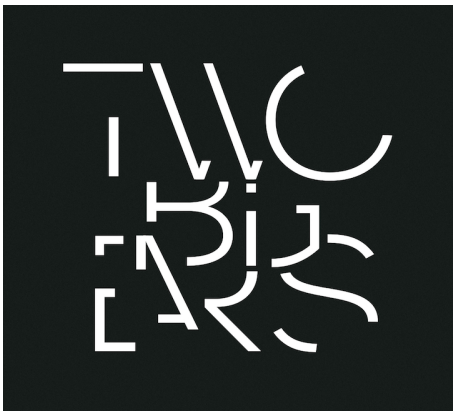
3Dception v0.3.0b

Two Big Ears



Getting Started in Unity 3D

TWO BIG EARS

3Dception – Getting Started in Unity3D

9/9 Brighton Street
Edinburgh, EH1 1HD
support@twobigears.com
twitter.com/twobig_ears

Introduction

Thank you for downloading 3Dception for Unity! Feel free to get in touch with us — support@twobigears.com — if you have ideas, bugs to report or need help.

Versions

3Dception is currently in beta. During this introductory period all desktop platforms (OSX, Windows and Linux) are bundled together. Depending on the product you downloaded, you might have some limitations:

- Non-Commercial: For non-commercial use only, with a limitation of 5 active binaural sources.
- Starter: For commercial use, with a limitation of 3 active binaural sources
- Basic: For commercial use, with a limitation of 5 active binaural sources
- Pro: For commercial use, with no limit on the number of binaural sources

License

Regardless of the product or version you downloaded, you should have been supplied with license keys in an email. Keep these safe as they are needed to activate 3Dception. Details on license activation are provided in the next page.

About Us

Two Big Ears is a Scottish company that designs audio tools for immersive and interactive applications. Stay updated at www.twobigears.com.

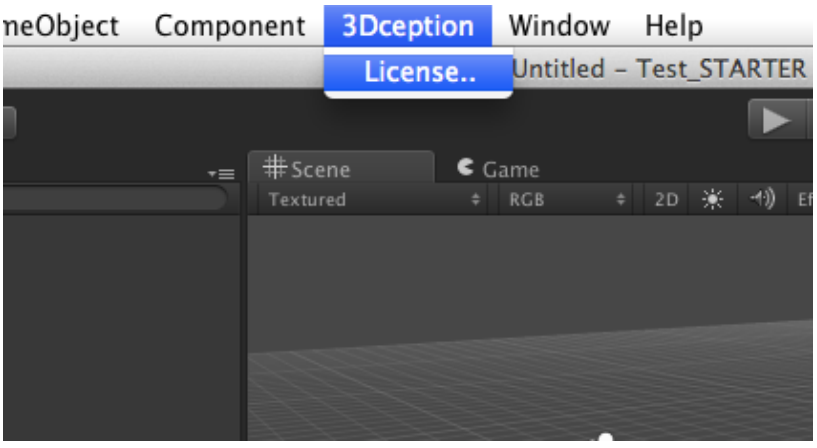
Updating

If you have installed a previous version of 3Dception in a project, follow these steps to update:

- Open your Unity project
- Go to the 'Assets' menu and choose Import Package > Custom Package
- Choose the new 3Dception Unity Package
- Click on import
- 3Dception will be updated to the new version

Activating Your License

- Import and setup the 3Dception Unity Package in your project (read the 'Using 3Dception' section later in this document for help)
- Click on '3Dception' in the menu bar. You might need to click elsewhere in the menu bar for '3Dception' to appear
- Click on 'License...'



- Enter your license key in the license pop-up window and click on authorise. **You must be connected to the Internet for this first verification.** Once authorised, you won't need to authenticate your computer again unless you deauthorise your license or receive a new one

Structure

3Dception consists of three primary components

- **Global Listener**

  A single instance must to be added to the main camera or the camera that controls the point of view of the player. If you are working with stereo visuals, you can place the GlobalListener on either of the two cameras. Make sure you have only ONE version active in a scene.

