

01 Color Capture

Code Community, 2017

Instructions: Follow the steps on your own or with a partner. Ask a tutor to help you if you get stuck. Or show a tutor how things work.


Project Goal: A simple matching game involving color.

Unity Knowledge: *scene, game object, component, MonoBehaviour, Script, Debug, Color, moving between IDEs and scripting.*

Integrated Programming Environment (IDE): panels: scene, hierarchy, inspector, project

Coding Knowledge OOP (object oriented programming) – classes and instances

Explore:

- 1) Locate the panels in the Unity IDE:
 - a. Hierarchy: is where the game objects are organized.
 - b. Project: is where game assets are organized.
 - c. Scene/Game/AssetStore/Animator panel: Click on Scene.
 - d. Inspector: Where you see the components of an object.
- 2) What Game objects are in the hierarchy? Click on 'Oliver' and take a look at his components in the inspector. What are the three components?
- 3) All three game objects (Main Camera, Oliver, TBall) have a Transform component that shows where they are on the screen. How big is Oliver? Hint: Look at 'Scale'
- 4) Where is Oliver – what are his coordinates?  Try moving him. You will need to click on the right most the game object panel: (Script)."
- 5) Oliver has a component "Oliver Shares" Check out the script by clicking on it. The comments in the script explain how a MonoBehaviour can be used .
- 6) Notice that the 'Oliver Shares' component has a *public* variable called 'Oliver Color'. You can change its value in the Inspector.
- 7) Check out the components for the TBall. Look at the MonoBehaviour for TBall.
- 8) If you read the scripts, you should know what keys will make things happen on the screen. Run the game (using the > button.) VERY IMPORTANT! If you change things in the IDE while the game is running, those changes won't be kept when you save the game.
- 9) One more thing: Notice that your game objects are part of a scene called 'OliverInSpace'. Always make sure you save the scene. In the hierarchy there will be a little * at the end of the name if it needs to be saved.

Practice:

- 1) Notice that the assets include a cat called 'Catherine Image.' Create a game object in the hierarchy that uses this image. Study the scripts for the TBall and Oliver so that Catherine changes her color to Oliver's based on a keystroke that is different from the ball. You can copy a script by highlighting it (clicking on it) and selecting Edit/Duplicate from the menu bar.
- 2) Make it a game! A very simple game for now. Have keys that change to colors other than green. Challenge your friend to find the key that matches a color. Just call out the color and have your friend guess which key is used.
- 3) Instead of setting the color to the start color, try using Random, or Random.ColorHSV. It helps to write you idea as a comment first. Ask a tutor for help debugging your idea. Don't forget to look at the documentation.

Resource Information:

The Unity IDE

The Four Main Panels: We'll be discussing Unity's standard layout, which should be the default when you start a new project. Here, we give a brief overview of the purpose of each panel. If you're interested in learning more about the panels, check out the documentation.

1. The scene/game/asset store/animator panel: In the center of the screen, there is a panel that has four tabs on the top (scene, game, asset store, animator). The scene tab allows us to move objects around. The game tab allows us to view the scene as it will look when the game starts. The animator tab is used to create animations.
2. The hierarchy panel: To the left of the scene panel, the hierarchy panel gives us a list of the game objects in the scene. Game objects can be organized into a hierarchy, like files on a computer, with some objects being stored under others.
3. The inspector panel: To the right of the scene panel, the inspector panel lists all the properties of a selected game object. Editing, adding, or removing these properties lets us change the behavior and appearance of a game object.
4. The project/console panel: Below the scene panel is the project/console panel. The project panel provides a typical file manager layout of all your assets. The console panel is a place where errors and messages from the code appear while the game is running.

An Object Framework

A unity project is referred to as a *game*, if it includes user interaction, and is referred to as an *animated video* if it does not. Unity3D projects consist of a set of *objects* that *extend* the *engine*. When you build a project you are defining *instances* of particular *object classes*. Examples of classes include *scenes*, and *game objects*. An instance of a class can have *components*; both *data* (information about the object) and *methods* ('actions' that the object can perform) that describe how it works.

In Object-Oriented Programming (OOP) code is written as a set of *classes*. A class organizes data and methods. An *instantiation* or *instance* of a class is an object that has specific values for that data, and can be invoked to execute one of its methods. Objects are linked through *has-a relationships*. For example a 2D object called a sprite has another object called a sprite renderer. Classes can have subclasses, where the subclasses *inherit* or have access to the parent class data and methods. This *encapsulates* information so that all of the details are not overwhelming. Most coding in Unity3D is creating instances of classes. The IDE includes an *inspector panel* lets you link objects together by making one object a *component* of another. Game construction involves creating game objects and giving specific values to the components. Sometimes that value will be a *MonoBehavior*, which is described through a script. In OOP its called a *method*. This ability to move between data and methods (values and scripts) is why Unity3D is such a great design tool. However, it requires that you understand how to move gracefully between the GUI and coding.