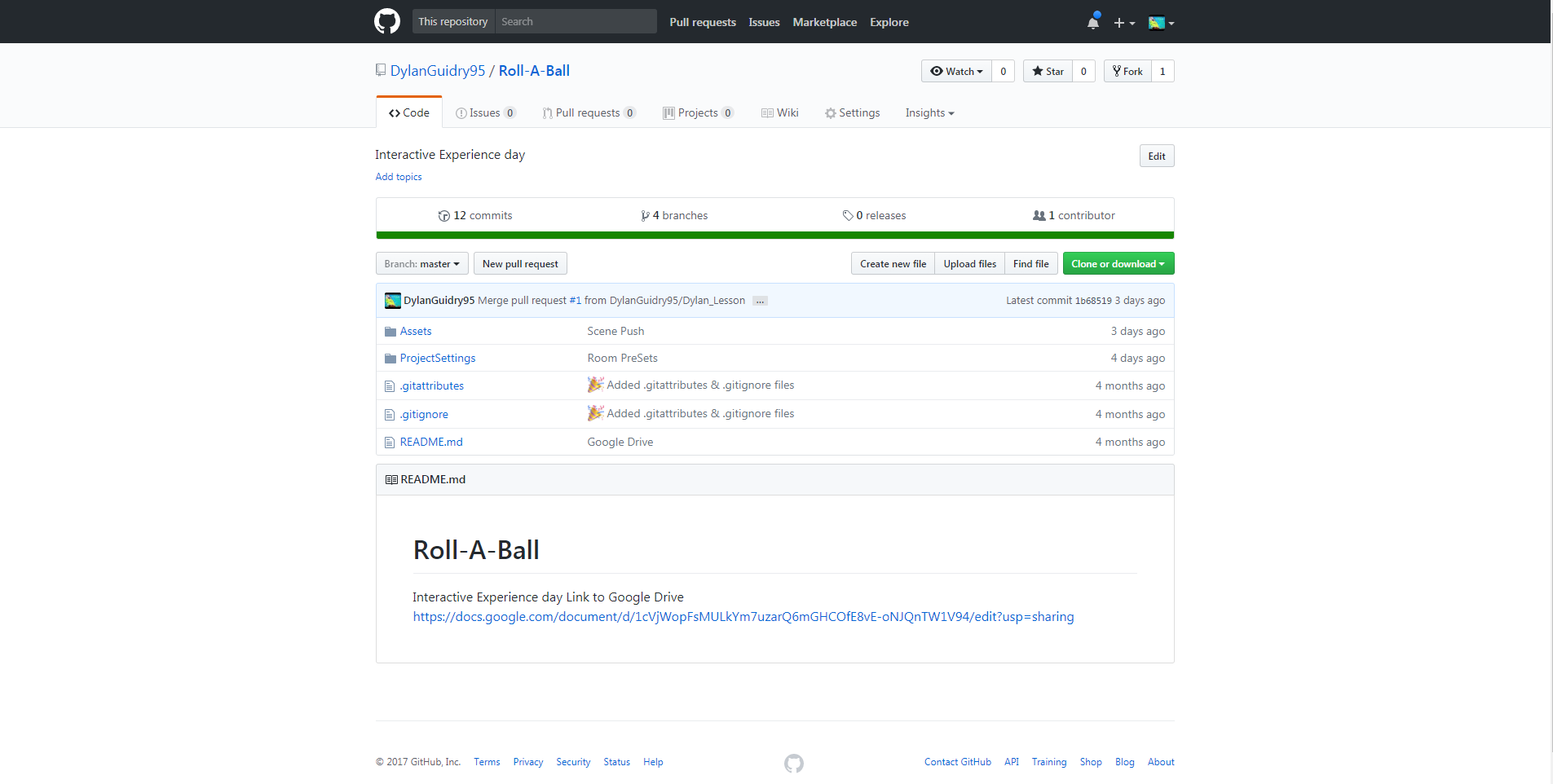
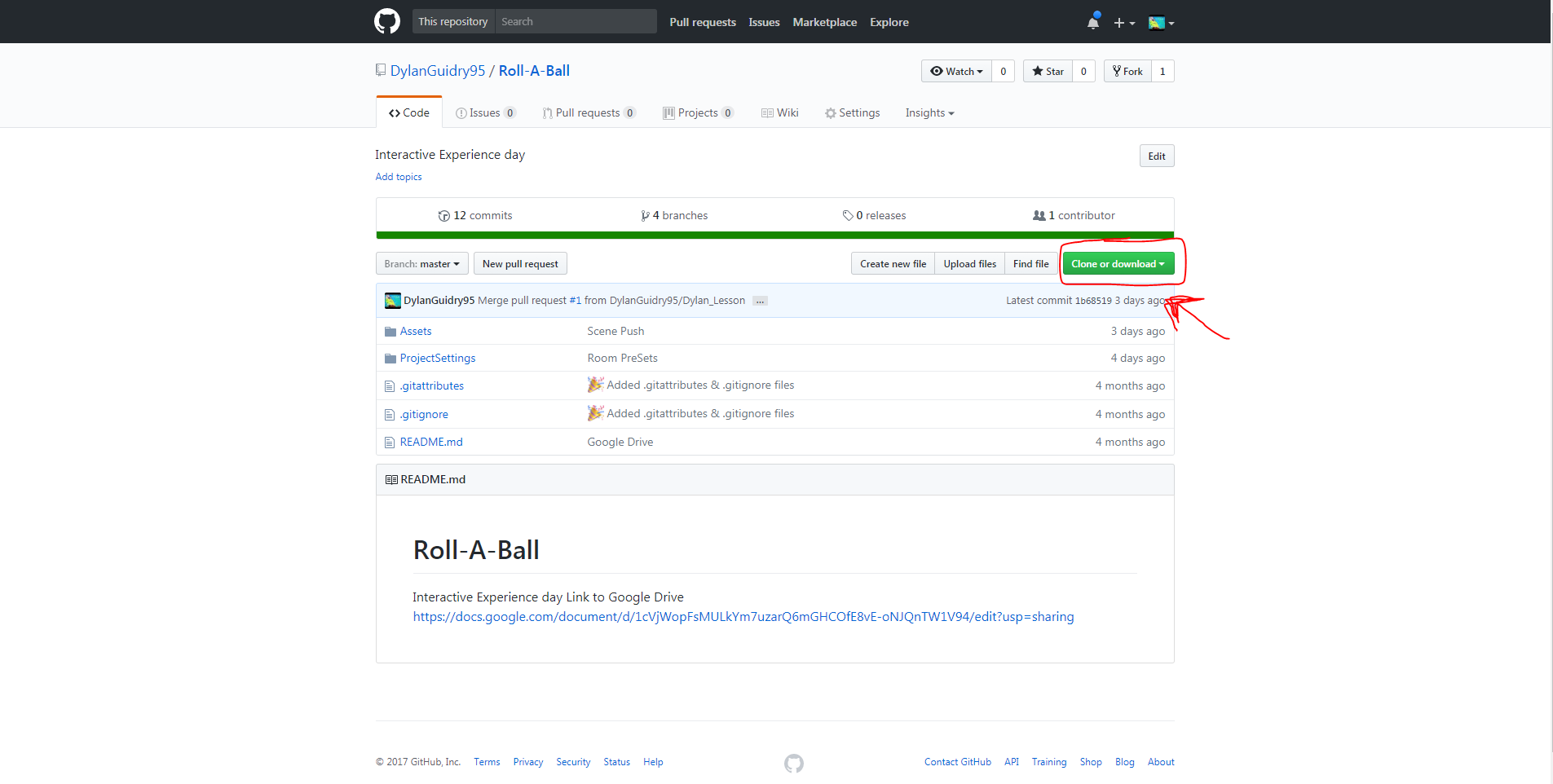
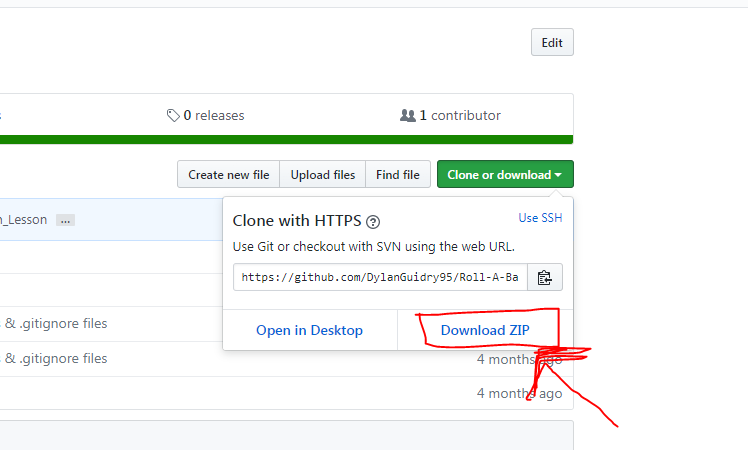
AIE Interactive Experience Day

