

Unit 7



Diogo Constantino

We load the game file with the road, and the teacher tells use in the scene view if we hole right-click then pressing WASD we can "fly", this is useful to move around the map.

In this lesson i was told that naming convetions are VERY important because if i want to code something, unity will try to find that name and if the name in the code is different in unity then the code won't work. Example:if i write Vehicle for my car in unity but i write vehicle without the capitalization, the code in C# won't work because the it can't find the name, that's why naming convetions are very important.

I learned that you can reset the position of an object on the transform(which all objects have) which is very useful if i want to the object to go back to it's original position.

The teacher told us that CTRL+P plays the game in unity, i already knew this but now i'm using it more often since it's very helpful by saving time for my project, a few seconds might not seem a lot but it does help in the long run. The layout on the upper right corner changes how unity looks, basically it adds another scene view, puts the hierarchy to the right and more depending on which layout you choose. This may not seem important but it helps a lot, example: if i am doing a FPS game i might need a scene view with the camera and another with the surroundings, it may be a specific thing to person to person, but it does help and it's nice to have.

I start by putting a red car on the road aligned in the middle and rename it "Vehicle", i put the camera behind the car. Next i put a create on the road and name it Object, then duplicate it 5 times. I adjust the objects weight/mass so that the crates move after the car hit it.

I test the cars and creates mass,at first the car goes flying in the sky,the reason this happened, i think is because the mass of the car was low, so i increased it and tested and now the creates were being sunken to the ground instead of being pushed to the side which is what i want. I do some more changes to the mass, it takes some time but i get the results i want.



My game in the beginning of the tutorial.

Lesson 1.2 - Pedal to the Metal

Tutorial • Beginner • 1 Hour 10 Mins • 13

Your progress

Where am I?

1. Introduction
2. Create and apply your first script
3. Add a comment in the `Update()` method
4. Give the vehicle a forward motion

Topics we'll cover

- For Educators
- Scripting
- Editor Essentials

The unity site i used to get help and reminders, if i forgot something.

On 13/10/19 i restarted the create with code project because i think i made a error by creating the project, by making unity 2019.1.0 instead of 2019.2.3, so that's why i made a clean plate.

I mostly was using the unity site to guide and help me make my game. It helped me a lot but there were some sections that were confusing, especially at the coding parts, at one point i did a code for the car to move, but it didn't work, by that i mean the car won't go forward. I think it must have been the position a putted the code in, i putted above void star instead void update.

I kept following the tutorial to make my game, i added a transform.rotate script for my car to actually rotate, then after that all i had to do was clean my code, i made all the public float privates and added some comments/notes to make my code understandable and since the turnspeed didn't have a value a i added one, i putted 45.0f since it's a good speed to have on turnspeed. I tested my game and it all went fine, the only problem i could fine was that my car was still "lagging", not driving very smoothly.

I learned that quotation marks ("") are used to create string literals in code.

This project was quite challenging but in a good way, i learned a lot about code.



My final game

What i will be working most of the time in terms of coding is C#, so i will do some extra study by reading and watching videos about C# to make me smarter as a game developer and programmer. I learned what Void start and update are, Void start basically only happens once

and is called before the first frame update. Void update is constantly checking the code and is called once per frame, it also starts after the Void starts. The teacher mentioned Void awake but didn't go into detail, so i did some research to get some small info about it, i went to a site about the difference in Void start and awake to get some info (<https://forum.unity.com/threads/what-is-the-difference-between-void-start-and-void-awake.139557/>) and basically Void awake starts when the MonoBehaviour is created. Essentially Void awake, "awakes the program", Void start "starts the program" and Void update continually updates the program. Void awake and start are the same with some differences such as Void awake is called when the script is enabled.

In code translate is basically another word of "more here" more stuff for my code i'm doing.

Public Float: is a code that will let anyone use it not only me and make it easier to change variables.

Privet Float: Only i can use and change this script/code.

The (//) are used as note for me and other people using the script with code.

The () key is used to end the code.

I create two code script in the script folder in unity,i create "FollowPlayer" and "PlayerController", the naming of these script are very important.

FollowPlayer script is for the camera to follow my car, i add the transform.position = player.transform.position; tested it and the camera was underneath the car so i add an offset to change it's position with: + new Vector3(0, 5, -7) this made the camera higher than the car. Then i made the vector3 a privet vector3 offset, so the new vecto3(0, 5, -7) = offset. I need the new vector2() because the 3 numbers could mean anything.

PlayerController script will let me control the vehicle and adjust it's speed and possibly more. On this script i adjusted the speed of my car by using Time.deltaTime * 20 this will slow down my car, i changed the number in the code and tested it to see how it affects the speed, because of this i found out that with a higher number the car doesn't go as slow but, if i use a lower number e.g 5, the car will go very VERY slower, this is good to know just in case.

The code i used was: transform.Translate(Vector3.forward * Time.deltaTime * speed);

I learned many things because of this project including:

-Start vs Update which i explain in a earlier time in sharepoint.

-C# Scripts: this Script is a special language that Unity can understand(although it understands multiple languages). And, it's through this language that we can talk to the engine and give it our instructions.

-Comments: a comment is used to get some info or reminder in code without affecting it, there are also multiple types of comments including: Multiline comments(which have one or more lines of narrative within a set of comment delimiters), Single-line comments(allow

narrative on only one line at a time, and also is the type we learned at class) and XML documentation comments(which start with a triple slash and are enclosed in XML tags, these can be very useful in keeping documentation up to date).

-Methods: A method is a code block that contains a series of statements. A program causes the statements to be executed by calling the method and specifying any required method arguments. Also method and function are basically the same, Method is the correct term for a function associated with a class.

-Pass parameters:Parameters enable a function to operate in many different ways or on different values without cluttering your code up with endlessly more variables that are used for just one purpose which can get confusing very fast, makes the code clean.

-Time.deltaTime: allows you to move things not by frame but by time. Time.deltaTime is essentially a fragment of a second or the time passed since the last frame. So if you move something with it involved it will move them by time rather than by frame. It can help you make a object move at a constant velocity *speed*, to do that you need to, multiply *speed* by Time.deltaTime and you will get exactly the distance moved by the object since last Update.

-Multiply (*) operator: the (*) is used to represent the multiplication operator. Operators are symbols in a programming language that tells the compiler or interpreter to perform specific operations on operands for producing the final output or result.

-Components: these components are found in the inspector window in a tab that says Components, and they are the Base class for everything attached to GameObjects. It's used to give objects components and "abilitys" for the player to use.

-Collider and RigidBody: These two are components, Collider make a GameObject be able to collide with other GameObjects, without it you the player and other objects will faze through it. Rigidbody will make a GameObject be affected by its surroundings and physics and not make it a visible "ghost".

-transform.Translate: it basically moves the transform in the direction and distance of translation. It make the object movable by "translating" the "transformation" of the game object.

-Vector3: is a Unity C# struct representing points and vectors in 3D space. And depending where after Vector3 example: Vector3.right or Vector3.forward the Vector will face the corresponding local-space of the object, if it's Vector3.right then it will face the right of the object. This structure is used throughout Unity to pass 3D positions and directions around. It also contains functions for doing common vector operations although it's not exactly a position more it's just 3 floats.

-Variables: are used to store information to be referenced and manipulated in a computer program, Variables often store simple data, like integers and literal strings,if you put a variable in a public float it will be viable in the inspector window, not only that but with unity you can change the value of variables while the game is running which is quite useful since it saves you some time.

-Data types: A data type classifies various types of data eg. string, integer, float, etc, the types of accepted values for that data type, operations that can be performed on it, the meaning of the data, and the way that data of that type can be stored, also all variable have

data types. There are mainly two data types: Value and Reference, almost everything in code is a data type.

-Access Modifiers: Access modifiers (or access specifiers) are keywords in object-oriented languages that set the accessibility of classes, methods, and other members. Access modifiers are a specific part of programming language syntax used to facilitate the encapsulation of components.

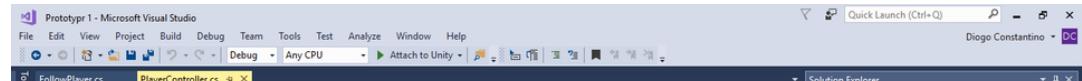
Fun Fact: C# has six Access Modifiers, higher than C++ and Java. the Access Modifiers are, public, protected ,internal, private, protected internal and private protected.

-Declare and initialize variables: For a variable, a definition is a declaration which allocates storage for that variable. Initialization is the specification of the initial value to be stored in an object, which is not necessarily the same as the first time you explicitly assign a value to it. Something happens when your variable is serialized. When you declare a variable public (or you use the SerializeField attribute) the field initializer becomes rather unimportant. It's only used when you create a object in the editor for the first time.

-Empty objects: These are objects that i created in the hierarchy, with these empty objects that i created i can do whatever i want with them or make them whatever i want them to be, it can be a prop, character, scenario, etc. More importantly i can have it have just one component or basic of the basic stuff in them for specific reasons that will help me make my game.

-User input: This code will help me control the car i am using, without it i will not have any control of my object, this code is essential for the basics of player interaction.

-Translate vs rotate: rotate make the vehicle actually turn by rotating the actual car, while translate just makes the car slide to the left or right.



A screenshot of a forum post from forum.unity.com. The post is titled "What is the difference between void Start() and void Awake()?" and has 5,577 posts. The first post, by user "BigBoy" (joined Jan 5, 2012, 91 posts), says: "The difference between Awake and Start is that Start is only called if the script instance is enabled, I never refuse this. But here I say if both the function have same instance which is first executed." The second post, by user "AngryAnt" (Keyboard Operator, Moderator, joined Oct 25, 2005, 3,042 posts), says: "Awake is invoked when the MonoBehaviour is created. You may view it as your default constructor. Start is a message which is sent to MonoBehaviours when all initialisation is done and the first frame for the behaviour is about to run. As mentioned, disabled behaviours do not get the start message as they are indeed not about to run their first frame. So the Start message is sent just before the first Update call on the same behaviour. At this point most dependencies should be fully initialised." The post also mentions: "A key difference between a function invoke which is used for Awake and a message transmission which is used for Start is that the handler for the latter can be turned into a co-routine." The footer of the page includes links to Apps, YouTube, Stick Page - Best O..., Newgrounds.com..., Global Provider of L..., GameFAQs - Video..., Watch Boku no Her..., GOG.com, and Outlook - free pers...

site i used to get extra info

The screenshot shows a Unity tutorial interface. At the top, there's a navigation bar with icons for 'Scripting Tutorials', 'Tutorial • DataTypes', and a 'Start Tutorial' button. Below the bar, the title '1. Data Types' is displayed. To the right of the title is a video thumbnail for 'C# DataTypes in Unity' from 'Beginner Scripting Tutorials'. The thumbnail features a red robot-like character in a futuristic setting. Below the video thumbnail is a code editor window with the title 'DatatypeScript'. The code in the editor is:

```

using UnityEngine;
using System.Collections;

public class DatatypeScript : MonoBehaviour
{
    void Start ()
    {
        //Value type variable
        Vector3 pos = transform.position;
        pos = new Vector3(0, 2, 0);
    }
}

```

tutorial from a site i used to better understand Data Types

Challenge 1 Plane Programming:

I opened Unity, and downloaded the challenge 1 using the direct link, and started doing the tutorials using this link(<https://learn.unity.com/tutorial/challenge-1-steer-a-plane-through-obstacles-in-the-sky?courseId=5cf96c41edbc2a2ca6e8810f&projectId=5caccdfbedbc2a3cef0efe63#5cb5ea6eedbc2a07f08ec24f>).

I went into the PlayerControllerX script so i can improve my plane, first i would make the plane go forward, i did that by changing the Vector3.back to Vector3.forward.

Then i made the plane slow down, i didn't change it in the script of the plane but in unity since the speed code was a public float, i put it at 1 speed for now, i might change it later. I check if the vertical input were correct in the project settings, i was a bit lost, i tried to change, transform.Rotate to transform.translate, but that only made the plane go strait to the right very fast, then i found out by looking at previous projects that i have to actually add verticalInput inside transform.Rotate. I tested my game and it didn't work, but later i realized that i

forgot to save in the script, (SAVING AFTER A CHANGE IN THE SCRIPT IS IMPORTANT, or else nothing will happen), and i was right, then it all worked, but not the way i would have preferred, because if you press the down arrow or S button the plane will go up, i preferred if it went down, and the same thing for the opposite control for the plane to down, you have to press W or the down key.

Next i change the position of the camera and make it follow the plane. I follow the instruction and make the cameras position: X=30, Y=0, Z=10, i thought it would be better if Z=0, because the camera would be line up with the plane, but then i realized that the other way around would be better since the player would be able to move in the beginning and have more time to react.

I made the cameras rotation Y= -90, so it would be facing the right side of the plane.

Then i made the camera follow the plane by putting the Player object in the component of the Follow Player X (Script), like the imaged showed to the right.

But the camera was too close so i had to offset it in the script. I added offset = new Vector3(30, 0, 10) in the code, but i think i added it in the wrong place, i added it in the Void Update, when i tested it a error appeared and i could not enter play mode, but then i realized i forgot to put () in the end of, offset = new Vector3(30, 0, 10), even the smallest detail can ruin a code or program, i'll have to be careful next time, i tested my game and it all worked well.

After that all i did was polished my game, i lowered the speed of the plane to 0.35 and lower the rotation speed to 90 and to Vertical Input 5, i decided not to change the vertical input because i'm use to it now and i don't want to break my game, even though you just have to change some buttons on the project setting but i don't want to risk it.

Then i started trying to make the propeller rotate, i first made a new C# Script called SpinPropellerX, i have to make it spin every frame around the Z axis.

I added the script to the Propeller object that is the child of Player, I googled around found the stuff i wanted and basically copied pasted it, but the code was overly complicated so i took it out of my script, but then i found a simple code that worked perfectly, `transform.Rotate(new Vector3(0, 0, 20) * Time.deltaTime);` although the speed of the propeller was a bit slow, so i increased it to 60 of the Z axis, it was faster but not fast enough for me so i increased it to a point i was satisfied, i ended up with (0, 0, 1000). With that i finished my project, i build my game even though i didn't need to do so but it's good practice, i will probably put it in my one drive so i may send it to my tutor.

This project didn't take too long to do and it was quite useful to me since it gave me practice for my coding skills, i enjoyed this.





Challenge 1 - Plane Programming - Unity Learn

learn.unity.com

Challenge Overview: Use the skills you learned in the driving simulation to fly a plane around obstacles in the sky. You will have to get the user's input from the up and down arrows in order to control the plane's pitch up and down. You will also ha...

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerControllerX : MonoBehaviour
6  {
7      public float speed;
8      public float rotationSpeed;
9      public float verticalInput;
10
11     // Start is called before the first frame update
12     void Start()
13     {
14     }
15
16     // Update is called once per frame
17     void FixedUpdate()
18     {
19         // get the user's vertical input
20         verticalInput = Input.GetAxis("Vertical");
21
22         // move the plane forward at a constant rate
23         transform.Translate(Vector3.forward * speed);
24
25         // tilt the plane up/down based on up/down arrow keys
26         transform.Rotate(Vector3.right * rotationSpeed * verticalInput * Time.deltaTime);
27     }
28 }
29
30
```

my script(PlayerControllerX) in the end of this project

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class FollowPlayerX : MonoBehaviour
6  {
7      public GameObject plane;
8      private Vector3 offset;
9      // Start is called before the first frame update
10     void Start()
11     {
12     }
13
14     // Update is called once per frame
15     void Update()
16     {
17         offset = new Vector3(30, 0, 10);
18         transform.position = plane.transform.position + offset;
19     }
20 }
21
22
```

My script(FollowPLayerX) in the end of the project after doing everything

Add comment

2 Replies · Add your reply

Sort:

 **Best Answer**
Answer by JeffreyD · Feb 16, 2014 at 04:57 PM
I'm using this below to do a simple rotation of a cube.
`transform.Rotate (Vector3.up Time.deltaTime 100, Space.World);`
So maybe it's the UP in Vector3 that you need. There also seems to be too many arguments.
<http://docs.unity3d.com/Documentation/ScriptReference/Transform.Rotate.html>
1 Try...
`transform.Rotate(Vector3.up speed Time.deltaTime, Space.Self);`
OR
`transform.Rotate(0, 0, speed*Time.deltaTime, Space.Self);`
I'm not clear on the use of Space.Self v.s. Space.World. I guess you can play with each to see which give you the effect you want.

[Add comment](#) · [Share](#)

 Answer by Epic_PotatoGuy · Aug 18, 2017 at 04:54 PM
use `transform.Rotate (new Vector3 (0, 0, 20) * time.deltaTime);`

0 Add comment · Share

Related Questions

- How can I rotate a tank's turret back to the starting turret position? 2 Answers
- rotating a tank turret while the tank is rotating 3 Answers
- question about turret on tank 1 Answer
- Lock rotation of object 4 Answers
- Rotating turret floats off tank in a sideways circle. Cannon floats out of turret when rotating up. 0 Answers

Your answer

The site and code that helped me make the propeller spin

LAB 1 - Project Design Document (PDD):

- Personal Projects: personal projects are for me to work on by myself with little to no instructions, it's for me to make whatever i want and use as much creativity as possible. It's around two goals the completeness of my game and how i made my game.
- Design Documents: This is to help me get concepts and ideas in a simplistic way and it also breaks down each component of my game, so i can focus on each aspect. Basically help me think about my personal project.
- Project Timelines: This helps keep me organize and give me steps along the way as i accomplish with the amount of time that i have. I'll have a order of things to do, that will help me make and complete my project, to simply break down all the work i have to do.
- Project Milestones and Backlogs: milestones are to motivate me into completing a certain aspect of my game within a deadline, backlogs are for me to add additional stuff, affects or gameplay for my game that i think would not be essential for my game but would be a nice touch if i have time. Also adding whatever that doesn't build into having a functional game that works, anything that doesn't get cover that gets moved to the backlogs so that i can work on it if i finish a milestone early or if i have free time.
- Minimum Viable Products(MVP): As soon as my game becomes functional and playable, it becomes a MVP, a bare bones of a game but a functional one that i will be happy with. I will also have a sketch of my MVP of what it will look like to help me visualize what my game would be, especially for a lot of the basic mechanics.

The Quiz i did for create with code unit 1.

The quiz was fairly simple and i remembered most of from what the teacher showed us with Kahoot.

100% COMPLETED!!!

Today(30/09/19) we started doing a RPG unity game we learned about how to do a 2D RPG game,at least the basic of how it works.

i learned what is ,the background,Midground,front and more about which priorities have the sprites and if you put them in the background tile palette the player will be in the front.

I found out that if you put a bush close to a animator cloud,the bush will also move with the cloud.

I learned how to do fading and how much and when it can be activated.

I discovered include dialogue and how i can make choices for those dialogue.Also every NPC object has the same character(the buff black guy)but i can change that but it won't appear like that in the edit view,so it is very important to rename a lot of game objects.

I added a quest as a "child" of the NPC,i put a dialog in right place in the inspector window to connect everything,i then added an item inventory and named it golden apple,which is very important to name it that.I then connect the item to the NPC or better the quest that is the child,made it count as 1 item instead of 0 otherwise i won't be able to get it since it would count as 0.

I then assigned the Quest to the NPC with it's dialog.

I learned that you need to scale the resolution aspect ratio into,1920 x 1080,for my game or their will be a red text saying that i need change the resolution,it's not a huge problem it but i do need the right resolution so i can play it right in terms of viewing it.

I added a reward to the NPC to give me(a chicken),for completing the quest,by right-clicking the NPC and giving it a inventory item,then i realized what that white box is,on the NPC,by adding a item to the PC it gained a small rectangle in the white box,with a total of 3.The rectangles show how many items,dialogs or stuff the NPC has in them.

I have to deactivate the chicken so i get it after i finish the quest.

Later i tested my game when trying to get the NPC's reward,it all worked out well except i forgot to add the chicken sprite to the rewarded,i got a golden apple instead,but then i fixed it later,then i edited the roof by increasing the box collider because i wanted the fading to happen earlier.

I added a story item which are text that share information with the player when they reach a specific location in the world.I pressed right click in my world to make a story item,then i can do **config**:where i add a text that the player will see and other key information.There are other options like,**story**:where i can create dependencies between story items,so they linked to each other.I can also choose **inventory**,which is where you can create a inventory requirement to access the story item.

I put "there is a strange person nearby" near my character,i test it and find it hard to encounter it since it's invisible even though i know where it is, and disable **Disable when discovered?** so that the story item remains even when the player has already seen it once, if not i'll see it and then it will disappear forever.

After i do that i move my NPC,items(trees,buses,etc) and add more stuff in my game so it look better and more polished.I put the story item,increase the edit collider so it's easier to activate it, and the NPC in the house because it will be a better and funny interaction.

I just found out why the bush moves with the cloud,it's because the bush was "in the cloud",the child of the cloud,making them connected and move together.

I added items tested it and went though it,realize i needed to add a box collider 2D for them.I change more stuff and add more items and fences and grass to make my game more appealing.I make build for my game,played it and realized that it is hard to exit the game,i had to use the windows key to exit the game.

In this lesson 07/10/2019 i added a bit more thing of the player to do,i added a new NPC the warrior to give the player a new quest,which is to give the warrior npc the chicken the archer gave you.It all worked well,i tested alot to see if it all works,i did worget some stuff but i eventually worked it out.I linked the chicken from the archer to the warrior.I did this because i thought it would be fun to link the NPC together.

I used this tutorial to help me make my game,it was very helpful,although i did add more stuff and change a few thing here and there.

I enabled the **Disable when discovered??** because for me it look weird for the story item to keep appearing when i activate it

feedback:

the learner has met goals for the RPG (role playing game) he has met the word count and documented where he found some thing harder to do, one of the main things that is note worthy about this work is the lay out, the reason i say this is that he has put his work out in a consistent fashion, with writing on the left and images on the right had side of the page, the one thing this i serest is that the learner should try and keep all RPG work together as a peace of the PRG is separated by the car game that we worked on, and it would help the document look better.

