CMPS 327: Introduction to Video Game Design and Development

Dr. Arun K. Kulshreshth

Fall 2020

Lecture 3: Introduction to Unity 3D

Announcement

- Project 1 is available on Moodle
  - Due: August 31, 2020 11:00 PM
  - Simple hello world project
  - Based on
    - todays lecture on Unity
    - GitHub lecture in the next class.

Unity 3D

- Integrated development environment (IDE)
  - Specialized for game development
  - Drag and drop visual interface
- Features
  - Game engine
  - Physics engine
  - Sound Engine
  - Animations
  - ….
- Create once and run anywhere
  - Mac, PC, smart phones, game consoles

Unity 3D Concepts

- Assets - building blocks of all Unity projects - graphics (textures), models, sound files. The files you use to create the scenario are stored in a folder called Assets

- Scenes - scenes are individual levels, areas of game content. Scenes can be loaded on demand.

- Game Objects - assets used in the scene become GameObjects. All GameObjects have at least one component - the Transform component.

Unity 3D Concepts

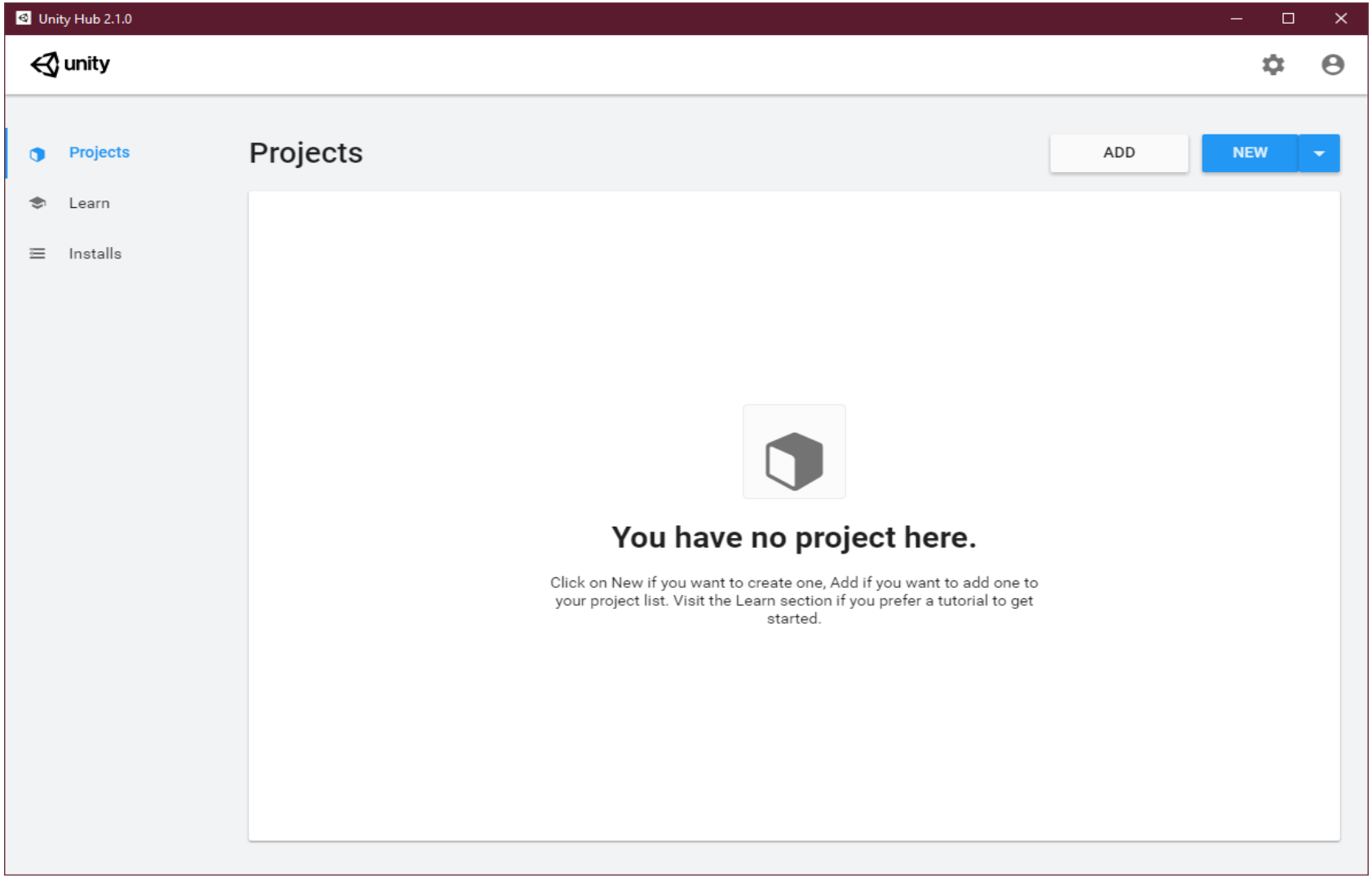
- Components - come in various forms.
  - Attach components to a game object
  - E.g. a physics component, or a script component

- Scripts - components used to add, extend or modify behavior of game objects.
  - Unity uses a Behavior class to facilitate the use of custom behaviors.