Roll-A-Ball

**Software Needed**

* Unity 2017.1.0f3 or newer
  + **Download:** <https://store.unity.com/download?ref=personal>
* Visual Studio 2015 or newer
  + **Download:** <https://www.visualstudio.com/thank-you-downloading-visual-studio/?sku=Community&rel=15>

**Getting Started**

* Before we can begin you will need to download a project from the website GitHub.
  + (insert note about github here)
* Open up your internet browser and type in <https://github.com/DylanGuidry95/Roll-A-Ball>
  + 
* Once on that page click the green Clone or Download button.
  + 
* A Dropdown menu will appear, once it does click Download Zip
  + 
* Once the file is downloaded extract the contents of the .zip files to your desktop.

**What is unity?**

* Unity 3D is a software development tool, primarily used for Game Development.
  + Games made in Unity
    - Hearthstone
    - Kerbal Space Program
    - Rust
    - Ori and the Blind Forest
    - Cuphead
* Unity is our primary game development tool here at AIE right next to Visual Studio

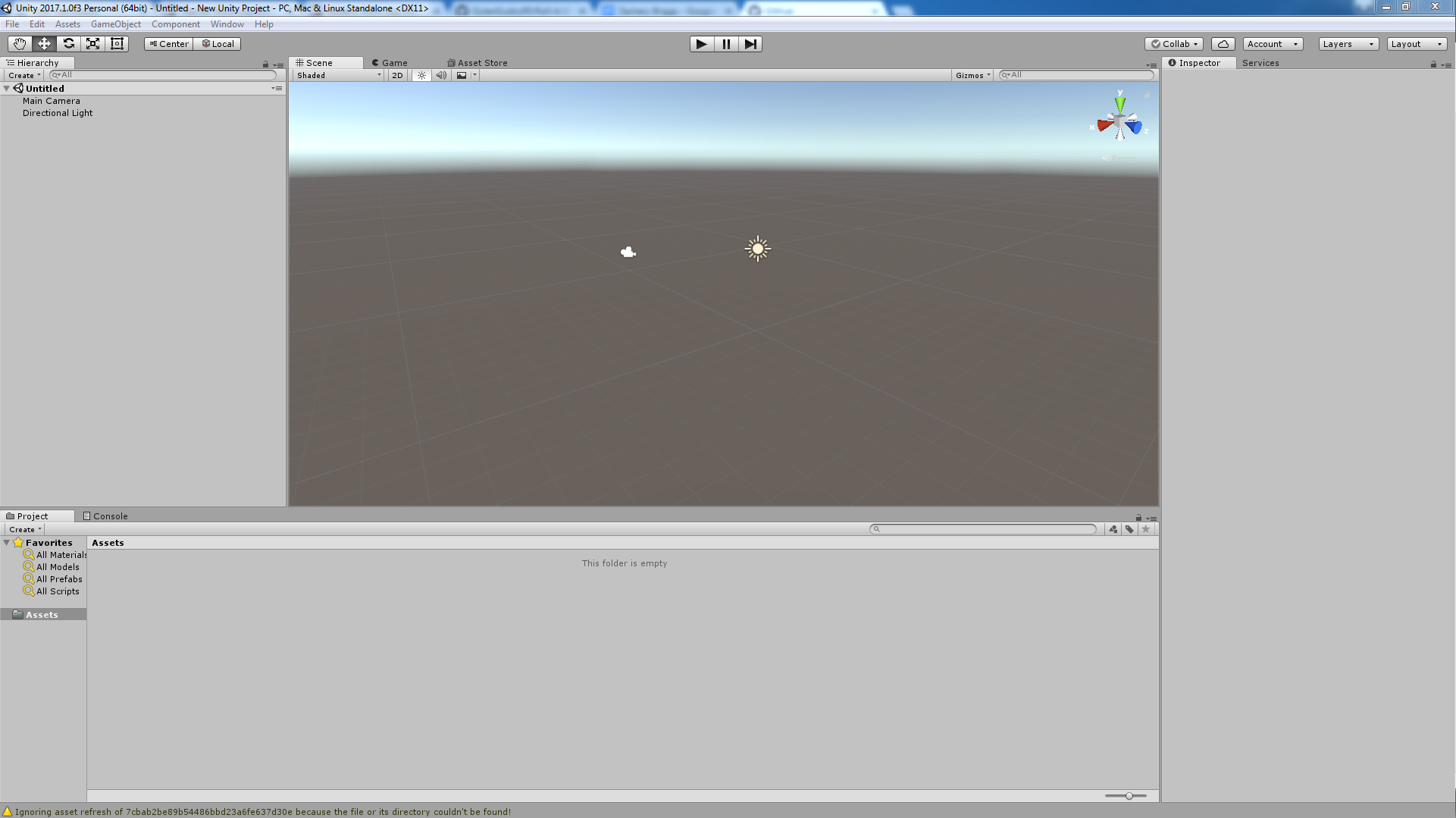
**What is Visual Studio**

* Visual Studio is an IDE (Intergraded Development Environment)
* Visual Studio is the most commonly used IDE in the software development industry because of all the tools is offers.
  + It allows us to Debug code as we work
  + Compile code whenever we want (assuming there are no errors)
  + Build an application

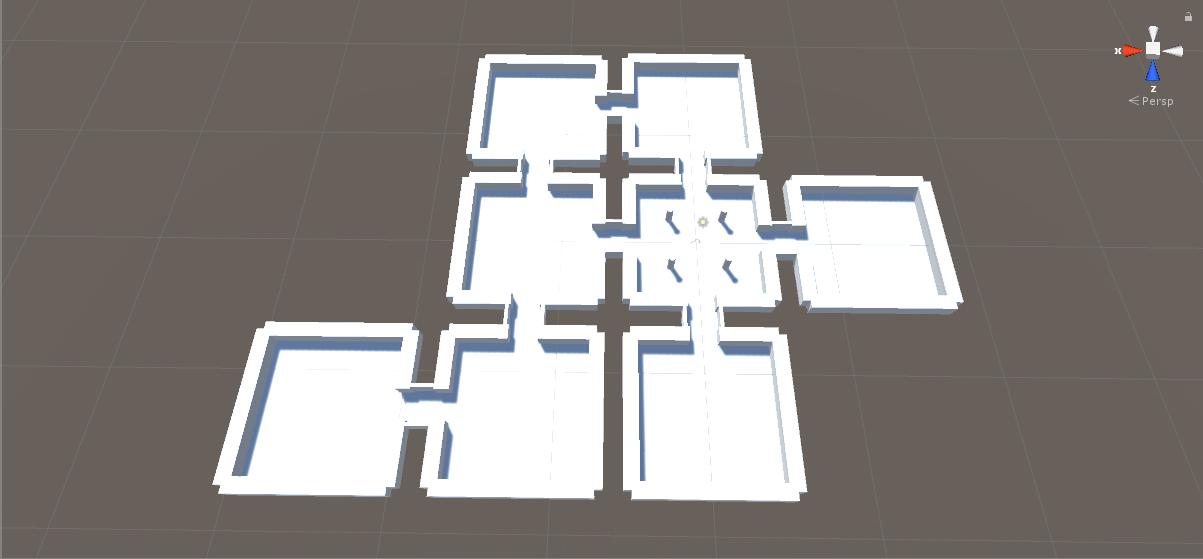
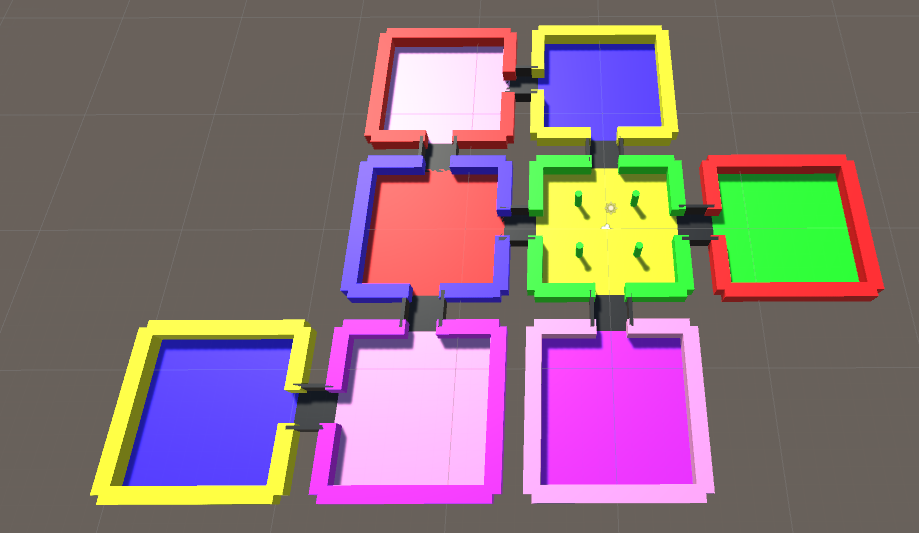
**What is Roll-A-Ball?**