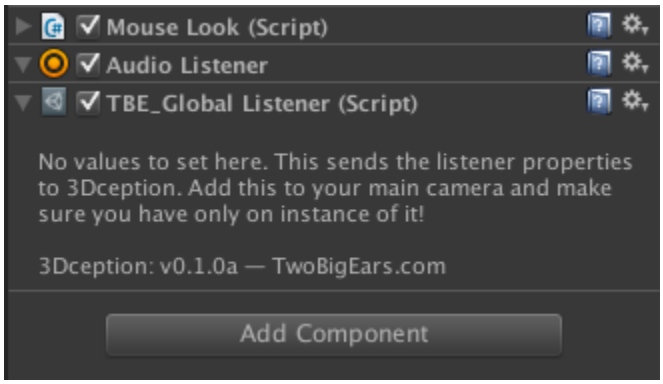
- **Source**

  The binaural audio source extends the functionality of Unity's AudioSource with binaural spatialisation options.

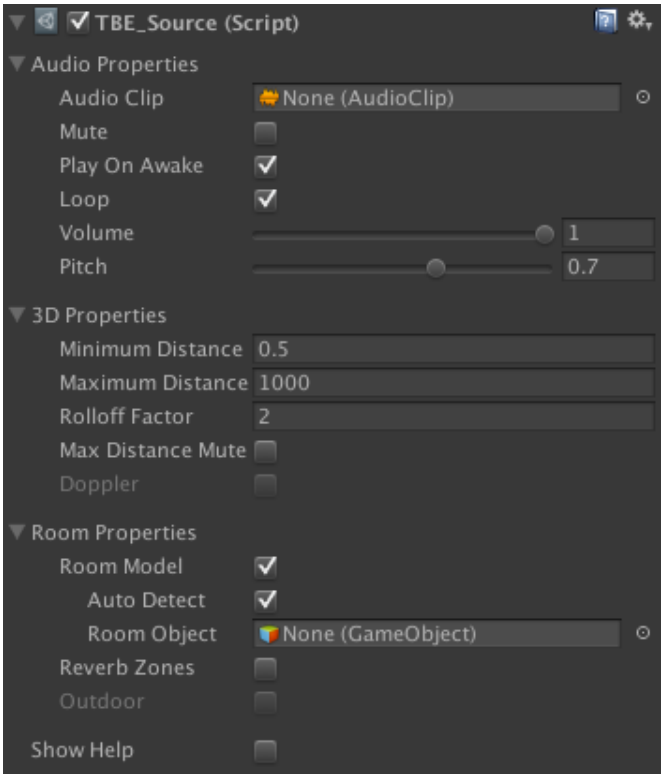
- **Room**

  A Unity prefab to setup room-modeling parameters

Global Listener The Global Listener has no properties to set. If you are using a first person controller, make sure the Global Listener is added to the camera and not the parent object in the hierarchy.



Source The Source must be placed on any game object that needs a binaural audio source. You will be familiar with its functionality if you have previously used the default AudioSource in Unity.



Audio Properties:

- Audio Clip: Specify the audio clip that needs to be played back
- Mute: Mute the source
- Play On Awake: Automatically play on game 'awake' or start.
- Volume: The volume of the source
- Pitch: The pitch of the source

3D Properties:

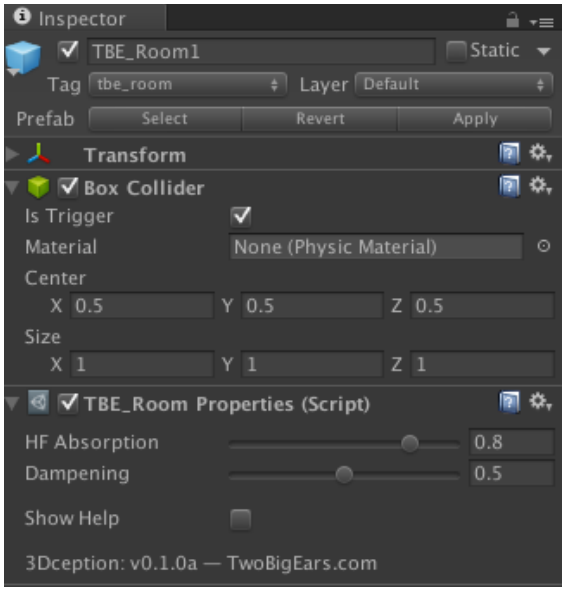
These properties determine how the amplitude of the sound changes depending on the position of the object on the game map. The further away a sound is from the listener, the lower its amplitude will be.

