The Unity interface

To start with we want to create a new unity project, customized for 3D. There is very little difference between selecting 2D and 3D, it just changes how things are set up at the start. For now we want to create a new 3D project.

***Create a new 3D project.***

This is what the unity interface looks like. If yours looks different to mine it may be because you are using a different layout. To change your layout use the drop down in the top right corner labelled ‘layout’.

***Use drop down to change layout, find one you like.***

Changing layouts just moves these panels around, you should still have a copy of the relevant panels.

## 

The unity interface consists of 5 main panels:

1. The scene view
2. The hierarchy
3. The game view
4. The inspector
5. The project view

# Scene View

The scene view is where you visually construct your game, adding and editing objects. When a Unity project is opened for the first time it starts with some objects already created in the scene; specifically, a camera, and a light (more on those later). This may look slightly different depending upon whether you chose a 2D or a 3D project.

Move around the scene view with:

* right click and mouse move to rotate camera,
* mouse wheel to zoom,
* middle click and mouse move to pan,
* selecting the hand icon from the bar below the title bar allows for panning with left click and mouse move

For now, lets add some objects to the scene.

***Add a 3D cube to the scene.***

***GameObject > 3D Object > Cube***

***OR***

***Right click in hierarchy panel > 3D Object > Cube***

### Manipulating objects in the scene view

Left clicking on an object in the scene view selects it. A selected object can be manipulated by using one of the 3 buttons on the left side of the editor window, below the title bar.



***Select one of the transform tools and drag on the handles attached to the cube in order to edit its position, rotation, or scale.***

An object can also be manipulated by directly setting the values of its transform component in the ‘Inspector’ window.

*X axis describes left to right or the horizontal plane*

*Y axis describes up and down or the vertical plane*

*Z axis describes forwards and backwards, or the depth plane.*

# Hierarchy panel

The hierarchy panel is a list of all the game objects currently in your scene. Selecting an object in the hierarchy also selects it in the scene view, and vice versa.

### Scenes

A game consists of one or more scenes. A scene is a container for objects and their current settings. Unity scenes are just like scenes in a movie; they contain all the people, things, and environment from a particular setting. A scene can sometimes correspond to a map or level from a game but can also be used to make a main menu.

When we start a new project Unity begins with an unsaved scene that contains our light and camera.

***Save the current scene.***

***File > Save scene.***

Scenes can be switched between during game play as a player progresses through the game. We can also switch between scenes while constructing our game. When a scene is loaded, the resources from the previous scene are released from memory (and will be loaded again when re-opening that scene).

# Game View

The game view is where you view and play your game during its construction. The game can be started or stopped by pressing the play button at the top of the editor, or paused by pressing the button next to it. While paused you can step through the game frame by frame with the button next to the pause button.



***Press the ‘Play’ button. Notice that the game view comes into focus. Notice that nothing interesting happens...***

Currently, nothing interesting happens when we play the game. This is because we haven’t yet told the game objects to do anything.

|  |
| --- |
| *NB:* You can make changes to the game while it is playing. **These changes will be lost when the game is stopped.** |

### Cameras

In unity cameras are objects that can be added to the game to show different perspectives of game play. This can be the same as a movie that cuts between different cameras OR like a picture in picture view. The game can have any number of cameras in it each providing a different view.

The game view and the scene view are viewed through different cameras. The game view is viewed through a camera that exists as a manipulatable game object in our game (Unity put it there for us); whereas, the scene view is viewed from a special ‘behind-the-scenes’ camera that **does not** exist in our game.

***Manipulate the ‘Main Camera’s position or rotation. Notice that the game view changes.***

|  |
| --- |
| *NB: The scene view camera is not listed in the ‘Hierarchy’ window because it is not a manipulatable object; likewise, the behaviour of the scene camera cannot be affected through scripting.* |

# Inspector

The ‘Inspector’ is where information about our currently selected game object is displayed. The inspector panel lists all the components attached to our game object and their current state.

### GameObjects and Components

A game object is a container for components. A game object consists of 1 or more components; each of which is like a unit of functionality. It is the components attached to a game object that give it its functionality.

*A renderer component allows an object to draw itself*

*A collider component allows an object to collide and be collided with*

*A rigidbody component allows an object to be affected by physics*



***Turn off the renderer component on the cube game object (see above image). Notice that the cube becomes invisible.***

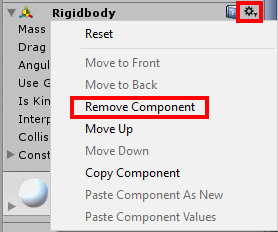
***Click the tick box next to the component’s name***

Components can be added and removed from game objects as you please, adding and removing functionality from the given game object.

***Add a Rigidbody component to the cube object:***

***Add Component > Physics > Rigidbody***

***Play the game. Notice that the cube falls.***



***Remove the Rigidbody component from the cube object (see above image):***

***Click the cog icon in the top right of the Rigidbody component > Remove Component***

|  |
| --- |
| *NB: A transform component incorporates a game objects position, rotation, and scale properties. It the one compulsory component that all game objects must have; it cannot be removed. When adding a game object in the editor, a transform component is automatically added.* |

### Scripts

Scripts are user made components. Scripts follow the same rules as the provided unity components; they effect the behaviour of the game object they are attached to. Scripts can have properties listed in the ‘Inspector’ window just like the Unity components we have seen. We will see how to write scripts in the ‘Falling cubes game’ portion of this tutorial.

# Project View

The project view is the view of all of our game assets (not only those in the current scene). These are all the scripts, prefabs, 3d models, sprites, etc saved on our system in our project’s asset folder.

### Prefabs

A game object can be saved as a prefab. A prefab is a preconfigured game object; it saves all the components and their state associated with our object. When instantiated in the game a prefab creates an identical copy of that preconfigured game object. This is useful for cloning objects in our game. In fact, the cubes and other objects we have been adding to our scene are themselves just prefabs.

We will see how to instantiate a prefab during the ‘Falling cubes game’ portion of this tutorial.