I started making my FPS game, put two green blob germs in the first level(area) to make the player immediately go into action. I then put another green germ but this time it was a germSpike, a different version of the other ones, I started modifying it and testing it like the unity tutorial told me to. The tutorial later told me to change the default keys in the project settings but it was alright for me so I didn't do anything.

I increased the health of the germSpike so there's some variation of enemy, even though it's very little, it will be a nice feature to add and will make my game more interesting. I also increased its sphere collider(it's "hitbox") so it becomes easier to shoot it, without sphere collider the bullets will just go through it, after that I put rigidbody on my GermSpike and disable "use gravity", so the object can float, and enable "Is Kinematic", which will stop (or anything else) from moving the object(GermSpike) around.

When I try to edit the level layout I noticed that the component was very small in the group section for no reason, I tried to fix it but to no resolve, so I decided to ignore it since it's not hindering my progress.

Then I started designing my own level with corridors and such, I turned off the audio gizmo otherwise it would clutter my level and it would be annoying, but since there wasn't a lot for me to use my "map" wasn't very big or complex. I wanted to have at least one green and purple "zone", as seen to the right with the image.

After I have done that I added a "fatBlob" which represents a door for my level. I put the door on the left of the brain and added a Key for the door, which I named Key 1 simple but effective.

I add a new weapon to my game by copying the old "GermOmatic" rifle and renaming it, I named it GermDestroyer. It's here I learned you that you can individually look at an object by double-clicking on them. I change the stats of my rifle to what the tutorial told me to do, then I make available for my character, because of that the character now has 4 weapons to choose, which is a lot in my opinion so I might change it in the future, I test it and it all went fine.

I then had to add a "LevelEnd" Gameobject, I created one on the hierarchy, renamed it Level END and added a box collider(I enabled "is trigger") to the player can touch it. I moved the box to where I wanted it to go, I increased its size, and added a component called End Checkpoint so the player can end the game and finish the Level, I tested it and it works.

After that I started to add more detail to my game by adding some props and more enemies. I added box colliders to some probes because it would look weird to have bones and teeth that you can pass through.

On the final level where you end the game I wanted to put a fatblob to the player needs a Key before he/she can finish a level, I struggled a little bit with it because the fatblob was always facing the inside of the level and would not spin to the other side, but I eventually fix it by manually turning the blob to face the right way. I put the Key for the door to the final level in the heart section of the map, I put a lot of red blood cell in there because I think it would be funny and interesting to have a lot of them near the heart like real life, hearts in real life pumps a lot of red blood cells. I also added some green germs of both kind in there with the blood cells.

On the stomach section a wanted to put a lot of enemies and a "Boss" of some kind in there, so i made a green germblob very big and increased it's health to 30 and i put Key 1 inside the Boss, like a reward for defeating it. There was a way i think of getting a key after defeating a certain number of germs but i didn't want to waste my time finding out how to do it, so i make a lazier way of simulating that.

On the final section, where you finish the level (to the left of the divided section) I putted a quite some time into it by adding a lot of props and i added one small blood cell because i think it would be for the player to finish the level and the last thing they see is a small blood cell making them think it's the cause the problem and the reason you, the Doctor, are in the human body. After i did all that i wanted test to see how every thing goes but as soon as i enter play test i immediately finish the level i quickly realize that the reason for this is because the red blood cell was in the box collider of the End Checkpoint component, now i realize that any object that has a similar property as the character the player plays, cannot be in the in the end checkpoint "zone" or the game will end before it even started. I solve this by making the box collider smaller, i tested it and now it works fine.

I did some more tests to see if everything is working and if there is any bugs. At one point while editing my game i eliminated one weapon which is the rifle i had copied from because having 3 guns was normal and better that way. I was testing and goofing off the level when i discovered a glitch or bug of some kind, When you jump with the rifle the gun disappears but you still can shoot the gun it just becomes invisible. Another glitch i found was that if you guns are in their last clip and you try to reload, the gun will disappear and you can no longer use it the only way for you to be able to shoot the gun again is to switch the weapon and then go back to it. After doing some more tests with the guns i found out that if you change weapons right after you start reloading and then switch back you will "skip" the reload animation your weapon will have it's clip full, basically you reloaded your gun without doing the reload animation.

After doing does tests i wanted to make my germs move or at least wander around the level, because i heard from a friend that he did it and i wanted to do it as well but i didn't figure it out, i even found a moving script but even that didn't work so i decided to leave it.

Overall after i did this project i found it very fun to make and leaned a lot from it, about testing your games a lot to find bugs, glitches and how to make a level interesting given little to work with.

3D Animation in MAYA

I made a simple animation with a ball bouncing using maya.

I used a simple 3D ball put it in the center of the grid in maya and made it 2D so that it would be easier to animate and move it, since i will only be using 2 dimensions.

I pressed the first frame when the ball was in the center and not in the "air". Then i moved the ball higher and forward, i pressed the fourth frame, tested the animation by pressing play on the bottom of the screen. It played too fast so i put the playback speed to real-time, the animation went pretty well even thou it looked a bit unnatural. Now all i had to do was complete the animation.

I made the ball bounce high while it's moving forward, going lower and lower with each bounce, losing it's kinetic energy at least that is what i was going for.

After finishing the animation a noticed that the balls speed didn't seem rights, when the ball was high it didn't lose speed like in real life, so to correct this i increased frames when the ball was high in the "air", this seemed to work, i made it ore natural but not as good of a "flow" as i wanted, because i'm not that good yet, but i will get there with enough experience.

I also added two extra frames in the final animation part so that the ball would stay still for longer.

I personally thing i did a reasonable performance while doing this simple project although i think a could have polished a little bit of the animation, but i didn't because a wanted to focus on more important projects.

Deadpool run animation

I started by putting deadpool on a 2D camera to help me animate but i will change it back and forth from 2D and 3D to help me animate.

I chose to do a normal run of 14 frames per step from a video i'm using to help do this animation(<https://www.youtube.com/watch?v=LByvCYBcOH4>).

I started doing the tutorial of Run Cycle with the Deadpool modal i have, i was basically copying what the tutorial was doing, i began focusing on the legs first, then the body, head and then arms.

I began by making 3 main position on 3 main frames(the first frame, the frame in which the body parts switch position from the initial frame and the final frame which is the same as the first frame), i then started putting a straight leg position, a down(pressing) position, a push position, a up position and then back to a straight leg position. Those were the main things, probably the easiest, i had to do for the running animation after that it's just polishing the running and making it smooth.

The hard part for me to follow was trying copy right side of the arms and legs, and mirroring them, then putting them in the opposite position, making deadpool run. Another challenge i had to face was coping the whole body and putting it in future frames but for some reason i could not do it, so i ended up manually changing the position of the body in the frames i want.

I had to manually copy the rotate and transition Z and Y so i can put one side of the limbs to another, manually on the Z and Y axis so it can become symmetrical which wasted a good portion of my time in this project.

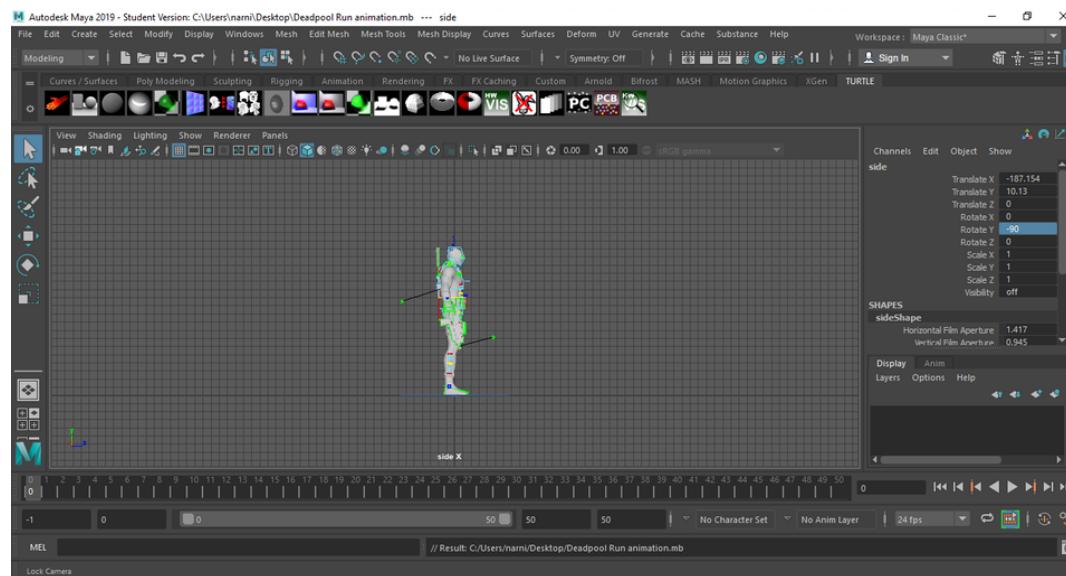
At one point near the end i wanted to get rid of the motion trail but i didn't find out how to do it, i even went to google for answers but all i learned was to press Ctrl+H to hide it that all, i ended up leaving it.

While i was moving and animation the arms of deadpool i realise, after watching the references from YouTube, that i was doing the arms wrong, i was putting the arms on a horizontal position instead of a more vertical position which is more human for Deadpool to do. The video references really help me understand how deadpool runs.

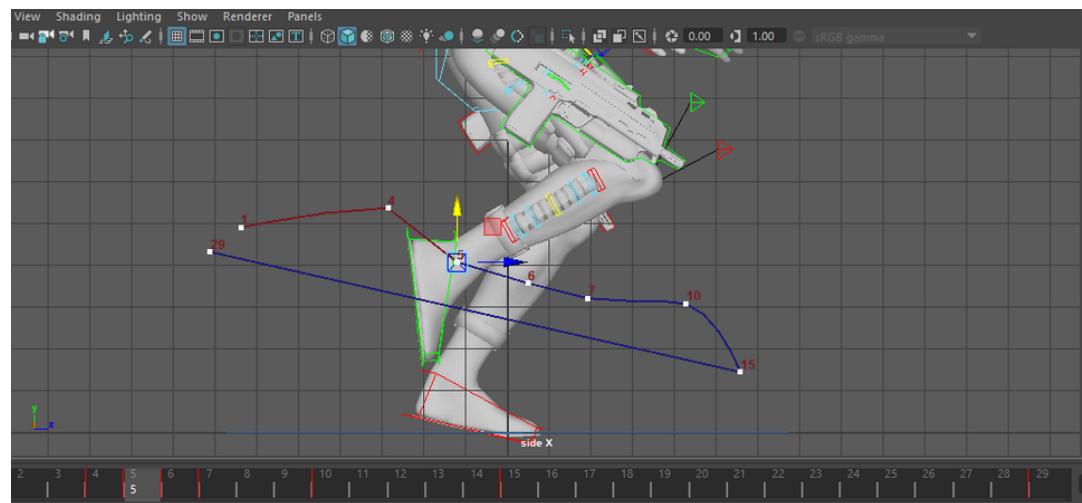
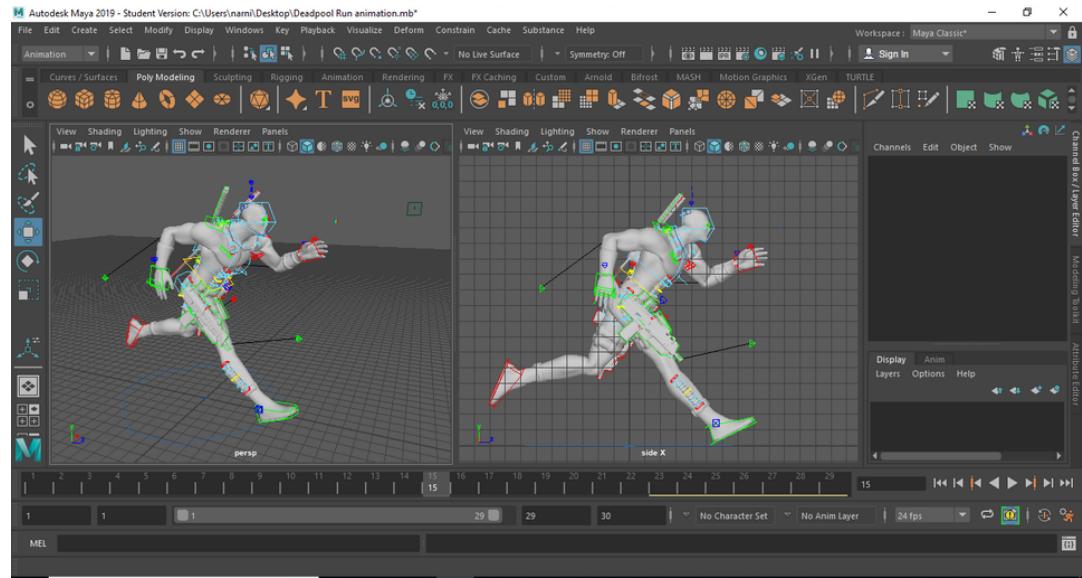
Compered to the arms and legs, the torso and head were the easiest to position and frame it, for my animation.

After i finished the animation i found out i made a mistake, in the beginning position, the right arm(green one) should have a closed arm when pulling back but it was open. I left it because my deadline was close for this homework.

I still have so much work and polishing to do but i think i did a passable job but i can do much better, i might improve on it later, using the motion trail and the graph editor better. Thanks to this project i did i found out that animation is not as easy as i though it was, which is a great eye opener for me in a good way.



How i positioned Deadpool



the motion trail to help me smooth out the running of Deadpool

On 11/10/19 we learned how to properly make a ball bounce, by losing height and length with each bounce.

2D: CONTEXTUALISED

What is being asked of me is that i make a 2D game in 2 weeks basically.

I should include relevant research, make a PDD, audience feedback, this could be from, friends, family or people who play and comment on my game. Ongoing reflection/evaluation, about how my project is going and how i am resolving my game. "making of" my 2D game showcasing the creation of the product. All this must have 1000+ words on sharepoint, not only that but i will put some images even though it is not necessary.

Finally i have to submit the build of the game on the VLE.

Create with Code 2:

I opened unity, used the direct link for create with code unit 2, i have everything set and ready to go, at this point i know how to set up create with code on my unity.

I start by putting a human object, i chose SimpleFarmer_FarmerWrangler_White, placed it in the hierarchy so it's position becomes (0,0,0,), if i putted in the scene view instead it would have a different position of any other.

I also learned that is you dragged the materiel in the folder to the scene view, the grass texture previously will became a different one of my choosing, so that's good to know.

Then i started putting animals in the scene, i took the animal, of my choosing, from the animal folder to the scene view instead of strait to the hierarchy, because i don't want the objects clustered together in the middle. i placed a animal in the scene but it was way too high in the scene, the reason for this was maybe because my camera was very high in the sky. I then added a food steak in the scene, because i preferred the steak instead of the pizza.

I renamed my Farmer "Player" and started moving the player and animals to the positions i like for them to face, i then rotated the animal so they face the player, i scaled the food steak up from 1 to 2 because it was too small.

I then created a folder called Scripts in the Assets folder and inside it i created a C# script called "PLayerController" This will be the Script where i'll be able to control the Player in unity. I placed the "PlayerController" script into the Player object with this i will actually be able to control the Player, since they are connected.

I then added public float horizontalInput; in the script i just created on top of the code because it cannot be inside void Start or Update. Then in the void Update i added horizontalInput = Input.GetAxis("Horizontal"); i tested it and it works fine, the player won't move because it doesn't have any speed not Time.deltaTime but i will add that.

After that i added a public float speed = 10.0f; then i added transform.Translate(Vector3.right * horizontalInput * Time.deltaTime * speed); in the Void Update, this will let me take control of my Player object.

I started adding the if-statement for my code and inside of the if-statement i added transform.position = new Vector3(-10, transform.position.y, transform.position.z); i tested my game and my character instead of being in (0,0,0) position it was at (-10,0,0) position, i tried to move it but it would not budge. I realized that the problem was because at play mode the Player would be at -10 in the X axis and then it would get stuck there, i tried to find out the problem with in my code and i found that it was because i placed a semicolon (;) in the end of my if-statement, which was confusing for me because i thought you put ; in almost every thing but i guess not. The little light bulb in my code did help me find the problem, which was very helpful.

I tested my game and it all worked well, the complaint i have is i would prefer that my Player be able to move more, it seems to short for me, i may change that later, it's simple i just have to increase the number of -10 to something like -12, easy.

Then i continued to follow the tutorial from the VLE has given to me, i basically copied and pasted the other if-statement then changed the < -10 to > 10, i test my game and it works fine.

I added a note: //Keeps the Player in bounds-----\ , above and below the if-statement, so i can separate the codes from each other, it's easier to find problems this way.

I then add two variables called public float LeftBoundary = -10.0f; and public float RightBoundary = 10.0f; and substitute my -10 and 10 numbers for these variables, to see if these work i test and it works fine, but later i find a better and cleaner version of this, called xRange and this is for both -10 and 10 variables. I substitute my numbers with xRange and for the negative numbers all i have to do is add a (-) sign in the beginning of xRange.

I created a new C# script, called "MoveForward" to put in the food object, then i a public float speed = 40.0f; , then i added transform.Translate(Vector3.forward * Time.deltaTime * speed); in the Void Update. I tested my game and it works, the steak moves forward, incredibly fast infact.

I created a new folder called "Prefabs" i placed my food steak in it, making a original Prefab. I go in my PlayerController script and added public GameObject projectilePrefab; i then in unity placed the steak that was in the Prefab to the new component in the Player Object, i

have to do this so unity always reuses this prefab otherwise, if i did it from the hierarchy the game would create a new steak every time and that my break the game.

Fun fact: you can rag objects while you are in play mode in unity.

I then add a separate if-statement like this: `if(Input.GetKeyDown(KeyCode.Space))`, then inside it i add `Instantiate(projectilePrefab, transform.position, projectilePrefab.transform.rotation)`; so the player can spawn steaks by pressing the space key. I test my game and it works perfectly.

I then try to make the animals into a Prefab, it's always good to make a object into what you want before making them into a Prefab, i then shift-click the animals and then apply the MoveForward script into the inspector window so they all have the script now, i was struggling a bit with the shift-click because i didn't realize you have to click on top then on the bottom not hold click. I then adjust the animals speed to my liking. I later make my animals into Prefabs, I also fine out that is you change the speed of the animals in the prefab, you change them on the hierarchy.

I create a new script called "DestroyOutOfBounds" this script will make my steak is "destroy" from my game when out of bounds, i then add a variable called `private float topBound = 30;` i add `if (transform.position.z > topBound)` in the Void Update, so if my steak is out of a specific range it will disappear from the game. I then override my food and click apply all, i tested it but something went wrong, i could not launch my projectiles, it said below in red in unity that i had destroyed the gameObjects or something, later i figure it out and every thing works, i think i did something to do with prefabs but i'm not sure.

I then do the some thing, of the out of bounds, with the animals, i add the DestroyOutOfBounds script into my animals, then in the script i add a private float called `lowerBound` wich is equal to -10, then i add a else-if statement below the if-statement, inside of it is `else if (transform.position.z < lowerBound) { Destroy(gameObject); }`

I test the game and it works fine, the animals are destroyed when out of bounds.

I started doing the Spawn Manager stuff, i first created a empty object in the hierarchy called `SpawnManager`, i also created a C# script called `SpawnManager` and i placed it in the `SpawnManager` object. I then typed `public GameObject` in the code then i used an array, `[] animalPrefabs;` , in front of `public GameObject`, so i can have a object that stores multiple objects that i can use later, to spawn in my game.

After that i placed a if-statement in the Void Update and made it recognize if i'm pressing down on the S button, by typing `if (Input.GetKeyDown(KeyCode.S))`, then i typed the code that will actually spawn my animals, with `Instantiate(animalPrefabs[animalIndex], new Vector3(0, 0, 20), animalPrefabs[animalIndex].transform.rotation);` , but i'll need a public int `animaldex` so i can substitute the 0's i had, I test this code in unity and it works fine but the only problem i found was that the new animals i spawned would not disappear when out of

bounds, but this was easy to fix, i just had to click in all the animals in the hierarchy, and override their prefabs and click apply all, after that it all works as intended, the new spawned animals will disappear when out of bounds.

Later i remove the public int animalIndex and put just int animalIndex in side the if statement, then i typed int animalIndex = Random.Range(0,animalPrefabs.Length); with this the animals i spawn will be random.

After that, i made two new private floats called spawnRangeX = 11; and spawnPosZ = 20; i made spawnRangeX equal 11 instead of 20 because 20 seemed too big of a range for me, then inside of the if-statement i placed, Vector3 spawnPos = new Vector3(Random.Range(-spawnRangeX, spawnRangeX), 0, spawnPosZ); in the code, now the animals will spawn at random locations.

I learned about the cameras prospective and the scene view prospective, if you change the game's camera projection from perspective to orthographic it will have a more top down, you will not be able to see the side of the animals much, in the scene view is the same, from perspective to isometric, this is useful if you want your game to have a different "feel". I chose to keep both cameras form the game and scene to perspective because it felt better that way.

Now i need to create my own custom function, it will be called void SpawnRandomAnimal() inside of it will be the same as if-statement in SpawnManager, then i substitute the whole code in the if-statement with SpawnRandomAnimal(), all of this will make my code a lot easier to read and manage by making sure that certain tasks that i need only for a specific purpose are kept inside on specific method and then, i can just reuse that method over and over whenever i want to spawn an animal for any use later.

After that i started making the animals spawn at time intervals, in my SpawnManager script i made two new private floats called startDelay = 2; and spawnInterval = 1.5f; then i placed a InvokeRepeating() in voidstart,inside of the invokeRepeating is ("SpawnRandomAnimal", startDelay, spawnInterval); the 2 in startDelay means how long it takes in seconds for the invokeRepeating to start and 1.5f is how long each spawned animal will take from each other.

