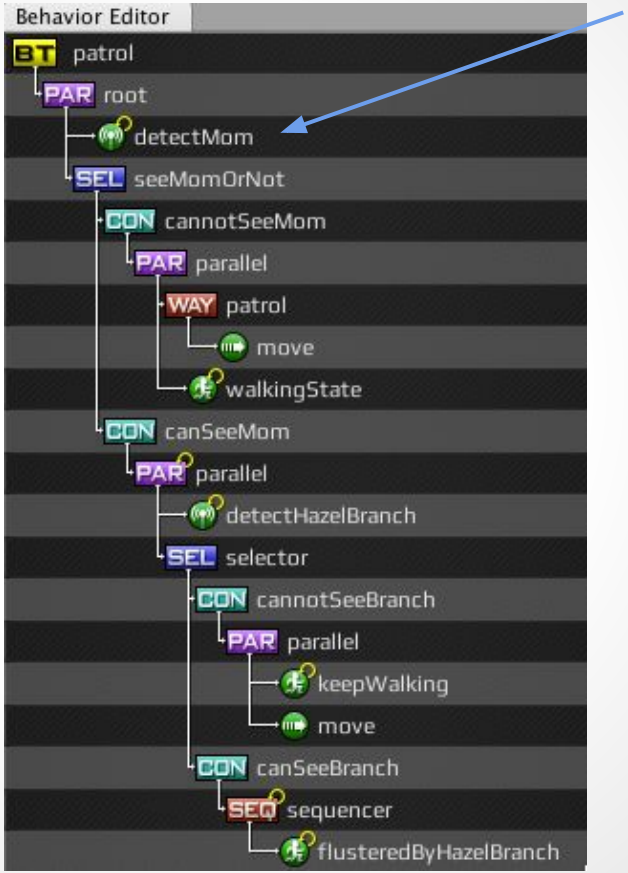
AI Breakdown

Tremaine Eto

CS188

The father (GameObject Dad_Unity) is our AI agent. The RAIN AI child is parented by him, and thus you must expand his GameObject to see the following Behavior Tree:

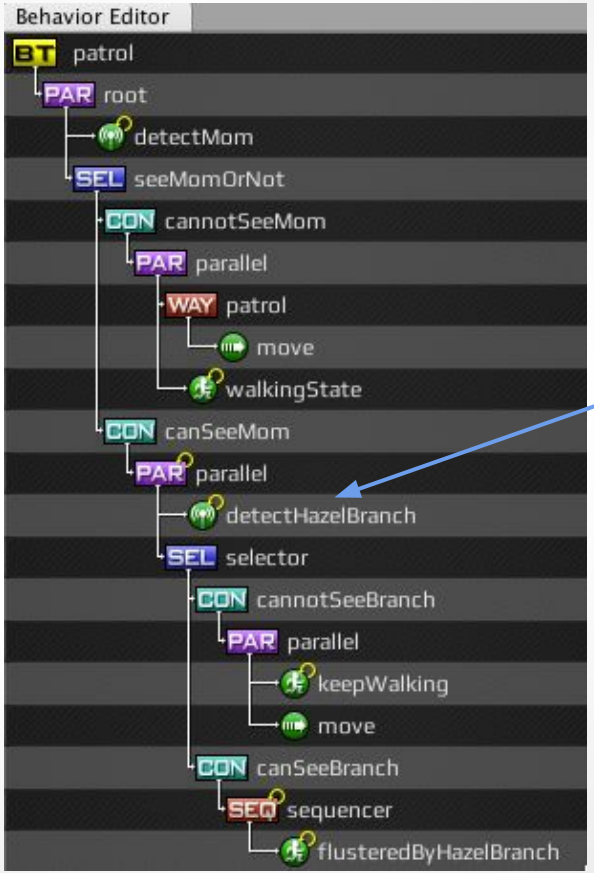
How does it work? Essentially, I wanted the AI to work into the story as much as possible. I didn't need any extraneous novelties, in other words. Thus, the AI had to center around the idea that the father is evil and you have to outsmart him.



The idea is simple: when the father sees you (the mom), he walks towards you ominously. You're not supposed to let him touch you. If you're out of his visual sensor (detectMom), he'll go back to his waypoint route.

