VO Manager Lite

Documentation

# Table of Contents

Setup

In order to use our VO Manager, we must first make sure that our VO Bank is present in the scene at all time. To set up a new VO Bank you can either:

1. Create a new empty GameObject in the Hierarchy and drag and drop the VO Bank or VO Manager script onto the newly made gameobject or;
2. You can go into the prefabs folder in the project panel and simply drag and drop the VO Bank prefab into the Hierarchy.
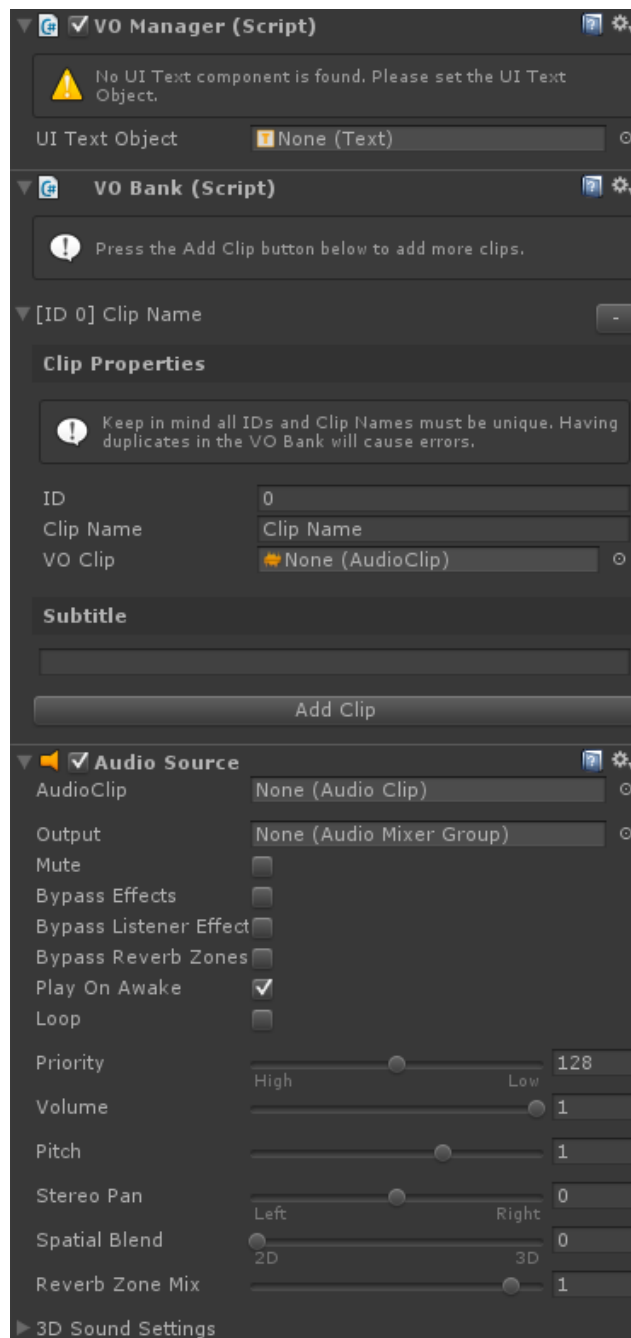