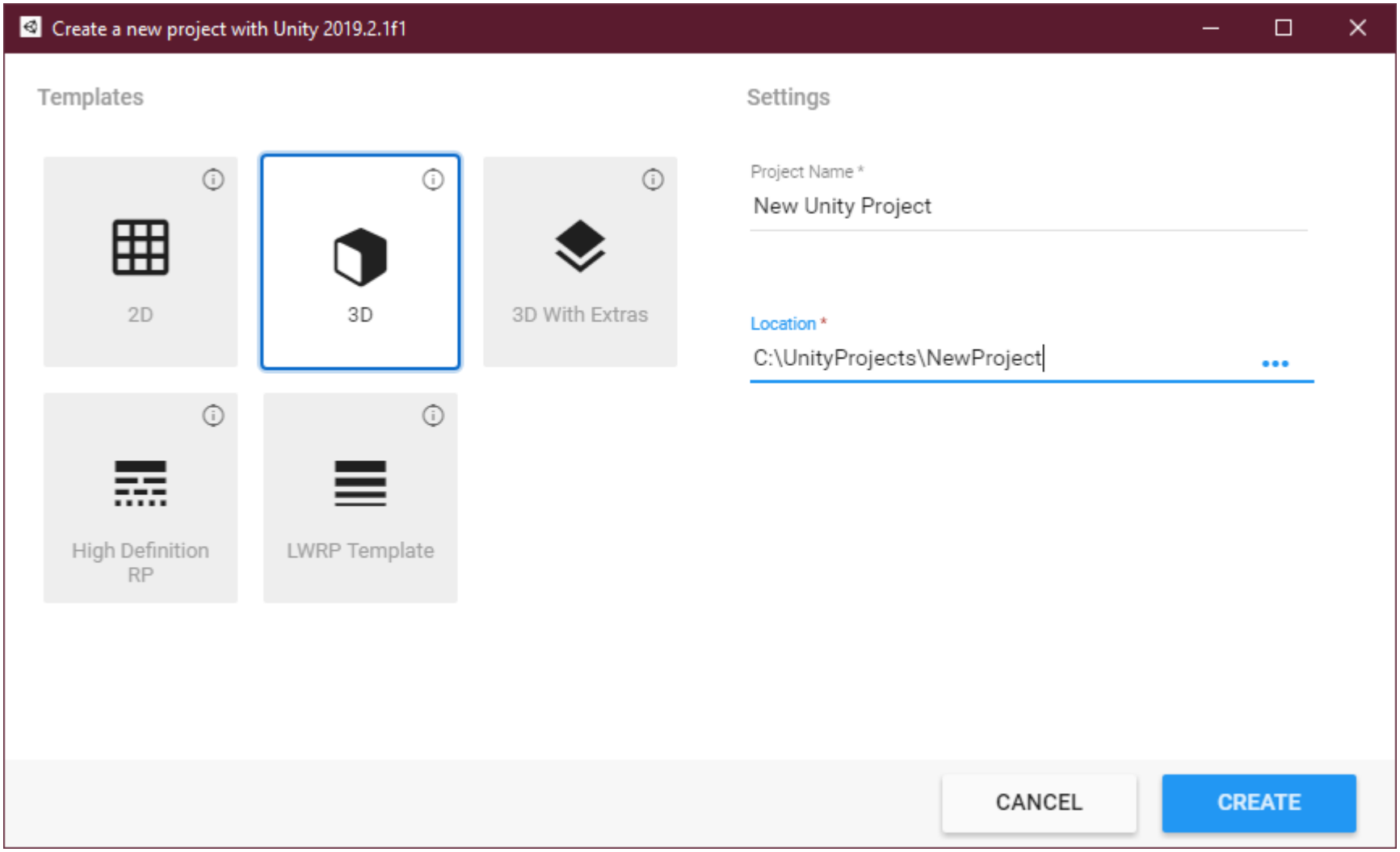
Unity 3D Concepts

- Prefabs - prefabricated game objects with stored associated components and configuration.
  - Prefabs allow functional game objects to be reused in scenes (spawned) or imported into other projects as external assets.
  - 'The First-Person Controller' is an example of a Prefab

