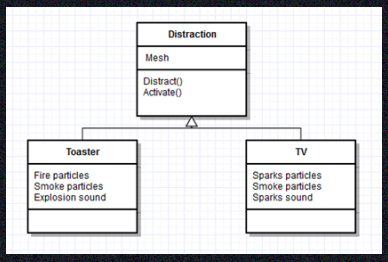
STANDBY

Distraction mechanic

The player can use the phone to activate objects in the house to make them malfunction, the AI will then prioritise the malfunctioning object, move towards it and fix it. This mechanic is used to give the some control over the AI by sending it somewhere which is essential to avoid capture.

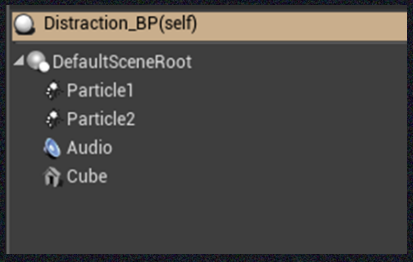
Microwave Distraction

Inheritance was used due to the many intractable distraction objects in the game. All the distractions are children of the parent distraction. The children could then be adapted to fit the specific distraction criteria.

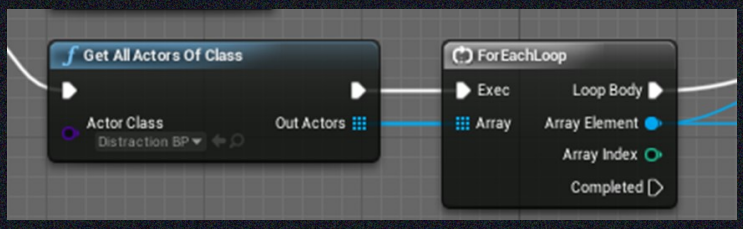


Inheritance diagram

A user could create a child of the "distraction" blueprint and then change the individual components. This method allows easy implementation of new distractions as well as giving users freedom to change the particle, audio and mesh's to their liking.



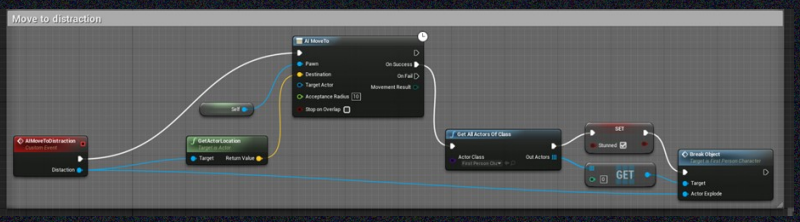
Distraction components



Get all distractions

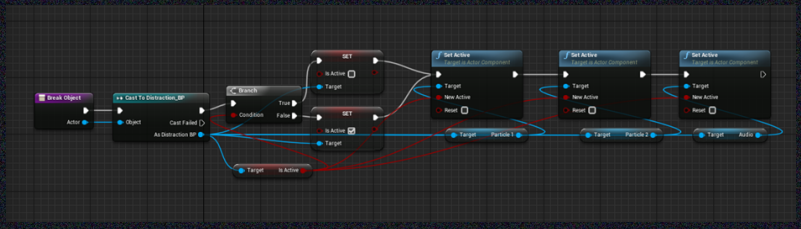
When adding a new distraction none of the code would have to be changed. This was achieved by creating an array in the code to get all actors of the class "distraction" (which also applies to the children of this class) and then loop through each to see if any have been activated.

The AI will then move to any distractions that have been activated, wait for a few seconds while fixing the object and then deactivate them using the same code that was used to activate them in the first place.



Move to distraction code

Continuous integration was used during the development process with the code being uploaded regularly. This was useful for the team to be able to see the results fast, find bugs and make suggestions during development.



Toggle active code