VIA University College

Course Introduction

An introduction to game development and Unity

Course Overview

Practical course information and a road map for the semester

Introduction to Game Design

Before we start programming we need to learn what a game is

Introduction to Unity

An overview of the game engine we are going to use

Exercises

Get started with your game development adventure!

Who’s teaching?

Jakob Knop Rasmussen

MSc. in Computer Science, Aarhus University
Associate Professor at VIA

Contact

E-mail:	jknr@via.dk
Phone:	87 55 43 34
Office:	C03.01

Who are you?

Course Overview

Your Expectations

Why did you pick this course?

What do you think is going to happen?

What do you expect to learn?

Course Description

Developing games in Unity

Once a week

New topic(s) each class

Class exercises and course project

Learning by doing

No oral exam - grading based on project

Course literature

Course description available on itslearning

Semester Plan

Available on

itslearning

Road Map

Course Overview

- Course Introduction
- Basic Scripting
- Vectors & Input
- Physics
- Graphics & Audio
- Animation
- User Interface
- 2D Game Development
- Intermediate Scripting
- Game Architecture

The Course Project

You must develop a game in groups of 1-4 throughout the course

Protip: Work on your project every week!

Deadline: ~ two weeks after last session (see itslearning)

- Hand in the source code using a public **GitHub** link on itslearning
- The GitHub repository must include a README with
  - A link to a **YouTube** video demonstration of your game
  - Sources of any third party assets/code
  - A link to a playable WebGL build of your game, hosted on GitHub Pages

The Course Project

The project must include elements from the following topics:

Input & Vectors (e.g. input systems, transforms)
Physics (e.g. rigidbodies, colliders, triggers)
Graphics & Audio (e.g. models, shaders, audio clips)
Animation (e.g. animators, animations)
User Interface (e.g. menus)
Intermediate Scripting (e.g. coroutines, events, optimization)
Game Architecture (e.g. game managers, SOLID principles)

Write Your Own Code...

It’s okay to rely on tutorials, but make your own game

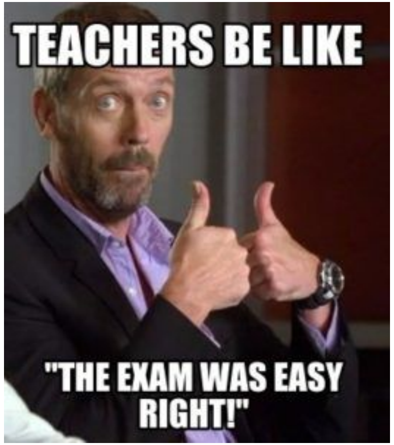
MAKE GAME THAT'S IDENTICAL TO THE TUTORIAL THEY FOLLOWED
"I'M SUCH A GOOD GAME DEVELOPER"
imgflip.com

The Exam

There is no oral examination. Your GMD grade is based on the course project.

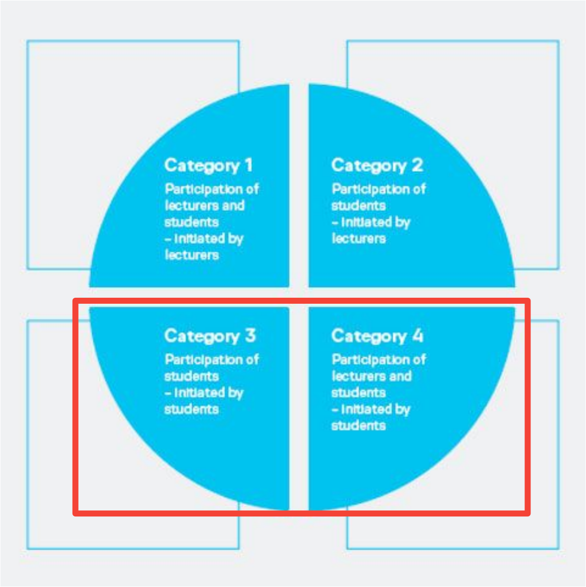
Additionally, you must hand in 1-2 pages (per group member) of individual reflections on your contribution to the project.

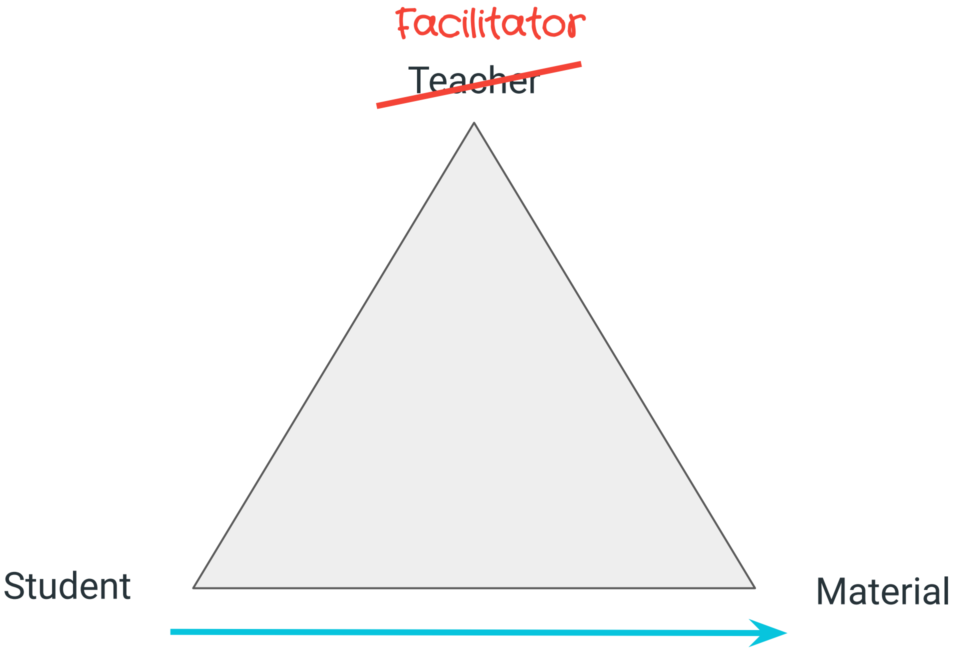
Remember to mark clearly who made what!



My Expectations

5 ECTS = ~140h of the student’s time = ~**9h/week outside of scheduled sessions**


Category 1
Participation of lecturers and students – Initiated by lecturers
Category 2
Participation of students – Initiated by lecturers
Category 3
Participation of students – Initiated by students
Category 4
Participation of lecturers and students – Initiated by students



My Expectations

Engaged students

who build cool stuff

and learn at the same time

Defining “Game”

What is a game?

What is a game?