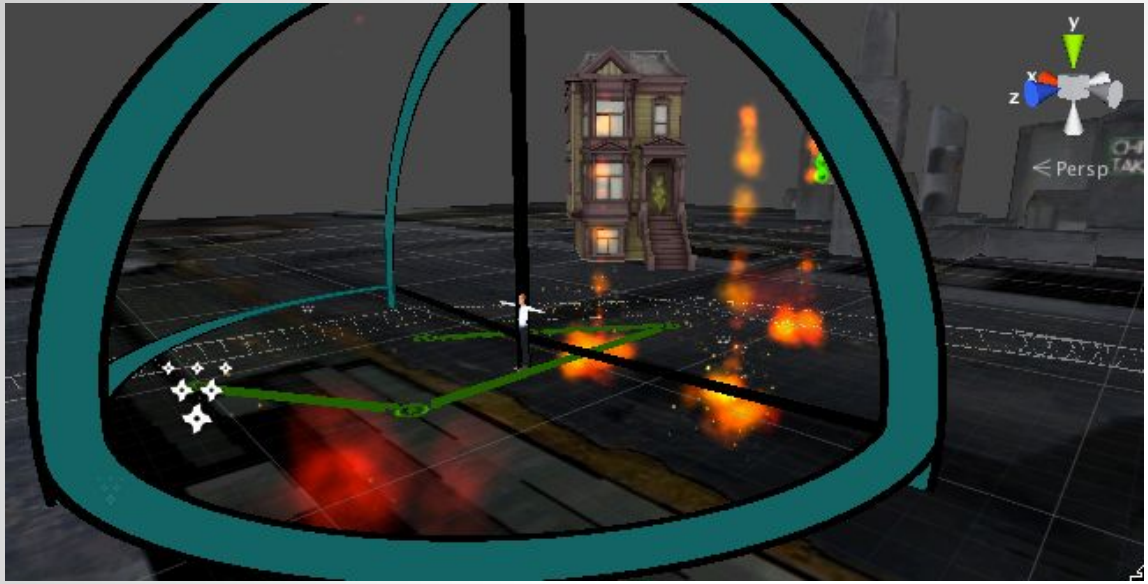
The father (GameObject Dad_Unity) is our AI agent. The RAIN AI child is parented by him, and thus you must expand his GameObject to see the following Behavior Tree:

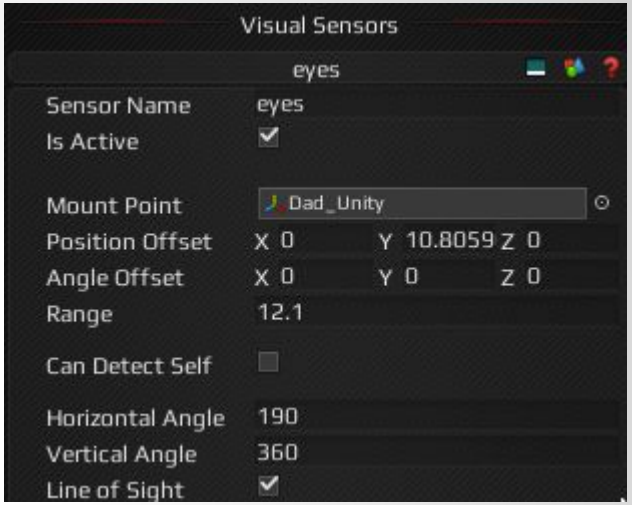
The detection of you, the mom, is only a part of the AI. I wanted the Hazel-Branch to be the central element of his AI. Thus, I have you as the first person mother go and grab the Hazel-Branch from the San Francisco house located in the nightmare scene.



Once you have the Hazel-Branch, the AI agent can detect it as a GameObject. If the father sees you with the Hazel-Branch, he suddenly is paralyzed, and his flustered animation fires. He no longer walks about the waypoint route. Thus, you can pass him with ease.