Next in Unity on the animals Prefabs i added a box collider and i adjusted it to my liking and enabled is Trigger. I also added a box collider to my steak object, also enabled is Trigger and i also added a Rigidbody to it, disabled gravity.

I created a new C# scrip called DetectCollisions this script will let me "destroy" the animals with my steak, i added this script to all my animals and food, then i overrided them and clicked apply all, so it affects new variants of the original Prefabs. Inside the script i created private void OnTriggerEnter(Collider other) in this i placed Destroy(gameObject); and Destroy(other.gameObject); so the animals and food destroy each other.

In the DestroyOutOfBounds script in the else-if condition that checks if the animals reach the bottom of the screen, i added a Debug.Log("Game Over!"), this will help me figure out if the animals going out of bounds works, i tested it and it works.

After that i removed the three animals in the hierarchy because it would make me get a "Game Over!" without me getting any chances. After that i started polishing my scripts so it becomes easy to understand and added some comments on it, i also changed some values to

make my game slightly better. After all this i build my game and ziped the folder so i can send it to my professor.

That was pretty much it, it was all thanks to these tutorials from this site and more(<https://learn.unity.com/tutorial/lesson-2-4-collision-decisions?courseId=5cf96c41edbc2a2ca6e8810f&projectId=5cdcc312edbc2a24a41671e6#5ce34ef9edbc2a2807e18bb9>) i enjoyed this lesson, on create with code unit 2, it was a bit hard and time consuming, but that's how you learn by doing the hard stuff.

-Adjust object scale: With the scale tool or transform component on the scale i can adjust it's size, but i can adjust a specific size of an object, i can make the object taller by just changing the scale on the Y axis, for example.

-If-statements: these are basically to create logic for my game, basically, if(condition)->Action, e.g. if(to far to the left)->Stop, if statement are one of the most important tools for adding logic to my projects. The if statement is used to check a condition and if the condition is true, we run a block of statements (called the if-block), else we process another block of statements (called the else-block).

-Greater/Less than operators: These are comparison operators, they compare values and return true or false, e.g. if X is greater then Y then Z happens.

-Adjusting Scale: Adjusting a objects scale is useful, because it might make the object better to see, more imposing or even a bigger target.

-Create Prefabs: You can create Prefabs by making a new folder in assets and what ever you put in there, will became a Prefab, any thing you change in the prefab will also change in the hierarchy if the same object is there. If you create a prefab of a existing object you have to create a original prefab.

-Override Prefabs: If i press override on a object that has prefabs, you will update the items in the hierarchy and all future objects that they also will became Prefabs.

-Test for Key presses: if (Input.GetKeyDown()) will check if a specific key is being pressed, use this if you want to press a key that's not your typical input like horizontal or vertical, e.g (Input.GetKeyDown(KeyCode.LeftControl)).

-Instantiate objects: Basically lets a object or Player spawn objects out of it, i used this so my Player(Farmer) spawns steaks from within it.

-Destroy objects: i used if (transform.position.z > topBound)

{

 Destroy(gameObject); Basically you can make a object disappear or be destroyed from the game after a certain condition is met, This is very useful if you don't want your game to be filled with useless unnecessary junk that just fill CPU space.

-Else-if statements: else-if is a conditional statement performed after an if statement that if true performs a function. These are used if the first if statement couldn't actually satisfy a different condition that needs to be covered.

-Original vs Variant Prefabs: Prefab Variants are useful when you want to have a set of predefined variations of a Prefab, For example, you might want to have several different types of robot in your game, which are all based on the same basic robot Prefab. However you may

want some robots to carry items, some to move at different speeds, or some to emit extra sound effects. The original prefab you create is original. Whatever changes to original, the variants will follow the original. Any changes to the variant will not affect the original, the original controls all variants of that prefab but not other originals of the same objects.

-Input.GetKeyDown: Returns true during the frame the user starts pressing down the key identified by name, you need to call this function from the Update function, since the state gets reset each frame. It will not return true until the user has released the key and pressed it again.

-GetKeyUp: Returns true during the frame the user releases the key identified by name, this is practically the same as GetKeyDown but in a opposite way.

-GetKey: Returns true while the user holds down the key identified by name, this might be used to confirm a key is used for auto fire, when dealing with input it is recommended to use Input.GetAxis and Input.GetButton instead since it allows end-users to configure the keys.

-KeyCode: These map directly to a physical key on the keyboard, Key codes can be used to detect key down and key up events, using Input.GetKeyDown and Input.GetKeyUp, e.g. (Input.GetKeyDown(KeyCode.Space)).

-Spawn Manager: Basically this lets you spawn multiple objects in a scene, you'll need a spawn manager script, you can make it spawn one or more objects and at random location if you want, and you can even make it automatic.

-Arrays: These are used to store multiple objects, multiple variables inside of it, it can be has many has i want. To do this in code you have to for example: public GameObject[] animalPrefabs;

-Keycodes: This helped me spawn all my animals, example: if (Input.GetKeyDown(KeyCode.S)).

-Random generation: I used this to produce random values for my array, i used int animalIndex = Random.Range(min, Max); this Return a random integer number between min [inclusive] and max [exclusive] (Read Only), with integer numbers, the last max number does not count for the randomness, e.g. Random.Range(1, 9), 9 does not count this is know as a exclusive number, it goes though 1 to 8, only with float number does the last number count, e.g (0.0f, 7.0f) 7 does count for the random generator.

-Local vs Global variables: local variables are inside of our method/function instead of our script and can only be used inside of that method/function, while Global variables are declared outside any function, and they can be accessed (used) on any function in the program.

-Perspective vs Isometric projections: Isometric is 3D objects drawn with no perspective, lines have no vanishing points. Perspective is 3D objects drawn with 1 or more vanishing points, often along, above, or below a horizon line to simulate realism.

-Integer Variables: Integer variables are variables that must take an integer value (0, 1, 2, ...). Basically only whole numbers are accepted. Fin fact: A special kind of integer variables is binary variables. Binary variables can only take the value 0 or 1. They are integer variables with a maximum of 1 on them (and don't forget there is always an implicit minimum of 0 on each variable).

-`.Length`: This is used to track the length of how large the array is, you can use this instead of typing the last number of total elements there are, it's good to use because in case you increase or decrease the number of elements in a array (`.Length`) will adjust itself, and example of (`.Length`) is: `Random.Range(0, animalPrefabs.Length)`;

-Create custom methods/functions: This will help me create and do something that's not in the program already, you should name it in a way that tells you the task that it's doing, you can do whatever you want with this custom code, even stuff you already did.

-`InvokeRepeating()` to repeat code: This will call a function multiple times, e.g `InvokeRepeating("SpawnRandomAnimal", 2, 1.5f)`; This function starts after 2 seconds and is called again every 1.5 seconds.

-Colliders and Triggers: A collider doesn't need to be exactly the same shape as the object's mesh - a rough approximation is often more efficient and indistinguishable in gameplay. Triggers are colliders that are capable of executing script when a collider touches the trigger, exits the trigger, or while the collider touches the trigger via `OnTriggerEnter`, `OnTriggerExit`, or `OnTriggerStay`.

-Override functions: I used this to override the `OnTriggerEnter` function from `MonoBehavior`, a child class inherits the data members and member functions of parent class, but when you want to override a functionality in the child class then you can use function overriding. It is like creating a new version of an old function, in the child class. I used this site to help me understand Overriding(<https://learn.unity.com/tutorial/overriding#5c893d46edbc2a0d28f48954>).

-Log Debug messages to console: The `Debug.log` file can record database operations, system processes, and errors that occur when executing a transaction or running unit tests.

-Compartmentalization / Abstraction:

-Box Colliders: The Box Collider is a invisible shape that is used to handle physical collisions for an object. I used this for may animals and food, that way they can collide with each other, with this i made them destroy each other.

-`OnTriggerEnter`: This is is called when the Collider other enters the trigger, this message is sent to the trigger collider and the rigidbody (or the collider if there is no rigidbody) that touches the trigger, `OnTriggerEnter` can be a co-routine, simply use the `yield` statement in the function.

-`Debug.Log`: A message will appear on the bottom of unity, the message can be what ever you want, telling you when something has happened in unity, example, if i wanted to know if a collision happened, i could use `Debug.Log` to see if it actually happened.

-`LogWarning`: `Debug.LogWarning` is a variant of `Debug.Log` that logs a warning message to the console, when you select the message in the console a connection to the context object will be drawn. This is very useful if you want know on which object a warning occurs. When the message is a string, rich text markup can be used to add emphasis. See the manual page about rich text for details of the different markup tags available.

-.LogError: This is a variant of Debug.Log that logs an error message to the console, when you select the message in the console a connection to the context object will be drawn. This is very useful if you want know on which object an error occurs.

The screenshot shows the Unity Editor's code editor with the DetectCollisions.cs script selected. The script contains the following code:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class DetectCollisions : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      void Start()
9      {
10
11
12
13      // Update is called once per frame
14      void Update()
15      {
16
17
18
19      // Objects that come in contact with each other will destroy one another
20      private void OnTriggerEnter(Collider other)
21      {
22          Destroy(gameObject);
23          Destroy(other.gameObject);
24      }
25  }
```

My final completed code after completing the lessons, all the codes from below are all the same except from different scripts.

The screenshot shows the Unity Editor's code editor with the SpawnManager.cs script selected. The script contains the following code:

```
4
5  public class SpawnManager : MonoBehaviour
6  {
7      public GameObject[] animalPrefabs;
8      private float spawnRangeX = 11;
9      private float spawnPosZ = 20;
10     private float startDelay = 2;
11     private float spawnInterval = 1.5f;
12
13     // Start is called before the first frame update
14     void Start()
15     {
16         InvokeRepeating("SpawnRandomAnimal", startDelay, spawnInterval);
17     }
18
19     // Update is called once per frame
20     void Update()
21     {
22
23
24
25     void SpawnRandomAnimal()
26     {
27         int animalIndex = Random.Range(0, animalPrefabs.Length);
28         Vector3 spawnPos = new Vector3(Random.Range(-spawnRangeX, spawnRangeX), 0, spawnPosZ);
29
30         Instantiate(animalPrefabs[animalIndex], spawnPos, animalPrefabs[animalIndex].transform.rotation);
31     }
32 }
```