Core elements? Rules? Win/lose conditions?

Interactive experiences?

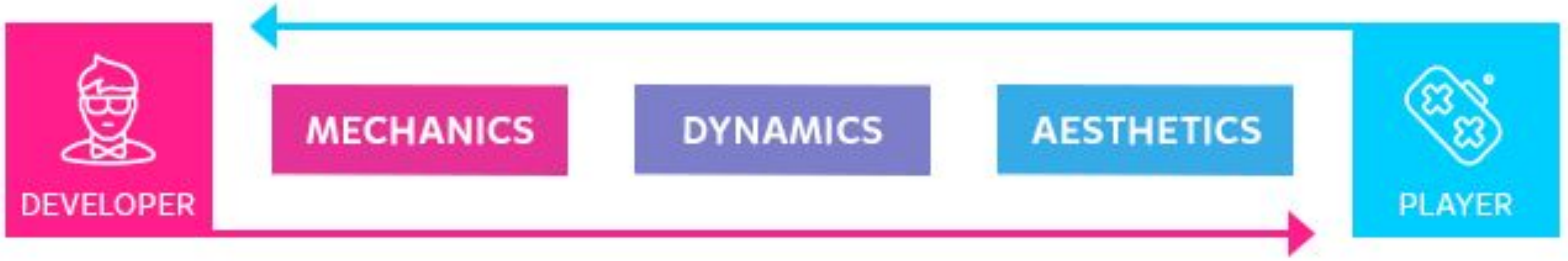
Game Genres

Can you give examples on game genres?

What games belong to each genre?

Is this a good way to categorize games?

A question about perspective



Mechanics are the base components of the game - its rules, every basic action the player can take in the game, the algorithms and data structures in the game engine etc.

Dynamics are the run-time behavior of the mechanics acting on player input and "cooperating" with other mechanics.

Aesthetics are the emotional responses evoked in the player.

A question about perspective

What makes a game fun?

The Aesthetics of Play

1. Sensation

- Game as sense-pleasure

2. Fantasy

- Game as make-believe

3. Narrative

- Game as drama

4. Challenge

- Game as obstacle course

5. Fellowship

- Game as social framework

6. Discovery

- Game as uncharted territory

7. Expression

- Game as self-discovery

8. Submission

- Game as pastime

9. Competition

- Game as domination

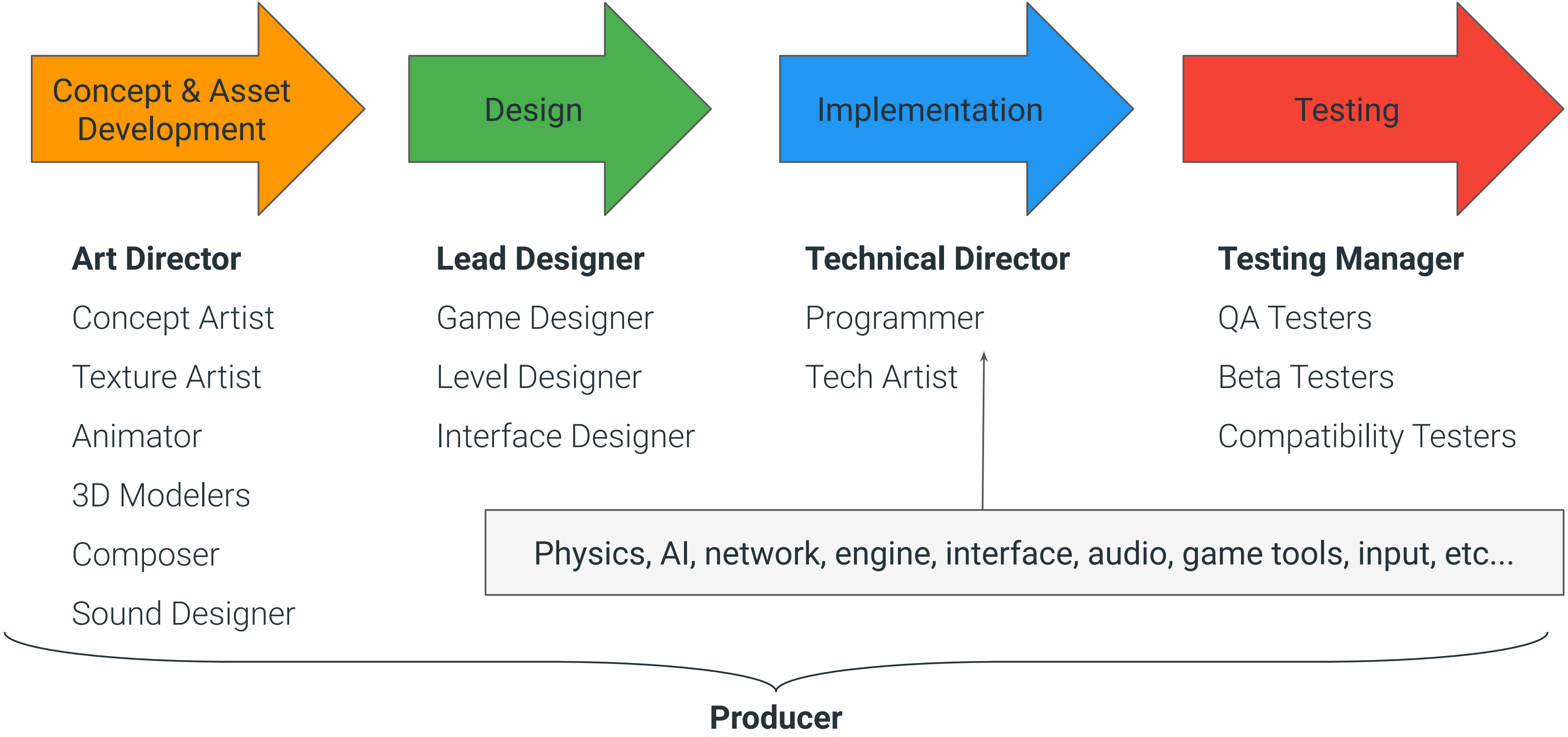
10. Mastery

- Game as self-improvement

What Does it Take to Develop a Game?

It’s hard work to make a game. It’s even harder to make a fun game!

Team Structure & Production Pipeline



Concept & Asset Development

Art Director

Concept Artist

Texture Artist

Animator

3D Modelers

Composer

Sound Designer

Design

Lead Designer

Game Designer

Level Designer

Interface Designer

Implementation

Technical Director

Programmer

Tech Artist

Physics, AI, network, engine, interface, audio, game tools, input, etc...

