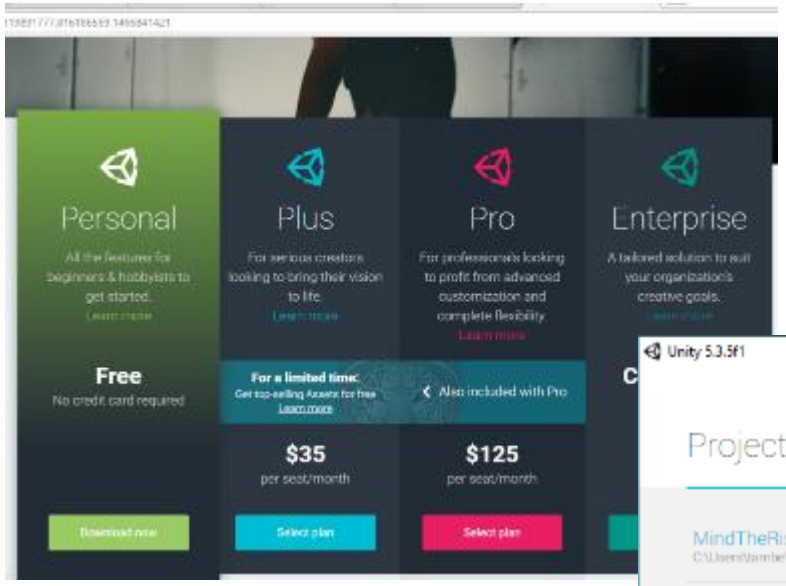
Unity introduction & Leap Motion Controller

