## Section 1: Unity Basics

To begin, you will need to download Unity from their website. Next, step-by-step instructions will be provided for creating a basic environment in which a keyboard is used for moving an object. Upon collision with another object, both the User controlled object and the designated Target object change color. Additionally, information will be provided for understanding the Unity interface including the basics about materials, scripts, and scenes. A tutorial environment can be downloaded that has all of the following information included.

- 1. Download Unity
  - a. Link: <u>www.unity.com</u>

- 2. Unity Hub
  - a. To load a project, select the Add button or press New to create a new project. This is where you will find the Tutorial project if downloaded from the MOCORE website.



b. If creating a new project, press NEW and create a 3D Template.

c. Find the Unity Project file folder saved on your local computer. Press Add File on the load screen when the Project file folder is selected.



## 3. Main User Interface

a. On the Main Screen you will see the basic dropdown menus at the Top, a Hierarchy tab on the left side where each object is identified, the Scene and Game windows split in the middle (which can also be overlaid if the Game window is dragged on top the Scene window, noted below), and the Inspector/Services tab on the right side for specifics about each object highlighted in the Hierarchy tab.



b. When the Main Camera is selected on the Hierarchy tab on the left side, object properties such as position, rotation, and scaling are listed under the Inspector tab.



c. *Optional:* To save monitor space while working on the environment, the Game window can be overlaid on top of the Scene window by selecting the Icon (see below) and moving the window to the same position on the Scene window. You can then navigate between these two screens when working on the project. The Camera Preview previews what is seen in the Game window by the Main Camera.



# Scene	C Game	6	😤 Ani	imato	r 🖆	i Asset	t Store
Shaded	*	2D		10)	* -	90	

Unity 2019.1.10f1 Personal - U	ntitled - Tutorial - PC, Mac & Linux Standalone <dx11></dx11>				(=)	D X
Con Assets GameObjec	Component Window Hep			Collab •	Account + Layers +	Layout +
II Hierarchy	= # Scene Came N Animator @ Asset Store			·= O Inspe	ctor Services	ä -=
Create * (0+A)	] Shaded * 20 ♥ 10 \$ * ∞ 0		🛠 🎫 * Gizmos * /Q*All		Main Camera	Static 💌
C Main Camera				Tag	MainCamera I Layer D	efault 1
<ul> <li>Directional Light</li> </ul>				Position	X 0 Y 1	Z -10
				A Rotation	n X0 Y0	ZO
		242		< Persp	A 1 1 1	6 1 5 -1 A
				Clear F	lags Skybex	1
				Backgro	ound Examples	1
				Coung	mask Everything	
				FOV Ax	is Vertical	1
				Field of	View	60
				Finyaica	Diseas Maar 0.2	
				Cappan	Far 1000	
				Viewpo	rt Rect X 0 Y 0	
				Depth	-1	
				Render	ing Path Use Graphics Setting	n 1
				Target	Texture None (Render Text on Culling 🖌	une) O
				HDR	Use Graphics Setting	15 1
				MSAA Allow D	Use Graphics Setting	<u>(i i</u>
			Camera Previ	Target	Display Display 1	
				0 Z A	udio Listener	🖬 🕫 e.
	/				Add Component	
D Project Console				a		
Create *	Assets >		(9,	2491		
T Assets						
> Scenes						
⊨ I Packages						
	Scenes					
				र्देशलीय	Auto Genera	ate Lighting Off

- 4. How-to: Create Objects
  - a. Right click on the Hierarchy tab or mouse over the GameObject tab at the top of the screen in the Dropdown menu to create a new 3D object.
  - b. For this tutorial we will be creating multiple objects including a table, and two spheres. One sphere will be controlled by the User and the second will be designated as a Target sphere.

