Tyrel Chance

My tutorials

**Tutorial 1**

Tutorial will be teaching you how to move your player around using the arrow keys. You will need unity 2019 or higher and Visual Studios for coding.

|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated A screenshot of a computer  Description automatically generated | **Step 1.**  Before moving to the coding, you need to firstly have objects for code to control. I firstly imported a plane and a cube the plane acts as the ground for the cube making sure it doesn’t fall through, the cube is what we will be putting the moving code on.  The plane is found GameObject – 3D Object – Plane (and so is the cube) |
| Graphical user interface  Description automatically generated | **Step 2.**  I then imported 2 materials this is to add colour to the objects so I don’t get confused on what is happening.  Left click in assets box (bottom of the screen) – create – material |
| A screen shot of a computer  Description automatically generated | **Step 3.**  I then add the colour to the objects the colours can be found on the right-hand side of the software. You will then pick the colour you want and drag it on which object you be that colour. |
| Graphical user interface, website  Description automatically generated | **Step 4.**  Now we move onto the coding. We firstly have to create a C# script this can be done the same way we imported the materials.  Left click in project – create – C# script |
| Graphical user interface, application, Word  Description automatically generated | **Step 5.**  This will then take you into Visual studios this is where we write out the code to make the cube move. We are going to focus on the update section of the code which is the bottom section this is because we want this moving code to constantly happen throughout the game and not cut out at one point.  Copy the code in the image this is telling unity what way the cube will be going in this case it is left and right (A and d on the keyboard). |
| Graphical user interface, application, Word  Description automatically generated | **Step 6.**  The next code does the maths for the previous bit of code. Controlling the speed of the code when you press a to go left and d to go right.  This will also make sure that the animation doesn’t overwork your computer it will adjust to the speed of your computer. Copy it carefully one wrong mistake can stop the movement from working and not play your game until fixed. |
| A screen shot of a computer  Description automatically generated | **Step 7.**  Save the code and head back into Unity this is because we are going to test to see if it works.    press the play button in the tool bar and this will take you into game mode showing you what the player would see.  Then press the left and right arrow keys and the cube should move side to side. If this does happen got back to the code and look for errors (errors are highlighted in red). |
| A screen shot of a computer  Description automatically generated | This is an after image of what the cube should do if done correctly. You can see it is no longer in the middle of the screen. |
| Graphical user interface, application, Word  Description automatically generated | **Step 9.**  Now we got it moving side to side we want to create a code that now makes it move forwards and backwards as well. Go back to the code and look at the section that says (Input.GetAxis("Horizontal") \* Time.deltaTime, 0f, 0f) this code is representing X axis so far we now need one for the z axis so it can move up and down. So copy that horizontal code and change the word to vertical. |
| Graphical user interface  Description automatically generated | **Step 10.**  I if done correctly then you cube should be able to move to the end of the plane. |
| A picture containing graphical user interface  Description automatically generated | **Step 11.**  As you can see the cubes moved to the edge of the plane. |

**2nd Tutorial**

In this tutorial I will be teaching you how to make a 3D person camera which you could use in Games. I will be teaching you how to create the movement for the character and then how to combine it with the camera.

