Dynamic Soundtrack Controller

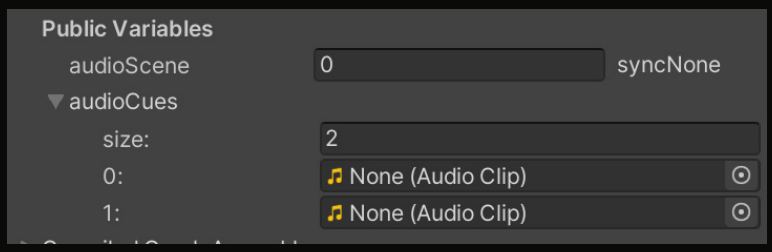
by Happy Time Institute | Made in Graphs

Concept / Setup

This is a prefab for audio scheduling that will help you create dynamic soundtracks that musically switch between sections/loops/bars in VRChat.

This system is built around the idea of **"audio scenes"** in which each scene equals a new cue for your soundtrack.

To play through your dynamic soundtrack - you can simply set/change the `audioScene` variable through any behaviours:



This system uses `dspTime` (recently exposed in Udon) for very precise audio scheduling! It handles moving between loops/bars, playing one-time transitions sections into loops, stingers, and can fire off external game events at the end of scenes if desired.

Basic Setup

- **Drag-and-drop the Dynamic Soundtrack Controller prefab into your scene hierarchy**
- **Populate the `audioCues` array with your audio clips/loops**
- **Change the value of `audioScene` at runtime + it will schedule/play through your loops automatically!**

Start Behavior

By default, Dynamic Soundtrack Controller will start playing the current `audioScene` (defaults to 0) almost immediately at runtime or when you enter playmode if you're using CyanEmu (which you should!)

This is not always ideal — in some cases, you may want to start the dynamic soundtrack late or even toggle it off/on.

To do this, you just disable the Dynamic Soundtrack Controller game object and enable it later at runtime.

Note: the system will still handle whether or not the current queued clip is a transition/stinger clip when it gets enabled (or re-enabled)

