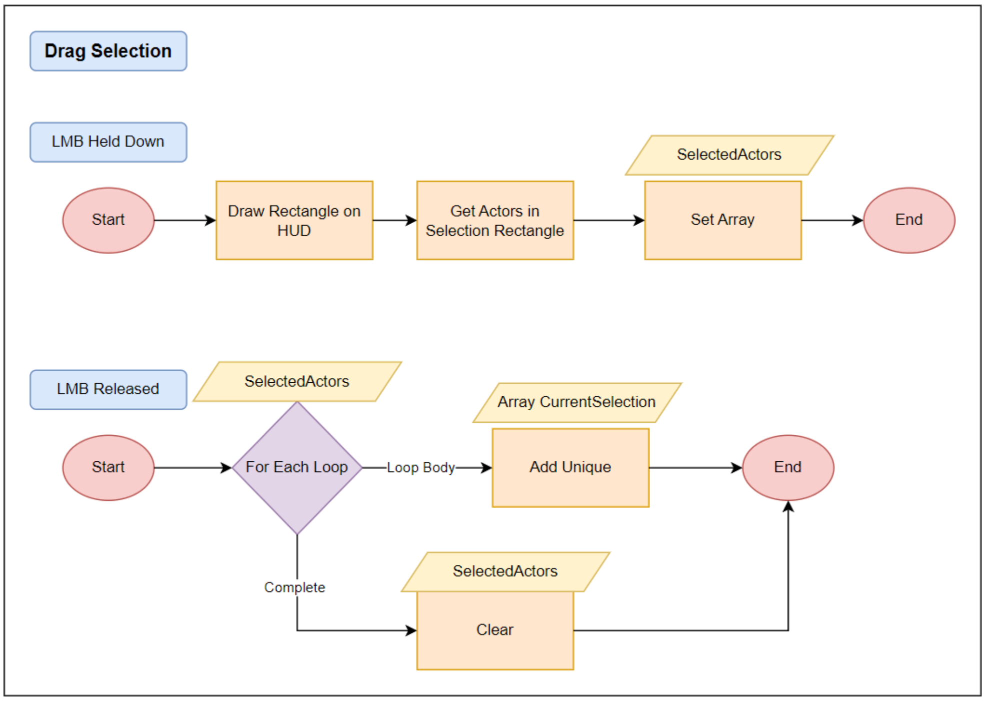
Group Selection allows the player to select multiple units with Shift + LMB. Currently selected units are not removed from the selection when this method is used.



Select All uses a keyboard command to add every controllable character in the world to the current selection.



Lastly, drag select draws a rectangle to the screen, selecting all units within the rectangle when the command is completed.

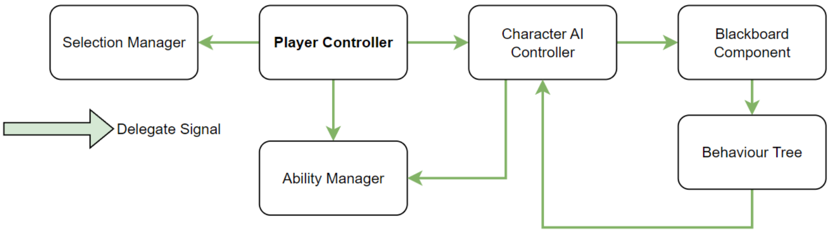


System 2 – Character Movement

This system controls how the player moves the selected units around the world.

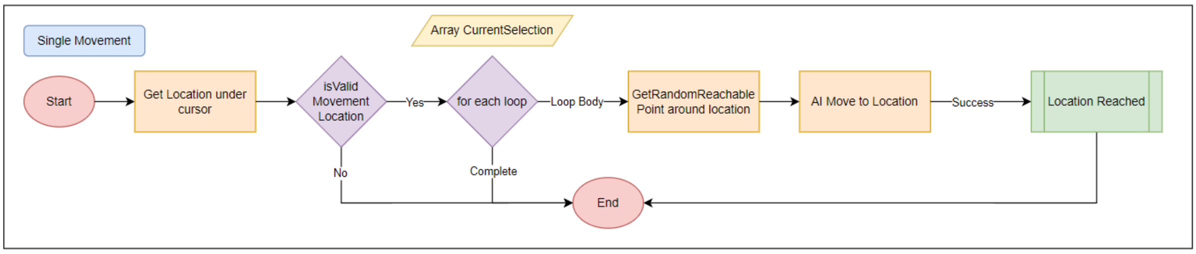
Communication Diagram

Move locations are stored in the Ability Manager which receives Communication from the AI controller. The Controller requests the next move location once the Behaviour Tree has communicated that the current Move Task is complete.

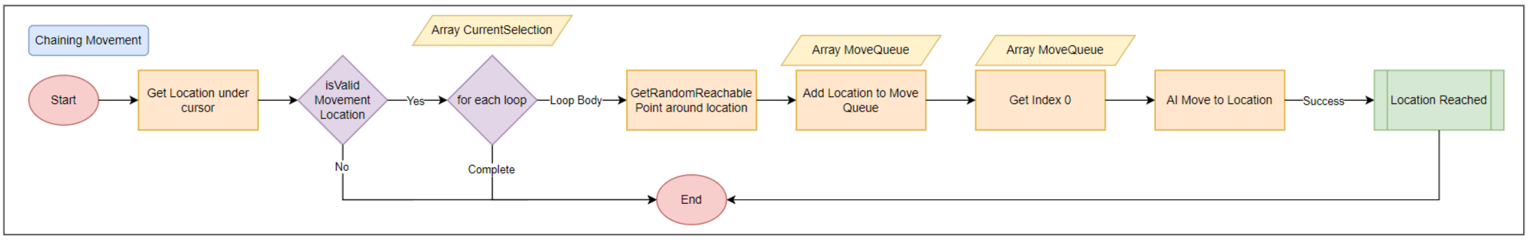


Flowcharts

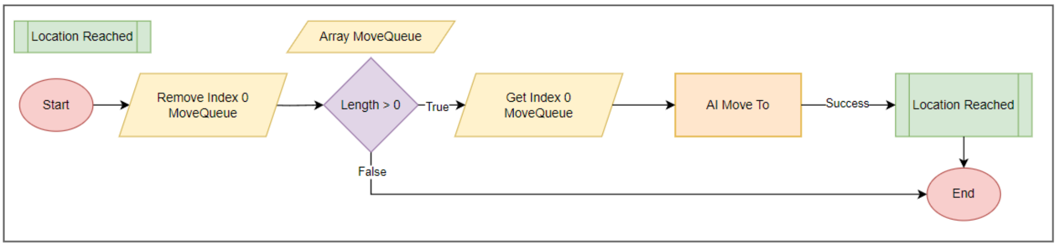
Single Move updates the selected units’ Move To command with the newly clicked location.



Chaining Movement allows the player to add locations to a Move Queue. Units will complete their current move command, then move to the next location in the Move Queue if there is one.



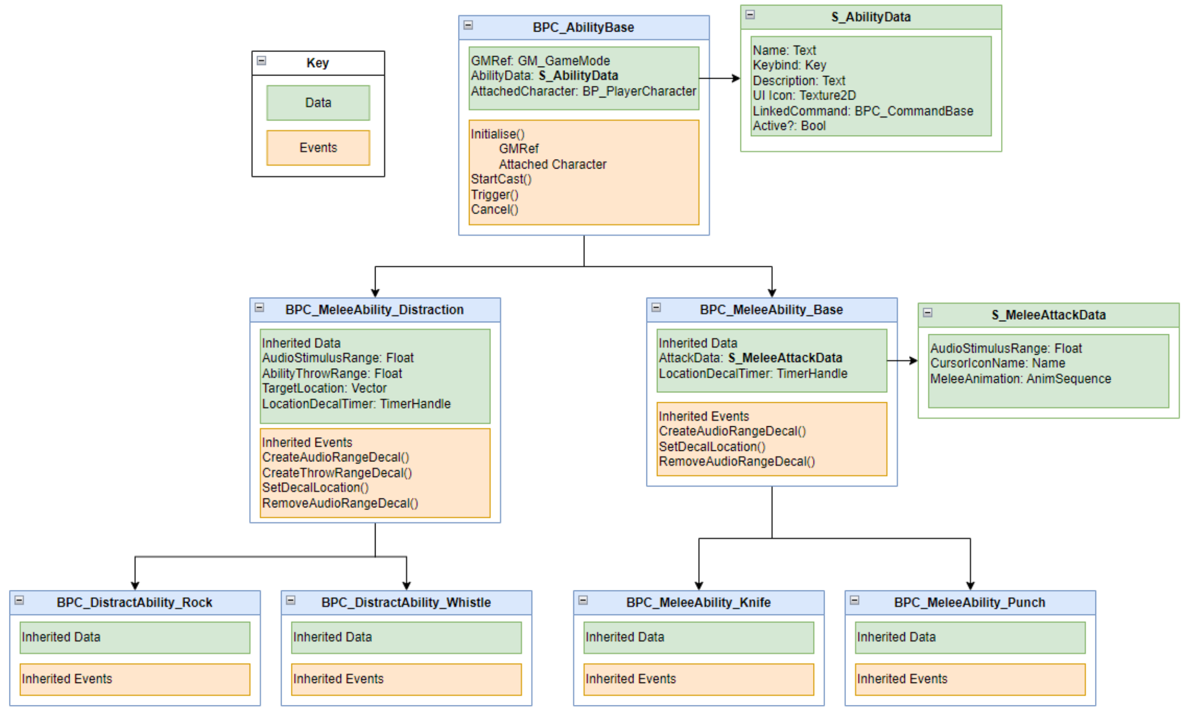
Each of these methods calls a function at the end called ‘Location Reached’. This function removes the current command, allowing the unit to continue movement if another command is present.



System 3 – Character Abilities

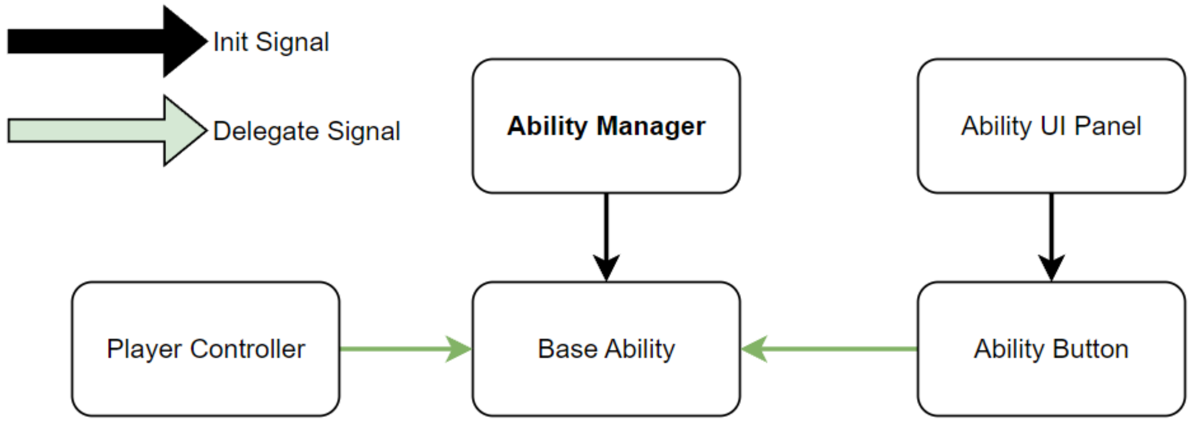
Class Diagram

Abilities are constructed using a hierarchy system with data and behaviour being stored in a parent class and inherited by its Children. The Class Diagram below shows where data and events are stored in each level of inheritance.



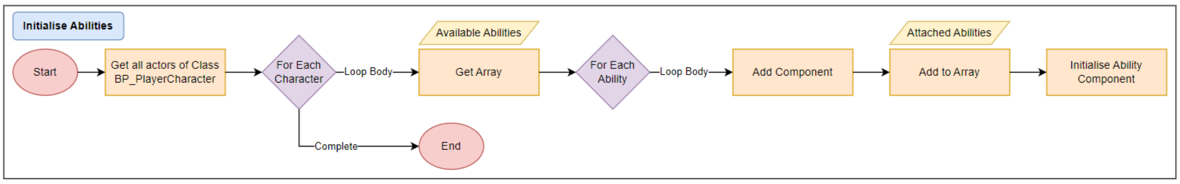
Communication Diagram

Abilities are initialised by the manager when attached to characters. Abilities can be triggered by receiving delegate signals from either the ability button or player controller when the corresponding hotkey has been pressed.



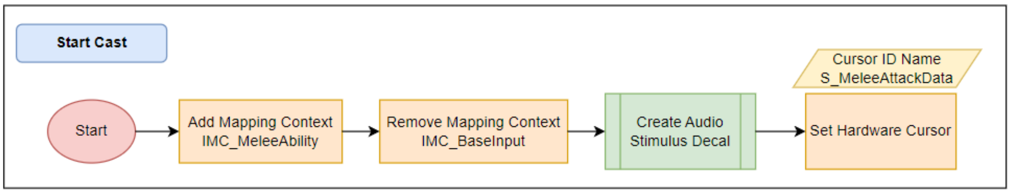
Flowcharts

Initialising the abilities retrieves the available abilities array from each character, adding the corresponding class.

