

Ambient Audio Implementation

1. Key Questions

- List of objects and their locations (Inside, outside, movable, etc)
- Will there be dialogue or music emitted from objects? (Radio, TV, hologram, etc)
- Are these objects important to the story or objective? If so, which ones and how?
- Can these objects be interacted with? (turn on/off, change station, set different dials, etc)
- Do the sounds of the objects act as cues to guide the player in specific directions? Should they?
- Are there specific features about the environment that would make it different from a regular Earth swamp? If so, do these features change from dawn, day, dusk, or night?
- Are there creatures or insects featured in this environment that could add to the ambience? Do they have a day/night cycle?

2. Implementation of Reverbs

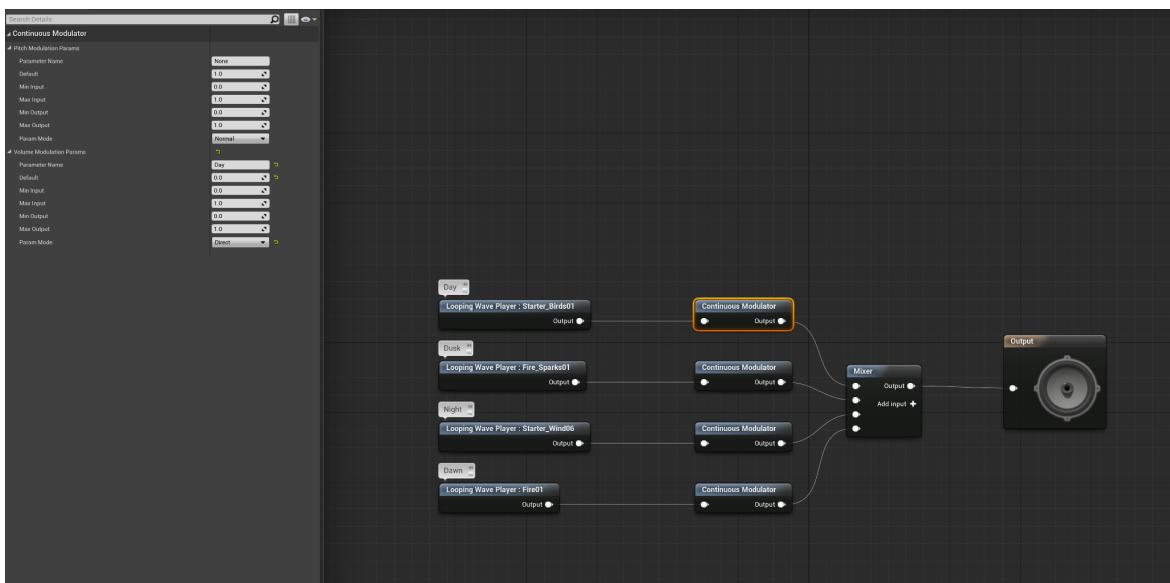
- Create an Audio Volume in the map and alter it to cover the entire outdoor portion of the level and place a Reverb Effect for the swampy wooded area.
- Next, create individual Audio Volumes for each of the buildings in the level and place Reverb Effects for those. Set their priority levels higher than the “outdoor” Audio Volume.

3. Sound Emitters on Objects

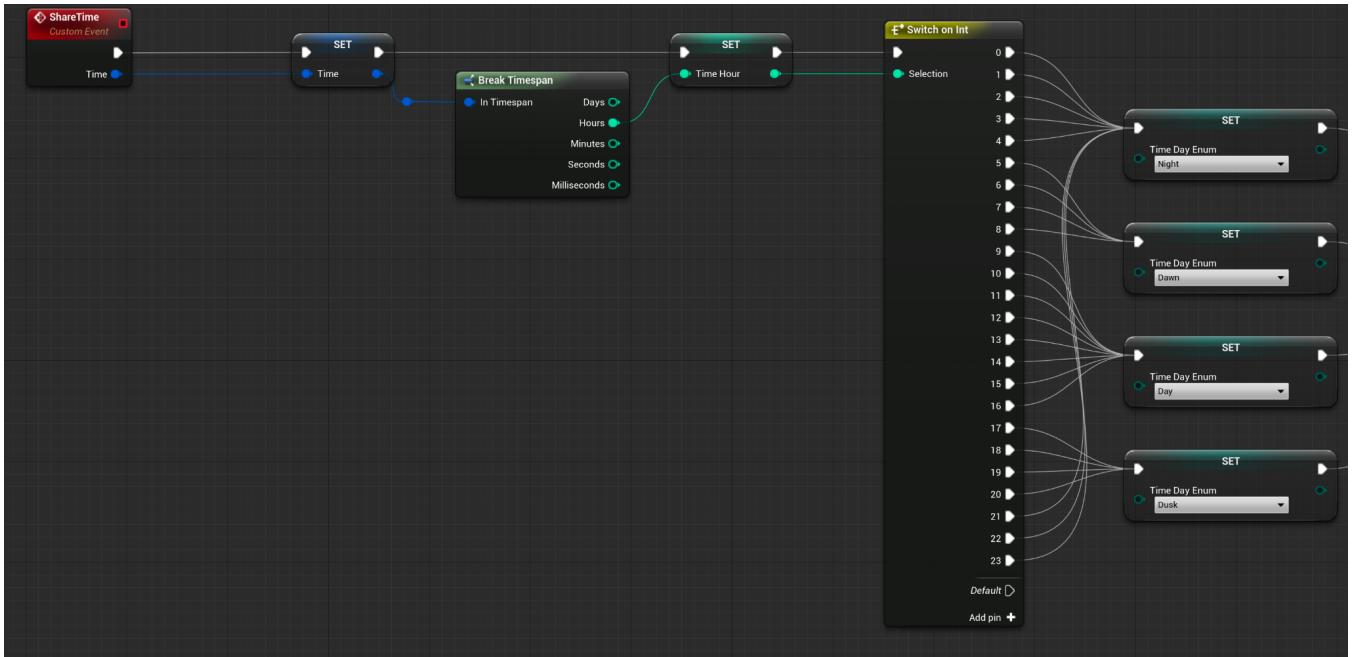
- Place looping sounds on each of the objects and spatialize with Sound Attenuation.
- If the object can be interacted with, implement the triggers and sound banks to represent the interaction. Ex: Interacting changes Radio station and a random song will play from the sound bank upon each interaction.