```
DetectCollisions.cs      SpawnManager.cs      DestroyOutOfBounds.cs      MoveForward.cs      PlayerContr
Assembly-CSharp
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class DestroyOutOfBounds : MonoBehaviour
6  {
7      private float topBound = 30;
8      private float lowerBound = -10;
9      // Start is called before the first frame update
10     void Start()
11     {
12     }
13
14
15     // Update is called once per frame
16     void Update()
17     {
18         // If an object goes past the player view in the game, remove that object
19         if (transform.position.z > topBound)
20         {
21             Destroy(gameObject);
22         }
23         else if (transform.position.z < lowerBound)
24         {
25             Debug.Log("Game Over!");
26             Destroy(gameObject);
27         }
    }
```

Quiz 2:

It was fairly easy but i did think for a bit, the only question i got wrong and struggled a bit was question 3, i chose the last answer but it was the 5th answer, i thought i was right because i saw in my code for create with code 2 that the Instantiate function was in the void Update, so i thought that's the only place it can be, but i was wrong, my initial choice was the 6th option, because i could not understand that you could do one or the other.

Challenge 2:

I make a new unity and import all the assets, then i start doing the tutorials, first i go to the Spawn Manager object and substitute the dogs that are in the Spawn Manager X (Script) component with Ball 1, Ball 2, Ball 3 in the Element 0, Element 1, Element 2 slots respectfully. Then i make the Player spawn dogs instead of ball, by putting a Dog Prefab in it's Spawn Manager X (Script) component with only one slot.

I then went and double-clicked the dog in the Prefabs folder so i could decrease the dogs box collider because it was WAY too big.

After that i went and tried to fix the out of bounds problem, i went to the DestroyOutOfBounds C# Script, i found out that the if (transform.position.x > leftLimit) code was wrong the symbol (>) should be (<), so i changed it to the lower then symbol, then i increased the private float leftLimit = 30; to 45 and i also changed the (leftLimit) to (-leftLimit), so that the dog would disappear because in the left of the scene view that's where values became negative, if i didn't do that the dogs would disappear instantly.

The problem of the balls not disappearing is because the else-if statement has a Z axis instead of a Y axis, since the Z axis will never go lower than -5 it'll never disappear, all i had to do was change this else if (transform.position.z < bottomLimit) to this else if (transform.position.y < bottomLimit), i tested all this and it works fine now.

Later i started trying to make different balls spawn randomly instead of just the blue ball, i typed int ballIndex = Random.Range(0, ballPrefabs.Length); in the void SpawnRandomBall (), but it still did not work when i tested it, i re-read the problem from the tutorial and looked at the code again and then i realized that i have to substitute the 0's in the Instantiate code with ballIndex, after that it worked when i tested my game.

I decided not to do bonus because i have other stuff to do, and i don't want to waste my time figuring stuff out even though that would be beneficial for me, but i have deadlines to complete.

Lab 2 New Project with Primitives:

Here i created a bare bones of my personal project, i opened a new unity, made a folder so i can save my project in it, named my folder accordingly it, ,then in unity i right-click in the Hierarchy and made a 3D Plane object, i reseted it's position in the transform component and Increase the XYZ scale of the plane to (5, 1, 5).

I then created a sphere and renamed it "Player", after that i went to the assets folder and i Created a Folder named "Materials", inside that folder i right-clicked and Created a Material and rename it "Blue", i changed the Blue materials colour by going into it's inspector window and clicking on the Albedo color box and changing it to a blue. I then dragged the material to the sphere and it became blue.

After that i changed the position the camera at (0, 10, 0) directly over the player and rotate it 90 degrees on the X axis, i did this for a top-down game feel.

Later i created more objects in my game, i created a cube, positioned it in a position i like, renamed it "Enemy", added a material, renamed it "Red" and painted it red, i then added the red material to the cube, after that i duplicated the cube four times, to simulate multiple enemies. I then created another cube, renamed it "Fence", put it between the enemy and Player, i scaled it making it long and thin, i then made a new material in the assets folder, called it "Brown", made it brown, then dragged said material to the Fence, simulating a fence, i positioned every thing in order to make sense.

I then saved my scene, then exported my game by Right-clicking on the "Assets" folder, clicked Export Package, then clicked Export, i made sure that include dependencies is enabled, i then created a new "Backups" folder in my Personal Project folder, then saved it with my name and the version number, Diogo_V0.1 and that was my experience doing Lab 2.

-Primitives: These are objects that we use as place holders, just so we can see how our game will work before we actually create any models that we might use or any animation.

-Create new materials: In this project i created new materials so that way i can see which of my objects are bad, power-up, an object or is my Player.

-Export Unity packages: This is good to do just in case any thing breaks, or if i want to pass a project i did in unity, to some one.

2D game:

Before i started the tutorial i did some research of the best 2D game of all times and i found this site (<https://cultureofgaming.com/the-top-25-2d-platformers-of-all-time-in-progress/>) that shows you 25 most popular games of all times, here there are a lot of diversity of games, which is good because it tells you why they are popular, i got a lot of information thanks to this site, the why it is popular and what its best aspects are.

I opened unity, i imported the 2D game kit then i created a new scene, named it, after that i tried following the tutorial it was given to me in the VLE, i went to window > 2D > Tile Palette, but i could not find the Tile Palette option in 2D, the only option it had was Sprite Packer, this delayed my progress because i could not progress without it, i went to google to find solutions an i was not the only one that has experience this problem, most people said that they fixed it by downgrading or upgrading their unity version, but i didn't want to do that since it took quite some time to install this kit, i even went to the shortcuts in Edits, after i search some more Googling and finding a different solution, to find the tile palette and use or make a shortcut so i can use it, i even went to the folders in my game and try to fine it by typing in the search bar but to no avail.

I eventually made a new unity with the 2019.1.0 instead of the 2019.2.3 version, after i imported every thing and after that i finally worked, i don't know why this has happened but i heard that this is a common bug and unfortunately i wasted a lot of my time because of it.

After all that i started doing the tutorial and i started painting a level, i choose a grassy tile, made some platforms i liked then i tested my game, i moved my Player, see how fast she goes, how high she jumps and more, i did found something interesting, if the Player jumps into a tall wall and keep running forward, it stay there, continuously trying to jumps of some sort.

Then i went and try adding a moving platform, i went to Assets>Prefabs>Interactables and chose the MovingPlatform object and dragged it to the scene window. I tested my game and it works, the platform moves, i even went in front of it, to see what would happen if i would stop it, bo it pushed me off the ledge.

Then i added another movingplatform, but this one will loop around, i found out that you can preview where the platform will go, you just have to go to the inspector window of the object and go to Preview position slider and set the completion of the movement of the platform. i made the platform go around in a square by adding two more nodes in the Moving Platform component, i adjusted the nodes to make a square although i was missing a node but i don't need an extra one since i can make them loop together by changing the looping from BACK_FORTH to LOOP.

After that i made more platforms to the left of scene so i can add a door and Pressure Pad. I went to the Interactables folder and placed a door in my scene, then i added a pressure pad next to my door to the right, i tested my game, and i can't pass though the door which is good, and the pressure pad does not work, i have to activate it in the inspector window and link it to the door.

I went to the PressurePad's Inspector, went to the Pressure Pad component, In the On Pressed list, i clicked the + at the bottom right to add a new event, i then drag the Door gameobject from the Hierarchy window to the **None(Object)** field in the event and then In the No Function dropdown i clicked on Animator > Play(string), a text box that appeared under the dropdown and i typed DoorOpening in it. I tested my game but nothing happened when i pressed on the pad, i re-read the tutorial in this section and it was supposed to make a sound when i stepped on it but that didn't happen when i tested it earlier, i looked more into the PressurePad gameobject and i realized that the box collider was too small or that i positioned it a bit too low on the platform, i then increased the box collider to be taller, i did a test and i could not go though to pressurepad because the collider was too tall but when i jumped on top of the collider a sound appeared and the door moved, so everything works now, all i have to do is shorten the height a little so the Player can pass through while still on top of it, this all worked without the "Is Trigger" being enabled.

I lowered the box collider on the pressurepad now i can trigger it without jumping on it, later the tutorial told me to try out a different method of opening the door using a reusable switch. I made a "hole" on top of the door there i placed my reusable switch, i added the door object to the interact on trigger component, i didn't need to type DoorOpenaning since it was already there for some reason, then i went to the No Function dropdown, found Animator > Play(string), saved then tested my game, i shot the switch with my gun, by pressing the "O" key, that the Player has, and it worked it played a sound and the door moved.

Later i went to the enemies part of the tutorial. I went to Prefabs>Enemies, i made more platforms to put the enemies, then i adder a chomper, i increased it's speed from 1 to 2, it's health from 1 to 2 and then i decreased it's FOV (field of view) view from 360 to 150, the rest i left it the same, but i might mess around with them to see what they do and what might happen. I tested my game and it all works, the enemy attacks when i'm near it, it can't see me if i'm behind it and it takes 2 shots from my gun to kill it but only one melee attack to kill.

After that i made a tall platform so i can kill a Spitter (enemy) by dropping a PushableBox on top of it, i added a spitter and i did some modifications to the spitter but not much, just the FOV and other stuff. I added the pushable box on top of the platform, then i added a component in its inspector window called "Damager", this "Damager" Component tells any GameObject that has a Damageable Component on it (like a Spitter or a Chomper) to give it damage, which is represented by a green collision box. I adjust the damager box collision to be the same as the pushable box, then i inside the Damager component on the HittableLayers dropdown i select the Enemy Layer, so now the pushablebox will damage the spitter, i test this and it works the spitter died.

Later i find out i can decorate my level with objects in Art>Sprites>Environment, the Kit also includes decorative sprites, which are all used in the example game "The Explorer" included in the Kit, a lot are stored in the sub-folders, so i shouldn't forget to expand the small arrows and see what other sprites are in that category, but i might use the decorations later in the creation of my game, i'll focus on the mechanics first then i'll decorate my level.

Later i am trying to make the player (Ellen) teleport in the same scene then teleport onto a different scene. First i added a Teleporter in my game, i'm mainly using it as a sprite, because by the seems of it the actual teleporting thing that will make Ellen teleport is invisible just a box collider. Then in the Project window, went to Prefabs > SceneControl, dragged a TransitionStart Prefab to the Scene View and placed it in front of the Teleporter, after that i dragged another TransitionStart in the scene but in a different position, i renamed the new TransitionStart to TransitionEnd, i selected the TransitionStart in the hierarchy, went to the TransitionPoint component, dragged Ellen GameObject from the Hierarchy into the Transition Point Component's Transitioning Game Object slot, then i Set the Transition Type to Same Scene. After that i dragged the TransitionEnd GameObject into the TransitionPoint Component's DestinationTransform slot and Set Transition When to On Trigger Enter, i could make the player teleport only when you press the interact key (E), by setting the **Transition When to Interact Pressed** but i prefer **On Trigger Enter**, since all you have to do is walk to it.

I closed my unity that had my, but i did save it, and when i opened it again, i had to make it in a 2019.2.3 version, which is good,it seems when you make a project that is of a older version, unity will want to make that project of your the newest version. I also check to see if i still had the same problem when i started my 2D game, but to my surprise, no the Tile Palette was there in window>2D, so every thing is good now.

I then went i tried to make a new teleporter but this time it will teleport the player to a new scene. I go to the Zone 1 scene in the kit, i placed a TransitionDestinaton prefab in the Zone 1 scene next to the ship, i set the Destination Tag to a letter(D) and then dragged the player GameObject (Ellen) from the Hierarchy window to the Transitioning Game Object slot, then i had to make sure that my destination Scene is in your Editor's Build Settings. To do this, go to File > Build Settings and click Add Open Scenes. I then created a new Transition Start in my game scene that i created , not Zone 1, i made a new platform purposely for this. I Set Transition Type to Different Zone, Set New Scene Name to the Scene you want to send it to(Zone 1) and then i Set Transition Destination Tag to the Destination Tag letter you set in the Transition Destination Component(D). I tested my game went to the TransitionStart that takes me to a new scene, i went into the trigger and then the game was loading then i appeared in Zone 1, it works, although my Character was a bit high in the spawn so i lowered the spawn the zone 1 later.

After all that i started making my game a bit better, using decoration and more enemies, i added two new enemies, a chomper and a spitter, then i added some decorations to my level like some plats, vines, mushrooms, some trees, rocks and a statue.

Then i wanted to try out the pre-made scenes to see how they work and if it's enjoyable, i started with the "start" scene, these scenes are in 2DGamekit > Prefabs > Scenes. I clicked on it, and it began loading, which took some time, i don't know if it was the games fault or my PC's fault the loading took some time to load, a main menu appeared, it was nice menu, it had a quit, options and a start options, the option menu was very basic. I press play and it loads for a bit then i enter Zone 1 which is basically the tutorial of the game, telling you what you can do. I go through a platform and it takes me to Zone 2, here is where i find a bit of a problem, on to the right of the scene there is a "cliff" you need to go, but i felt instinctively that it was not the right way since it looked like a cliff, but turns out it was, you don't go down the cliff you'll get near the cliff and then the game will transfer you to the next level, it felt weird, like you are trying to pass a line but the game stops you and teleports you to a different location. The way i would fix this is by either extending the "cliff" or path so it gives you the illusion of continuing running to the next level, while running, so you don't get the feeling of being stopped, or you can make the camera stop following the Player at a certain position, probably to the far right or left, then you character will go "through" the edge of the screen and then the Player will be transferred to the next level.

Later i did the other Zones of the game and they were fun to play, Zone 4 was a bit challenging, which is good, i even died there mainly because the spacing between the pushable boxes in the green acid water were a bit too far, i did do it but i needed to try multiple times. After i collected all three keys i went back to Zone 2 to get to the final Zone, Zone 5, there i fought a Boss, the fight itself was a good challenge very simplistic but challenging i even almost died, after i defeated the Boss i gained access to the teleporter, when i entered the teleporter i was teleported to Zone 1, it was anti-climatic and disappointing that that was the closure i got when i passed through the teleporter, so anti-climatic in fact that i did the boss battle again to see if i would get a different "ending" but no i still was teleported to Zone 1, i kinda get it since this is a 2D game kit from the asset store but still i wanted some closure. Overall this game i played was well made, simple but well made and not very complex.

Throughout the times i was doing this game i did some additional research of how to make a good 2D game, these are some researches i have done (<https://www.youtube.com/watch?v=2MlmgNvgrNg> and <https://www.youtube.com/watch?v=on9nwbZngyw>) i also viewed more videos and articles on this matter.

This tutorial and project i did that is a 2D game was a good experience to get, i learned many things, like how a transition of an gameobject to go to one place to another using the TransitionStart and TransitionDestination, making an object deal damage to a specific object, adjusting an enemies stats(health, speed, damage and more) and their FOV and much more. After i did that i had all the freedom to decorate and improve my game but because of my own time mismanagement i did not have time to do a lot of decorations or additional

stuff to put in my 2D game, overall i had fun and i learned a lot in terms of the 2D genre, which is one of my favorite genre, this has help me a lot in becoming a game developer.

3D Game:

I downloaded the 3D kit so i can make my own 3D game, i create a new scene by going to Kit Tools > Create New Scene and i named it "3D game by diogo".

I press play in Unity to see how the game work and it's controls but an error "All complier errors have to be fixed before you can enter playmode!" on the bottom of unity there is a error in red "Effect Resonance Audio Renderer could not be found. Check that the project

contains the correct native audio plugin libraries and that the importer settings are set up correctly" it would seem i need to fix the audio of the game.

I showed this to the teacher and he told me to go to the Level 1 of the 3D game kit, a scene that was already made, and an error still appeared but the reason was different, then the teacher told me to close unity and go to his computer and play the game that was already made.

I played all the pre-made scenes of the kit, after playing the games i can tell that the game is well made, beautiful and well executed, but there are some critics i have to give, the attack you do with the staff is a bit short in my opinion, the mobility isn't very good, when you jump and fall sometimes you will roll when you fall at a somewhat high altitude, slightly higher then your jump, this roll is uncontrollable so if you land near an ledge you will roll and fall, it's annoying because you can't control the direction of the roll. You can also get a bit lost in the game, you don't a good sense of direction in the game especially in the caves.

On 4/11/2019 i downloaded the 3D game kit in unity in my home computer so i can do my work also at work. After imported all the assets i went to kit tools and created a new scene, named it, then i play it and it all worked well, because of this i am going to make sure to save my game and also putting a save file in my one drive so i can continue my work i did at home to college, since the 3D game kit at college i created there did not work.

After that i added a MovingPlatform in my scene view, by going to Assets > 3DGameKit > Prefabs > Interactables and dragging the designated gameobject to the scene, i made the gameobject go up and down by adjusting the End Point Tool on the MovingPlatform. To make the platform loop forever i have to make it's Loop Type into Ping Pong so it doesn't do it only once. In the Simple Translator Component, i also ticked the Activate box, so that the platform activates and moves, if not it would not move.

After i adder the moving platform i realized that it was too big, so i made the plane (the ground) of my scene bigger, i tested my game and it looked good, but then a problem happened, at a specific place on the plane where the ground was low the Player would die, the reason for this i think is some parts of the plane where low enough for the out of bounds collider to contact the player, killing it. I used the Quick Start - Unity Learn site (<https://learn.unity.com/tutorial/quick-start/?projectId=5c514897edbc2a001fd5bdd0&tab=materials#>) i re-watched a video, especially in the part where the man doing the tutorial does the scaling of the plane to see if i was doing it right, i practically was doing the same except that i did not flatten my plane, i instead increased the Y axis position so that the lower parts of my plane would not be in contact with the "out of bounds collider", i also increase the Ellen gameobject, Player, and movingPlatform positions accordingly. I test my game and the plane is fine, i don't die in any place of the plane, i jump off the plane to see if i can still die and i discovered a problem. When i die i spawn below the plane, and since there's nothing preventing me to fall again i die again, in a endless cycle of death, i quickly understood the problem. The Player's spawner was below the plane so all i had to do was make it higher, it was a easy fix.

I then placed a small door in my scene, adjusted it a bit so it doesn't seem like it's floating in mid air, then i added a PressurePad next to the DoorSmall gameobject. In the PressurePad gameobject, in the Send on Trigger component i added DoorSmall to the Interactive Object slot in the Send on Trigger Enter, a white arrowed line showing a connection from PressurePad to DoorSmall. I press play and it works perfectly.

Later i added a enemy in my scene, specifically the "Chomper", i placed it at the back of the platform near the edge, i press play to see if the enemy works and when i got near it tried to attack me but it didn't move because i hadn't implemented that on it just yet, i later moved it to a different location because i didn't like were it was.

I then adjusted its stats in the inspector window, i expanded the Player Scanner, in Detection Radius, i set the value to 3 and i turned the Detection Angle slider to 180. I then needed to make the chomper able to move, to do this, in the Plane gameobject i needed to add a component called NavMesh Surface, i set Agent Type to Chomper, i set Include Layers to Environment and Vegetation only, and i finally clicked "Bake" and after that the plane became blue, this will help the chomper "see" what it can move to, i test my game and it works, it will follow the Player and try to attack and will stop pursuing you after 3 seconds of losing its sight of the Player. Later i added more health to the Chomper from 1 to 3 so i can have more of a challenge, i test my game to see if it works and to "play" around some more, and while i was testing my game i found out that if you press restart in the menu after pressing ESC you will be below the Plane and die, then you will spawn above the Plane thanks to the checkpoint. I believe i know how to solve this, i increase the Y axis position from 0 to 2.83 of the TransitionDestination GameObject, i play my game restart my game on the menu and now it works fine, Ellen will respawn normally every time you restart the game.

After that i went to the Interactables folder in side Prefabs, and dragged a DestructibleBox Prefab into my scene. I place it above my Chomper when i press play it fell onto the chomper but it did nothing, not even damage it, i went and added, in its inspector, a box collider to the Box and enabled Is Trigger, i then added Contact Damager in the Contact Damager Component i set the Amount to 3, i test my game and it works the chomper dies when the Box hits it on top of it and after that the Box bounces off it. Later to make the DestructibleBox break when it hit the Chomper i had to add a Contact Damager Component in it, set the amount to 1 and the Damaged Layers to Enemy, it worked well in the test i did when playing.

Next the tutorial's next lesson to teach me was to decorate my scene but i didn't want my scene to be messy when i do the final lesson which is teleporting the player to one location to another, i want to save the decorating for later.

I start to go to the Project Window go to Prefabs > Environment > Structures and drag the GateWayHugeTeleporter Prefab to my scene, and it was HUGE, so i scaled it down to a reasonable size, i then select TeleportPlane in the GatewayHugeTeleporter gameobject, the TeleportPlane is the child of the huge gate. I then add a component, BoxCollider and tick Is Trigger later i add a Teleporter component to it as well and change the Layers to Player, i go to the Checkpoints object in the hierarchy and its child is opened, which is Checkpoint, with the TeleporterPlane selected in the Hierarchy i drag Checkpoint (The child) into the Destination Transform field of the Teleporter Component. This will let me teleport to the Checkpoint through the gate. I later expand the object Ellen in the Hierarchy, inside it

was RespawnParticles, i have to drag the RespawnParticles to the Teleporter component in the TeleportPlane, but there was a mistake in the text the lesson wrote, it told me to "With GateWayHugeTeleporter selected in the Hierarchy" to "Drag RespawnParticles into Enter Effect on the Teleporter" and "Drag RespawnParticles into Exit Effect on the Teleporter" but the problem is that the GateWayHugeTeleporter object doesn't have Exit Effect or Enter Effect it's child has, TeleportPLane, it was a bit confusing for me but i soon understood what it wanted to mean. So i dragged the RespawnParticles into Enter Effect and Exit Effect on the Teleporter, I test my game and the the gate didn't have a "blue aura" in front of it which was weird for me, i went toward it to see if it works and it does, and the particle affects work too. But the portal was "invisible" in my game and the images from the site that helped me make my game had a blue color, and i wanted that too, i try to see if there was any problem in my huge gate, if i did something wrong? the box collider did something? or something, i check around it and i found the problem. It was in the opposite direction, so i turn it around and now i can see the blue light it does when i spawn to test my game.

It looks good now since i see the gate properly and the particle affects look cool, even when i die they appear now. After that i start decorating my scene with plants, trees, rocks and more. I went to Prefabs > Environment > [...] and chose the decorations i wanted and place them in my scene, i then learned how to use the Rock and Vegetation Painter, i have to go to the Hierarchy and locate VegetationPainter or RocksPainter, then click on the GroundCover and then click on my scene to place rocks or vegetation, i can choose the different types of rocks or vegetation by going in the Instance Painter component where are pictures of each prefab in the painter and choosing the ones i want, The selected prefab will be grey. Later i learn the controls of the Painter which are:

-Left Click = Paint;

-Ctrl + Left Click = Delete;

-Alt + Scroll = Increase Brush Size;

-Space = Randomize positions and rotation.

Any object i paint on will be stored as a Child of the Parent.

After i did my decorations to my game, like adding grass, rocks, big plants and small, more enemies, pillars and more, i "baked" my map so the enemies know where to go. I test my game to see if every things is alright and i found a few "hole" where you can see the outside through the walls i placed, i fixed this by adding something to cover this.

I then builded my game and after a long time i decided to play it, and i realised that i builded also the pr-made scenes, which is the reason it took so long to build my game, I rebuild it with just my scene, i did this by unchecking the other boxes and just leaving my game. Throughout the duration of this project i learned after you close your computer (not shutting down, just closing), if your unity is open with your game you won't be able to collabe it, so you are going to close your computer might as well close your unity.

I fairly enjoyed this 3D project, i learned a lot of things that help me understand how 3D games work and how to apply it in Unity. It wasn't overall hard but one lesson did take some time was the decorations, even though it wasn't a very important task, but i did learn through experience how decorations can work with each other. I had some problems but i overcame them. although i did have some problems with Unity itself, one example was Unity kinda crashed, it was all white, i could still edit and use the bar on the very top and save my game but i couldn't see my project, i simply closed my Unity and then reopened it so i could resume me work, i also can't collabe my game because it's too big, i also can't post my game in the VLE because it's too so i'll ask my teacher another way to send it.

I think i did a fairly good job while doing this project, i did what was asked of me and did all the lessons from the site of making a 3D game with the 3D game kit, with some slight modifications. On the decorations i did it all by my self, the site just showed me how to use it. I wish i did some more research and some audience feedback, the reason i didn't do some audience feedback was because my game didn't seem right for it, you just wander around with now objective, so it wasn't a great deal for me to get feedback, even though i should get it to improve myself, a great way to get feedback is through itch.io, even though it's a lot of work for a uncompletable game of low quality, then again i could get some feedback through my family so i might try that next time i do a big project.

I did some research on how to make a 3D game, these are some of the sites and videos i went to get my information (<https://www.gamedesigning.org/learn/3d-design/>), (<https://www.youtube.com/watch?v=p9vwPmOZhiQ>) and (https://www.youtube.com/watch?v=1wn5Ur1_vKg) this has help me understand Unity and how 3D game are made properly.

Create with Code 3:

I installed the Prototype 3 assets in to my Unity, then i started doing unit 3 with create with code, I was to change the back ground to my liking, i left the background has it is because i preferred the town background over the others. I then had to add a Character to the scene, i added the "SimplePeople_RoadWorker_Brown" gameobject to the scene and renamed it to "Player", i added a Rigidbody and Boxcollider components to the Player, i had to edit the box collider because it was short and too low to be accurately representative, so i adjusted the collider, after that i added a new folder called "Scripts" in the assets folder and in scripts i added a C# script called "Player Controller".

In the PlayerController i made a private Rigidbody playerRb; and inside the void start i added `playerRb = GetComponent<Rigidbody>();` (this code will make my game look for the Rigidbody) and `playerRb.AddForce(Vector3.up * 1000);` (this will apply the force upwards). I test it and it works at the start the Player object will jump very high and thanks to the in game gravity physic, or better it's Rigidbody, it will come down and not go upwards infinitely.

Later i added a if statement in my code so i can control the Player, it was: if (Input.GetKeyDown(KeyCode.Space)), this will let me control the player by pressing the Space button, then inside the if statement i copied the playerRb.AddForce(Vector3.up * 1000); from the void start and cut the one that was in the void start, i then modified it to become playerRb.AddForce(Vector3.up * 10, ForceMode.Impulse);

After that i started tweaking the jump force and gravity, i did this by adding in my code a public float jumpForce = 10; so i can replace the 10 in the if statement and so it doesn't become hard coded. Then in the void start i added a Physics.gravity *= gravityModifier; the (*=) makes the code simpler because it's the same as Physics.gravity = Physics.gravity * gravityModifier; saying Physics.gravity twice could be really confusing and is redundant, we can short-form this by saying times equals (*=). I then had to make a public float gravityModifier; so i can mess with it on play mode. I play test my game, i press space and the player flew up to infinity because i didn't add gravity in the player controller component, it was 0, i then messed with the component a bit, added more gravity, less jump force, etc.

After that i made a public bool isOnGround = true; , Bool is short for Boolean which is another value of data type, just like an integer is for whole numbers, floats are for decimals. I then changed inside the void Update and the if statements to become if

```
(Input.GetKeyDown(KeyCode.Space) && isOnGround) {  
    playerRb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);  
    isOnGround = false; }
```

Below this i added: private void OnCollisionEnter(Collision collision) {
 isOnGround = true }

I play test my game and it works, i even test to see what happens if i uncheck the "Is On Ground", that the script component has now, what happened was i couldn't make the Player jump, and the same happened if i unchecked it mid-jump.

I then had to add a obstacle and make it move left, i added a Prop_Barrier02 gameobject in my scene, i made it align with the Player so the player doesn't miss it. I renamed the object i placed "Obstacle", added a Rigidbody and Box Collider in it, then i edited the collider so it's aligned with the obstacle.Later i made a folder in Assets called "Prefabs", i dragged the Obstacle into it and made it into a original Prefab, then i went to the Obstacles inspector window and went to Overrides, on the top of the inspector and clicked apply all, so all future Obstacles are the same as this one i created.

After that i created a new C# script called "MoveLeft" and dragged it to the Obstacle in the hierarchy. Inside the script i added a private float speed = 30; , then inside the void Update i added transform.Translate(Vector3.left * Time.deltaTime * speed); and i might change the speed value in the future if am not happy with it. I then applied the "MoveLeft" script to the Background Object in the hierarchy, i press play and it all works, they both (Obstacle and Background) move to the left, although it is quite fast.

Later i added a empty GameObject callecd "SpawnManager" and i also made a C# script called "SpawnManager" as well, then dragged the C# script to the SpawnManager GameObject. Inside the script i added a public GameObject obstacalePrefab; and a private Vector3

```
spawnPos = new Vector3(25, 0, 0); , then inside the void Start i added Instantiate(obstaclePrefab, spawnPos, obstaclePrefab.transform.rotation);
```

I test my game and it didn't work because i forgot to place the Obstacle Prefab into spawn manager component, so i do that and now it works. But there has been a problem, i typo i made, in the spawn Manager component, below cript it says Obsticale Prefab then the Obstacle Prefab i added in, the reason for this was because i made the typo in the script, i thought that maybe if i change the typos in the SpawnManager script it would be alright but i just made it so i can't even play my game, i decided to leave it since i don't want the chance that i might break something. After that i deleted the Obstacle i had in the hierarchy, since i don't need it any more.

Inside the SpawnManager script i added two private floats: startDelay = 2; and

repeatRate = 2; in the void start i placed `InvokeRepeating("SpawnObstacle", startDelay, repeatRate);`. I made a new method called Void SpawnObsticle() and inside it typed: `Instantiate(obstaclePrefab, spawnPos, obstaclePrefab.transform.rotation);`; but before it was perfect without any typos i went through a long time of rewriting obstacles to obstacales and vice-versa until i found out that my changes, in the script, did change the spawn manager component in unity, so i made everything with the correct spelling and i had to redrag the Obstacle object to the component, i test my game and now everything works great.

After all that i made a new C# script called "RepeatBackground" and dragged it to the Background GameObajct, this will help me make the background endless, making the player run infinitely.

Inside the script i added a private Vector3 startPos; (this will be very helpful, so that i know where i should reset my background when it reaches the point that i need to put it back and make it look like it's repeating.) and in void start i added: `startPos = transform.position;`, then i made a if statement which is: `if (transform.position.x < startPos.x - 50) { transform.position = startPos; }`

All this made the Background repeat itself after it reached a certain point when moving to the left. Next i added a Box collider to my Background object, this will help me be more precise when making the background repeat it self, then i added Declare a new "private float repeatWidth" variable and i also added `repeatWidth = GetComponent<BoxCollider>().size.x / 2;` (this will make the repeating of the Background as smooth as possible), then next i replaced the hard coded 50 in the if-statement with repeatWidth. I play test my game and it works really well, it's very smooth.

In Unity i select the Ground gameobject, and in the inspector on the top that says **Tag** i click it and i have many options, nut what i want to do is to create a new one and name it "Ground". I then go to the Obstacle prefab and click open Prefab, then i go to it's Tag option, make a new Tag and name it "Obstacle", then in the "PlayerController" i added and additional public bool gameOver = false; and then i added, if (`collision.gameObject.CompareTag("Ground")`) {
 `isOnGround = true;` }

