

# Maker Skill Unity

SID2a

# Ciska Vriezenga

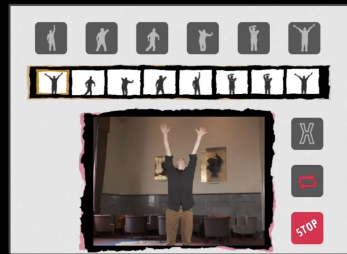
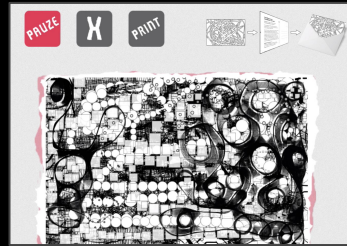
## Design Ontologist

... Playful Interaction Designer, Software  
Developer, Researcher, Lecturer

*Creative design practices, Sonic Interaction  
Design, Adaptive games*



School of  
Music & Technology  
Games



# ONTWERP & ONDERZOEK

structure #1



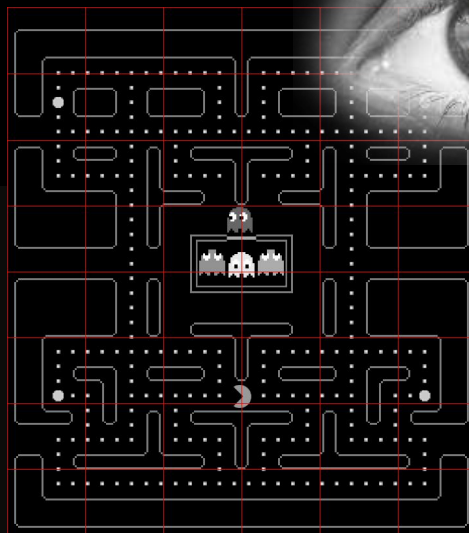
structure #2



structure #3

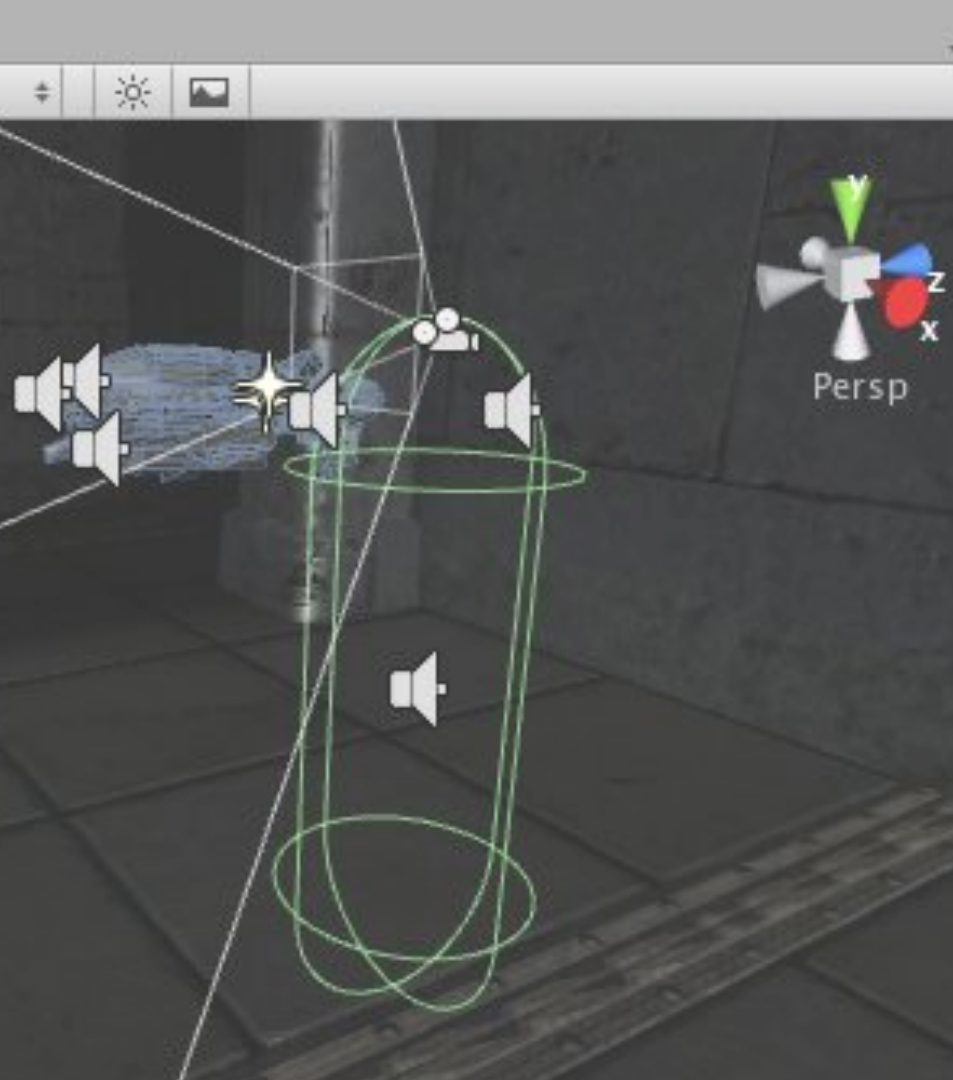


structure #4



Overview

Introduction

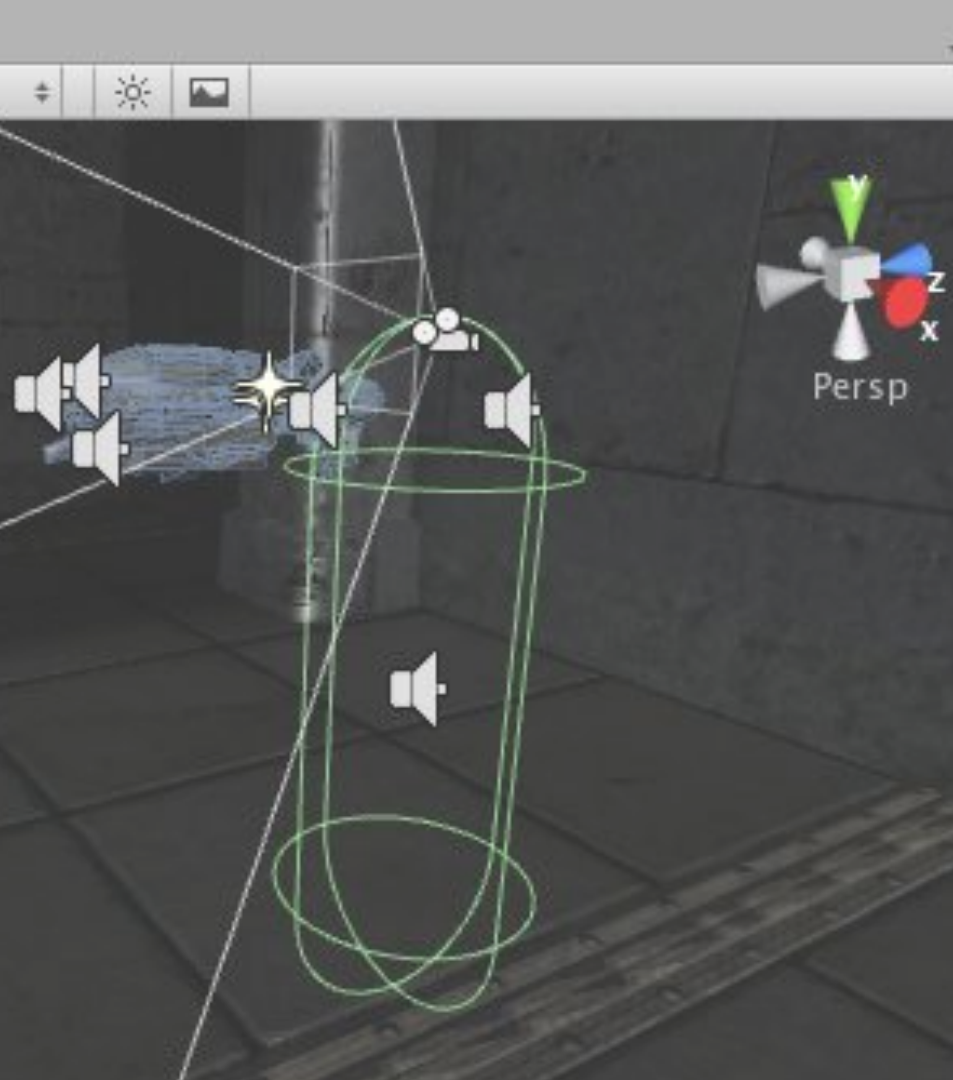


# Maker Skill Unity - Course

## Overall goal

**Understanding** of and **experience** with

- Audio sources
  - Audio Clips
  - Audio Random Containers
- Audio Listeners
- Audio Mixers
- Reverb zones