4. Global Ambience

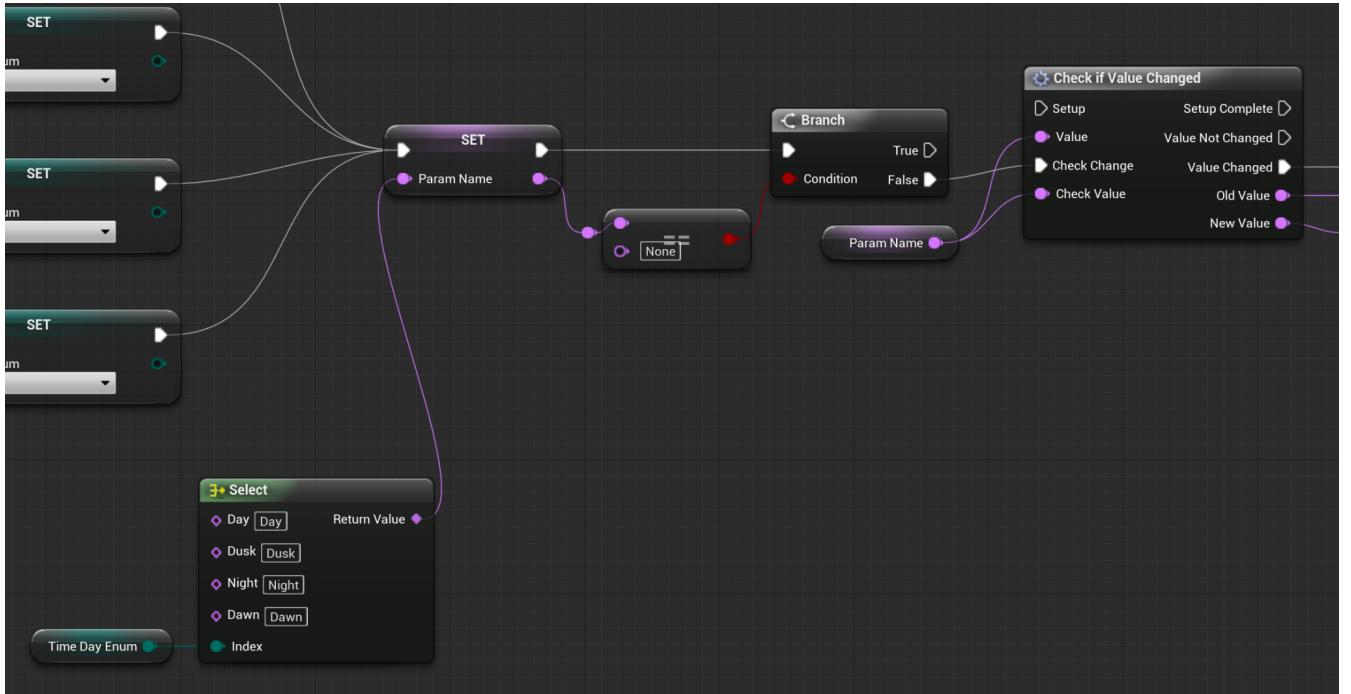
- Create multiple looping ambiances representing Dawn, Midday, Dusk, and Midnight in a Sound Cue. Run each through a Continuous Modulator and give each Volume modulation a parameter name.



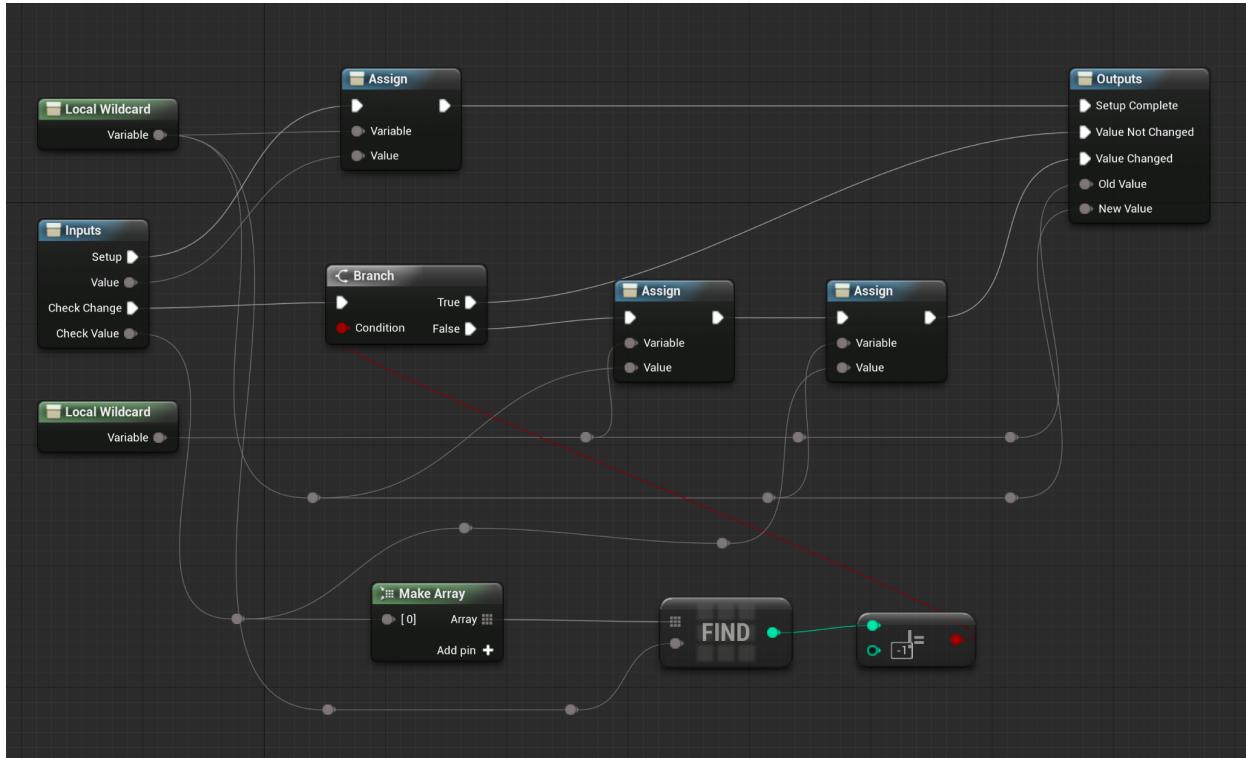
- Create a Blueprint and integrate the time parameter from the Level Blueprint into your Blueprint. Create an Enum to represent each of the Time of Day game states and set them accordingly as a variable.