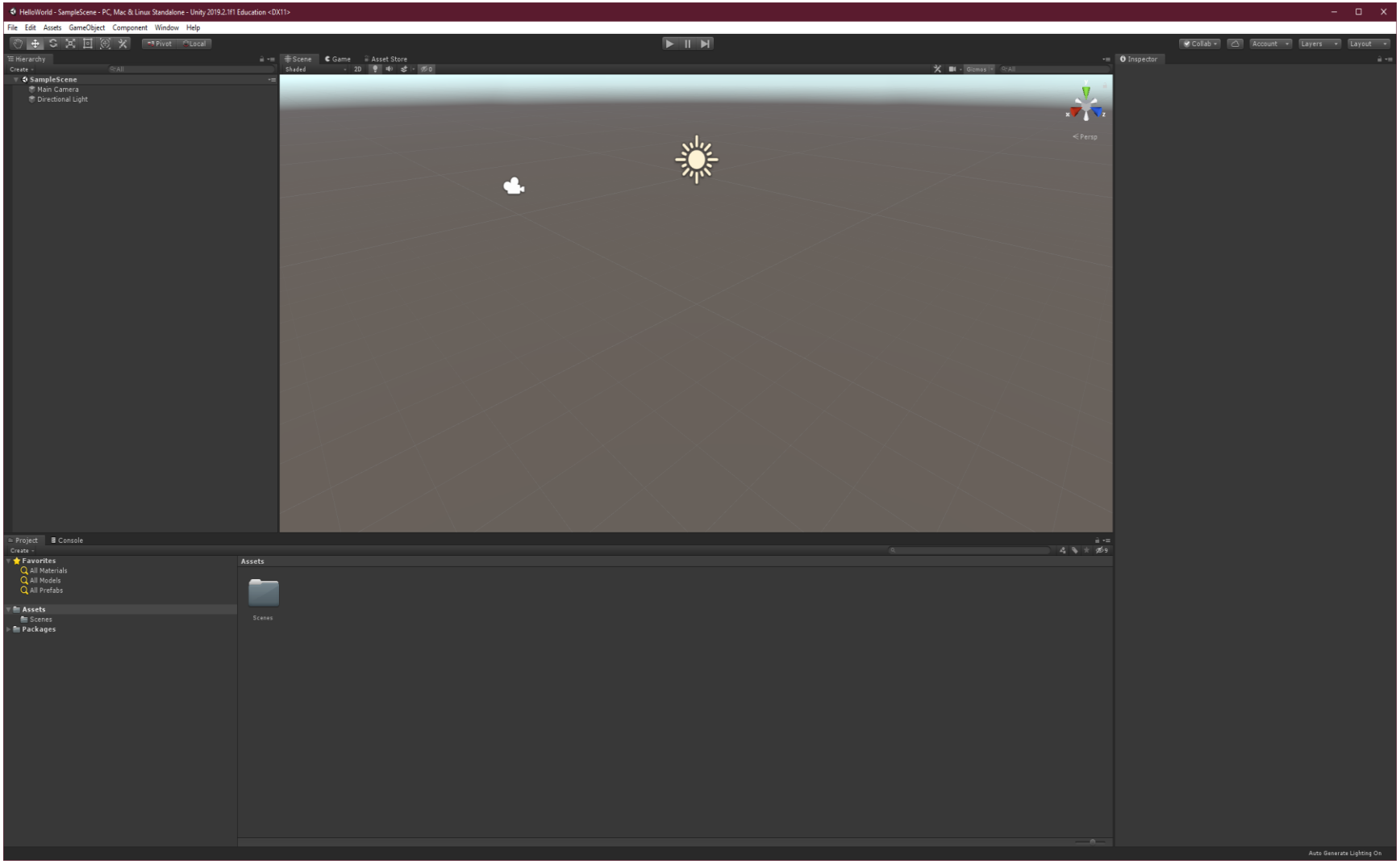
The Project Tab



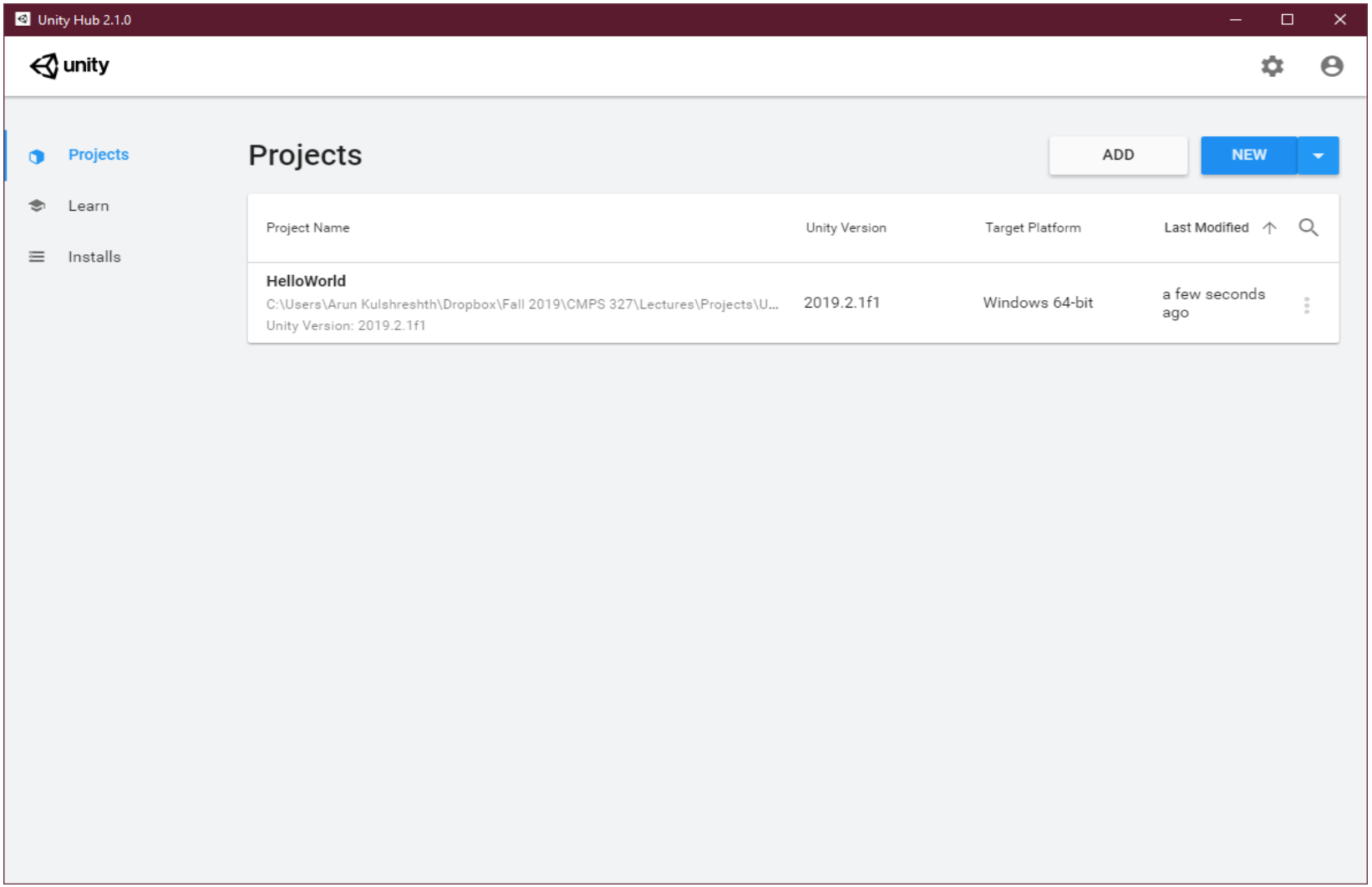
Creating a New Project



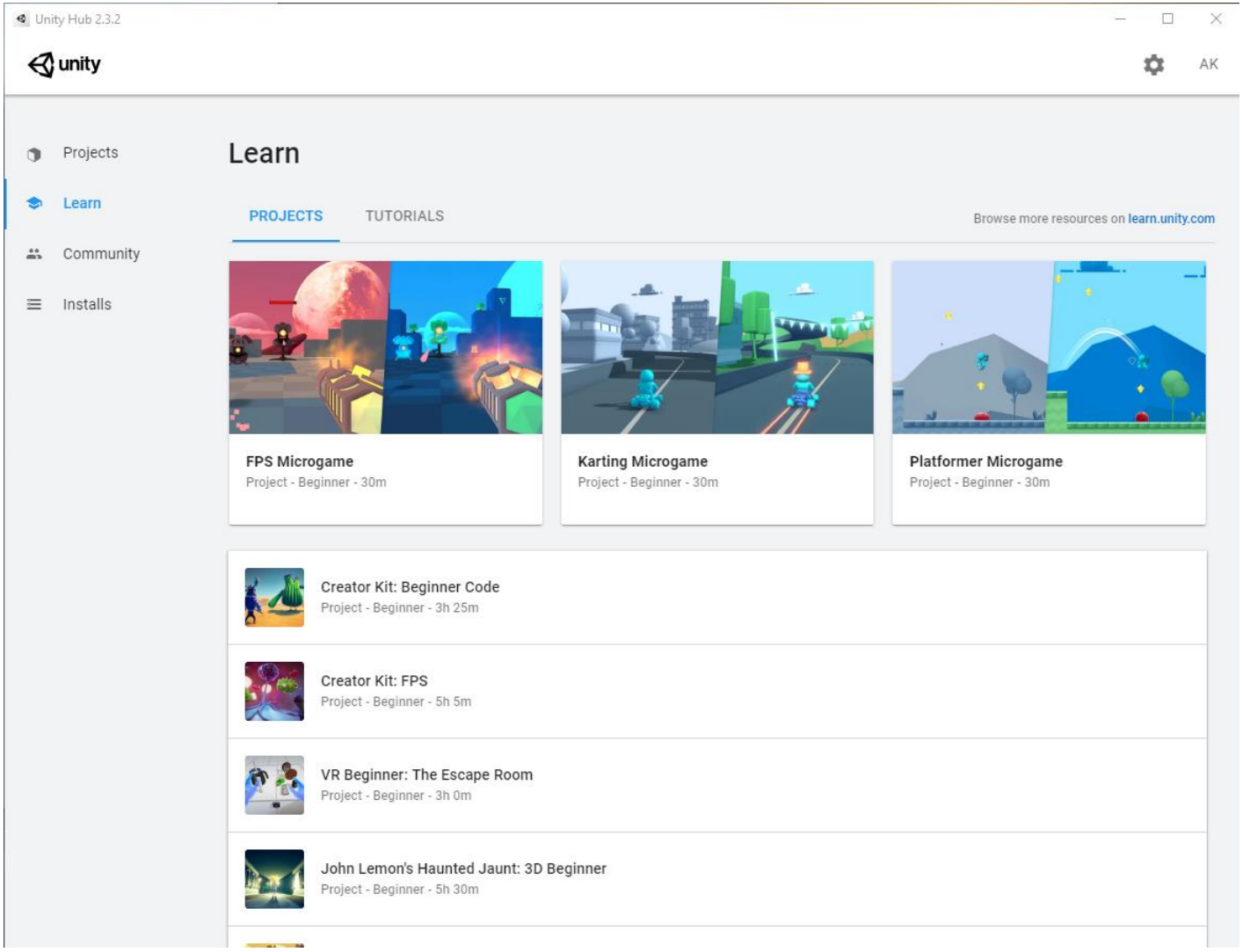
New Project

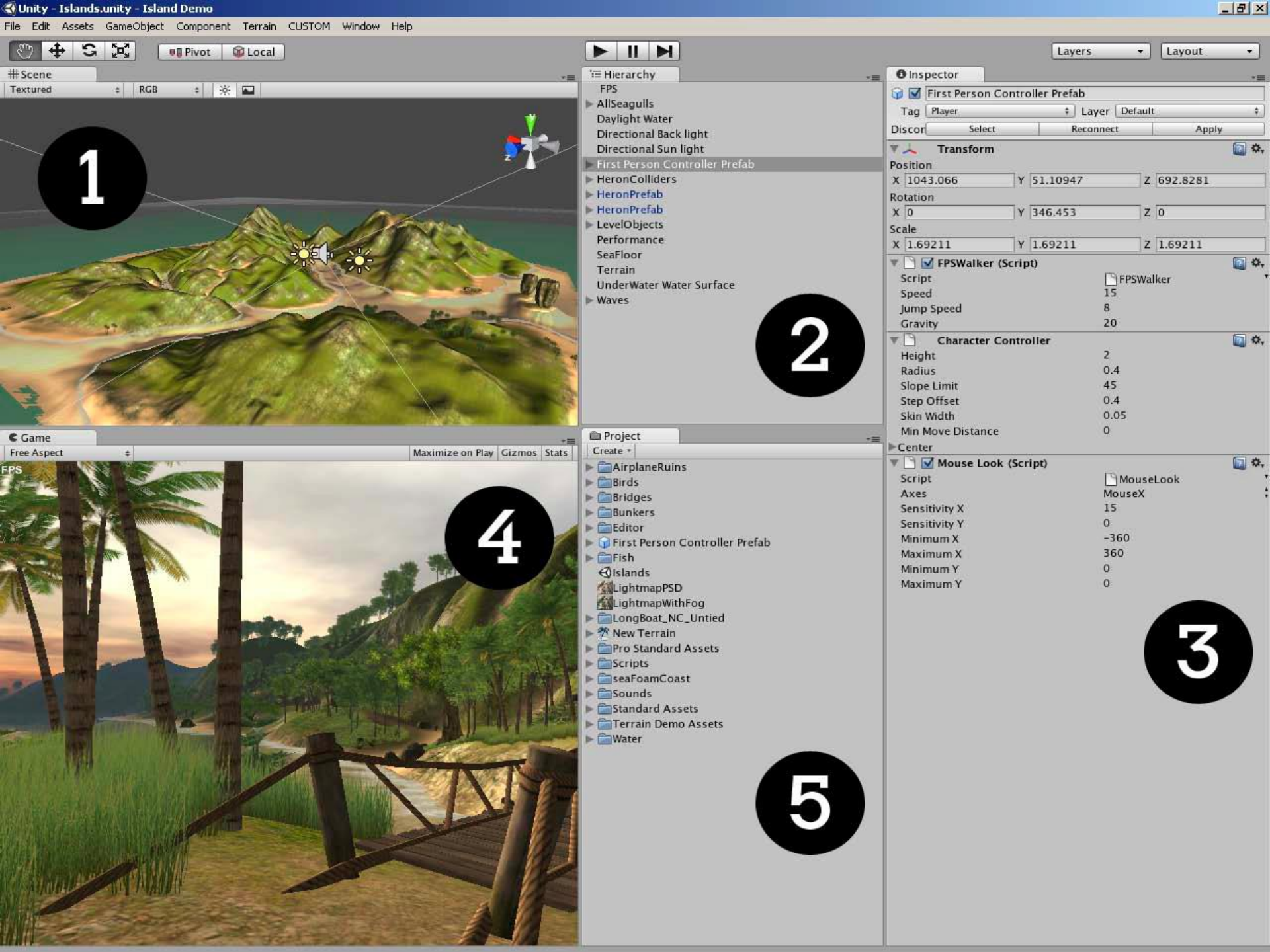

HelloWorld - SampleScene - PC, Mac & Linux Standalone - Unity 2019.2.1f1 Education <DX11>
File Edit Assets GameObject Component Window Help
Hierarchy
SampleScene
Main Camera
Directional Light
Scene
Game
Asset Store
Persp
Inspector
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
Packages
Auto Generate Lighting On

Open an Existing Project



Learn Tab


Unity Hub 2.3.2
unity
AK
Projects
Learn
Community
Installs
Learn
PROJECTS
TUTORIALS
Browse more resources on learn.unity.com
FPS Microgame
Project - Beginner - 30m
Karting Microgame