Before we can begin using the VO Manager we must first start by adding VO (voice-over) clips into our VO Bank and any subtitles that we may want to go with it. To do so, click on VO Manager in the Hierarchy and press the "Add Clip" button under VO Bank. You should see a similar interface below once you click on the add clip button.
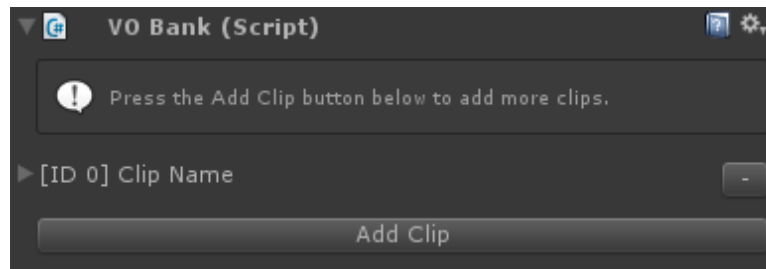
Go ahead and click on the expand icon by "[ID 0] Clip Name" to expand the panel.
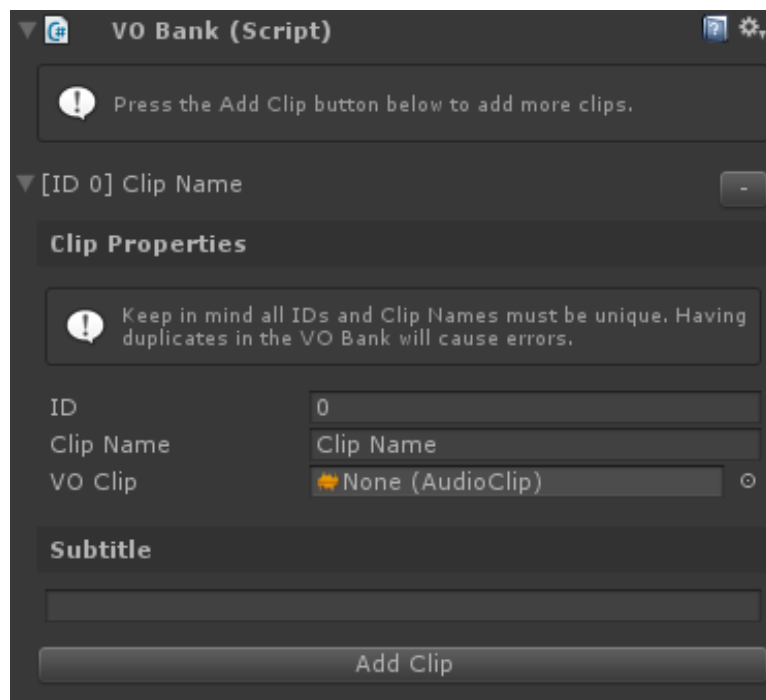
Once expanded you should see a few more options. Here is a list of what they are and do:

- **ID** - An integer value that can be manually assigned. This is used as a reference to call up your clip's properties.
- **Clip Name** - A string that can be manually assigned. Like the ID, this is used as a reference to call up your clip's properties.
- **VO Clip** - The audio clip.
- **Subtitle** - The text that appears on screen when you play the audio clip.

Once the VO Bank is set up there is one last step we need to perform before we can use our VO Manager. In order to use our subtitle system, we must first go to the Hierarchy and create a new Text object. To do this go to Create -> UI -> Text. By default (if you have no Canvas elements in the scene) it will automatically create a Canvas and EventSystem object for you like so:
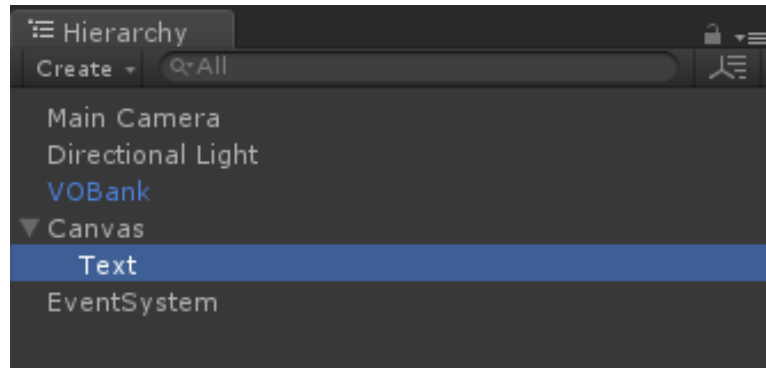
Setup the Text component properties and values the way you'd want it to appear (by default you can keep the text field of the text component empty). Once you are done click back on the VO Bank object and drag and drop the Text component we just created (highlighted in the hierarchy above) into the UI Text Object field in our VO Manager. Once you have at least one sound clip added into your VO Bank it should look like the following:
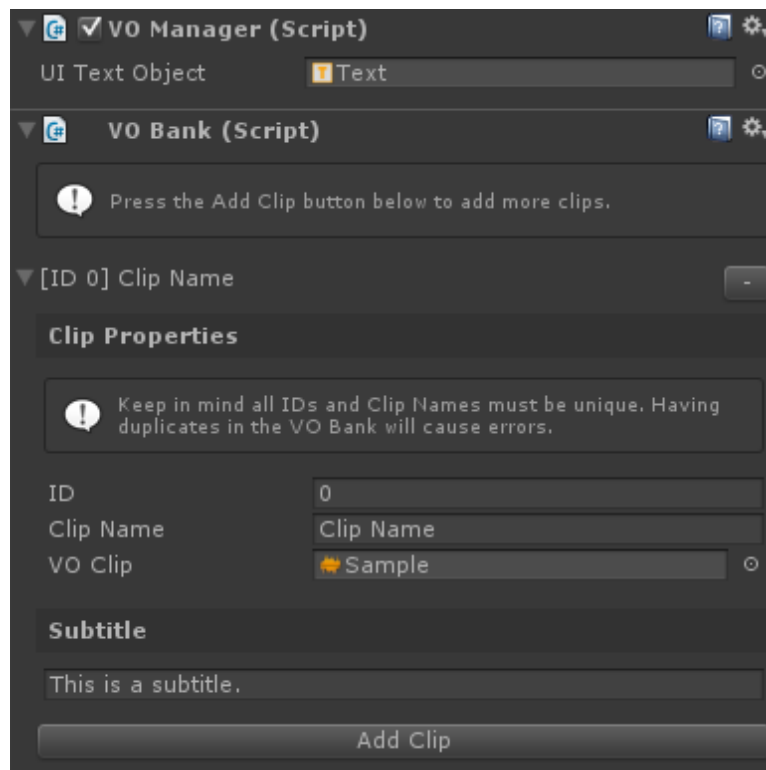
Once the set up for the manager has been completed, save your scene and do one of the following:

1. If you used the VO Bank prefab you can either Apply the current prefab or;
2. Create a new prefab by dragging your VO Bank object into the Project panel (this should be done if you've made your VO Bank by creating a new GameObject) to save a backup prefab.

You can also remove audio clips by clicking on the "-" button beside each audio clip name.

Usage

There are 4 main functions you can call up through the VO Manager.

1. Play Mode
2. Force Play Mode
3. Checks
4. Stop

Play Mode

Play mode can be called in 4 different ways and triggers audio calls normally. This means that once you make a call to the play function, you cannot make another one until the function is done playing.

```
VOManager.Instance.Play(int _id);
VOManager.Instance.Play(string _name);
```

The example above both plays audio the same way except they each take in a different type of parameter. One takes in an integer while the other takes in a string. You can call an audio to play by passing through its ID or its assign Name.

```
VOManager.Instance.Play(AudioSource _audSrc, int _id);
VOManager.Instance.Play(AudioSource _audSrc, string _name);
```

The second set is very identical to the ones above except it takes in a second parameter which is an audio source. By passing through an audio source you can play an audio from an external source instead of the one attached to the VO Bank. This allows for more control over 3D sound.