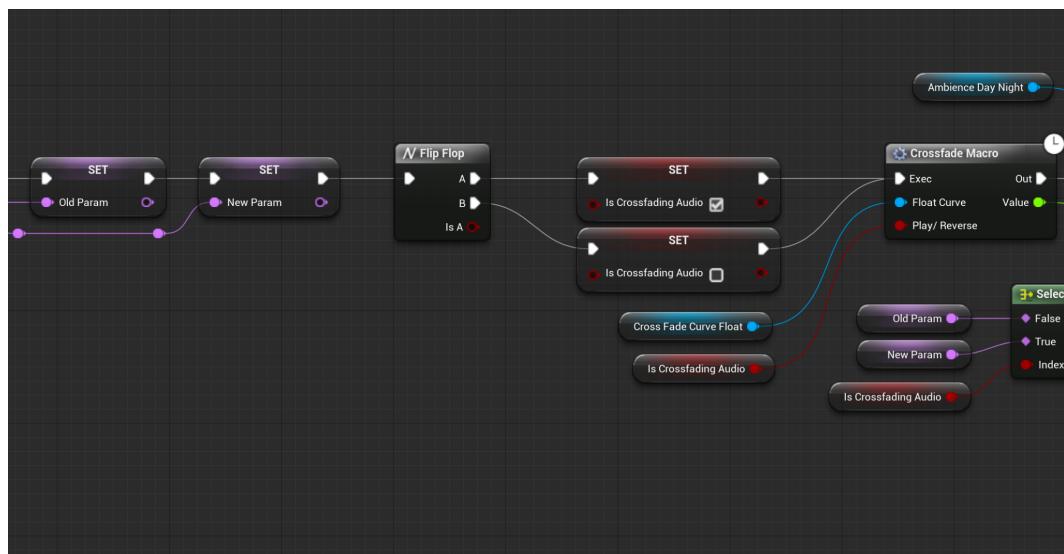
- Convert the Enum output to Name using Select and set as a variable.

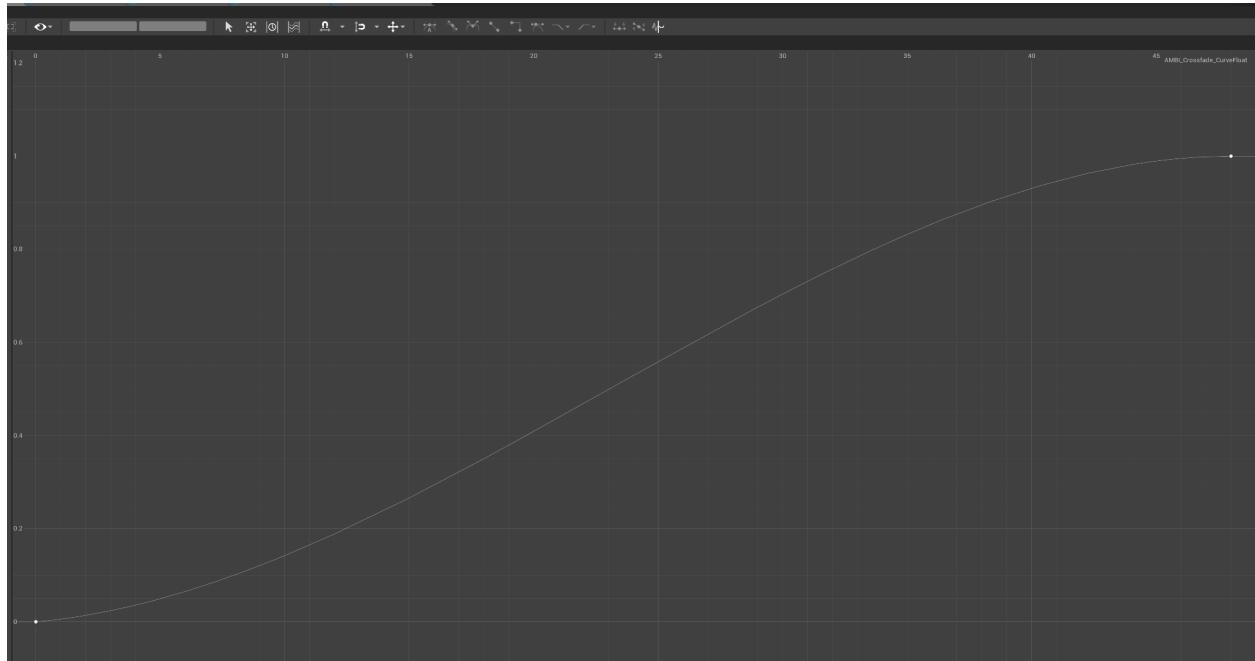


- Check if value has changed and output both parameters: “Old Parameter” (what it was changed from) and “New Parameter” (What it was changed to).