- Minimum Distance: The distance value after which the amplitude attenuation starts.
- Maximum Distance: The distance value after which the amplitude attenuation stops.
- Rolloff Factor: The distance attenuation curve. A value of 1 is a 6dB drop every time the distance is doubled (this is how most sounds work in the 'real' world). A value greater than 1 results in a steeper rolloff.
- Max Distance Mute: Mutes the audio source if the distance of object from the listener is greater than the maximum distance.

Room Properties:

- Room Model: Enable or disable room modeling
- Auto Detect: Automatically detect which room prefab the source is in
- Room Object: Manually specify the room prefab object
- Reverb Zones: Use Unity reverb zones

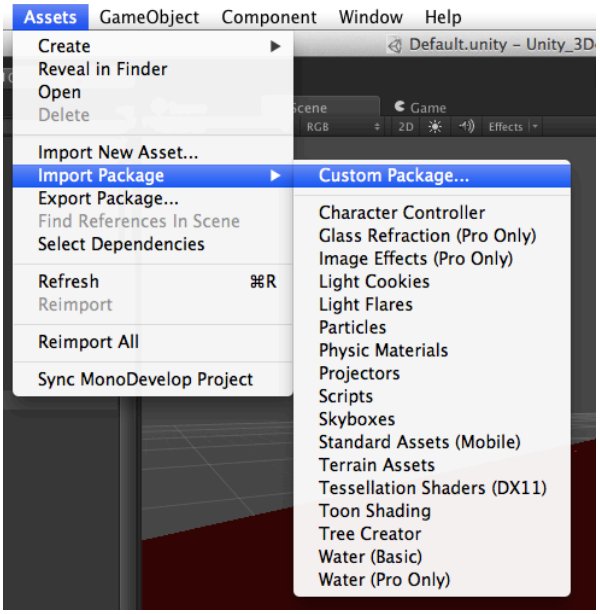
Room The room modeling properties are set using a prefab called 'TBE_Room', found in the 'TBE_3Dception' folder under 'Assets'. Instructions for setting this up are provided below.



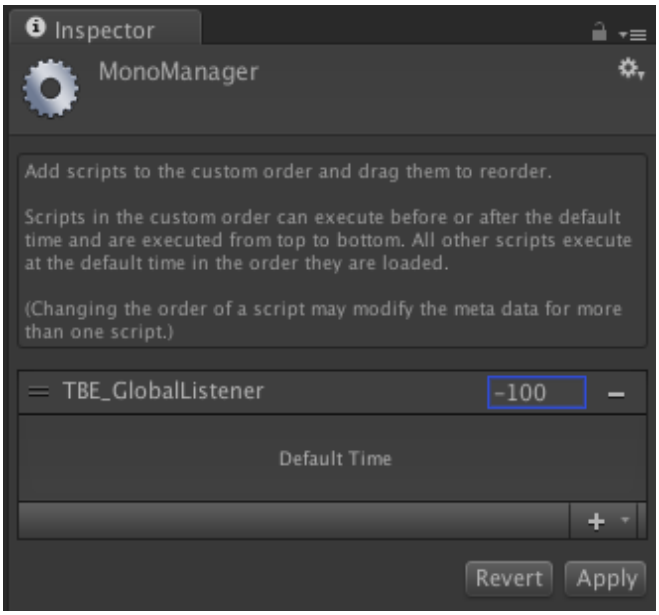
Using 3Dception

In just a few steps you can have 3Dception integrated into your Unity scene:

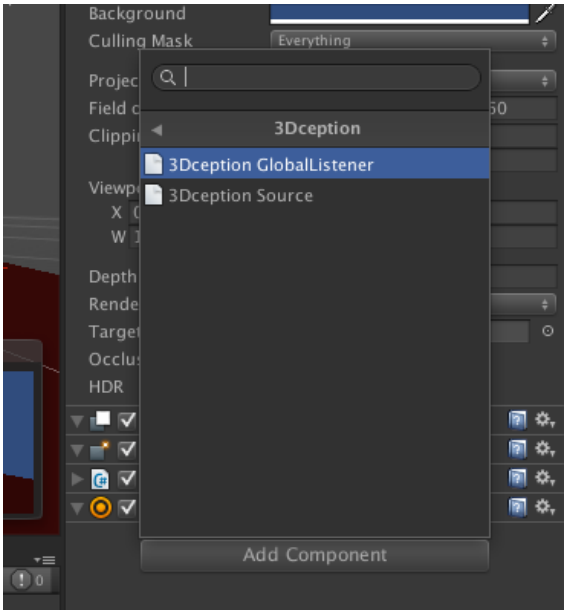
1. Open Unity and load your project and scene
2. Go to the 'Assets' menu and choose Import Package > Custom Package