Project - Beginner - 30m
Platformer Microgame
Project - Beginner - 30m
Creator Kit: Beginner Code
Project - Beginner - 3h 25m
Creator Kit: FPS
Project - Beginner - 5h 5m
VR Beginner: The Escape Room
Project - Beginner - 3h 0m
John Lemon's Haunted Jaunt: 3D Beginner
Project - Beginner - 5h 30m


Unity - Islands.unity - Island Demo
File Edit Assets GameObject Component Terrain CUSTOM Window Help
Pivot Local
Layers Layout
Scene
Textured RGB
1
Hierarchy
FPS
AllSeagulls
Daylight Water
Directional Back light
Directional Sun light
First Person Controller Prefab
HeronColliders
HeronPrefab
HeronPrefab
LevelObjects
Performance
SeaFloor
Terrain
UnderWater Water Surface
Waves
2
Inspector
First Person Controller Prefab
Tag Player Layer Default
Discor Select Reconnect Apply
Transform
Position
X 1043.066 Y 51.10947 Z 692.8281
Rotation
X 0 Y 346.453 Z 0
Scale
X 1.69211 Y 1.69211 Z 1.69211
FPSWalker (Script)
Script FPSWalker
Speed 15
Jump Speed 8
Gravity 20
Character Controller
Height 2
Radius 0.4
Slope Limit 45
Step Offset 0.4
Skin Width 0.05
Min Move Distance 0
Center
Mouse Look (Script)
Script MouseLook
Axes MouseX
Sensitivity X 15
Sensitivity Y 0
Minimum X -360
Maximum X 360
Minimum Y 0
Maximum Y 0
3
Game
Free Aspect
Maximize on Play Gizmos Stats
FPS
4
Project
Create
AirplaneRuins
Birds
Bridges
Bunkers
Editor
First Person Controller Prefab
Fish
Islands
LightmapPSD
LightmapWithFog
LongBoat_NC_Untied
New Terrain
Pro Standard Assets
Scripts
seaFoamCoast
Sounds
Standard Assets
Terrain Demo Assets
Water
5