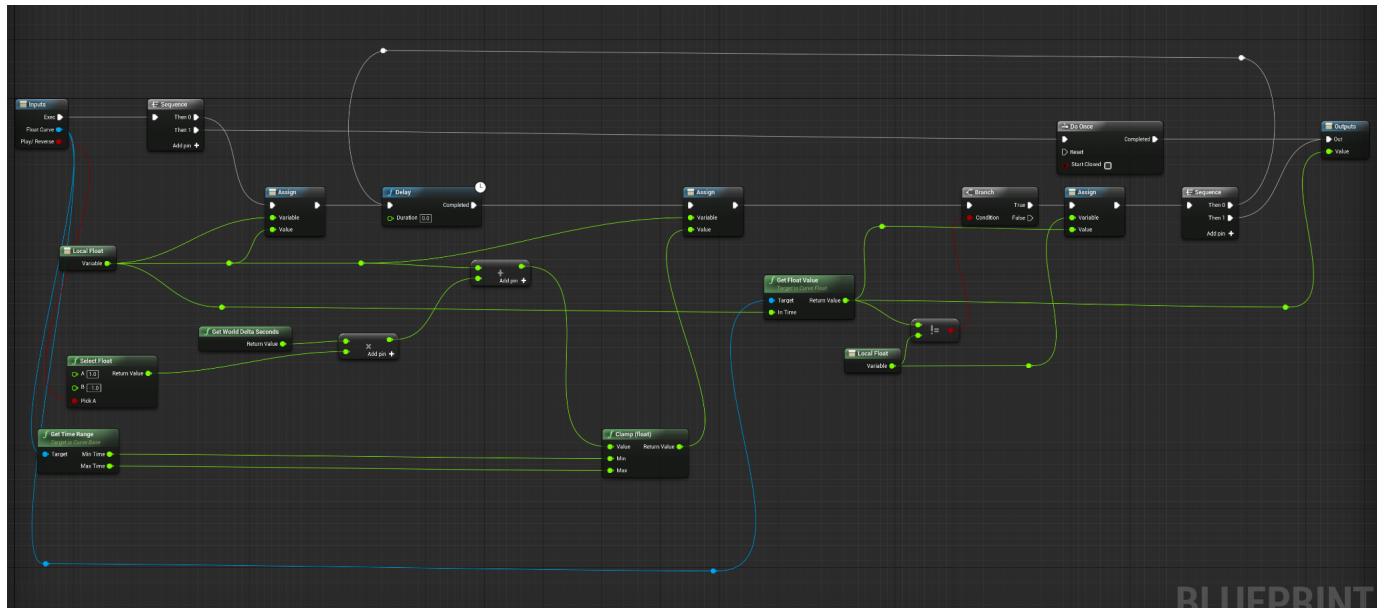


- Move these Name parameters into their own variables. Then create a CurveFloat for the Crossfades. For a 30 min day where an hour is about 75 seconds, anywhere between 45 seconds and 90 seconds would be a nice and easy transition that would be barely noticeable--kind of like how you don't really notice the sound transitions in the real world. Perhaps, longer/shorter depending on how quickly it goes from light to dark. Ideally, the sounds of Dawn and Dusk have elements of Night and Day to make the transition even less noticeable. This can be altered and massaged until desirable results have been reached. (In my example, the curve is 48 seconds long)

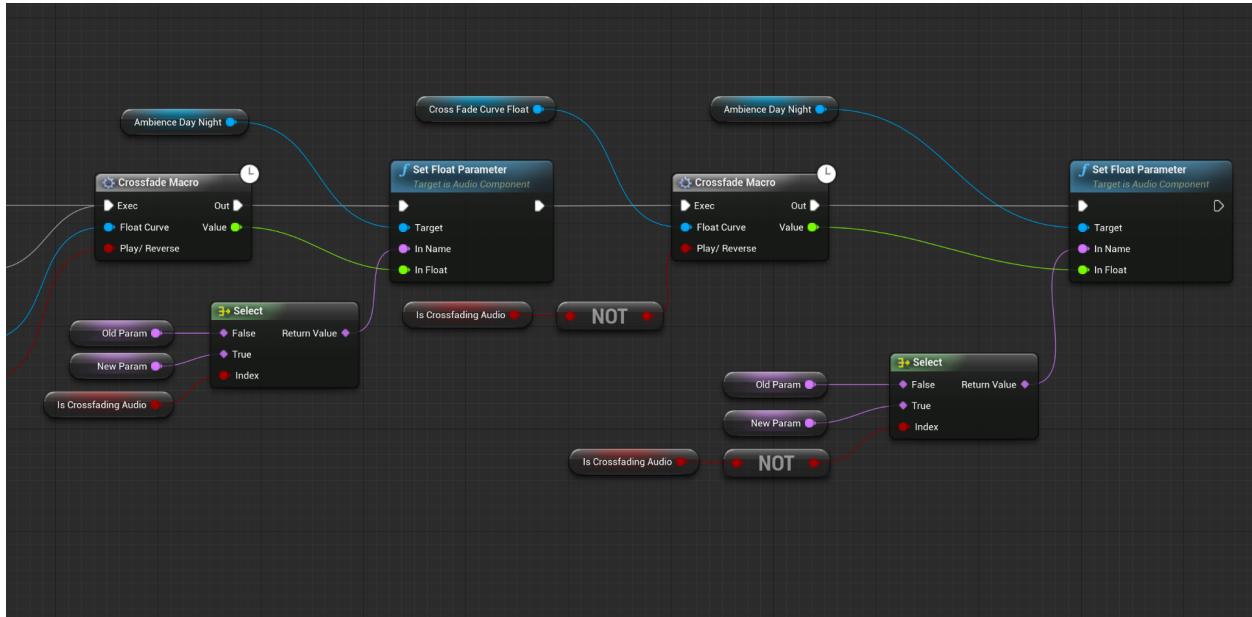




- Create a system that can play the cross fade both forwards and backwards depending on the boolean. Below is the macro that I created.