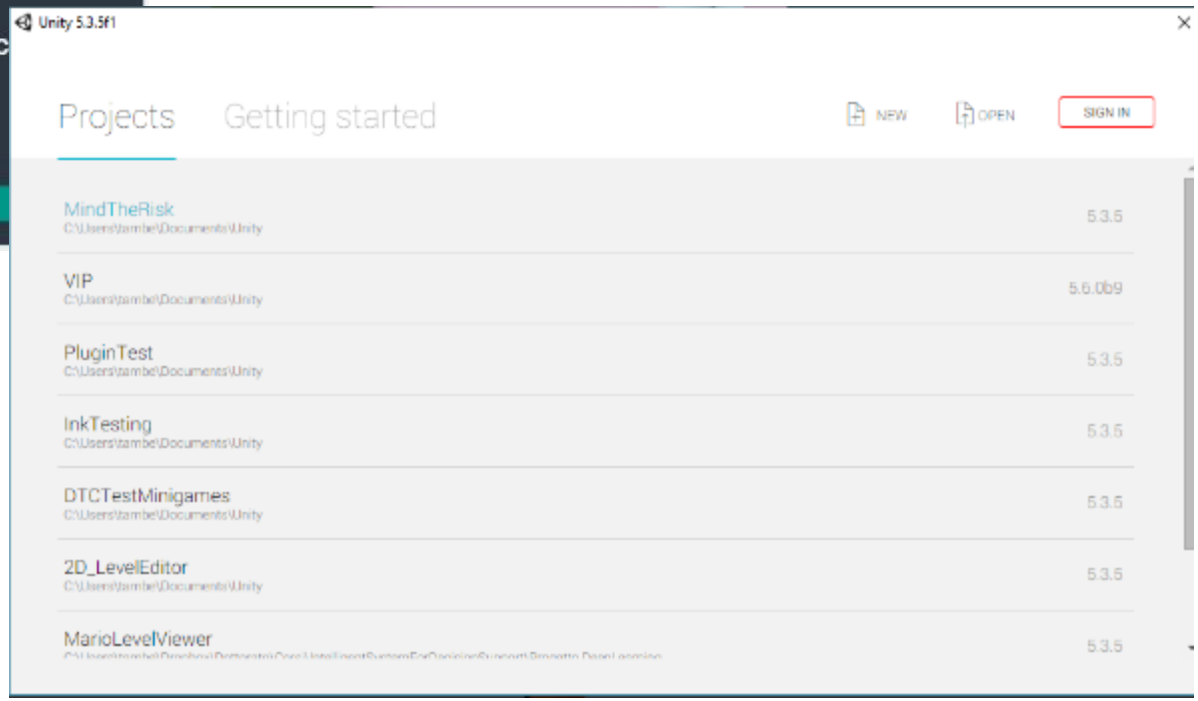
Renato Mainetti
Jacopo Essenziale

renato.mainetti@unimi.it
jacopo.essenziale@unimi.it

Lab 04

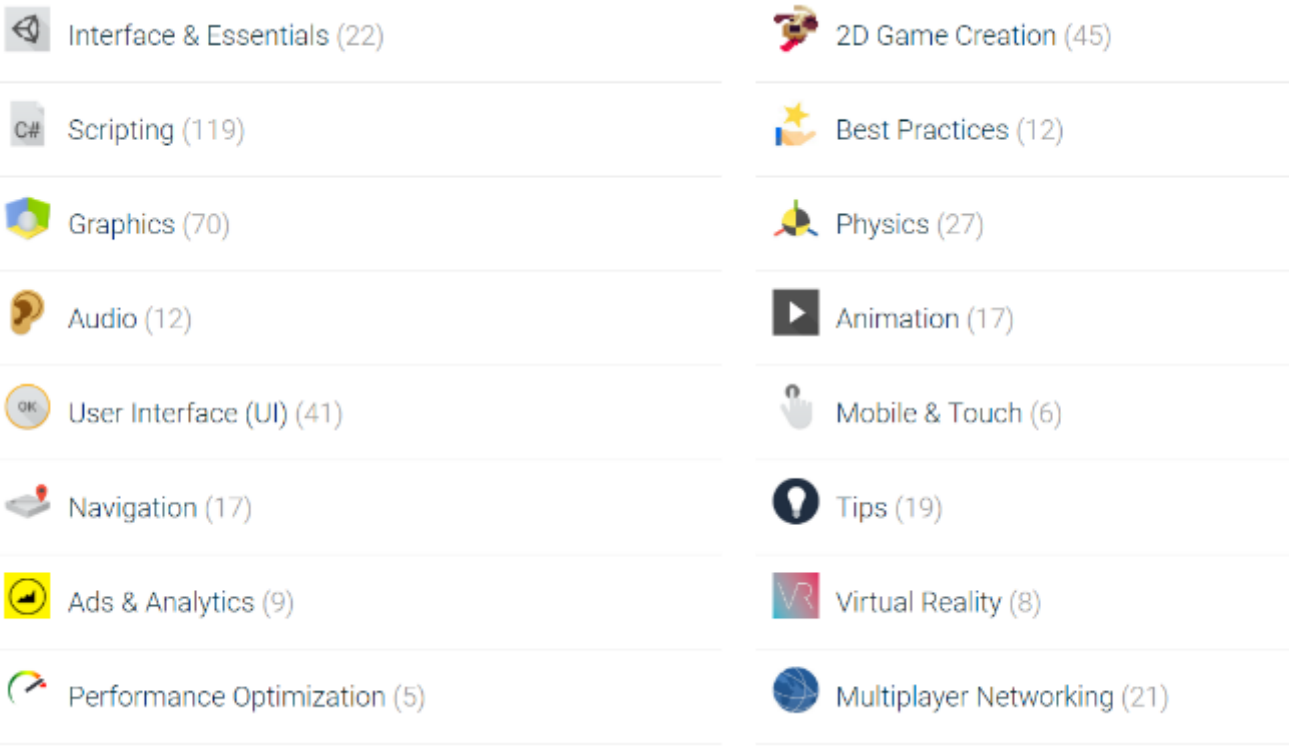
Unity 3D Game Engine



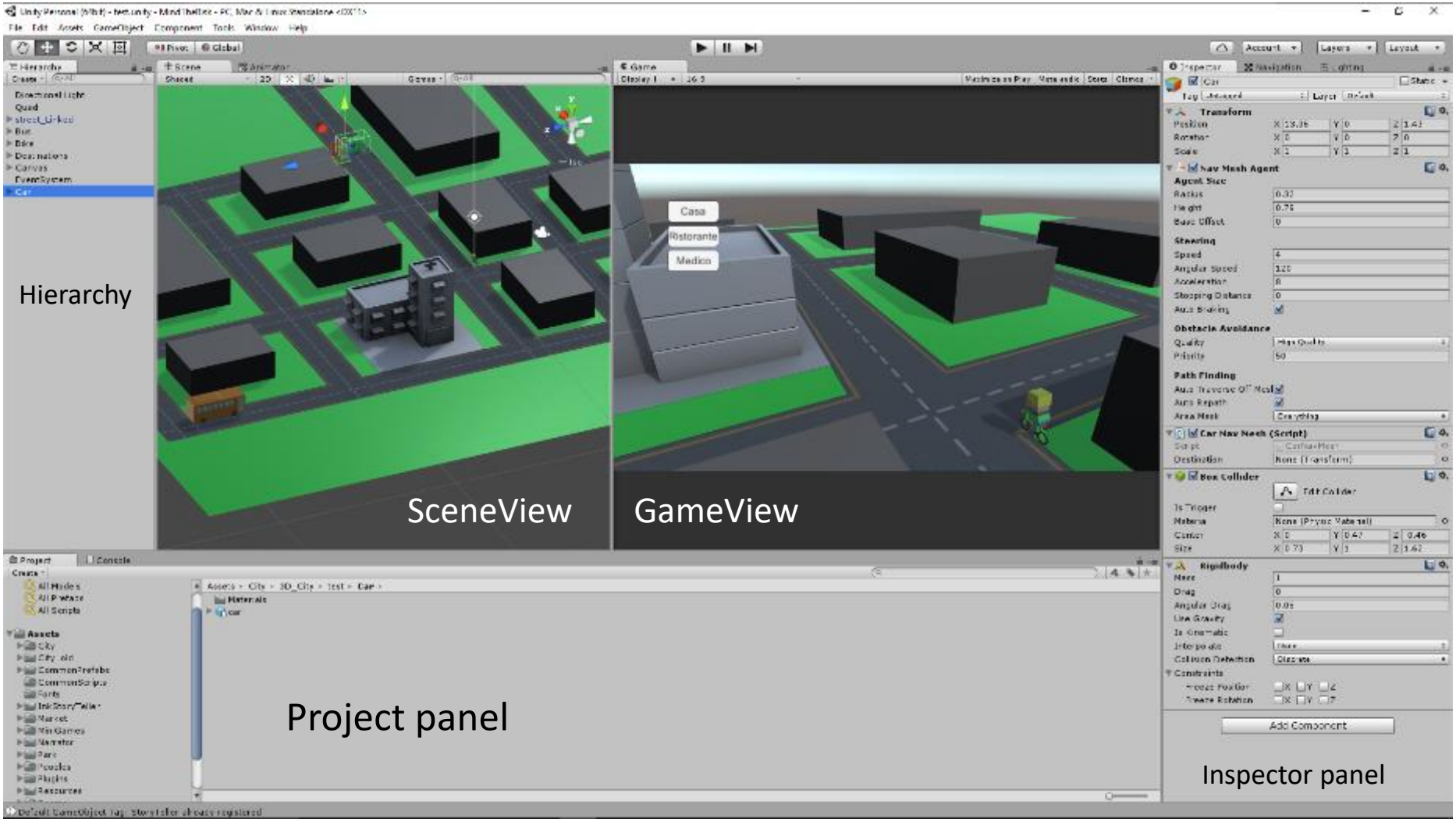


Official Unity 3D Tutorials

- https://unity3d.com/learn/tutorials/



Interface Overview



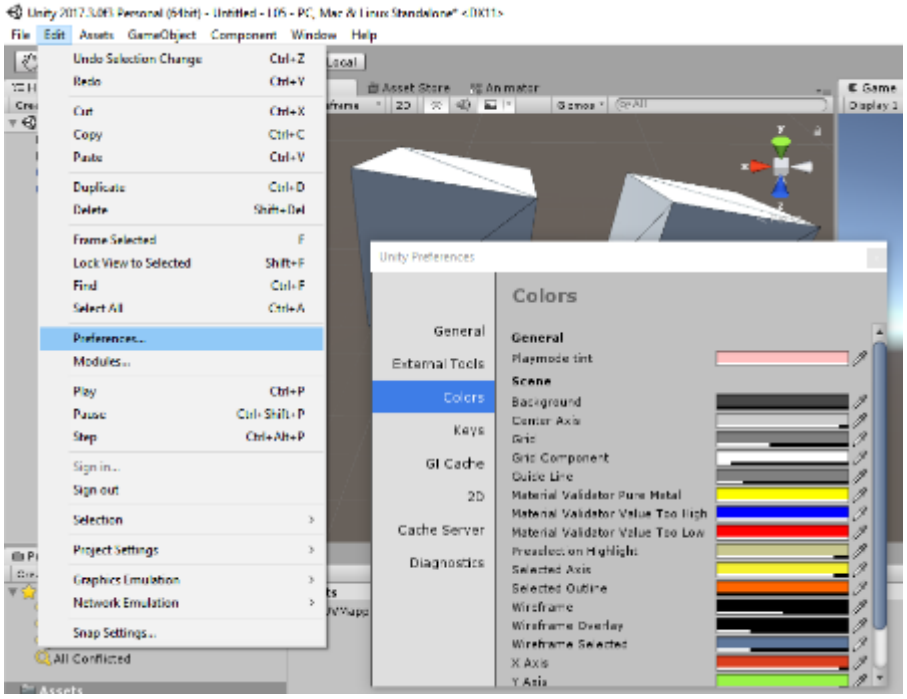
Scene View:

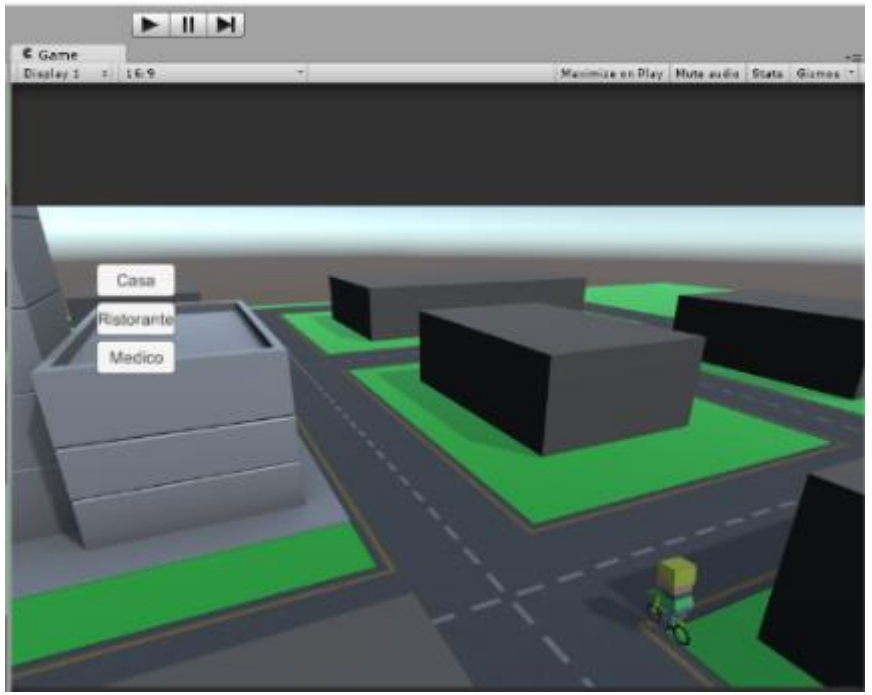
- qwert -> 
- Left mouse click = PAN
- Right mouse click = Look around like FPS
- Alt + Left = Orbit around your looking point
- Alt + Right = Zoom
- wer = position - rotation - scale
- Center-Pivot, Global-Local 

- Focus to an object: dbl-click in hierarchy or press f
- Several display mode (Shaded,wireframe, etc…)
- Search scene, fade out all the other objects
- Light on, sky and audio preview
- Perspective 3D view to Ortho 

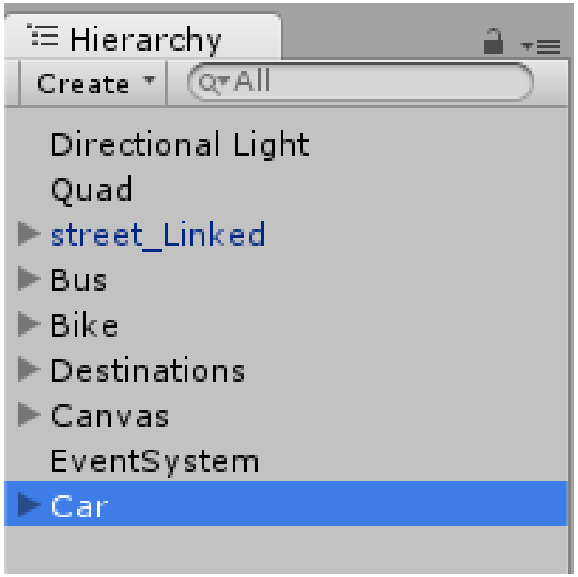
Game View:

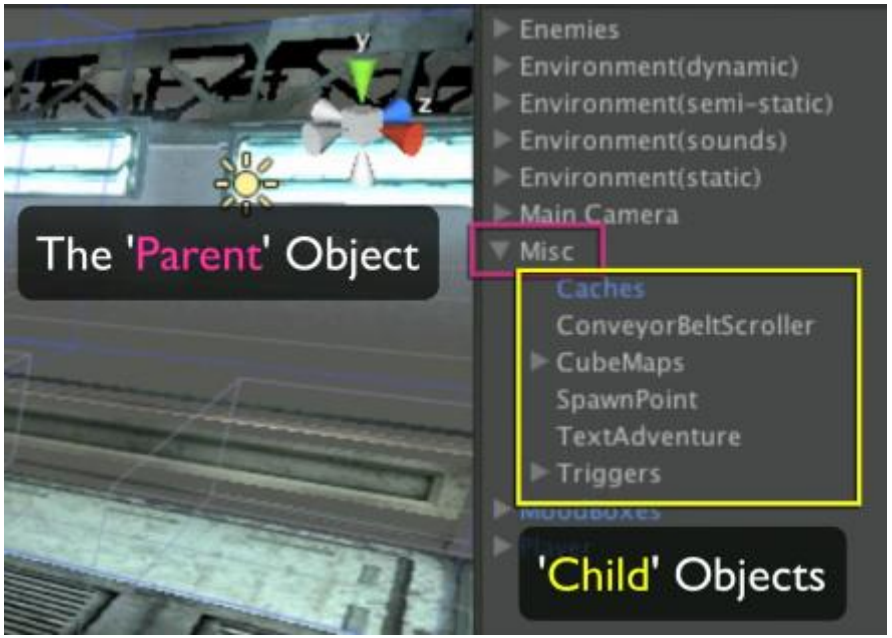
- Active when play button is pressed (playmode tint)
- When in play mode every change is not saved
- Paused and stepping forward frame by frame
- Aspect ratio and resolution (to test deploy platforms)





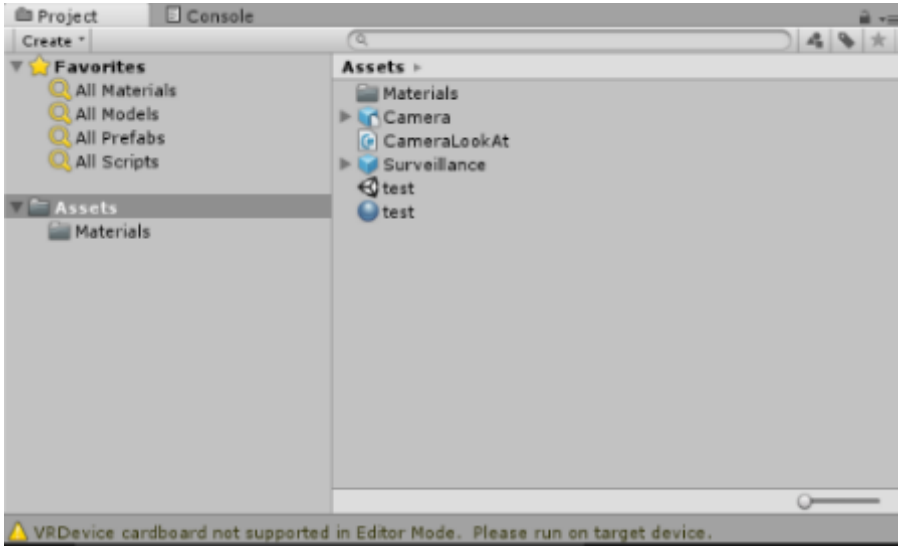
Hierarchy Panel:





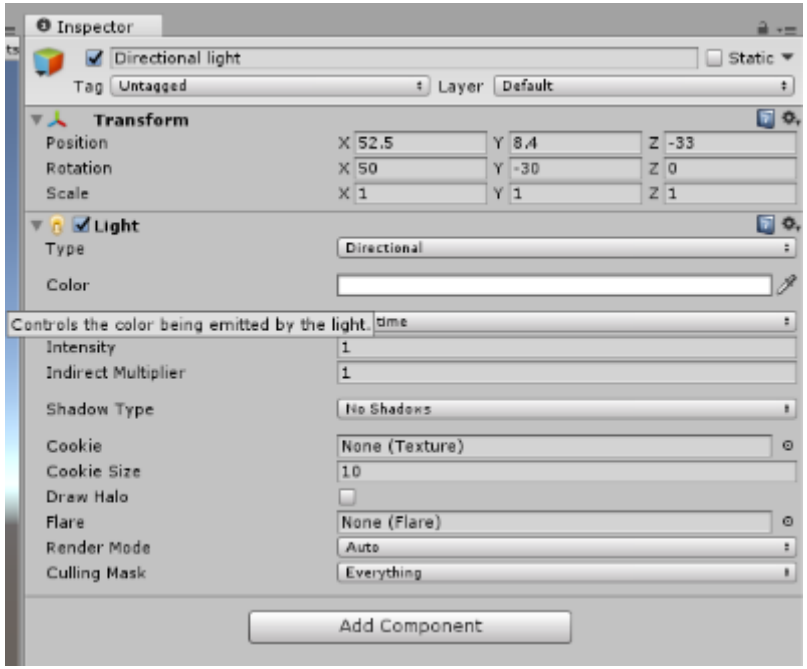
- If the parent is the world, RTS relatively to the world
- If it's child, RTS relatively to it's parent
- Create button is the same as -> game object – create other
- Search like the one in the hierarchy view
- Search could be filtered by type: t:Light, t:AudioSource

