Unity Game Engine

Unity Download, Install, and Interface

Unity Game Engine

- Unity is a professional-quality game engine used to create video games targeting a variety of platforms.
- It's also one of the most accessible modern tools for novice game developers create video games targeting a variety of platforms.
- It provides a plethora of features that are useful across many different games such as physics simulation, normal maps, screen space ambient occlusion (SSAO), dynamic and so on.
- It provide an extremely productive visual workflow, and a high degree of cross-platform support.

Unity Download

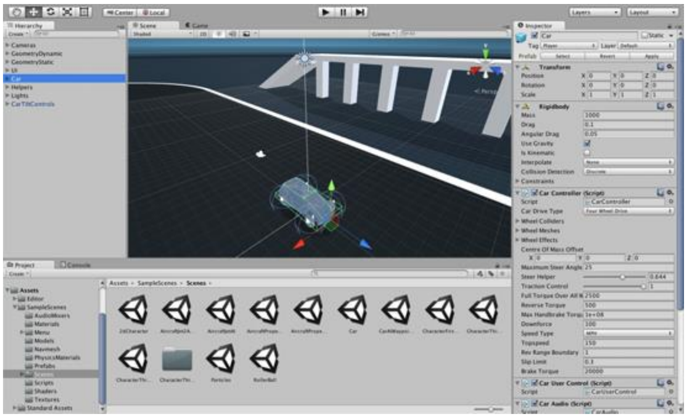
- Go to the website www.unity3d.com to download the software.
  - Unity Personal (the base version – free)
  - Unity Plus and Pro (with advanced features and commercial licensing terms)
- The URL (www.unity3d.com) is a leftover from Unity’s original focus on 3D games; support for 3D games remains strong, but Unity works great for 2D games as well.

Unity Install

- If you haven’t done so already, download the program from www.unity3d.com and install it on your computer
- After you install it, launch Unity to start exploring the interface.
  - Utilize sample tutorials to learn Unity on your own.

Unity Interface - Sections

- Window > Layouts > Default



Car ☑ □ Static
Tag Player Layer Default
Prefab Select Revert Apply

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Rigidbody
Mass 1000
Drag 0.1
Angular Drag 0.05
Use Gravity ☑
Is Kinematic □
Interpolate None
Collision Detection Discrete
Constraints

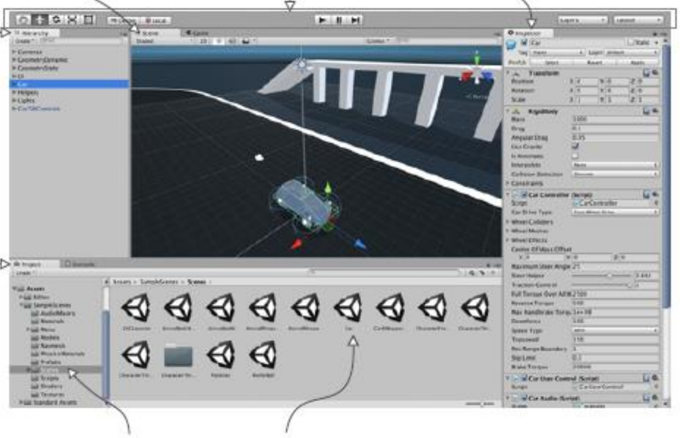
☑ Car Controller (Script)
Script CarController
Car Drive Type Four Wheel Drive
Wheel Colliders
Wheel Meshes
Wheel Effects
Centre Of Mass Offset
X 0 Y 0 Z 0
Maximum Steer Angle 25
Steer Helper 0.644
Traction Control 1
Full Torque Over All W 2500
Reverse Torque 500
Max Handbrake Torqu 1e+08
Downforce 100
Speed Type MPH
Topspeed 150
Rev Range Boundary 1
Slip Limit 0.3
Brake Torque 20000

☑ Car User Control (Script)
Script CarUserControl

☑ Car Audio (Script)
Script CarAudio

Sections

- Scene tab - place objects in the 3D scene being viewed.
- Game tab - test playing your game in Game view while watching error output in the Console tab
- Toolbar - has controls for working with the scene
- Hierarchy tab - drag and drop object relationships
- Inspector - lists information about selected objects
- Project tab - browse through all the files
- Console tab



Scene and Game are tabs for viewing the 3D scene and playing the game, respectively.

The whole top area is the Toolbar. To the left are buttons for looking around and moving objects, and in the middle is the Play button.