Unity 3D Interafce

1. Scene
   —where the game is constructed
2. Hierarchy
   —a list of GameObjects in the scene
3. Inspector
   —settings for currently selected asset/object
4. Game
   —the preview window, active only in play mode
5. Project
   —a list of your project's assets, acts as a library

Layout of Unity Editor

- The Unity editor works like any other fully dockable GUI editor.
- Each window (or view) has a tab at the top, which allows you to drag/dock each view.
- You can dock windows under one another, or dock them in certain areas of the editor, or even on separate monitors.

Hierarchy

- Contains a list of every game object inside your scene.
- Filter the items using the search box at the top
- “Create” menu allows you to create many primitive game objects.
- Interact with the game objects in your hierarchy view by
  - right-clicking them
  - using common keyboard shortcuts with them selected (Ctrl+c Copy, ctrl+v paste, ctrl+x cut, ctrl+d for duplicate, f2 rename).

Inspector

- All important information about the currently-selected game object, either through the scene view, or hierarchy view.
- Top left of the inspector is a checkbox, this allows for a very simple way to “disable” game objects.
  - They aren’t deleted, they just aren’t currently interacting with anything in the game.
- Top middle also allows you to name the game object. Right underneath are dropdowns for Tags and Layers.

Inspector: Tags and Layers

- Tags are just a simple way for us to mark or group game objects together
- Layers allow us to do sort of the same thing, but with more power, with the Unity physics system.
- In both the Tags and Layers menus, you can add new tags or layers at the bottom of the dropdown menu.

Inspector: Transform

- Every game object has one.
- The Transform is a collection of all the physical information about the game object's position, scale, and rotation in the game scene.
- We will talk more about this later

Inspector: Components

- After transform, all of the game objects unique Components are listed
- Each component serves a specific purpose.
- They can also all be turned on and off (disabled/enabled) like the game object.
- At the bottom, there is an Add Component button. This pops up a list of literally every component you can add to an object.

Project View

- Shows all of the assets being used by the current project.
- There is a big difference between Game Objects in the scene, and Assets.
- Game objects have no tie to outside of the scene, they can't be "saved"
- When you import models, textures, animations, audio clips, etc. to the project, you would find them in this Project View
  - Assets in the Project View can be found on your hard disk

Project View

- Assets can be imported through the Unity Editor, or simply by dragging them into the Assets folder for that project in your file explorer.
- You can right click inside the Project view, select the Create menu, and create a new folder, or entire assets.
- You can do any other things you usually do with files (like the hierarchy view).

Project View

- If you create a new asset here, it will also appear in your Assets folder for this project outside of the editor.
- It is almost always better to manage your file structure INSIDE of unity, not outside of unity.
- Unity tracks everything with separate .meta files, and they're sometimes finnicky.
- You can move/copy/delete/whatever every asset and folder from within Unity.

Console

- Like most IDEs/editors, unity comes with a default console.
- When scripting, you can debug straight to this console

- Debug.Log("String")
  - works like println

- This is also where compile errors, game errors, etc. pop up.

Scene View

- Its best to think of a scene as a level in Unity.
- Scenes can be used for other things besides just a game level, but at the lowest level, a scene is simply just a grouping of assets (game objects and their information).
- If you added any new game objects, you'd see them added to the scene.

Scene View

- This is where you'll spend most of your time in the unity editor

- When you click on a game object, the transform anchors light up, and you're able to drag them along their axes (or free-form drag the object) around the scene view.
  - Watch the inspector values for the object's transform as you do this.

Scene View Navigation

- When in the scene view, there's five main tools for interacting with objects.

- Pan, Axis Move, Rotate, Scale, and Transform.

- Their keybinds are Q-W-E-R-T respectively.

More on scene view navigation:
https://docs.unity3d.com/Manual/SceneViewNavigation.html

The Hand Tool

- When the Hand tool is selected (shortcut: Q), the following mouse controls are available:

- Move: Click-drag to drag the Camera around.

- Orbit: Hold __Alt__+click, and drag to orbit the Camera around the current pivot point.
  - This option is not available in 2D mode, because the view is orthographic.

The Hand Tool

- Zoom: Hold Alt__+right-click, and drag to zoom the Scene View.
  - On Mac you can also hold Control__+ click and drag instead.

- Hold down Shift to increase the rate of movement and zooming.