Force Play Mode

Like our regular play mode, force play mode takes in the same type of parameters. The parameters passed through can either be an integer or a string. The distinction between Play and Force Play mode is that calling the force play function immediately cuts off the current audio that is playing and plays the new on that is called.

```
VOManager.Instance.ForcePlay(int _id);
VOManager.Instance.ForcePlay(string _name);
```

Force play mode also takes in an external audio source.

```
VOManager.Instance.ForcePlay(AudioSource _audSrc, int _id);
VOManager.Instance.ForcePlay(AudioSource _audSrc, string _name);
```

Checks

If at any given time you need to check if an audio is playing, you can call the is playing function which return a boolean value of true or false.

```
VOManager.Instance.IsPlaying();
```

Stop

Calling the stop functionality simply stops all current audio sources that are playing and subtitles drawn on-screen.
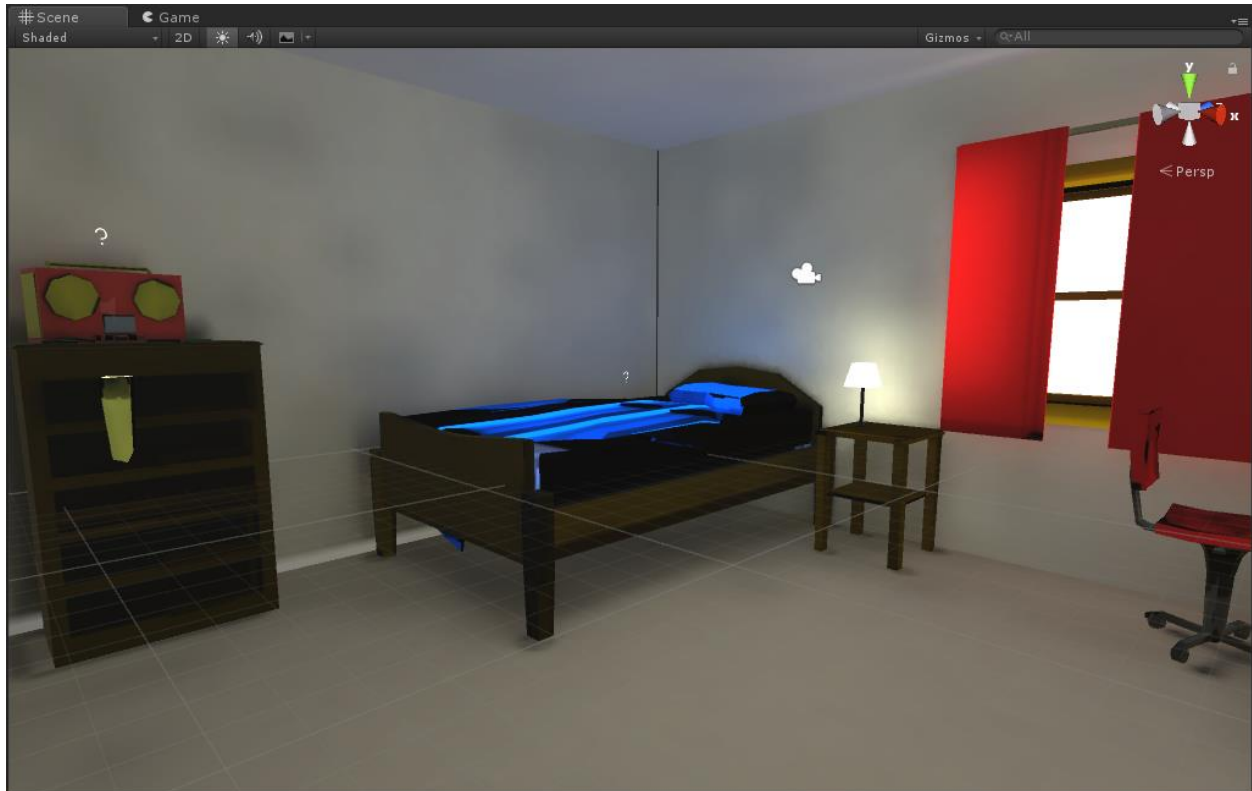
```
VOManager.Instance.Stop();
```

Demo

Fixing the Scene Lighting

VO Manager comes with a small demoscene. Upon your first time opening the demo scene you may see something similar to below:

Fear not! This is quite normal. To fix this simply open the Lighting window by going to Window -> Lighting. Once the window is open if the **Auto** box is not checked, press **Build** or **Generate Lighting** (in later versions of Unity) and let it process the scene.
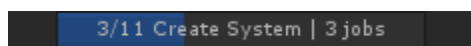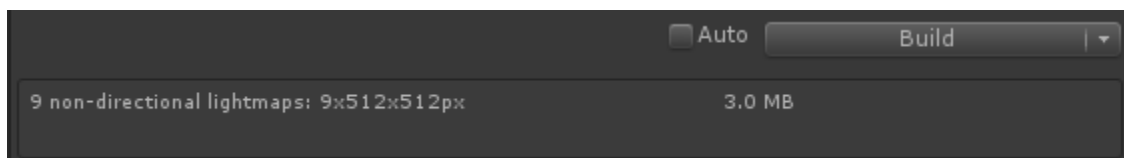
Once the lighting is baked you should now see your scene all fixed up like so:
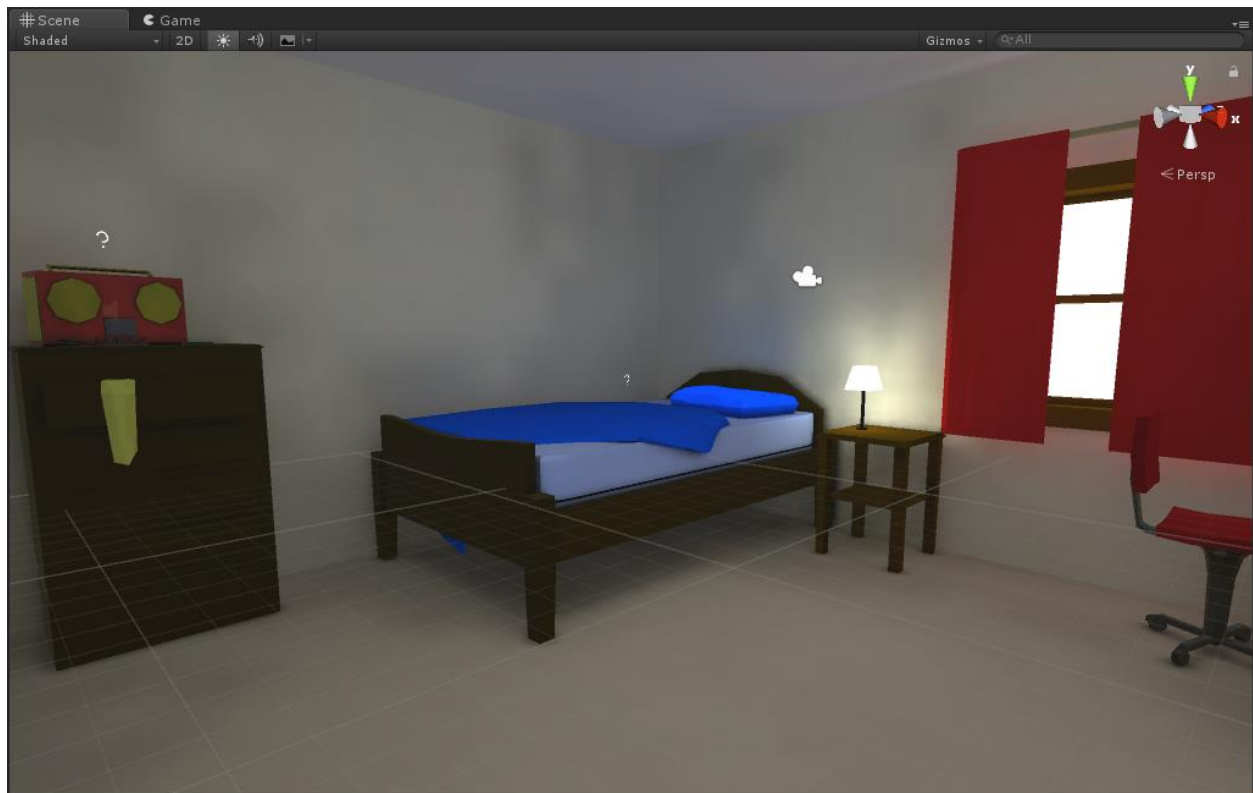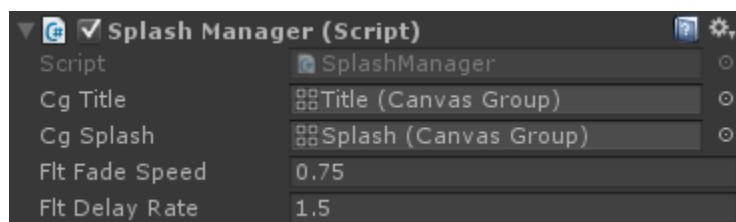

