

GD 1133 - Game Development



Reminders

Installing Unity 5

- Navigate to <https://unity3d.com/get-unity/download>
- Click on "choose your unity + download" button
- Click "download now"

Unity 3D Installation

B. Components needed

- Unity 5.4.x
- Standard Assets
- Example project
- Android Build Support.
- WebGL Build Support
- Windows Build Support *(if you are using a windows machine)*

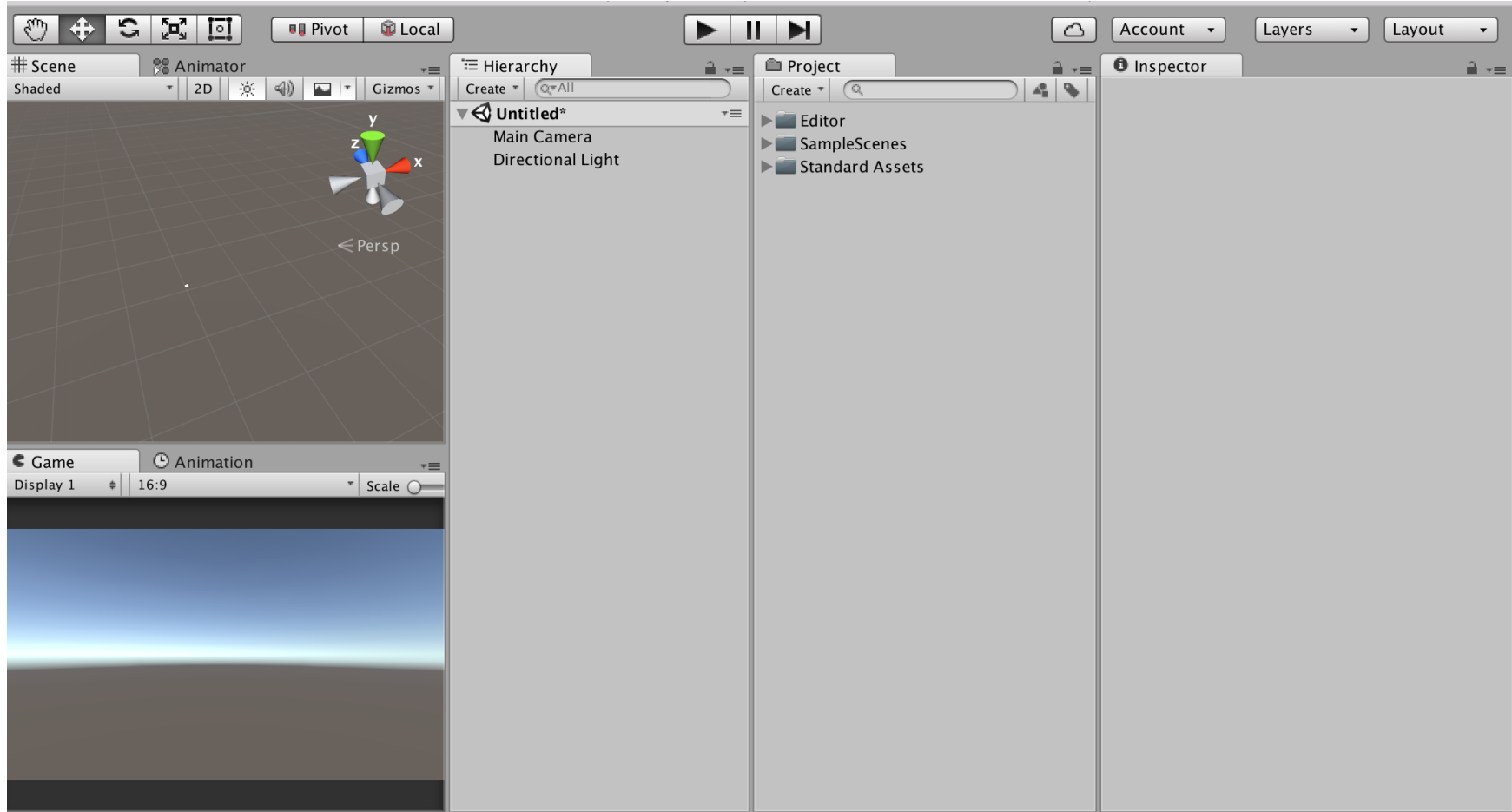
Exploring Unity 3D

- Launch Unity application
- Create a Unity account (if you are a new user)
- Sign in using your credentials
- Explore the Example project by Unity
(/Users/Shared/Unity)

Configuring Workspace

- Customize as you see fit
- There are different layouts available:
 - Window > Layouts > 2 by 3
- You can also change project column layout:
 - Project tab > One column layout

My screen layout



Useful Shortcuts

- https://cdn.tutsplus.com/active/uploads/legacy/tuts/270_IntroToUnityPart1/unity-cheat-sheet.pdf
- <https://docs.unity3d.com/Manual/UnityHotkeys.html>

Editor

- Game mode
- Editor mode
- Hierarchy view
 - game objects list
 - zoom (F key), move, scale objects

Solar System Project

- Download the solar system assests
- Unzip the folders on your computer
- Open Unity
- Create a new project
- Import new assets by dropping all folders in Project column.

