

# **Artificial Intelligence for Games**

---

**Unreal Engine 4.23**

June 2020  
Digital Learning Games  
Tallinn University

# Introduction

This project implements AI Behaviour Tree and Blackboard included in Unreal Engine 4 (UE4). This is made for the 'Artificial Intelligence for Games' course in Tallinn University.

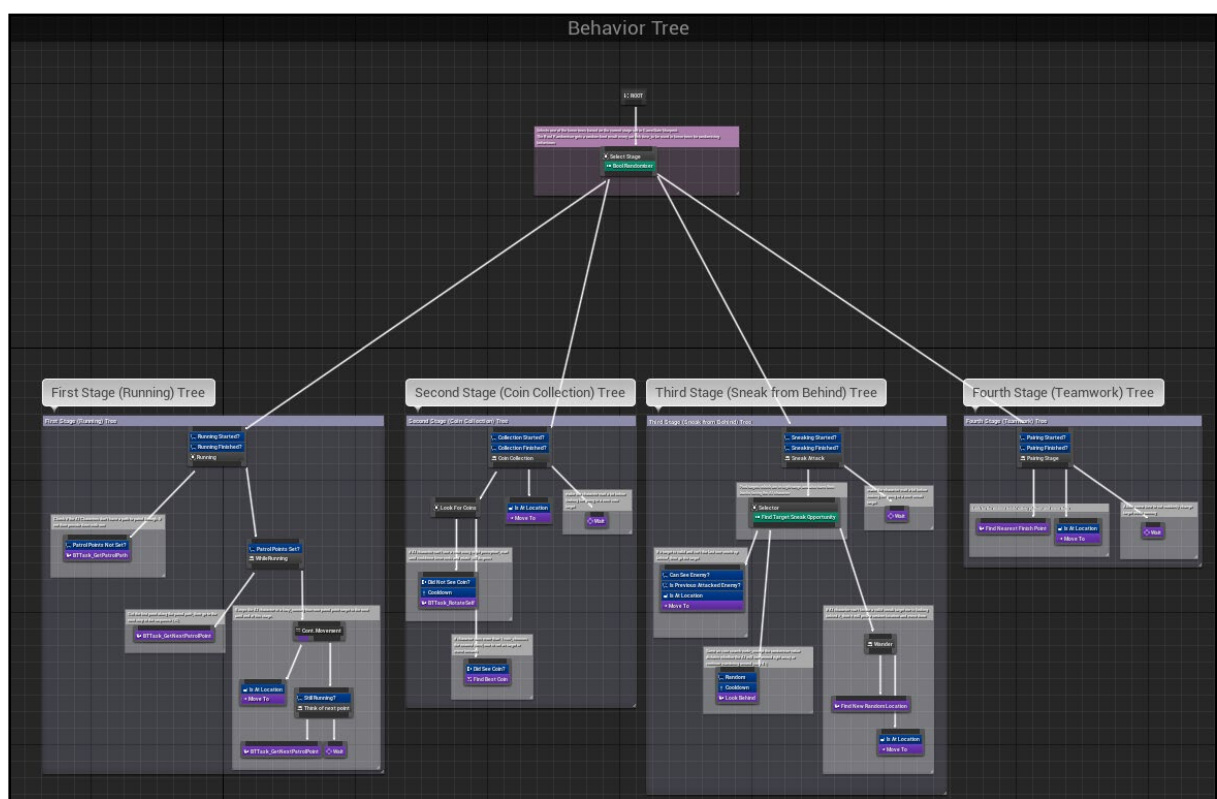
## Prerequisites

- UE4 editor version 4.23
- Github Desktop (optional)

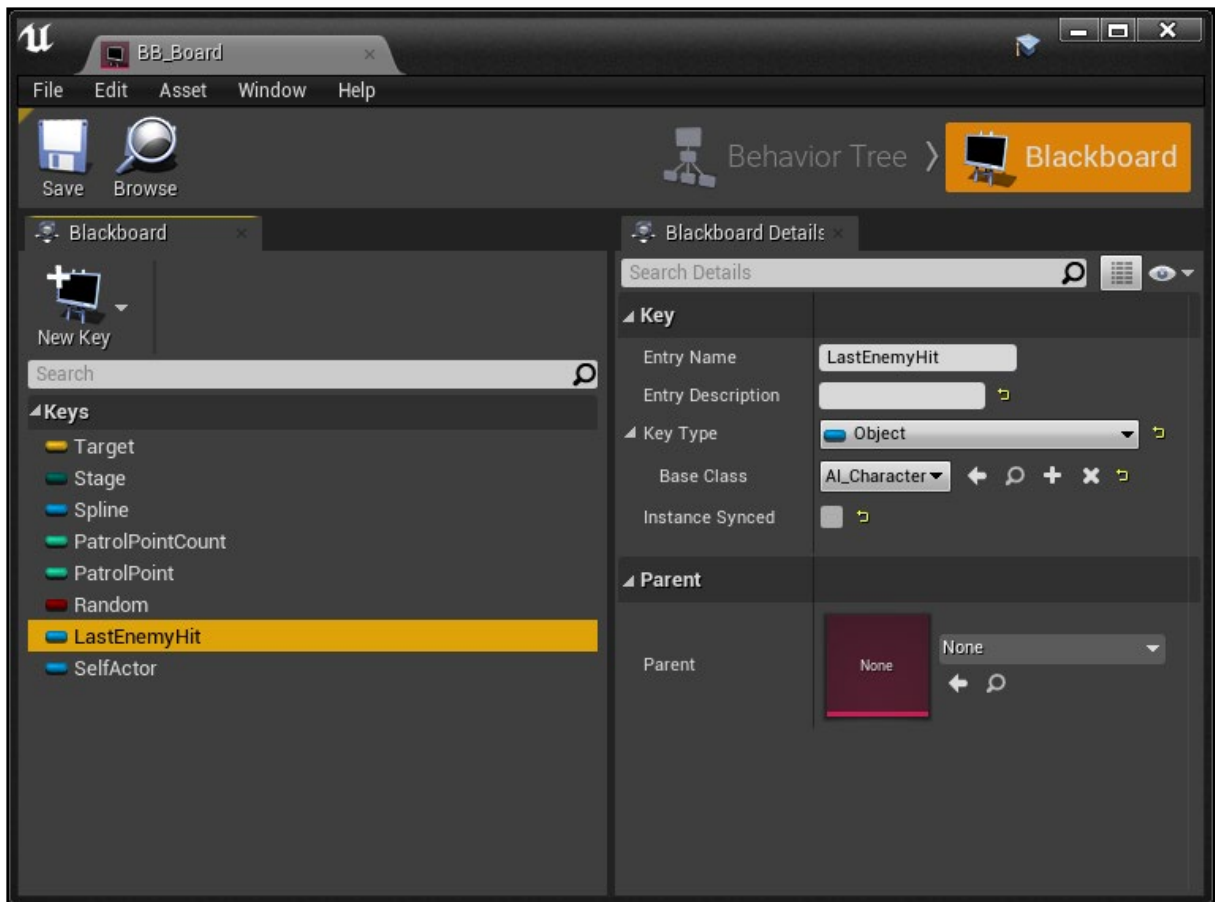
## Built With

- All code was built using UE4's visual scripting blueprints.
- Specific AI behaviour was built using UE4's Behaviour Tree, Blackboard, Environment Query System (EQS) and blueprints.

Note: AI-related blueprints and Behaviour Tree are commented within the project as well.



One Behaviour Tree used in this project



One Blackboard used in project – One key highlighted with details displayed on right pane

## Access to Project

The project is uploaded on GitHub. At the time of this document, it has two branches: 'master' and 'Refined\_AI\_Behaviour'. The first attempt of this project covered all videos provided by the course instructor and was saved in the 'master' branch. Afterwards, each of the 5 stages in the project (explained below) was added upon to a certain degree; to experiment with more complex AI behaviour and try new things. The GitHub repository of the project can be accessed through [this link](#).