3. Choose *3Dception_0.3.0b_system.unitypackage*, where 'system' would be OSX, Windows or Linux. For example, if you use OSX it would be named *3Dception_0.3.0b_OSX.unitypackage*.
4. Go to the 'Edit' menu and choose Project Settings > Script Execution order. In the Script Execution inspector, add 'TBE_GlobalListener' using the "+" on the bottom right and change the time value from 100 to -100. Click on apply.

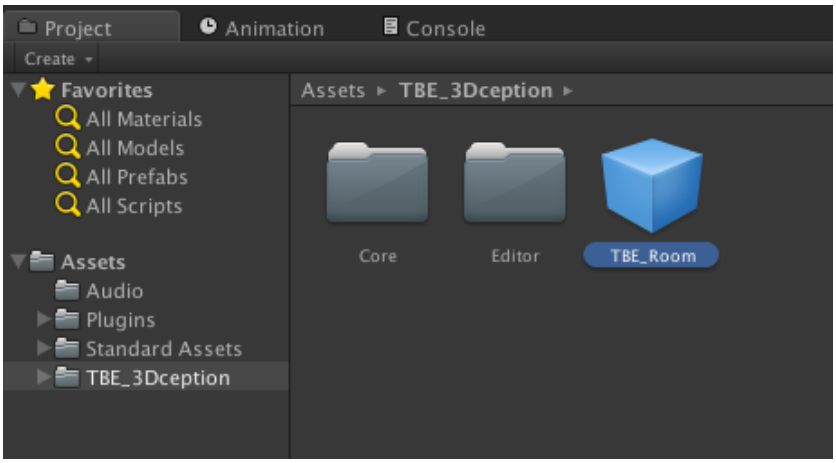


5. Select your main camera and click on the 'Add Component' button in the inspector. From the contextual menu, click on '3Dception' and choose '3Dception GlobalListener'

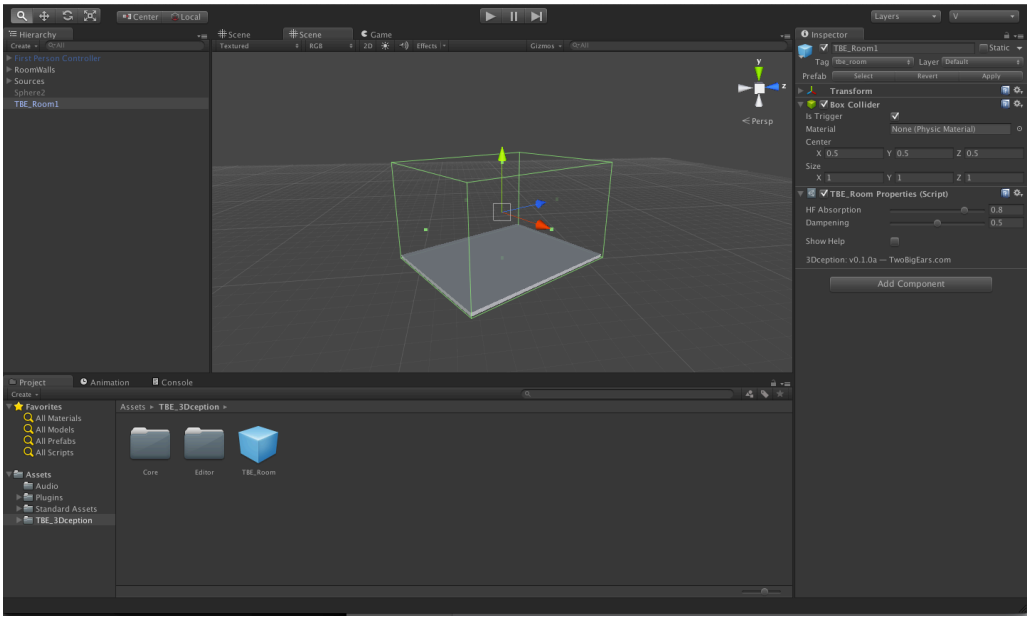


6. Select your game object and click on the 'Add Component' button in the inspector. From the contextual menu, click on '3Dception' and choose '3Dception Source'

