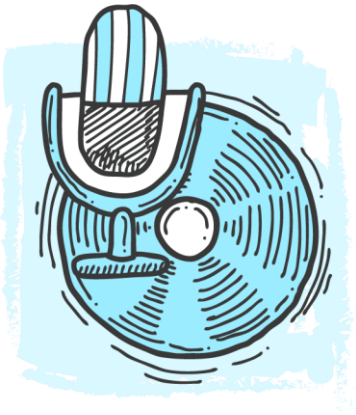


VO Manager Lite

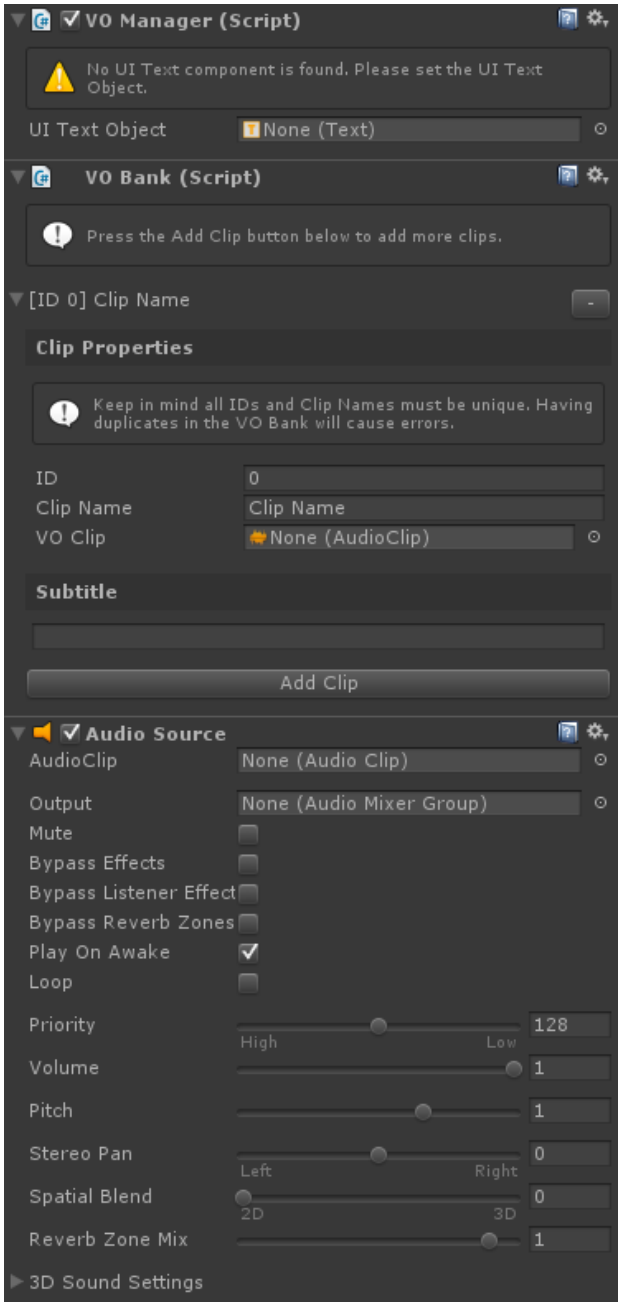
Setup and Usage Guide

Table of Contents

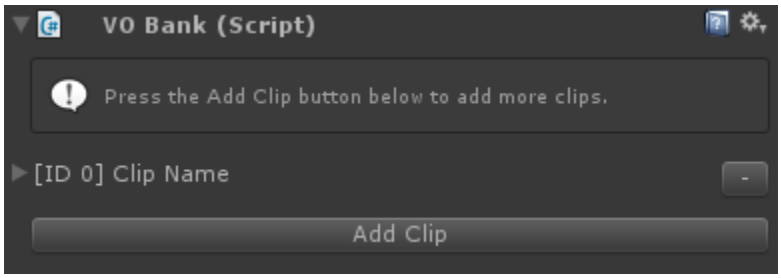
Setup

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

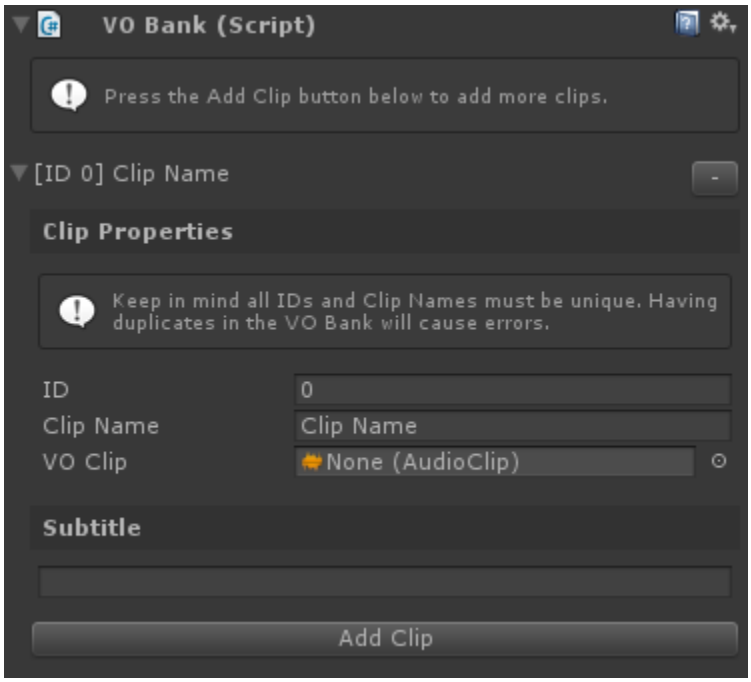
1. Create a new empty GameObject in the Hierarchy and drag and drop the VO Bank or VO Manager script onto the newly made gameobject or;
2. You can go into the prefabs folder in the project panel and simply drag and drop the VO Manager prefab into the Hierarchy.