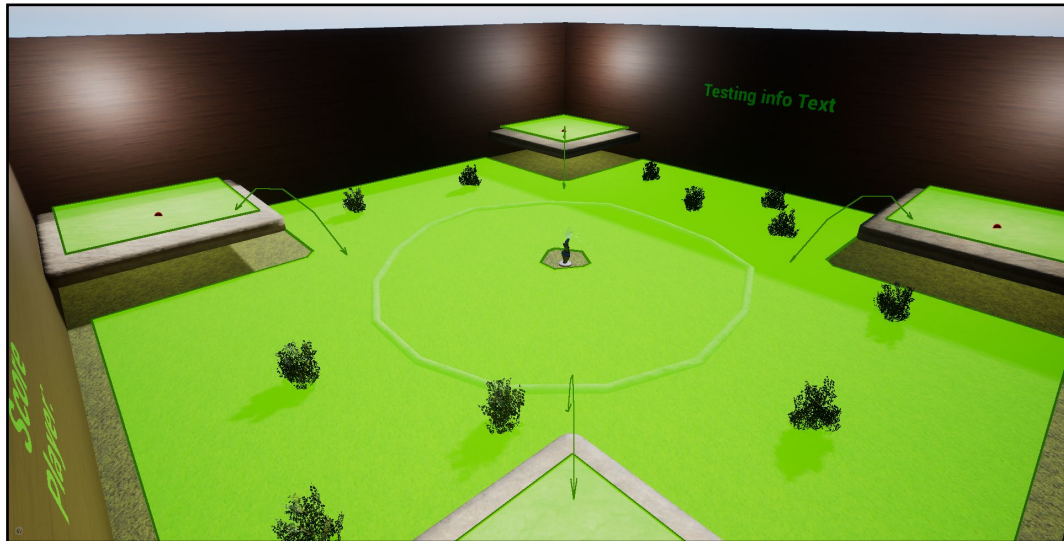
## AI Behaviour

In this project, there are 5 AI-controlled characters. The level goes through 5 different stages as follows:

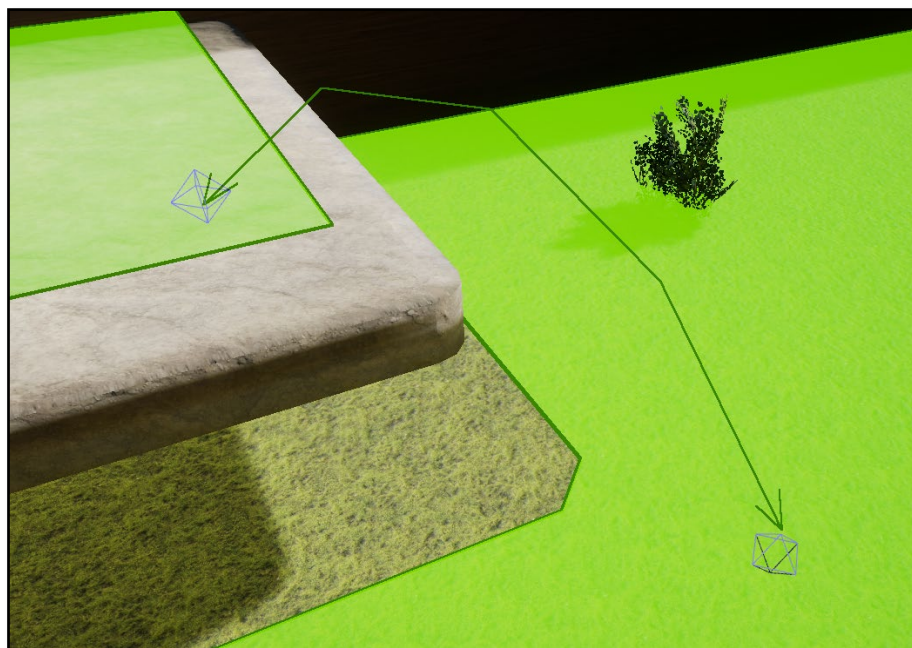
- 1- Running along a specific set path
- 2- Collecting objects (represented as coins) randomly generated in the level area
- 3- Running up to behind each other to try landing a sneak/back attack

- 4- Grouping with teammates at safety points
- 5- End of level and final score calculation

The characters navigate the level using the built-in Navmesh system, and can jump onto higher platforms using Navlinks setup – also included as part of UE4.



Navmesh System: Green overlay showing parts accessible by AI characters



Navlink Example: Diamonds wireframes represent start & end points, arrows show expected jump trajectory

# Initial Run

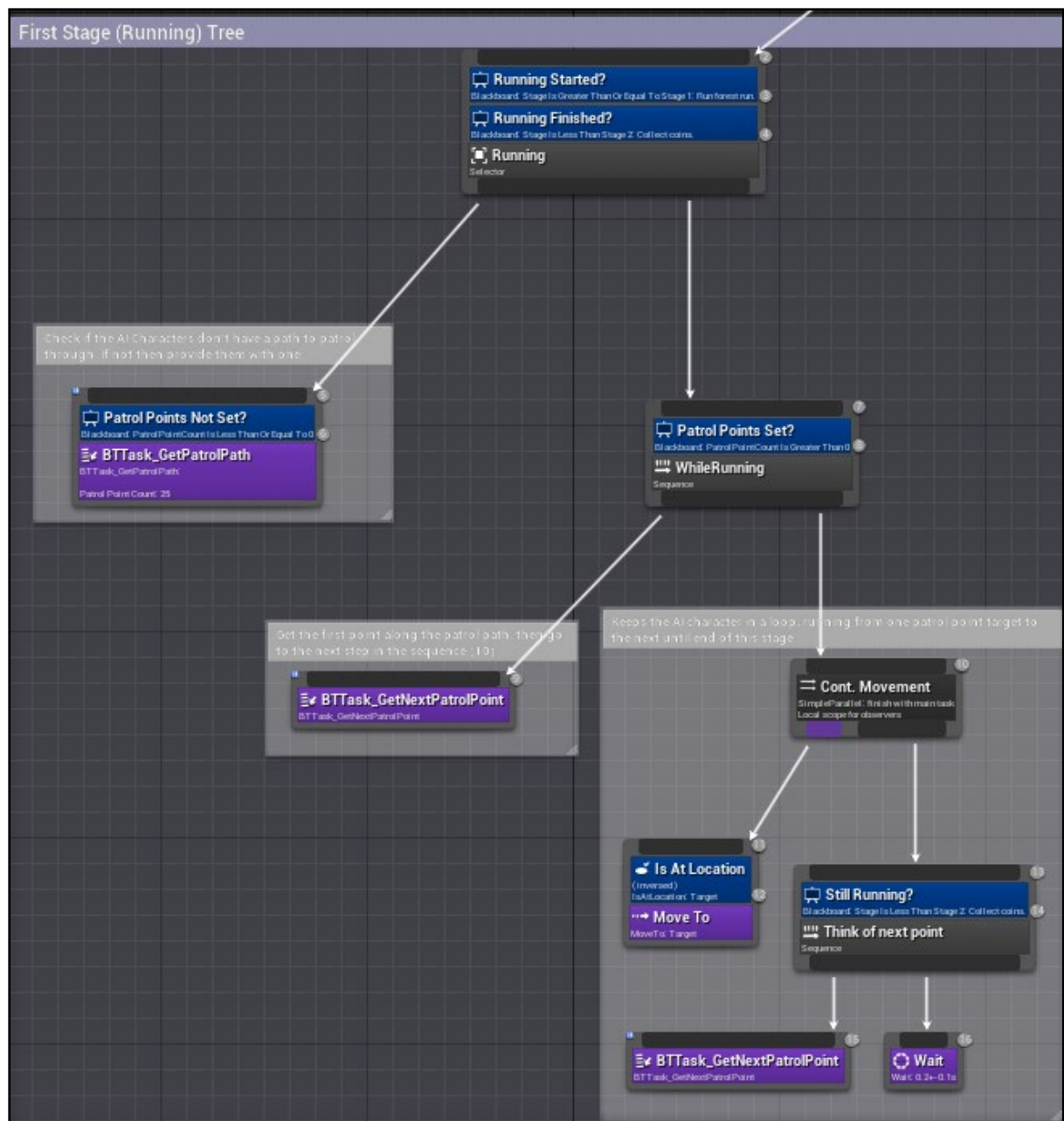
The following describes the initial project setup and the planned and executed AI behaviour according to the video instructions provided by the course instructor. These are stored in the 'master' branch of the GitHub repository.