Flythrough Mode

- Use Flythrough mode to navigate the Scene View by flying around in first-person, similar to how you would navigate in many games.
- Steps
  - Click and hold the right mouse button.
  - Move the view around using the mouse, the WASD keys to move left/right/forward/backward, and the Q and E keys to move up and down.
  - Hold down Shift to move faster.
- Not available in 2D mode

Other tools in Scene View

- View mode
  - Shaded
  - Wireframe
  - Tons of other options here when dealing with visual effects.

- The 2D button
  - switches your camera to a 2D or 3D camera.

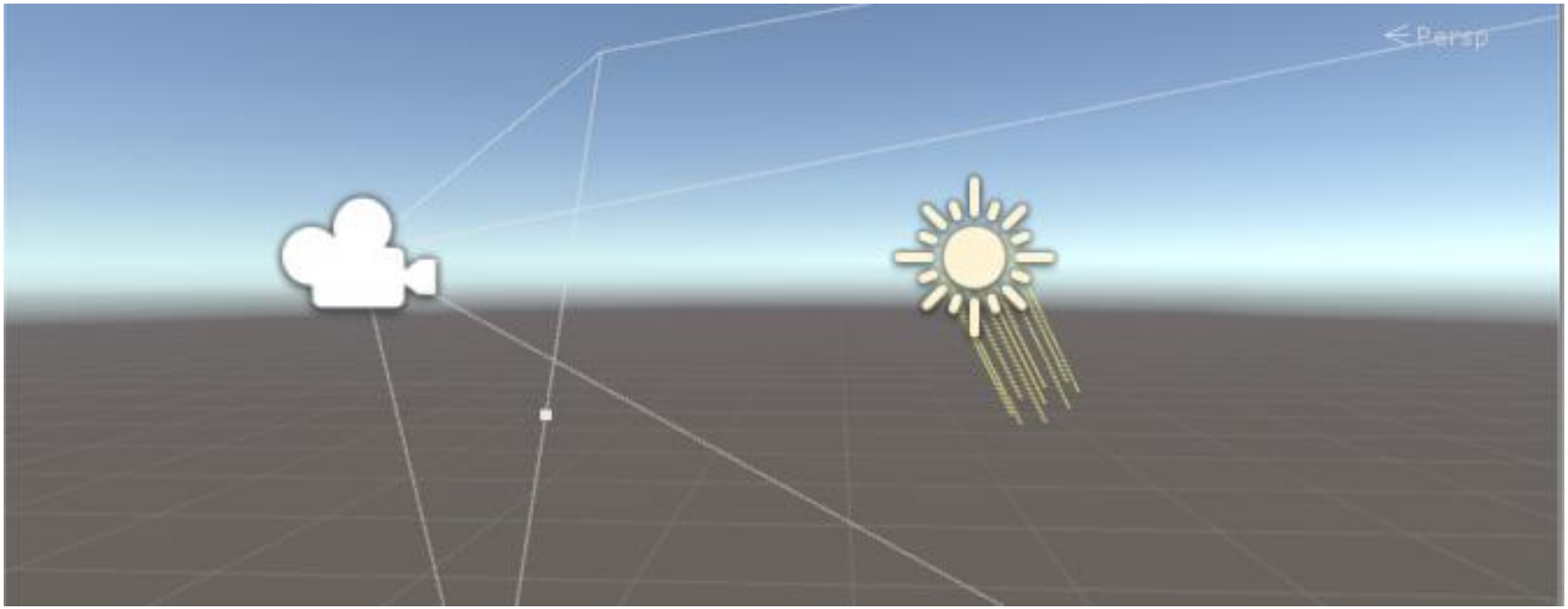
Other tools in Scene View

- Turn on/off
  - scene lighting
  - scene audio
  - environmental effects (click drop down to select specific ones).

Gizmos

- Gizmos are graphics associated with GameObjects in the Scene.
- Some Gizmos are only drawn when the GameObject is selected, while other Gizmos are drawn by the Editor regardless of which GameObjects are selected.
- They are usually wireframes, drawn with code rather than bitmap graphics, and can be interactive.
- You can also create your own Gizmos using script.

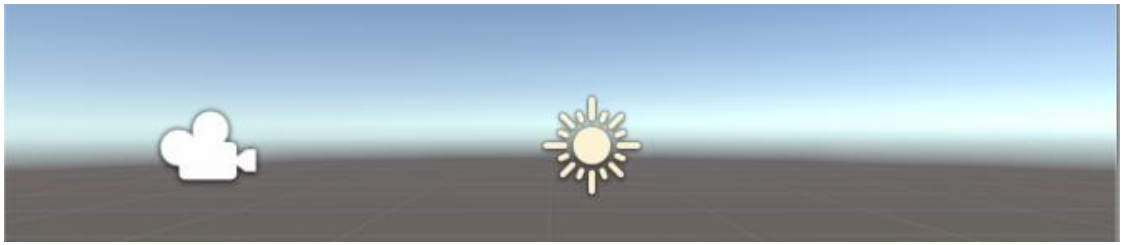
Camera and Light Gizmo



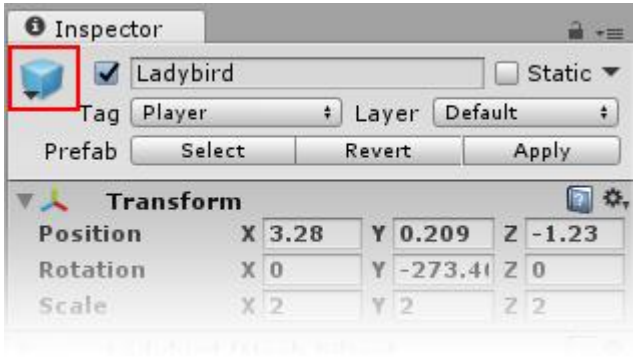
More on Gizmos: https://docs.unity3d.com/Manual/GizmosMenu.html