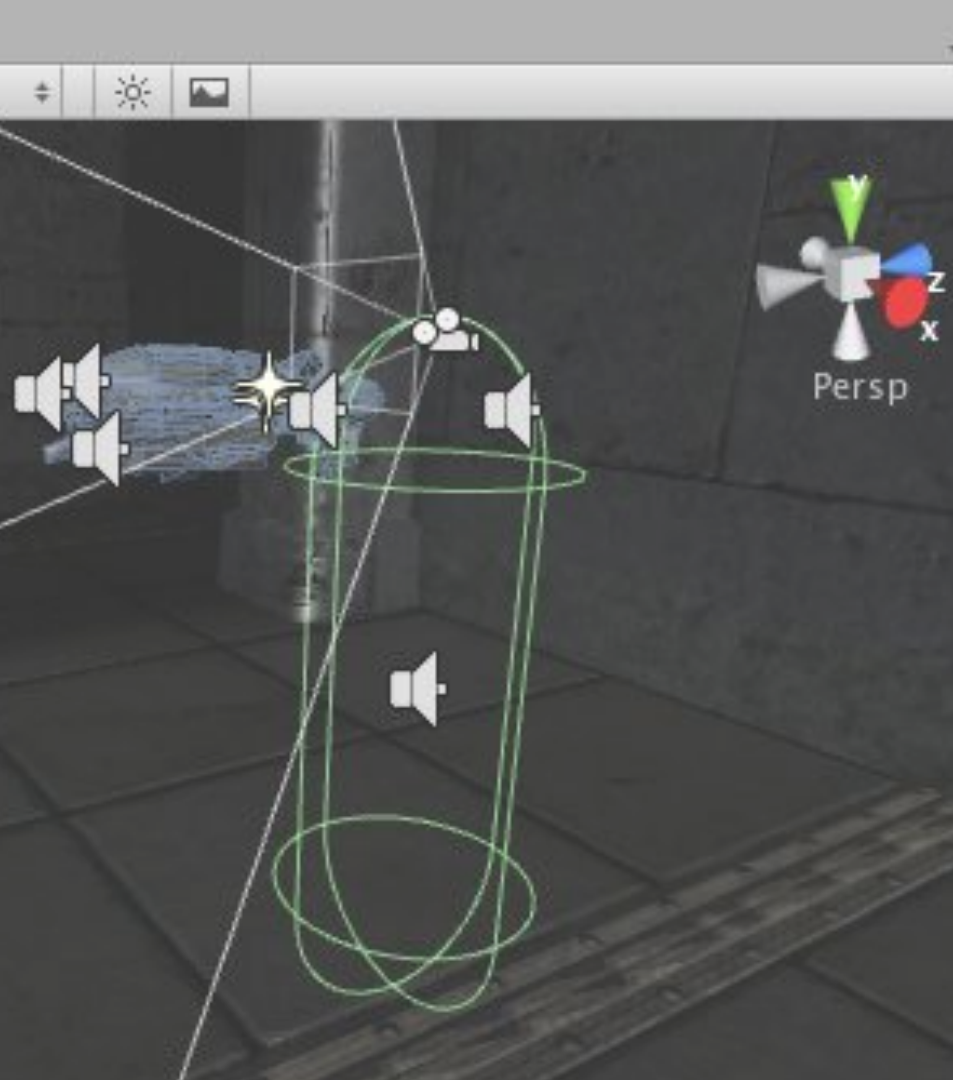


# Maker Skill Unity - Course

## Other topics

- Stingers
- Vertical remixing based on location
- Horizontal resequencing
  - Transitions, e.g. cutting; cross-fade; transition matrix.
  - Timing, e.g. immediate; on the beat; bar; phrase; custom markers

You will try to implement a selection

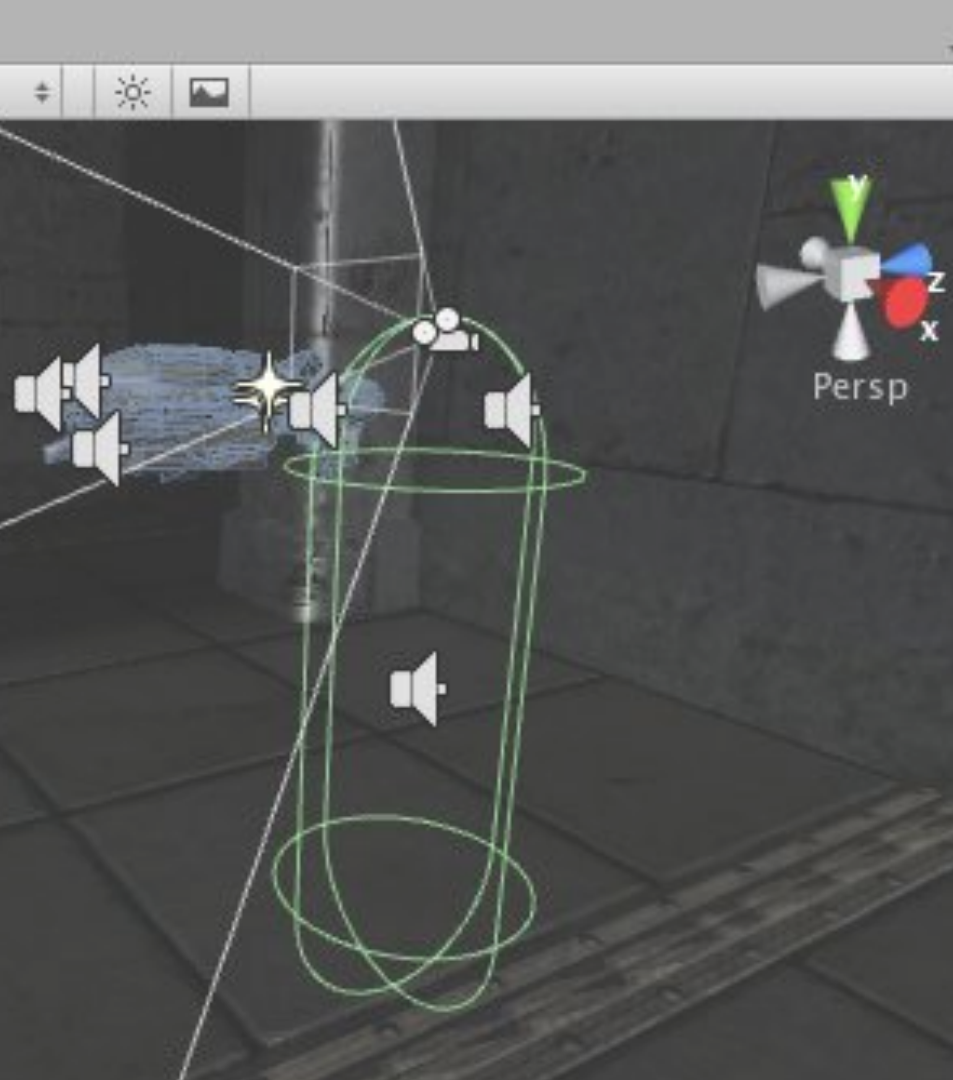


# Maker Skill Unity - Course

However ... course is **not** meant as a 'tutorial' / 'How to guide'

**Want those?**

→ Unity Learn environment



# Maker Skill Unity - Course

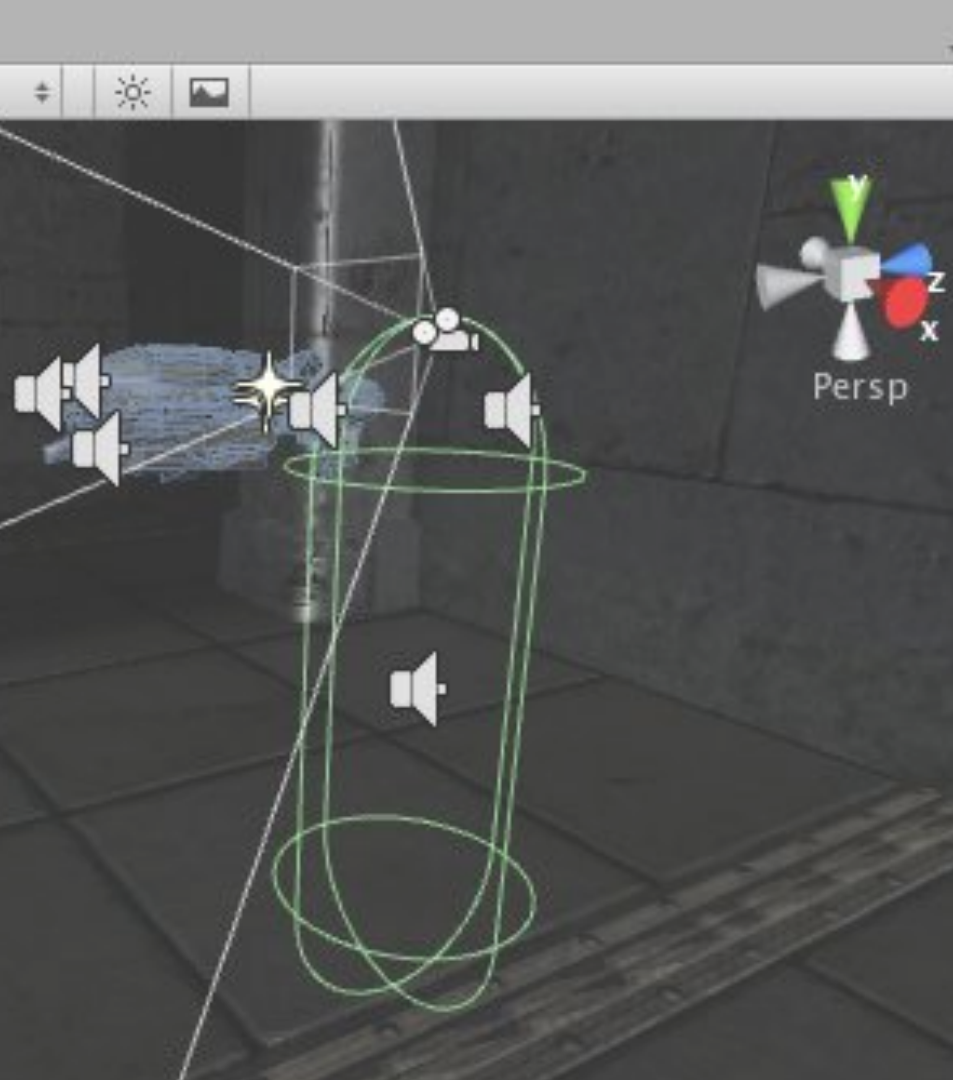
## Overall goal

**Understanding** of and **experience** with

- Audio sources
  - Audio Clips
  - Audio Random Containers
- Audio Listeners
- Audio Mixers
- Reverb zones