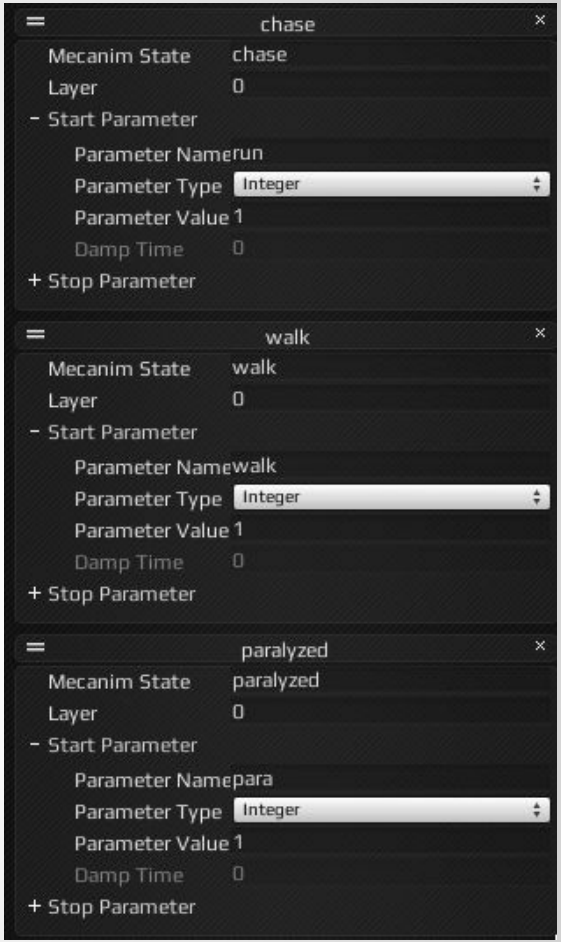
For the behavior tree to even work, I had to attach it to a body, which is my GameObject of Dad_Unity. Then, I had to attach the behavior tree itself to the RAIN AI GUI.