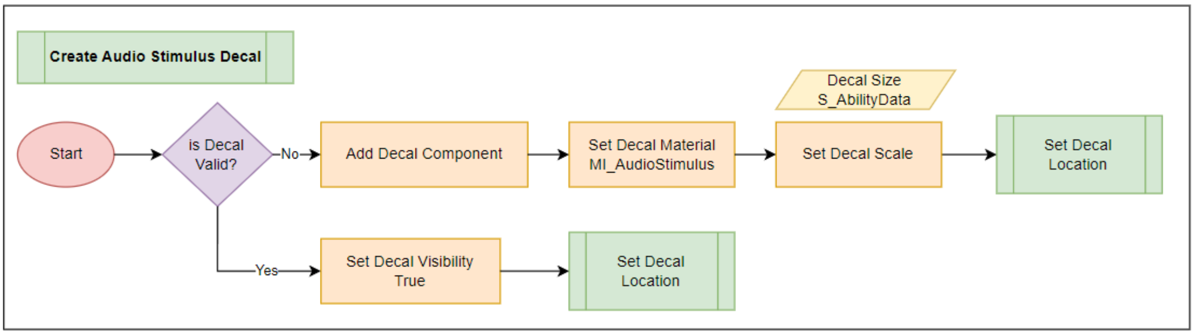


Melee Abilities

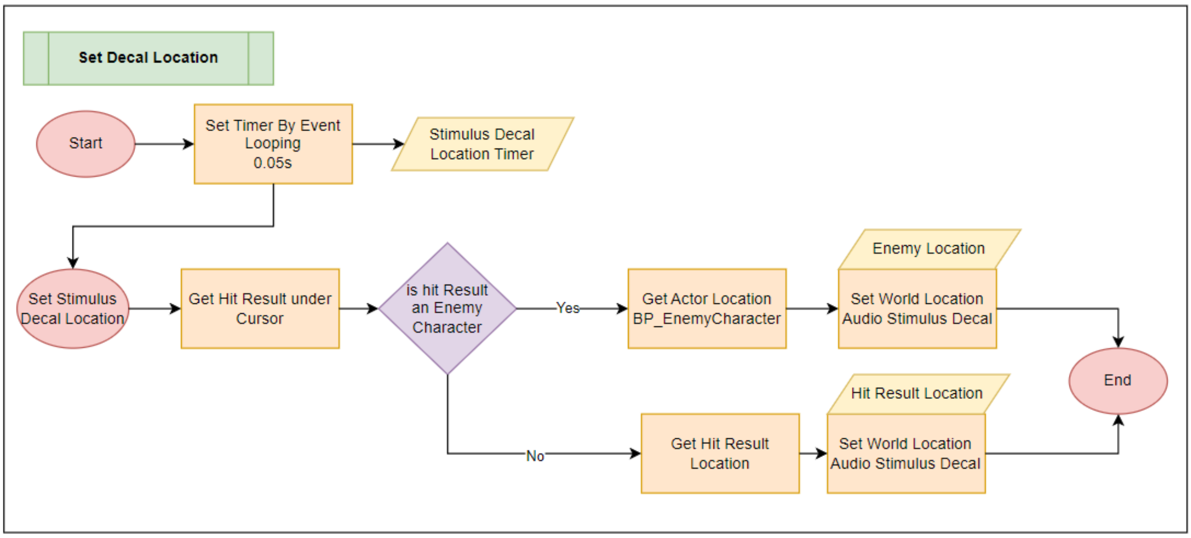
Start cast sets up the ability, adding the ability mapping context and setting the cursor


Start Cast
Start
Add Mapping Context IMC_MeleeAbility
Remove Mapping Context IMC_BaseInput
Create Audio Stimulus Decal
Cursor ID Name S_MeleeAttackData
Set Hardware Cursor

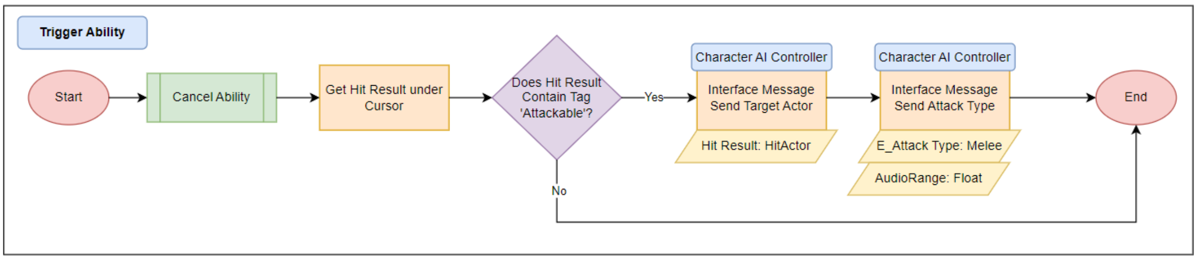
A decal is created to represent the audio stimulus of the ability.



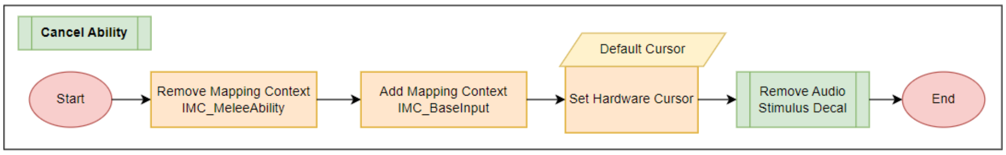
The decal's location is updated on a looping timer to follow the cursor location.

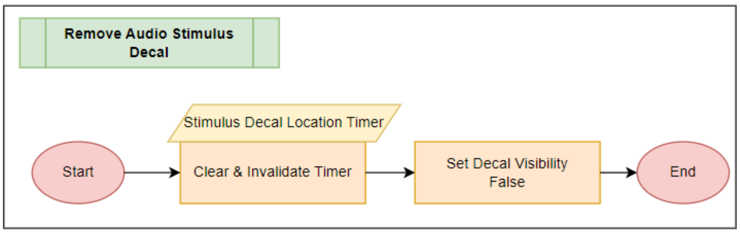

Set Decal Location
Start
Set Timer By Event Looping 0.05s
Stimulus Decal Location Timer
Set Stimulus Decal Location
Get Hit Result under Cursor
is hit Result an Enemy Character
Yes
Get Actor Location BP_EnemyCharacter
Enemy Location
Set World Location Audio Stimulus Decal
End
No
Get Hit Result Location
Hit Result Location
Set World Location Audio Stimulus Decal

When the ability is triggered, a couple of interfaces are sent to the AI Controller.

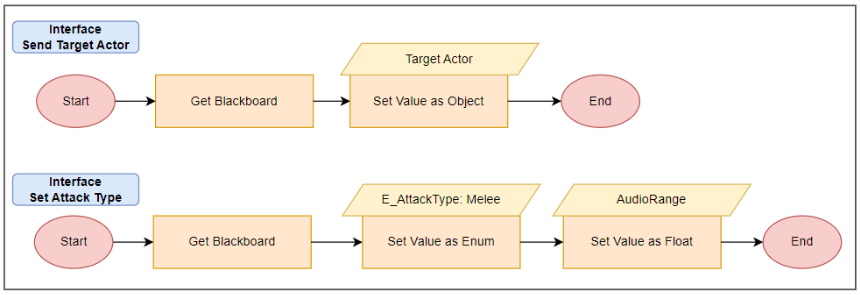


The Cancel Ability function is called on Trigger as it resets the ability, removing the mapping context and hiding the audio decal.

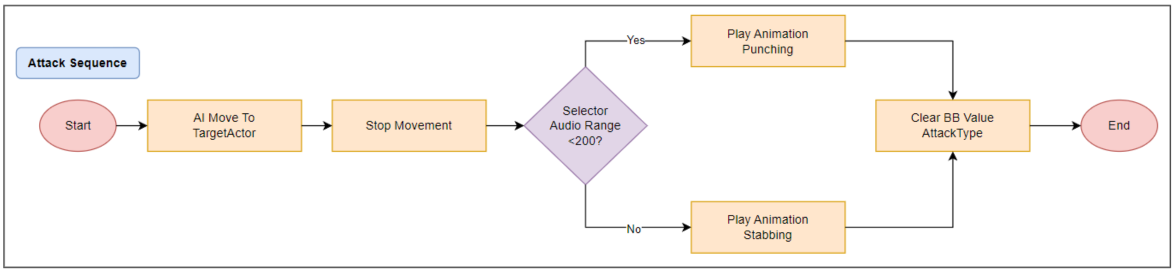

Cancel Ability
Start
Remove Mapping Context IMC_MeleeAbility
Add Mapping Context IMC_BaseInput
Default Cursor
Set Hardware Cursor
Remove Audio Stimulus Decal
End


Remove Audio Stimulus Decal
Start
Stimulus Decal Location Timer
Clear & Invalidate Timer
Set Decal Visibility False
End

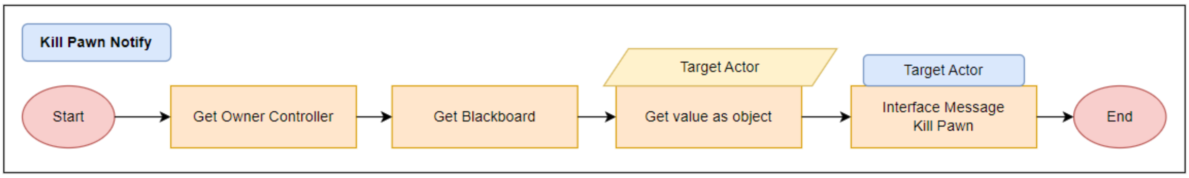
The received interfaces update Blackboard values to trigger behaviour in the Behaviour Tree.



The Attack Sequence moves the character to the target, performing an animation and clearing the attack type key.

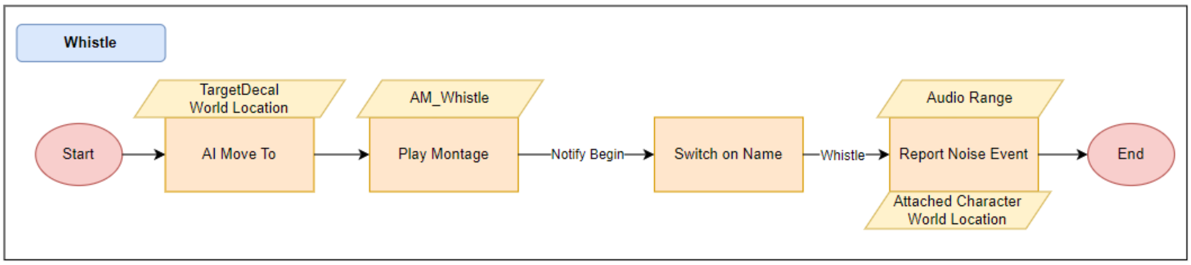


Each animation has a notify which retrieves the Target Actor from the Controller’s Blackboard, sending an Interface message to kill the pawn.

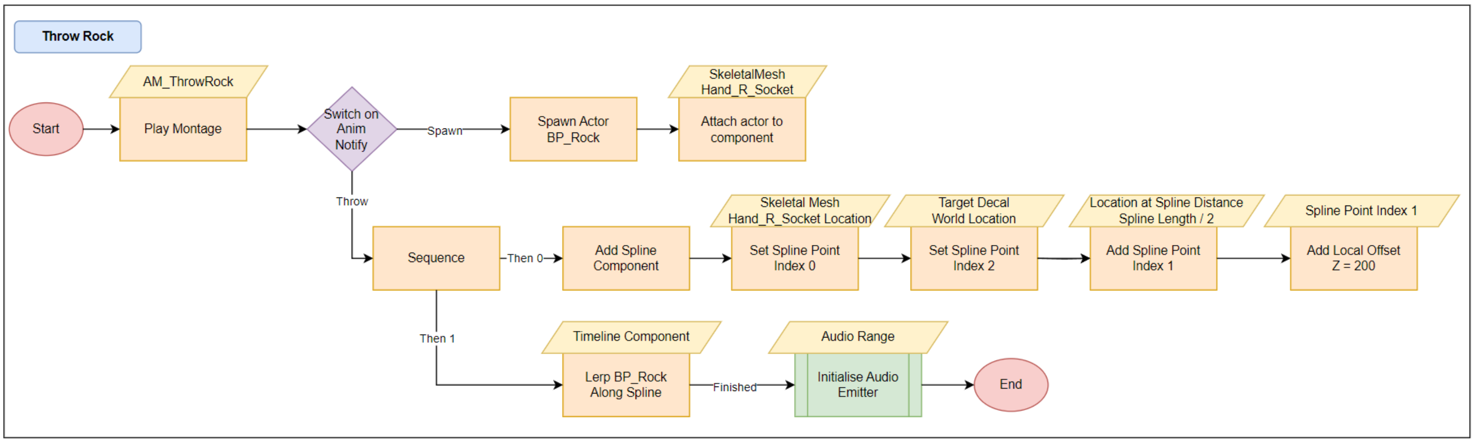


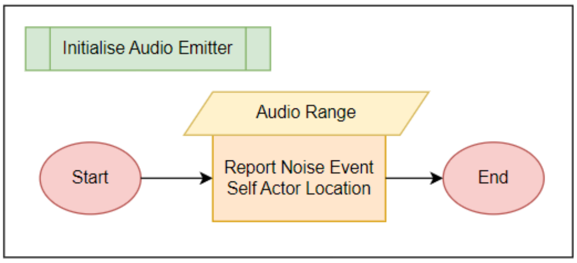
Audio Distraction Abilities

The Whistle is a straightforward audio distraction ability that reports a noise event on a Notify name during an animation montage.