Testing

Testing Manager

QA Testers

Beta Testers

Compatibility Testers

Producer

Why did the engineer get fired?

“A young aerospace engineer is at his first day of work. He goes into the boss’s office with a gleam in his eye, and declares, "I’ve got the greatest idea for a new kind of airplane."

The boss is intrigued. "Explain," he says.

The young engineer takes on a visionary expression and stares into the distance. "The passengers board hassle-free in five minutes. Then the plane takes off, silently, with barely a bump, as the passengers enjoy martinis in their private booths. As they soar over the Atlantic, a young couple enjoys the view in one of the plane’s many bubble canopies, and a cute kid gets a tour of the cockpit. The captain chuckles as the kid asks why they can’t fly to the moon. By the time they touch down, love has been found, lessons have been learned, and everybody is ready for whatever awaits them at their destination."

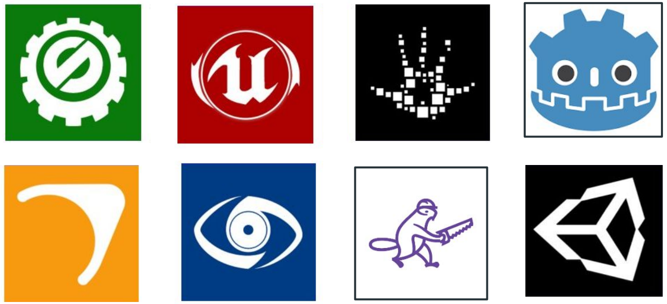
The boss leans back in his chair and takes a long drag on his cigar. "You’re fired," he says.”

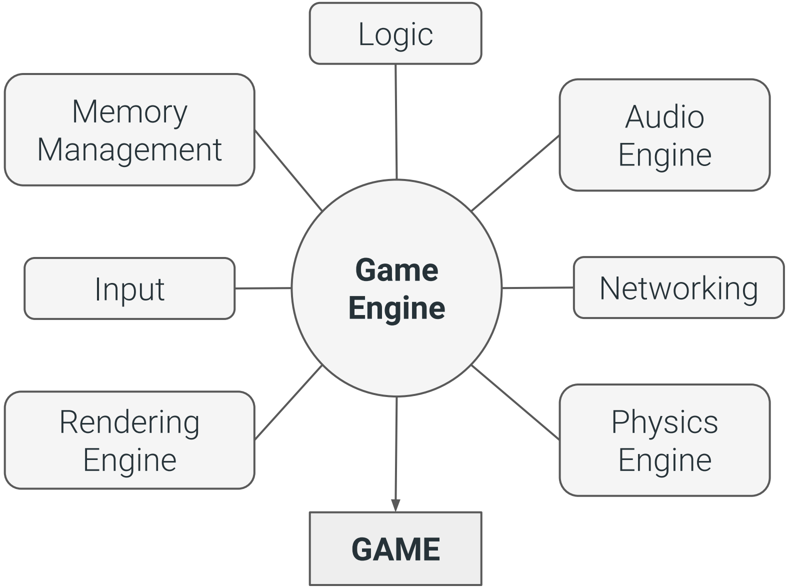
"Designing Games: A Guide to Engineering Experiences" by Tynan Sylvester

Game Engines

What is a game engine?

What engines are used in the industry right now?





Logic

Memory Management

Audio Engine

Input

Game Engine

Networking

Rendering Engine

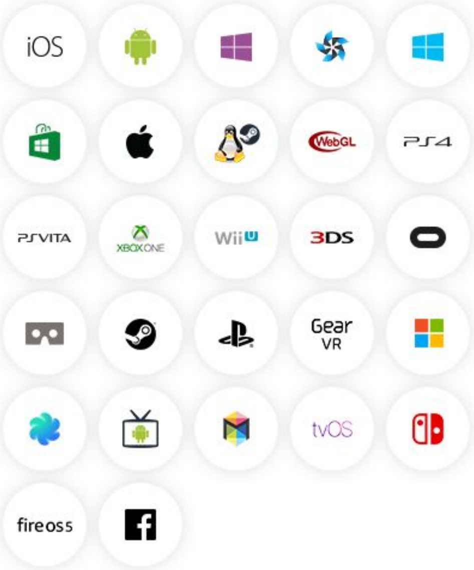
Physics Engine

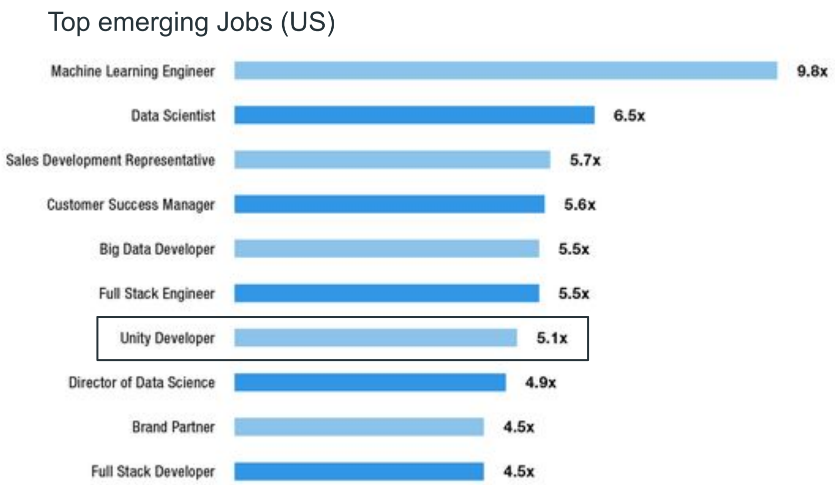
GAME

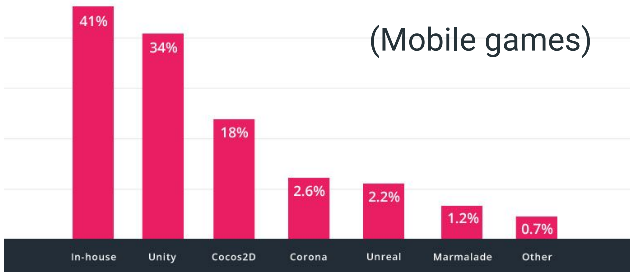
The Unity Engine

What is Unity?

