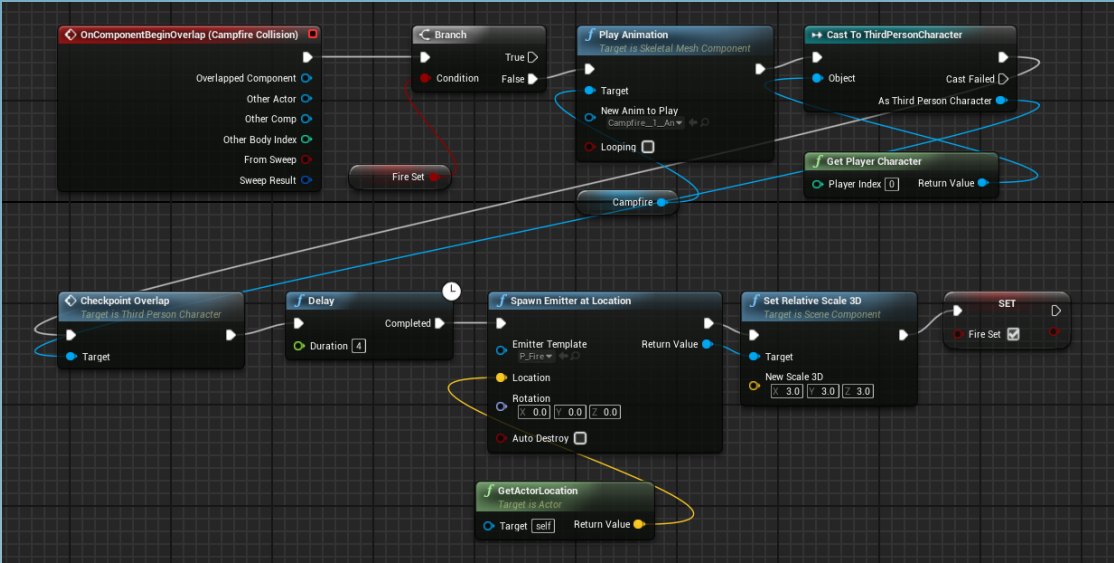
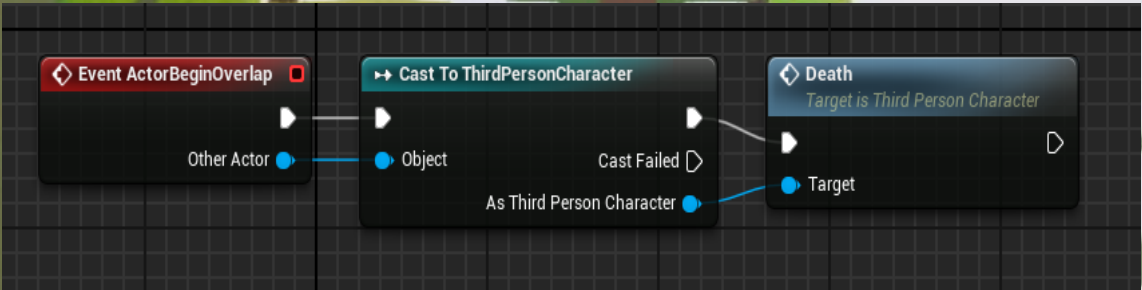
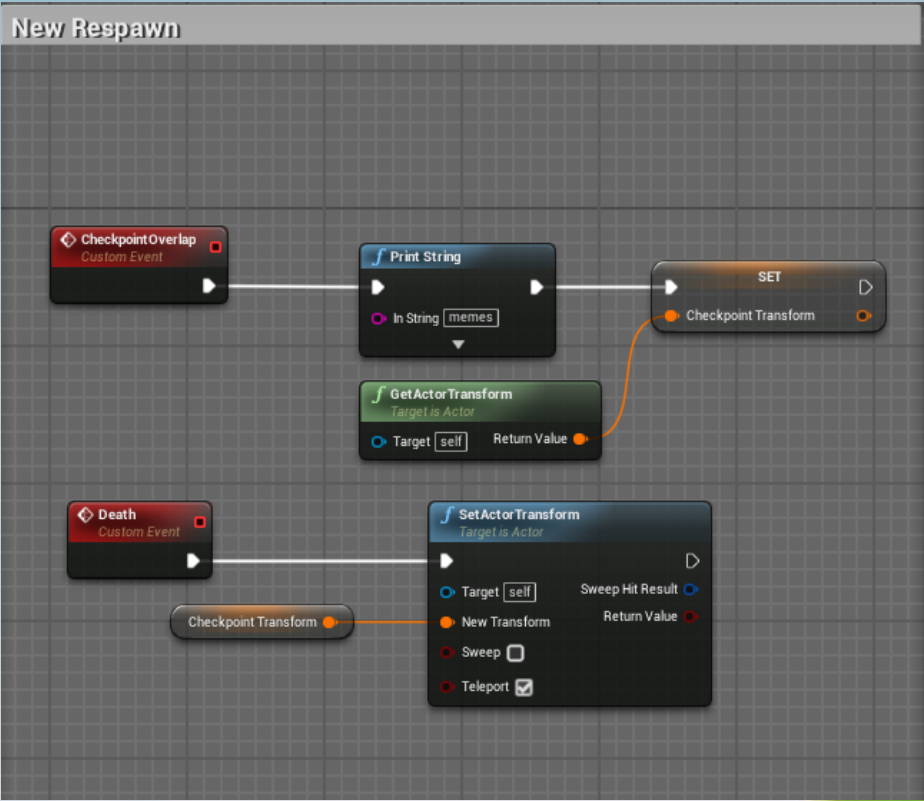


# Player Respawn System - Jack Maber



To make sure the player could resume progress after some of the harder puzzles, I implemented a respawn system so that players are teleported back to a marker when they die, this blueprint is part of the “campfire” blueprint, which is the progress marker that the players make contact with to save their progress, and as can be seen in the image, when the player overlaps the model, it checks to see if that fire has already been triggered as not to reset the animation and spawn location, if it hasn’t, the unfolding animation is played to let the player now that it has been set, the “checkpoint overlap” custom event is then cast to the player which sets the respawn location and saves their progress, a delay allows the animation to finish and then an emitter is spawned to create fire in the campfire as another way of letting the player know it has been set, and thanks to my use of OOD, the fire set Boolean for that particular object is then set to true, meaning it can’t be reset again by the player hitting it.

This part of my system was inside the player blueprint and was mainly to do with setting the respawn location, and when the player dies. As seen in the last part, when the player collides with the checkpoint, the custom event is triggered, which sets the pre-existing “checkpoint transform” vector in the player character blueprint, which saves the player position as it is the same as the checkpoints. The death custom event is called when the player makes contact with one of the trigger boxes in the world, these are placed in areas such as the water below the jumping puzzle, and when the player makes contact with them, they are simply teleported back to the last campfire they touched, allowing them to carry on from their last major point of progress.



This part of the system is part of the trigger box blueprint, and it all it does is when the player overlaps with it, the “death” custom event is cast to the player, which will then teleport the player back to the last checkpoint they touched. The trigger boxes all work on the same principle and simply trigger a custom event, so they utilise Inheritance, where the child copies gain the attributes of the first produced object or “Parent”, which made it much easier for me to implement the system into the game as it meant that I didn’t have to recreate the same blueprint on every trigger box that I needed, which saved me a considerable amount of time.