***In Unity, but even better ...***



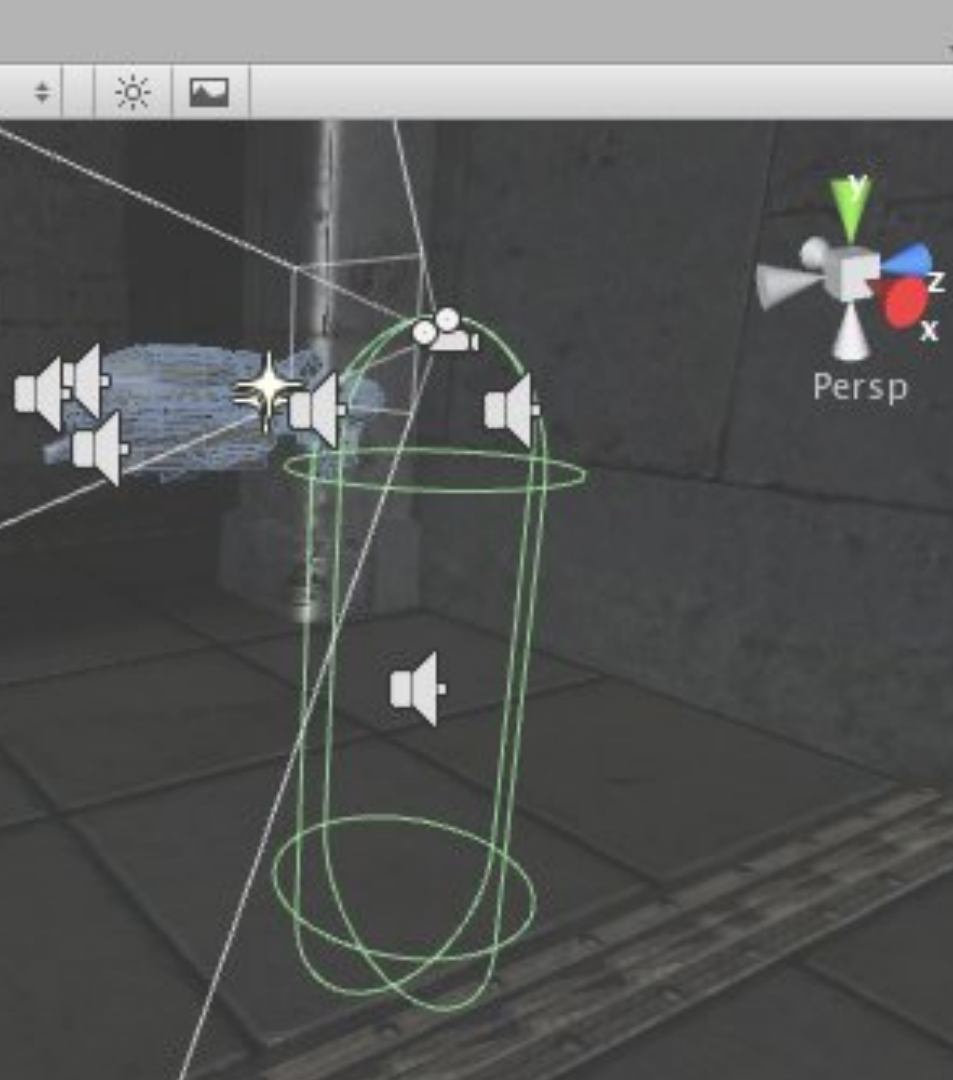


# Maker Skill Unity - Course

## Overall goal

Understanding of **the concepts** of

- Audio sources
  - Audio Clips
  - Audio Random Containers
- Audio Listeners
- Audio Mixers
- Reverb zones

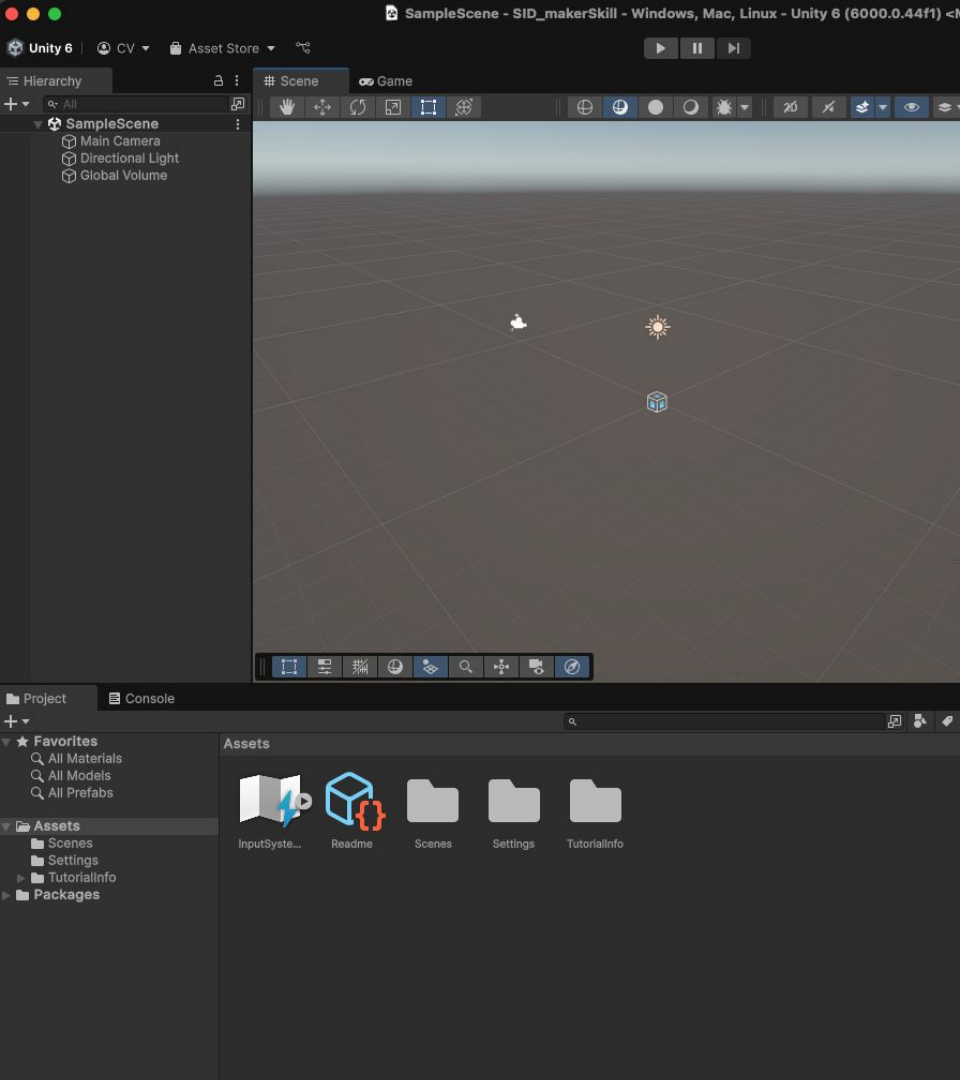


# Maker Skill Unity - Course

Today - we start by **constructing** the **'mental model'**!

Start download

But first things first



# Fetch Unity

- Create unity student account
- Install Unity Hub, open and login
- Install most recent Unity version  
→ *start download*

&

- Voor c-sharp code:
  - *Visual Studio Code*
- OF
  - *Jetbrains' Rider (github student account nodig)*

Mapping functionality

Mental model

# Mapping functionality

*“What functionality is required in a game environment to implement a simple game audio design?”*



Audio Sources and Listener

E.g.:

- Trigger a sound when a diegetic game event happens
- Change the music track based on ...

**Map** as many as possible!  
[individual] - 15 minutes

# Mapping functionality

*“What functionality is required in a game environment to implement a simple game audio design?”*

E.g.:

- Trigger a sound when a diegetic game event happens
- Change the music track based on ...