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.

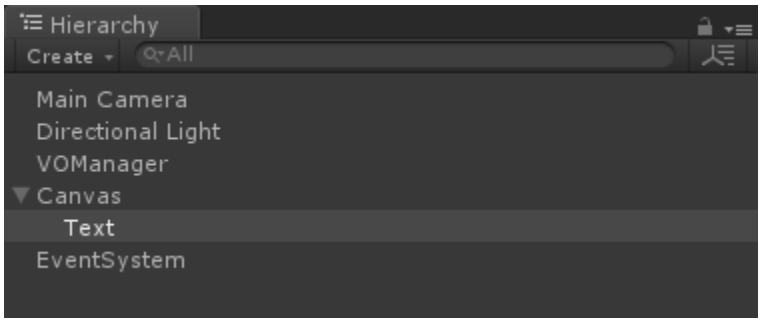


Once expanded you should see a few more options. Here is a list of what they are and do:

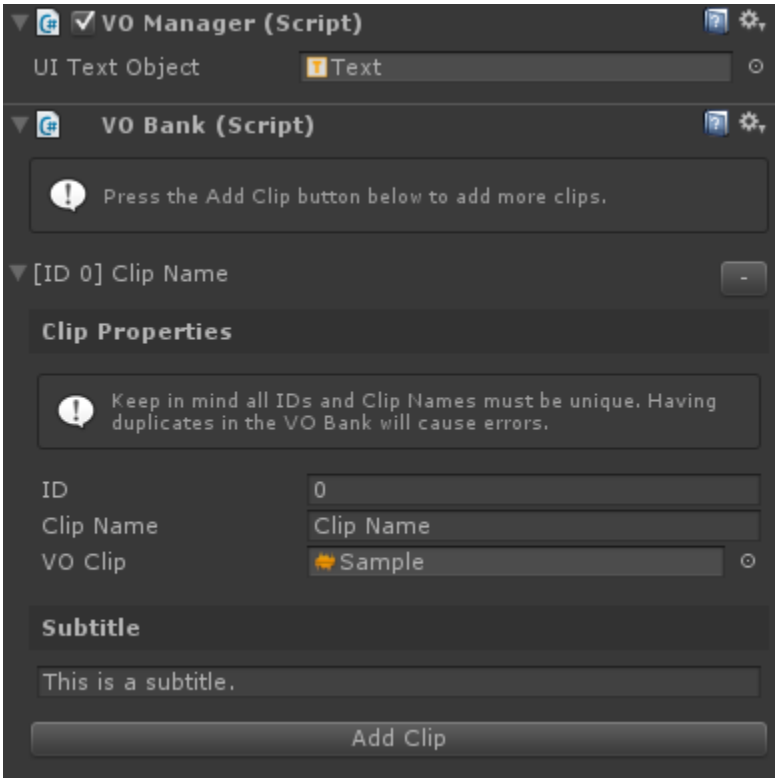
- **ID** - An integer value that can be manually assigned. This is used as a reference to call up your clip’s properties.
- **Clip Name** - A string that can be manually assigned. Like the ID, this is used as a reference to call up your clip's properties.
- **VO Clip** - The audio clip.
- **Subtitle** - 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 have to 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 Manager 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 Manager prefab you can either Apply the current prefab or;
2. Create a new prefab by dragging your VO Manager object into the Project panel (this should be done if you made your VO Manager 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 currently two ways to trigger an audio through the VO Manager. Those are:

1. Play Mode
2. Force Play Mode

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 can not make another one until the function is done playing.

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

The example above is a set of the same except the major difference is that they both 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 are able to play an audio from an external source instead of the one attached to the VO Manager. This allows for more control over the 3D sound.

Force Play Mode

Just like our regular play mode, force play mode takes in the same type of parameters to trigger an audio. This parameter takes in either an integer or a string. The biggest difference 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);
```

Much like the functions above, 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 just stops all current audio sources that are playing and subtitles drawn on-screen.

```
VOManager.Instance.Stop();
```

Useful Links

If you have any questions, feedback, or issues, please feel free to contact me via:

- Email at tvledesign@gmail.com or;
- Go to the GitHub Repo at https://github.com/tvledesign/vo-manager-lite

Additionally to learn more or find tutorials you can go to:

- Website at https://www.tvledesign.com
- Blog at https://www.tonyvle.com
- YouTube Channel at https://www.youtube.com/channel/UCeto2S7J0vwAeNAdonRH80w

Credits

Special thanks to these wonderful people who were on my team at the Global Game Jam 2016 which inspired me to create the VO Manager:

- Cameron Cintron
- Jessica Borlovan
- Joe Song
- Neal Shaw

Last but not least these wonderful organizations that made this idea possible:

- Global Game Jam
- Petal et Al