The Rock spawns a rock at the Spawn notify, throwing it at the Throw notify. A spline is generated first, starting at the hand bone and ending at the target location with an arc in the middle. Once generated, the rock is lerped along the spline.



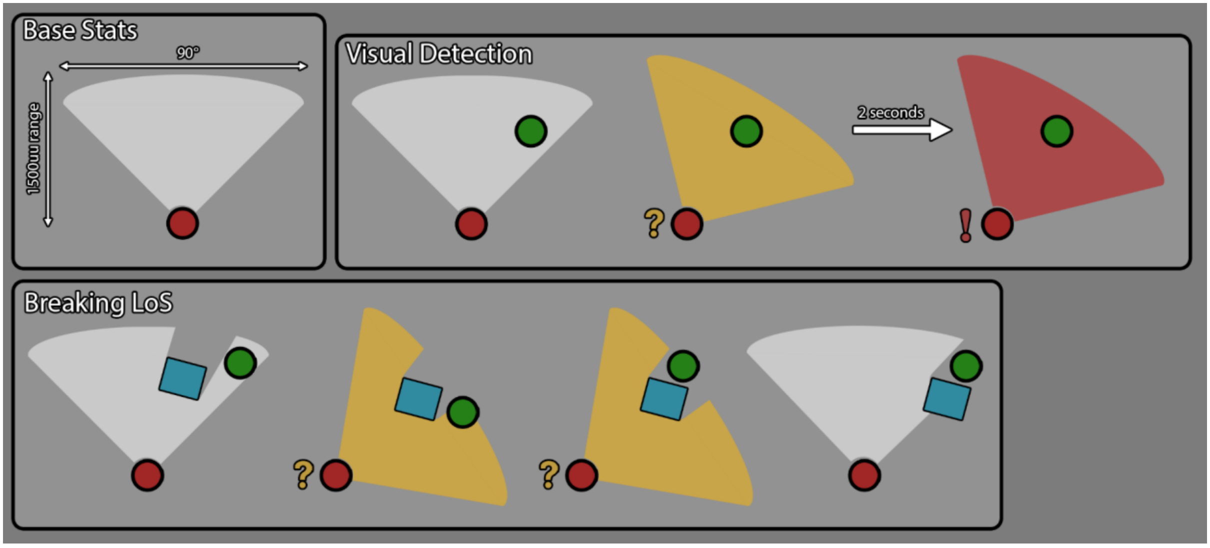


System 4 – AI Behaviour & Perception

Design Diagrams

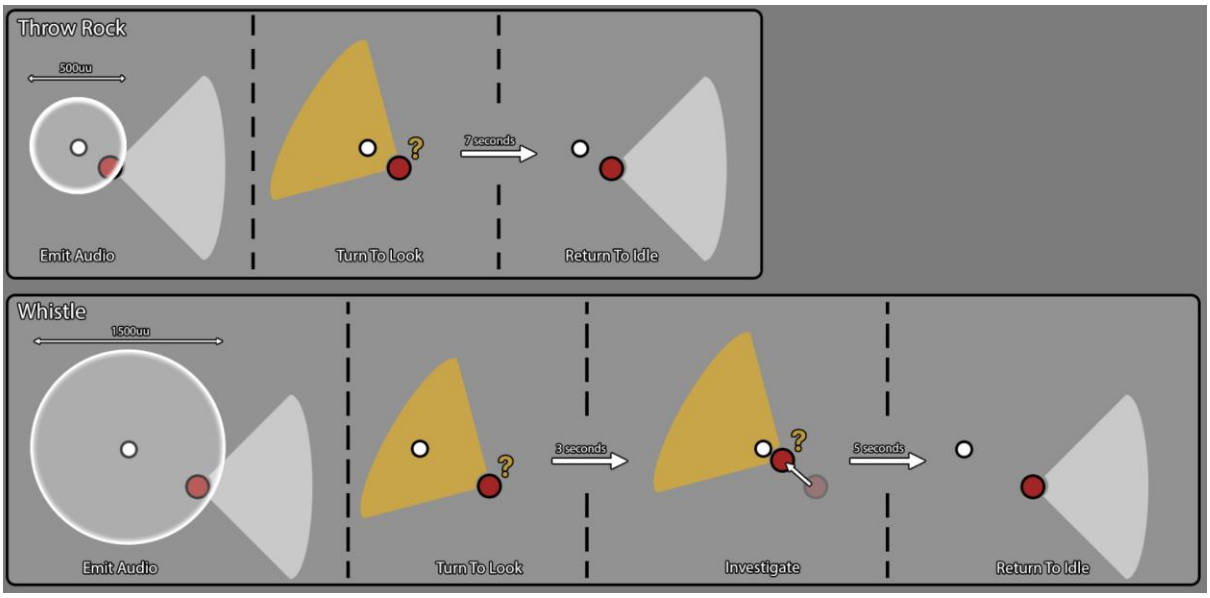
Below are diagrams explaining the Visual and Audio perception systems in the prototype.

Visual Perception



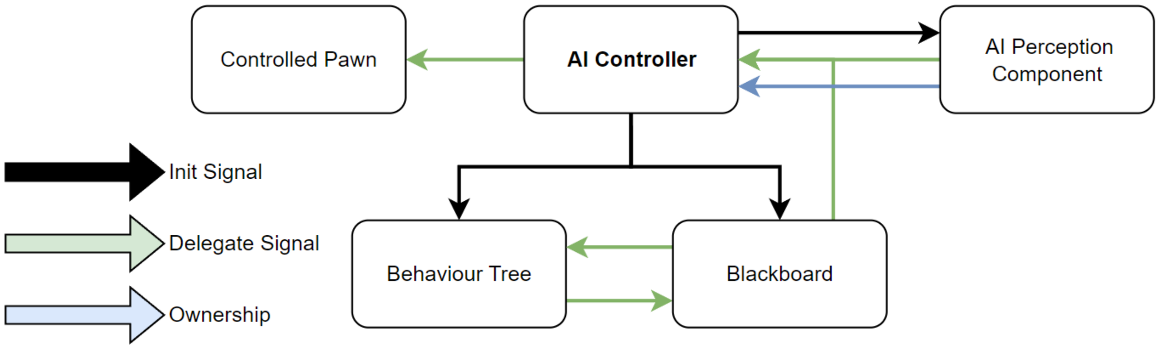
Audio Perception

AI Agents have different behaviours depending on what ability produced the audio. Agents only turn to face the audio generated from throwing a Rock, however will move to investigate the Whistle audio.



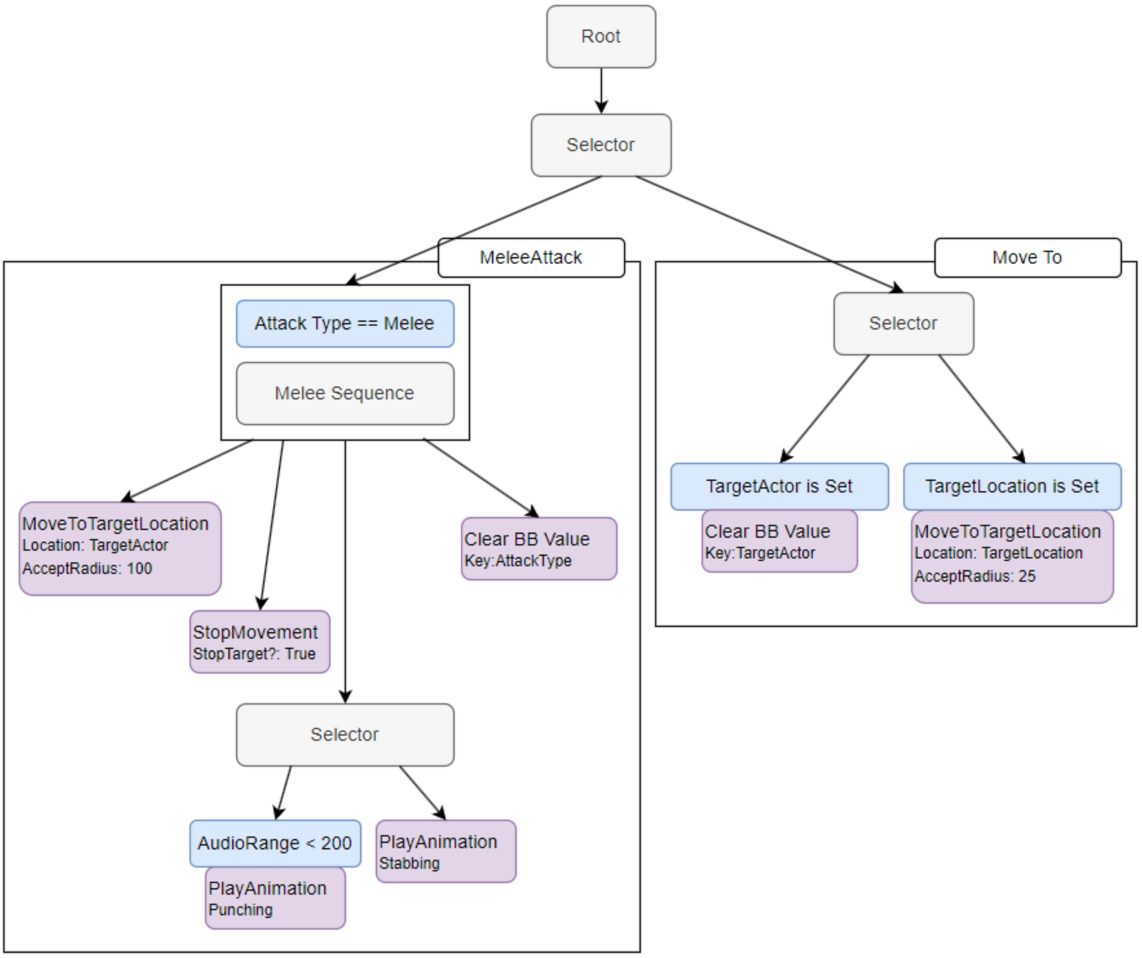
Communication Diagram

The AI Controller acts as a main hub for AI Communication, sending and receiving data from the Behaviour Tree, Blackboard, and Perception Component, which will then send any data to the pawn that it requires.

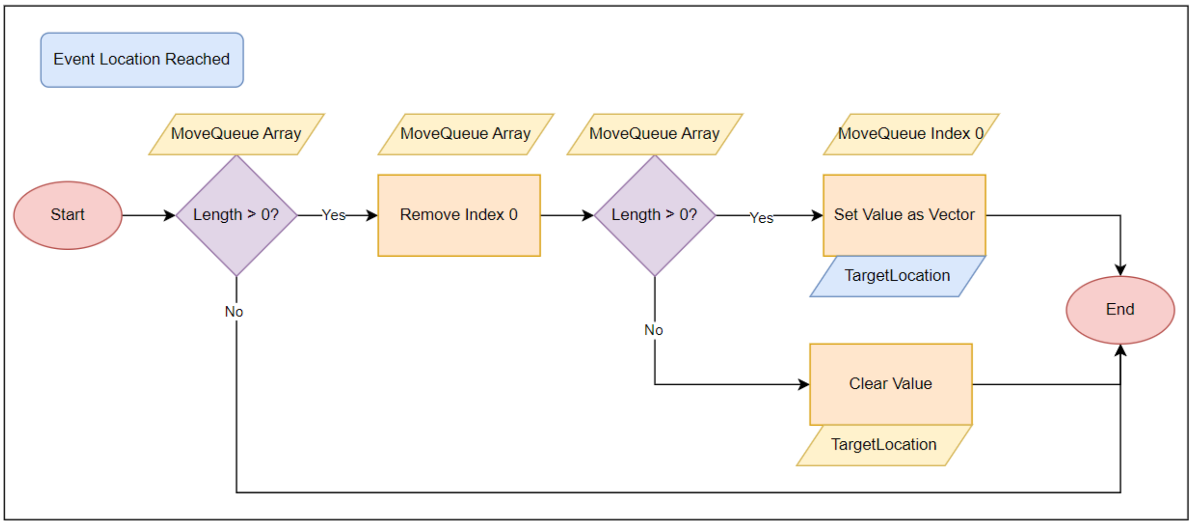


Player AI

The only required sub-trees for the Player Character Behaviour Tree are Melee Attack & Move To Location as the player themselves makes many of the decisions for the characters.