## Stage 1 - Run, Forrest, Run!

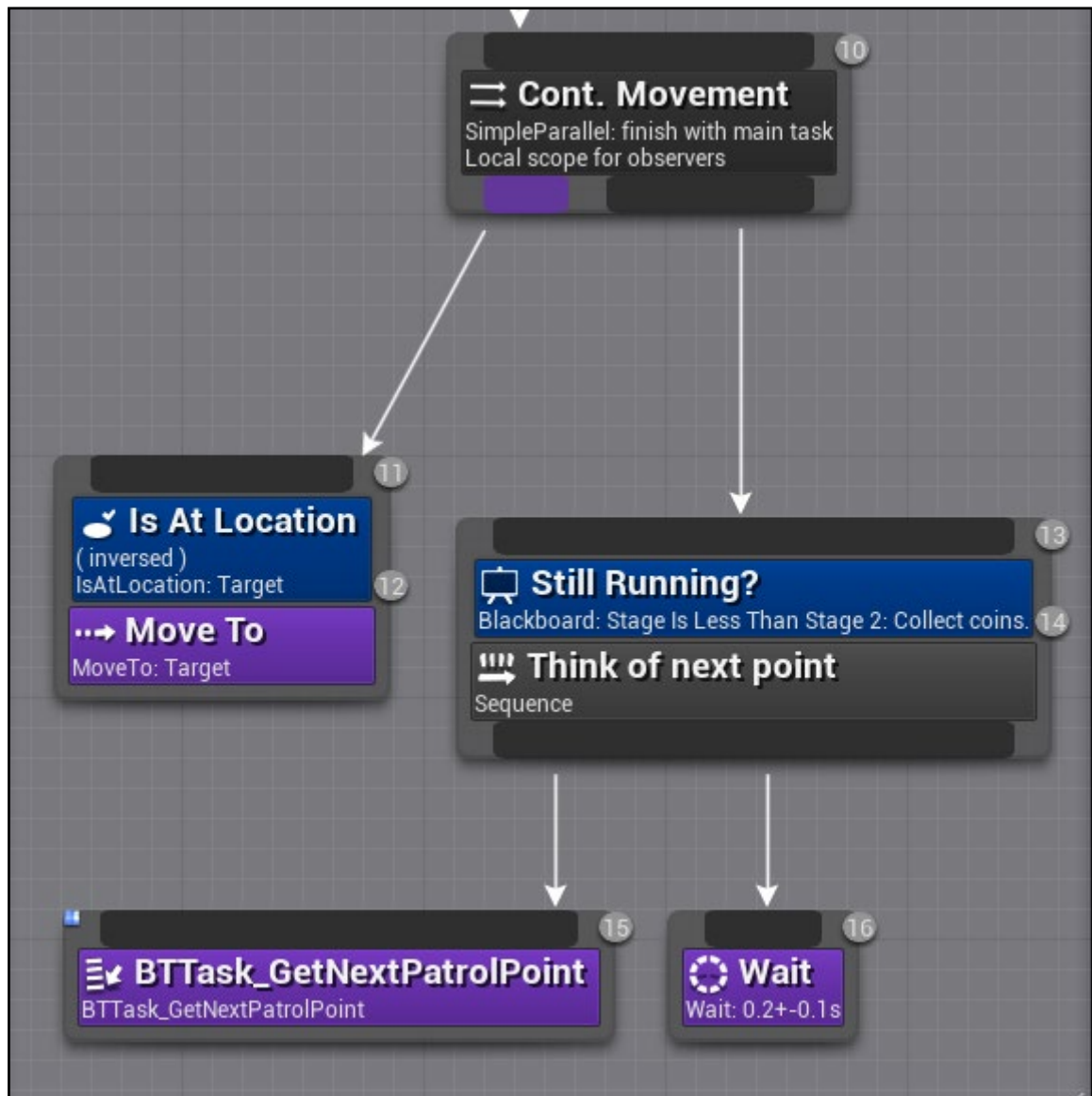


In this stage the AI characters are given a target point along a set spline, and are expected to run towards that point. While that point is the active target and they're running towards it, they are also looking up the next point to move to after reaching the current one. This 'parallel' calculation ensures the characters are continuously running without stuttering; which can be caused by the character running to a point, stopping, finding the next point then running again - it's most obvious when the target points are very close to each other, like in our level.





Behaviour Tree: First Stage



'Cont. Movement' node is a parallel type calculation (composite in UE4) telling the AI to perform two processes in parallel ensuring smooth movement