Project Panel:



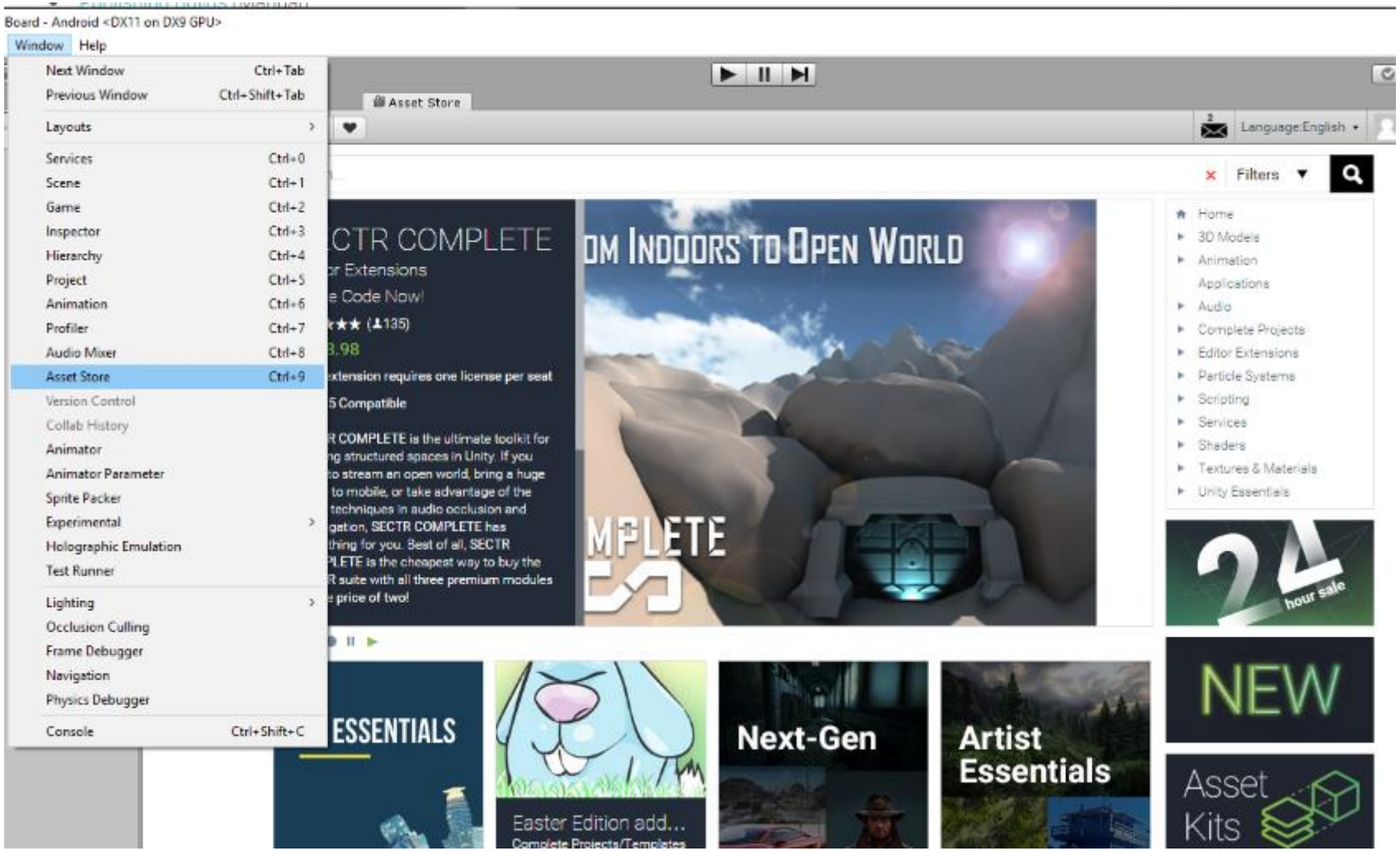
- Shows all the contents in Asset folder
- Used to organize all the assets
- New object could be imported(into the folder or by drag and drop) or created by right click mouse
- Search for assets
- Thumbnail using the zoom slider
- Filter by type and labels

Inspector Panel:

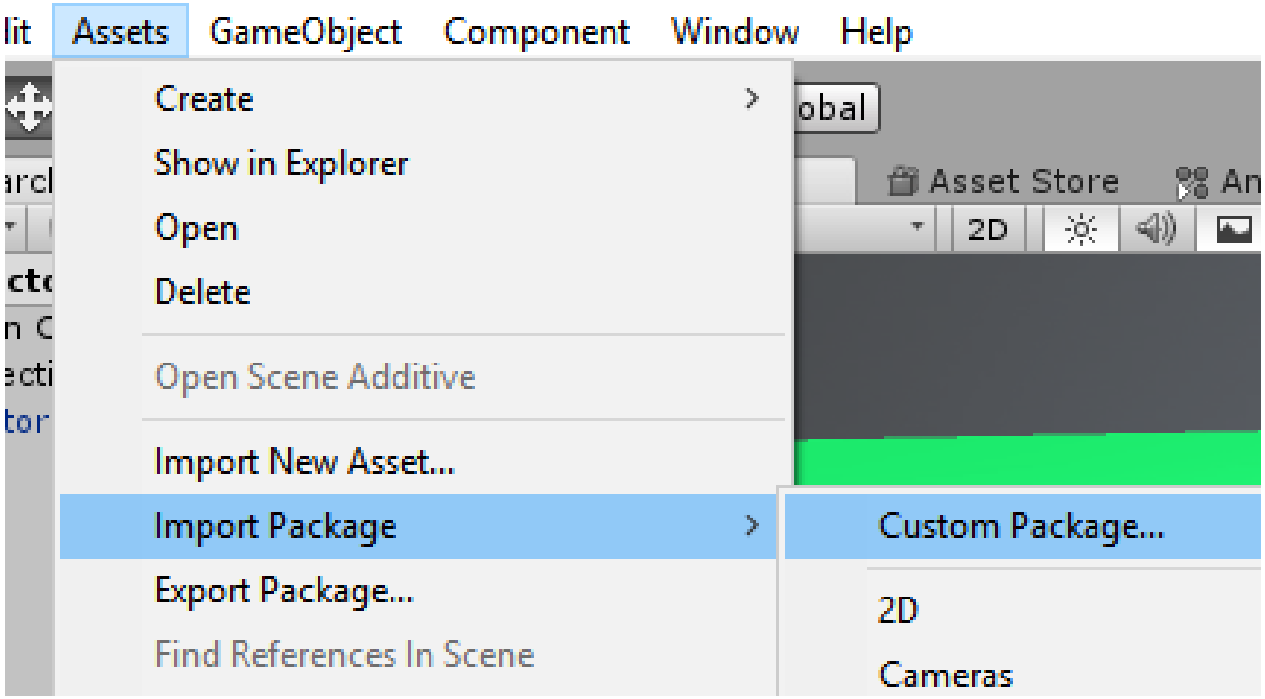


- Contextual
- Static flag is used for static object in the scene (used to bake light, evaluate navigation path, etc.)
- Tag (find objects by code)
- Layer (render, collision, etc.)
- List all components attached to the current selected object
- Lock inspector to current object (useful to copy components properties between objects)