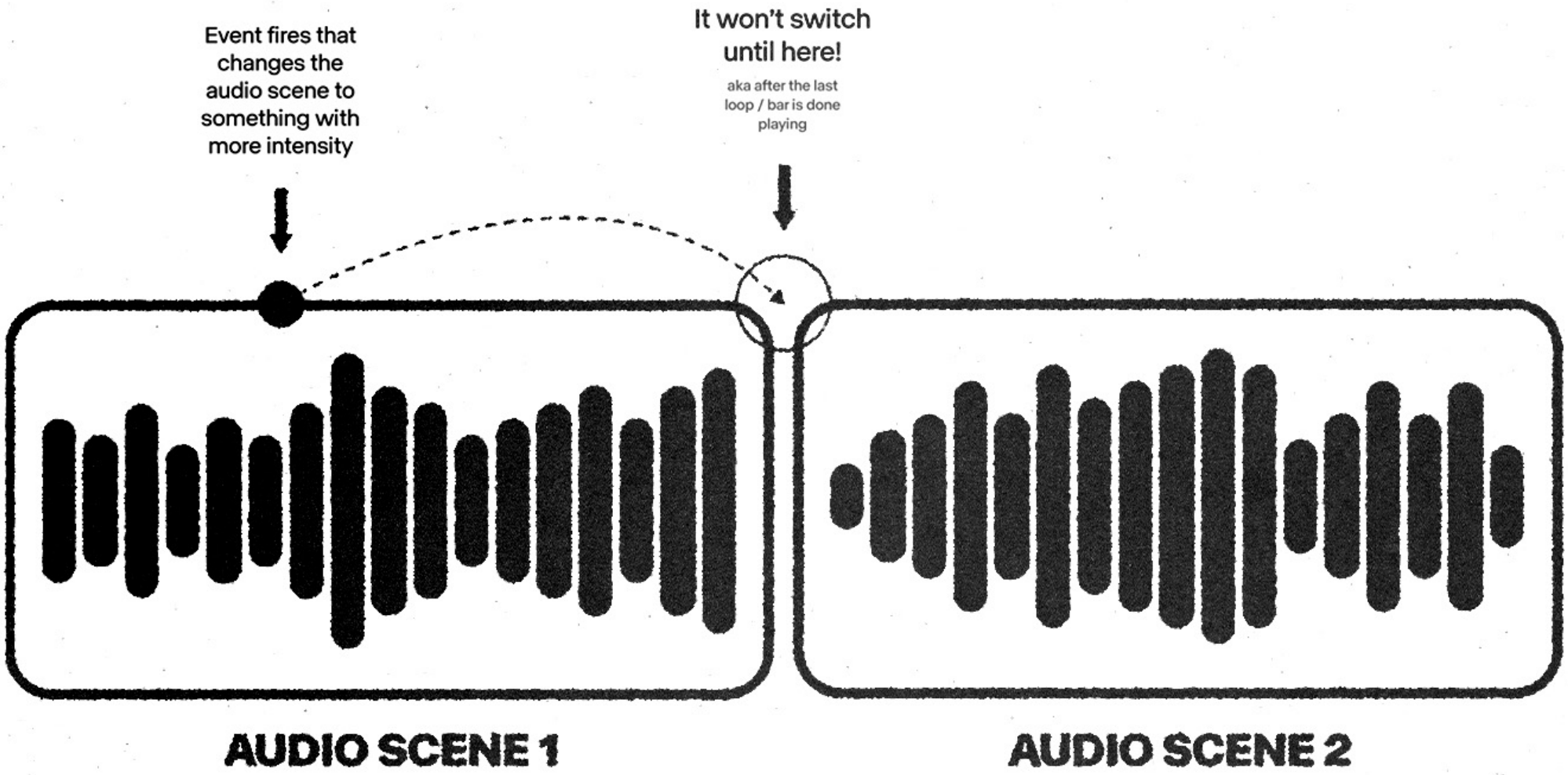
Even if you don't use much of the system here,
this should be an okay jump-off point for playing with `dspTime` in Udon Graphs!

If you do plan to use these things -
it's important that you understand the whole "audio scenes" thing and the benefit of scheduling
because I'm going to keep referring to it... so peep the diagram on the next page

Cool!
Now What The Fuck Does It Do?

Here's a basic example to sum it up:

Let's say you wanted to change the intensity of your score naturally *without* a crossfade or cut...



Transition Cues

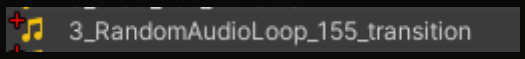
There's often two cases for transitions.

You might want a musical transition where you have an intro or a swell that segues into another piece that you can loop — in our case, we're gonna call these **Quantized Transitions.**

Quantized Transitions

Okay, so you're on *Audio Scene 3* + at the end of the current loop; you want to play *Audio Scene 4* **once** and then instantly move into *Audio Scene 5* + have that continue to loop until you (or the player) switches it again.

Super easy. The only thing you have to do is rename + write **"_transition"** at the end of the audio clip you plan to use for the transition (in this example - for Audio Scene 4)



The name should end like this!

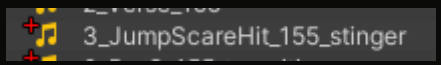
On the other end, you might want a transition that cuts through what is happening right now and surprises the player. We're going to call these **Instant Transitions**

Instant Transitions

Alright - back again. You're on *Audio Scene 3* but you want to ***SLAM*** into *Audio Scene 6* because...someone slammed through a door with A KNIFE in your game and you need some shrieking music to punch in (y'know jump scare shit!)... then you want to move into a high intensity loop that happens in *Audio Scene 7* for some big chase.

That's easy too. Just rename + write **"_stinger"** at the end of audio clips you plan to use for these big sting moments.

Then when those big event happens in game, you update the `audioScene` variable to 7. It'll jump right into it to scare the shit out of the player [and then move onto the looping audio scene right after it]



The name should end like this!