* Roll-a-ball is an intro level project for new programmers or programmers who are just trying to get familiar with the features Unity has to offer.
  + You can find Unity’s official tutorial here <https://unity3d.com/learn/tutorials/projects/roll-ball-tutorial>
* Today we will be doing a hybrid of the Unity Tutorial with an AIE twist.

**Unity Window**

* At first glance the Unity window maybe intimidating with all the windows it has inside but don’t worry we will go over them shortly.
  + 
* Hierarchy
  + Located in the TopLeft
  + Displays the names of all object currently in the scene.
* Scene
  + Located in the middle
  + Allows us to move objects in the scene and place them where we want them. Think of it as your canvas
* Game
  + Located in the middle (Select the Game tab on top the scene window)
  + Will become the active window when we run our game for testing.
* Inspector
  + Located in the right
  + Displays all the components and values associated with the currently selected object.
* Project
  + Located in the bottom
  + Displays all assets currently in our project (Scripts, models, textures, etc.

**Let’s Make a Game**

* So the project I made you download from GitHub has a bunch of premade assets for you to use.
  + Models
    - Rooms
      * 0\_Door\_Room
      * 1\_Door\_Room
      * 2\_Door\_Room
      * 2\_Door\_Room\_Corner
      * 3\_Door\_Room
      * 4\_Door\_Room
      * 4\_Door\_Room\_Pilliar
      * Room\_Connector
    - Players
      * Boxy
      * Rollie
    - Pickups
      * Coin
      * Diamond
  + Materials
    - Black
    - Blue
    - Green
    - Pink
    - Purple
    - Red
    - White
    - Yellow
  + We will use these assets to make a dungeon for our game. (Follow along on with instructor)
  + Placing the rooms
    - Ctrl + V to snap objects together
    - 
  + Once you have your rooms all set up you can use the materials to color them.
    - Simply drag the color you want onto the object.
    - 
* Once we have our dungeon made let’s get a player rolling around in there
  + Go into the Models/Players folder and select which model you want to represent your player.
  + Once you have selected it selected drag it and place it in the scene where you would want to spawn.
    - You may need to raise the model 0.5 unities in the y so that it is not sitting in the floor.
  + Now that we have our player in we need to set up the camera. We all ways want our camera to be following our player for this example.
    - To do that we can parent the camera object to our player object by dragging it onto it in the hierarchy.
    - Once we have it parented we can move the camera to there we want it when the game runs.
    - Use the camera preview window to see a preview of what the camera will look like in game view
      * Use the Q,W,E,R keys to cycle the different types of transform values you can manipulate.
        + Q