```
else if(collision.gameObject.CompareTag("Obstacle")) {  
    Debug.Log("Game Over");  
    gameOver = true; }
```

The isOnGround = true is now in the if-statement not in the void anymore.

After that, in the "MoveLeft" script i added some stuff, like a private PlayerController playerControllerScript;, in void start i added playerControllerScript = GameObject.Find("Player").GetComponent<PlayerController>();, this will connect my reference to my script to my actual player controller script in my game. Later i made a if-statement: (playerControllerScript.gameOver == false) and inside that i take the transform.Translate(Vector3.left * Time.deltaTime * speed); that was in the void Update and place it here, in the if statement.

After that i went to my SpawnManager script and add private PlayerController playerControllerScript;, then i add a playerControllerScript = GameObject.Find("Player").GetComponent<PlayerController>();, in the void Start(), these two codes will help the script identify the Player and make the Obstacles stop spawning. In the void SpawnObstacle() i added a if-statement and inside it i placed the Instantiate(obstaclePrefab, spawnPos, obstaclePrefab.transform.rotation); and removed it from its previous position within the code. I test my game and it works i get stopped when hit and the obstacles stop spawning.

Then i went back to my MoveLeft script in order to make the obstacles disappear when they are off the screen, so they don't clutter the memory of any computer, i add, inside the void Update: if (transform.position.x < -15 && gameObject.CompareTag("Obstacle")) {

Destroy(gameObject); } i then substitute the -15 with a private float leftBound = -15; so it doesn't become hard coded. The && gameObject.CompareTag("Obstacle") will make sure that the background doesn't also get deleted when moving to the left. I test this and it works, the obstacles disappear when "out of bounds".

After that i moved the ground object itself a bit to the left because when you played the game you could see a little "hole" where you see behind the background, it wasn't a big problem but i wanted to fix that and it was no trouble at all, i fixed it in seconds.

After that i double-clicked on the Player's Animation Controller, then explored the different Layers, double-clicking on States to see their animations and Transitions to see their conditions.

Later i went to the Parameters in the animator window and then i have to make the Speed_f into 1 or greater than 0.5, and then i have to set the Run_Static as a Layer Default state, meaning the animation will start immediately since it's the character's default animation now. I then increased the speed 1 to 1.5 of the Run_Static itself.

Then i went to my PlayerController C# script so i can connect the jump running animation when i press the jump button. I did this by adding in the script a new private Animator playerAnim; and a playerAnim = GetComponent<Animator>(); in the void Start, then i have to playerAnim.SetTrigger("Jump_trig"); in void Update and inside the if-statement. The way i found the "Jump_trig" is by going to the Layers in

the Animator window, then click on Jumping, click on the arrow that connect the Run to Running_Jump and then look at the inspector window at the bottom. The trig part represent trigger so because of that you have to put .SetTrigger in order for it to work. I press play and it works, the animation plays when i press Space, but it looks weird because the Player will keep running in the air after the animation stops.

Later i, in my Player Animator window in the Jumping Layer, changed the Speed value of 1 to 0.62 of the Running_Jump.I also changed the values of the mass (rigidbody), Jump Force and Gravity Modifier(Player Controller), with the values of 60, 700 and 1.65 respectively, i got these values after a lot of testing, trying to get the jump just right.

I then went to the Death layer and chose Death_01 (which it's name is Death_b, which is important) for the animation when the player hits an Obstacle. In the PlayerController script i added a playerAnim.SetBool("Death_b", true); and playerAnim.SetInt("DeathType_int", 1); inside the if-statement. I have to set "Death_b" to true because it's a boolean and i also have to set "DeathType_int" to 1 because it's a integer.

After that i added `&& !gameOver` inside the if-statement, `&& gameOver == false` is the same as `&& !gameOver` which is also the same as `&& gameOver != true`.

After all that i started adding and doing stuff with particles and sound. First i went to Course Library > Particles and dragged FX_Explosion_Smoke to the hierarchy, then i fiddled a little with the Play / Restart / Stop buttons to preview it and i played around with the settings to get my particle system the way i want it, but in the end the changed i left was the increase the speed of the Velocity over lifetime from 2 to 10, i also made sure that the Play on Awake* was unchecked. Later i made the FX_Explosion_Smoke become the Child of the Player so that the effect happens when the player does something. Then i changed the Y axis of the particle effect because it was a bit too high for me.

After that i went to my PlayerController C# script and added a public ParticleSystem explosionParticle; then in the if-statement i added `explosionParticle.Play();` after that i added the FX_Explosion_Smoke to Explosion Particle of the Player Controller component, i test my game and it works the smoke effect happens when the player hits an obstacle, when i was testing my game, the speed of the smoke seemed too fast for me, so i decreased it's speed from 10 to 5, i test again and it looks fine to me now.

I went and dragged the FX_DirtSplatter gameobject to the Player for it to become the child of it, i then had to add public ParticleSystem dirtParticle; `dirtParticle.Stop();` (inside the void Update and also with in the if-statement), `dirtParticle.Play();` (inside the private void OnCollisionEnter and if-statement) and `dirtParticle.Stop();` (inside the private void OnCollisionEnter and in the else if statement). After that i had to put the dirt splatter to the Player controller component inside the Player. I also adjusted the dirt splatter's directions since it was splatting to the left of the Player instead of behind it ,i then tested it and it works.

Later i went i clicked on the Main Camera gameobject and added a Audio Source component, i then chose cron_audio_8-bit_modern01, from Course Library > Sound > Music. Then i dragged the music clip onto the AudioClip variable in the inspector, i test my game and it

works by it could be better, i later decreased the Volume of the music from 1 to 0/5 and made the music loop by checking the loop option.

In the PlayerController script i added public AudioClip jumpSound; and public AudioClip crashSound; so that i can actually place some sound effects to my Player. then i chose the SFX i wanted to the jump and crash sound.

After that i again went to the PlayerController script to add, private AudioSource playerAudio;, playerAudio = GetComponent<

-GetComponent: GetComponent is the primary way of accessing other components. From JavaScript the type of a script is always the name of the script as seen in the project view. You can access both builtin components or scripts with this function. Basically we can get the component of something for our Player.

-ForceMode.Impulse: Apply the impulse force instantly with a single function call. This mode depends on the mass of rigidbody so more force must be applied to push or twist higher-mass objects the same amount as lower-mass objects. This mode is useful for applying forces that happen instantly, such as forces from explosions or collisions. This is similar to Transform.Translate, but instead of moving something physically just using positions, it can apply forces to things just like the real world using physics to make something move.

-Physics.Gravity: This helps us modify the gravity of a engine, i used this so for my Player to determine how much he can jumped or how fast he will fall do to gravity.

-Rigidbody constraints: This will freeze a specific axis of an GameObject, i used this so my Player doesn't topple over when it hit a obstacle, making it still.

-Rigidbody variables: Here in one of my script i changed some variable through the script, thus changing the rigidbody in unity.

-Booleans: This is a data type which is used to store true or false. I used this to make sure if the Player was on the ground or not.

-Multiply/Assign (*) Operator: I used so multiple equals, so that way i can add physics gravity is set to the gravity times our gravity modifier.

-And (&&) Operator: These (&&) are two ampersand symbols, i used this in a way to see if the space bar is down and the players on the ground. The logical AND operator (&&) returns the Boolean value TRUE if both operands are TRUE and returns FALSE otherwise. The operands are implicitly converted to type bool prior to evaluation, and the result is of type bool.

-OnCollisionEnter(): I used this Instead of using triggers to adjust the tech collisions. OnCollisionEnter is called when a collider/rigidbody has begun touching another rigidbody/collider.

-Repeat background: i made a background repeat itself by typing in the RepeatBackground C# script startPos = transform.position; repeatWidth =

```
GetComponent<BoxCollider>().size.x / 2; and if (transform.position.x < startPos.x - repeatWidth) {  
    transform.position = startPos; }, the box collider also help by making things more precise.
```

-Get Collider width: This is used to return or set the size of the BoxCollider component of a GameObject. Unity measures the size in the GameObject's local space. Changing the BoxCollider size scales it by the Transform's scale.

-Script communication: This help the code to get references of other scripts, in my scene, as well as figuring out how i can find those scripts, so that i can actually use the variables in that script,to be able to control the state of my game, using if-statements.

-Equal to (==) operator: Also know as equality operators, help you to check if something is equal to something else.This was useful to me because without it, it would be much harder to Obstacles stop spawning, i would probable needed to make a new script without it.

-Tags: A Tag is a reference word which you can assign to one or more GameObjects the fundamental object in Unity scenes, which can represent characters, props, scenery, cameras, waypoints, and more.

-CompareTag(): These let the code only do things when it has certain game objects that is's interacting with, in a way.

.size.x: size.x is the width of a collider, this helped my script to precisely find the width of my background, without it, the background would reset too early or too late.

-Animation Controllers: An Animator Controller allows you to arrange and maintain a set of Animation Clips and associated Animation Transitions

for a character or object. In most cases it is normal to have multiple animations and switch between them when certain game conditions occur.

-Animation States, Layers, and Transitions: Animation layers hold animation in your scene, with each layer containing animation curves for attributes that have been assigned to it. The animation layers appear stacked in the Animation Layer Editor, and depending on various settings, blend together to create the result animation that plays in your scene.

Animation States are the basic building blocks of an Animation State Machine. Each state contains an individual animation sequence (or blend tree) which will play while the character is in that state. When an event in the game triggers a state transition, the character will be left in a new state whose animation sequence will then take over.

Animation transitions allow the state machine to switch or blend from one animation state to another. Transitions define not only how long the blend between states should take, but also under what conditions they should activate. I used all these things to connect the animation to the scripts, adjust the value of an animation like its speed and where to find these animation.

-Animation parameters: Animation Parameters are variables that are defined within an Animator Controller that can be accessed and assigned values from scripts. This is how a script can control or affect the flow of the state machine. I used this to make my player run even faster.

-Animation programming: I learned how to control how i can activate all of those different parameters in code. How every thing connects with each other.

-SetTrigger(), SetBool(), SetInt(): I used the these triggers for the specific animations, for example: i can only put setTrigger in ("Jump_trig") because it will only accept a setTrigger because it has trig, which represents trigger, another example is "Death_b" with this i can only use SetBool().

-Not (!) operator: Example gameOver != true is the same as: if gameOver is NOT equal to true, then something will happen. Basically != means "not equal to".

-Particle systems: For effects like moving liquids, smoke, clouds, flames and magic spells, a different approach to graphics known as particle systems can be used to capture the inherent fluidity and energy. I used this in my game to make my Player "make" dirt splatter when he runs and a big smoke appear when he crashes.

-Child object positioning: I added the particle systems as the child of my Player since they are both attached now the particles will appear inside the player when he does something, because they are connected.

-Audio clips and Audio sources: The Audio Source plays back an Audio Clip in the scene. The clip can be played to an audio listener or through an audio mixer. The audio source can play any type of Audio Clip and can be configured to play these as 2D, 3D, or as a mixture (*SpatialBlend*). I used this, by adding it to my Player and Main camera gameobjects so they can actually play sound, without it they can't play any sound

Audio Clips contain the audio data used by Audio Sources. Unity supports mono, stereo and multichannel audio assets (up to eight channels). I used music and sound effect that i wanted then i placed them onto the objects they wanted and then audio source will play the sound.

-Play and stop sound effects: I used this so i can precisely see the particle effects better by playing, repeating and stopping it.


```
roller.cs ✘ SpawnManager.cs RepeatBackground.cs MoveLeft.cs
y-CSharp PlayerController
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    private Rigidbody playerRb;
    private Animator playerAnim;

    private AudioSource playerAudio;
    public ParticleSystem explosionParticle;
    public ParticleSystem dirtParticle;
    public AudioClip jumpSound;
    public AudioClip crashSound;

    public float jumpForce = 10;
    public float gravityModifier;
    public bool isOnGround = true;
    public bool gameOver = false;

    // Start is called before the first frame update
    void Start()
    {
        playerRb = GetComponent<Rigidbody>();
        playerAnim = GetComponent<Animator>();
        playerAudio = GetComponent<
```

The private AudioSource playerAudio; and the playerAudio = GetComponent() i
added to my PlayerController script.

```
28     }
29
30     // Update is called once per frame
31     void Update()
32     {
33         if (Input.GetKeyDown(KeyCode.Space) && isOnGround && !gameOver)
34         {
35             playerRb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);
36             isOnGround = false;
37             playerAnim.SetTrigger("Jump_trig");
38             dirtParticle.Stop();
39             playerAudio.PlayOneShot(jumpSound, 1.0f);
40         }
41     }
42
43
44     private void OnCollisionEnter(Collision collision)
45     {
46
47         if (collision.gameObject.CompareTag("Ground"))
48         {
49             isOnGround = true;
50             dirtParticle.Play();
51
52         } else if(collision.gameObject.CompareTag("Obstacle"))
53         {
54             Debug.Log("Game Over");
55             gameOver = true;
56             playerAnim.SetBool("Death_b", true);
57             playerAnim.SetInteger("DeathType_int", 1);
58             explosionParticle.Play();
59             dirtParticle.Stop();
60             playerAudio.PlayOneShot(crashSound, 1.0f);
61         }
62     }
63 }
```

The playerAudio.PlayOneShot(jumpSound, 1.0f); and playerAudio.PlayOneShot(crashSound, 1.0f); i added to my script.

Challenge 3 - Balloons, Bombs & Booleans:

I download the challenge 3 file into my empty Unity, i play the game and i see all these problems and errors that i have to fix.

First i can't control the Balloon which is the Player, so i went to the PlayerControllerX C# script to fix this problem. I thought about it and decided to add a playerRb = GetComponent<Rigidbody>(); in side the void Start, but turns out it didn't work, i tried some other thing to make it work like changing some public floats to private and other stuff but nothing worked. I decided to move on with the challenge, i went to MoveLeft script and i made the if statement into if (!playerControllerScript.gameOver) i simply added a (!) in the begging to represent NOT, i test my game it works the background starts when i play the game and stops when i die, but the game still has lot of errors. i try to go back and fix the player movement and while doing so and error appeared when i tried to test it and i couldn't play my game. After try to fix stuff and putting things back the way they were, even thought i can't because i think i broke something, i decided to make a new unity and re-download the challenge 3 again and do it again but this time i'll be more careful and re-read the work i did in the past to help me.

I then started doing the challenges one by one, i added the get component for the balloon to float, i made sure the back ground moves when i'm playing the game, i correctly spelled PrawnsObject to SpawnObjects so objects like money and bombs spawn, i moved the fire works particle on the Player since it was to the side and we don't want that. I changed the Y axis to the X axis so that the background repeats properly, i made the player not go too high up by making a boolean, called isLowEnough, and if-statement, i made the Player bounce off the ground and added sound for that bounce.

After that all i did more was some more testing and polishing, like adjusting some values, i made the float force for the player lower from 50 to 40, moving some objects in the correct position, deleting the two prefabs in the hierarchy, because i din't need them in the beggining since they spawn later and more.

I found a bug or a problem of some sort, when i clicked on the player object(the inspector of that object appeared) then press play the Float force was very low, very hard to go high, but if you press play without clicking on the Player, making the inspector window empty, it would be easy to play the game, the float force was very high, this was a weird bug that cost me some time because i thought i did something wrong, turns out i just can't play the game with the Player's inspector window open.

At the end of doing everything my project was finally done, after some hard thinking the challenge 3 was completed.

Lab 3:

In my personal Project i added a Rigidbody component to my Player object, then i made a new folder called "Scripts" and inside it i made a C# script called "PlayerController" after that i dragged the C# script to my Player Object. Then inside the script i'll put the movement code that i want, it will probably be the same as the guy's game in the tutorial, just horizontal and vertical movement.

I then added code to my C# script, first i added private float speed = 10.0f; and private Rigidbody playerRb; in my PlayerController script. After that in the void Start i added playerRb = GetComponent<Rigidbody>(); so the game know that the Player has the Rigidbody.

In the void Update() i added float horizontalInput = Input.GetAxis("Horizontal"); and float verticalInput = Input.GetAxis("Vertical"); so that i can press inputs for my player to move. Then i also added playerRb.AddForce(Vector3.forward * speed * verticalInput); and playerRb.AddForce(Vector3.right * speed * horizontalInput); These code will actually make the Player move. I tested my game and it worked, not only that the Player's movement was constraint because of the "fence" i added, i thought because of this that i might not need to do the next lesson of "Constraint the Player's movement" and so i asked the teacher if i need to do the tutorial of Constrain the Player's movement, and he said no and that, and i quote "less code is best code", since i had the fence up to block my Player.

Even though i didn't need to do the tutorial i did take a screenshot of how to do it.

After that i wanted to move my camera since in the game view you can't see the whole fence and enemies, so i clicked on the Main Camera object and made it higher in the air so it can bet a better view, but it also saw some outsides of the plane so i made the enemies closer to the fence, then i made the cameras FOV from 60 to 65, i still could see the outsides of the plane, the black background, i finally made the Plane larger it the X axis, i don't know why i just didn't do that in the beggining instead of unnecessarily increasing the cameras FOV, but since i already did that might as well keep it.

I then started doing clean up and export backup, i first made an empty GameObject and renamed it to "Fences", i then placed all of my fences inside it, in order to make my hierarchy more clean. Then i went to my PlayerController script and i cleaned it, i added a note or better two explaining what this code does, with some "arrow" where the code is, the reason i add two notes and arrow is to make sure i can show people who read this script where the codes are and what they do, and it's also to help me to make a "block" of code separated from other codes making it easier to find problems.

I then tried to copy the what the guy was doing, i tried to move a block of code into a new void function called MovePlayer(), but for some reason it didn't work, i even tried to play my game and it wouldn't allow me to play, so i decided to simply added notes to my block of code so it can become more readable.

Later when every thing was fine i just remembered to do something that i forgot to do and that was to freeze the rotations of the X and Z axis. After i did that i tested my game and i noticed a small change in the movement, there were less momentum, since there was no rotation. Before if you stopped moving the Player would still move a little bit because of the momentum, now after the changes you can stop completely.

After that i exported my game and placed it in my BackUps folder and named it "Diogo_V0.2".

-Program in C# independently: I referenced some of our old code that wrote, and followed the instructions from the guy doing the Lab 3.

-Troubleshoot issues independently: I solved the problems i faced except the part in the cleaning of the code i didn't fix what i experienced even though i should but i thought it was unnecessary.

Quiz 3:

On question 1 i chose the third option but it was the second option, i made a mistake because i forgot that the bool code need to be public.

On question 3 i also got it wrong because i chose `setInt("Speed_f", 1);` since 1 is greater than 0.25 but i can't choose a integer number. the right answer was `setFloat("Speed_f", 0.3f);`

On question 5 i also got it wrong because i chose "The Enemy object does not have a Rigidbody component" but the answer was "There is no object named "Player" in the scene" since you can't "Find" an object that is not in the scene, you will get a "NullReferenceException" error.

I failed question 8 because i didn't understand the question it was given to me, i thought it wanted me to find the ones that are wrong.

I also made a mistake on question 9 since i thought it was from running to walking so i chose Speed_f is Greater than 0.5 when it was Speed_f is Less than 0.5.

The rest i had it correct especially the animator ones when easy.

Create with Code 4:

I made a new Unity game and called it Prototype 4, then i downloaded and imported the files to make my game from the direct download.zip, clicked on prototype 4 and deleted the SampleScene.

Then i left-clicked on the hierarchy window and make a 3D sphere and renamed it to "Player" and reset it's position and increased it's XYZ scale from 1 to 1.5. Later i added a Rigidbody to the Player and also added a texture to my Player called PolygonPrototype_Texture_Grid_Glass_01, which is the black one with white lines, i did this by going to Library > Textures and dragging the texture i want to my Player. I then tested to see if everything was working and it was, nothing is broken and i can see the fog below the island.

After that i created a focal point for my camera, i made an empty gameobject in the hierarchy and renamed it to "Focal Point", i then placed it in the Main Camera Object so it can become it's child, i also reset the position of the Focal Point. Later i made a C# script called "RotateCamera" and attach it to the Focal Point.

Then inside the "RotateCamera" script i added a public float rotationSpeed; and then i added inside the void Update, float horizontalInput = Input.GetAxis("Horizontal"); so i can control the camera itself, transform.Rotate(Vector3.up, horizontalInput * rotationSpeed * Time.deltaTime); is to make the camera actually move. Then in Unity i added 40 to be the value of the Rotation speed in the Rotate Camera component, i test my game and it doesn't work, i look back at my script and everything is fine, i rewatched a previous video lesson on created with code 4 and i realized what the problem was, the Main Camera was suppose to be the child of the Float Point, not the other way around, i do that, and i test my game and now it works and also the speed of the rotation is great so i'm going to keep the speed at 40.

I made a new C# script called "PlayerController", i dragged it to the Player GameObject then inside the script i added two floats: private Rigidbody playerRb; and public float speed = 5.0f;. I also added playerRb = GetComponent<Rigidbody>(); in the void Start, then i added float forwardInput = Input.GetAxis("Vertical"); and

playerRb.AddForce(Vector3.forward * forwardInput * speed); so that the force is actually applied to the Player. I then went to Unity and tested my game and it didn't work, I couldn't move the Player. I went to the Player's inspector window because I thought I might have forgotten to add the PlayerController script to it, just to make sure, but turns out the speed value was 0, even though I made a public float speed = 5.0f; with the value of 5, but turns out it didn't add it to Unity, I'll have to check if values are added in Unity in the future. I added 5 to the speed and test my game and it works.

Later, I also added in my script a private GameObject focalPoint; then in the void Start I added focalPoint = GameObject.Find("Focal Point"); after that, I Replaced Vector3.forward with focalPoint.transform.forward, this will make so I move in the direction that the camera is facing, I test it and it works, although it's a bit hard to control.

After all that I created a new Sphere, rename it "Enemy" reposition it, and dragged a texture onto it, specifically a white texture(PolygonPrototype_Texture_Grid_05). I then added a new RigidBody component and adjust its XYZ scale, then tested it and it works, the enemy moves when the Player hits it. I then made a new folder called "Physics Materials", then inside it I created a Physics Material and named it "Bouncy", in "Bouncy" I increased its Bounciness to, changed Bounce Combine to "Multiply" and then I applied it to my player and enemy. I then created a new C# script called "Enemy", attached it to the Enemy GameObject, inside it I added public float speed = 3.0f; private Rigidbody enemyRb; and private GameObject player; then in the void Start I added enemyRb = GetComponent<Rigidbody>(); and

player = GameObject.Find("Player"); then in the void Update I added enemyRb.AddForce((player.transform.position - transform.position) * speed); this will apply the force that the Enemy needs. But it goes very fast, the reason for this is because as the player gets farther away from the enemy, the length of that actual vector, the distance between those two positions becomes larger and so what it actually does, is it tries to multiply it over a very large amount of distance, so it actually increases in force over time. And to fix it I changed my code I just did, by adding a standard strength, and make it into this: enemyRb.AddForce((player.transform.position - transform.position).normalized * speed); this will make the Enemy not go as fast when after the Player, I test this and it works.

Later I cleaned my Enemy script by adding one variable, which is: Vector3 lookDirection and inside it I added (player.transform.position - transform.position).normalized; I then replaced where this code was with lookDirection.

Then I dragged the Enemy into the Prefabs folder, which I created, to create a new Prefab, then delete Enemy from scene. I created a new "Spawn Manager" object, attach a new "SpawnManager" C# script, and open it, inside it I declared a new public GameObject enemyPrefab variable then assign the prefab in the inspector, later I added in the void Start, Instantiate(enemyPrefab, new Vector3(0, 0, 6), enemyPrefab.transform.rotation); I then went to Unity and pressed play and it worked a enemy spawns and tries to chase me.

After that I first made a private float spawnRange = 9.0f, I did this first unlike the guy in the video lessons who did that later in his code after putting hard coded numbers. Then I made float spawnPosX = Random.Range(-spawnRange, spawnRange); and float spawnPosZ =

Random.Range(-spawnRange, spawnRange); to show where the enemy can spawn, later i added Vector3 randomPos = new Vector3(spawnPosX, 0, spawnPosZ); and replaced the new Vector3 in the Instantiate with randomPos, to make the code more clean, i tested this and it works, every time i press play the enemy spawns in a different location.

I then created a new function Vector3 GenerateSpawnPosition() and inside it is what i added before in the void Start except for the Instantiate, also in this code, below it i added return randomPos; this will "activate" the Vector3 GenerateSpawnPosition(). I replaced the code in my Instantiate call with my new function name: GenerateSpawnPosition(), specifically i replaced the randomPos in it. I test my game to see if it works and it works just fine, later i change the camera's speed from 40 to 55 because after a lot of tests with this speed i realized it becomes difficult to control the Player like that, a high speed is easier to control, this is what i realized after some tests with this speed.

After all that i went to Course Library > Pickups and then i choose Gem_01, dragged it to my scene then i renamed it to "Powerup", i then scaled it from 1 to 2, and then i reset its position and made the Z axis 4. Later i added a Box collider to it and made sure the "Is Trigger" checkbox is checked. I then created a new "Powerup" tag and apply it to the powerup, then i dragged the Powerup into the Prefabs folder to create a new "Original Prefab".

In my PlayerController script i added a public bool hasPowerup; then i made a private void OnTriggerEnter(Collider other) inside of that i made a if-statement:

```
if (other.CompareTag("Powerup")) { hasPowerup = true; Destroy(other.gameObject);}
```

The if statement will allow Unity to know if the Player has a Powerup or not and if it does the Powerup will be destroyed. I then tested this and it works.

After that i created a new "Enemy" tag and applied it to the Enemy Prefab, then i went to my PlayerController script and added a private void OnCollisionEnter (Collision collision) {} and inside that was { Debug.Log("Collided with: " + collision.gameObject.name + "with powerup set to " + hasPowerup); } this will let me know if a collision happened and if i had a powerup or not. I check in the tutorial video if i did everything all right, and i realized i forgot to add the if statement which is if(collision.gameObject.CompareTag("Enemy") && hasPowerup), i added the if statement, and i realized that the allows the code to only display a "Log" when the Player collides with a Enemy, and the info of the Log will be different if the Player has a powerup or not, this is . Later i tested my game and the Player collected the powerup and then hit the Enemy a Log appeared below. I realized something interesting, when i tested my game before i had the if-statement in my code, after the Player collected the powerup a Log appeared below, it said: "Collided with: Island with powerup set to False" and now that i had the if-statement, it no longer appears when i collide with the powerup.

I then added, in my PlayerController script, a private float powerupStrength = 15.0f; then i added this:

```
Rigidbody enemyRigidbody = collision.gameObject.GetComponent< Rigidbody>();  
Vector3 awayFromPlayer = collision.gameObject.transform.position - transform.position;  
enemyRigidbody.AddForce(awayFromPlayer * powerupStrength, ForceMode.Impulse);
```

In my if (collision.gameObject.CompareTag("Enemy") && hasPowerup). I tested my game and it works, the Enemy bounces off far and fast from the opposite direction of my Player.

Later i added in my PlayerController script, a IEnumerator PowerupCountdownRoutine() and inside it i added yield return new WaitForSeconds(7); and hasPowerup = false; these codes will make the powerup disappear after 7 seconds. Then in the private void OnTriggerEnter(Collider other) i added a StartCoroutine(PowerupCountdownRoutine()); this code will make the timer possible, it will activate the count down timer. I tested my game and it works the powerup wears off after 7 seconds.

After that i dragged an object, from the library, to the hierarchy and increased it's X and Z scale from 1 to 3 and i lowered it's position to be near the ground. I learned what the check box, the box near the name in the inspector window at the top, does it sets an GameObject as active or inactive, when it's off it disappears, like it never existed, i unchecked my Powerup Indicator.

I then i added a public GameObject powerupIndicator; in my PlayerController, then i dragged the Powerup Indicator to the Player's inspector window in the empty component that i made thanks to what i just made in the code. Later i added powerupIndicator.gameObject.SetActive(true); in the if (other.CompareTag("Powerup")) statement, this code will make the circle appear when the Player collects the powerup, i also added powerupIndicator.gameObject.SetActive(false); in the IEnumerator PowerupCountdownRoutine() this will make the circle disappear after the timer has ended. I tested all this and it works, the circle appears after i touched the powerup and it disappears after 7 seconds. Later i added more code in my script like: powerupIndicator.transform.position = transform.position + new Vector3(0, -0.5f, 0); in the void Update, this code will make the powerup indicator follow the Player, i tested this in Unity and it works a circle appears, follows the Player and then disappears after 7 seconds.

Then i, in my SpawnManager C# script, added a for-loop in my code, in void Start(), replaced the single Instantiation with a for-loop that spawns 3 enemies, then i Moved the for-loop to a new void SpawnEnemyWave() function, that i made, then called that function from Start(). I test this and it works, 3 enemies spawn when i pressed play.

Later i, in my SpawnManager script, added a parameter int enemiesToSpawn to the SpawnEnemyWave function, i replaced the 3 it had with from i < __ to i < enemiesToSpawn, then i added a 3, in the brackets, to the SpawnEnemyWave() i tested this in my game to see if it works and it works fine, everything is alright.

I then added if(transform.position.y < -10) {Destroy(gameObject);} in the void Update. Then i added, in the SpawnManager C# script, a variable public int enemyCount;, then i added a enemyCount = FindObjectsOfType<Enemy>().Length; and if(enemyCount == 0) {SpawnEnemyWave(1);} these two where in the void Update. I test all this and it works, one enemy appears when there are zero enemies left.

After that in the SpawnManager script i made a public int waveNumber = 1; variable, and i also changed inside the if-statement to waveNumber++; SpawnEnemyWave(waveNumber); this will make 1+ enemy when there are 0 enemies left, so each "round" will have 1+ enemy. I play test this and it works i even went to fighting 4 enemies at a time.

Later i added a public GameObject powerupPrefab; with this i needed to go back to unity and place a prefab of the powerup into the Spawn Manager component and into an empty slot, i also delete the Powerup in the hierarchy since it's unnecessary now. In the script i also simply added the Instantiate(enemyPrefab, GenerateSpawnPosition(), enemyPrefab.transform.rotation); into the void Start and Update, this will make the spawning of the powerup possible, i test this and there was two enemy instead of one enemy and a powerup, i went to my script and realized the problem, i had enemyPrefab instead of powerupPrefab in my code, in the void Start and Update, i fix that and i test again, and now it works great i played around and had fun.

-Texture Wraps: I "wrapped" my Sphere in a texture of my choosing, and i just dragged and drop the texture i wanted to my Object of choosing, here i chose my Player.

-Camera as child object: I made an empty gameobject then i dragged the camera into that empty gameobject, making the camera its child.

-Global vs Local coordinates: I learned with the focal point, the difference between a global and a local direction, so that way, when i actually want to get the direction that the actual focal point is pointing towards, i can use the local position, but if i ever want to reposition the focal point to a different place in the world i can use the global location to actually move it around and set it to a specific place that i want it to.

I used a video to help me understand Global and Local space (<https://www.youtube.com/watch?v=DosisUyIYIk>) which helped me a lot.

-Get direction of other object: In my script i learned how to get a reference for my focal points that i could actually use its Local direction to move the player in that direction.

-Physics Materials: Physic Materials are used to adjust friction and bouncing effects of colliding objects. To create a Physic Material select Assets > Create > Physic Material from the menu bar. Then drag the Physic Material from the Project View onto a Collider in the scene. I use a physics materials to add a extra bounce on my Player and Enemy objects.

-Defining vectors in 3D space: With this i did a little bit of vector math, subtracting the players position by the enemy's current position.

-Normalizing values: When normalized, a vector keeps the same direction but its length is 1.0. If this vector is too small to be normalized it will be set to zero. I used this in my game in a way so that way the enemy doesn't just speed directly at the player.

-Methods with return values: I used this so that my code is able to return values that i need to use whenever i need data back from them, like my generate spawn position. That gives me a Vector3 whenever i call that method.

-Bounciness property and Bounce Combine: The physics materials have multiple property like: Dynamic Friction(The friction used when already moving. Usually a value from 0 to 1. A value of zero feels like ice, a value of 1 will make it come to rest very quickly unless a lot of force or gravity pushes the object), Static Friction(The friction used when an object is laying still on a surface. Usually a value from 0 to 1. A value of zero feels like ice, a value of 1 will make it very hard to get the object moving), Bounciness(How bouncy is the surface? A value of 0 will not bounce. A value of 1 will bounce without any loss of energy), Friction Combine(How the friction of two colliding objects is combined), Bounce Combine(How the bounciness of two colliding objects is combined. It has the same modes as Friction Combine Mode) and much more.

-Functions that return a value: Methods that return a value, has the purpose of giving us information after a calculation. An example in my case is that GenerateSpawnPosition, actually gives me a position that i need to use. In fuctions that return a value we need to add a specific type of data that we want to come out of this method, e.g. Vector3, outputType, etc.

-Debug concatenation: I used this in a way i could send myself more accurate messages whenever something happens, using strings and variables.

-Local component variables: I used more Variables by using the enemy's rigid- body when the Player collides with them, i was able to get their component, as well as getting their position.

-IEnumerators and WaitForSeconds(): IEnumerator is a .NET type that is used to fragment large collection or files, or simply to pause an iteration. WaitForSeconds suspends the coroutine execution for the given amount of seconds using scaled time. I used these to make a count down timer involving my powerup.

-Coroutines: Coroutine is a Unity type that is used to create parallel actions returning a IEnumerator to do so. I used this to to actually start a method, so that way i can actually have my countdown timer happen.

-SetActive(true/false): This basically Activates or deactivates an object, where true activates the GameObject and false deactivates the GameObject. I used this in my code to make the Powerup Indicator on or off depending on what the Player has done.

-For-loops: A For-loop is a programming language conditional iterative statement which is used to check for certain conditions and then repeatedly execute a block of code as long as those conditions are met. I used this in my code to keep control over the number of times i want something to execute, basically so the code can spawn more and more enemies when the Player defeats one.

-Increment (++) operator: A increment (++) operator increases the value of a variable by 1 and decrement (--) operator decreases the value of a variable by 1. I used this to actually make the increase of the numbers of enemies spawned after the Player defeats them.

-Custom methods with parameters: I made a custom method called void SpawnEnemyWave() but this time with a parameter to pass in, so then that way the script could keep track of the number of enemies to spawn based on the wave number.

-FindObjectsOfType: Returns a list of all active loaded objects of Type type. It will return no assets (meshes, textures, prefabs, etc) or inactive objects. I used this so that i could get any object with a type of our enemies script. So then i could actually get the number of enemies that are in our scene.

The screenshot shows the Unity Editor's code editor with the file `Enemy.cs` open. The tab bar at the top shows `PlayerController.cs` and `RotateCamera.cs` are also open. The code itself is as follows:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Enemy : MonoBehaviour
{
    public float speed = 3.0f;
    private Rigidbody enemyRb;
    private GameObject player;

    // Start is called before the first frame update
    void Start()
    {
        enemyRb = GetComponent<Rigidbody>();
        player = GameObject.Find("Player");
    }

    // Update is called once per frame
    void Update()
    {
        Vector3 lookDirection = (player.transform.position - transform.position).normalized;

        enemyRb.AddForce(lookDirection * speed);

        if(transform.position.y < -10)
        {
            Destroy(gameObject);
        }
    }
}
```

What i did in my Enemy script, i added and if-statement, in order for my Enemy object be destroyed when it goes off the cliff.

The screenshot shows the same `Enemy.cs` script with a syntax error. Line 20 contains the code `enemyCount = FindObjectOfType<Enemy>().Length;`. The word `Length` is underlined with a red squiggly line, indicating a spelling mistake. The code block is as follows:

```
16
17     // Update is called once per frame
18     void Update()
19     {
20         enemyCount = FindObjectOfType<Enemy>().Length;
21
22         if(enemyCount == 0)
```

I made a mistake when doing this code, when typing visual studio will let you autocorrect for you and i accidentally choose the wrong one, i choose the "FindObjectOfType" when it show have been "FindObjectsOfType" with an s, i'll have to be more careful next time. Even the smallest things can ruin my code.

The screenshot shows the Unity Editor's code editor window. The tab bar at the top has tabs for "SpawnManager.cs", "Enemy.cs", "PlayerController.cs", and "RotateCamera.cs". Below the tabs, there's a breadcrumb navigation bar with "Assembly-CSharp" and "SpawnManager". On the right side of the editor, there's a search bar with the placeholder "Update()". The main area displays the C# code for the "SpawnManager" script:

```
6
7     {
8         public GameObject enemyPrefab;
9         private float spawnRange = 9.0f;
10        public int enemyCount;
11
12        // Start is called before the first frame update
13        void Start()
14        {
15            SpawnEnemyWave(3);
16
17        // Update is called once per frame
18        void Update()
19        {
20            enemyCount = FindObjectsOfType<Enemy>().Length;
21
22            if(enemyCount == 0)
23            {
24                SpawnEnemyWave(1);
25            }
26        }
27
28        void SpawnEnemyWave(int enemiesToSpawn)
29        {
30            for (int i = 0; i < enemiesToSpawn; i++)
31            {
32                Instantiate(enemyPrefab, GenerateSpawnPosition(), enemyPrefab.transform.rotation);
33            }
34        }

```

The additions I made in my SpawnManager script, so that my project makes one new enemy when their are 0 enemies left.

```
SpawnManager.cs  X  Enemy.cs  PlayerController.cs  RotateCamera.cs
Assembly-CSharp  ▾  SpawnManager

6   {
7     public GameObject enemyPrefab;
8     private float spawnRange = 9.0f;
9     public int enemyCount;
10    public int waveNumber = 1;
11
12    // Start is called before the first frame update
13    void Start()
14    {
15      SpawnEnemyWave(waveNumber);
16    }
17
18    // Update is called once per frame
19    void Update()
20    {
21      enemyCount = FindObjectsOfType<Enemy>().Length;
22
23      if(enemyCount == 0)
24      {
25        waveNumber++;
26        SpawnEnemyWave(waveNumber);
27      }
28    }
}
```

The additions I made in my SpawnManager script in order to spawn 1+ more enemy after the Player has defeated them.

Quiz 4:

I failed question 1 because i didn't know you had to add "return age;".

I also failed question 7 because i thought i needed a void if i'm to create a method, but turns out i don't.

Question 8 was failed because i didn't realize the "public class Enemy" and since it said "if(other.CompareTag("Spike"));" i thought it was any object but the Enemy.

With question 10, i failed it because, D and C i knew were wrong but with the others i wasn't so sure, i did consider the, they are all wrong option but i choose the first option because i thought that A had to be right but i was wrong.

Challenge 4 - Soccer Scripting:

I first try to fix the problem of hitting an enemy sends it back towards the Player. I went to the PlayerControllerX.cs and i tried to add an enemy and player, in the Vector3 awayFromPlayer, in front of the transform.position, but it didn't work the code even showed an error. I then thought about it and looked back at my previous work, then i tried to simply move the other.gameObject to the left so it be

subtracted, like this (other.gameObject.transform.position - transform.position), no error appeared so i test the game and it worked, i'm surprised that the answer can be that simple.

Later i went to the SpawnManagerX script to fix enemies spawning when you pick the powerup. I looked at the code and i found the problem, in the void Update, in the enemyCount it had "Powerup" instead of "Enemy" so i do that and then i test the game and it works, enemies spawn when there are no enemies left.

I then went back to the PlayerControllerX and look at the IEnumerator and i started doing small modifications to it, like changing setActive to true, adding a number in front of PowerupCooldown, and some other stuff, all of which didn't work, so i look at my previous work and i found out that i need to add StartCoroutine(PowerupCooldown()); in my code, so i add that in the IEnumerator and it still doesn't work when i tested it, so i looked more closely into my previous work and lessons and i realized that i have to place the code into the if (other.gameObject.CompareTag("Powerup")) not IEnumerator, after i did that the timer works now.

After that in the SpawnManagerX i figured the problem of two enemies spawning in the beginning, the for (int i = 0; i < 2; i++) had a tow in the middle, so i thought to fixed this was to simply change the 2 to 1, i did that then i tested the game, it appeared 1 enemy but when i defeated it only 1 appeared instead of tow, there should be 1+ enemy for every "round", i dig deeper into the code to see how to fix it and i found out that i need to substitute the 1 from i < 1 to enemiesToSpawn, which is the code that makes 1+ more enemy every new "round", i did that then i tested it and it now works perfectly.

Later i went to the EnemyX script to fix the problem of the enemies not moving to the Player's goal. I looked at the code, i tried to do some things in the void Update because of the comment, it didn't work, so i went back to the lesson 4 to see what i did, and i realized i had to add playerGoal = GameObject.Find("Player Goal"); which makes sense because the "playerGoal" isn't really showed what it is or what it does, and GameObject.Find makes the enemy go and find a specific object. And inside the brackets with quotations you put what object the Enemy needs to find. I did this but for some reason it didn't work, i relooked at my previous works to see if i misread but it looked fine to me, i tried to add a value to speed, because it didn't have one, i tested it and it didn't work. I even changed the way the grammar was spelled for "Player Goal", i took out the space, changed some big letters to small letters, i even tried to add a new tag called "Player Goal", but nothing worked, i placed everything in the way they were, except the playerGoal in the void Start, and started thinking again to figure out the problem. I went to unity and clicked on the Enemy prefab and went to the EnemyX component to see if there was anything wrong with it, and i saw that there wasn't a value on the speed component, i thought to myself "this isn't right", i added a value in the script and it still didn't work, so i decided to add a value, like 100, to the speed, i pressed play and then it finally worked, the enemies went after the player's goal. I did some more test and changed to value from 100 to 50, since for me 100 was a bit too fast and hard.

Next i decided to try to add a speedboost to the Player when you press the Space button. I went to the PlayerControllerX to do this, and created a new private float speedBoost = 10; then i made a if-statement in the void Update which is: if (Input.GetKeyDown(KeyCode.Space)), this code will make it possible for the space button to do something, later in it i copied

playerRb.AddForce(focalPoint.transform.forward * verticalInput * speedBoost * Time.deltaTime); and added to the if statement, i test my game and i think it didn't work so, maybe i thought to myself i need to modify this code, i went back to my previous works and i googled a bit and i found out that this code works: playerRb.AddForce(focalPoint.transform.forward * speedBoost, ForceMode.Impulse); i think it makes sense because the hint told me that i needed some type of impulse, so that helped. I test this new code and it works the Player goes really fast and i can do it as many times as i want, but i did find a sort of glitch, when the Player hits a wall at a very high speed you go through it.

I then tried to add a particle affect to the speed boost, what i tried to do was, add a new public float GameObject Smoke; but when i did this the code didn't like what i did, saying that there is something wrong here, i later realized that i can't place a float there so i take it out, now it's fine. I then went to the void Update and added Smoke.Play(); to it, but there was an error in Unity and in the code about the ".Play" i couldn't even play the game, so i tried to figure out the problem, i placed my mouse on the .Play with the red line underneath it so i could get some info about it, since visual studios will usually tell you that somethings wrong with the code, it said: "GameObject does not contain a definition of "Play"....". I thought about it and i realized that the problem was that you can't use ".Play" on GameObject, it had to be a ParticleSystem. I went and changed it so now it looks like this: public ParticleSystem Smoke; then i place the Smoke_Particle as the child of the Focal Point, since i forgot to do that sooner, then i added that Particle in a empty slot in the PlayerControllerX component, i test it and the smoke appears although it's far from the player so i drag the Smoke_Particle into the Player, and lower it a bit so it looks like the smoke is coming from the ground and not from the Player.

I tried to the the last bonus challenge, i went to previous works and even went to the internet but i gave up, i decided to not wast too much time into this. I had other stuff to do.

Lab 4:

I created a new GameObject called "Bullet" in the form of a yellow 3D capsule, i decreased it's size, made a new material for it and also made it's rotation from the Z axis 90 so it becomes horizontal. I did this because i think i need something for my enemies to shoot, just like how the Player from create with code 3 needed a type of food to shoot.

Later i added a Rigidbody to my Enemy object and freezed it's Y position, then i made a new C# script called "MoveAround" apply it to my Enemy. Then i made all my Enemies and Player Y position from 1 to 0.5 because i didn't like the way they dropped when you start the game. Then in the script i added public float speed = 5.0f; and private Rigidbody objectRb; then in the void Start i added objectRb = GetComponent<Rigidbody>(); and in the void Update i added objectRb.AddForce(Vector3.forward * -speed); just to see if it works, i tested it and the Enemies didn't move, i went to my code and rewatched the video about basic movement and i did everything correct, i tried to change the -speed to speed and that still didn't work, so i thought to myself that maybe it was because the enemies being too close to the ground, so it might be clipping, and so i increased it Y position and then i pressed play and now it works, the problem was that it was too close to the ground making it stuck, i left the enemies Y position to be 0.55 it seems good for me. . Then later i enabled the Is Trigger for the Bullet Object because i want it to interact with the Player, meaning taking the life of the Player.

Something i realized after doing some tests with the Enemies movement, i realized that if you don't freeze the Y position of an enemy Object in their Rigidbody, the enemy doesn't move for some reason, it only moves after the Y position is frozen.

Next what i had to do in the lesson was "Destroy objects off-screen", i only have one use for this, and that's for my Bullets coming from the enemies. I made a new C# script called "OutOfBounds", dragged it to my Bullet GameObject, and inside it i added two variables: private float zDestroy = 30; and private float xDestroy = 10; these are the limits of where the bullets can go, then i added two if-statements: if(transform.position.z > zDestroy){Destroy(gameObject); the outer if statement has the x instead of the z.