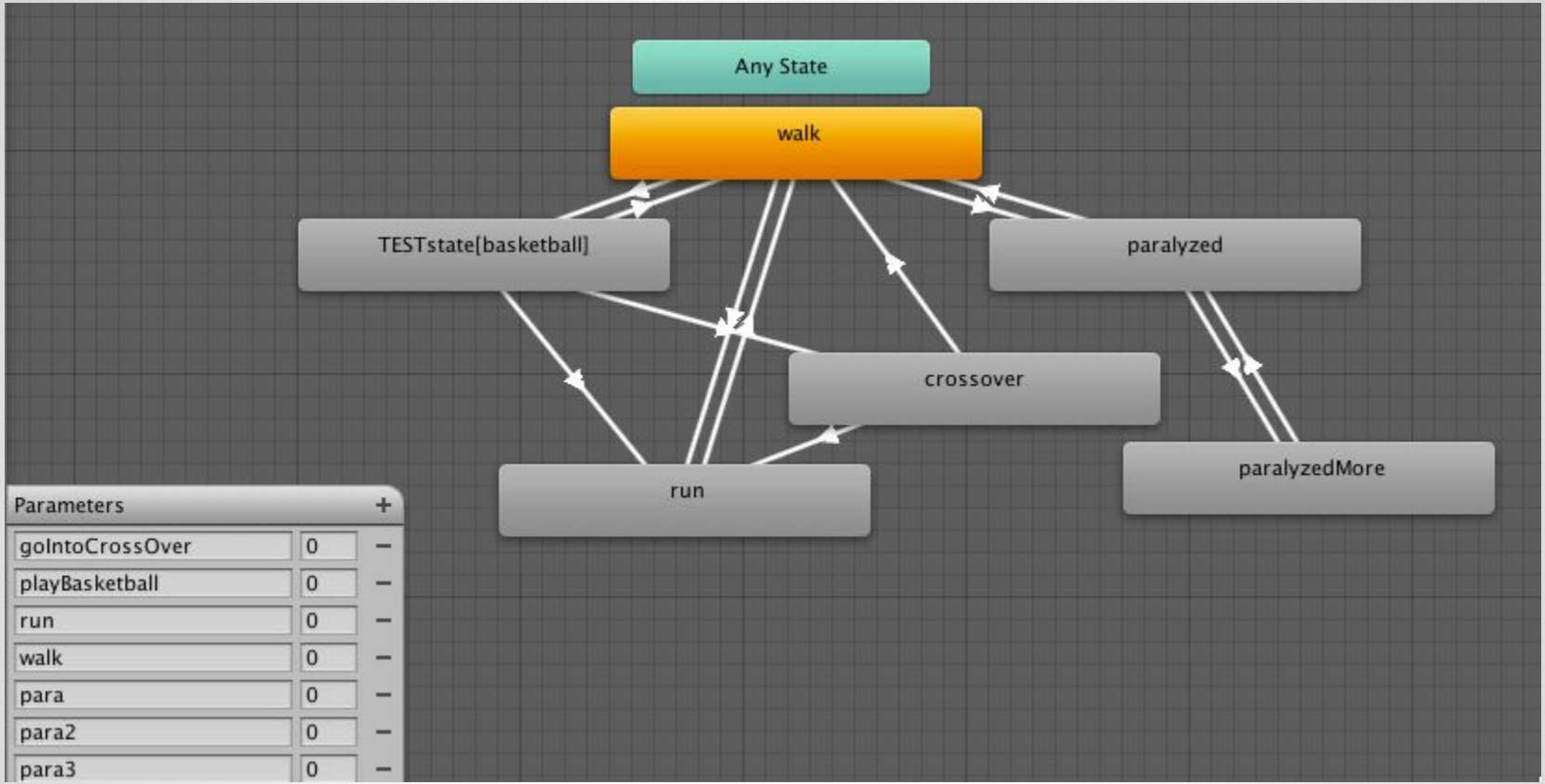




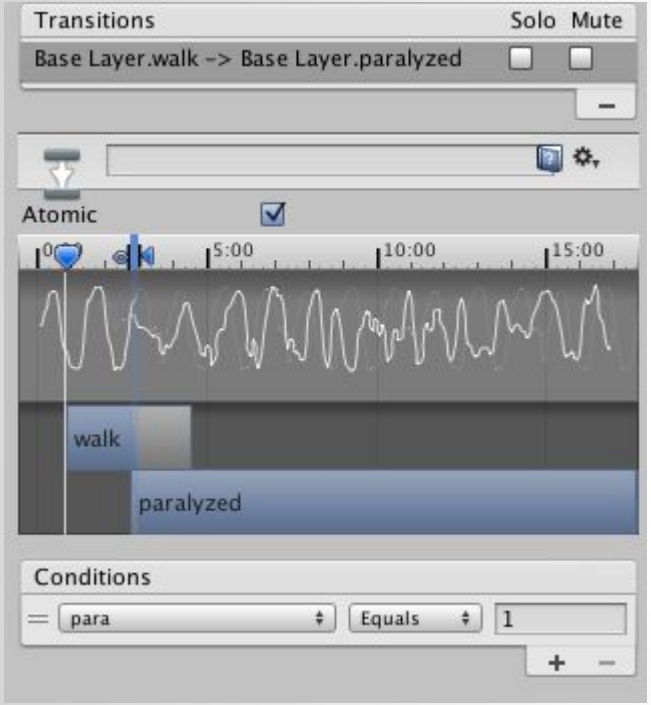
Next, I had to create my father character's visual sensor with the appropriate settings. I made his peripheral vision pretty impressive: 190 degrees. I didn't want him to see you all across the scene (for reasons such as the flames and smoke blocking his view) and so his range is about 12.1. This makes gameplay not too frustrating as you can be decently far from him but also don't have to get too close for his AI to fire.

This is the part where it gets quite interesting, and it's figuring out how to get the AI to interact with the mechanim state machine I created for my father character. You have to select "MecanimAnimator" in the "Animator" dropdown, and you can add Animation States according to what you already have in the Animator Controller on that particular GameObject. Then, you can access the parameters you defined for your Animator Controller and change the values accordingly. To make things easier, I set all my parameters to integers and made them fire when I set them to 1. The same effect can easily be used with a Boolean.





The “debugging” part that can cause the most issues is the construction of the mecanim state machine. This is completely separate from the state machines I made for earlier scenes, which only dealt with animations. As can be seen, parameters are defined on the transitions to know when to go from one to another.



Here is an example of a parameter binded to a condition. As you can see, if the para parameter equals 1, then we move from the base layer's walk state to the base layer's paralyzed state. The RAIN AI then can fire and set the para parameter to 1 and thus navigate through the mecanim state machine accordingly.

Those are the general steps I took to implementing the AI!
It is not the easiest thing to bind the mecanim state machine to RAIN AI, but using this method is a great way to get it going.
To see more details on how I did it, please click through my Unity project file and see for yourself.