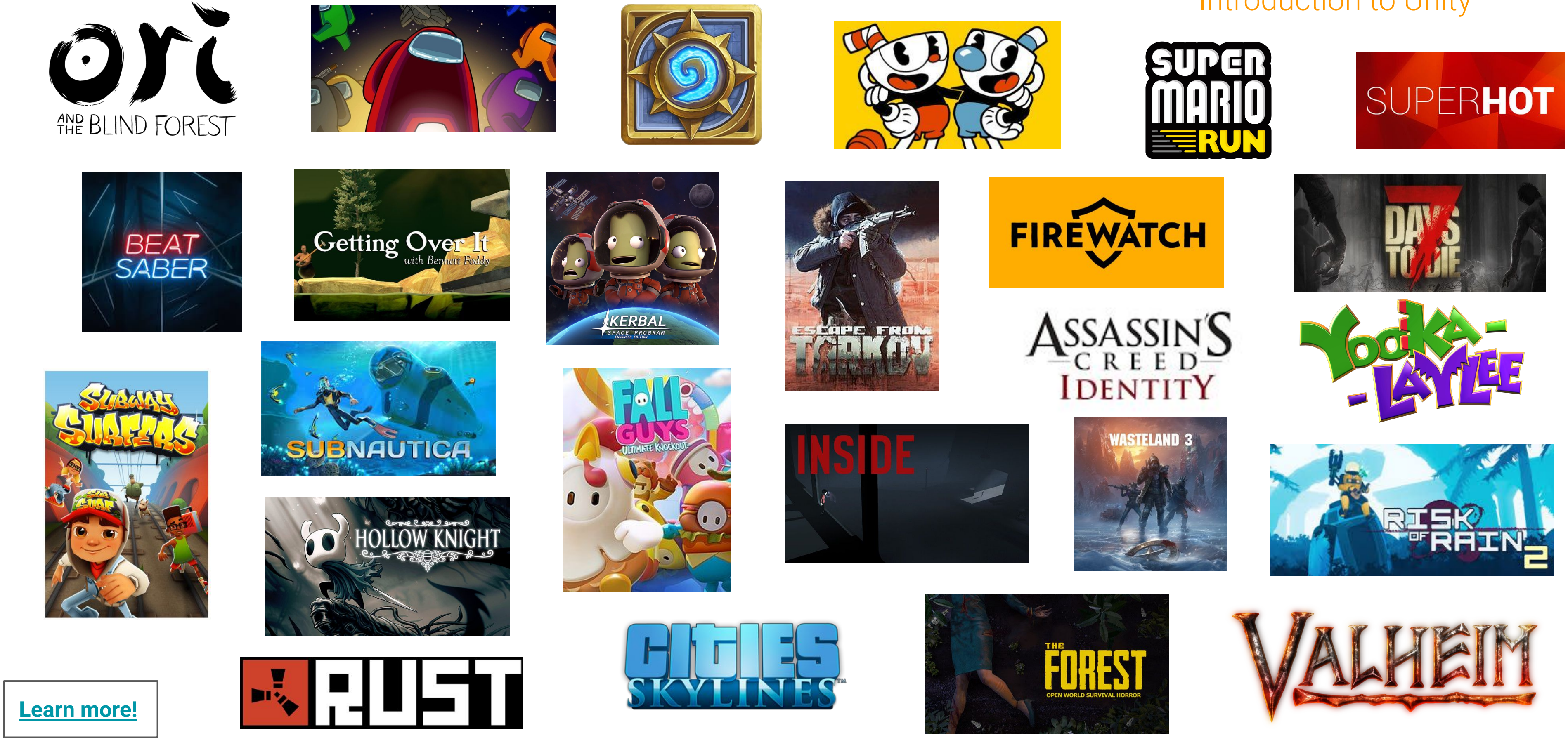
Why use it?




Top emerging Jobs (US)


(Mobile games)

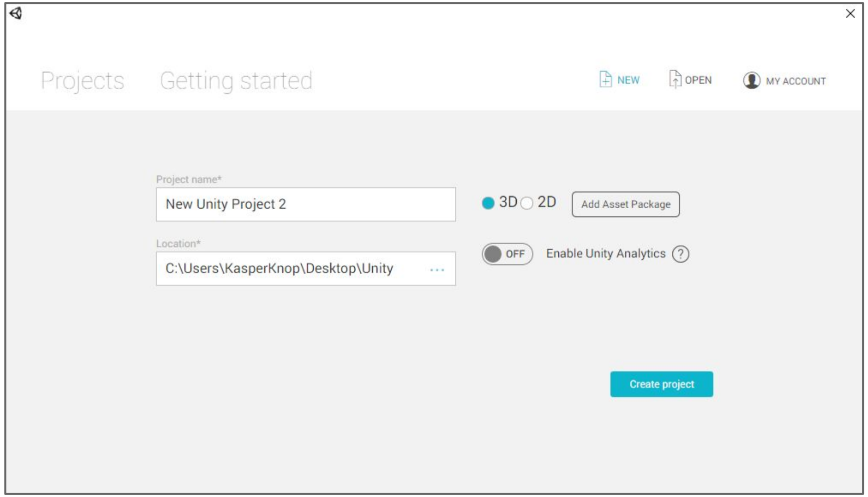
Made With Unity



Learn more!

Starting a new Project

You decide the arrangement of the views in Unity!

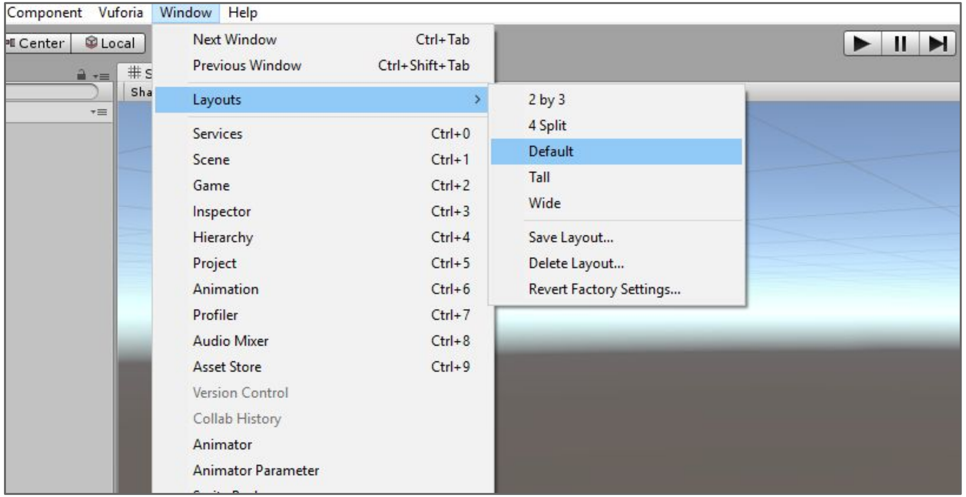


Project name*
New Unity Project 2

● 3D ○ 2D

Location*
C:\Users\KasperKnop\Desktop\Unity ...