I went to Unity to test this, and a simple way to test this instead of making new code for the bullet to move, is to move the the bullet manually while you are in play mode, you have to be on the Scene tab to do this, i did this and i can test my bullet without having to add code. When i tested my bullet i noticed that it got removed far up and close to the right, when in fact it should be the opposite. It was a simple fix, i just had to swap the values in the code.

I didn't like that i had two if-statements, so i went to the internet to figure out a way for my if statements to do multiple things in it, and i found out that you can put a sort of OR logic in code which is (||), || = OR, this was the site i found this information (<https://answers.unity.com/questions/1230388/how-to-destroy-object-after-it-moves-out-of-screen.html>) this was quite helpful. With this my code won't be so messy, now all i had to do was add another if-statement but this one will have the negative value like this: if(transform.position.z < -zDestroy || transform.position.x < -xDestroy) and with the Destroy(gameObject); now the Bullet will disappear when it reaches out of bound from all sides, not just from the top and right any more. Although i do wish i did a better job at making this code short and simple, because i think there is a simpler way of doing this, i don't know if i should just placed all the code in one if statement, i don't want a single code to be too long, i left the code the way it is because i don't want to waste too much time optimizing a single code. I tested my Bullet if it disappears on all four sides if it goes far enough and it does.

I then added tags to almost all my GameObjects, i added the Player tag to my Player, i made a new tag called "Enemy" and added it to my Enemy object, i made "Fence" and gave it to Fences and i made "Bullet" and gave it to my Bullet object The Fence tag was maybe unnecessary but i made it nonetheless.

```
In the PlayerController script i added private void OnCollisionEnter(Collision collision) { if (collision.gameObject.CompareTag("Bullet"))
{Debug.Log("Player has collided with bullet")}}
```

This will let me know if my Bullet has collided with my Player, later also in my script i added something that will destroy the bullet when it comes in contact with the Player, that something is: private void OnTriggerEnter(Collider other) inside of it is a if (other.gameObject.CompareTag("Bullet")) { Destroy(other.gameObject); } this code will actually destroy the bullet. I test this in my game and it works, the Bullet disappears after my Player comes in contact with it.

I didn't add a Physics material in my game because my game doesn't need one, i also deleted 3 of 4 of my enemies, because i just need one really good enemy and then make it into a prefab and then make more if i need so. Later while i was testing my game some more i realized that the Debug.Log didn't appear, when the Player contacted with the bullet, i believe the reason for this is because the bullet gets destroyed before the code can deliver the message, i thought about deleting that part of the code, but then i thought i might reuse it for

something different. Instead of when the Player hit the bullet, it could be for when the Player hit the fence, that's when the Debug.Log shows. I made some modifications to my code, the if (collision.gameObject.CompareTag("Bullet")) { Debug.Log("Player has collided with bullet.");} is now: if (collision.gameObject.CompareTag("Fence")) {Debug.Log("Player has collided with fence.");} i test this and at first it didn't work so i thought that maybe i have to add the Fence tag to all my Fence objects, of which are the child of Fences, i did this and it now works.

After that i made a new folder called "Prefabs" and dragged my Enemy and Bullet GameObject into it. Then i tested my game and to see if they still work i dragged the Bullet from the Prefabs folder to my scene while the game was running, and everything worked fine. I still left my prefabs in the hierarchy but i will delete them later when my game is almost finish.

Later i finally made a new script called "BulletMove", this script will make the Bullet follow the Player, inside this script is: public float speed = 3.0f; private Rigidbody bulletRb; and private GameObject player;, I gave my Bullet a Rigidbody for this, but its gravity is turned off. In the void Start i added bulletRb = GetComponent<Rigidbody>(); and player = GameObject.Find("Player"); then in the void Update i added Vector3 lookDirection = (player.transform.position - transform.position).normalized; and bulletRb.AddForce(lookDirection * speed); all this will make the Bullet follow after the Player specifically. I had to look up my previous work with create with code 4 in order to do this, which helped my a lot.

Then i created a new empty gameObject called "Spawn Manager", then i made a new C# script called "SpawnManager" and inside it i started typing what the guy from the lab 4 lesson was doing, but half way through i thought to myself that i didn't want this type of spawning. I wanted my spawn manager to spawn the bullets from the Enemies, so i went to the internet to look for some solutions, and i found some, i found a video about how to spawn objects from prefabs, called "Unity3D 101: Spawning Objects from prefabs, resources, and pools" this is the video(<https://www.youtube.com/watch?v=9KOHclqSmR4>), i basically copied the code it had, about spawning objects from prefabs, i did struggle a bit making the spawning just right but eventually i got it. This code didn't make the bullets spawn from my enemies but it did spawn from the Spawn Manager object itself which is close enough for me, because i can simply position the Spawn Manager GameObject to be the same as the Enemy GameObject.

I tested this in my game, and a crazy bug appeared, hundreds of bullets were being spawned every second, it looked like a yellow snake. I went to my script to figure out the problem and i found out that the problem was, i had added SpawnBullet() into the void Update, and the void Update updates every frame the spawn manager was spawning bullets every frame. I solved this problem by simply removing the SpawnBullet() from the void Update, turns out that the SpawnBullet() in the void Update was unnecessary.

After i solve the spawning problem and tested my game, another problem arrived, the bullets spawning after the first spawn were vertical and not horizontal, their rotation were different. I looked at my code to see where the issue was, then i went to the internet to look for solutions and i found a site about some people that had the same issue as i'm having, the site was this (<https://answers.unity.com/questions/146754/object-rotation-not-working-as-such.html?page=1&pageSize=5&sort=votes>), the people in

this site had multiple ways for solving this problem and i chose the placing an object into a empty object and making that into a prefab. At first i did what the solution said and it worked, when the bullets spawn they appeared horizontally but i gained another problem, that was the Bullet disappeared but not the empty object i made, it even wouldn't be destroyed after i dragged the OutOfBounds script into it. I tried a lot of other stuff in order to make it disappear and work properly, like adding the same tag as my Bullet object has, changing the code a bit and even swapping the objects making the Bullet the child or the parent with my empty object, but nothing worked. I decided to go back to the way of my bullets spawning into vertical bullets, it may look weird but at least it works and i don't want to mess or break my game.

After all that i stopped messing with my SpawnManager script and started improving and polishing my game in Unity. I duplicated my Enemy so now i have four enemies, 1 up top, 1 left, 1 right and 1 bottom, then i duplicated mt Spawn Manager GameObject from the hierarchy, with this there will be multiple bullets after the Player which is more challenging, i placed the positioned of the Spawn Managers to be the same as the Enemies, so it looks like the enemies are shooting. I play tested my game and it works well, it may not be great but it's functional, after i did my tests i exported my assets and named them "Diogo_V0.3".

-Non-player objects prefabs have basic movement: I added my Enemy some movements, but later i didn't want my Enemies to move so i took that out of them. But i did make my Bullet follow the Player so it look like it's attacking it.

-Objects are destroyed when they leave the screen: I made a script that can make an object disappear after i goes past a certain distance from the X and Z axis.

-Collisions between objects are handled appropriately: I made so that the Player and the Fences collide properly, i also made a Debug.Log to see if the Player collided with the Fence. I made so that when a Bullet collides with the Player it gets destroyed.

-Objects are spawned at the appropriate locations on time-based intervals: I made a Spawn Manager so that it can spawn multiple bullets that attack the player. I made four of them and placed them with their corresponding Enemies, which are four also, making it look like the enemies are shooting the bullets.

Create with code 5:

I made a new empty 3D Unity game in the unity hub and named this "Prototype 5", i installed the file for the crate with code 5 and added it to my unity.

I first double-clicked on Prototype in the Assets folder and deleted the sample scene, then i clicked on the 2D icon in scene view to put my scene view in 2D, i decided to not change the visual of the borders and background.

Later i, from the Library, dragged 3 "good" objects, a fish cut in half, a apple and a sandwich, into my scene. And i also dragged a 1 "bad" object which i a bomb, into my scene, i then renamed them "Good 1", "Good 2", "Good 3", and "Bad 1", i later added Rigid Body and Box

Collider components, then i made a Scripts folder in which i created a new "Target" C# script, i attached it to my Target objects. At last i dragged all 4 targets into the Prefabs folder to create "original prefabs", then delete the ones in the hierarchy.

I then, in the Target script, declared a new private Rigidbody targetRb; and initialize it in Start(), in Start() i added targetRb = GetComponent<Rigidbody>(); this will get the Rigidbody that the code need from the objects. Then i added targetRb.AddForce(Vector3.up * Random.Range(12, 16), ForceMode.Impulse); this will make the objects go up with variant degrees of force, weak or strong, still in he void Start() i added targetRb.AddTorque(Random.Range(-10, 10), Random.Range(-10, 10), Random.Range(-10, 10), ForceMode.Impulse); this will make the objects rotate in different ways, and finally in the Start() i typed transform.position = new Vector3(Random.Range(-4, 4), -6); i tested this in Unity by being in play mode and dragging my prefabs into the hierarchy, and it works the objects went up, some higher, some lower, and all had different torque, some torqued to the left, right, up or down, and all their position when they spawned were different every time.

After that i cleaned my code a bit, i added these variables:

```
private float minSpeed = 12;  
private float maxSpeed = 16;  
private float maxTorque = 10;  
private float xRange = 4;  
private float ySpawnPos = -6;
```

I then replaced the hard coded numbers accordingly, then i made a new method called Vector3 RandomForce() {} and inside it i added return Vector3.up * Random.Range(minSpeed, maxSpeed); i'll then replace the code in start() with RandomForce(). Then i added float RandomTorque(){} inside it was return Random.Range(-maxTorque, maxTorque); and i also added Vector3 RandomSpawnPos() { return new Vector3(Random.Range(-xRange, xRange), ySpawnPos);} i then replaced the codes in the void Start() with the methods i made in order to make the code short, clean and readable.

I learned that with RandomForce() the reason it doesn't have the "new" keyword, is because it's multiplying a Vector3 that already exists, and multiplying it by values, i can return that Vector3 immediately. In my RandomSpawnPos, because i am actually creating a new Vector3, i actually need to use the new keyword to create a new Vector3 variable with values that the code can then use.

Then i made a new empty object called "Game Manager", and attached a new "GameManager" script, then opened it and inside it i typed: public List<GameObject> targets; and then in Unity i added a 4 to the new Targets component so that i can add my 4 prefabs. I also learned that arrays and lists are similar, and example of these two is: public List<GameObject> targets;(Lists) and public GameObject[] targets2;(Array), with the list itself i can actually pass it in the type of thing that i want, unlike with the array i have to tell the array what that thing is before we make the array itself. Lists and arrays have slightly different functionality in terms of the ways that they work.

Later i added private float spawnRate = 1.0f; then i made a new method that is: IEnumerator SpawnTarget() and inside it i typed { while (true)

```
{ yield return new WaitForSeconds(spawnRate);
```

```
int index = Random.Range(0, targets.Count);
```

Instantiate(targets[index]); } } this will make random objects from different x range spawn from below the scene. Then in void Start() i added StartCoroutine(SpawnTarget()); to actually start the code of making the objects spawn.

After that, in the Target C# script, i made two new methods: private void OnMouseDown() { Destroy(gameObject); } and private void OnTriggerEnter(Collider other) { Destroy(gameObject); }. These will make the objects disappear after i click them with my mouse or after they collide with the Sensor collider, below the boards and panels. I also made the ySpawnPos = -2 insted of -6, I tested all this and it works, although my objects now go a bit too high sometimes, but i like this, the player will have to remember if to click on the objects that went too high, it's a cool feature.

-2D View: Since i am play a 2D it's a lot easier if i just look at my entire view through the 2D one, no need to complicate things. To do this you simply need to press the button that says "2D" on the top of the scene view.

-AddTorque: I applied Torque to the rigid bodies of my objects in my target script. This will make the objects i applied to spin in a direction i want or to all direction, randomly.

The force can be applied only to an active rigidbody. If a GameObject is inactive, AddTorque has no effect. Wakes up the Rigidbody by default. If the torque size is zero then the Rigidbody will not be woken up.

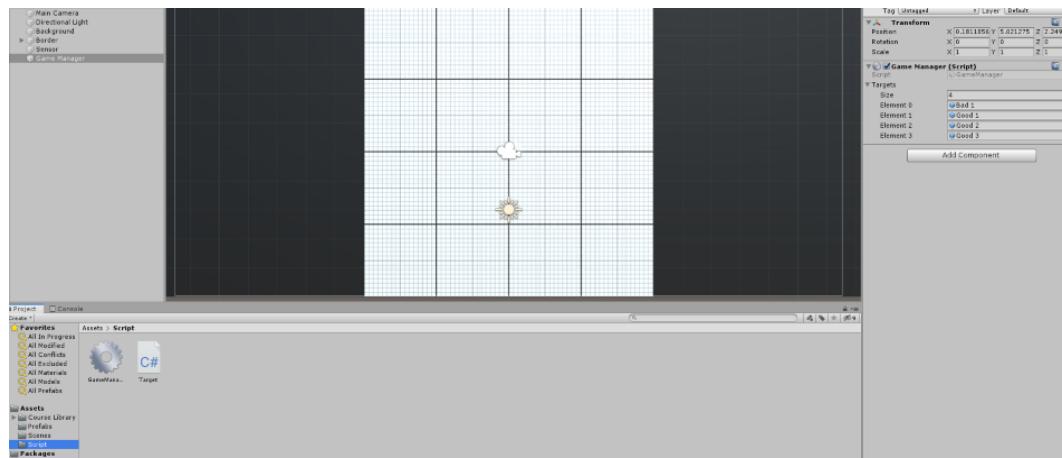
-Game Manager: I created a Game Manager object instead of a Spawn Manager, because i'll be using this actually control the state of my game as i add more to it.

-Lists: In the Game Manager script i learn how to use a list in place of array to be able to get different objects.

A list is a different kind of data structure from an array. The biggest difference is in the idea of direct access Vs sequential access. Arrays allow both; direct and sequential access, while lists allow only sequential access. And this is because the way that these data structures are stored in memory.

-While Loops: While loops are basically a fusion between a for loop and an if statement. So while loop will execute your code continuously, but instead of giving it a defined number of times to run, we can actually use a condition to tell it when it should actually stop running. I used this in my code to be able to control while something is happening to keep iterating over my blocks of code and keep doing them over and over again.

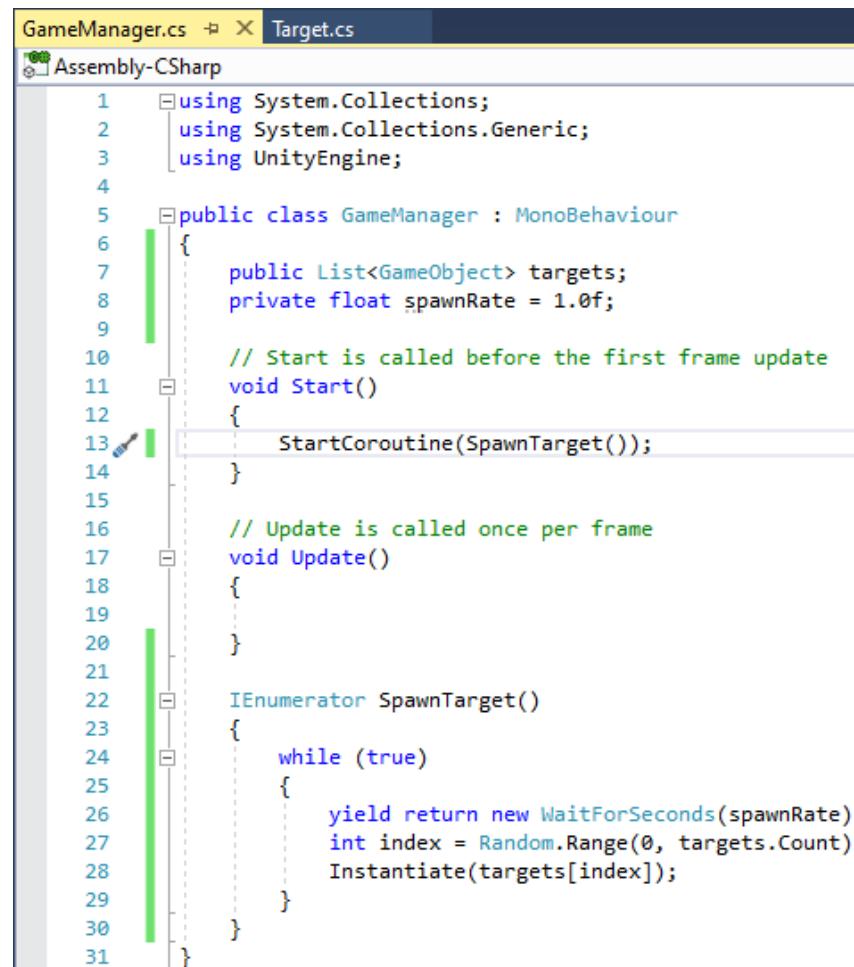
-Mouse Events: I learned how to use "OnMouseDown", so that whenever i get my mouse clicked i can actually immediately know when it's over one of my objects and i can destroy the game object as a result.



The GameManager script i added in my Unity game, and the empty game object called Game Manager that has the script within, and the component that has Lists, i made the targets 4 and added my prefabs into them.

```
GameManager.cs  X Target.cs
Assembly-CSharp
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class GameManager : MonoBehaviour
6  {
7      public List<GameObject> targets;
8
9      // Start is called before the first frame update
10     void Start()
11     {
12     }
13
14
15     // Update is called once per frame
16     void Update()
17     {
18     }
19
20 }
```

The List i added into my GameManager C# script.



