

Unity Project Guide

Philip Stenmark

April 12, 2019

1 Structure

The project consists of various folders for resources, prefabs and scenes and other editor assets. Make sure to place your assets in the correct folder and subfolder upon creation. If applicable, you may also create a subfolder for your content in any of the provided folders. The folders that are configured by default are:

- **Prefabs** contains all completed prefab assets, such environmental objects, players and pickups.
- **Scenes** holds all live scenes, such as menus and levels. Be careful modifying these, since they do not merge correctly in version control.
- **Scenes/Experimental** is where all experimental and not final scenes may be placed. You may favorite this folder in Unity for easier access.
- **Resources** contains all *in-game* resources, such as scripts, models, textures and shaders. Note that the Resources folder is a Unity special folder, meaning any resource may be loaded using **Resources.Load** in scripts.
- **Gizmos** is a Unity built-in folder that allows access for Scene View graphics through **Gizmos.DrawIcon**.
- **Editor Default Resources** contains all *in-editor* resources that can be loaded directly from Editor scripts using **EditorGUIUtility.Load**.
- **Plugins** contains all external addons. You may not modify any content of the Plugins folder explicitly.
- **StreamingAssets** contains resources that requires streaming from the filesystem, such as videos and sounds.

2 Tags

The project only provided with the Unity built-in tags. Additional tags may be added during development using the tag manager. Out of all built-in tags, make sure you are aware of some of their usages:

- **EditorOnly** objects will not appear when building and running the project.
- **MainCamera** objects will be accessed globally in code using `Camera.main`.
- **Player** may be useful for accessing all players in the scene.

3 Layers

Similar to tags, all layers provided in the project are configured by Unity default settings. Some of these are:

- **IgnoreRaycast** objects will be ignored by any raycast transmitted through the physics engine.
- **TransparentFX** may be applied on transparent objects, thus making them appear correctly when rendered using some post-processing effects, such as lens flare.
- **UI** should be applied on in-game user interface elements.

4 Inputs

Unity uses virtual inputs to allow for seamless implementation of bindable keyboard, joystick and gamepad inputs. By project default, some pre-configured inputs are created according to the functional specification. Make sure you implement your input handling according to the conventional inputs provided.

- **Horizontal** used for horizontal movement (A/D).
- **Vertical** used for vertical movement (W/S).
- **Fire** used for firing weapons (left mouse button).
- **Use** may be used for interacting with in-level objects, such as doors, players and pickups (E).
- **Reload** used for reloading weapons (R).
- **Mouse X/Y** is to be used for the player looking around and interacting with user interface elements.