Audio Sources and Listener

**Combine** the mapped functionality  
[duo's] - 10 minutes

# Mapping functionality

*“What functionality is required in a game environment to implement a simple game audio design?”*



Audio Sources and Listener

E.g.:

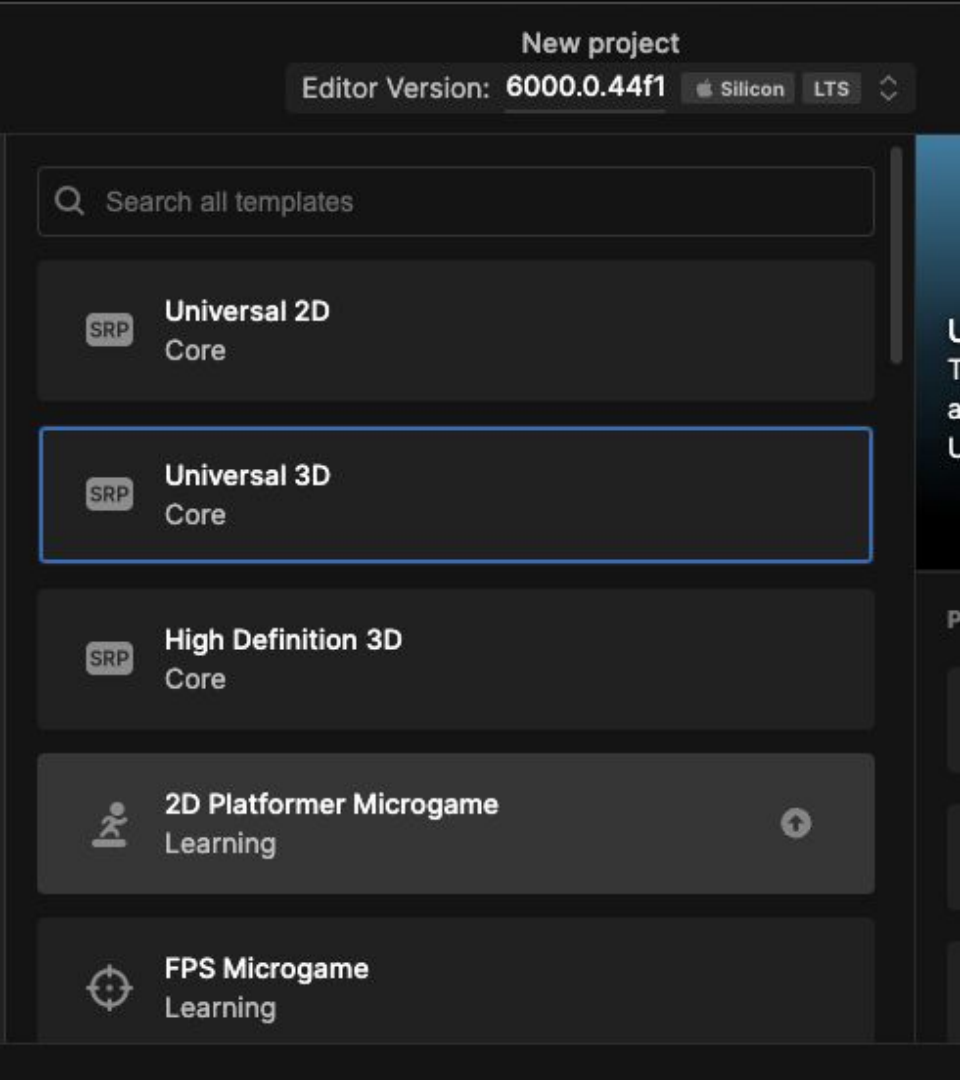
- Trigger a sound when a diegetic game event happens
- Change the music track based on ...

**Reflection!**  
**REFLECTION!**



Hello world

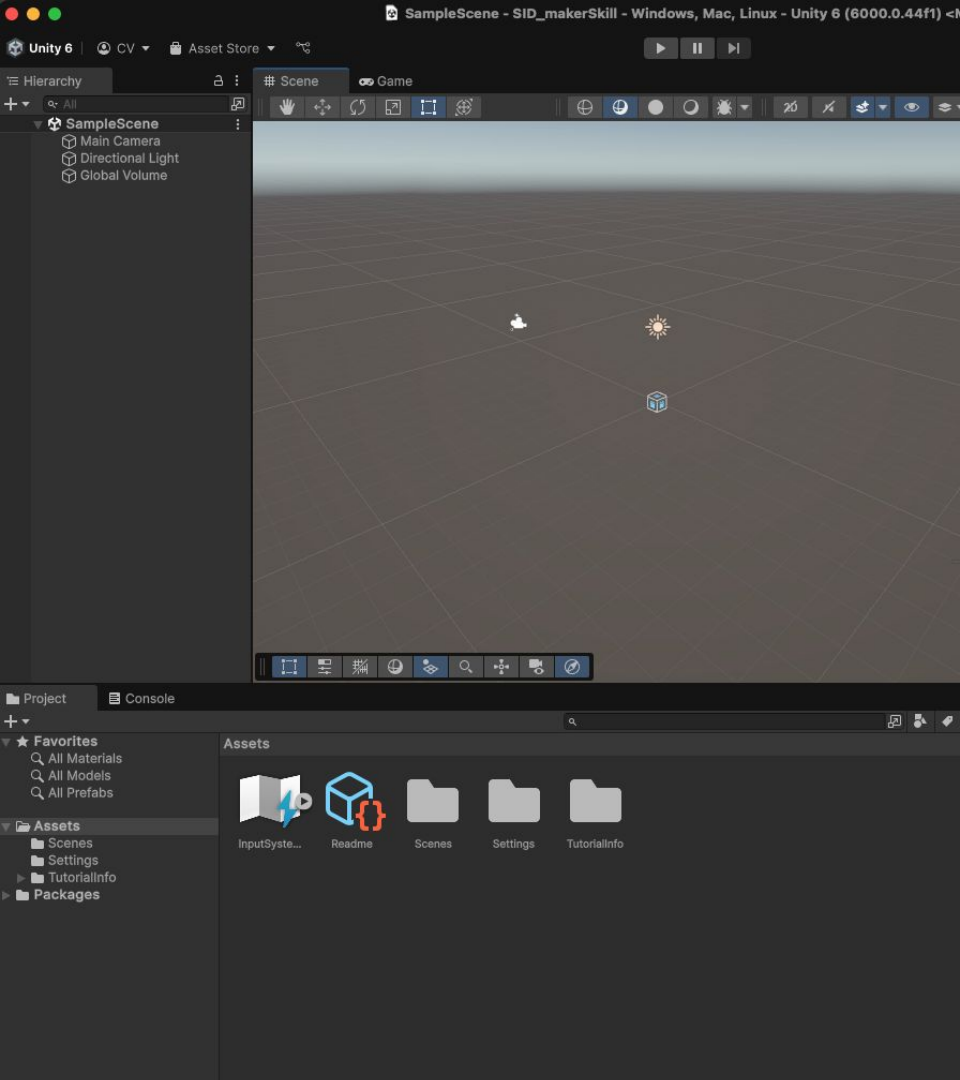
In sound



# Start new project

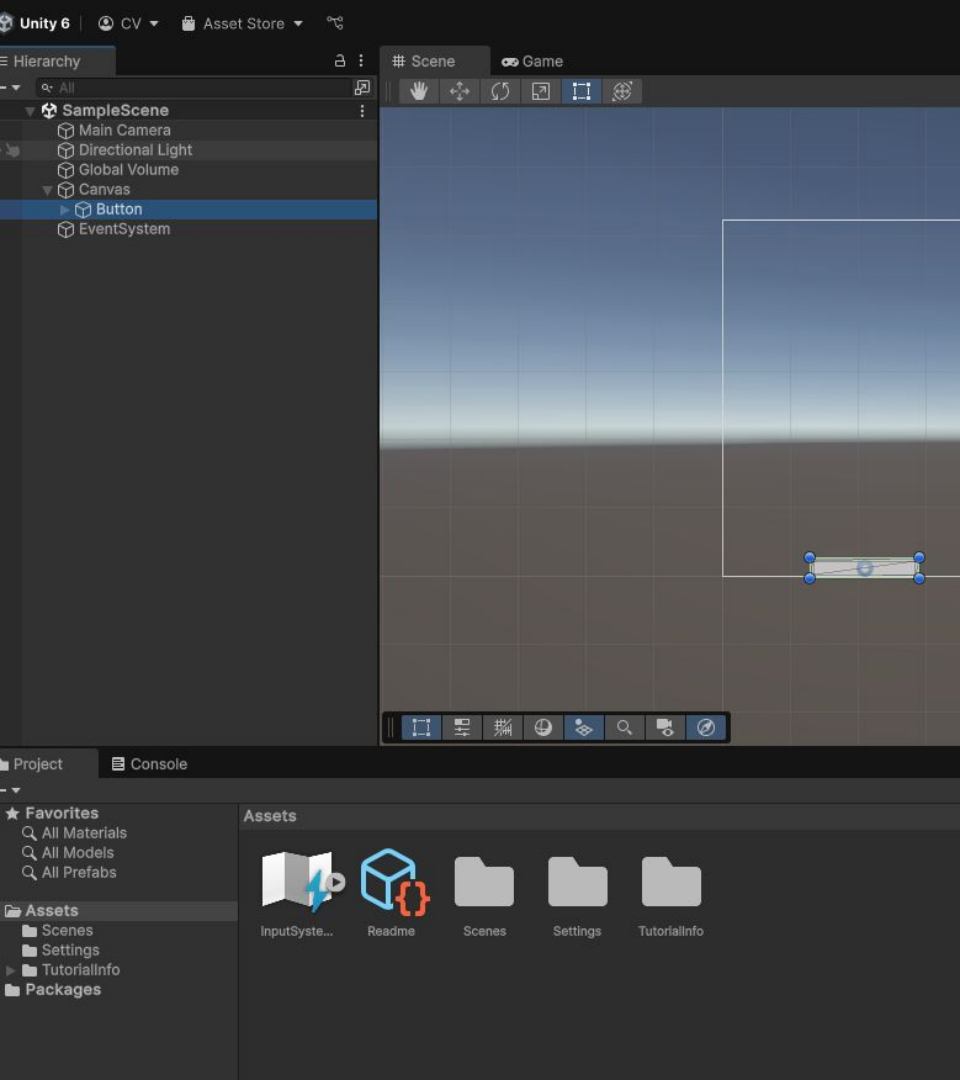
## In Unity Hub 'Projects'