OFF Enable Unity Analytics ?



Interface Overview

Layouts and Overview

Views

- Scene View
- Hierarchy
- Game View
- Inspector
- Project View
- Asset Store
- Console
- … you will learn even more later!

Build and Player Settings

Let me show you in Unity!

You can learn more in the documentation.

Working with the Scene View

Navigating the Scene

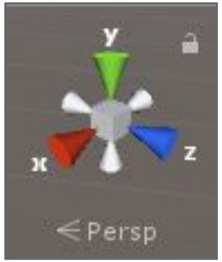
Pan: Middle mouse

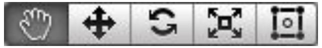
Zoom: Scroll wheel (or Right-click + Alt)

Free navigation: Right-click + WASD

Orbit: Left-click + Alt

Focus: F



Gizmos 

Changing tools - QWERT

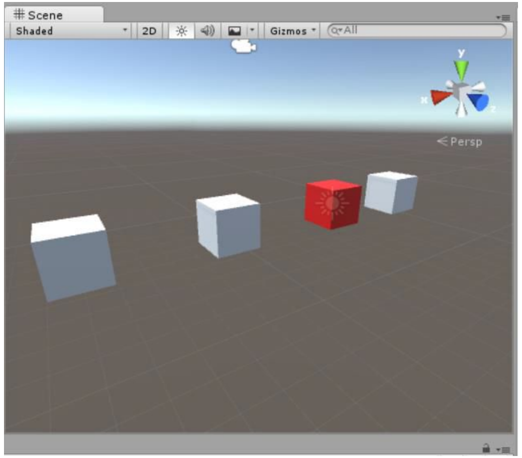
Searchable

Viewport configurations

Representation options

Perspective/isometric

Toggles



Hierarchy

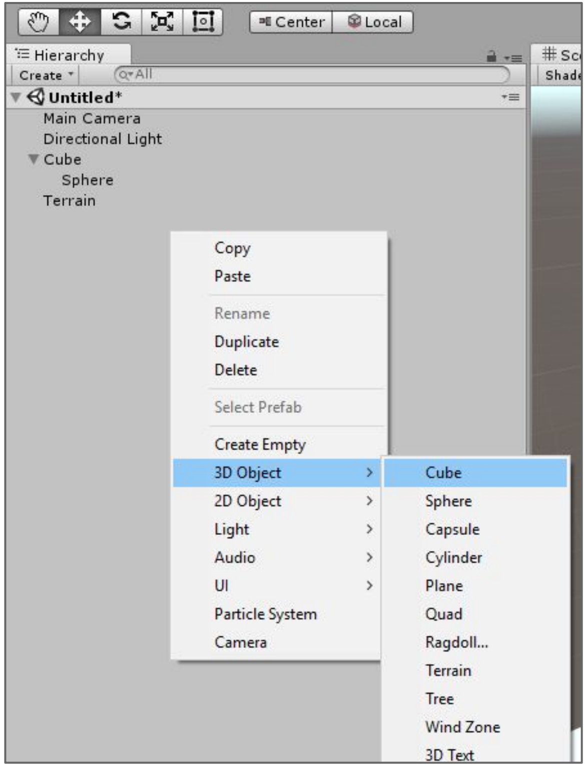
Listing all GameObjects in your Scene

Selecting in the Hierarchy also selects it in the Scene

Parenting (inheriting the transform)

Search by name and/or type

Can be used to create new GameObjects



Game View

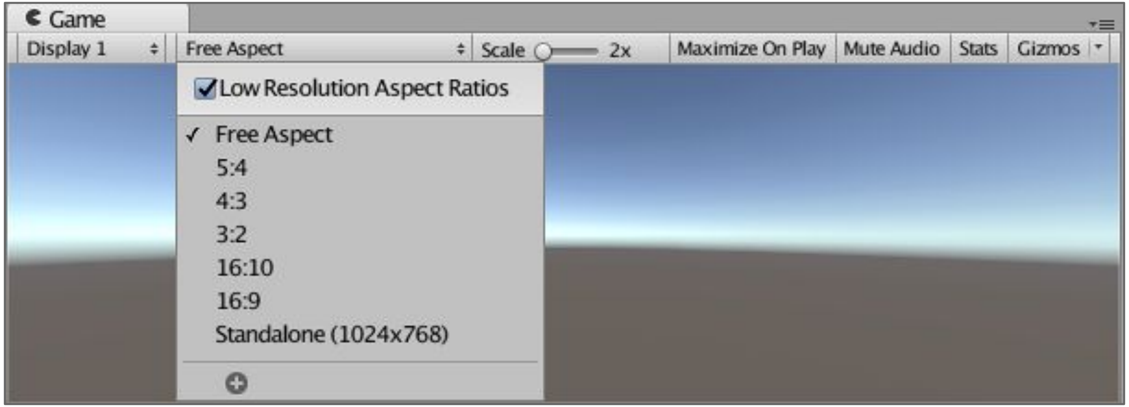
Quick testing - play your game in the editor

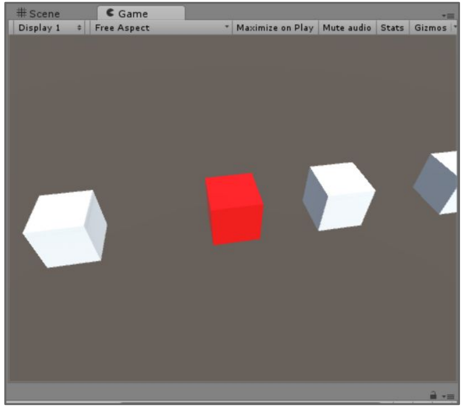
Play-, pause- and next frame button

While in play mode, **any changes you make are temporary**

Shortcut: CTRL + P (CTRL + SHIFT + P for pause)

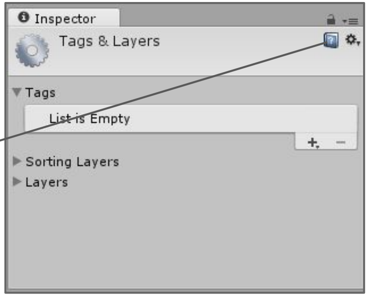
Various configurations and information on the top bar





Inspector

- Detailed information about your selection
- GameObject
  - Active/Inactive
  - Name
  - Static
  - Tags & Layers
  - Components and exposed variables
  - Asset
- Project Settings
- Locking
- Asset preview


