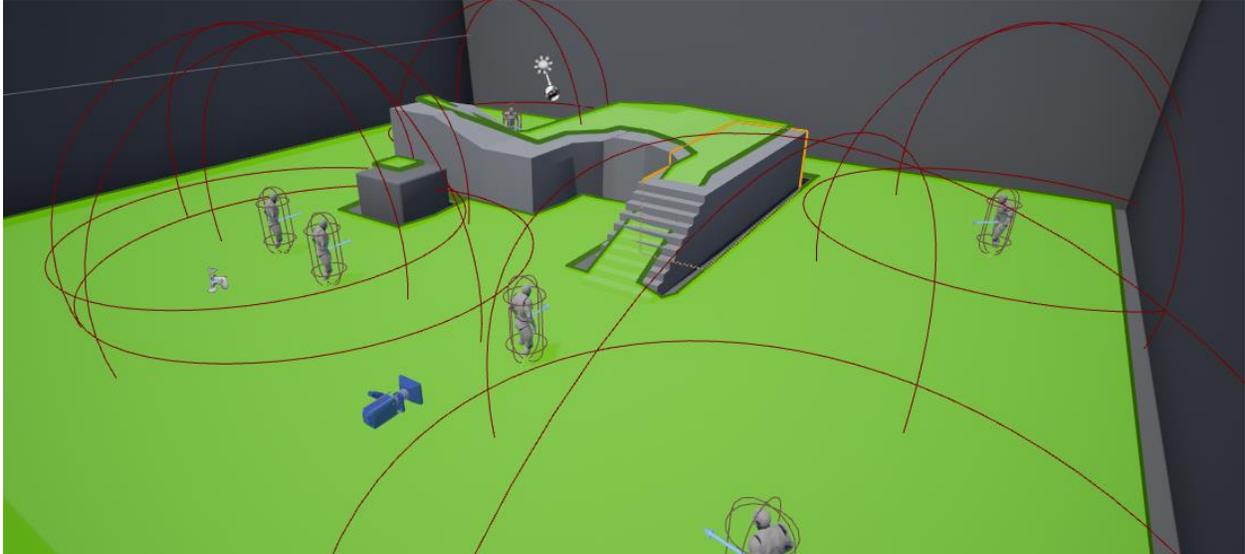


# Andreas Nordbø

3D Game Design - Noroff Stavanger 2017

Artificial Intelligence (ARI)



## Links and info:

Link to gameplay video:

<https://www.youtube.com/watch?v=L5t0J2fuT4Y>

Blogpost for this project:

<http://andreasgamedesign.com/artificial-intelligence-ari/>

My blog:

<http://andreasgamedesign.com/>

Unreal Engine version:

4.14.2

Reflection on next page!

## AI's Intended behavior:

### Chase and Patrol AI:

- Roaming around when the player is not close
- Chases player when in range
- Goes back to roaming/patrolling when player is out of range

### Fleeing AI:

- Flees when seeing player, runs to optimal hiding spot

## Reflection:

For this assignment that had a span over two weeks, our aim was to understand and create a simple but effective AI using Bipedal "bots" using Unreal Engine 4. The task required me as a student to make sure the AIs in the level had the functionality to patrol an area and reacting if it senses any form of threat. Basically, the way I decided my bots to work was to first make them move around in the level by walking towards certain locations, stopping in-between for 1-3 seconds (randomly generated) then continue to next spot. The reason I decided to make the "idle" phases random duration was to make it look more realistic, so every bot does not stop the same time when having multiple bots in the level. If the player is close range to any of the bots, the bot then chases the player and increases run speed. The bot also stops chasing the player after getting out of range/vision, it then returns to roaming around again.

The tools I ended up using was behavior trees, blackboards, navigation meshes and AI controllers combined with blueprints, to make everything work together as a whole.

At the start of this assignment I created a unreal engine project (using Unreal Engine 4 - 4.14.2) using the included third person template. Then I added a navigation mesh that the bots use to determine where it is possible to walk/run. Later I duplicated the "ThirdPersonCharacter" blueprint with mannequin, collision capsule, movement and animations included. I removed the camera arm, camera itself and all the nodes from the Event Sheet that I did not need to use. Then I set up a simple AI Controller blueprint and added this to the AI Character as an AI Controller Class. Later, I created a blackboard where I stored the variables destination (as vector, used to decide where to walk) and "isThreat" Boolean to determine if the bots sees the player. I made two simple Blueprint Task blueprints, one to calculate where to move when roaming, I called this "RandomMovement". The second Task class I named "Threat" this class was used to determine movement from bot to player. In the end, I made a behavior tree that acts as a Finite State Machine, this is the "brain" of the bot that controls everything I just described I've set up in terms of blueprints, and so on.

In the last week of this assignment I decided to add a second AI, utilizing EQS (Environment Query System) and sensing systems (AI Perception). I decided that this AI should behave different than the first AI, so I made him use EQS to find the optimal hiding spot away from the player (ThirdPersonCharacter). It took a bit more time and effort to get this working, but I'm happy with the result I ended up with, I also learned a lot from doing this. I did some experimentation with "hearing sensing" but ended up scrapping it for this project as I found it unnecessary.

I encountered an issue where the bots were not able to get up the stairs in the standard third person blueprint. The way I solved this issue was to move the stairs a bit until the navigation mesh did not cut off in the middle of the stairs, this issue seems to be engine related. I wanted to implement a system that the bots jumped off ledges, but found this way too difficult and out of the scope of the assignment so I decided to not implement this feature. I would've been cool to add some sound effects, but I decided to focus more on functionality on this assignment.