|  |  |
| --- | --- |
| A screen shot of a computer  Description automatically generated | **Step 1.**  We are going to be creating a 3rd person camera but first we will need objects for the camera to follow. Start off by importing a Plane and cube, you can do this by going to GameObject– 3D Object – cube. The plane can be found in the same place. |
| A screen shot of a computer  Description automatically generated | **Step 2.**  I will then be adding different colours to the object, so I don’t get myself confused to do this left hand click in the asset box scroll to create and then scroll down to Material. You should see a sphere appear in your assets folder, click on it, in the inspector you will see a white box. This is representing the colour, click on the box and choose the colour you want then drag the sphere to the object you want to be that colour to be. |
| A screenshot of a computer  Description automatically generated | **Step 3.**  I then want you to add another cube to your scene. This time minimize it by press R and use the x, y and z axis arrows, once done press w and place it at the top edge of the cube (still using the arrows given). This is so we can see the direction the cube is facing. |
| A screenshot of a computer screen  Description automatically generated | **Step 4.**  We will now have to start labelling the cubes. label the large cube “Player” and the small one “Direction” in the hierarchy (top left of the screen) . Then drag the direction cube in the player cube. This will make it a child attaching it to the main cube, so if I click the Player cube it will automatically select the Direction cube as well. |
| A screenshot of a computer screen  Description automatically generated | **Step 5.**  I then want you to make an empty game object and drag it inside the cube. Once that is done label it targets and drag it to the Player label making it a child. There should be an arrow on the left-hand side of the Player label click on it and it will show you the Direction and Target are inside, leave it open and drag the main camera into the target label. This will snap the camera to the target making it focus on the cube as well. |
| A screenshot of a computer screen  Description automatically generated | **Step 6.**  Now it is time to start the coding go to your assets left click scroll up to create and click on C# Script. I want you to create 2, one called “ThirdPersonCharacter” and the other “ThirdPersonPlayer”. Remember no spaces or numbers write it exactly how I have. |
| A screenshot of a computer screen  Description automatically generated | **Step 7.**  Next click on the ThirdPersonCharacter script this will bring you in to visual studios once in copy this code to the left, what this code will do will allow you to move the object around using the keys on your keyboard the time.DeltaTime section allows your game to run Smoove on any computer no matter the speed of strength of it. |
|  | **Step 8.**  Go unity and press on the play button this will put your scene into play mode. Use the arrow on your keyboard to move the cube around. If it doesn’t work, go back into your code and look for areas highlighted red as they are seen as issues in visual studios. Issues could be that there isn’t capital letter when you are writing “Mouse X & Mouse Y” in your code, This is very important if you don’t the code will not work. |
| A screenshot of a computer screen  Description automatically generated | **Step 9.**  We will now move on to the Camera code. I want you to follow the code to the left, this code will allow you to move the camera when you move around your mouse. A lot of this code is copy and paste as sometime the same code they just need a few adjustments to do a different action for example move left, right, up and down.  Now test your camera in unity but pressing the play button and moving the mouse around. |

**Tutorial 3**

In this tutorial I will be teaching you how to create and enemy AI. I will be showing you how to make it patrol and area, chase an enemy and attack an enemy when close.

|  |  |
| --- | --- |
| A screenshot of a computer screen  Description automatically generated | **Step 1.**  Firstly, I am going to create a C# script by left clicking scroll to create and click on C# Script. Once created click on it and it will open up Visual Studios. |
| Graphical user interface, text, application  Description automatically generated | **Step 2.**  I then what you to erase the public update and the void start as you will not need them at this point. Then repeat what’s on the script to the left this is going to be the start point to making your enemy move around without the need of controlling them. |
| Text  Description automatically generated | **Step 3.**  In the same script underneath what you just wrote copy this section. This is the attacking section of your script this will calculate the number of times the enemy has attacked. |
| Text  Description automatically generated | **Step 4.**  This section of the code is how far the enemy will be able to see from and when the player is in range of the enemy it will attack them. Write this below the attacking code. |
| Text  Description automatically generated | **Step 5.**  This code is a private code meaning that it still does something, but you won’t be able to see it In Unity. This is explaining to the game that this code is for more than one enemy so automatically connect this code to anything with the name “Enemy”. |
| Text, letter  Description automatically generated | **Step 6.**  These are if statements list out ever scenario that the enemy could do for unity. For Example, if the player is insight of the enemy the enemy will chase the player. Copy this code underneath the last section of code we wrote. |
| Graphical user interface, text, application, Word, email  Description automatically generated | **Step 7.**  The code here is what is going to allow the enemy to search for the player and attack the player when within range copy it underneath the last section of code. |
| Graphical user interface, text, application  Description automatically generatedText  Description automatically generated | **Step 8.**  This is the final section for coding the attacking for the enemy. This is so the attack resets after every attack we are also going to add in projectiles for the enemies so after copying this code go back up to the attacking code and add a “public GameObject” and call it “projectile;” “ this will make a tool for the projectile where you can add in your own object for it. |
| Graphical user interface, text, application, email  Description automatically generated | **Step 9.**  Add this code at the bottom of your script this creates a health for the enemies and allows the projectiles to give damage to the player. |
| A screenshot of a computer  Description automatically generatedA screenshot of a video game  Description automatically generated | **Step 10.**  Now go back to unity. We are going to now create a Nav map of the enemy. This just lets the enemy know how big the map is and how far they can travel. To do this go to windows AI and navigation. A table will appear on the right-hand side of your tool bar, click on bake and then bake again this will create the nav map for you. |
| A screenshot of a video game  Description automatically generated | **Step 11.**  Go to the enemy that you have created and on the right-hand side of the screen scroll down to add component. Once you are there type in “Nav mesh agent” this is going to help the enemy move around the game environment. |
| Graphical user interface, text, application  Description automatically generated | **Step 12.**  Adjust the settings on the agent to how you would like and then drag and drop your script you made into the enemies’ tool bar. Adjust the setting to what you want for example who the enemy is chasing. |
| A screenshot of a computer screen  Description automatically generated | **Step 13.**  Once that is all done press play and the enemy should be chasing and patrolling the environment. |