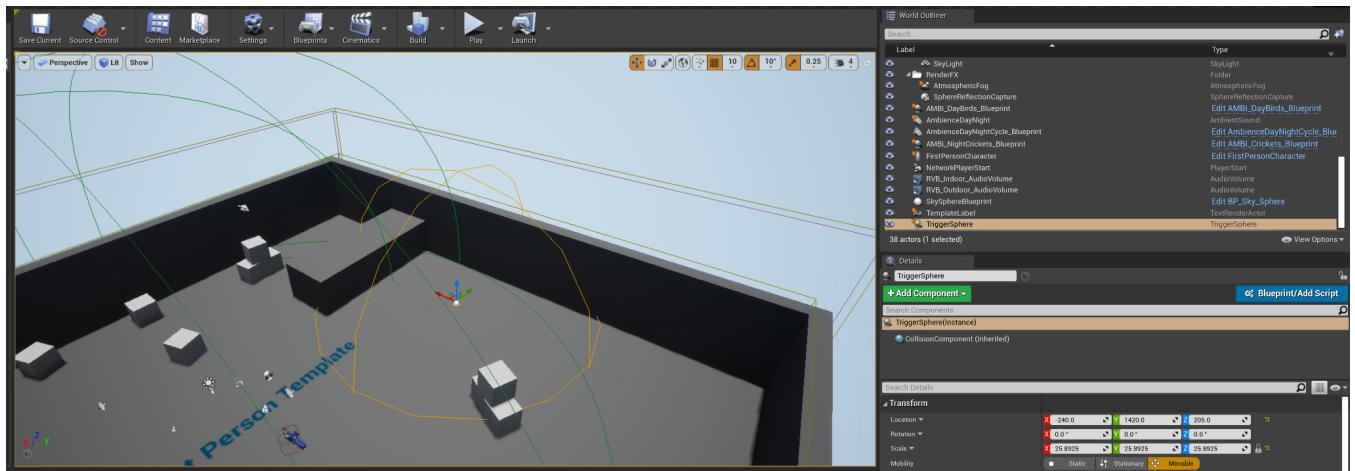
- Finally, use the accumulated variables and parameters to modulate the Volume Parameters in the Sound Cue. For the New Parameter, play the crossfade forwards. For the Old Parameter, play the crossfade backwards.



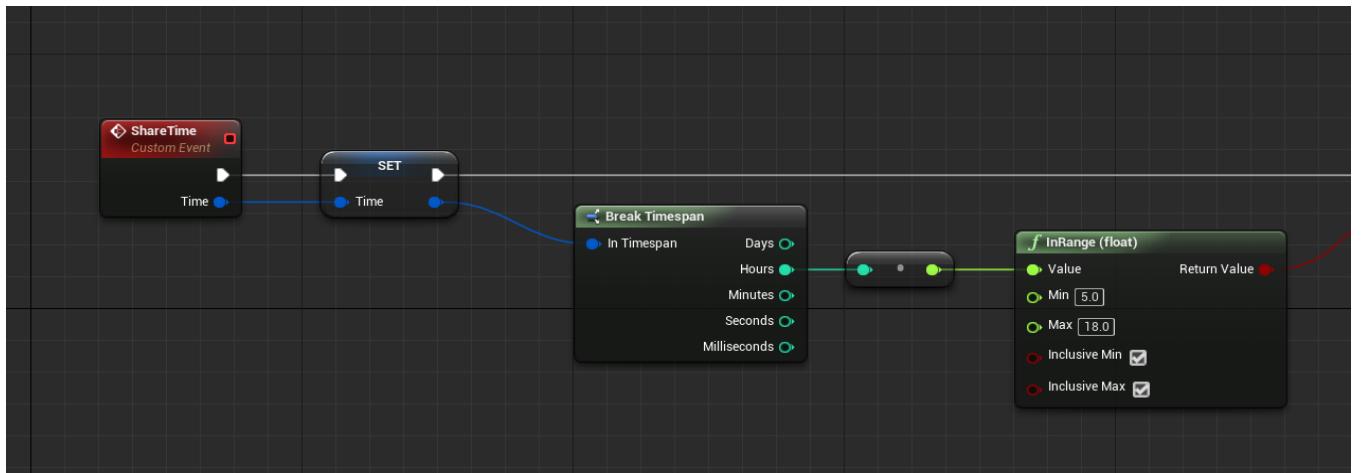
- I would probably make certain variables public so it is easier to manipulate and test, such as swapping out the CurveFloat