🚭 Unity 2019.1.10F1 Personal - Untitled - Tutorial - PC, Mac & Linux Standalone* <dx11></dx11>		- 0 X
File Edit Assets GameObject Component Window Help		
Concerner Outral Street Street		C Lovers - Lavers - Lavers - Lavers -
I Couse' (50) (20) (10) (20) (10) (20) (10) (20) (10) (20) (10) (20) (10) (20) (10) (20) (10) (20) (10) (20) (10) (20) (10) (20) (10) (20) (10) (20) (20) (20) (20) (20) (20) (20) (2	2 ■ + Ganet + (TAL)   Digiting 1 = 1   Prov August	t   Galle () Le Maximize On Hay   MarcA
	al True	
Copy Paste		
Renome Duglicate Delate		
Select Children Select Prefab Root		
Create Empty	Create 2D abiast	
2D Object > Cone 2D Object > Sphere Effect > Conce	Create 3D object	
Light > Cylinder		
Video > Qued		
UI > Text - textMeshPilo		
Persin Tree		
Wind Zone 3D Text		
M Project Concole	6	
Favorites Assets >		
M Scates		
Packages		
Scenes		
		Auto Generate Lighting Off



- c. Additional Step: Moving Camera and Screen Orientation
  - i. For better viewing and control of individual objects, move the orientation of the Scene screen by pressing the Middle Mouse or Right Mouse button to move and rotate the screen, respectively. The Scroll Wheel zooms Scene screen in and out.
  - ii. To move individual objects, first select the object and use the Green, Blue, and Red arrows the change the objects Position.
    - 1. You can also manually adjust the position by changing the numbers with the Inspector tab.
  - iii. To rotate the object use the Green, Blue, and Red circles.
  - iv. To change the dimensions of the object use the Green, Blue, and Red cubes.
  - v. In the top left corner there are different buttons for different object controls related to movement, rotation, or movement+rotation.
  - vi. Move the Main Camera to a position the looks down upon the Table object at a comfortable height and rotation (~45 degrees).



- d. Next, create two spheres of equal size and label one 'User' and the other 'Target'. Move both spheres to directly above the Table Top.
- e. Select the User ball, press Add Component at the bottom of the Inspector tab, and select RigidBody.
  - i. By adding the Rigidbody Component the sphere now has a specified Mass and other physical properties.
  - ii. Ensure that the Use Gravity box is checked on.
- f. Repeat the last step for the Target ball.





- 5. How-to: Create Materials
  - a. Different material properties can be specified for each object such as color and transparency. For this tutorial, a basic introduction to different materials for different colors will be presented.
  - b. This can also be found under the Assets tab at the Dropdown menu in the top left corner.



c. After you create the material, select the white square next to the label Albedo under the Inspector tab to manually change the color.



- d. Create four colors. A brown one labeled Table, a black one labeled User, a green one labeled Goal, and a red one labeled Contact.
- e. Drag and drop the material onto the object you want to apply the material too.
  - i. Apply the Table color to the Table, the User object to the User sphere, and the Goal color to the Target object. The Contact object will be used at a later time.



- 6. How-to: Create Scripts
  - a. Now that the environment is created the final step is creating scripts that help run the task or for creating event triggers such as sound cues or color changes that correspond with your desired effect.
    - i. We will create the following 2 scripts:
      - 1. One to move the User controlled sphere.
      - 2. One to change the material (and color) of both the User and Target spheres when they collide into each other.



- b. Create a script and name it MoveCube. This Script will contain code written in C# within the Visual Studios program for controlling the User sphere with the computer keyboard.
  - i. *Author note:* There are many ways to move an object. A google search for 'moving an object in Unity' will supply many examples. For example, you could apply a directional force that rolls the sphere across the table. In this tutorial we will present a simple code for changing the position in the X and Z directions based upon a set scaling factor.



- c. Double click on the new Script to load Visual Studio.
- d. At the basic level, anything added following 'void Start' will be applied immediately upon playing the scene while 'void Update' will continuously run and be updated every frame of the scene (~50 Hz).

Image: Project Build         Debug         Test         Analyze         Tools         Extensions           O         -         -         -         -         -         -         -         -         -         -         -         -         -         -         -         -         -         +         -         -         +         -         -         +         -         -         +         -         -         +         -         -         +         -         -         +         -         -         +         -         -         +         -         -         +         -         -         +         -         -         +         -	Window Help Search Visual Studie (Chri-C) 🔑 Tutorial ky - 🏂 - 1:5 대 또 또 제 제 제 제 제 구				SS — D × 년 Live Share &
<pre>bucches = X Dasemby/SBAW 1 Evansy System.Collections: 2 using System.Collections.Generic; 3 using UnityEngine; 4 1 reference 5 Epublic class Collision : MonoBehaviour 6 { 7 { 9 // Start is called before the first frame 8 E void Start() 9 10 11 12 12 14 15 16 17 17 17 17 17 17 17 17 17 17</pre>	- ≪talloon update <b>void Start</b>				Solution Episore
13         10         0 monose is caree once per mase           14         e         void bplate()           15         (         (           17         )         )           18         )         19	void Update				Sendaran Espinore 'Team Espinore Proporties - 9 × 22 (24 /#
111% • @ No more frond / d •	· ·			Visual Saudio 2	19 update ×
Entire Solution • (20 Errors (1) 0 Warnings (1) 0 Messages Build + Inte Code Description Error List Output	forme -	Pro	ject File	Sea Version 1665 e	downloaded and ready to install. View details
ET Ready		Ln 19 0	Cel 1 Ch 1	PIS .	🕈 Add to Source Control 🔹 🐴