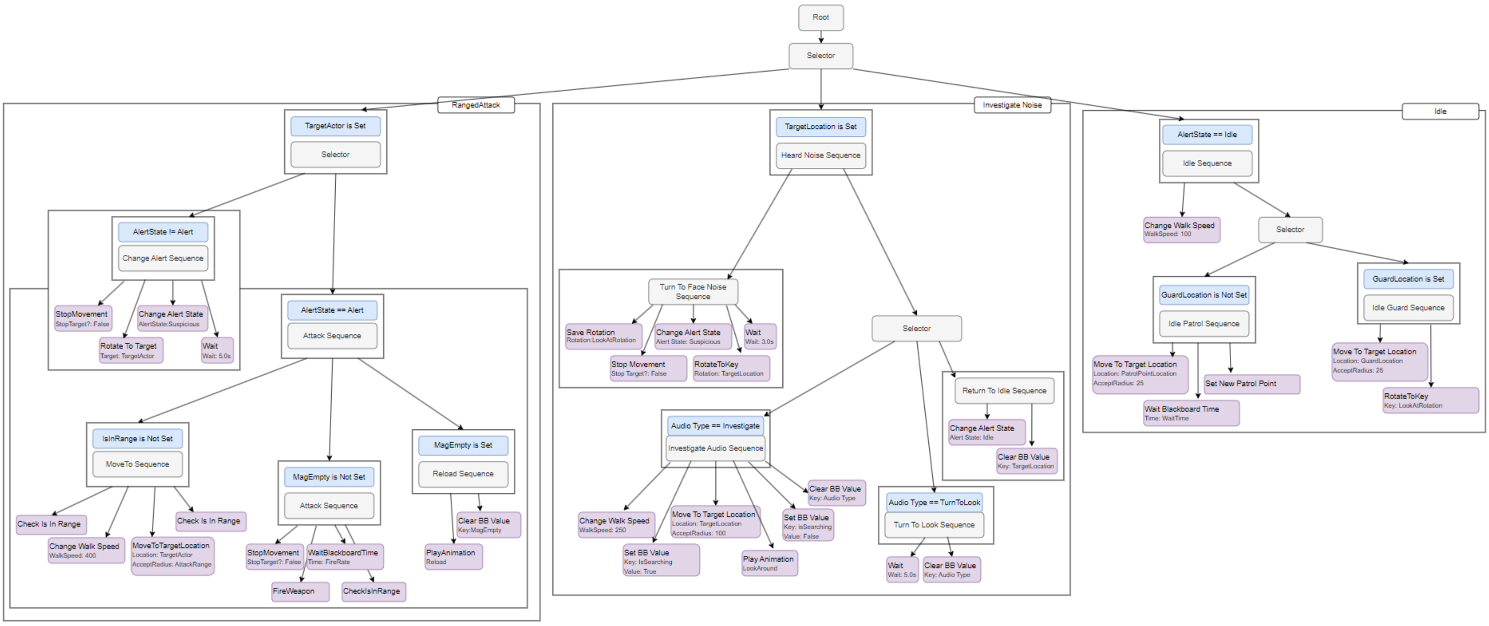
When the AI agent has reached the target location, this function removes it from the array, and checks to see if more locations exist, setting the new location if true or clearing it if false.



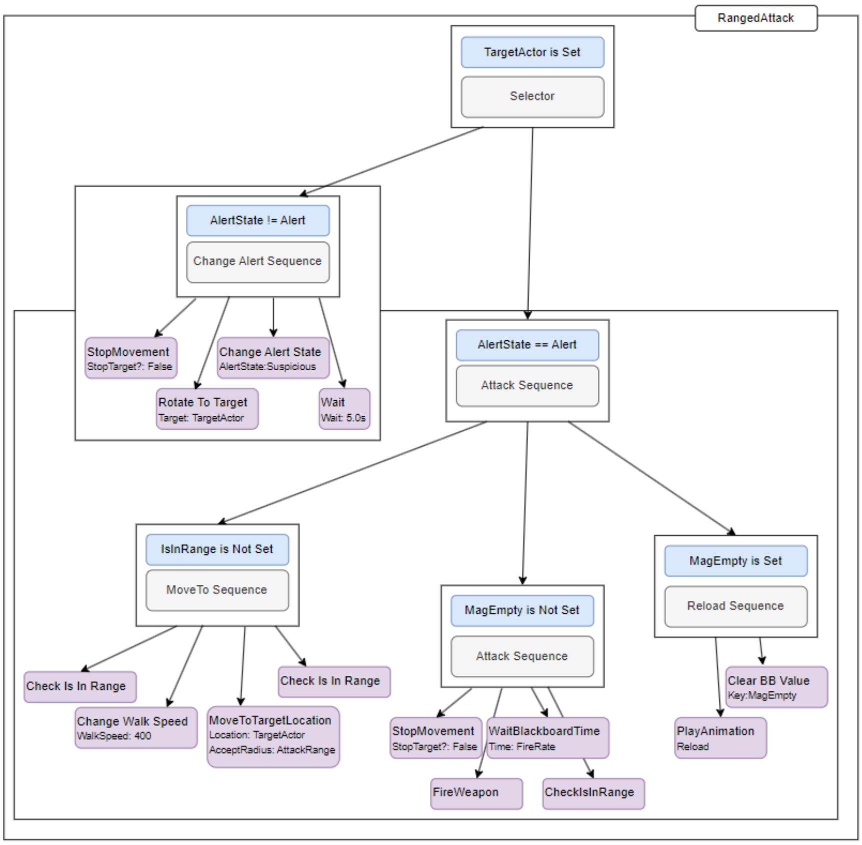
Enemy AI

Behaviour Tree

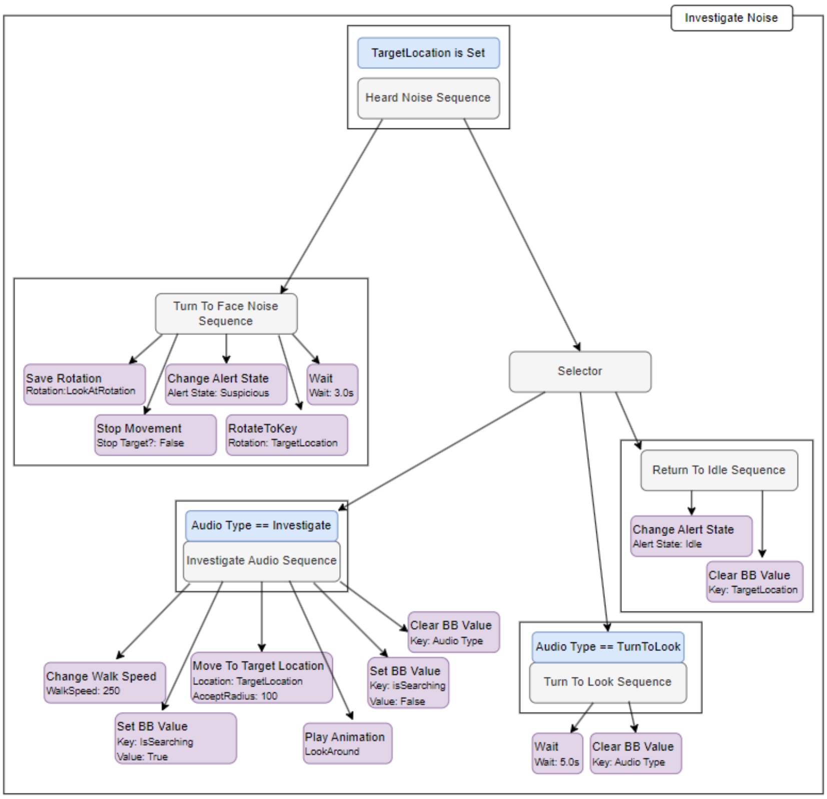
The Enemy AI's Behaviour Tree is visualised below.



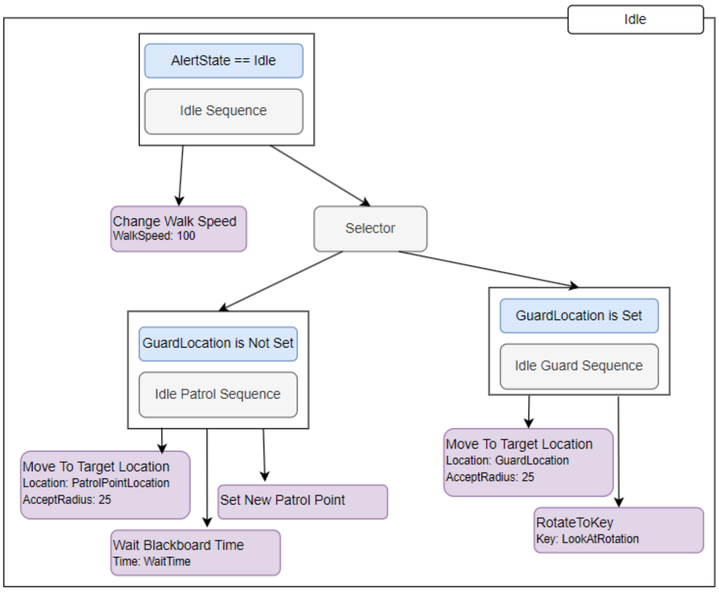
The Ranged Attack sequence is entered if the Target Actor is set and controls how the enemy behaves when attacking a hostile (Player Character)



When a Noise Event is reported in range of an enemy, the target location will be set and this sub-tree will be executed.

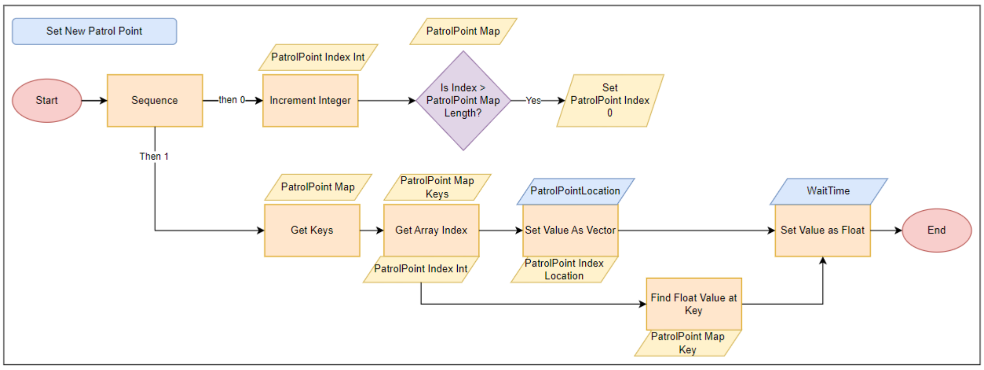


Lastly, the enemy's idle behaviour is dependent on if they are in the state 'patrol' or 'guard'.

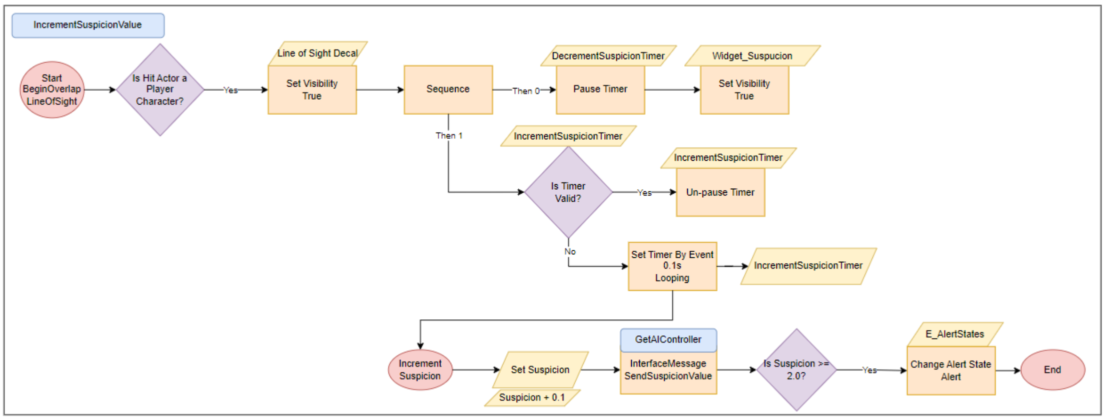


Flowcharts

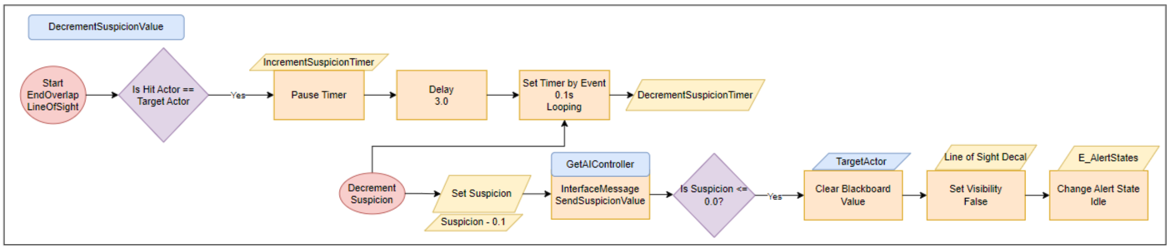
When a patrolling enemy has reached a patrol point, the next point in the map will be selected by incrementing an integer (setting it to 0 if it reached array length).



