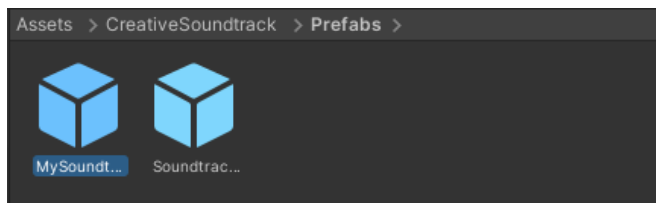


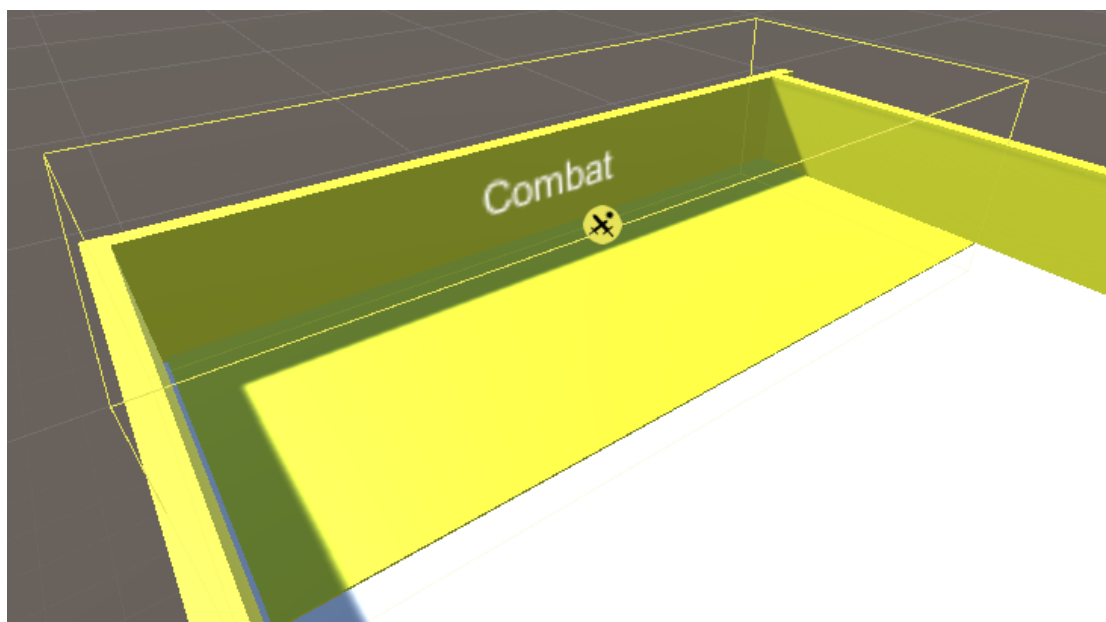
1. Setting up Scene

First of all, you want to add the prefab MySoundtrackManager to scenes you want to use our tool.



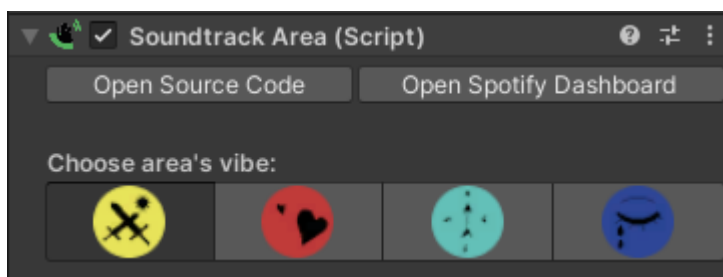
2. Inserting SountrackAreas to the Level

Just drag the prefabs into the scene, and a new object will be created, looking more or less like this:



That means that when the player enters this area, the player's songs that match the defined vibe will start to play.