5. Environmental Stingers

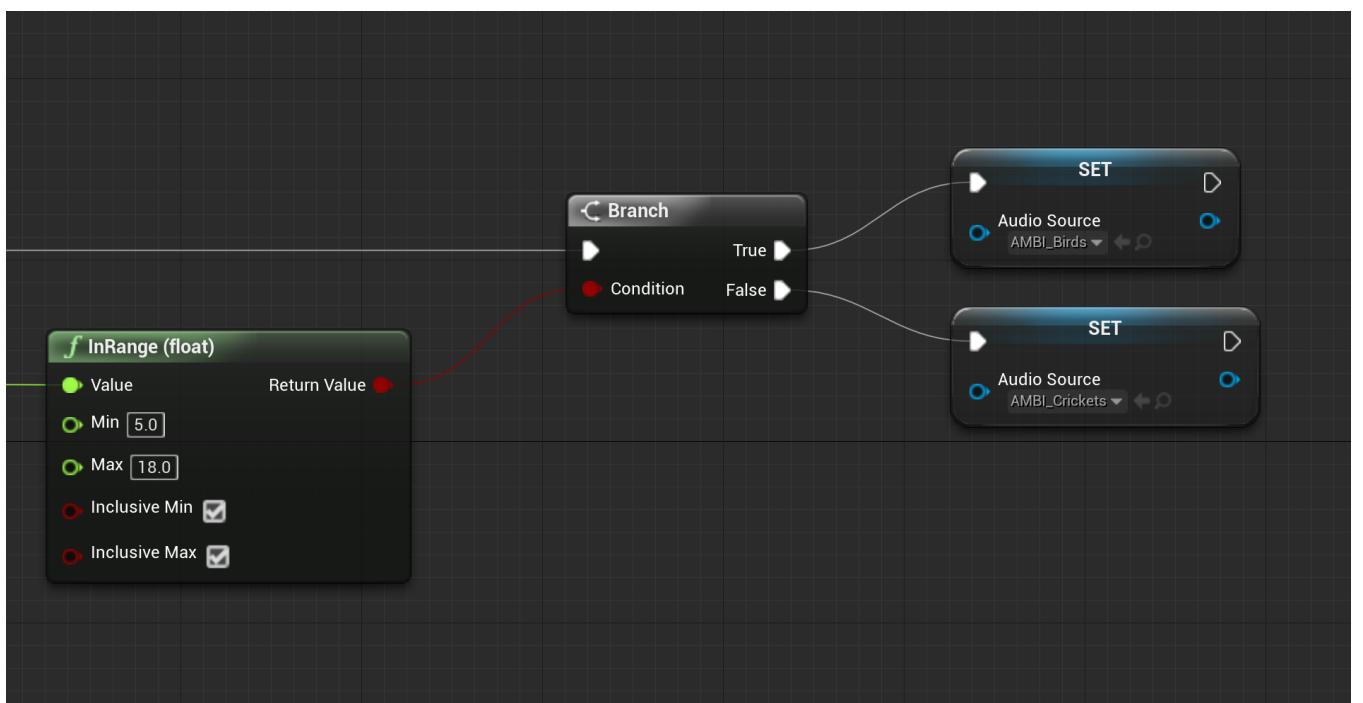
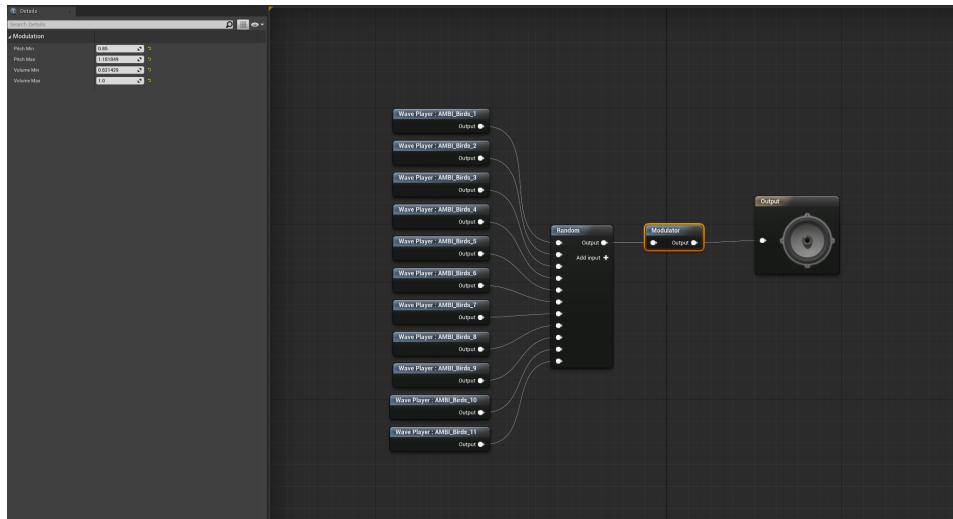
- Create Sphere Triggers in the Level Map where there are to be environmental stingers to play in the world.



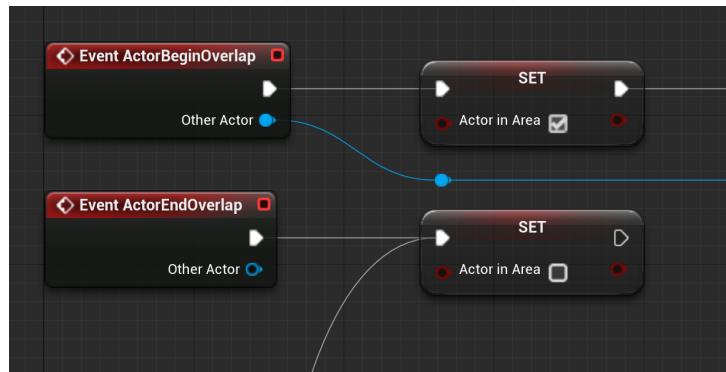
- Add Blueprint. Integrate the time parameter from the Level Blueprint into it and use the InRange to determine at what times of day these stingers should be firing.