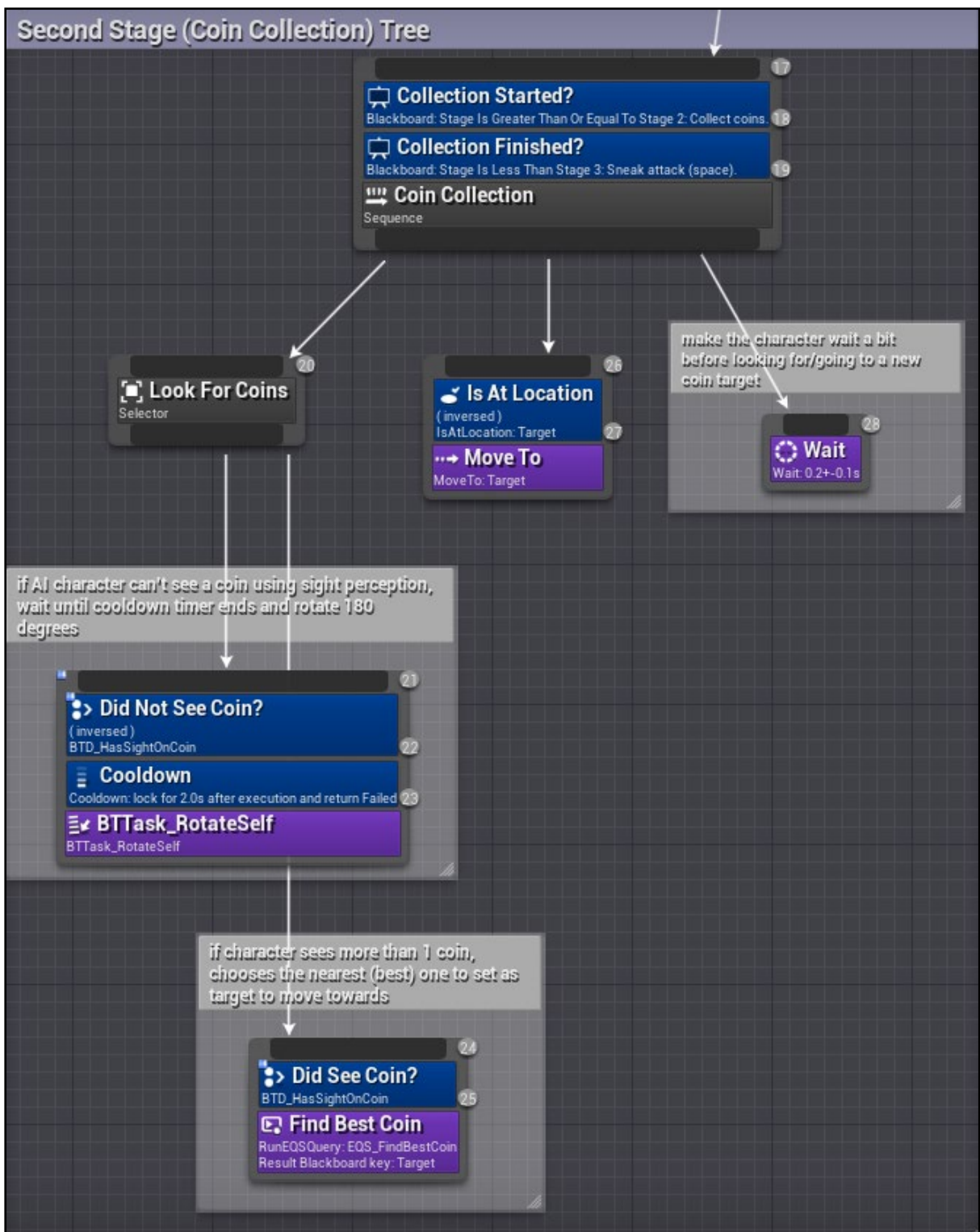
## Stage 2 - Collect Coins



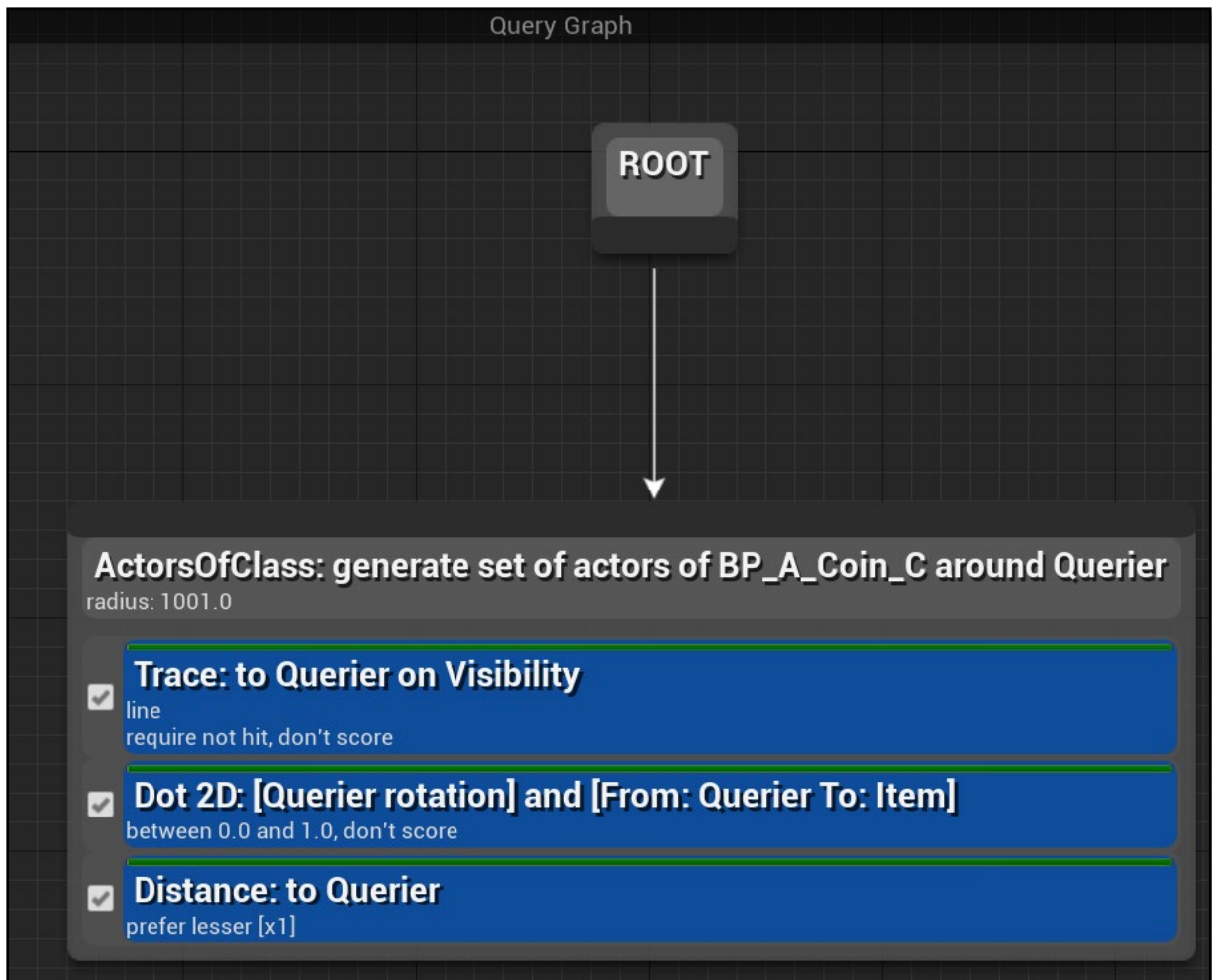
Here, 9 coins are spawned randomly on the ground area every 6 seconds. The AI characters are required to use their site perception (included in UE4) to locate all coin objects. If they cannot see any coins within their sight radius (180 degrees), they will turn around to look behind them. Once they sight 2 or more coins, the EQS for this step kicks in: It measures the distances to all perceived coins, then scores each coin based on how far it is from the character. The nearer a coin is to the character, the higher the score it receives. The AI then chooses the coin with the highest score - i.e. nearest to the character and runs towards it to pick it up. It waits a very short time (simulating a human player looking around) and repeats the process again.

It is possible for a character to not run to any coin. This case happens when all spawned coins are too far away from the character and beyond their sight range (currently at 1000 units).





Behaviour Tree: Second Stage



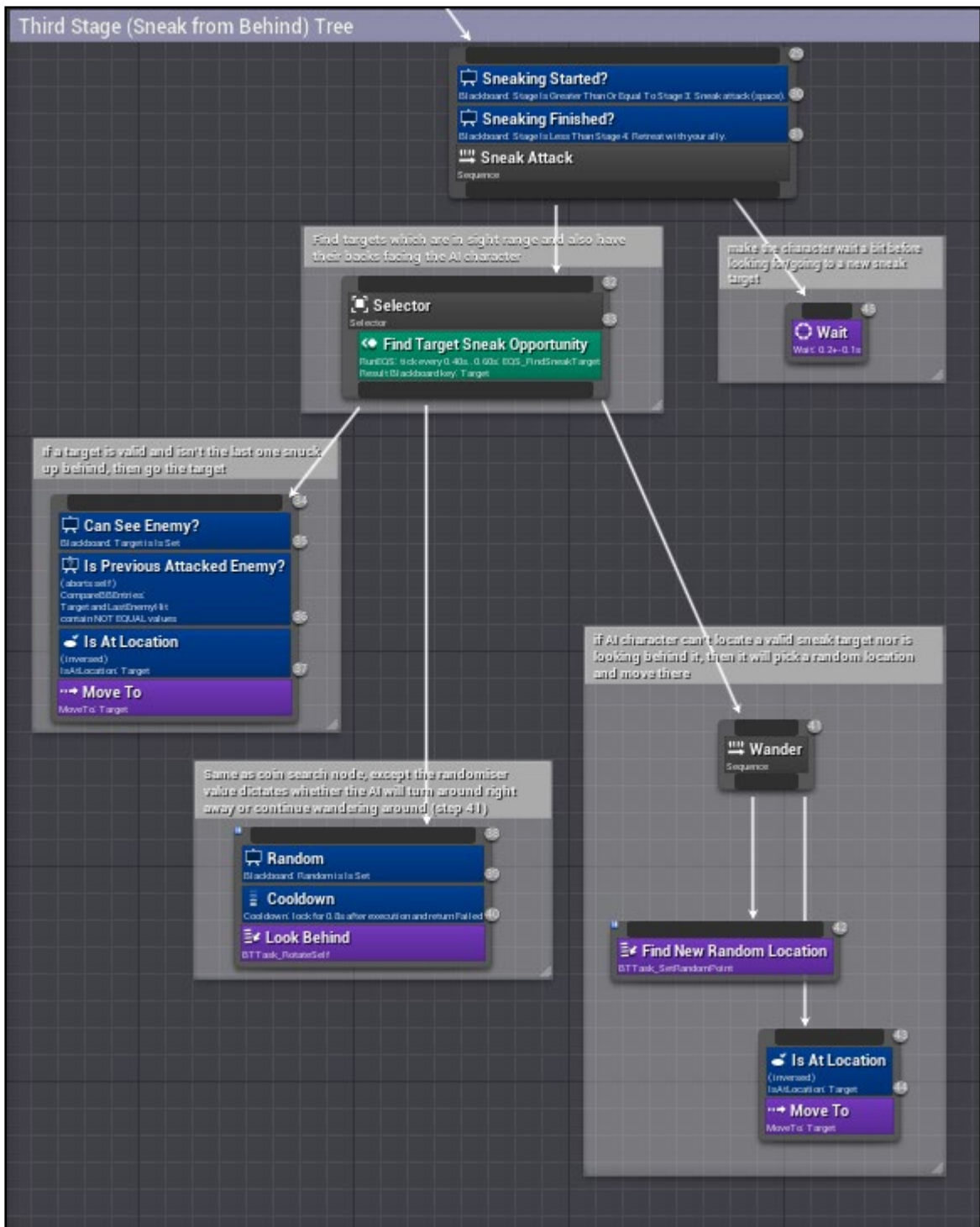
EQS for filtering coin objects and ranking them based on distance