- [New project] - *click!*
- Select 'Universal 3D'
- Configure
  - Name: [SID2a](#)
  - Location: *... suit yourself*
- [Create Project] - *click!*



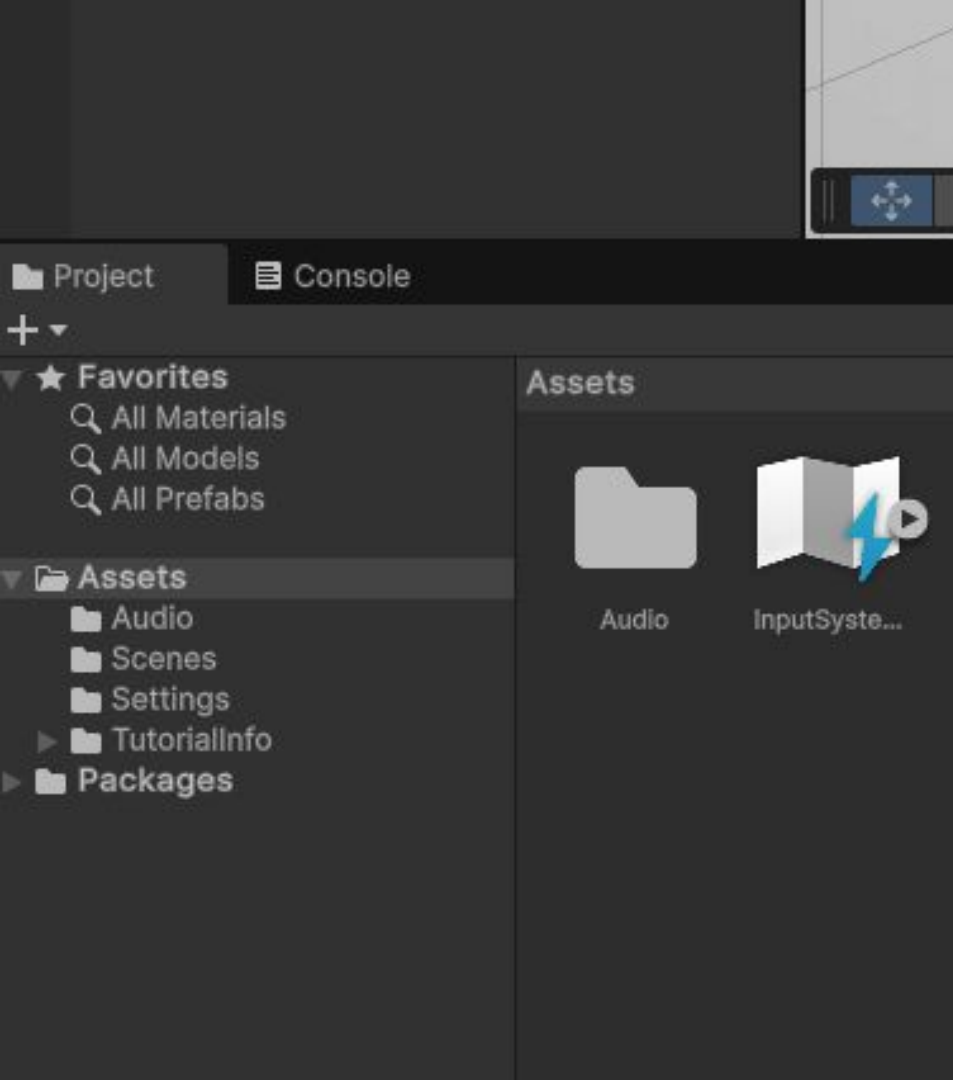
# “Hello Bleep!”

You now have a new project.



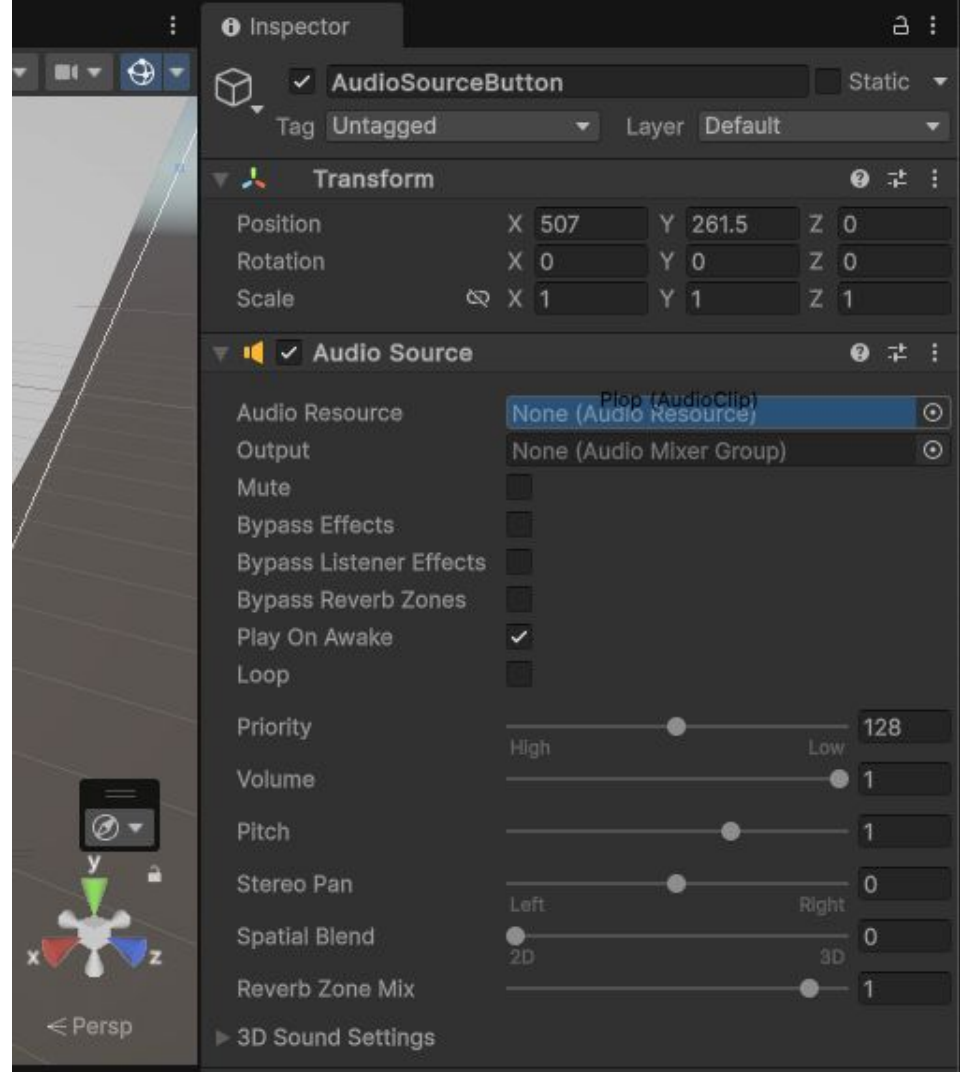
# “Hello Bleep!”

- Add a button  
*Think, search, use google, but no AI*
- Switch to 2D  
*Search for the [2D] button*
- In the Hierarchy panel, select Canvas and then press F  
*F → make fully visible in [scene panel]*
- Position button to your liking  
*1. Select button.  
2. Select move tool @top scene view  
OR use the Inspector @right*



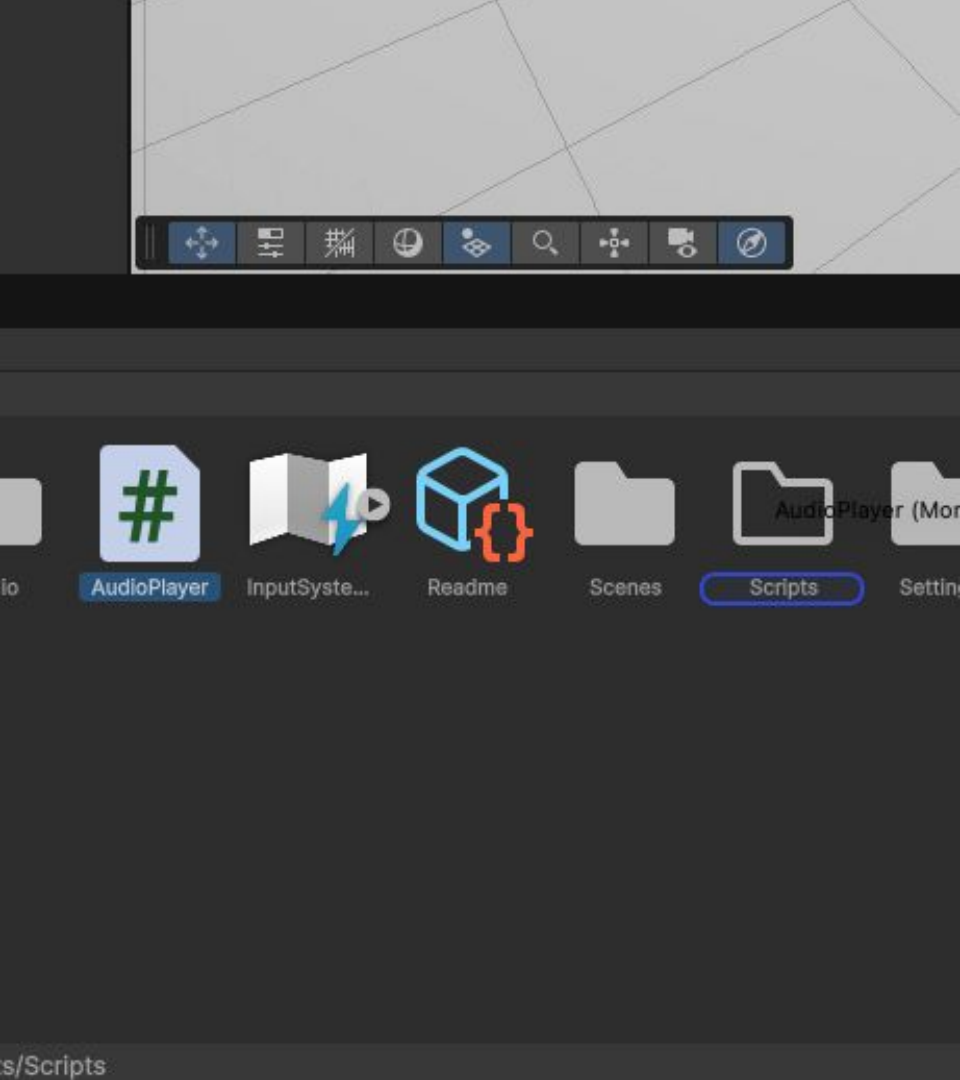
## “Hello Bleep!”