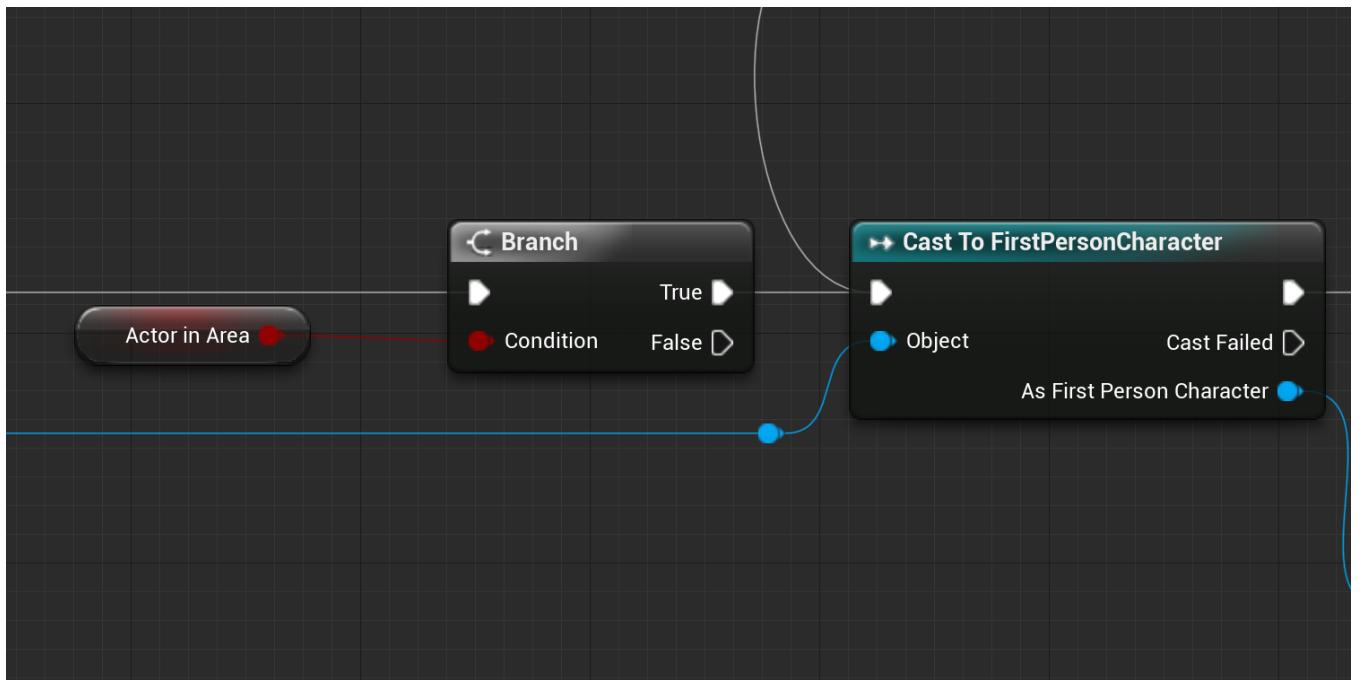
- Create SoundCues (with randomization and modulation) for each time of day (in this case “Day” and “Night”) and make a SoundBase Variable in the Blueprint. Set the different sound cues to play at different times.



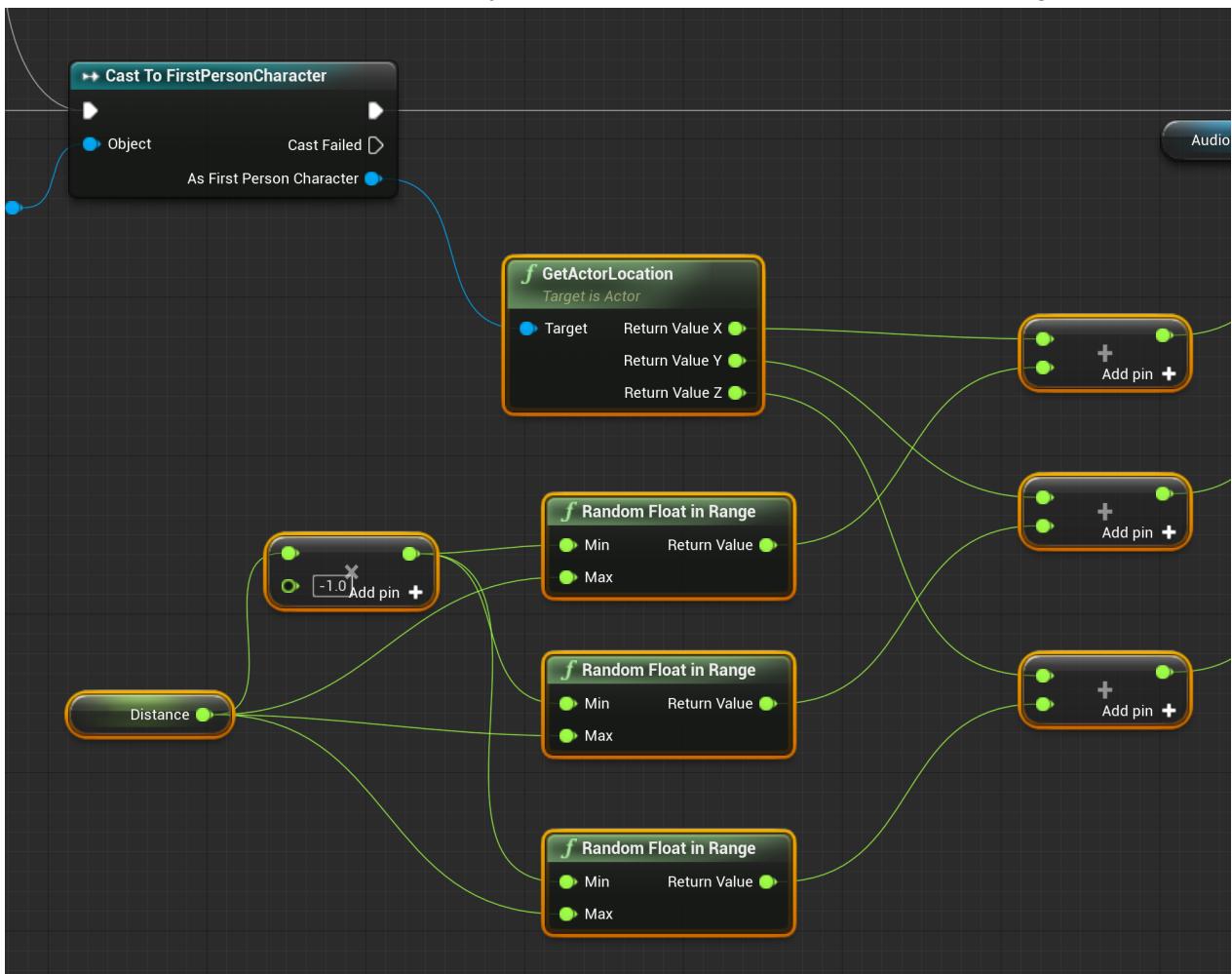
- Add Events on ActorBeginOverlap and ActorEndOverlap and create a boolean variable for “Actor in Area” for them to set as true or false.