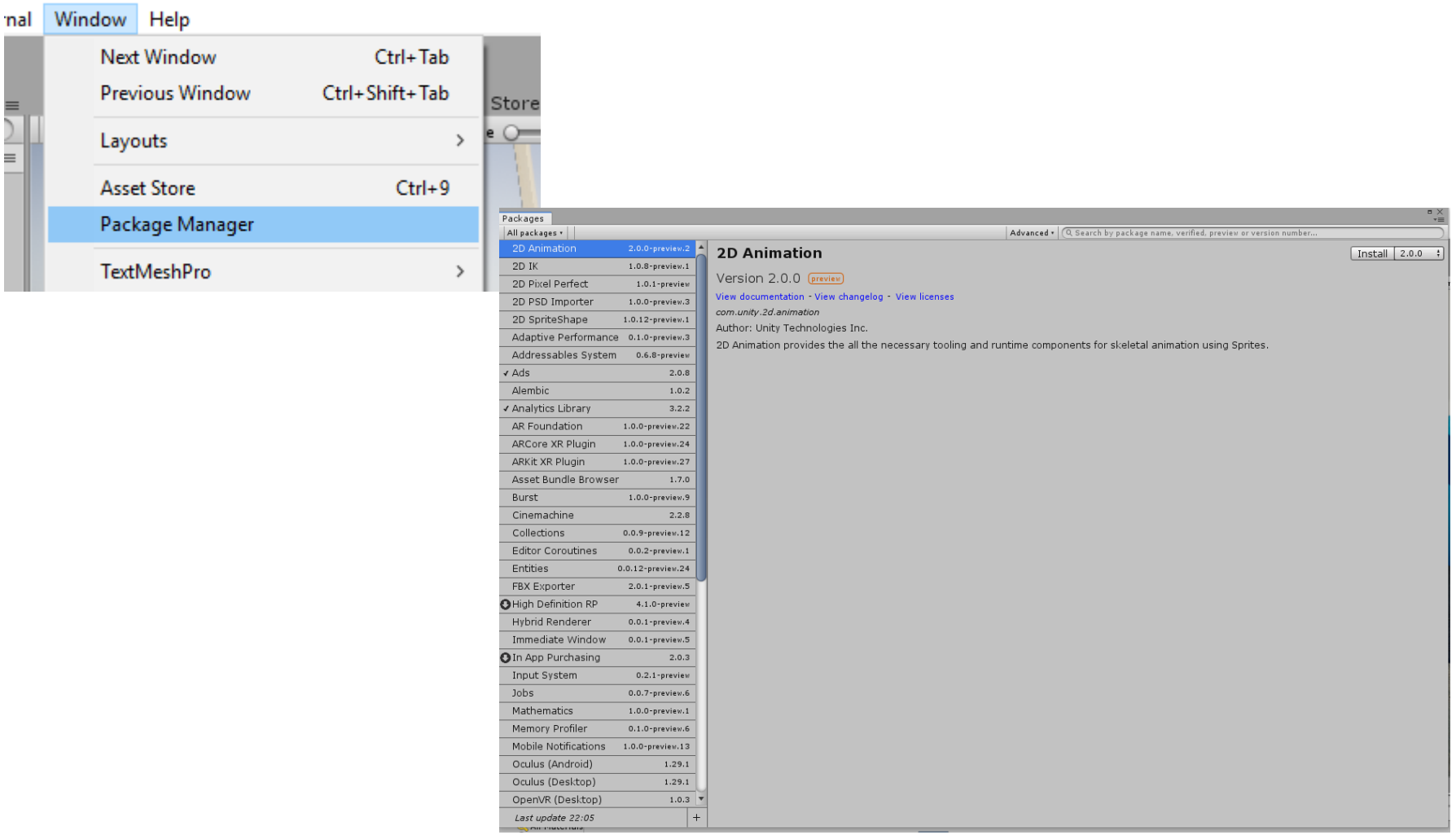
Asset store:


Board - Android <DX11 on DX9 GPU>
Window Help
Next Window Ctrl+Tab
Previous Window Ctrl+Shift+Tab
Layouts
Services Ctrl+0
Scene Ctrl+1
Game Ctrl+2
Inspector Ctrl+3
Hierarchy Ctrl+4
Project Ctrl+5
Animation Ctrl+6
Profiler Ctrl+7
Audio Mixer Ctrl+8
Asset Store Ctrl+9
Version Control
Collab History
Animator
Animator Parameter
Sprite Packer
Experimental
Holographic Emulation
Test Runner
Lighting
Occlusion Culling
Frame Debugger
Navigation
Physics Debugger
Console Ctrl+Shift+C
Asset Store
Language:English
Filters
Home
3D Models
Animation
Applications
Audio
Complete Projects
Editor Extensions
Particle Systems
Scripting
Services
Shaders
Textures & Materials
Unity Essentials
CTR COMPLETE
Extensions
Code Now!
3.98
5 Compatible
COMPLETE is the ultimate toolkit for
structured spaces in Unity. If you
stream an open world, bring a huge
to mobile, or take advantage of the
techniques in audio occlusion and
gation, SECTR COMPLETE has
thing for you. Best of all, SECTR
LETE is the cheapest way to buy the
suite with all three premium modules
price of two!
OM INDOORS TO OPEN WORLD
MPLETE
24 hour sale
NEW
Asset Kits
ESSENTIALS
Easter Edition add...
Next-Gen
Artist Essentials

Import custom packages: .unitypackage

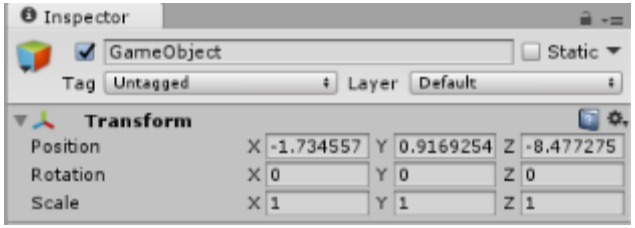


Extending Unity: Package Manager

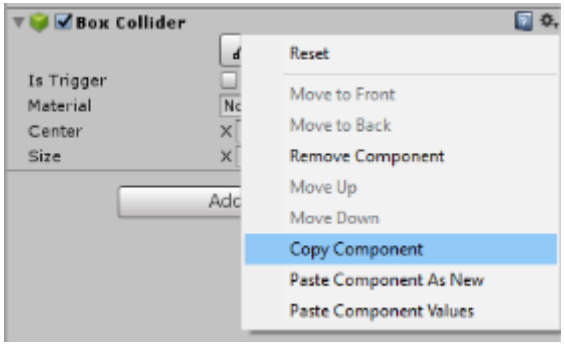


Game Object:

- Populate the Unity scene (visible in the hierarchy)
- Game object are made up by several components that could be added through menu component add, or through the add component button in the inspector
- Every game object (also an empty) has a transform component
- The game object could be activated/deactivated

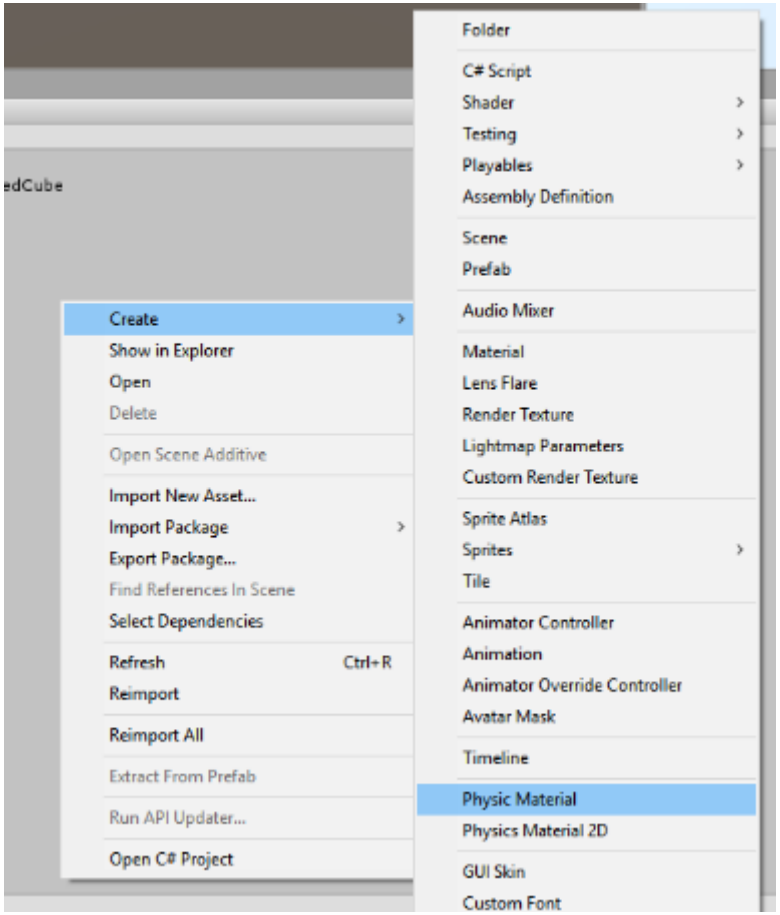


- Component could be added to the game object
- Component could also be copied and pasted to another game object



Physics:

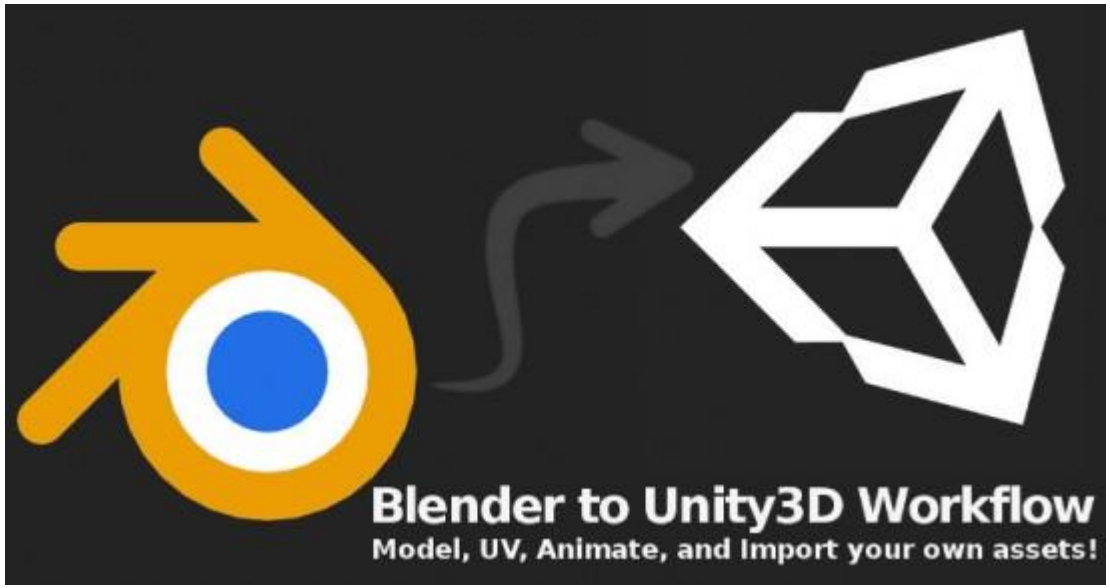
- Add a plane
- Add a cube
- The default plane and cube already come with a box collider attached
- Add a Rigid Body component (Physics)
- Run the scene
- Let’s now try adding a physic material like rubber



Import Blender models into Unity3D:

Blender models could be imported into Unity 3D:

- You can export the model from Blender using the .fbx format
- You can drag the .blend file into the Assets folder (blender required) Unity uses blender to auto export an fbx.



Scripting:

Two languages could be used as scripting language in blender:

- C#

- Scripts are mainly used to implement the game mechanics

- They could also be used to customize the editor or to create for example PCG.

- A script could be attached to a game object dragging it over the inspector or using the component add menu

Scripting:

- Two important methods:
- Start() -> called when the object is initialized
- Update() -> called once per frame

- How to move a cube?

```
Void Update()
{
        transform.Translate(new Vector3(1.0f, 0.0f, 0.0f));
}
```

Scripting:

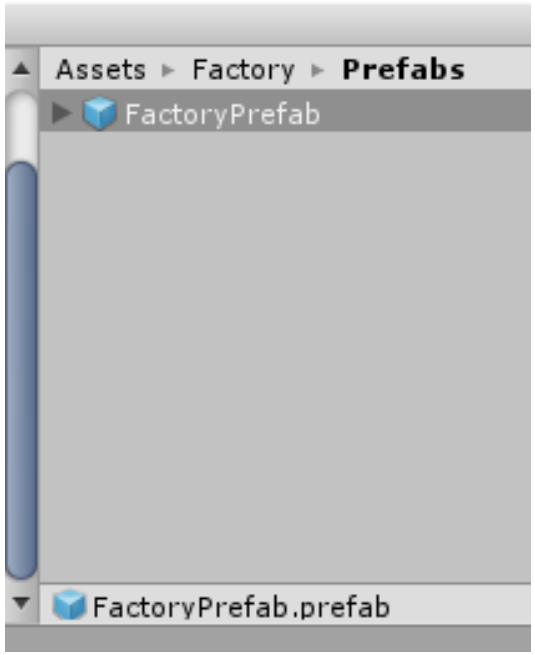
- Two important methods:
- Start() -> called when the object is

- Update() -> called once per frame
- How to move a cube, fps independent way?

```
Void Update()
{
        Debug.Log(Time.deltaTime);
        transform.Translate(new Vector3(Time.deltaTime, 0.0f, 0.0f));
}
```

Prefab:

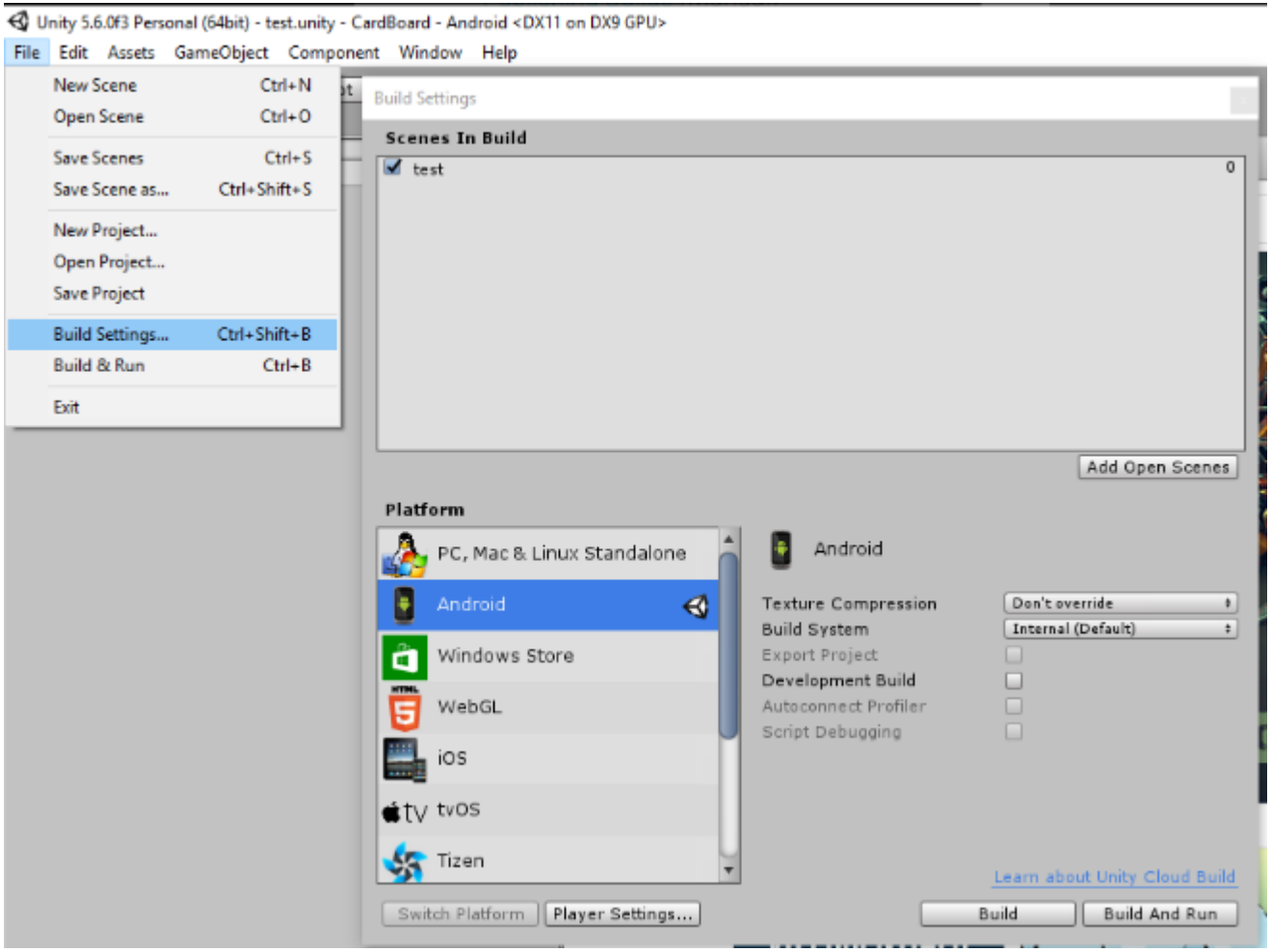
- To store a game object and its components as an asset we can create a prefab
- To create a prefab simply drag the game object from the hierarchy view to the asset folder
- The prefab could then be instantiated runtime or added again to the scene



UI: a simple score viewer:


```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class scoring : MonoBehaviour {

    public Text txt_score;
    public int score;


    // UpdateTxtScore is called when the score changes
    void UpdateTxtScore () {

        txt_score.text = score.ToString();
    }


    public void IncreaseScore()
    {
        score += 10;
        UpdateTxtScore();
    }

    public void DecreaseScore()
    {
        score -= 10;
        UpdateTxtScore();
    }
}
```

Deploy:


Unity 5.6.0f3 Personal (64bit) - test.unity - CardBoard - Android <DX11 on DX9 GPU>
File Edit Assets GameObject Component Window Help
New Scene Ctrl+N
Open Scene Ctrl+O
Save Scenes Ctrl+S
Save Scene as... Ctrl+Shift+S
New Project...
Open Project...
Save Project
Build Settings... Ctrl+Shift+B
Build & Run Ctrl+B
Exit
Build Settings
Scenes In Build
test 0
Add Open Scenes
Platform
PC, Mac & Linux Standalone
Android
Windows Store
WebGL
iOS
tvOS
Tizen
Android
Texture Compression Don't override
Build System Internal (Default)
Export Project
Development Build
Autoconnect Profiler
Script Debugging
Learn about Unity Cloud Build
Switch Platform
Player Settings...
Build
Build And Run

Leap Motion Controller

https://www.leapmotion.com

An Hand Tracking Device

- It allow us to track the position of the user hand in real-time, pretty much like kinect do with our body.
- It’s specialized in hand tracking, we can get the position of each bone of our fingers!
- It’s very fast! (up to 300 fps, consider that kinect 2 RGB camera works at 30fps).
- It’s simple (both in design and ease of use).
- It’s cheap! You can get one for 70€!

HARDWARE (1/2)

Inside the leap

- 3 Infrared (IR) leds
- 2 IR wide angle cameras

Thanks to the wide angle lenses of the camera the leap motion is able to monitor an imaginary interaction box of 60 w x 60 h x 60 d cm.

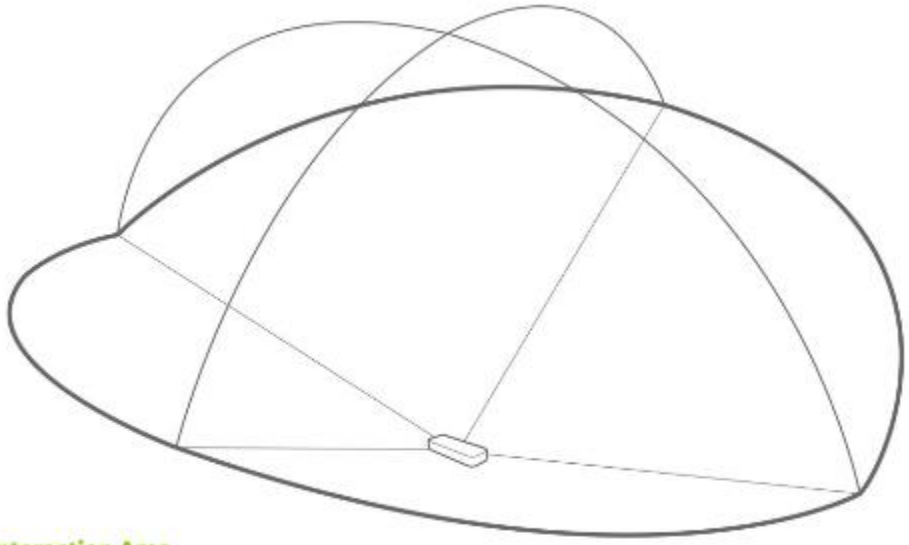
HARDWARE (2/2)

How does it work?

http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/

Leds light up the scene with IR light (outside human visible spectrum), both cameras capture the image.