Hides the transform in scene view

* + - * + W

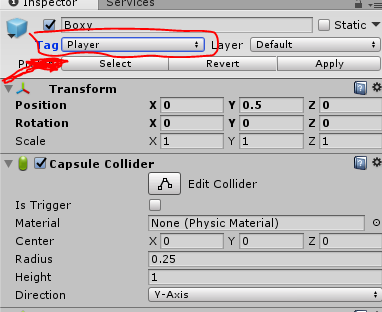
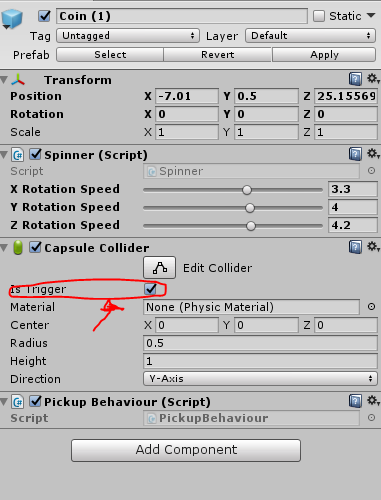
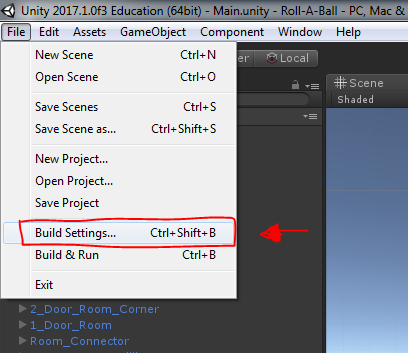
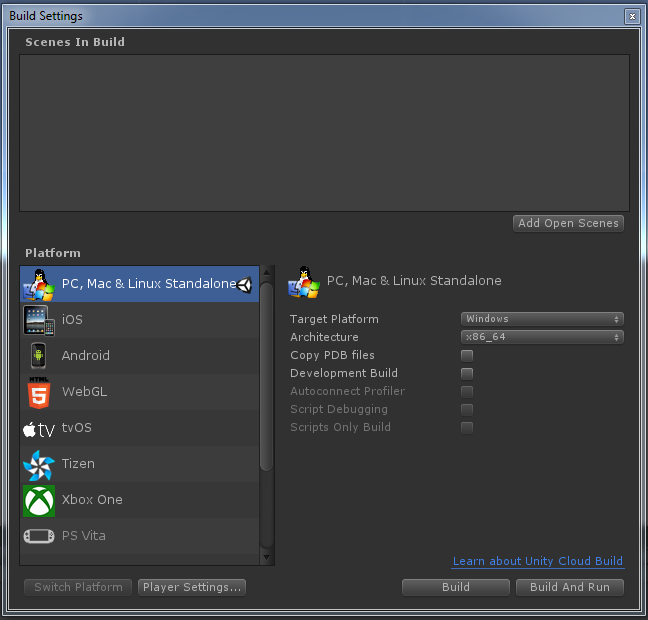
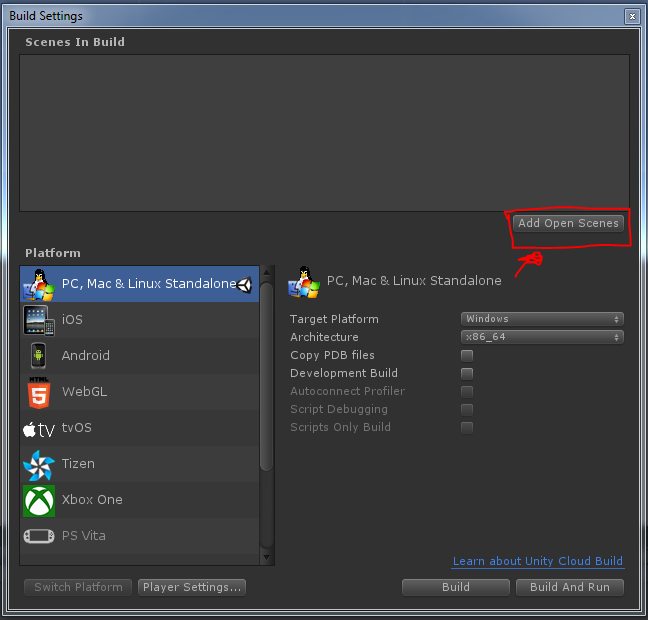
Changes the position