Firing Custom (Game) Events Off Audio Scene Changes

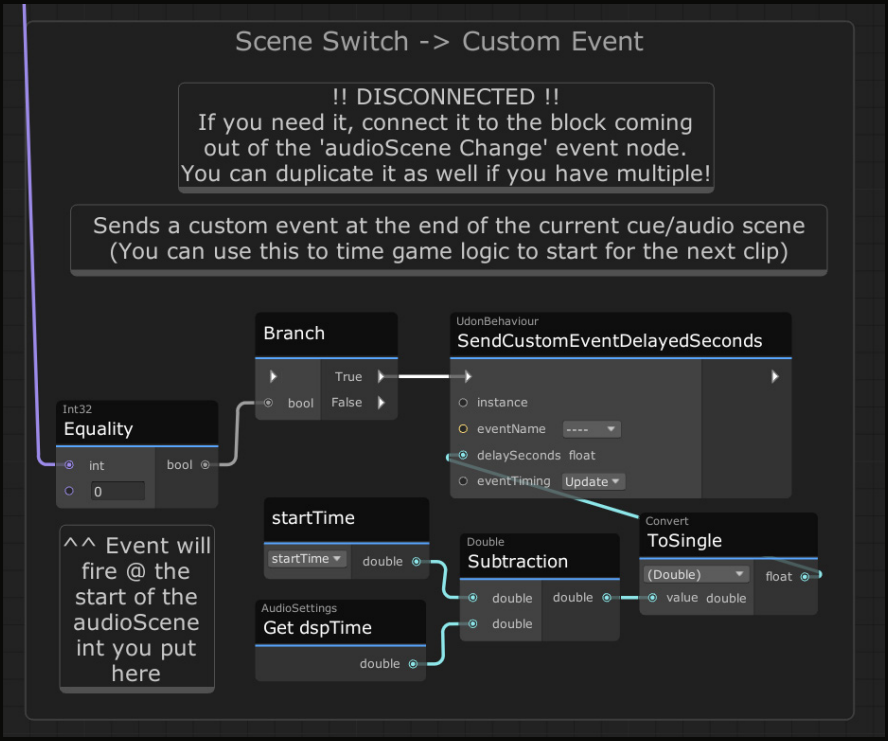
This might be more rare as this definitely wasn't made to be a system for handling something like a rhythm game, but there are definitely general cases where you want events to happen at the end/start of an audio scene, loop, bar, etc.

An example might be making sure an animation doesn't start until the next music cue starts... or maybe you want to teleport the player, but not until the next queued Audio Scene starts playing so the music matches the new environment.

Setup

For this one, you'll need to open up the graph but it's still simple to setup.

Look for the group labeled **'Scene Switch -> Custom Event'.** It's decently commented to tell you what to do.