## Stage 3 - Sneak Attack

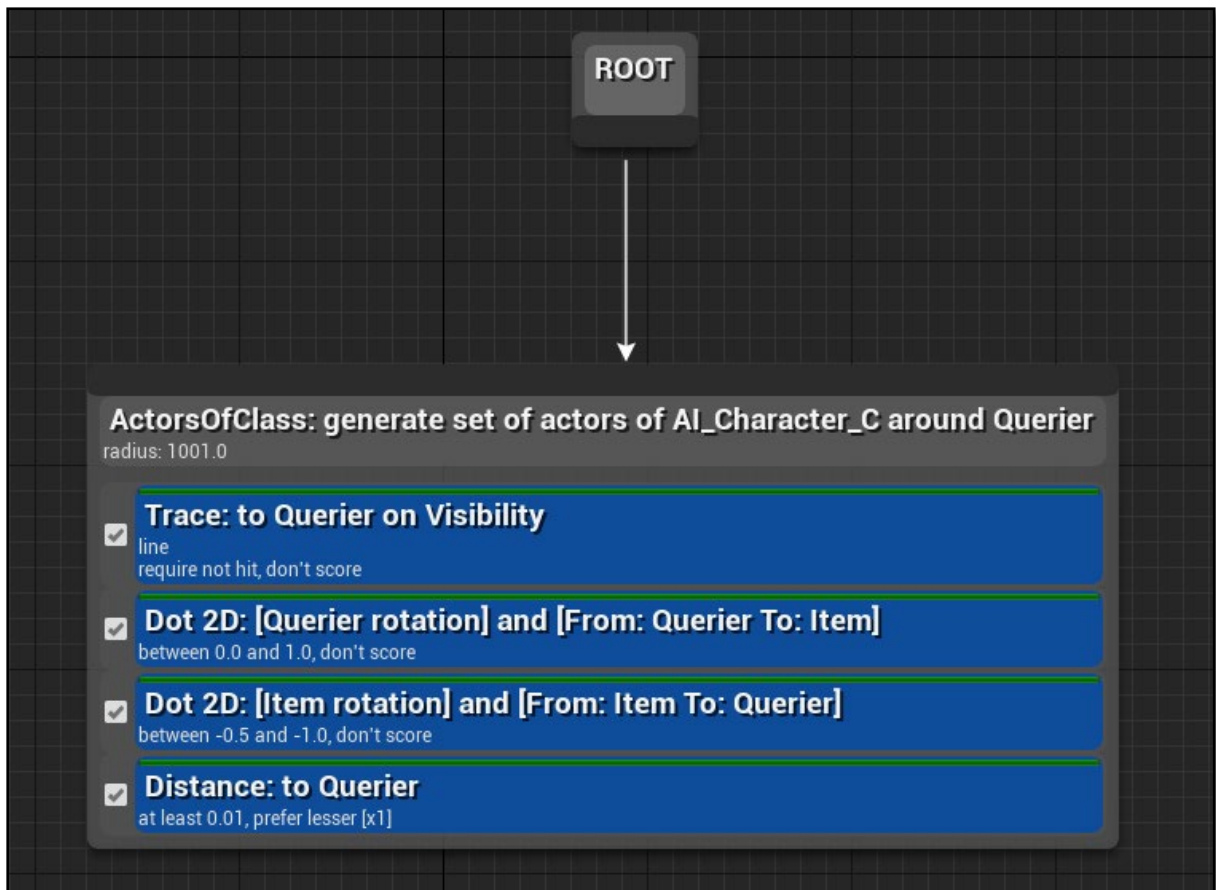


At this stage, the AI characters are expected to move towards each other but only if the other character they sight has their back to them - simulating a sneak attack from the back. The main functionality is similar to the coin collection, with a few exceptions:

- The AI character cannot sneak behind the same 'enemy' twice; this is to avoid having it stuck behind one enemy and not moving away so long as the stage is active
- Although the character can turn around (like in the coin collection stage) if it doesn't see an enemy in front of it, this action is randomised here and when the AI isn't allowed to perform it, it instead picks a random location point anywhere in the level and moves to that point before trying again to find an enemy. This is to simulate a wandering around behaviour, more typical to a human player.
- The AI character now needs to do two checks in the EQS, instead of the one in the coin collection. First, it needs to be able to see an enemy in front of it. Second, the enemy must be looking the opposite way and have their backs facing the AI character. Without this rule, the attack will be head on, and can't be considered sneaky anymore.



Behaviour Tree: Third Stage



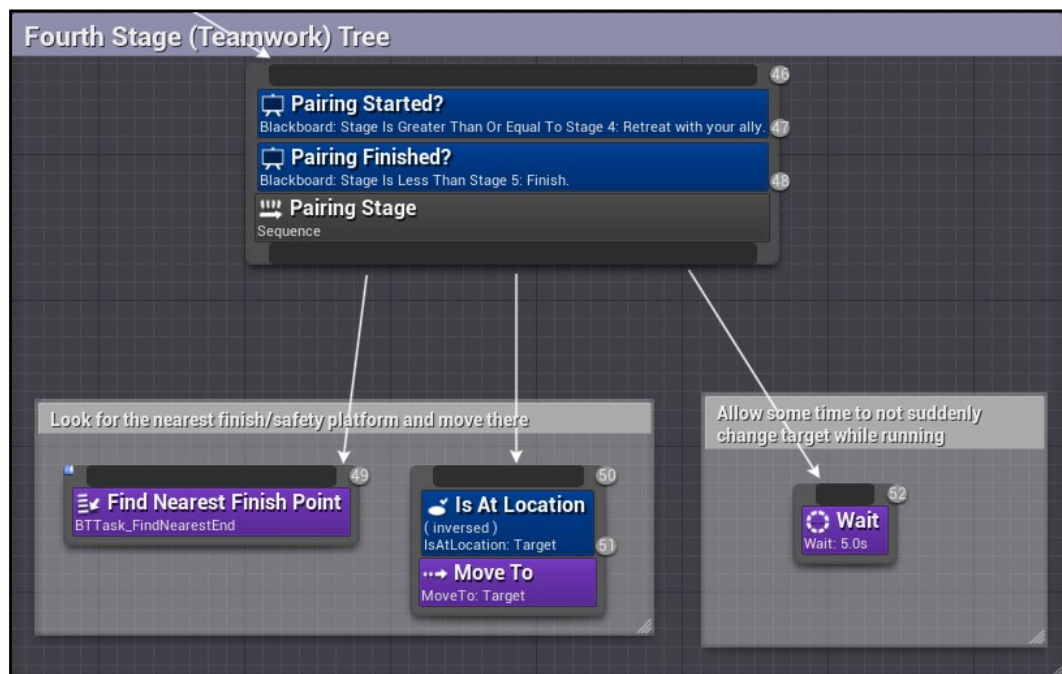
EQS for filtering other characters based on sight and directional orientation, then ranking based on distance



## Stage 4 - Retreat with your Ally



In this final stage, each 2 characters (including the player) are assigned a team and matching colours. The AI is expected to choose one of the safety/end point spots and stand there with their teammate. To do this, each character first finds their teammate, then they find the nearest End Point to both of them and run to that point. Once they get there, they wait until the final stage.