* + - * + E

Change the rotation

* + - * + R

Scale

* + Play around with the transforms of some of you object in your scene.
    - You can all ways press Ctrl + Z to undo your changes
  + Now we can give our game a run
    - You will notice you cannot move so let’s do that next.
* Player Movement
  + In the project we have provided you with a script for player movement called TankMovement
    - Open the script in visual studio by double clicking on it. The green text or comments is what we call them will explain what is going on.
    - This script mimics the movement of a tank where it can only move forward or backwards on one axis. It must rotate on its Y axis in order to move in a different direction.
    - Movement Controls
      * W
        + Forward
      * S
        + Backward
      * A
        + Left
      * D
        + Right
  + Attaching a script to an object
    - Select your player object in the hierarchy. Once you have it selected you should see all the components attached to the object appear in the inspector.
    - Now there are two ways can add the script to the object.
      * You can click the add component button in the window and search for the script you want to add.
      * Drag the script from the project window into the inspector
  + Once you have the script attached you can see all the values that we can manipulate from the inspector.
    - Rotate Speed
      * Controls the speed at which the object rotates on the Y axis, has a cap of 10
    - Movement Speed
      * Controls how fast the object moves on its Z axis, has a cap of 20
  + Go ahead and give your game a run and modify the values of Rotate Speed and Movement Speed to your liking and move around the map you made.
* Pickups
  + The main goal of Roll-A-Ball is collect some sort of pickup items and each one you pick up increases your score by some amount and once you pick up all the coins or your reach some score you win.
  + In the project you have been provided with there are two pickup objects you can use one is a diamond and one is a coin. Drag them into the scene and place as many as you like.
  + Once you have them in the scene you can color them.
  + So these pickups look boring let’s add some flare to them by making them move in place.
    - Look in the scripts folder for the Spinner script
    - Open it and follow along with instructor
  + Once we have the script you can add it to your coin objects in the scene.
  + When you add the script to an object you will notice 3 sliders these sliders will control how fast the object moves on each axis. If the value is negative the object will rotate counter clockwise.
    - Modify the values for each coin and see how the behave
  + Now that we have our coins looking energetic we need to be able to pick them up
  + Look for the PickUp script in our scripts folder and open it
    - Follow along with instructor
  + Once we have our script completed place it on all the pickup objects in the scene
  + You will notice that this script has no modifiable values in the inspector this is because we private all class variables in the script because we don’t want to be able modify them in the inspector.
  + Now you can run your application and move the player over the coin.
    - You may not be able to pick them up you will need to ensure that your player is tagged Player and the collider on your pickup are set to IsTrigger
    - Tag
      * 
      * If it is not set to Player hit the drop down and select it
    - IsTrigger
      * 
      * Make sure the box is checked
    - Now you can give the game another run and you should see the pickups disappear when you collide with them.
* Score Keeper
  + In the pickup script you noticed we made a variable of type ScoreKeeper.
  + That type another class that we have created that we will look at next
  + Look in your scripts folder and open the ScoreKeeper script
    - Follow along with your instructor
  + Before we can put the ScoreKeeper on an object we need to set up some UI elements to display our score to our player.
  + Inside the Models folder there is a UI folder there is a ScoreKeeperUI drag that into your scene.
  + Once you have it in your scene put your ScoreKeeper script on it and follow along with our instructor.
    - You will need to assign a value for the Score Display and Game Over Display variables
* Let’s Play
  + Once you have your ScoreKeeper in place run your game and try to get all your pickups once you do you should be prompted with Game Over on the screen.
* Building The Game
  + So as we all know great games aren’t played in Unity they are launched from your desktop or steam.
  + So let’s get us an executable built so we can take our game to our friends and family house and show them without having to download unity.
  + First click file at the top of the screen and select Build Settings
    - 
  + You will see a screen that looks like this
    - 
  + Hit the add open scenes button to add your scene to the build
    - 
  + Click the build button and save it in a new folder on your desktop
  + Once it is saved click the executable and you can now play your game on and Mac, PC, or Linux machine.