- Drag a short audiofile into the assets plane @Bottom
- Clean up assets folder structure; create ‘Audio’ Folder and drag audiofile into it



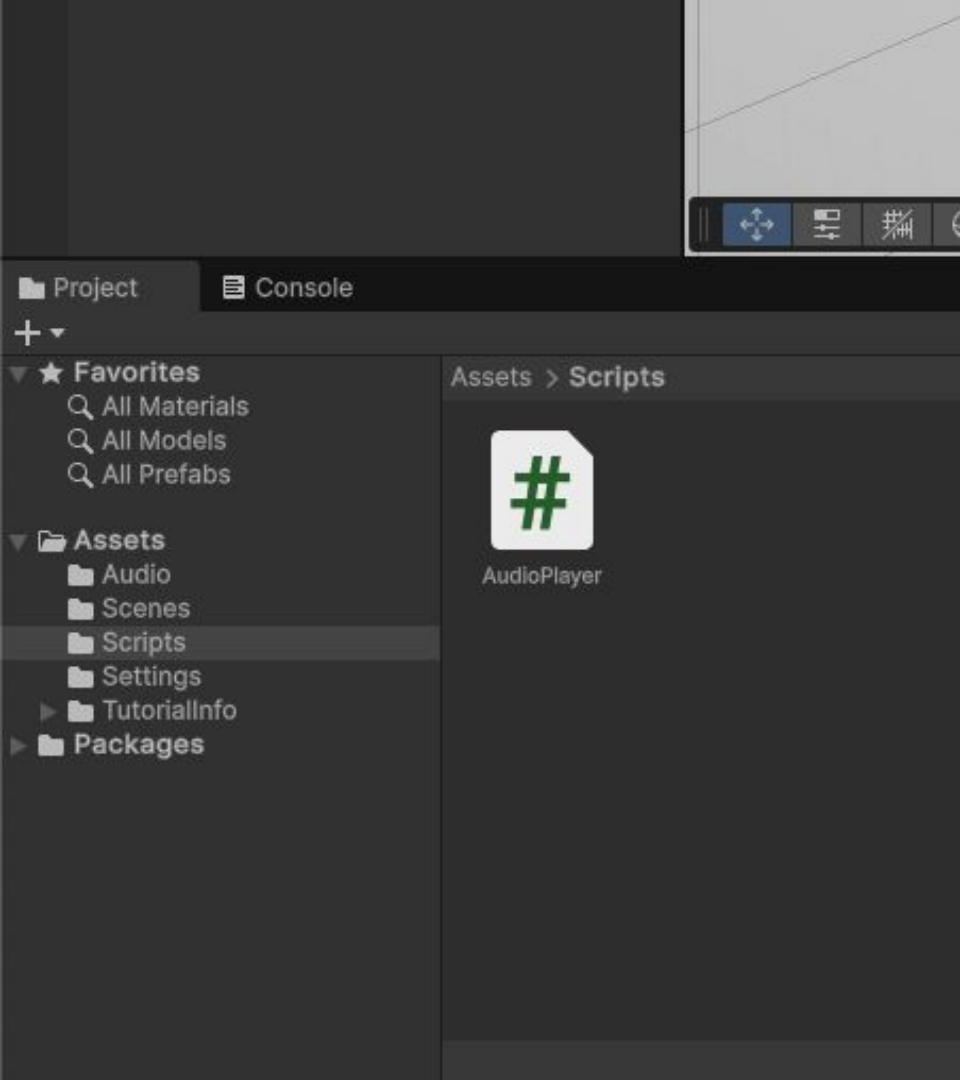
# “Hello Bleep!”

- Create empty object in Hierarchy  
→ rename ‘AudioSourceButton’
- Add an Audio Source to AudioSourceButton
  - *AudioSourceButton selected in Hierarchy*
  - *[Add component] - click!*
  - *[Audio/Audio Source] - click!*
- Deselect “Play On Awake”  
*See Inspector AudioSourceButton*
- Drag imported audiofile (*prev. slide*) to Audio Resource



## “Hello Bleep!”

- Create empty object in Hierarchy  
→ rename ‘AudioPlayer’
- Add a script to AudioPlayer
  - *AudioPlayer selected in Hierarchy*
  - *[Add component] - click!*
  - *[New script] - click!*
  - *Name: AudioPlayer*
- Clean up assets folder structure;  
create ‘Scripts’ Folder and drag  
AudioPlayer script into it



# “Hello Bleep!”

- Install IDE to edit scripts  
*Unity supports Visual Studio, Visual Studio Code, JetBrains Rider*  
<https://docs.unity3d.com/6000.1/Documentation/Manual/scripting-ide-support.html>  
*For Rider → arrange github account, apply for student pack, get free access to JetBrains software*
- Open AudioPlayer script - *2xclick!*



# “Hello Bleep!”

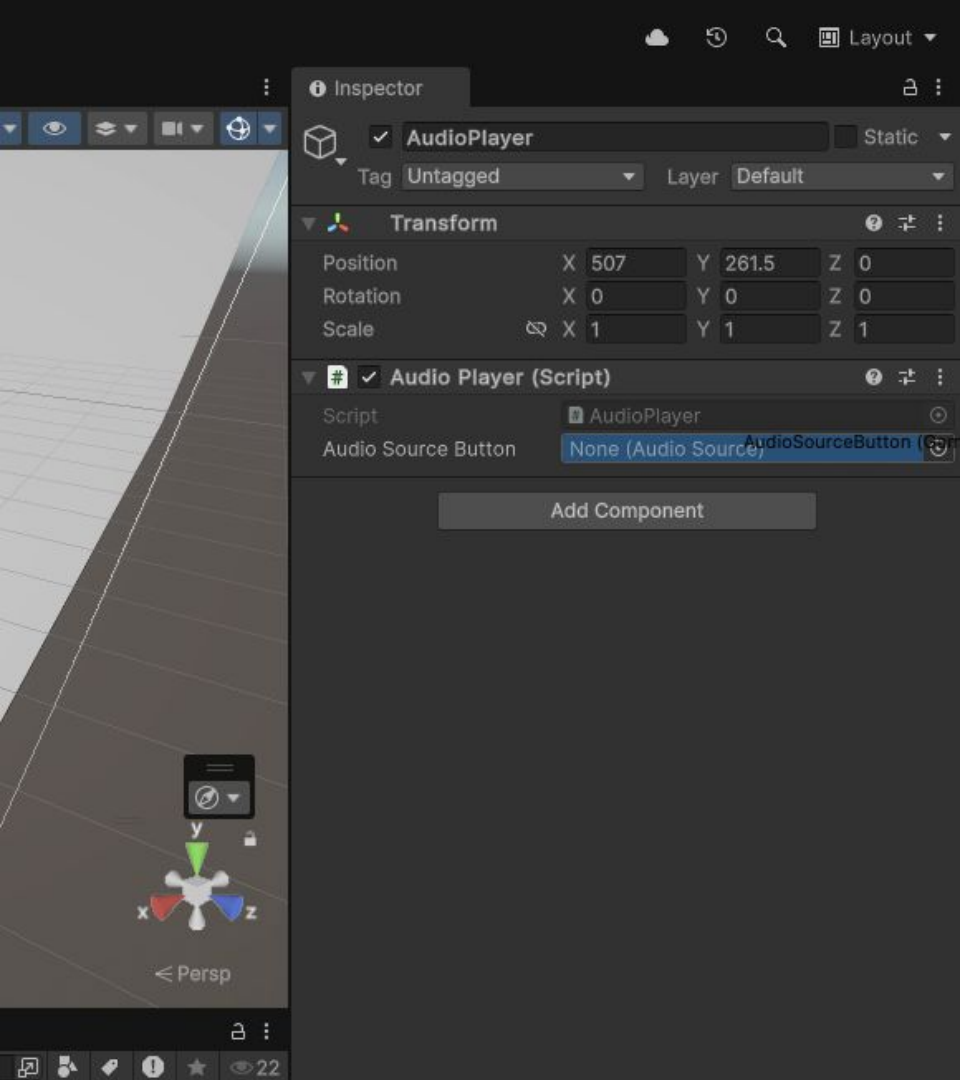
*In the AudioPlayer script*

- Remove Start and Update methods
- Add reference to an object of the class AudioSource.
- Add function to play AudioSource