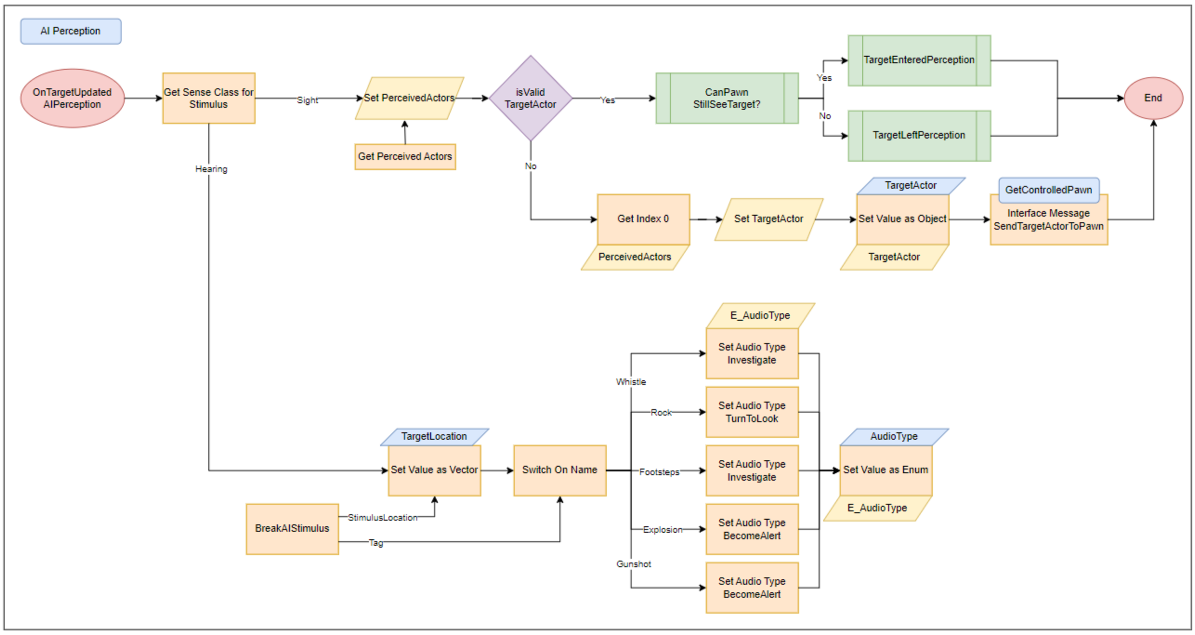
Enemies have a suspicion meter that will increase when a player character is overlapping with the Line-of-Sight material. This increments a float on a timer, changing their alert state if the suspicion value reaches 2.0



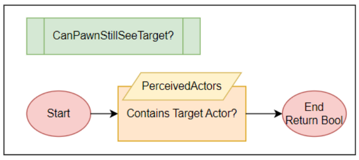
Decrementing the suspicion occurs when players have left line of sight.

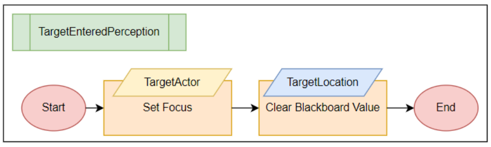


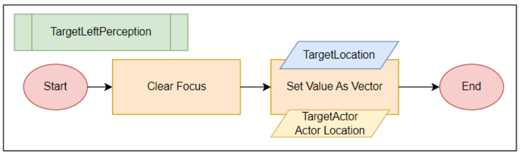
The AI Perception component handles visual and auditory perception. Visual perception will set the perceived actor as the target if a target is not already valid. If it is valid, a check will be performed to see if the agent can still see the target. Audio perception will receive the type of audio perceived and update the Audio Type Enum, affecting how AI agents react to audio stimuli.



The performed checks on visual perception will update the enemies' blackboard values depending on if the Target Actor is still perceived.



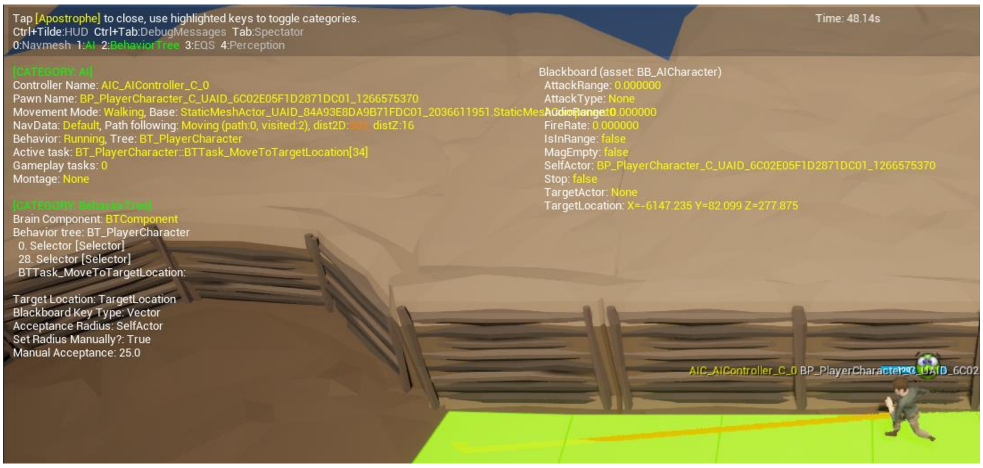




Optimisation and Profiling

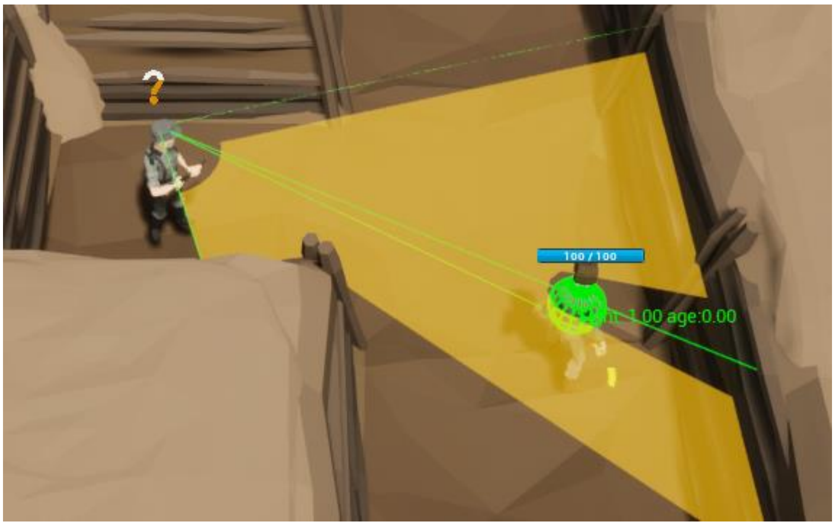
Profiling Systems

To test functionality of the AI in the prototype, the AI Debugger has been used extensively. A few examples of this include being able to see Blackboard variables update in real-time and what actors the AI perception component can perceive. This helps in understanding how the AI systems are communicating and identify any bugs or issues.

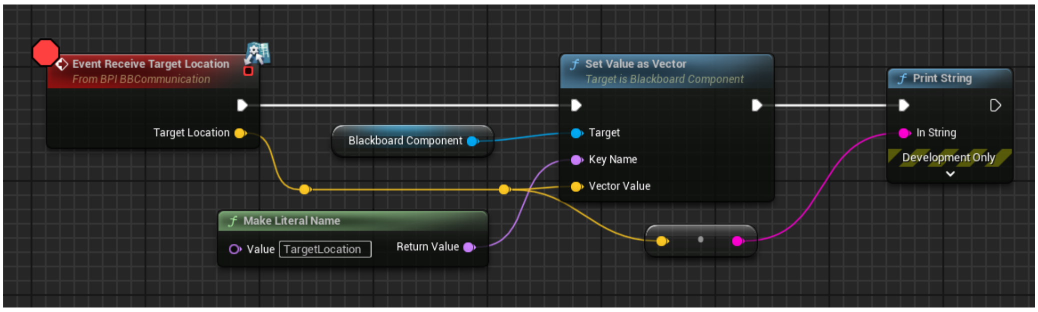


```
Tap [Apostrophe] to close, use highlighted keys to toggle categories.
Ctrl+Tilde:HUD  Ctrl+Tab:DebugMessages  Tab:Spectator
0:Navmesh  1:AI  2:BehaviorTree  3:EQS  4:Perception

[CATEGORY: AI]
Controller Name: AIC_AIController_C_0
Pawn Name: BP_PlayerCharacter_C_UAID_6C02E05F1D2871DC01_1266575370
Movement Mode: Walking, Base: StaticMeshActor_UAID_84A93E8DA9B71FDC01_2036611951.StaticMes
NavData: Default, Path following: Moving (path:0, visited:2), dist2D:[illegible] distZ:16
Behavior: Running, Tree: BT_PlayerCharacter
Active task: BT_PlayerCharacter::BTTask_MoveToTargetLocation[34]
Gameplay tasks: 0
Montage: None

[CATEGORY: BehaviorTree]
Brain Component: BTComponent
Behavior tree: BT_PlayerCharacter
 0. Selector [Selector]
 28. Selector [Selector]
 BTTask_MoveToTargetLocation:

Target Location: TargetLocation
Blackboard Key Type: Vector
Acceptance Radius: SelfActor
Set Radius Manually?: True
Manual Acceptance: 25.0
```

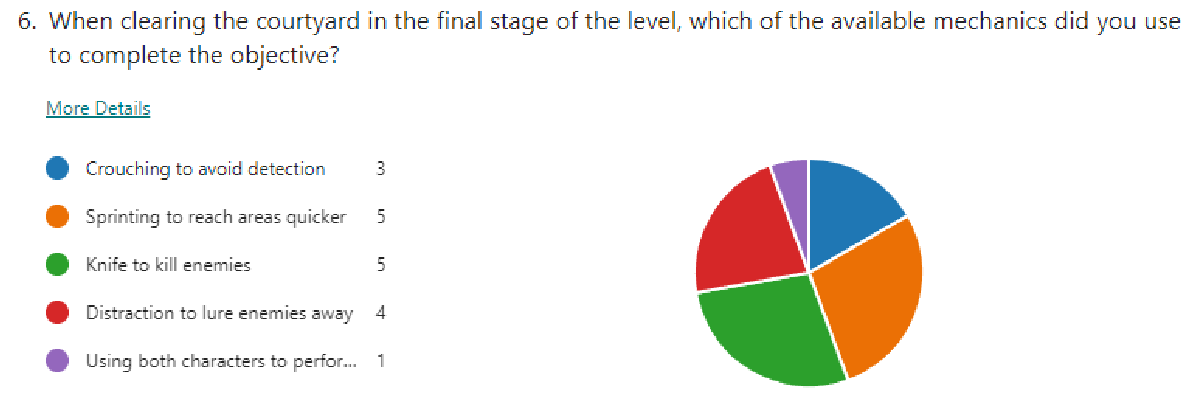
```
Time: 48.14s

Blackboard (asset: BB_AICharacter)
 AttackRange: 0.000000
 AttackType: None
 AudioRange: 0.000000
 FireRate: 0.000000
 IsInRange: false
 MagEmpty: false
 SelfActor: BP_PlayerCharacter_C_UAID_6C02E05F1D2871DC01_1266575370
 Stop: false
 TargetActor: None
 TargetLocation: X=-6147.235 Y=82.099 Z=277.875
```



Breakpoints and Print Strings have also been used to ensure logic is being executed and data is being communicated across the various systems that make up the prototype's functionality.



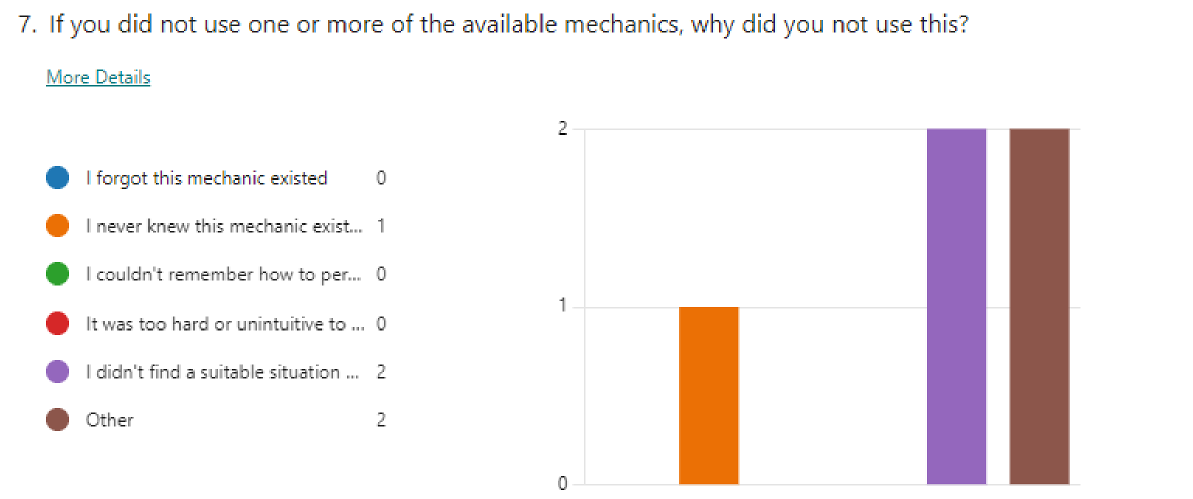
Lastly, user testing has taken place to gather feedback on the prototype. Participants played the prototype then answered a questionnaire on their experience. This feedback was then analysed, and actionable steps were identified and implemented aiming to improve the players' experience.



6. When clearing the courtyard in the final stage of the level, which of the available mechanics did you use to complete the objective?