Scene Setup

Splash Manager

The Splash Manager can be found under the GameManager object in the Hierarchy.

The Splash Manager controls the splash screen that we see fade into the demo scene upon play.

If you look at the Inspector with the GameManager object highlighted you'll see that our splash manager contains four parameters.

- **Cg Title** – The title screen canvas that contains the logo and text.
- **Cg Splash** – The splash screen canvas that contains the title canvas and the background.
- **Flt Fade Speed** – The speed of which the title fades in and out.
- **Flt Delay Rate** – The delay rate before the title screen begins fading out after it fades in.

Back in the Hierarchy, if you expand the default Canvas, you'll see a Splash object and within that a Title object. Expand these objects and make the necessary changes to take full advantage of the splash screen feature.

Game Manager

On the same GameManager object in the Hierarchy, you'll notice we also have a Game Manager script attached to it.

This script controls the entire game scene and can be broken down into smaller working pieces.
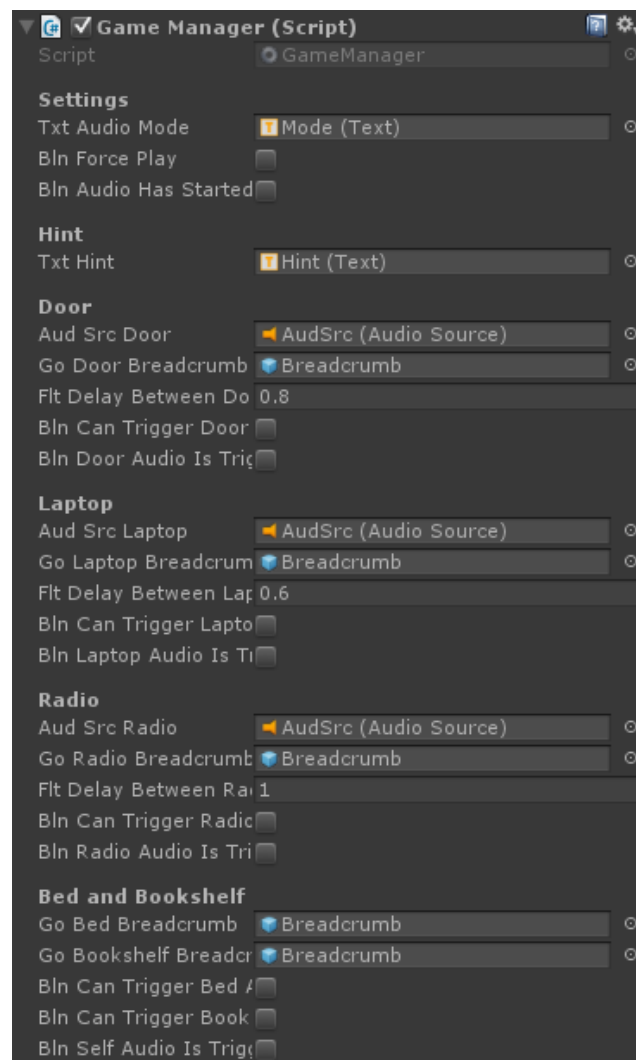
- **Settings** - Check which audio mode is currently being used and displays that information onto the UI. It also checks when an audio clip has started playing.
- **Hint** - Handles displaying useful information on the screen to inform the user what to do. This parameter contains one of our UI text object.
- **Door, Laptop, Radio, and Bed and Bookshelf** - These sections handle their own audio calls and information. Each section checks when their audio source is triggerable and if it is triggered or not. Some also have their own delay rate to determine when the next audio clip in the loop is played as well as if it is being played from another audio source. All sections also have breadcrumb to indicate in the game environment what objects are interactable.