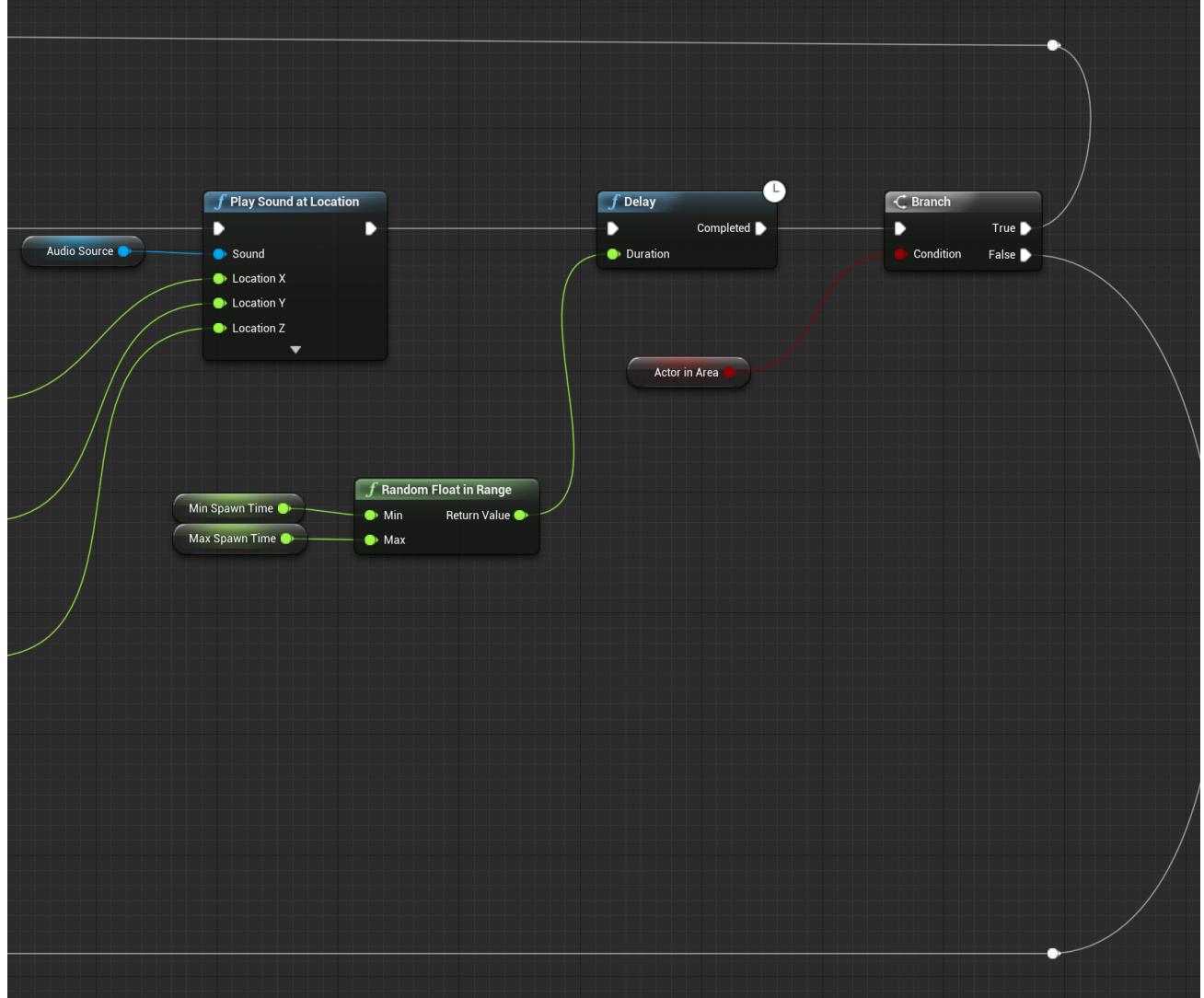
- Get Actor in Area variable, and if true, cast to Character.



- Make a Float variable set it to ~1000--that will determine the max distance a stinger will spawn (make it public so it can easily be manipulated). Plug it into the max nodes of 3 Random Float in Range and a multiplier *-1 that will plug into the min value nodes (this is so the sounds will also play on the right side of the character). Get Actor location and add the return values of x, y, and z values to the Random Float in Range values.



- Use Play Sound at Location and input the x, y, and z values respectively. Get the Sound Base that was set by the time. This will play each sound at a random fixed location around the character and should therefore be spatialized. Place a Delay after the Play Sound with the duration set by a Random Float in Range. It would be smart to make these inputs variables so they can be set outside of the Blueprint. After the Delay, check to see if the Actor is still in the area, and if True, loop back to Cast to Character. If False, loop back to set Actor in Area boolean to False.



- If a character starts out inside a sphere trigger, you could add an Event Tick to continuously check the Actor in Area boolean. It depends on how the level is laid out, it would probably be beneficial to have stingers going constantly in the level, but certain areas of the level with other stingers. Maybe birds near wooded areas, frogs near wet areas, etc. But then gusts of wind through trees going throughout the entire level.
- You could also create 2 Play Sound at Locations with the different Sound Cues already selected and modulate the Delay times depending on the time of day, such as during Dusk, as it gets later, bird stingers become more sparse and the inverse for crickets.