More Details

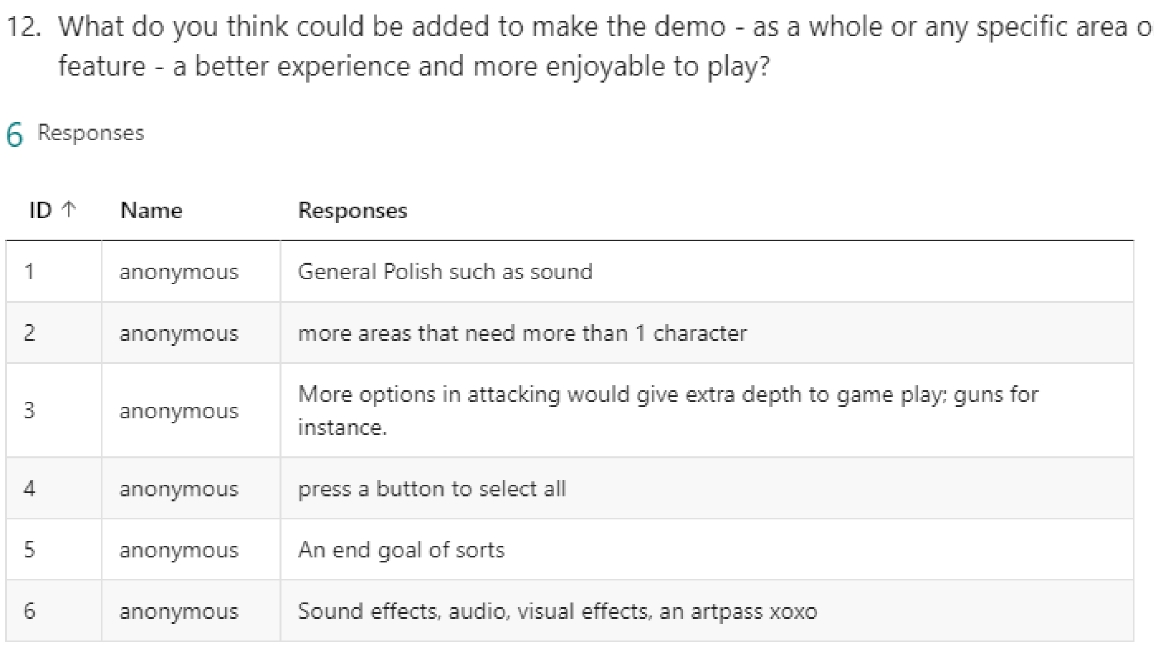
●	Crouching to avoid detection	3
●	Sprinting to reach areas quicker	5
●	Knife to kill enemies	5
●	Distraction to lure enemies away	4
●	Using both characters to perfor...	1



7. If you did not use one or more of the available mechanics, why did you not use this?

More Details

●	I forgot this mechanic existed	0
●	I never knew this mechanic exist...	1
●	I couldn't remember how to per...	0
●	It was too hard or unintuitive to ...	0
●	I didn't find a suitable situation ...	2
●	Other	2



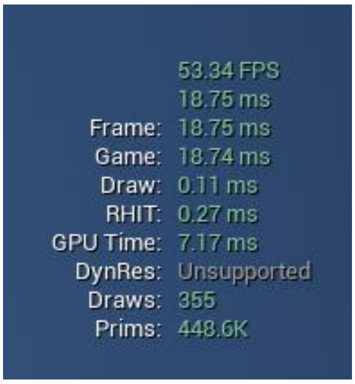
12. What do you think could be added to make the demo - as a whole or any specific area or feature - a better experience and more enjoyable to play?

6 Responses

ID ↑	Name	Responses
1	anonymous	General Polish such as sound
2	anonymous	more areas that need more than 1 character
3	anonymous	More options in attacking would give extra depth to game play; guns for instance.
4	anonymous	press a button to select all
5	anonymous	An end goal of sorts
6	anonymous	Sound effects, audio, visual effects, an artpass xoxo

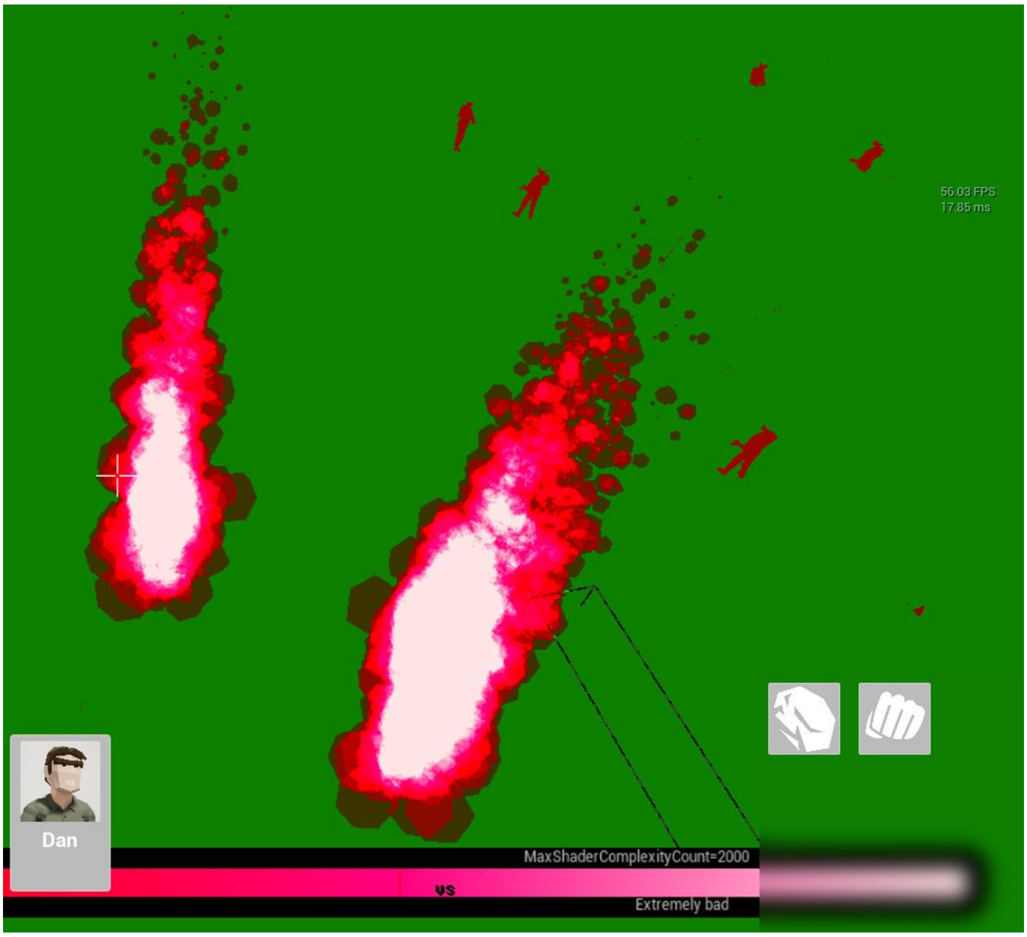
Profiling Graphics

Console commands 'Stat FPS' and 'Stat Unit' were used regularly to debug the Framerate and Graphics Performance during gameplay. This identified if any specific areas of the level or interactions with systems negatively affected performance, which could then be investigated further.



```
          53.34 FPS
          18.75 ms
    Frame: 18.75 ms
     Game: 18.74 ms
     Draw: 0.11 ms
     RHIT: 0.27 ms
 GPU Time: 7.17 ms
   DynRes: Unsupported
    Draws: 355
    Prims: 448.6K
```

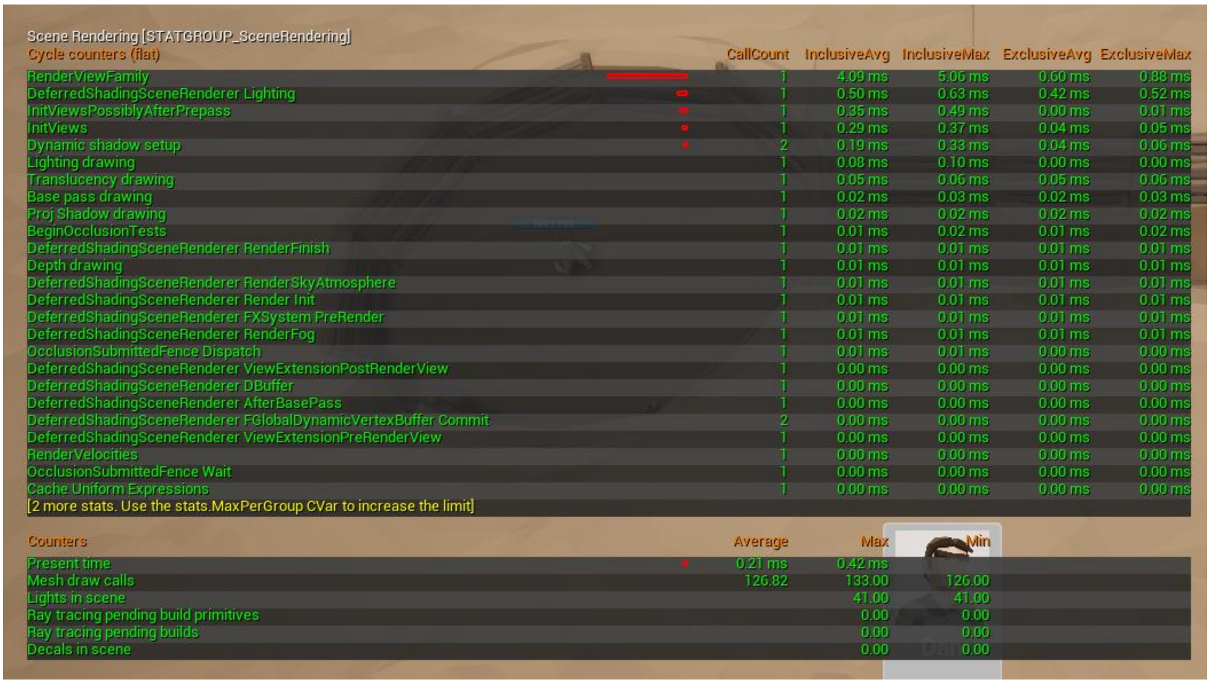
Shader Complexity was analysed using the keyboard shortcut F5 in-game. This identifies any areas using complex shaders that may hinder performance. In the image below, these flames & smoke FX are complex shaders, however using Stat FPS you can see this does not noticeably affect framerate.



```
56.03 FPS
17.85 ms
```

```
MaxShaderComplexityCount=2000
```

Lastly, Stat SceneRendering was used occasionally to view the stats of various graphics processes during gameplay. Whilst much of this debugger's contents is difficult to decipher, any unusually high values could be researched and investigated further to identify graphics issues.

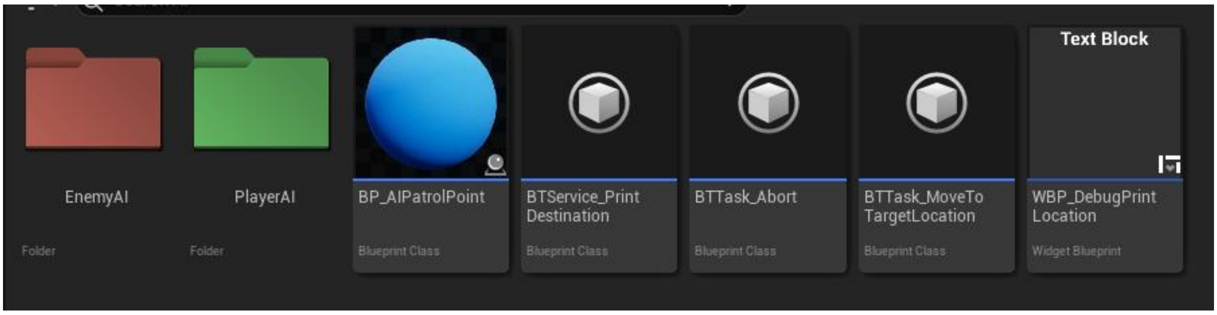


```
Scene Rendering [STATGROUP_SceneRendering]
Cycle counters (flat)                                         CallCount InclusiveAvg InclusiveMax ExclusiveAvg ExclusiveMax
RenderViewFamily                                                      1     4.09 ms      5.06 ms      0.60 ms      0.88 ms
DeferredShadingSceneRenderer Lighting                                 1     0.50 ms      0.63 ms      0.42 ms      0.52 ms
InitViewsPossiblyAfterPrepass                                         1     0.35 ms      0.49 ms      0.00 ms      0.01 ms
InitViews                                                             1     0.29 ms      0.37 ms      0.04 ms      0.05 ms
Dynamic shadow setup                                                  2     0.19 ms      0.33 ms      0.04 ms      0.06 ms
Lighting drawing                                                      1     0.08 ms      0.10 ms      0.00 ms      0.00 ms
Translucency drawing                                                  1     0.05 ms      0.06 ms      0.05 ms      0.06 ms
Base pass drawing                                                     1     0.02 ms      0.03 ms      0.02 ms      0.03 ms
Proj Shadow drawing                                                   1     0.02 ms      0.02 ms      0.02 ms      0.02 ms
BeginOcclusionTests                                                   1     0.01 ms      0.02 ms      0.01 ms      0.02 ms
DeferredShadingSceneRenderer RenderFinish                             1     0.01 ms      0.01 ms      0.01 ms      0.01 ms
Depth drawing                                                         1     0.01 ms      0.01 ms      0.01 ms      0.01 ms
DeferredShadingSceneRenderer RenderSkyAtmosphere                      1     0.01 ms      0.01 ms      0.01 ms      0.01 ms
DeferredShadingSceneRenderer Render Init                              1     0.01 ms      0.01 ms      0.01 ms      0.01 ms
DeferredShadingSceneRenderer FXSystem PreRender                       1     0.01 ms      0.01 ms      0.01 ms      0.01 ms
DeferredShadingSceneRenderer RenderFog                                1     0.01 ms      0.01 ms      0.01 ms      0.01 ms
OcclusionSubmittedFence Dispatch                                      1     0.01 ms      0.01 ms      0.00 ms      0.00 ms
DeferredShadingSceneRenderer ViewExtensionPostRenderView              1     0.00 ms      0.00 ms      0.00 ms      0.00 ms
DeferredShadingSceneRenderer DBuffer                                  1     0.00 ms      0.00 ms      0.00 ms      0.00 ms
DeferredShadingSceneRenderer AfterBasePass                            1     0.00 ms      0.00 ms      0.00 ms      0.00 ms
DeferredShadingSceneRenderer FGlobalDynamicVertexBuffer Commit        2     0.00 ms      0.00 ms      0.00 ms      0.00 ms
DeferredShadingSceneRenderer ViewExtensionPreRenderView               1     0.00 ms      0.00 ms      0.00 ms      0.00 ms
RenderVelocities                                                      1     0.00 ms      0.00 ms      0.00 ms      0.00 ms
OcclusionSubmittedFence Wait                                          1     0.00 ms      0.00 ms      0.00 ms      0.00 ms
Cache Uniform Expressions                                             1     0.00 ms      0.00 ms      0.00 ms      0.00 ms
[2 more stats. Use the stats.MaxPerGroup CVar to increase the limit]

Counters                                                        Average         Max          Min
Present time                                                    0.21 ms     0.42 ms
Mesh draw calls                                                  126.82      133.00       126.00
Lights in scene                                                               41.00        41.00
Ray tracing pending build primitives                                           0.00         0.00
Ray tracing pending builds                                                     0.00         0.00
Decals in scene                                                                0.00         0.00
```