Audio Sources

There are 4 audio sources being used throughout the demo scene. Each source comes from the:

- Door
- Laptop
- Radio
- VOBank

These audio sources can be found under each of their respective object in the Hierarchy.
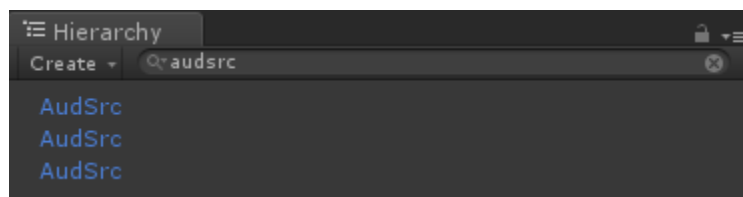
To quickly find the audio sources you can also type in "audsrc" into the search bar in the Hierarchy just like the screenshot above. These audio sources also make use of 3D sound settings and audio mixers.

Play Controls

Our scene also contains a very basic character controller. If we click on the Player object in the Hierarchy you'll have a couple settings you can adjust in the Inspector.

- **Flt Speed** - The speed of which the player is moving.
- **Flt Gravity** - The gravity applied onto the player object to keep it grounded.
- **Flt Mouse Sensitivity** - The mouse sensitivity and speed of which the camera turns.

Script Calls

This section will only give you a brief overview of how each audio clip is called to play for each respective object. To get a better understanding of how everything works please take the time to read through the scripts which are thoroughly commented.

Door

The Door audio clips are triggered on Line 194 and 215 in the GameManager.cs

Calling the normal play function triggers line 215 which loops through an array of IDs calling one play function after the next as they are completed. This is how a conversation is played using the audio clips. You can find the array of IDs on line 32. Each ID in this array is related to an ID in the VO Bank.

Line 194 only gets executed when Force Play Mode is on. This instantly plays the first ID in the array and follows through to line 215 and triggers the audio events afterward normally in the appropriate order.

Laptop

The Laptop audio clips are triggered on Line 281 and 302 in the GameManager.cs

Calling the normal play function triggers line 302 which loops through an array of IDs calling one play function after the next as they are completed. This is how a conversation is played using the audio clips. You can find the array of IDs on line 42. Each ID in this array is related to an ID in the VO Bank.

Line 281 only gets executed when Force Play Mode is on. This instantly plays the first ID in the array and follows through to line 302 and triggers the audio events afterward normally in the appropriate order.

Radio

The Radio audio clips are triggered on Line 368 and 389 in the GameManager.cs

Calling the normal play function triggers line 389 which loops through an array of IDs calling one play function after the next as they are completed. This is how a conversation is played using the audio clips. You can find the array of IDs on line 52. Each ID in this array is related to an ID in the VO Bank.

Line 368 only gets executed when Force Play Mode is on. This instantly plays the first ID in the array and follows through to line 389 and triggers the audio events afterward normally in the appropriate order.

Bed and Bookshelf

Unlike the other three actionable calls, the Bed and Bookshelf audio clips are called through an array of strings instead of IDs. This is just a demonstration of another way to call audio clips.

The Bed array can be found on line 62 and the Bookshelf on line 63.

Lines 438 and 456 gets triggered when a normal play mode is being used while line 485 and 503 gets triggered when force play mode is being used.

Each of these lines increments in order of their arrays. So each audio clip gets the chance to be played at least once and starts over once it is done each time it is triggered.