Behaviour Tree: Fourth Stage

## Stage 5 - Finish

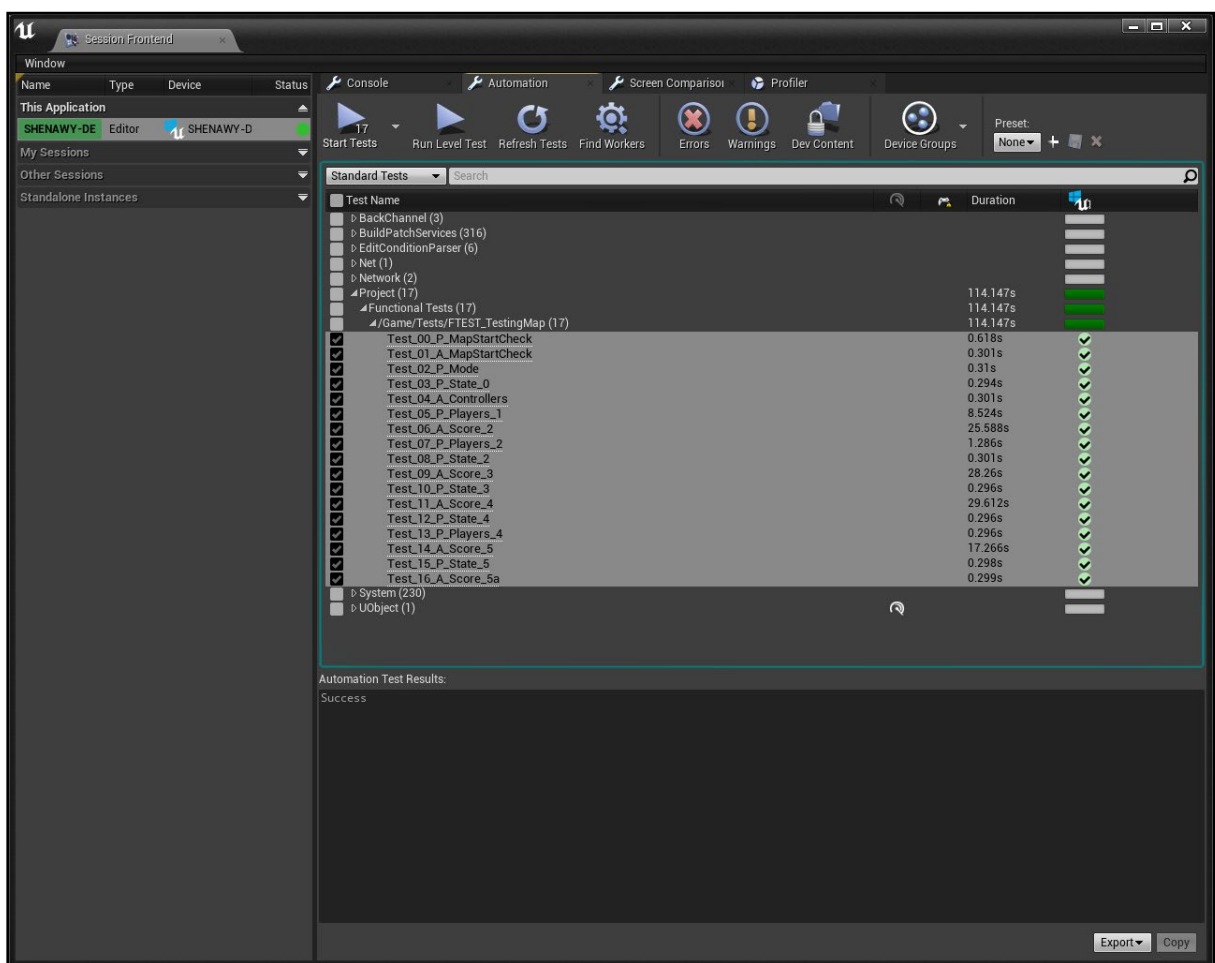
No AI behaviour takes place here. The scores of all 4 previous stages is summed and displayed and the game (and tests if running) end.

## Running the Tests

There are passive and active tests included in the project. Each of these tests is meant to check that all AI conditions exist and have been accurately implemented.

To run them, go to Window -> Developer Tools -> Session Frontend, then click on the 'Automation' tab and click 'Run Level Test'. The included tests will then run automatically with each test ending in 'success' or 'failed'. If the test fails, some information will be shown in the lower Automation Test Results message window.

With the current build, all tests were passed with success by the AI implementation.

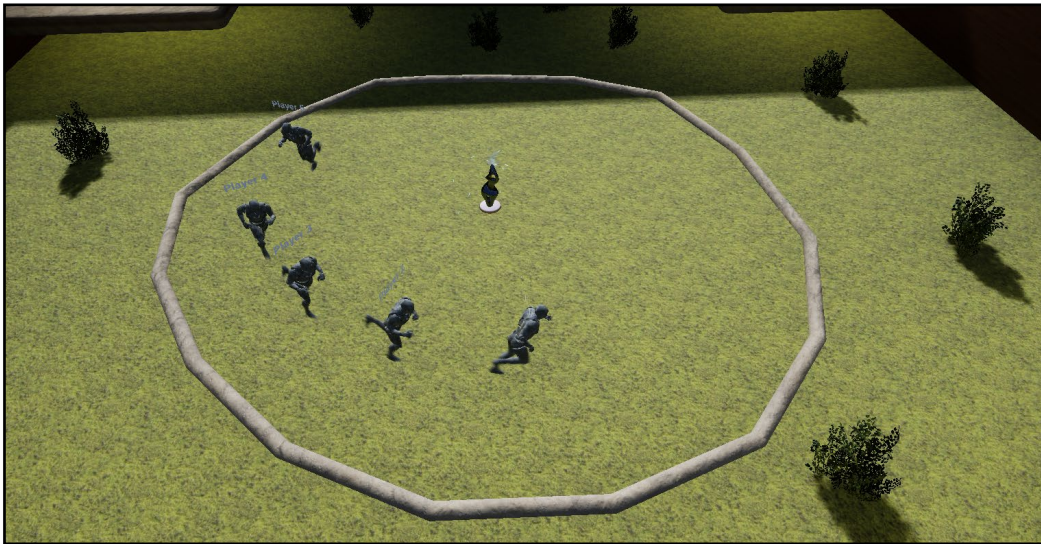


Passive & Active tests returning positive results

# Additional AI Behaviour

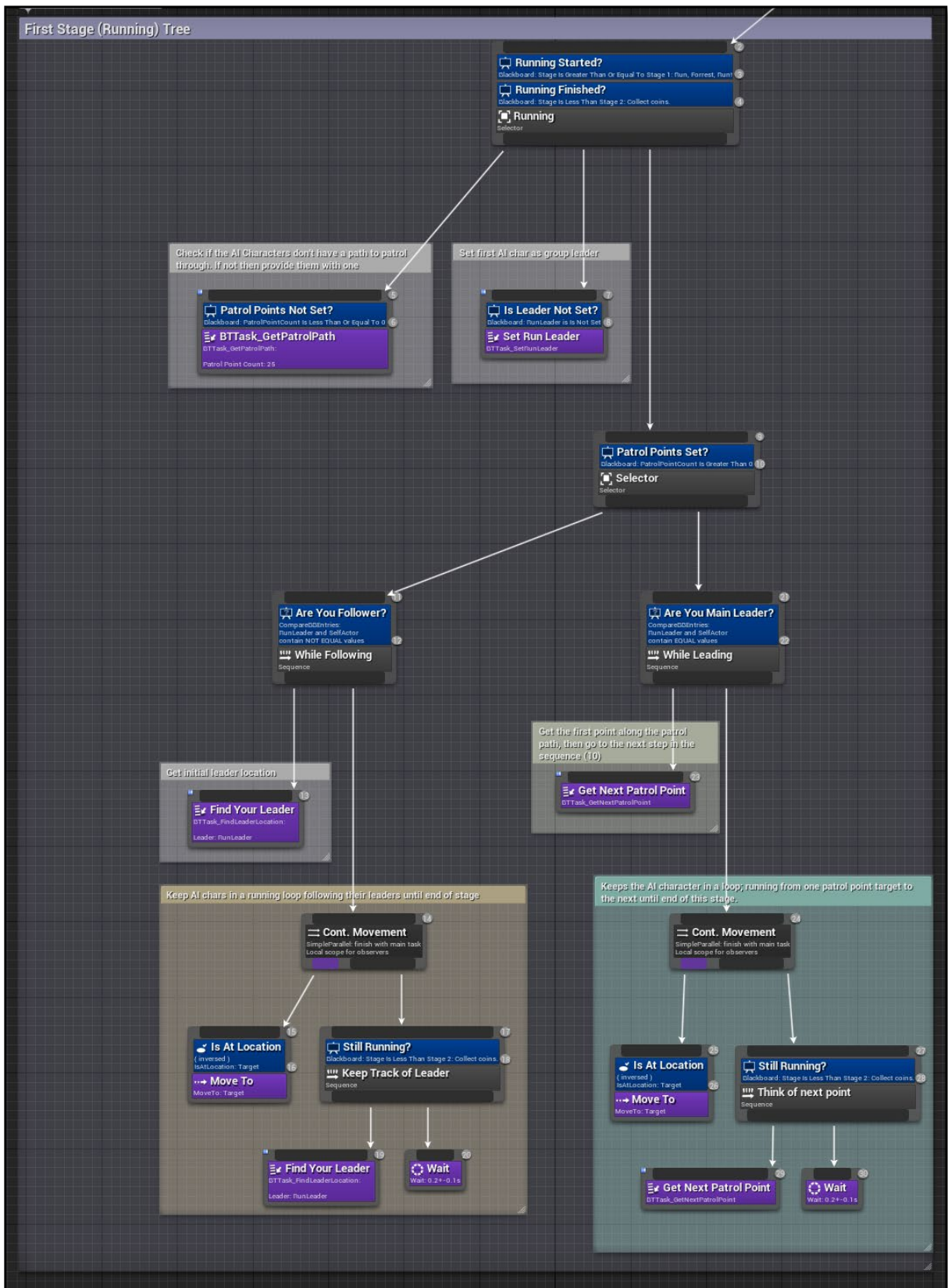
In the 'Refined\_AI\_Behaviour' branch in the GitHub repository, some experimentation with the AI was done to show more complex behaviour. Changes are explained below.