**Tutorial 4**

In this tutorial we will be learning how to program bullets and programming how to fire them in games. You will ned Unity that has the 2019.4 update or higher. You will also need visual studios to create the code.

|  |  |
| --- | --- |
| Graphical user interface  Description automatically generated | **Step 1.**  First off open up unity and go to the tool bar at the bottom that says “Assets”. Then left click, scroll to create and click on C# Script. Once that is done click on the script to open it up. |
| Graphical user interface, text, application  Description automatically generated | **Step 2.**  Once you’ve clicked the script it should open up visual studios for you. If it doesn’t then simply go to unity’s preferences in the top left corner and scroll down to external tools and check to see if Visual Studios has a tick next to it. If it doesn’t just click on the tick and it should now work.  Now that you are in visual studios copy this exactly as this is going to make the setting section for your weapon, meaning you will be able to adjust the speed of you gun, the force and much more. |
| A picture containing graphical user interface, text, application  Description automatically generated | **Step 3.**  Write this underneath the code we have already made as this is still adding the adjustable setting to unity the “//” basically gives you a short explanation of what they do not counting it as code.  The bottom piece of code is just to sort out any bugs or issues that come along the way. |
| Text  Description automatically generated | **Step 4.**  This section of the code will make sure that your magazine is full when you reload the weapon. Put this underneath the last code you wrote out. |
| Graphical user interface, text, application  Description automatically generated | **Step 5.**  This section of the code will allow you to hold down the left section of your mouse when you are firing your weapon.  However underneath the if statement put the else statement that you see to the left underneath if. This will allow the user to have an option of whether to hold or tap the mouse when firing. Remember there are weapons that only shot one bullet at a time. |
| Graphical user interface, text, application, email  Description automatically generated | **Step 6.**  This is our shooting variable and showing us what will happen when the player runs out of bullets. We are trying to make it automatically reload when the magazine hits 0. So we need to make sure in that reload period no bullets can be shot. Copy this underneath the last bit of code. |
| Graphical user interface, text, application, email  Description automatically generated | **Step 7.**  In between the “readyToShoot” and “bulletsLeft” section I want you to type in this ray section of the script. This help unity know when you shoot at a target or if your shooting in the, it also creates a spread for the bullets if you want it. If you don’t delete the float x and y section. |
| Graphical user interface, text, application, email  Description automatically generated | **Step 8.**  I want you to add the rest of this code in between as this controls the shot direction and the force of the bullets. So for example how hard they hit do they push the enemies back? |
| A picture containing text  Description automatically generated | **Step 9.**  Underneath the force code you just created copy this section of code this controls the reloading of the gun allowing you to control the time between each shot and time it takes to reload and the magazine size. |
|  | **Step 10.**  This section of code I want you to scroll back up to the“//shooting“ section and write this code above it. This code does allows you to press the “R” button to reload and if you try to shoot when there is no ammo it automatically reloads the gun. |
| Graphical user interface, text  Description automatically generated | **Step 11.**  This section has to do with the graphics that appear when you fire and making sure it doesn’t crash the game. |
| Diagram  Description automatically generated | **Step 12.**  Copy out the section circled as this does the calculation for the ammo and how it gets rid of the bullet number after one has been fired. |
| Graphical user interface, text, application, email  Description automatically generated | **Step 13.**  Copy this code as it is for the muzzle flash make sure it’s in your section. |
| Text  Description automatically generated | **Step 14.**  We have now finished all of the code so go back to Unity and start creating the assets that you are going to use for this code. To the left you will see my layout I have done for my level:  PlayerObj - capsule  Fps - Camera  Cylinder is the weapon  Attack point – empty game object |
| Graphical user interface  Description automatically generated | **Step 15.**  I then want you to click on you player object and drag and drop your script into the preferences of the object. Make your setting how you want them for your game. |
| A picture containing outdoor, flying, plane, water  Description automatically generated | **Step 16.**  Now when you press play you should see the bullets flying out of the weapon. |

**Reflection**

I began to see as I went completed each tutorial the number of issues I encountered began to decrease. This is because all these tutorials are linked in some way which meant that you would start to see the same errors if you made a mistake. Because of that I knew how to handle and fix the error because of past encounters.

**References**

**Walk code & 3rd Person camera -**<https://www.youtube.com/watch?v=7nxpDwnU0uU&t=165s>

**Moving AI -** <https://www.youtube.com/watch?v=UjkSFoLxesw&t=1s>

**Creating a weapon -** <https://www.youtube.com/watch?v=wZ2UUOC17AY>