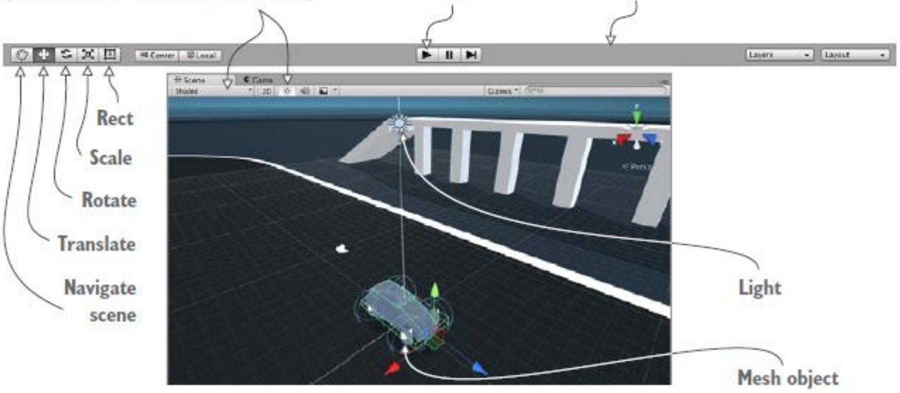
The inspector fills the right side. This displays information about the currently selected object (a list of components mostly).

Hierarchy shows a text list of all objects in the scene, nested according to how they're linked together. Drag objects in the hierarchy to link them.

Project and Console are tabs for viewing all files in the project and messages from the code, respectively.

Navigate folders on the left, then double-click the Car example scene.

Scene view, Game view, and the Toolbar



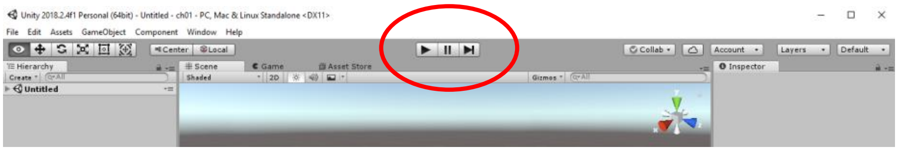
Options for aspects of the scene to display (e.g., toggle button to show lighting)

Scene view, Game view

- Scene view is where you can see what the game world looks like and move objects around. Note that the view you're seeing here isn't the same as the view in the running game—you're able to look around the scene at will without being constrained to the game's view.
- Game view: When the game is running, what you see in this view is the game.

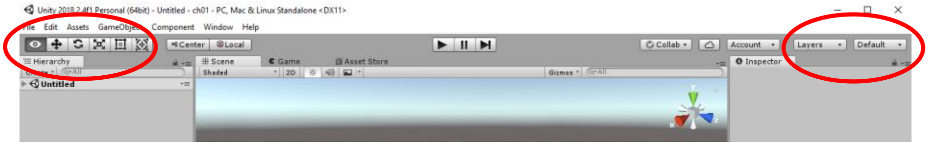
Toolbar

- To run game, hit the Play button just in the Toolbar.



Toolbar

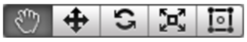
- At the left side of the Toolbar are buttons for scene navigation and transforming objects.
- The right side of the Toolbar is where you'll find drop-down menus for layers and layouts.

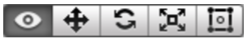

Unity 2018.2.4f1 Personal (64bit) - Untitled - ch01 - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component Window Help
Center
Local
Collab
Account
Layers
Default
Hierarchy
Scene
Game
Asset Store
Inspector
Shaded
2D
Gizmos
Untitled

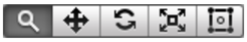
Toolbar - Scene Navigation

- Scene navigation is primarily done using the mouse, along with Alt key.
- Three main navigation maneuvers : *Move* (translate the camera), *Orbit* (rotate the camera), and *Zoom* (scale the camera).

<u>When the Hand tool is selected</u>,

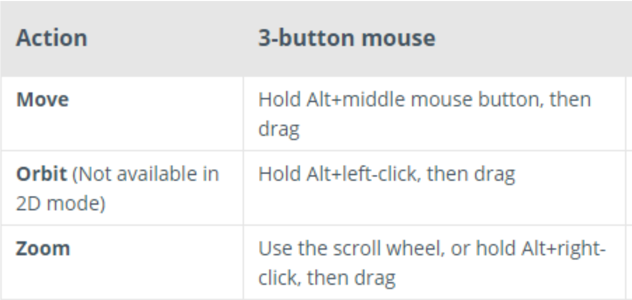
 Move: left-click and drag

 Orbit: Alt + left-click and drag

 Zoom: Alt + right-click and drag

Movement Shortcuts

- For extra efficiency, all of these controls can also be used regardless of which transform tool is selected. The most convenient controls depend on which mouse or track-pad you are using:



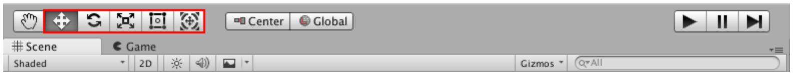
Action	3-button mouse
Move	Hold Alt+middle mouse button, then drag
Orbit (Not available in 2D mode)	Hold Alt+left-click, then drag
Zoom	Use the scroll wheel, or hold Alt+right-click, then drag

Centering the view on a GameObject

- To center the Scene View on a GameObject, select the GameObject in the Hierarchy, then move the mouse over the Scene View and press **F**.
- To lock the view to the GameObject even when the GameObject is moving, press **Shift+F**.

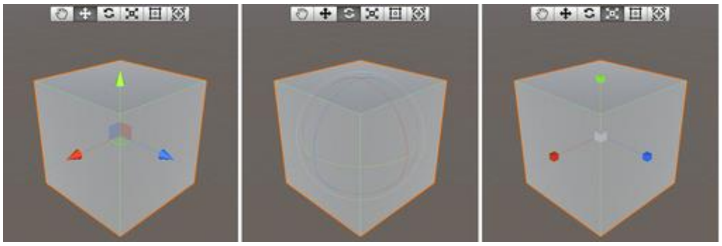
Toolbar - Object Transformation

- Select a GameObject in the Scene view.
- Selected object can be transformed by Transform tools.



Toolbar - Object Transformation

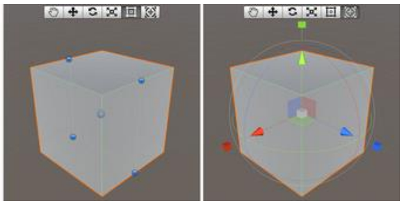
- Translate, Rotate, and Scale.



Select an object in the scene. Select a transformation tool and manipulate Transform Gizmo using mouse click & drag.

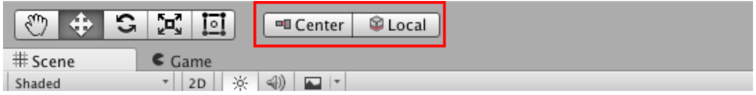
Hot keys: W (move), E (rotate), R (scale)

Toolbar - Object Transformation



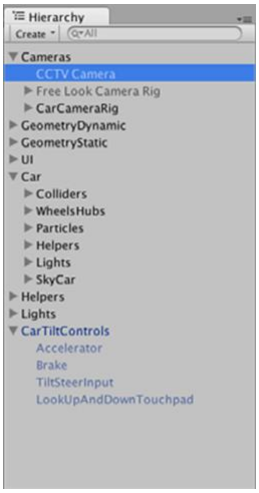
- **RectTransform is for moving, scaling, and rotating** 2D element of the object**.**
- The **Transform** tool combines the **Move**, **Rotate** and **Scale** tools.

Toolbar - Gizmo handle position toggles



- To define the location of any Transform tool Gizmo.
- **Pivot/Center** toggle
  - **Pivot** positions the Gizmo at the actual pivot point of a Mesh.
  - **Center** positions the Gizmo at the center of the GameObject's rendered bounds.
- **Local/Global** toggle for rotation
  - **Local** keeps the Gizmo's rotation relative to the GameObject's.
  - **Global** clamps the Gizmo to world space orientation.

Hierarchy tab



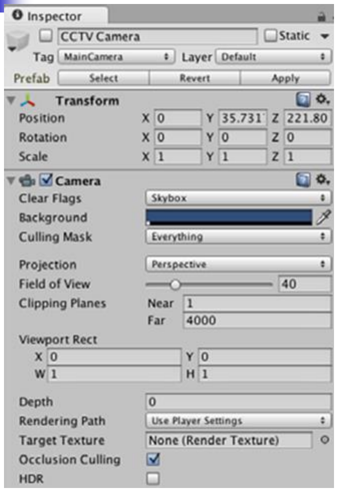
Hierarchy

Create ▾ | Q▾All

- ▼ Cameras
  - CCTV Camera
  - ▶ Free Look Camera Rig
  - ▶ CarCameraRig
- ▶ GeometryDynamic
- ▶ GeometryStatic
- ▶ UI
- ▼ Car
  - ▶ Colliders
  - ▶ WheelsHubs
  - ▶ Particles
  - ▶ Helpers
  - ▶ Lights
  - ▶ SkyCar
- ▶ Helpers
- ▶ Lights
- ▼ CarTiltControls
  - Accelerator
  - Brake
  - TiltSteerInput
  - LookUpAndDownTouchpad

- Hierarchy is a list view with the name of every object in the scene listed, with the names nested together according to their hierarchy linkages in the scene.