Coordinate Systems

- 2D Bitmap Graphics:
 - integer based
 - 1 unit = 1 pixel
 - jpeg, png, gif, etc.
 - device specific
- 2D Cartesian:
 - Real-based based
 - 1 unit = 1 meter
 - device independent

Points / Vectors

- Point = location
 - $P = (x_1, y_1)$
- Vector = displacement in space
 - displacement from origin
 - $\text{Vector2}(x,y)$

Unity Vectors

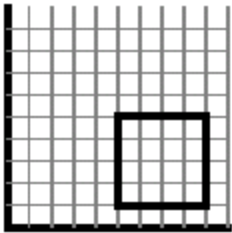
- 3D Cartesian Coordinate System:
 - `Vector3(x,y,z)`
- 3D Transform represented by:
 - position
 - rotation
 - scale

Transform

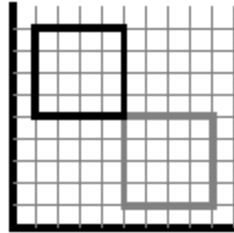
- Position = $\text{Vector3}(x,y,z)$
 - represents displacement from origin of world coordinate system
- Rotation = $\text{Vector3}(x,y,z)$
 - degree rotation around x, y, and z axis.
- Scale = $\text{Vector3}(x,y,z)$
 - scale factor on each axis
 - 1 unit = 100%

Transformations

original

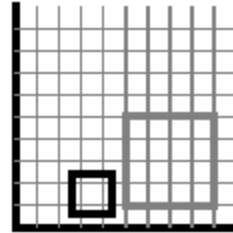


translation



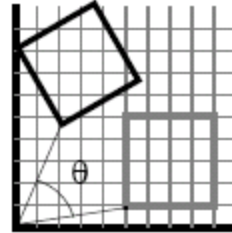
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

scaling



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

rotation



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Create Game Objects in Unity

Creating game objects for:

- sun
- earth
- moon

Sun

Steps to create sun:

- GameObject > 3D Object > Sphere
- Rename Shpere to Sun (in Hierarchy View)
- Set position transform to Vector3(0,0,0)

Earth

Steps to create earth:

- GameObject > 3D Object > Sphere
- Rename Sphere to Earth (in Hierarchy View)
- Transform:
 - Position = Vector3(2,0,0)
 - Rotation = Vector3(0,0,0)
 - Scale = Vector3(0.5,0.5,0.5)

Moon

Steps to create moon:

- GameObject > 3D Object > Sphere
- Rename Sphere to Moon (in Hierarchy View)
- Transform:
 - Position = Vector3(1.5,0,0)
 - Rotation = Vector3(0,0,0)
 - Scale = Vector3(0.1,0.1,0.1)

Main Camera

Change Camera Transform

- Select Main Camera from Scene View
- You can zoom in with (F key or Double click)
- Change the transform to:
 - Position = `Vector3(0,0,-3)`
 - Rotation = `Vector3(0,0,0)`
 - Scale = `Vector3(1,1,1)`

Save Scene

Let's save the current scene.

Steps:

- File > Save Scene
- Save as "Solar System" in the Assets folder (default)
- Create "Scenes" Folder in Project View
- Place the new "Solar System" file in "Scenes" folder.

Unity Behaviors

- Add script component to move the object
- Sun:
 - Add "RotateAround" script to sun
 - Add "Sun" as Target
 - Set speed to 5
 - Test by going into "Play Mode"

Unity Behaviors (1)

- Earth:
 - Add "RotateAround" script to Earth
 - Add "Earth" as Target
 - Set speed to 10
 - Test by going into "Play Mode"
- Moon:
 - Add "RotateAround" script to Moon
 - Add "Moon" as Target
 - Set speed to 15
 - Test by going into "Play Mode"

Making Earth Rotate Aaround Sun

- Add another "RotateAround" component to Earth
 - Add "Sun" as Target
 - Set speed to 40
 - Add "Moon" as a child on Earth
 - Test by going into "Play Mode"
 - Then, add another "RotateAround" component to moon
 - add "Earth" as it target
 - set speed to 5