Inspector

Tags & Layers

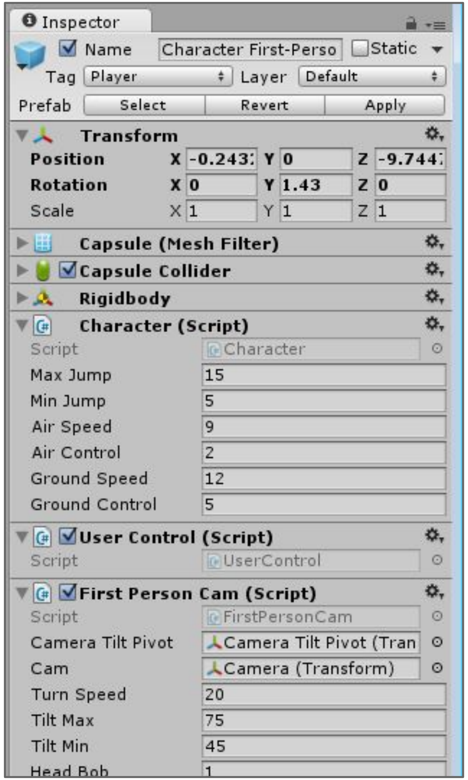
▼ Tags

List is Empty

▶ Sorting Layers

▶ Layers

Click to access documentation!


Inspector

☑ Name Character First-Perso □ Static

Tag Player Layer Default

Prefab Select Revert Apply

▼ Transform

Position X -0.243: Y 0 Z -9.744:

Rotation X 0 Y 1.43 Z 0

Scale X 1 Y 1 Z 1

▶ Capsule (Mesh Filter)

▶ ☑ Capsule Collider

▶ Rigidbody

▼ Character (Script)

Script Character

Max Jump 15

Min Jump 5

Air Speed 9

Air Control 2

Ground Speed 12

Ground Control 5

▼ ☑ User Control (Script)

Script UserControl

▼ ☑ First Person Cam (Script)

Script FirstPersonCam

Camera Tilt Pivot Camera Tilt Pivot (Tran

Cam Camera (Transform)

Turn Speed 20

Tilt Max 75

Tilt Min 45

Head Bob 1

Project View

Project View = Assets Folder

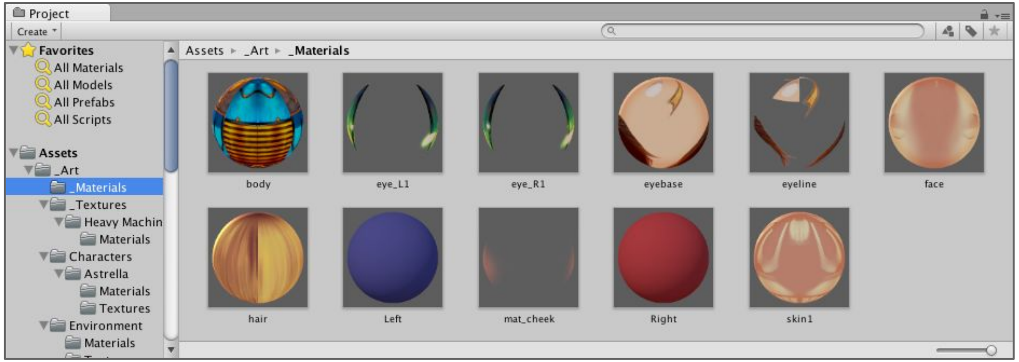
Labeling

Searching

Favorites

Importing

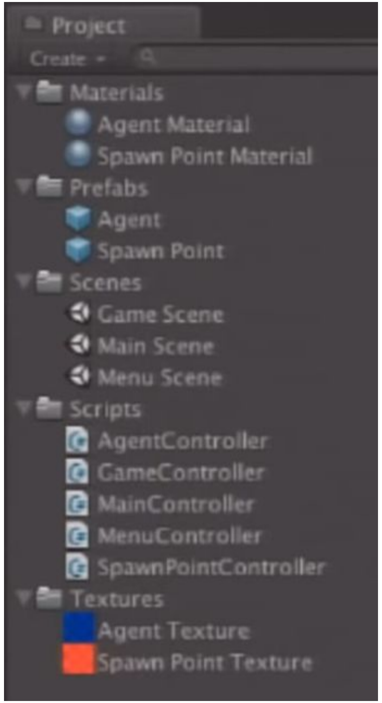
Creating new assets



Learn more!

Remember Structure

YES! :D



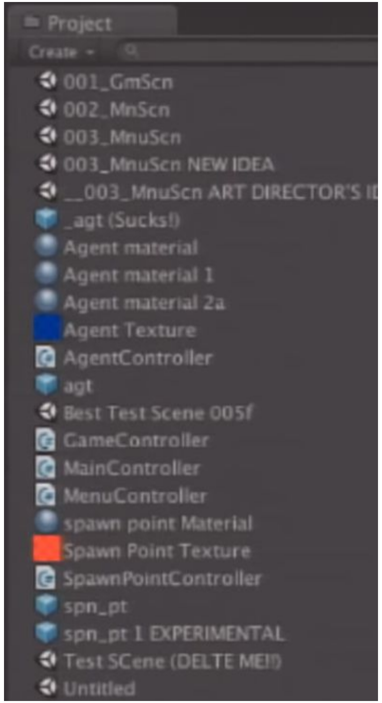
Project

Create ▾

- Materials
  - Agent Material
  - Spawn Point Material
- Prefabs
  - Agent
  - Spawn Point
- Scenes
  - Game Scene
  - Main Scene
  - Menu Scene
- Scripts
  - AgentController
  - GameController
  - MainController
  - MenuController
  - SpawnPointController
- Textures
  - Agent Texture
  - Spawn Point Texture

VS

NO!! >:(



Project

Create ▾

- 001_GmScn
- 002_MnScn
- 003_MnuScn
- 003_MnuScn NEW IDEA
- __003_MnuScn ART DIRECTOR'S I
- _agt (Sucks!)
- Agent material
- Agent material 1
- Agent material 2a
- Agent Texture
- AgentController
- agt
- Best Test Scene 005f
- GameController
- MainController
- MenuController
- spawn point Material
- Spawn Point Texture
- SpawnPointController
- spn_pt
- spn_pt 1 EXPERIMENTAL
- Test SCene (DELTE ME!!)
- Untitled