Coding Standards

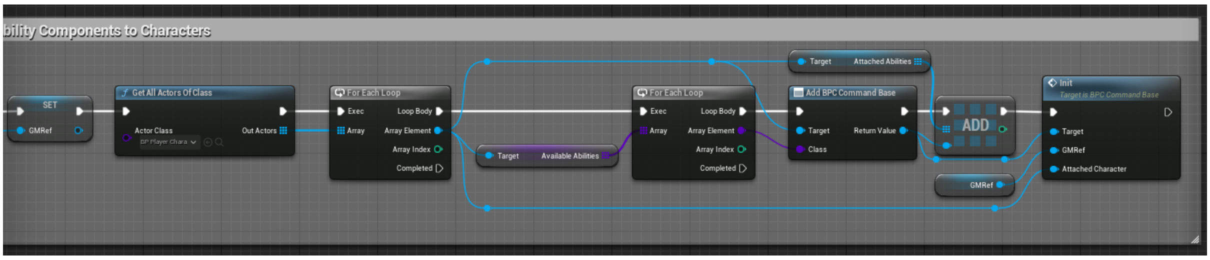
Programming Standards

Suitable coding standards have been adhered to during development. Appropriate naming conventions and prefixes have been used to ensure both readability and understanding for other developers viewing the project, as well as being able to locate blueprint classes quicker due to logical organisation, resulting in a more efficient workflow.

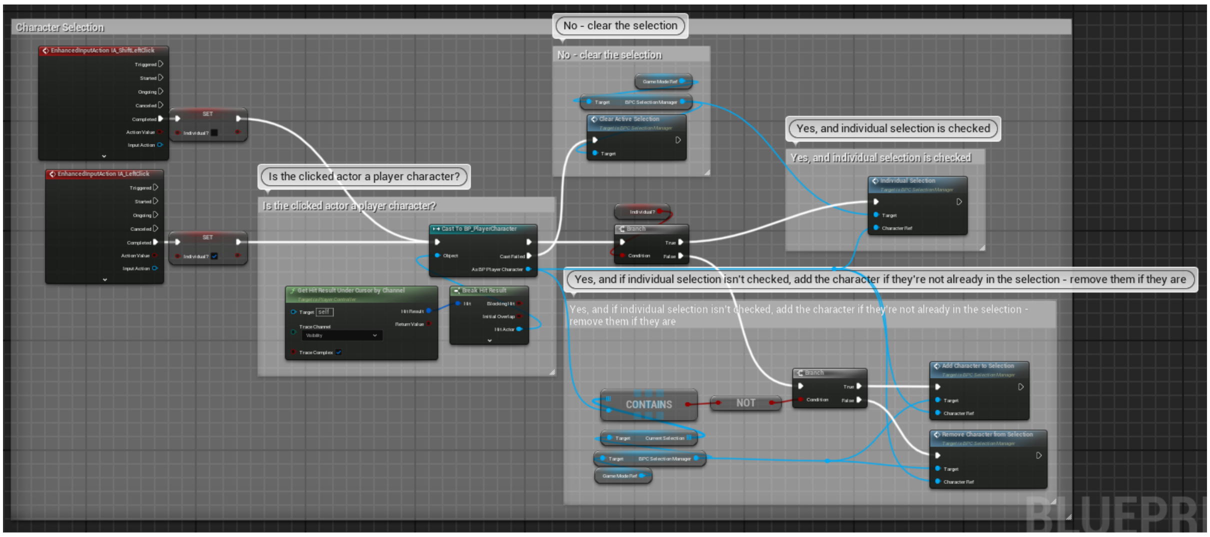


Style Guide & Commenting Rules

Blueprints have been organised neatly, making sure nodes aren't too cramped and good use of reroute nodes to visualise a logical flow of execution and a more readable, understandable blueprint.



Comment boxes group chunks of logic together with text to explain what the logic does, and comment bubbles ensure the comments are readable when zoomed out so other developers viewing the event graphs can quickly identify and navigate to the script they wish to see.

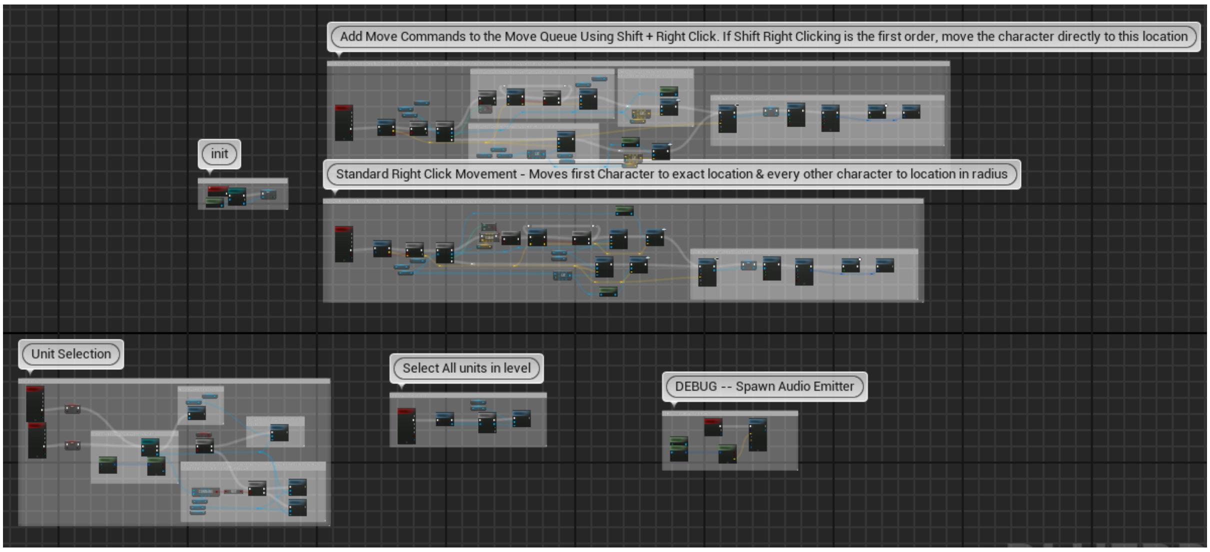

Character Selection

No - clear the selection

Is the clicked actor a player character?

Yes, and individual selection is checked

Yes, and if individual selection isn't checked, add the character if they're not already in the selection - remove them if they are


Add Move Commands to the Move Queue Using Shift + Right Click. If Shift Right Clicking is the first order, move the character directly to this location

init

Standard Right Click Movement - Moves first Character to exact location & every other character to location in radius

Unit Selection

Select All units in level

DEBUG -- Spawn Audio Emitter

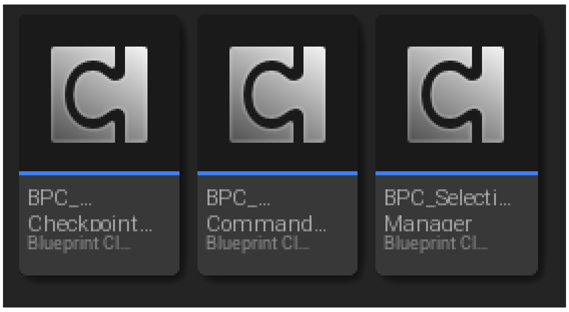
Naming conventions are used to ensure blueprint names are logical and consistent, making navigating to blueprints quicker and optimising development. For example:

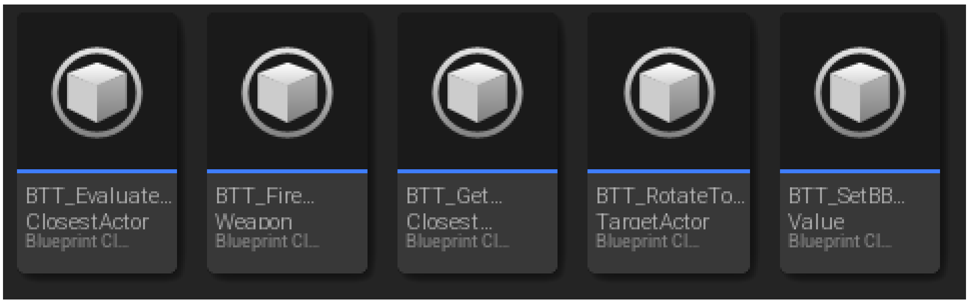
Actor Blueprint: BP_

Blueprint Component: BPC_

Behaviour Tree Task: BTT_

AI Controller: AIC_


BPC_... Checkpoint... Blueprint Cl...
BPC_... Command... Blueprint Cl...
BPC_Selecti... Manager Blueprint Cl...


BTT_Evaluate... ClosestActor Blueprint Cl...
BTT_Fire... Weapon Blueprint Cl...
BTT_Get... Closest... Blueprint Cl...
BTT_RotateTo... TargetActor Blueprint Cl...
BTT_SetBB... Value Blueprint Cl...

Code Review Procedures

Each system in the prototype will be planned before development & reviewed once implemented to ensure suitable practices have been adhered to during development. This ensures logic is dynamic in construction and systems can be added and modified in an efficient manner.

Logic will be reviewed and refactored at key milestones during development. For example, once all 'Must Have' features have been implemented, they will be reviewed and refactored if it is decided that the system has been built inefficiently. This negates the knock on effect bad coding practices may have on snowballing problems in latter stages of development.

Production Overview

Moscow

The Moscow analysis outlines features that are critical for the project (Must Have) alongside features the prototype Should & Could have, whilst also outlining features that tend to feature in existing Real Time Tactics games but won't be included in the prototype.

Must Have
Basic unit selection – click units to select them
Move selected units by clicking locations around the map
Give commands to units using UI buttons i.e Stop, Hold Fire, Attack
AI Enemies that guard & patrol when idle
Enemies have perception/line of sight ranges
Enemies investigate visual/audio stimulus
AI Enemy states – Idle, Suspicious, Alert
Enemies attack friendly units in range
Cover system that AI can interact with to avoid damage/remain undetected
Friendly/Hostile units can be damaged & killed

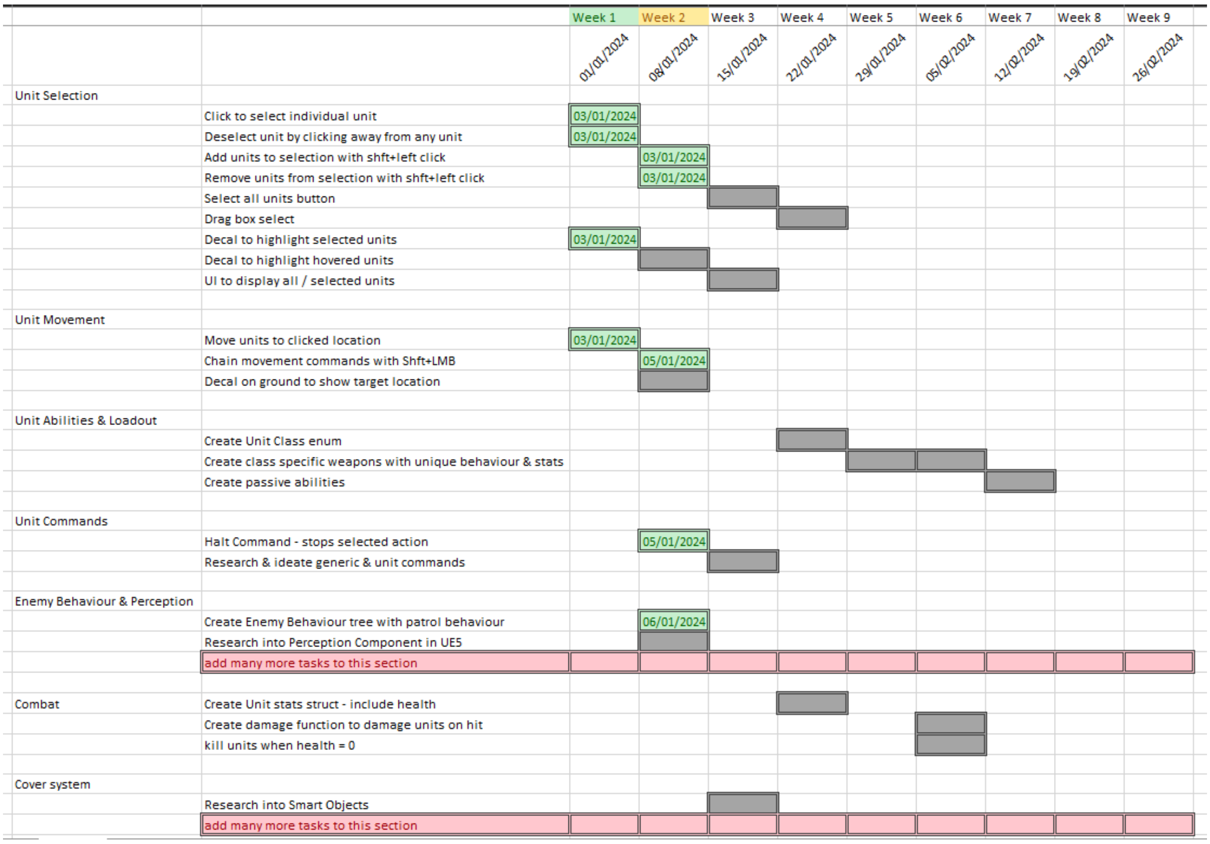
Should Have
Add/remove units from selection using a chorded input action
Units have unique abilities depending on class i.e Scout has binoculars, rifleman has grenade etc
Units have unique loadouts with weapons that have different stats & behaviour
Animations to accompany behaviours – running, crouching, attacking etc
EQS system for more advanced/natural AI movement & interaction with the environment
Firing, explosions, & other audio origins alert nearby enemies