## Modified Stage 1



In the initial run, each AI character ran independently of the other AI characters around the circle. The new behaviour makes the first AI character the leader of the second one, the second becomes the third's leader, and so on. Each AI character follows its leader, while only the first character runs around the circle. This simulates the behaviour of moving as team with a set leader and all walking/running in a line together and in sequence, which also gives the feel that each AI character is aware of the others instead of running randomly and blocking each other.



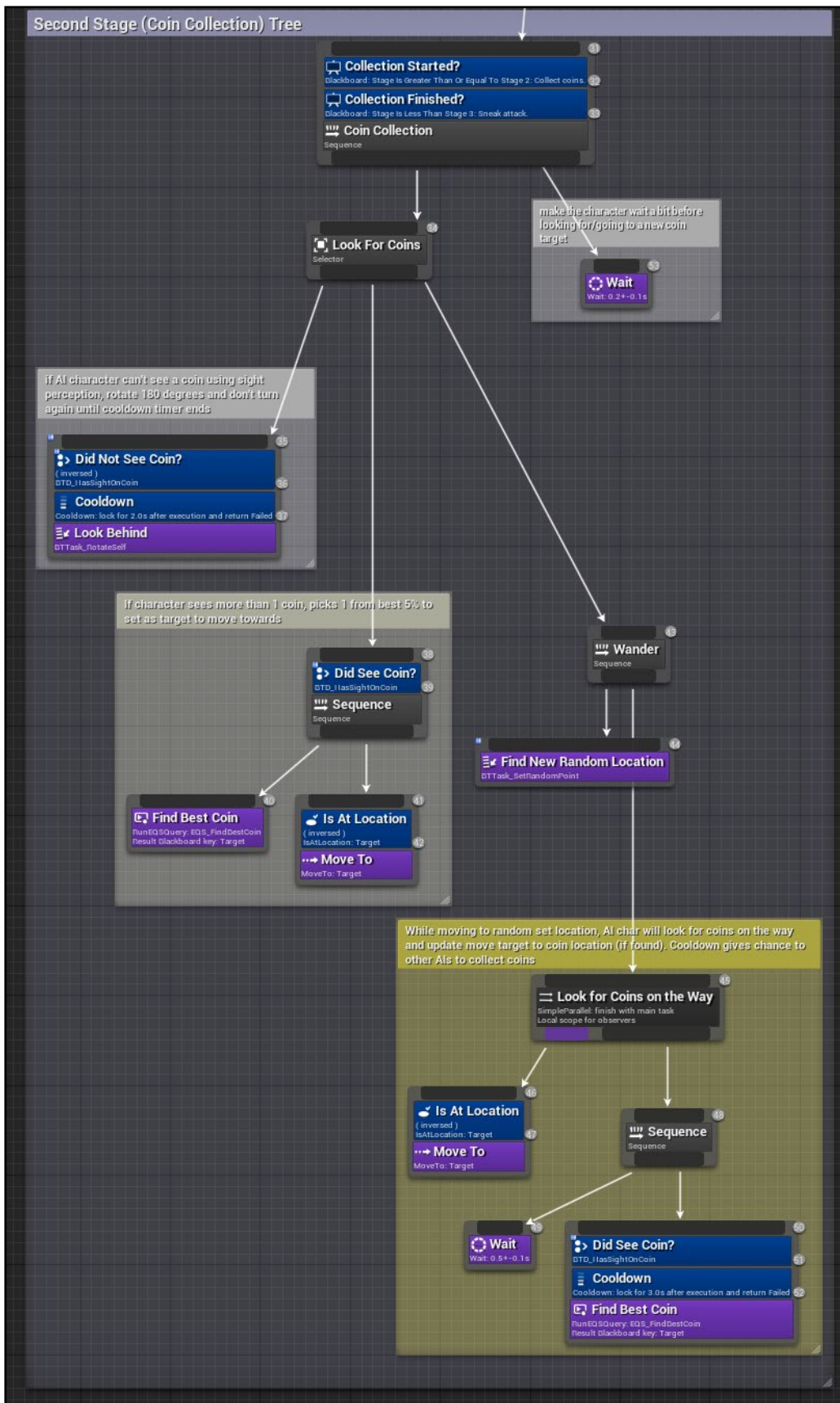


Modified Behaviour Tree: First Stage

## Modified Stage 2

In the original Behaviour Tree (BT), the AI would either locate a coin in their perception range or not. If it located a coin or more, it would calculate the best option to move to and proceed. If it didn't locate any, it would turn around itself and search again. Most of the time, the coins were out of perception range, and it can be sometimes irritating to the player that a coin is ahead of the AI character and yet it's not moving towards it – this is because the coin is outside perception range, but human players don't know that. The BT has been modified in a way that if the AI didn't locate any coins even after turning around, they would then wander around the area to a random point. If, while moving around, they happened to locate another coin, they would then change their target from the random point towards the new found coin. This seems more believable in the eyes of human players, but it also makes the AI more efficient and challenging; as it clears the area of coins much faster than before.