Use descriptive names, spaces, capitalization and a folder structure

Asset Store

Free and paid assets

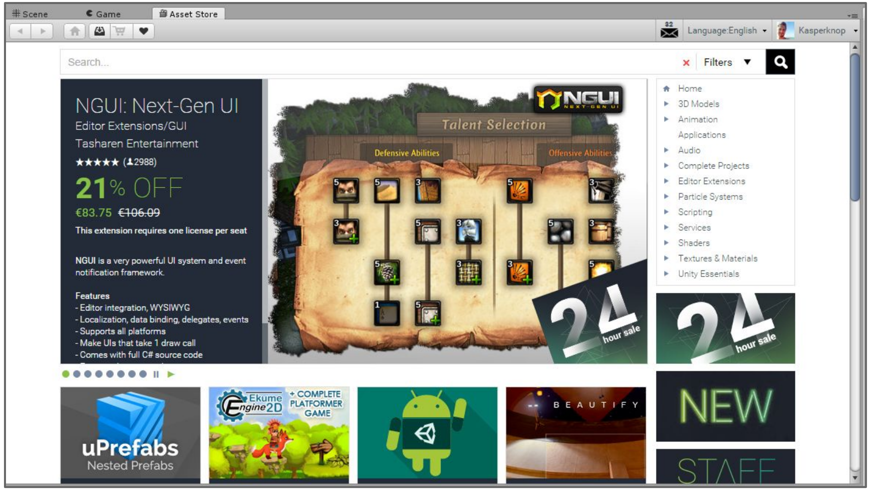
Scripts, 3D models, editor extensions, etc…

Good to get up and running fast

Integrated download

Often licensed per-team

Free to distribute in games



NGUI: Next-Gen UI
Editor Extensions/GUI
Tasharen Entertainment
★★★★★ (2988)
21% OFF
€83.75 ~~€106.09~~

This extension requires one license per seat

NGUI is a very powerful UI system and event notification framework.

Features
- Editor integration, WYSIWYG
- Localization, data binding, delegates, events
- Supports all platforms
- Make UIs that take 1 draw call
- Comes with full C# source code

Learn more!

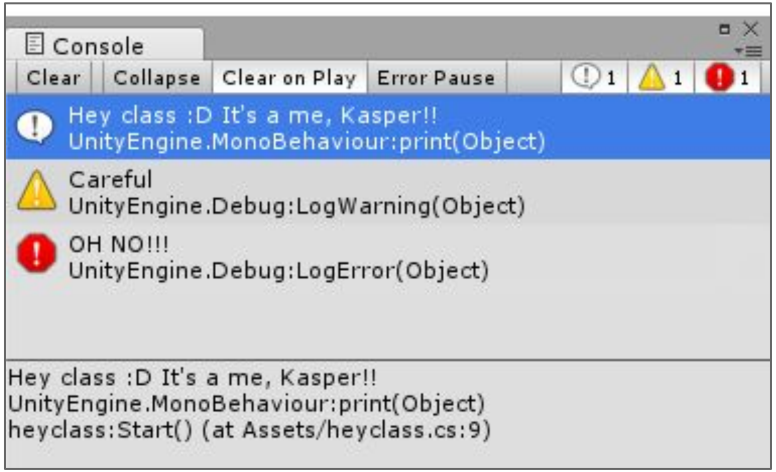
Console

Window displaying information, warnings and errors

Mainly used for fast debugging and displaying runtime/compiler errors

Left-click a message to get additional information

Double-click a message to go to the place it was invoked in the code



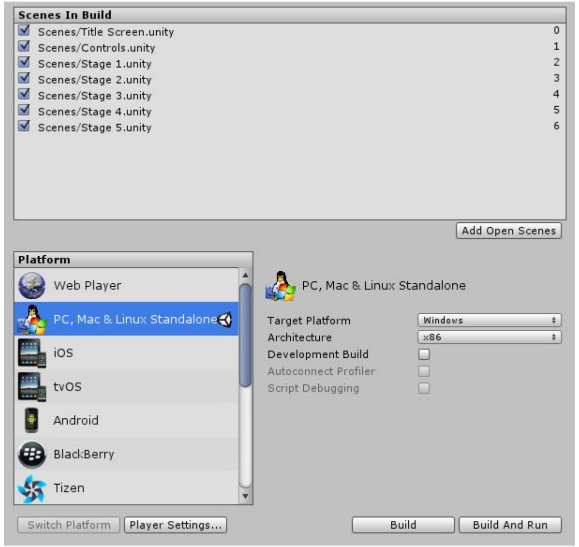
```
Hey class :D It's a me, Kasper!!
UnityEngine.MonoBehaviour:print(Object)
Careful
UnityEngine.Debug:LogWarning(Object)
OH NO!!!
UnityEngine.Debug:LogError(Object)
```

```
Hey class :D It's a me, Kasper!!
UnityEngine.MonoBehaviour:print(Object)
heyclass:Start() (at Assets/heyclass.cs:9)
```

Build and Player Settings

A build is an exported executable form of your project.

Since Unity supports multiple platforms, we have to specify what platform we want as our build target, and provide the necessary details in the player settings.



Scenes In Build

☑ Scenes/Title Screen.unity 0
☑ Scenes/Controls.unity 1
☑ Scenes/Stage 1.unity 2
☑ Scenes/Stage 2.unity 3
☑ Scenes/Stage 3.unity 4
☑ Scenes/Stage 4.unity 5
☑ Scenes/Stage 5.unity 6

PC, Mac & Linux Standalone

Target Platform: Windows
Architecture: x86
Development Build ☐
Autoconnect Profiler ☐
Script Debugging ☐

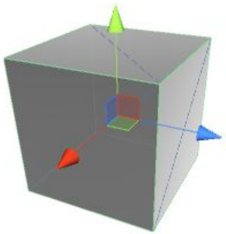
Learn more!

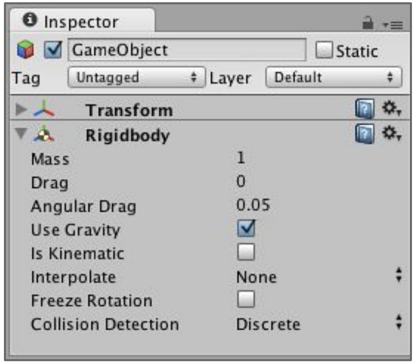
GameObjects

Live in Scenes

GameObject categories:

- 3D Object
- 2D Object
- Effects
- Camera
- Volumes
- Light
- Audio
- Video
- UI
- Empty GameObjects





Inspector

☑ GameObject □ Static

Tag Untagged Layer Default

Transform

Rigidbody

Mass	1
Drag	0
Angular Drag	0.05
Use Gravity	☑
Is Kinematic	□
Interpolate	None
Freeze Rotation	□
Collision Detection	Discrete

All GameObjects are essentially Empty GameObjects with components attached!