In the inspector, you can change the vibe associated with the area:

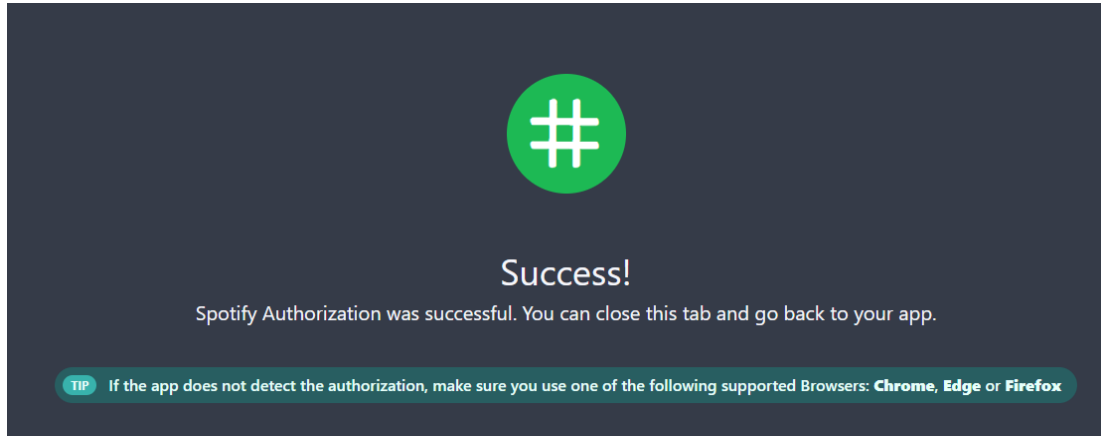


3. Playing and Logging In

Whether it is you testing or the player playing your game, when `MySoundtrack.Connect()` is called for the first time, a browser page will open asking you to log in into your spotify account. After the login, the SpotifyAPI

will ask for some permissions, in order to visualize songs and control playback states.

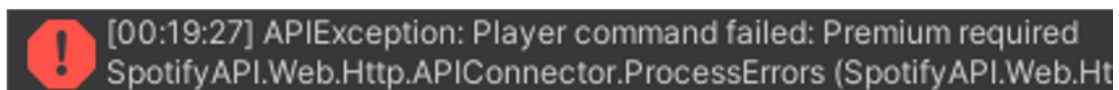
After you log in and accept the permissions request, you will be redirected to a success page:



Now you can close the browser page and enjoy MySoundtrack with your game!

Important Notes

- Connect() is called on Start by default, but you may want to change it.
- Depending on your scene, and how many SoundtrackAreas you put in, it may take a few seconds to properly initialize all the areas. Performance improvements are currently under progress :)
- **Player must be a Premium user of Spotify**, or else, it will not be possible to play songs on the account via the API. If while testing or playing, you log in into a free Spotify account, you might get the following exception when trying to play player's songs:



- If when importing the package, some compilations errors appear, talking about Newtonsoft package version, you may ignore it and clear the console. This error happens because of a conflict between the versions of Unity's Newtonsoft.Json package and SpotifyWebAPINET's one. **This does not stop the package from working properly.**