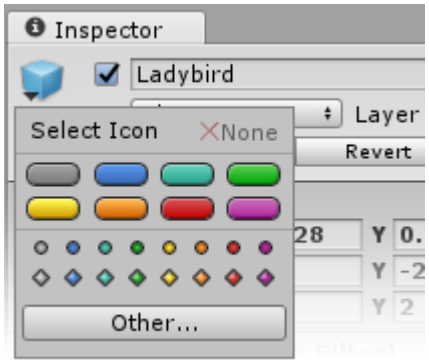
Icons

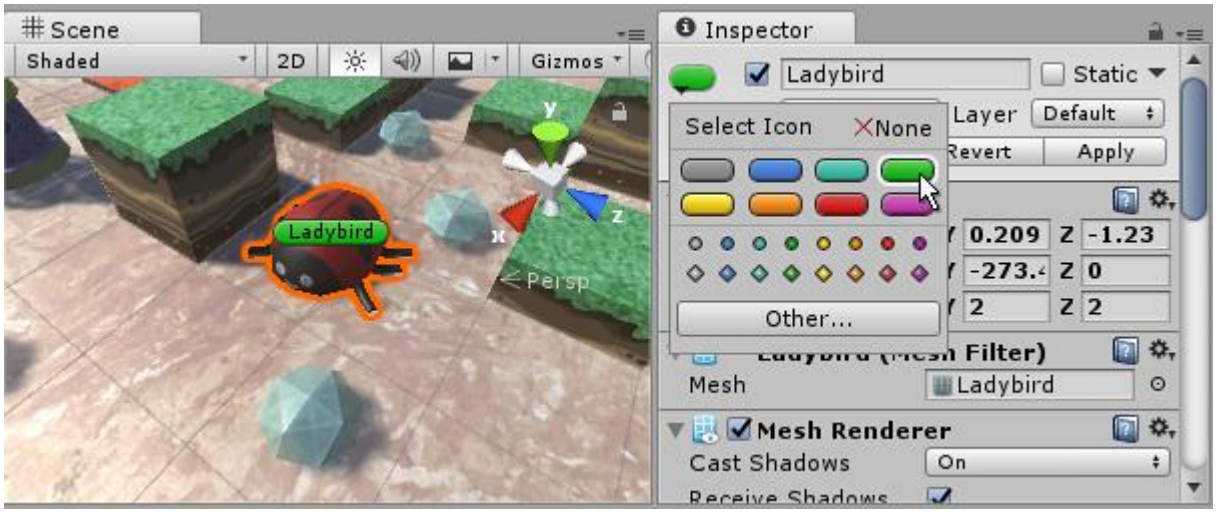
- You can display icons in the Scene view or Game View.
- They are flat, billboard-style overlays which you can use to clearly indicate a GameObject's position while you work on your game.
- The Camera icon and Light icon are examples of built-in icons; you can also assign your own to GameObjects or individual scripts



Assigning Icons







More on assigning icons: https://docs.unity3d.com/Manual/AssigningIcons.html

Game View

- This is what the camera(s) see currently.
- When you are editing your game, you cannot interact with the game view.
- Top left game view -> aspect ratio dropdown
  - You can either allow it to scale with the view or conform the view to a locked aspect ratio.

Game View

- Top right -> maximize on play button
  - Maximizes your game view window when you hit “play”.
- Mute audio button.
- Stats button.
- Gizmos button
  - same as the scene view.

Game Play Buttons

- Three buttons
  - Play, pause and step forward
- Play: plays the game, Ctrl+P
- Pause: pauses the game, Shift+Ctrl+P
- Step forward: go forward one frame, and pause the game

Game Play Buttons

- During play mode, if you change ANYTHING about your game
  - scene objects, values in inspector, etc.
  - the changes will NOT be reflected once you're out of play mode.
  - Nothing is saved.

Game Play Buttons

- If you find properties that you want to save during play mode
  - you should copy the properties (or the entire component, select object in Hierarchy or Scene view, look at inspector, right click on the component, and click “Copy Component”.
  - When outside of play mode, go to that object, right click on that component, and click “Paste component values”.

Unity C# Script Methods

- Awake is called when the script instance is being loaded.
- Start is called on the frame when a script is enabled just before any of the Update methods is called the first time.
- Update is called every frame, if the MonoBehaviour is enabled.

More Methods: https://docs.unity3d.com/ScriptReference/MonoBehaviour.html

Example Script

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

public class testScript : MonoBehaviour {

    void Start(){
            Debug.Log("This is printed from the Start Method");
    }

    // Update is called once per frame
    void Update () {
            Debug.Log("This is printed from the Update Method");
    }
}
```

Summary

- Unity 3D Concepts
- Unity Editor views
  - Scene
  - Inspector
  - Hierarchy
  - Project
  - Game
- Different tools and buttons
- Unity script
- Next class: GitHub Basics

Additional Resources

- Go through the following tutorial about Unity's Interface
  - https://unity3d.com/learn/tutorials/topics/interface-essentials/interface-overview?playlist=17090