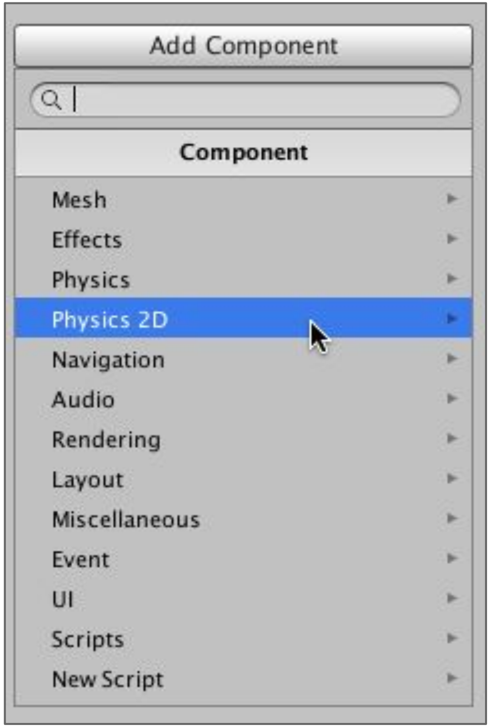
Components

Live on GameObjects

Components = Attaching behaviour to your GameObjects

Common examples

- Transform
- Collider
- Mesh Renderer
- Mesh Filter
- Camera
- Audio Source, Audio Listener
- Particle System
- **Custom scripts**



How to add components?

Assets

Live in your project folder

C# Script

Scene

Material

Sprite

Prefab

Animation

Texture

3D model

Audio Clip

… A LOT more! (you will learn some of them throughout the semester :))

You will be importing many of these assets so that we can focus on writing code!

Assets - Prefabs

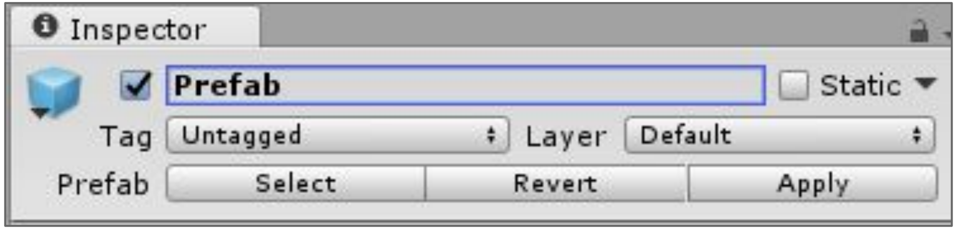
If a GameObject is set up exactly how we want it, we can save that GameObject and make copies of it

We call the saved file a Prefab (because it is prefabricated)

You can have multiple instances of the same Prefab in a Scene

They can be instantiated or cloned, to create instances of them at runtime

Prefabs have all manners of usage: Rockets, enemies, procedural levels, etc…



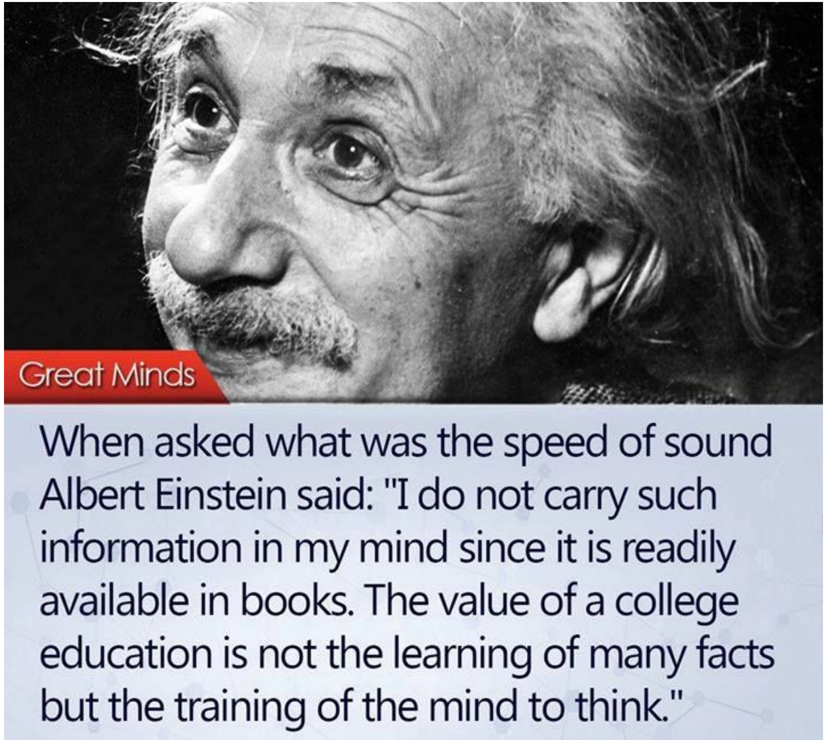
Inspector

☑ Prefab ☐ Static

Tag Untagged Layer Default

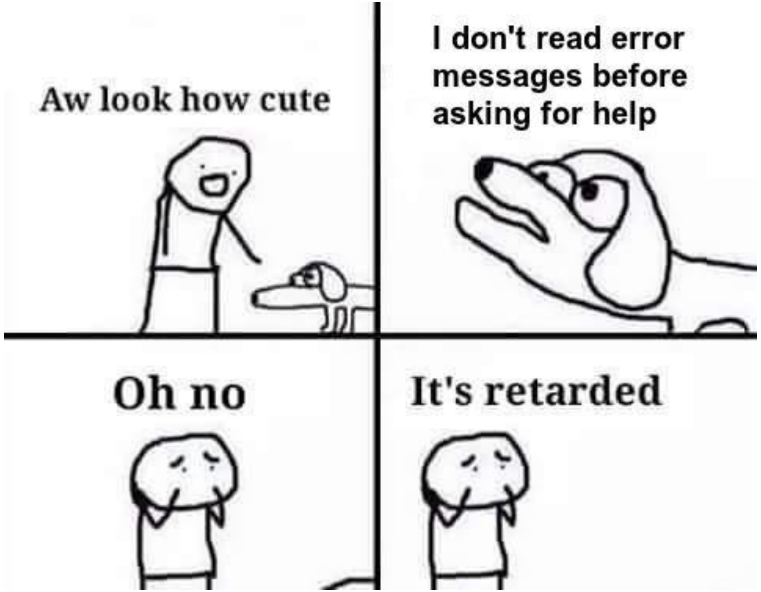
Learn more!

Exercise Tips



Great Minds

When asked what was the speed of sound Albert Einstein said: "I do not carry such information in my mind since it is readily available in books. The value of a college education is not the learning of many facts but the training of the mind to think."



Exercises

Set up your development environment

- Install Unity **(Personal Edition)**
- Install Rider (or VSCode/Visual Studio)

Get started on the exercises