Playground [beta]

# Welcome

Whether you're an aspiring maker or an educator in game development: **welcome to Unity!**
Unity Playground is a framework to create 2D physics-based games, and it's perfect for teaching to quickly enable beginner game developers to make games in Unity.

This project is intended to be as flexible as possible, not enforcing a specific game genre apart from being 2D and physics-based. It contains scripts that perform "atomic" tasks (that is, they do mostly only one thing) so you can combine them to create any type of gameplay.



This said, before using the Playground you should already have an idea of Unity's core concepts. A very good resource is the Interactive Tutorials that you can find in the editor launch window (under the Learn tab).
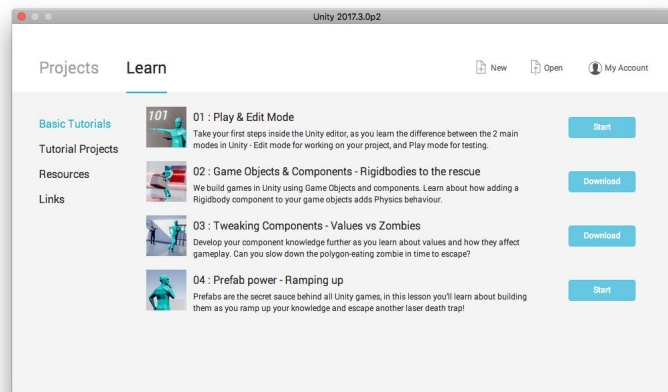


Enjoy using the Unity Playground!

## Quick Start

To use the project in a workshop, you just need to distribute the /Assets and /ProjectSettings folders to the students.

The /Assets folder contains a readme file called Readme.rtf that explains what each scripts does in a one-liner. This file is intended as a quick-reference, to give the students a quick outlook of the possibilities at their disposal. Read more about scripting on the Scripts page.

The /Sprites and /Particles folders contain graphic assets which could be used to compose the scene, but the developers are free to import new graphics if they so desire.

Scripts are in the /Scripts folder, organised by category. Most of them should work out of the box, although some require objects to be tagged in a specific way to work. For instance, /Attributes/HealthSystemAttribute.cs requires the object to be tagged as "Player" in order to show the health in the UI (read more about them on the Tags page).

## Contributing to the Playground

For comments, questions, suggestions, feel free to email Ciro Continisio. If you want to contribute to the project, check it out on Github, fork it, and create pull requests.

# Table of Contents

# General Concepts

## Tags

Tags are used by some scripts to filter objects and decide when to produce their effects. Some scripts filter objects `OnCollisionEnter2D` or `OnTriggerEnter2D`, while others (like HealthSystemAttribute) behave differently depending if the object is tagged as Player or not.

If you import the `ProjectSettings` folder at the start, some extra tags are already defined. Specifically:

- Player and Player2 have to be used on the two players to enable the UI to work properly, damaging of players, scoring points, etc.
- Enemy, though not currently used in any script, is useful to define enemies on which to apply the effects of bullets and weapons.
- Bullet for projectiles.
- Ground used for checking what is ground when the player jumps.

You can filter Tags on Conditions, and they update automatically to show the whole list of Tags so there's no possibility of making typos.

## User Interface

...

## Graphics

...

# Scripts

## Movement

Auto Move

AAA

Auto Rotate

AAA

## Creation

AAA

## Destruction

Aaa

## Attributes

Aaa

## Conditions

Aaa

## Actions

aaa