Inspector Tab



Inspector

□ CCTV Camera □ Static ▾

Tag: MainCamera | Layer: Default

Prefab: Select | Revert | Apply

▼ Transform

Position X 0 Y 35.731 Z 221.80

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

▼ ☑ Camera

Clear Flags: Skybox

Background

Culling Mask: Everything

Projection: Perspective

Field of View: 40

Clipping Planes: Near 1

Far 4000

Viewport Rect

X 0 Y 0

W 1 H 1

Depth: 0

Rendering Path: Use Player Settings

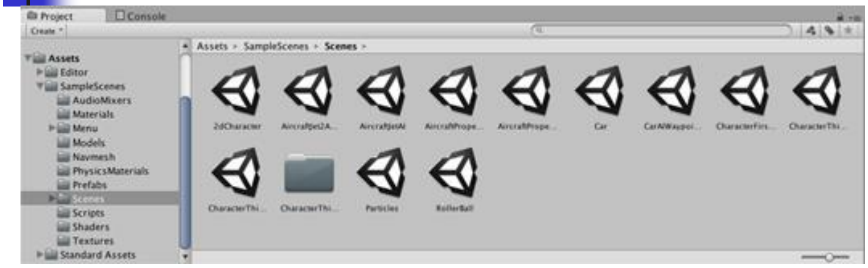
Target Texture: None (Render Texture)

Occlusion Culling: ☑

HDR: □

- The Inspector shows you information about the currently selected object.

Project and Console tabs



- Project shows all the assets (art, code, and so on) in the project. On the left side of the view is a listing of the directories in the project; when you select a directory, the right side of the view shows the individual files in that directory

Create a New Project

- Launch Unity and create a new project;
- Menu navigation: File > New Project
- Type a name for the project, and then choose where you want to save it. Choose “3D”.

Create a New Project (cont'd)

- Your game execution will start from code files linked to an object in the scene. (Game objects are built up as a collection of components, and that collection can include scripts to execute.)
- In Unity, scripts *are* components—not all scripts-only scripts that inherit from `MonoBehaviour`, the base class for script components.
  - Unity supports a few programming languages, in particular JavaScript and C#. There are pros and cons to both choices. You will be using C#.

Create a basic script

- Menu Navigation: Assets > Create > C# Script
- Or right-clicking Assets in the Project view > Create > C# Script.
- You will see the minimal boilerplate code.
  - When you open the C# script, it will launch Visual Studio.
  - Set your favorite IDE for C# editor through Edit > Preferences > External Tools > External Script Editor

Code template for a basic script component

- a new C# script: the minimal boilerplate code that defines a valid Unity component.

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

  public class MyScript : MonoBehaviour {
    // Use this for initialization
    void Start () {
    }

    // Update is called once per frame
    void Update () {
    }
}
```

Code template for a basic script component

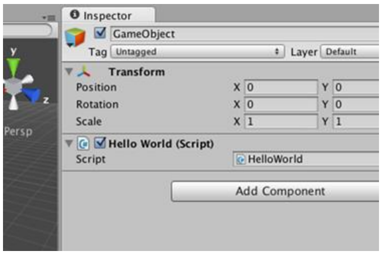
- `Start()` is called once when the object becomes active (which is generally as soon as the level with that object has loaded)

- `Update()` is called every frame. Thus your code is run when you put it inside these predefined methods.

- A *frame* is a single cycle of the looping game code.

Create an empty object

- You also need an object in the scene to attach the script to.
- Select “GameObject > Create Empty”
- Blank GameObject will appear in the Hierarchy list.
- Add the script to the GameObject.
  - Drag the script from the Project view over to the Hierarchy view and drop it on the empty GameObject.
  - Or use “Add Component” in the Inspector view.



Inspector

☑ GameObject

Tag Untagged   Layer Default

Transform

Position	X	0	Y	0
Rotation	X	0	Y	0
Scale	X	1	Y	1

☑ Hello World (Script)

Script HelloWorld

Adding a Console Msg

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

  public class MyScript : MonoBehaviour {
    // Use this for initialization
    void Start () {
      Debug.Log(“Hello World!”);
    }

    // Update is called once per frame
    void Update () {
    }
}
```

Hit Play in Unity and switch to the Console view. You’ll see the message “Hello World!” appear.

Scene navigation and keyboard shortcuts

- Scene Navigation: http://docs.unity3d.com/Documentation/Manual/SceneViewNavigation.html

- Hot keys: http://docs.unity3d.com/Documentation/Manual/UnityHotkeys.html