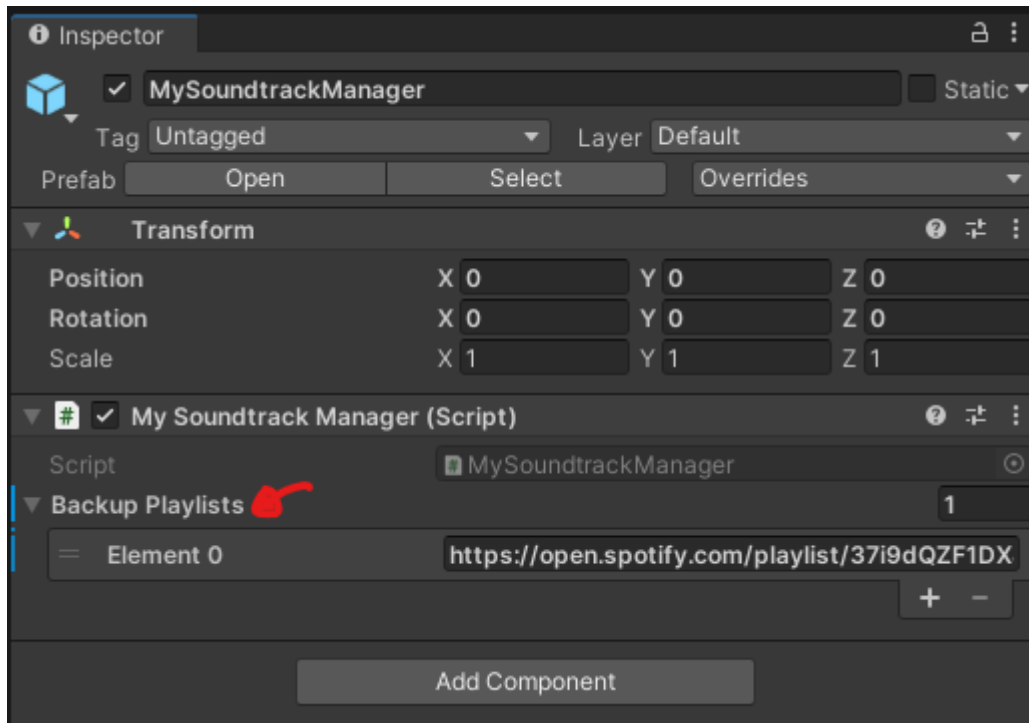
And that's all you have to do to use MySountrack and start using the player's songs together with the game's experience!

Thank you all for downloading. This package is in its initial stages, more updates will be coming soon :)

Optional Features

1. Backup Playlists and Features Playlists

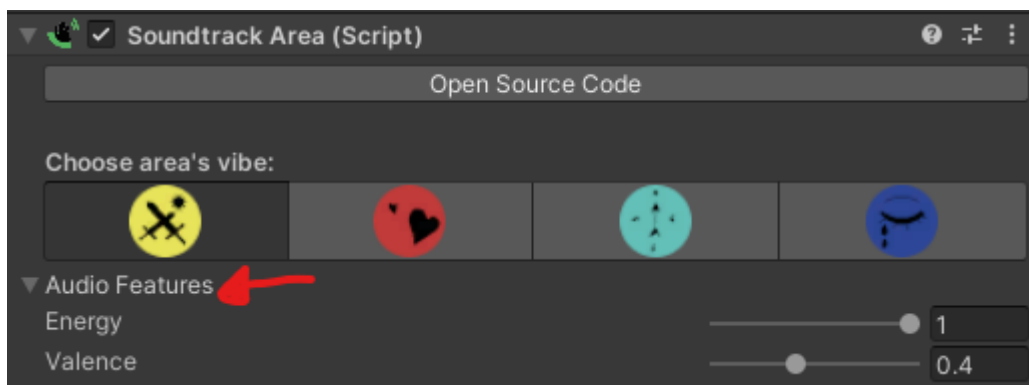
You can choose any number of playlists to work as a backup if the player does not have enough liked songs. You can do that in the MySoundtrackManager, for example:



If the player doesn't have enough liked songs **and** no backup playlist was defined, MySoundtrack will choose among player's Features Playlists. These playlists are typically those that appear on the "Browse" menu on Spotify.

2. Customize SoundtrackArea's vibe

If the four pre-defined vibe presets don't please you, you may customize the vibe yourself, by accessing the "Audio Features" menu on the SoundtrackArea component:



3. Playing songs programatically, at any moment

If you, instead of using SoundtrackAreas, want to call a certain vibe through code, you may use MySountrackManager's PlaySongsOfVibe() method, like so:

```
MySoundtrackManager.Instance.PlaySongsOfVibe(energy: 0.5f, valence: 0.5f);
```