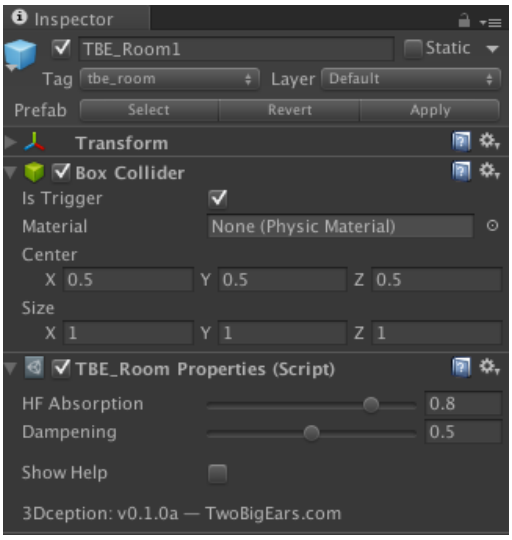
7. Under the 'Project' tab in Unity, navigate to Assets/TBE_3Dception. Drag the 'TBE_Room' prefab into your scene.



8. Scale this prefab around the room within your scene. If you have multiple rooms, add multiple instances of the prefab around each of the rooms. These prefab rooms do not get rendered, but act as detectors in 3Dception to setup the room modeling properties. Think of it as acoustic paint on the walls in the room. If you have a room that isn't a cube, you can still use TBE_Room as an approximation of its dimensions.



9. Clicking on the TBE_Room prefab will give you access to the room reflection properties



10. **Done!** If you have a Unity Reverb Zone setup in your scene, you can use it with 3Dception for enhanced spatialisation. Just enable 'Reverb Zones' on the 3Dception Source properties.

Room Modeling

The room modeling system in 3Dception works by creating the reflections that are important for spatialisation. This means that it could be used with any other reverberation system without any problems. It is advisable to use it in conjunction with another reverb system for best results. Adding a small predelay to the reverb system (30-60ms) can greatly enhance the binaural spatialisation.

Performance

3Dception is optimised for maximum results with as little CPU usage as possible. In most cases our measurements have shown the algorithms take up a lesser percentage of CPU compared to just playing back an OGGVORBIS audio file in Unity!

Unfortunately, the audio system in Unity 4 is greatly limited in flexibility and you will see overheads that have nothing to do with 3Dception but to do with Unity moving audio data around. This is currently being investigated.

3Dception Unity API

You can control or change any of the properties on a 3Dception binaural source using scripts. The methods are similar to Unity's *Audio Component*.

GET / SET methods:

```
GetComponent<TBE_3DCore.TBE_Source>().erRoomObject
GetComponent<TBE_3DCore.TBE_Source>().isPlaying
GetComponent<TBE_3DCore.TBE_Source>().loop
GetComponent<TBE_3DCore.TBE_Source>().maxDistanceMute
GetComponent<TBE_3DCore.TBE_Source>().maximumDistance
GetComponent<TBE_3DCore.TBE_Source>().minimumDistance
GetComponent<TBE_3DCore.TBE_Source>().mute
GetComponent<TBE_3DCore.TBE_Source>().pitch
GetComponent<TBE_3DCore.TBE_Source>().playOnAwake
GetComponent<TBE_3DCore.TBE_Source>().reverbToggle
GetComponent<TBE_3DCore.TBE_Source>().rollOffFactor
GetComponent<TBE_3DCore.TBE_Source>().roomToggle
GetComponent<TBE_3DCore.TBE_Source>().time
GetComponent<TBE_3DCore.TBE_Source>().timeSamples
GetComponent<TBE_3DCore.TBE_Source>().volume
```

Functions:

```
GetComponent<TBE_3DCore.TBE_Source>().Play();
GetComponent<TBE_3DCore.TBE_Source>().PlayDelayed(float delay);
GetComponent<TBE_3DCore.TBE_Source>().PlayOneShot(AudioClip clip);
GetComponent<TBE_3DCore.TBE_Source>().Stop();
```