The intersection of the binocular cameras field of view creates a shape of an inverted pyramid which defines our interaction area



SOFTWARE

The real magic happen here…

- Hardware just send simple raw images to Leap Motion runtime service through the USB cable
- Images are processed using CV algorithms to reconstruct 3d position of what the camera sees.
- Techniques are pretty similar to the ones used for MOCAP: we got two images, we try to detect the same point in both of them and then we triangulate to detect position.
- Software also tries to infer occluded objects position.
- Some filtering is used to guarantee data coherence between frames.

SDK + Unity Assets

https://developer.leapmotion.com/orion/

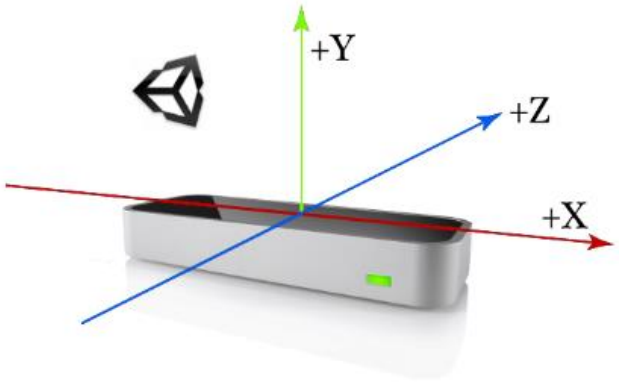
- Leap Motion provides an easy to use SDK to access controller data within your applications
- Supported languages:
  - C++
  - C#
  - Objective C
  - Java
  - Python
  - JavaScript

https://developer.leapmotion.com/documentation/

- Some useful assets for game engines are also provided, we'll have a look at them in a moment!
  - Supported game engines:
    - Unity
    - Unreal Engine

HANDS ON – LITERALLY ☺

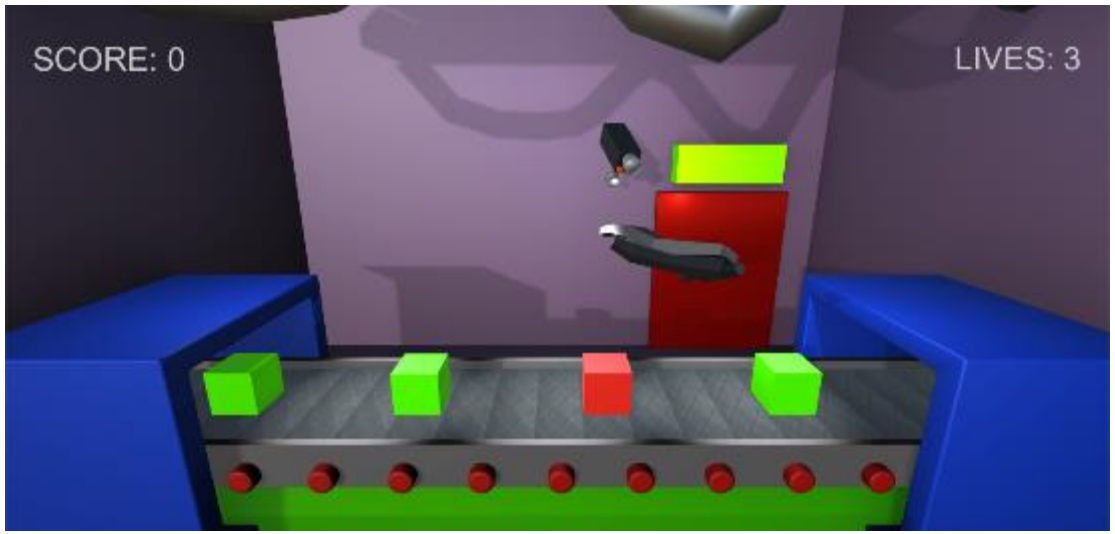
- Requirements:
  - Leap motion runtime installed
  - Unity project created, Leap .unitypackage imported
- EX#1: Let's build our first leap controller, we want to detect the position of the palm of the hand and make the security camera follow it.
- EX#2: Virtual hands visualization.
- EX#3: Interact with the environment.
- EX#4: Interact with the environment (in a real game).
- EX#5: Grasping Objects (Physics Issues).
- EX#6: Grasping Objects (Interaction Engine).



Leap Motion Unity Assets

Leap Motion SDK also contain a unity package with a lot of ready to use stuff we can integrate in our games without bothering about writing our own controllers…

- Hand Controllers
- Hand Models
- Grasping utilities
- ….



Something about the user experience:

- We should not forget that the emotional part makes a game/story fascinating
- We are going to create a little game with a very simple mechanic.
- The player impersonate an assembly line workman
- He should grab bad pieces leaving on the line the good ones.
- A surveillance camera is active…someone is watching us…

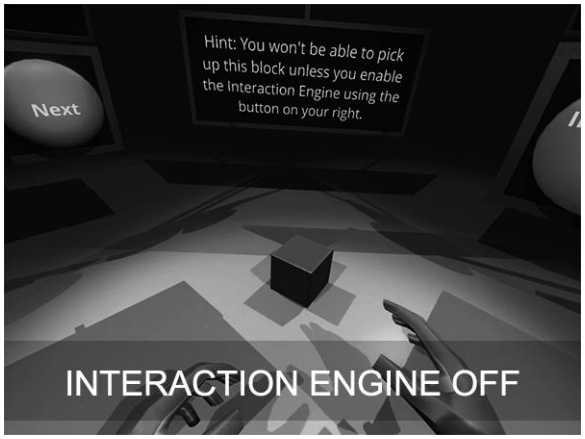
Leap motion and VR

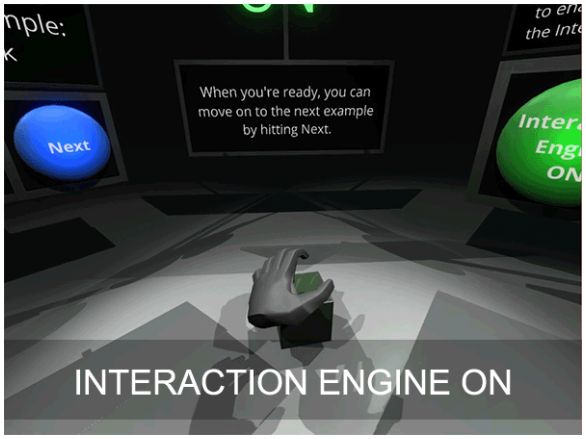
- Leap motion can also be used vertically and mounted on an Head Mounted Display (HMD) such as the Oculus Rift
- Orion SDK are thought with VR in mind.
- You can try this combination for your final project in the lab!

https://www.youtube.com/watch?v=rnlCGw-0R8g

- Interaction engine

http://blog.leapmotion.com/introducing-interaction-engine-early-access-beta/





Future of leap motion

- Android VR, embedding the leap motion in visors
- https://developer.leapmotion.com/android#107