*public → necessary to allow access from within the Unity Editor*

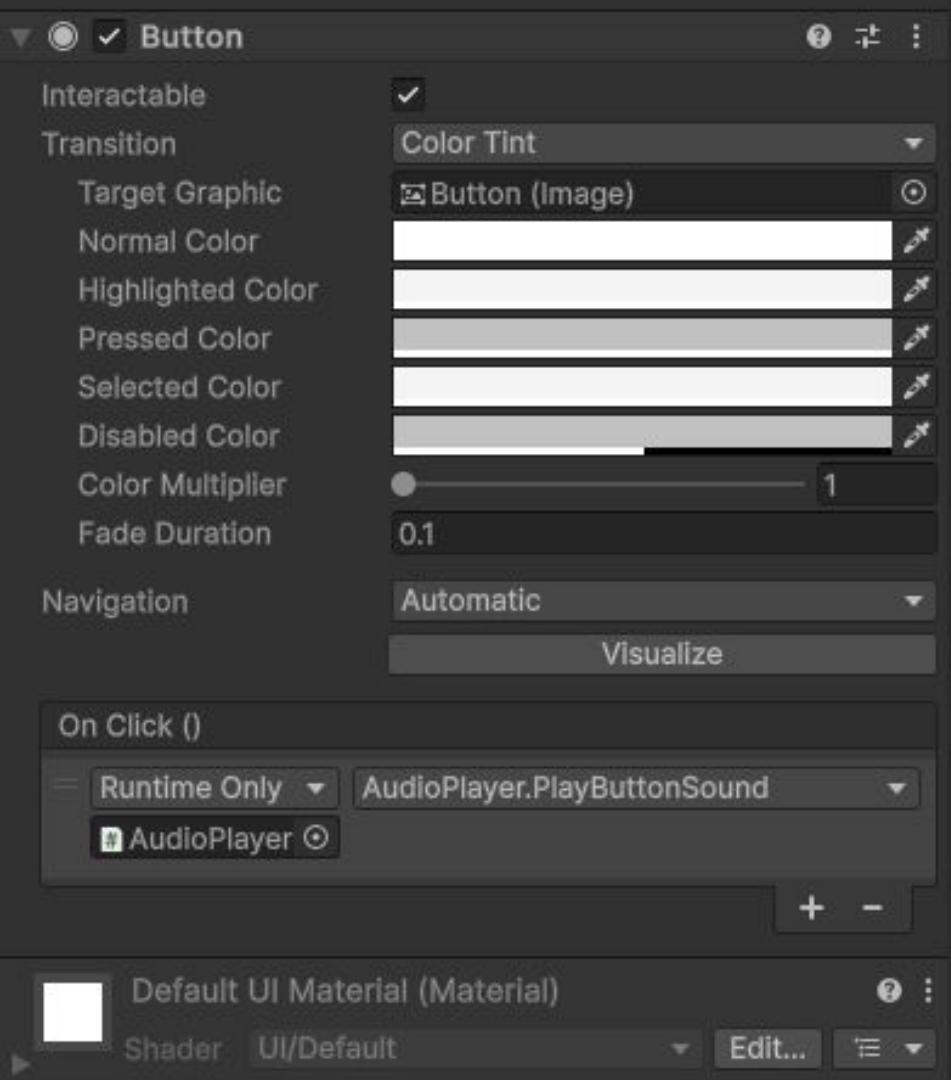
C# AudioPlayer.cs x

```
1 using UnityEngine;
2
3 ^o 1 asset usage
4 public class AudioPlayer : MonoBehaviour
5 {
6     public AudioSource audioSourceButton;
7
8     1 asset usage
9     public void PlayButtonSound()
10    {
11        audioSourceButton.Play();
12    }
13 }
```



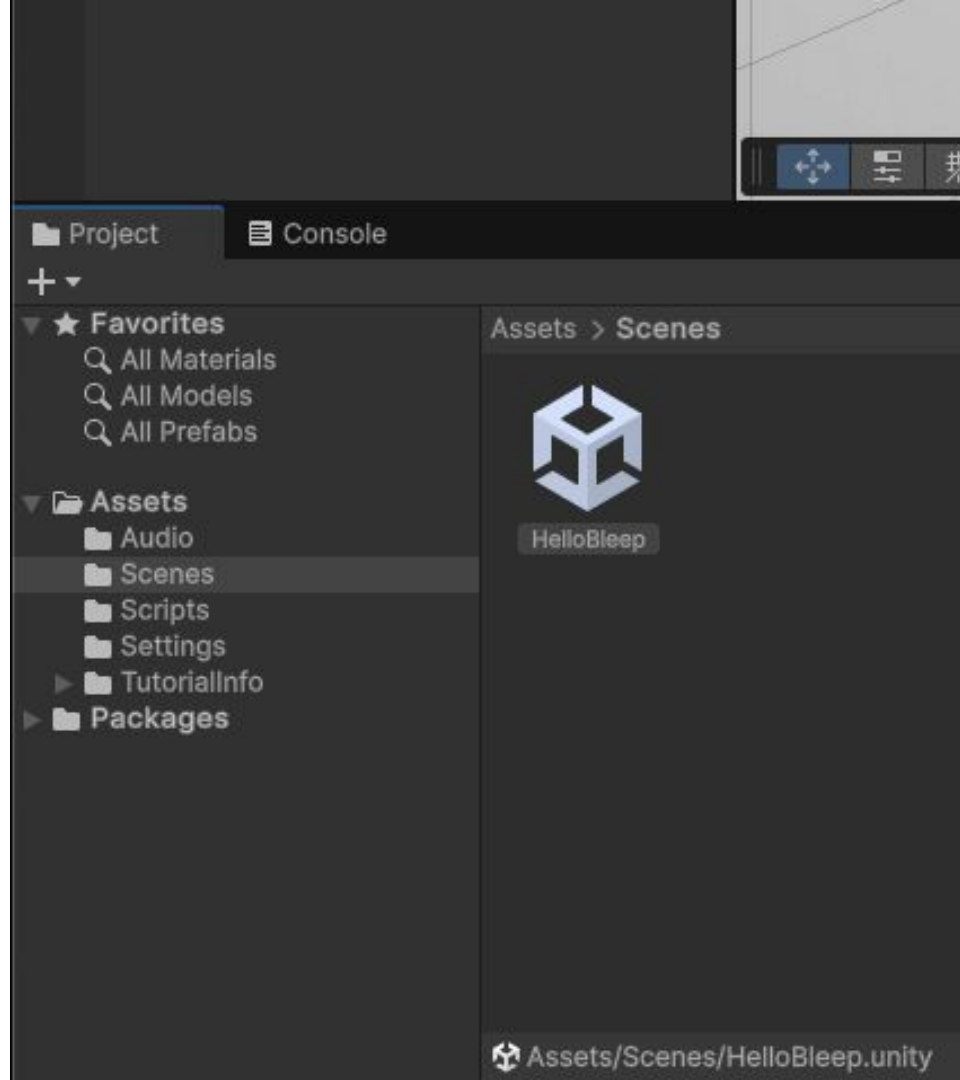
# “Hello Bleep!”

- Select AudioPlayer
- Drag AudioSourceButton from Hierarchy to AudioPlayer's [Audio Source Button] in inspector



## “Hello Bleep!”

- Select Button
- Select AudioPlayer in On click () element and select the `AudioPlayer.PlayButtonSound` function



# “Hello Bleep!”

- In Assets panel rename ‘SampleScene’ in Scenes folder to ‘HelloBleep’

DONE!