Could Have
Drag Box Select
Basic Objective system – kill all enemies
Units have unique passive abilities depending on their class
Stealth kills – undetected units can approach enemies from behind for a silent kill
Item pickups – units can pick up items throughout the world i.e ammo, health packs, grenades
Tactical Pause – pauses the game, allows the player to give commands that will execute when unpaused

Won't Have
Character selection to select unit before taking them into the mission
Advanced objective system – capture areas, defend locations etc
Different types of enemies – Infantry, Tank etc
Advanced stealth system – hide bodies in cover, use bushes to remain hidden
Multiple levels/maps

Timeline

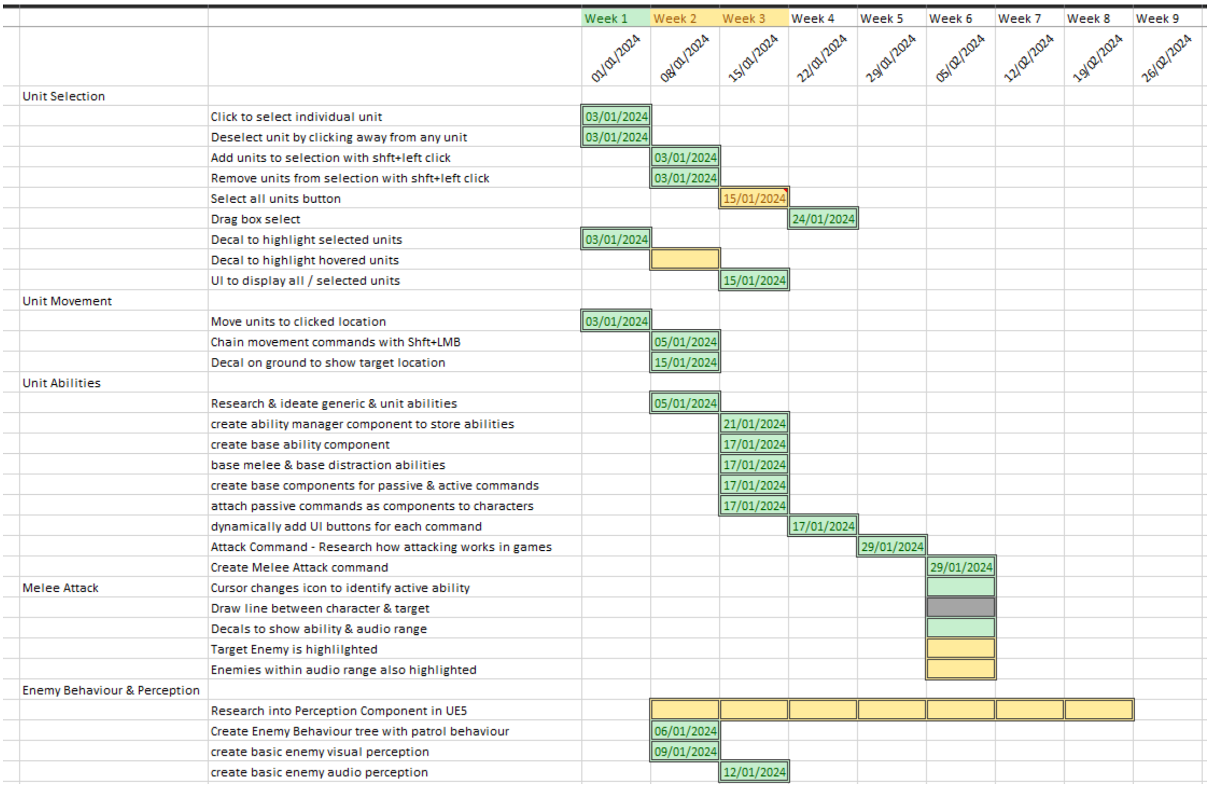
Iteration 1 of the Gantt Chart, created at the start of the project, outlines the completion date and timeframe of all the known tasks. However, Enemy Perception has many unknown variables and require extensive research before the specific tasks that need to be completed are known.



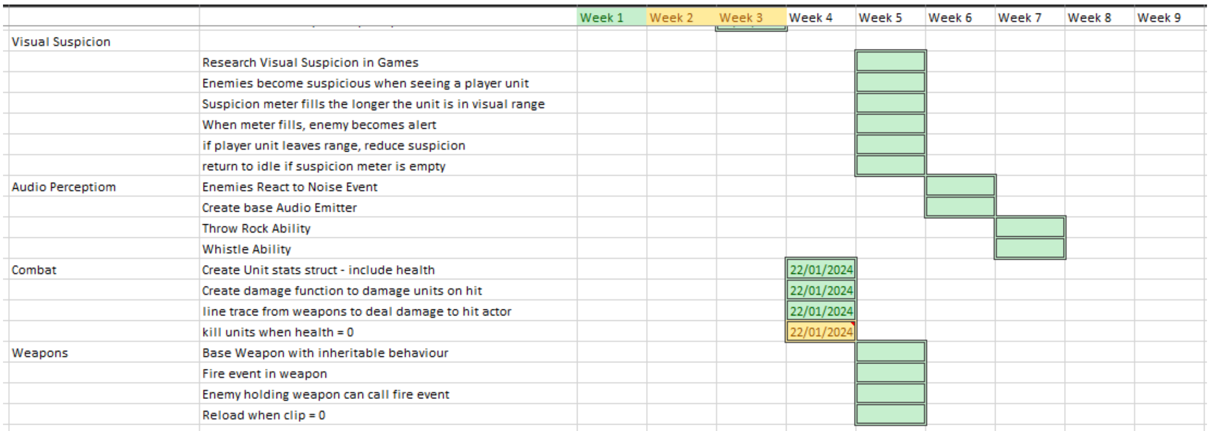
		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9
		01/01/2024	08/01/2024	15/01/2024	22/01/2024	29/01/2024	05/02/2024	12/02/2024	19/02/2024	26/02/2024
Unit Selection										
	Click to select individual unit	03/01/2024								
	Deselect unit by clicking away from any unit	03/01/2024								
	Add units to selection with shft+left click		03/01/2024							
	Remove units from selection with shft+left click		03/01/2024							
	Select all units button			▨						
	Drag box select				▨					
	Decal to highlight selected units	03/01/2024								
	Decal to highlight hovered units		▨							
	UI to display all / selected units			▨						
Unit Movement										
	Move units to clicked location	03/01/2024								
	Chain movement commands with Shft+LMB		05/01/2024							
	Decal on ground to show target location		▨							
Unit Abilities & Loadout										
	Create Unit Class enum				▨					
	Create class specific weapons with unique behaviour & stats					▨	▨			
	Create passive abilities							▨		
Unit Commands										
	Halt Command - stops selected action		05/01/2024							
	Research & ideate generic & unit commands			▨						
Enemy Behaviour & Perception										
	Create Enemy Behaviour tree with patrol behaviour		06/01/2024							
	Research into Perception Component in UE5		▨							
	add many more tasks to this section									
Combat	Create Unit stats struct - include health				▨					
	Create damage function to damage units on hit						▨			
	kill units when health = 0						▨			
Cover system										
	Research into Smart Objects			▨						
	add many more tasks to this section									

Throughout development the vision of the project changed somewhat, meaning the Gantt Chart had to be updated with more relevant tasks. Many tasks were removed, modified, and added, and a new Gantt Chart was created to better represent the tasks required to fulfil the criteria.

The new vision for the prototype leaned towards the Stealth sub-genre of RTT games, removing many combat elements and focussing heavily on movement, abilities, and AI Perception.



		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9
		01/01/2024	08/01/2024	15/01/2024	22/01/2024	29/01/2024	05/02/2024	12/02/2024	19/02/2024	26/02/2024
Unit Selection										
	Click to select individual unit	03/01/2024								
	Deselect unit by clicking away from any unit	03/01/2024								
	Add units to selection with shft+left click		03/01/2024							
	Remove units from selection with shft+left click		03/01/2024							
	Select all units button			15/01/2024						
	Drag box select				24/01/2024					
	Decal to highlight selected units	03/01/2024								
	Decal to highlight hovered units		🟨							
	UI to display all / selected units			15/01/2024						
Unit Movement										
	Move units to clicked location	03/01/2024								
	Chain movement commands with Shft+LMB		05/01/2024							
	Decal on ground to show target location		15/01/2024							
Unit Abilities										
	Research & ideate generic & unit abilities		05/01/2024							
	create ability manager component to store abilities			21/01/2024						
	create base ability component			17/01/2024						
	base melee & base distraction abilities			17/01/2024						
	create base components for passive & active commands			17/01/2024						
	attach passive commands as components to characters			17/01/2024						
	dynamically add UI buttons for each command				17/01/2024					
	Attack Command - Research how attacking works in games					29/01/2024				
	Create Melee Attack command						29/01/2024			
Melee Attack	Cursor changes icon to identify active ability						🟩			
	Draw line between character & target						▨			
	Decals to show ability & audio range						🟩			
	Target Enemy is highlilghted						🟨			
	Enemies within audio range also highlighted						🟨			
Enemy Behaviour & Perception										
	Research into Perception Component in UE5		🟨	🟨	🟨	🟨	🟨	🟨	🟨	
	Create Enemy Behaviour tree with patrol behaviour		06/01/2024							
	create basic enemy visual perception		09/01/2024							
	create basic enemy audio perception			12/01/2024						



		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9
Visual Suspicion										
	Research Visual Suspicion in Games					🟩				
	Enemies become suspicious when seeing a player unit					🟩				
	Suspicion meter fills the longer the unit is in visual range					🟩				
	When meter fills, enemy becomes alert					🟩				
	if player unit leaves range, reduce suspicion					🟩				
	return to idle if suspicion meter is empty					🟩				
Audio Perceptiom	Enemies React to Noise Event						🟩			
	Create base Audio Emitter						🟩			
	Throw Rock Ability							🟩		
	Whistle Ability							🟩		
Combat	Create Unit stats struct - include health				22/01/2024					
	Create damage function to damage units on hit				22/01/2024					
	line trace from weapons to deal damage to hit actor				22/01/2024					
	kill units when health = 0				22/01/2024					
Weapons	Base Weapon with inheritable behaviour					🟩				
	Fire event in weapon					🟩				
	Enemy holding weapon can call fire event					🟩				
	Reload when clip = 0					🟩				

Budgeting

System	Feature	Time Budget (hrs)	Financial Budget (£)
Unit Selection	LMB to select individual unit	1	0
	Shft+LMB to select multiple units	2	0
	Select All Units	0.5	0
	Selection Decal	1	0
	Selection UI to display all / selected units	6.5	0
	Subtotal:	**11**	**0**
Unit Movement	Move units to clicked location	1	0
	Chain movement commands with Shft+LMB	2	0
	Target Location Ground Decal	1	0
	Subtotal:	**4**	**0**
Unit Abilities	Base Command to initialise & execute command logic	1	0
	Passive Command attached to characters on spawned	1	0
	Dynamically create UI for each attached command	3	0
	Base Melee Ability	6	0
	Base Distraction Ability	8	0
	Whistle Distraction	4	0
	Rock Distraction	6	0
	Subtotal:	**29**	**0**
Enemy Behaviour & Perception	Idle Patrol Behaviour	6	0
	Visual Perception to Detect Player Units	6	0
	Audio Perception to react to noise stimuli	8	0
	Become suspicious & attack enemies in visual range	12	0
	Investigate audio stimulus	10	0
	Subtotal:	**42**	**0**
Weapons & Combat	Create Base Weapon & Data	4	0
	Damage & Kill Enemies & Players	2	0
	Fire Weapon	4	0
	Reload Weapon on Mag Empty	2	0
	Subtotal:	**12**	**0**
	Total	**98**	**0**