- e. The following code has been created to change the position of the designated object based upon a scaling factor labeled as 'delta' according to arrow directions from a keyboard.
- f. The variable labeled delta is identified as a 'public' variable. This means it is accessible from the User Main Screen within Unity and therefore can be changed. If the variable is set as 'private' it will be hidden from control within Unity and only accessible within Visual Studio.



- g. Upon each Update,
  - i. IF a Left Arrow is detected the object moves in the -X direction
    - 1. By a factor of delta (specified within the Unity Program).
  - ii. IF a Right Arrow is detected the object moves in the +X direction.
  - iii. IF an Up Arrow is detected the object moves in the +Z direction.
  - iv. IF a Down Arrow is detected the object moves in the -Z direction.



Author note: When creating this tutorial I accidentally swapped the User and Target sphere. Which sphere is which is completely up to you, but note here that the Target and User names are in different order from the previous steps.

Previously the upper sphere was renamed as User, in this case the black sphere which is positioned below, is labeled as 'User'. This will be corrected at a future date for consistency.

- h. Apply the MoveCube script to the User sphere
- i. There are multiple ways to apply a script to an object.
  - i. You can press Add Component and find the correct Script you created.
  - ii. You can drag and drop the script onto the component at the open area at bottom of the object information in the Inspector tab.
- j. The 'delta' label can now be seen within the MoveCube script within the Inspector tab and can be changed between trials for modifying the scaling factor of the User sphere position change.



- k. Now is a good time to test your Project if you have not already. Press the Play button at the top of the screen and wait for Unity to load. Once the program has loaded you should be able to control the position of the black User sphere by pressing the arrow keys on your keyboard. Each keypress is a movement command.
  - i. *Note:* A 'delta' of 0.01 will be extremely low and movement will be tough to see at first. A recommended delta is between .1 and 1.

- 1. Next, create a second script and label it CollisionCube.
  - i. This script will contain event triggers that will cause a color (material) change when the User sphere contacts the Target sphere.
- m. The following image is an example script but as noted before there are multiple ways this can be done.
- n. First, we created two public variables, on is a Renderer labeled Target and one is a Material labeled Contact.
  - i. The Renderer label specifies which gameObject in the virtual environment is the 'Target' and will change color upon collision.
  - ii. The Material label specifies which Material property will be used as the new material upon object collision.
    - 1. Because both objects are set as public they must both be set within the Unity environment. See below for further instructions.
- o. The 'void Start' and 'void Update' were removed and a 'void OnCollisionEnter' was used.
  - i. This is a code that exists within the program and must be called upon and used in a specific way. *Google 'Unity oncillisionenter' for further documentation*.
- p. Finally, the code below states that IF the gameObject (which is the object this script is attached to) makes contact with an additional gameObject named "Target" that both the gameObject and Target will change materials to the specified material, in this case named 'Contact'.





q. Apply the CollisionCube script to the User sphere.

r. Click on the User sphere within the Hierarchy tab. Drag and drop the Target sphere from the Hierarchy tab to the Target option within the CollisionCube script within the User sphere. Additionally, drag and drop the Contact material to the Contact option within the same script.



- s. Press Play to test the environment.
  - i. When the User sphere contacts the Target sphere both objects should immediately change to red.





Congratulations! You have completed the MOCORE tutorial for creating a Unity project. Additional projects can be found in the following sections and downloaded from the MOCORE website. The basics will be removed, and the instructions will detail the specific differences with each project and the additional information not covered in this tutorial.



Optional: You can create folders to better organize your materials, scripts, and scenes.