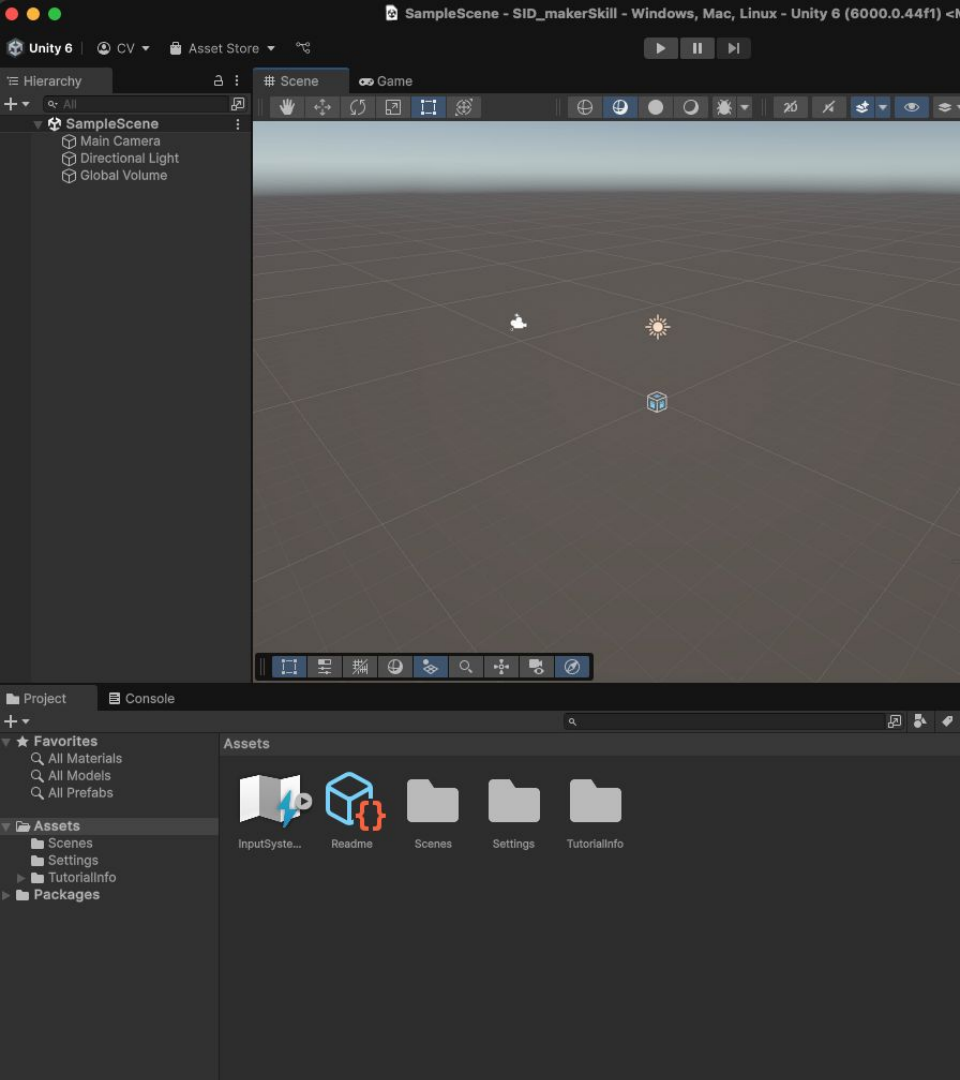
*click play button @top*



*click button in Game view*

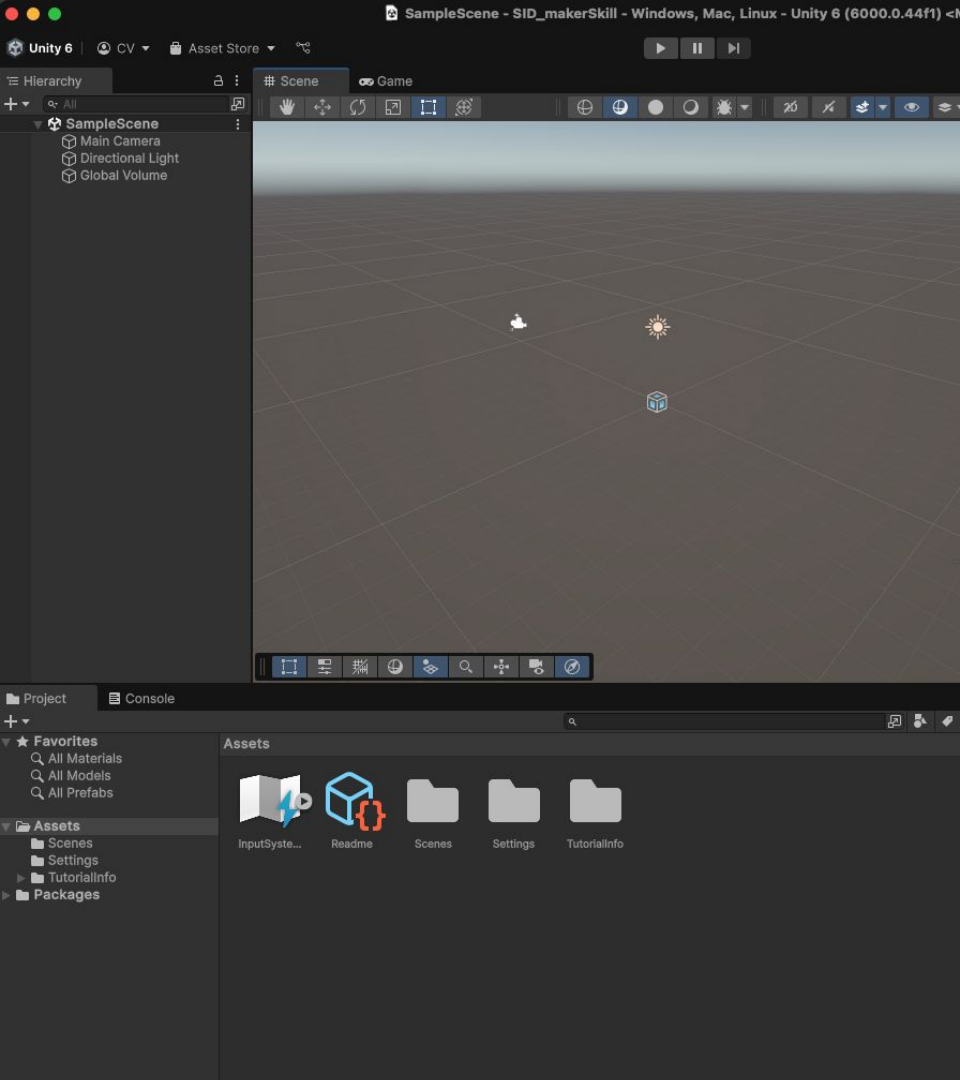
Unity elements

In summary



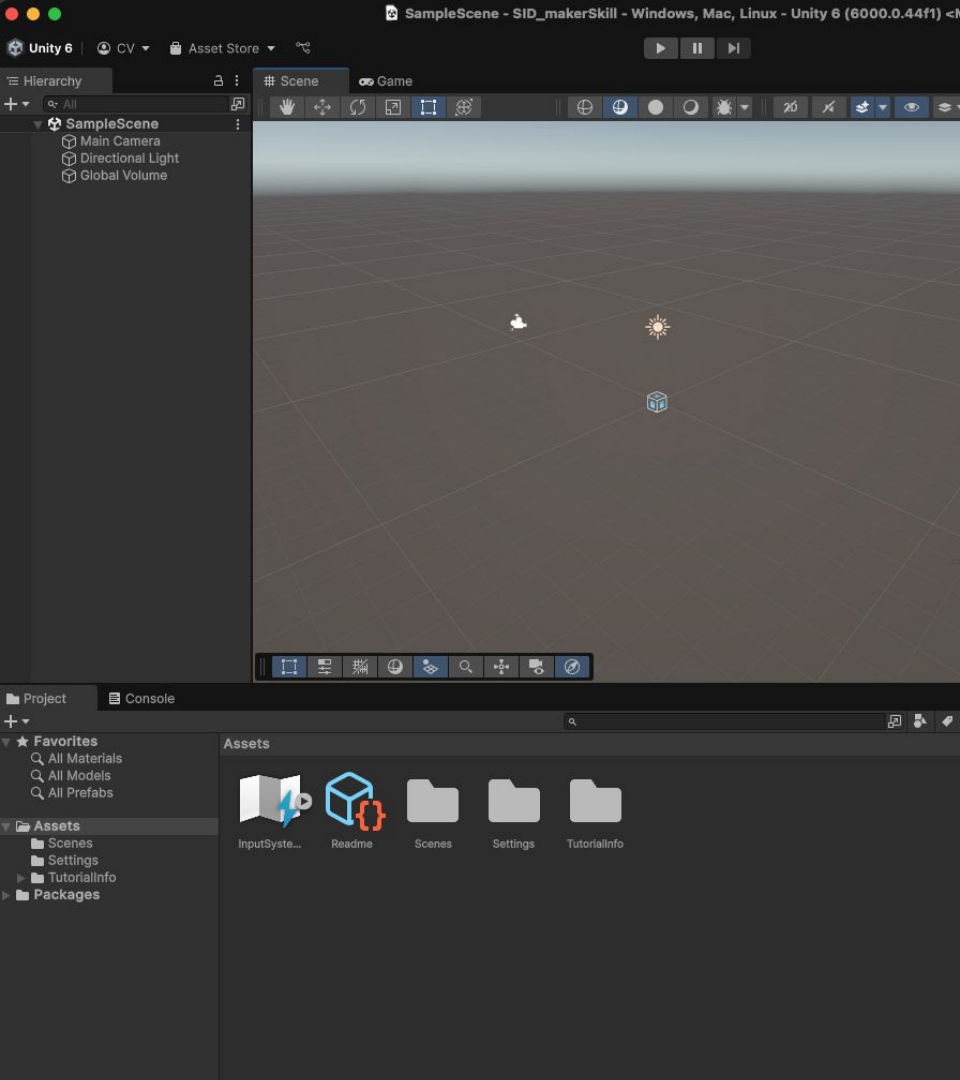
# Disclaimer

Slides were intended as starting point  
→ Follow-up sessions no extensive slides!



# Unity elements

- Hierarchy
- Inspector
- Assets
- Scene view / Game view
- Components
  - Audio Source
  - Script
- References
- Functions



# Up next

Design a 3D experience, with at least

- Audio sources
- Audio Listeners
- Audio Mixers
- Reverb zones

And possibly

- Randomization
  - (Audio Random Containers & effects)
- Stingers
- Vertical remixing based on location
- Horizontal resequencing
  - Transitions, e.g. cutting; cross-fade; transition matrix.
  - Timing, e.g. immediate; on the beat; bar; phrase; custom markers