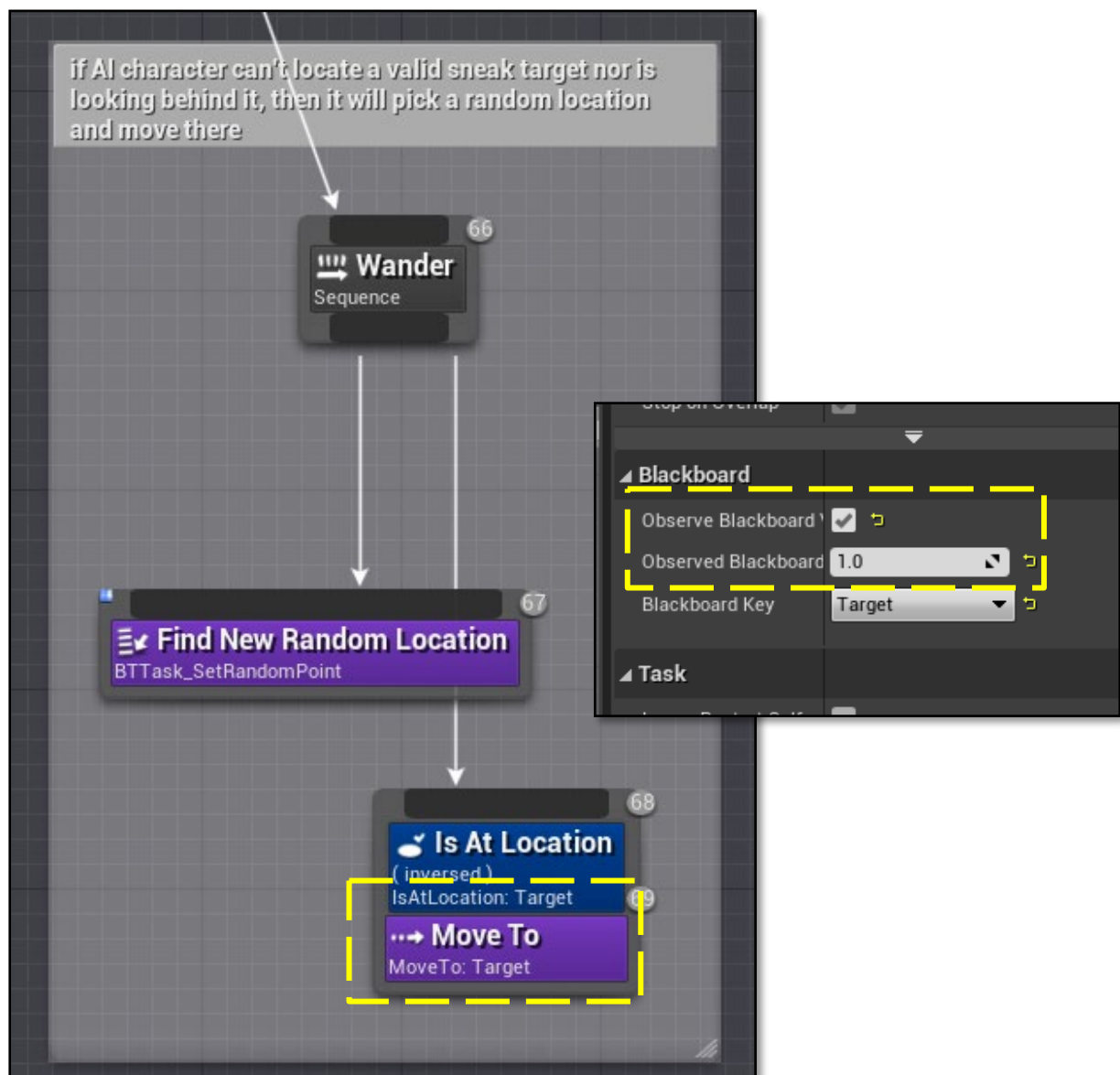




Modified Behaviour Tree: Second Stage

## Modified Stage 3

Not much was changed in this stage's BT. The 'Move To' task which ran if the AI couldn't find an enemy target made the AI character go straight to a random new location, before starting to look for enemies again. This made it seem strange during game play, because the AI, while moving to the new location, would come across potential targets. But it ignores them completely until reaching the target location. Since the EQS service on this stage's branch was continuously running, the Move To task was set to constantly observe the blackboard value set by the EQS. This way, while the AI character is moving to a new location, if it happens to cross a new valid enemy target, it will attempt to attack it instead.

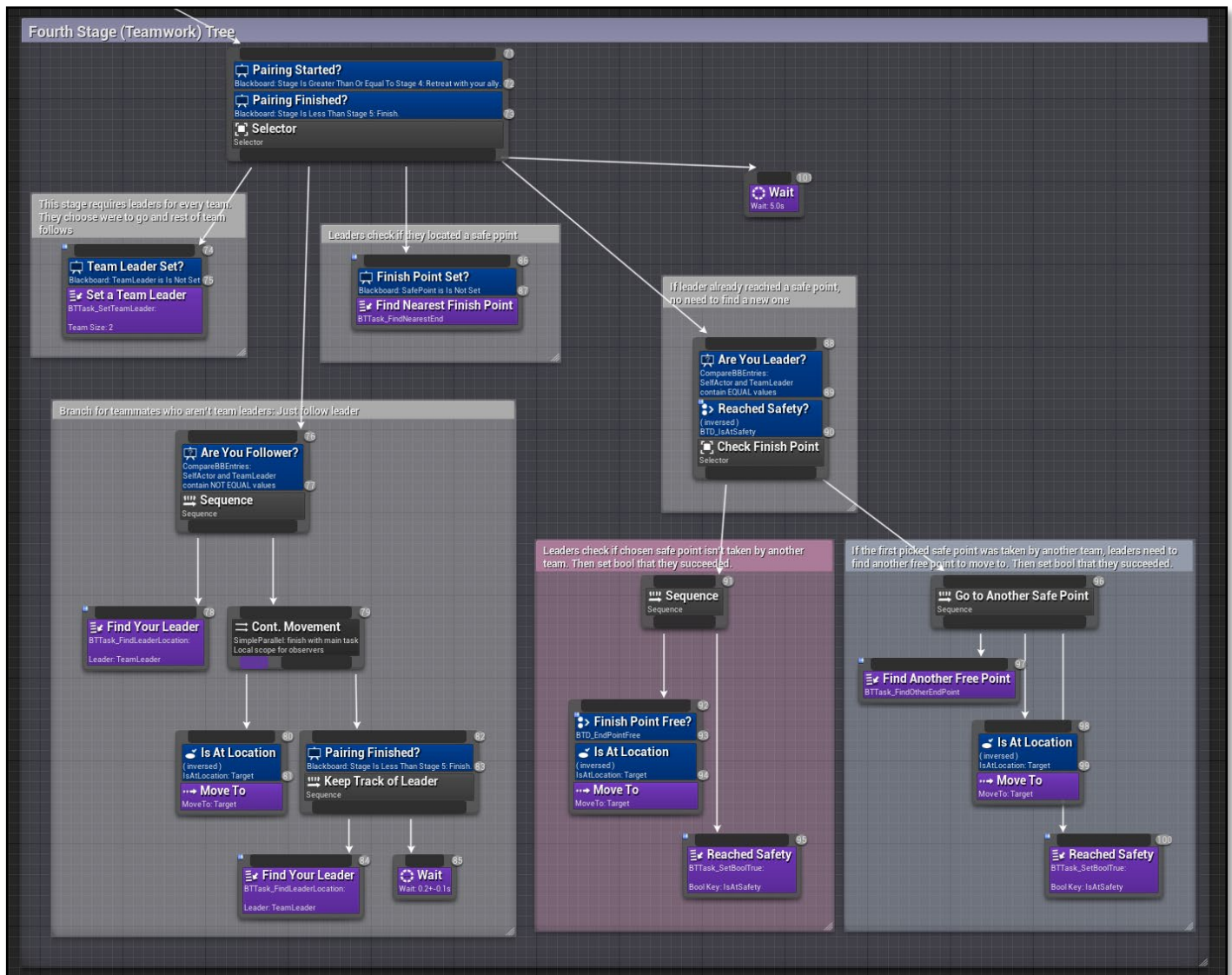


Modified Behaviour Tree: Third Stage

## Modified Stage 4



In the initial BT, each AI-controlled team would go to a common end point (safety point) out of the available 4 end points. Teams ignored whether other teams were also at the same point or not. If all AI characters were on the same side, this behaviour wouldn't be an issue and would look fine to a human player. However, if AI teams were opponents or for some other reason could not share the same end/safety point, this behaviour won't make sense. The modified BT was made (along with additional Tasks and Decorators) so that each team will have a leader, and the other team's AI characters would follow that leader. After each team leader picks an end point, they will have to check whether that point is also picked by another team (the first team leader to pick a point gets to have it and mark it as occupied). If the point is free, they would move to it along with the rest of the team. However, if the point was taken by another team, they will search for a new end point that isn't occupied by another team and go to it. The related tasks can also work for teams of any size. The only current downside is that any AI on a human player's team will be their own leader and are highly likely to not stand together at one end point; it's hence recommended that, if team sizes are bigger than 2, AI characters set the human player as their leader and follow their decisions.



### Modified Behaviour Tree: Fourth Stage

## Team Members

- Anastasiia Tcepeleva
- Daniel Gumnikow
- Ahmed Mohamed Said Anwar ElShenawy

## Acknowledgments

- [AI for Games video playlist by Roman Gorislavski](#)