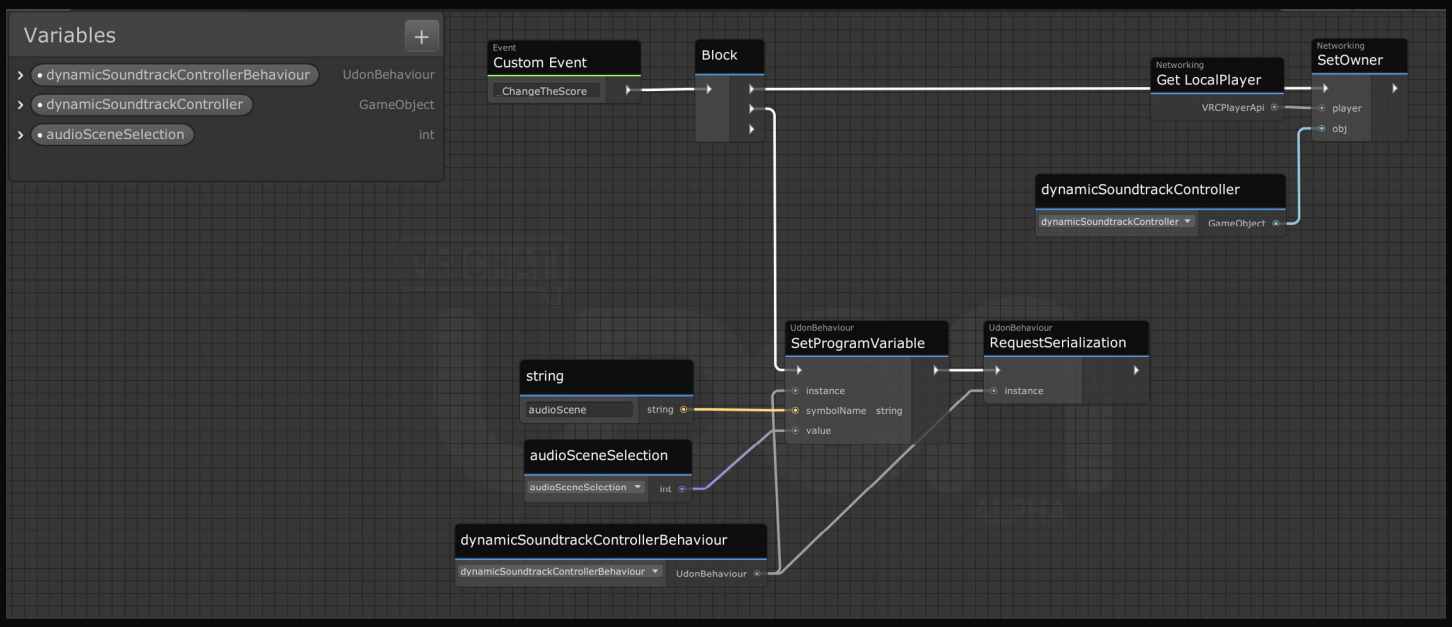
You just need to connect the output flow of the **'audioScene Change' Block** into the **Branch** node in that group, choose/set the `audioScene` number that will fire the event, and then set your custom event's name and target Udon Behaviour!

Networking / Syncing

If you plan to sync the state of the music/score for all players, it's as simple as syncing the `audioScene` variable! This is definitely a place where you'd want to use manual sync (which the prefab is set to already)

VRChat already has some great docs + videos on how to handle networking. The basic gist is that you want to set an owner of the Dynamic Soundtrack Controller game object, have them make the changes to the `audioScene` variable, and then request serialization so everyone else gets that updated variable!

Here's a simple example graph:



Although, you should consider the fact that you might not need to sync this system if you're syncing some higher-level core game mechanic or event mechanic that could easily propagate the 'audio scene' changes for each player "locally".

Changing audioCues array at runtime

This is more so for the advanced nerd shit...but if you plan to change (add or remove) audio clips from the `audioCues` array at runtime; you'll need to call the `_RebuildArrayLength` event on the Dynamic Soundtrack Controller behaviour immediately after doing it.

This is because we store the length of the `audioCues` array in a variable at `Start()`

Authoring Assets

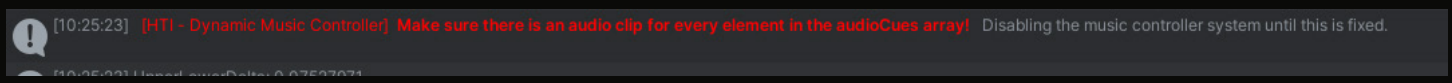
There's no real rules here - but a good starting point is to just export a loop file per bar. It'll depend on your use case so I suggest using markers or locators in your DAW to mark your loop start + end points for export!

Here's an example from Ableton... but I know there's equivalents in Reaper, FL Studio, Audition, Logic, etc.

Common Errors / Troubleshooting

Generally, this system will break if you have a row in the audioCue array that isn't filled / is left blank.

We have a check that will print this into your console + disable the system until it's fixed:



If you find any other strange bugs - make an issue on the repo or reach out to me on Discord!
KidKwazine#8416

Notes

Feel free to extrapolate this / make it better / remake it in U# or whatever so the inspector can be pretty.

I wouldn't call myself a programmer, but my attempts at bullying some programmer nerd friends into making this failed so I had no choice. It's probably shit!

It worked well for me, though.

have fun, love u
xoxo rob (aka kid kwazine)