A screenshot of a code editor showing the `GameManager.cs` script. The window title bar says "GameManager.cs" and "Target.cs". Below it is a tab labeled "Assembly-CSharp". The code is as follows:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class GameManager : MonoBehaviour
6  {
7      public List<GameObject> targets;
8      private float spawnRate = 1.0f;
9
10     // Start is called before the first frame update
11     void Start()
12     {
13         StartCoroutine(SpawnTarget());
14     }
15
16     // Update is called once per frame
17     void Update()
18     {
19     }
20
21     IEnumerator SpawnTarget()
22     {
23         while (true)
24         {
25             yield return new WaitForSeconds(spawnRate);
26             int index = Random.Range(0, targets.Count);
27             Instantiate(targets[index]);
28         }
29     }
30 }
31 }
```

The code i added in my GameManager in order to make random objects from,
bad 1 to good 3, to spawn.

After all that i right-clicked in the hierarchy and went to UI > TextMeshPro text, then clicked on the button to Import TMP Essentials, then i renamed the new object "Score Text" and changed the Anchor Point so that it is anchored from the top-left corner in the inspector window,

i then repositioned my Score Text to be in the top left corner.

Later in the inspector i changed the text from "New Text" to "Score:" then i changed it's font style to Margarine-Regular SDF, i left it's Font size and vertex color as it is.

I then went to my GameManager script and added a "using TMPro;" at the top of the script, underneath all the other using statements. Then i added a new public TextMeshProUGUI scoreText; and private int score; variables. Later in the void Start() i added score = 0; and scoreText.text = "Score: " + score;, all this will make my game have a score with numbers and all. I then, later dragged the Score Text to the new empty Score Text slot in the component of the Game Manager, i then tested my game and a 0 appeared next to the score.

After that i created a new private void UpdateScore() method, this and what's inside it is what will update the score of the game. Inside UpdateScore i added scoreText.text = "Score: " + score; and score += scoreToAdd;, then in Start() i added UpdateScore(0); , after that i typed UpdateScore(5); in the While(true) this will actually add 5 "point" every time objects spawn, this feature won't stay it's just to test if the UpdateScore works. I press play and it works.

Later in the GameManager script i made my void UpdateScore(int scoreToAdd) into a public method do that my Target script can call it. I also deleted the UpdateScore(5); in the while loop because it's no longer necessary, after that, in the Target script i added a new private GameManager gameManager; then in the void Start i added, gameManager = GameObject.Find("GameManager").GetComponent<GameManager>(); and in the private void OnMouseDown() i typed gameManager.UpdateScore(5); so that the Target script can call on the GameManager script and make it update the score by 5 after the player clicks the objects with the mouse. I went in Unity and tested this and it works, the score gains points by 5 each time when i click on the objects, even the bad objects give the score point, i'll fix that in future lessons, in this create with code 5.

After that i assign a point value to each target, i did this by going to my Target script and adding public int pointValue; then inside the gameManager.UpdateScore(5); i replaced the 5 with "pointValue". Then in each prefab, there is now a Point Value in the Target component, i added values to my prefabs, i added 15 to my Good 1, 10 to Good 2, 5 to Good 3 and -10 to Bad 1. I tested if my code works, in Unity and it works, different points are added to the score depending what i click.

I then added a public ParticleSystem explosionParticle, in the Target script, and i also added Instantiate(explosionParticle, transform.position, explosionParticle.transform.rotation); these will make a particle effect appear after i click on a object. I then went to Unity and assigned a explosion particle, from the course library folder in the particles folder, to my prefabs in the empty new slot they have now, i dragged the black particle to my Bad 1 prefab, the blue particle to my Good 1 prefab, red particle to my Good 2 and a yellow particle to my Good 3 prefab. I tested this in my game and it works, a explosion particle happens when i click on the objects in the scene.

-TextMeshPro: I used this to be able to create smart UI elements. TextMeshPro provides Improved Control over text formatting and layout with features like character, word, line and paragraph spacing, kerning, justified text, Links, over 30 Rich Text Tags available, support for

Multi Font & Sprites, Custom Styles and more.

-Canvas: I used this to actually position all of my text on my screen. The Canvas is the area that all UI elements should be inside. The Canvas is a Game Object with a Canvas component on it, and all UI elements must be children of such a Canvas.

-Anchor Points: I used this to be able to tell TextMeshPro whether i want something to be located always within the center of our screen or in one of the corners of our screen. That way whenever a player plays my game on a different device like a phone or a computer, the text will align itself accordingly. In this project i made the anchor point make the text always be on the top left corner.

-Import Libraries: I used the using -statement so that the GameManager can be called and imported to the Target script.

-Custom methods with parameters: The custom method with parameters i made were really used it for its public key word, this was public void UpdateScore(int scoreToAdd).

-Calling methods from other scripts: In my target script i called the GameManager script, specifically it's UpdateScore method, gave it a parameter to update on and then updated the score on my game.

After all that, in Unity, i Right-clicked on the Canvas, created a new UI > TextMeshPro - Text object, and renamed it "Game Over Text", then i move the text so it's more in the centre of the screen. In the inspector i made the text to "Game Over", the font style to be the same as the score text, i made the text color red, increased it's size and then set the "Wrapping" setting to "Disabled", so that the word "Over" isn't under the word "Game". I looked at and it looked good.

Later i Uncheck the Active checkbox to deactivate the "Game Over" text by default, then in my GameManager script, i added a new public TextMeshProUGUI gameOverText; and assigned the Game Over object to it in the inspector, of the Game Manager component. I then added, in the void Start gameOverText.gameObject.SetActive(true); this will activate the Game Over text, that is deactivated in Unity, and make the it appear in the middle of the screen.

After that i deleted the scoreText.text = "Score: " + score; in the void Start() because i realised that it's no longer necessary, since that code is in the public void UpdateScore(int scoreToAdd), it's a waste of space to have two codes in different part of the script.

I then, still in the GameManager C# script, created a new public void GameOver(){}, then i copied and cut the gameOverText.gameObject.SetActive(true); and pasted it in the new function i just made. Then in the Target script i added gameManager.GameOver(); to the private void OnTriggerEnter(Collider other), after that i went to my Unity and added a tag to my Bad 1, that tag is called "Bad". Later i went to my Target script again, and added a if (!gameObject.CompareTag("Bad")) {gameManager.GameOver();}, i took out the gameManager.GameOver and placed it inside the if-statement. I tested this and it works, a

Game Over text appears in the centre of the scene when a good object touched the collider below, and a Game Over text did not appear when a bad object touched the collider below.

Later i created a new public bool isGameActive; and in the while (true) i replaced the "true" with "isGameActive", then i added isGameActive = false; in the public void GameOver(). In the void Start i added isGameActive = true; but here the order of the code is actually important now, i also move score = 0; so now isGameActive = true; and score = 0; are above StartCoroutine(SpawnTarget()); and UpdateScore(0);. In the Target script i created: if (gameManager.isGameActive) {} and inside it i copied and cut what code was in the void OnMouseDown() previously. I went to my Unity game to test it and every thing works as intended.

After that i tried to right-click on the Canvas and create > UI > Button, but for me the UI wasn't there, i thought i miss clicked but not i really isn't there, i can't continue doing these Unity lesson because i can't create a Button, i even asked my tutor but he couldn't help or figure out what the problem was. But later when i was at my home computer i tried to create a button and it worked. For some reason, right-clicking on the Canvas and create > UI > Button, actually works, i still don't know why this didn't work on the other computer at college. After that i continued with the work i had to do in this project, after i created a Button i renamed it to "Restart Button", i temporarily reactivate the Game Over text in order to reposition the Restart Button nicely with the text, below the Game Over text, then deactivate it again. I then selected the Text child object, then edit its Text to say "Restart", and i changed it's Font to Margarine-Regular, i increased it's font size to 18 and made it's colour more dark.

Then i added using UnityEngine.SceneManagement; on the top of my GameManager script, i also added a new public void RestartGame() function that reloads the current scene you are playing, on the bottom of the script, inside this function is:

SceneManager.LoadScene(SceneManager.GetActiveScene().name); then in the Button's inspector, i clicked on the + to add a new On Click event, dragged it in the Game Manager object and selected the GameManager.RestartGame() function, after that i tested my game and it works, the restart button restarts the game, unfortunately it restart when ever i want but i will fix that later.

After that i unchecked the "Active" checkbox for the Restart Button in the inspector, then i added using UnityEngine.UI; in the GameManager script, and a public Button restartButton;. The new using statement will allow me to actually interact with the buttons. Finally i added restartButton.gameObject.SetActive(true); in the public void GameOver(), then i connected a reference of the button in the scene, to the button variable that i created. Tested this in unity and it works, the Restart button only appears after a Game Over, and the button works properly.

-Game states: I managed my game states and actually give the players feedback when they do so with my Restart button and "Game Over" text.

-Buttons: I crated a button by right-clicking Canvas and create > UI > Button, and made some modifications like, repositioned the button itself, made the text bigger, changed it's font style, made it darker and more.

-On Click events: I made the Restart button actually do something, and that something is restarting the game when i click on the button.

-Scene management Library: I learned how to use some new libraries, the new libraries i used were: "using TMPro;", "using UnityEngine.SceneManagement;" and "using UnityEngine.UI;". The scene management itself was used to load and reload my scenes.

-UI Library: I used this to can actually interact with different elements like my buttons in my scene.

-Booleans to control game states: I used Booleans to control my game states. An example would be, in my case, when my game is active it'll run a whole bunch of different things like making my objects, Good and Bad, spawn. But then i can also set it so that the game is over, like in my target script where, when the game is running, then the player can interact with different objects, but then when the game is over, then the player can't.

Later after that, in Unity, i duplicated my Game Over text and renamed it to "Title Text", i changed its colour to blue and its Text to "Food Clicker". I then duplicated my Restart Button and edited its attributes to create an "Easy Button" button, its text was "Easy" and i positioned the button to the left for space for the other buttons. Then i duplicate and edited the new Easy button to create a "Medium Button" and a "Hard Button" all with different text and positions.

I then, for all 3 new buttons, in the Button component, in the On Click () section, clicked the minus (-) button to remove the RestartGame functionality. After that i created a new "DifficultyButton" C# script and attach it to all my 3 buttons, then inside the script i added using UnityEngine.UI to my imports, i also created a new private Button button; and, in the void Start, button = GetComponent<Button>();.

Later i added void SetDifficulty(){} and inside it i added a Debug.Log so i can know if my code knows when i'm clicking on the difficulty button and which one, the code is: Debug.Log(gameObject.name + " was clicked"); and i also added, in the Start(), button.onClick.AddListener(SetDifficulty);. After that i, in unity, i tested this, clicked on the buttons and a Log on the console appeared, informing about the buttons i clicked.

I then went to my GameManager script and made a new public void StartGame() and inside this i cut and copied the code that was in void Start(), then in the DifficultyButton script i created private GameManager gameManager; variable, and i made sure the script could find the gameManager in the scene with gameManager = GameObject.Find("Game Manager").GetComponent<GameManager>(); in the void Start. Then, in void SetDifficulty() i added gameManager.StartGame(); so that the game actually starts when i click any difficulty button. I tested this and in the start, without me pressing anything, the prefabs don't start spawning only after i click on of the buttons.

After that i right-clicked on the Canvas and Create > Empty Object, rename it "Title Screen", and drag the 3 buttons and title text onto it. In the GameManager.cs i added a new public GameObject titleScreen; then in unity i dragged the Title Screen to the new empty "Title

Screen" slot in the inspector of the Game Manager. Later in the script i added titleScreen.gameObject.SetActive(false); to my public void StartGame() method, i tested my game and everything works perfectly as intended.

Later in the DifficultyButton script, i created a new public int difficulty variable, then in the Inspector, i assigned the Easy difficulty as 1, Medium as 2, and Hard as 3. After that i added "difficulty" to the gameManager.StartGame() in order for the difficulty to actually work. Then in the GamaManager script i added int difficulty in void StartGame(), inside the brackets, and inside the curly brackets i added spawnRate /= difficulty; so that the spawn rate becomes faster. I went to unity to test this, and in the video i learned that i could enter play move with the scene on full screen, i tested my game, in full screen, to see if the spawn rate becomes faster the high the difficulty, and it works fine, although it's becomes almost impossible to play more then 10 seconds on hard difficulty.

-AddListener(): I used button.onClick.AddListener(SetDifficulty), basically when the button is clicked, the game needed to listen, to pay attention, to when it's actually been clicked.

-Passing parameters between scripts: I used methods from other scripts that i wrote with parameters that we can pass into them, like the public void StartGame(int difficulty).

-Divide/Assign (/=) operator: I used this to basically divide numbers, an example i used was spawnRate /= difficulty; i used the value of the difficulty code to divide the spawnRate.

-Grouping child objects: I figured out how to disable all of my title screen elements, by nesting them underneath my title screen game object, and turning it on and off.

Quiz 5:

I failed question 1 because i got confused whether the sentence in red or in brackets needs to have a Title Case.

I failed question 3 because i simply didn't see the one (=) when it should be two like this (==).

Question 7 was failed because i didn't know that the "UnityEngine.UI" library was a essential for the class like "Button".

I failed question 8 because i didn't know that the first two required parameters are Vector3 variables.

I also failed question 10 because since lists need [] i thought that but i was wrong.line 1 was the answer

Challenge 5:

I first tried to fix the problem of The difficulty buttons look messy, i went and look at the texts in the hierarchy of the difficulty buttons, and i noticed that their alignment is wrong and different from each others. I fixed this by going to the Text objects of each buttons, went to Paragraph > Alignment and made all of them the same, in the centre.

Then i went to TargetX.cs and i immediately found the problem of the food not being destroyed when the player clicks on it, only when the mouse touches it. The problem was that the script had private void OnMouseEntre(), when it should have had private void OnMouseDown(), this will make the objects be destroyed after you click on them, i placed the correct code, then tested the game and it works.

I went to the GameManagerX to fix the score display on the scene, the problem is the code scoreText.text = "score"; i fixed this by replacing it with scoreText.text = "score: " + score; all i did was added the "+ score", then i tested this in unity and it worked, numbers appear when i click on the objects.

Still in the GameManagerX script i already fixed the Game Over button not appearing, all i did was change the false, in restartButton.gameObject.SetActive(false); , to true.

After that i tried to fix the problem of the difficulty buttons not changing the difficulty of the game. First i looked at the code then i looked at my code i did in Create with Code 5, and i realised that i have to add "difficulty" from the public int difficulty, in the DifficultyButtonX, then i went to the GameManagerX and on the public void StartGame() i added "int difficulty" in it, later in the curly brackets of the public void StartGame(int difficulty) i added spawnRate /= difficulty; which will actually divide the spawnRate. After that i went to test the game and it works, the game will get faster when you click on the medium button, and even faster when you click on the hard difficulty button. The problem for me is that the game is way to fast even on the medium difficulty, so i went to the GameManagerX and changed the private

float spawnRate speed from 1.5 to 2, i tested this and to me it was still a bit fast so i increased the spawnRate to 3, and after that the speed of the spaenRate seemed fine to me.

Lab 5:

On this lab 5 i basically made my game look nice, i first downloaded the Course Library from the Tutorial materials on the Lab 5 site, i then looked at objects and stuff that the Library has.

Later i actually replaced my objects with the new assets, but first i dragged my Player GameObjects to the Prefabs folder, in order to make it into a prefab. Then on the Player's prefab editor I dragged the asset I wanted my player to be, I choose the Farmer, the farmer was now a child of the Player, i reposition and scaled it to be the almost the same as the blue sphere, then i gave it a box collider, edited it to be the same as the farmer. Later i unchecked the Mesh Renderer and the sphere collider of the blue sphere which is the Player. I tested my game and at first the player didn't move so I adjusted it's position in the prefab editor because I thought it was because the farmer or player was too close to the ground, after testing, it was moving but slowly so I decided to uncheck the box collider of the farmer and check the sphere collider on the player. After that speed of the player was back to normal but I think it's a bit faster than usual but that's ok, it's fun to be fast.

After that i added assets to my enemies and basically they all have different animals asset, a moose, a doe, a horse and a stag each with their own box collider. I made the enemies(the red box) invisible so that in game you could only see the asset model. Then i changed the rotation of the animals, so that they are facing the Player, i tested my game and they seemed fine, they even faced in the right direction but they started moving, I don't know why, but I fixed this by freezing there position and rotations in the inspector of their Rigidbody.

I then went to the asset store and searched for low poly bullets (free ones to be specific), I couldn't find any bullets that I liked so I searched for rockets instead and I found one I liked a lot. I downloaded the assets, and imported some of it since I won't be using all of it's assets. After that I double-licked on my Bullet Prefab and added a red small rockets as the child of the bullet, I repositioned and resized it to be the same as the capsule bullet, then I gave the rocket a capsule collider and I unchecked the mesh renderer of the bullet so it becomes invisible. I then tested this and at first the bullets wouldn't pass the fences and the reason for this is because I didn't enable the "It trigger", I

did that and now it goes through the fences, then I decided I didn't like how the rockets looked so I tried to make the rockets spawn in a different rotation, more horizontal, again.

I first tried to change the rotation of the object in the prefab editor but that didn't work, so I tried to change the rotation of the Spawn Manager where the Bullets come from and after that it worked, the rockets started spawning horizontally. I did that to all my Spawn Managers, with slight differences in the rotation some have, -90 or 90, and others have their rotation on the Z instead of the X, but in the end they all looked good for me, they all looked like the rockets were aiming for the Player.

After doing some more tests, I decided to change the speed of the Player to be higher because I didn't like how it felt, I increased the speed for 10 to 13.

I then went and added the Prob_Barrier03 to my Fences, of which I made them all prefabs, in the Prefabs editor. I changed the position and scale of the Prob to be the same as the fence, then I unchecked the Mesh Renderer, and on some of the Prob_Barrier03 I made the Y axis rotation to be 180 in order for all the fences to be facing player. Then I added a Brown material to all my fences so they become brown not just white, then I tested to see how the fences look in-game and they look very nice, perfect for my game.

Later I went and added a texture to my plane, I chose the Gravel texture, I then made the material less shiny, in the Material properties, I unchecked the "Specular highlights" and "Reflections" settings, I left the Tiling of the Gravel as it is because it looked nice in my game. I tested my game and everything worked and it looks fine.

-Art workflow: I made my game look nice and appealing to the eye. I basically downloaded some assets from the asset store and from the link the lesson on Lab 5 gave me and implemented them into my game, it's all about the visuals here, nothing to do with the mechanics.

-High vs. Low Poly: High Poly means a high polygonal count on your 3D model, while Low Poly is the opposite. Each has its value and time. High Poly gives you more details to play with but renders slowly, therefore it is used in Films where we have enough time to render. While low poly is needed in games for example to establish a high enough frame rate for real time rendering of the 3D models. I used low poly for my game in order for it to run well.

-Asset Store: I downloaded something in the asset store, a low poly rockets cartoon pack, then I imported them into my game, although I didn't import everything because I wasn't going to use all of the pack.

-Nested Prefabs: You can include Prefab instances inside other Prefabs. This is called nesting Prefabs. Nested Prefabs retain their links to their own Prefab Assets, while also forming part of another Prefab Asset. To do this all you have to do is, double-click on a prefab and in the prefab editor you simply dragged an object you desire into the hierarchy, and then that object becomes the child of that prefab.

-Material properties: I added the Gravel texture to my Plane, but I had to make the material less shiny, so in the Material properties, I unchecked the "Specular highlights" and "Reflections" settings, there were many other things I could do in the settings but I

just did that.

Create with Code 6:

I opened my my create with code 1 project, the car game, then i opened the PlayerController.cs script so i can change the code to be more clean.

On PlayerController.cs i added [SerializeField] to my private float speed = 30.0f; and private float turnSpeed = 50.0f; then i saved and went to Unity to see if the variables are now visible, and they are. I tried to change the values in Unity while in Play mode, and it works, the values do change and do have an impact on the vehicle an example is if i increase the speed value the vehicle will move faster, if i lower the value the car will move slower, if i change the Turn speed it will also change the speed of how fast the car will turn accordingly.

I then added the [SerializedField] to my FollowPlayer C# script, specifically in front the offset code, after i did that i went to see if i could see the values of the offset in the inspector window of the Main Camera GameObject. Of which i can see, i can see 3 values one with the X axis, Z axis and Y axis, then i tried to see if i can edit the values while in play mode and see if they do anything, and they in fact do change how the camera will see, if i change the X axis the camera will move to the left or right accordingly, the same happens with the different axis but the Y will move the camera up and down and the Z will move it front and backwards.

After that i started applying the "readonly", "const", or "static" attributes, to my code and i noticed that when you added them to your code the value in the inspector windows disappear, which is interesting. I also deleted the private attributes on my code so that just the [SerializedField] is there and not any static readonly or public keywords, just to see what it do, if there will be a different in the game or will everything stay the same. After i did a few tests i realized that nothing happened and that everything stayed the same. To be honest i don't even know if i should keep the private next to the [SerializedField] or not what are the benefits of it, if any.

Later i duplicated my Main Camera and renamed it to "Secondary Camera", i unchecked the Main Camera's box on the top of the inspector so it's deactivated. Then i moved the Secondary Camera to be in front of the vehicle and i added it's Transform position values to the Offset in the Follow Player component. After that i play tested my game and i noticed that the car shakes a lot, like a earthquake, i also noticed that if you move forward that the camera will fall back and be inside the boxes the car is holding, the camera will move even more forward you drive backwards, still shaking a lot.

Then i went to my PlayerController.cs and changed the void Update() to void FixedUpdate(), i saved, then went to Unity to test it and now with the new void FixedUpdate the camera stops shaking and the car runs smoothly now with is great.

Later i changed the void Update() to LateUpdate() in the FollowPlayer C# script, then i tested y game to see how my camera works, and now it works just fine, it no longer shakes violently and everything is very smooth when in motion.

I then deleted the void Start() in both FollowPlayer and PlayerController scripts, so it's more clean and organized that way, then i disabled the Secondary Camera and re-enabled the Mani Camera. I tested my game one more time to see how it felt and looked in a 3rd person view and it looked good and felt very smooth.

Later i went to my prototype 2 game and made a back up of it by going to, Assets > Export Package, i made sure everything was selected, clicked export, then i saved it on a new folder i made called "Prototype 2 Completed", all of this was done so i can use object pooling.

I then imported the Object Pooling Package into my project, which is from the direct link download on the Tutorial Lesson 6.1 - Project Optimization site i was using to do all of this project and optimization.

Later i, after i imported the Object pooling, i had to re-add some scripts to my prefabs and Player, i had to drag the PlayerController script to my GameObject Player, i had to drag the DestroyOutOfBounds and the DetectCollision scripts to my Prefabs. After i did that added the new ObjectPooler C# script to my Spawn Manager, so that my Food steak can be activated and deactivated accordingly, and increased the Amount to Pool from 0 to 10 and dragged my Food_Steak_01 to the Object to Pool slot. I didn't do anything to Pooled Objects and it's size since it's not important right now. I then went to test my game and after i shoot my meat i couldn't shoot it any more, i looked at the errors in the console and saw the instructions on the tutorial again, and i realized that the the food_steak was not suppose to have a DetectCollision, so i took it out on the prefab of the food_steak. After i did that and tested my game everything worked fine, my game has no problems whatsoever.

-Optimization: I have optimized my game and the code within the scripts of the prototype 1, i did this by removing the void Start of both scripts, changing the void Update to FixedUpdate, in the PlayerController.cs, and changing void Update to Late Update on the FollowPlayer.cs. Then i cleaned my code a bit more by adding [SerializeField] instead of private in the code, then i went to prototype 2, made a backup, and added object pooling to the game so that it's more optimized and doesn't push on your computers process power.

-Serialized Fields: Serialization is the process of taking an object in ram (classes, fields, etc...) and making a disk representation of it which can be recreated at any point in the future. When you apply the SerializeField attribute to a field, it tells the unity engine to save/restore its state to/from disk. You is mostly use serialization for the editor, and especially when building your own editor windows and inspectors. It is also used for showing a private variable's value on Inspector. You can easily change the value like public variable. But None can access this value from another script or places.

When i added [Serialized Fields] into my PlayerController.cs, i could see the values of the speed and the turn speed on the vehicle inspector window. So serialized field is a really helpful way of defining which variables you want to use just in the editor, but you don't want to ever use in a different class.

-readonly / const / static / protected: The keyword "protected" is a mix of private and public. So with private, I can use this variable in this script, in this class, but I can also use this variable in any other script that actually inherits from player controller. It also replaces the public or private keywords.

The PlayerController inherits from mono behavior. And so the Start and Update voids are now both protected methods, thanks to this protected keyword, this allows me to use them in my player controller script. However, if i made a brand new script, that didn't inherit from mono behavior, i wouldn't be able to use these methods. Basically protected, allows you to see variables, or methods outside of the current class. But only to any script, or class that inherits from this main class.

Another keyword i can use is "const". And const is just short for constant, and so that means that i can never change a value because of this keyword. Basically the value is constant, so for example, I could not write turnSpeed = 5f; , in the void Start or Update because turnSpeed is a constant variable, that can never change. const is added into a code with a variable, it does not replace the public, private or even the protected keywords. example: [SerializeField] private const float turnSpeed = 45.0f;

"readonly" is also a keyword i can use in my code, and when you first instantiate an object and use the readonly keyword you can set that variable, when you create that object, but then can never change it ever again, versus const (which is used after you create a variable), is defined as soon as you create that variable. This keyword also doesn't replace the private, public or protected keywords.

The readonly keyword is used to declare a member variable a constant, but allows the value to be calculated at runtime. This differs from a constant declared with the const modifier, which must have its value set at compile time. The real advantage of this keyword is to generate immutable data structures.

Finally the keyword "static" can also be used if you want to have something like a global variable to use anywhere in your code at all times. This keyword also doesn't replace any other keywords

static, in C#, is a keyword that can be used to declare a member of a type so that it is specific to that type. The static modifier can be used with a class, field, method, property, operator, event or constructor. It is mostly used when the data and behavior of a class do not depend on object identity.

-Event Functions: Events are specialized delegates that are useful for when you want to alert other classes that something has happened, events function very similarly to public multi cast delegates.

The Update function is the main place for this kind of code in Unity. Update is called before the frame is rendered and also before animations are calculated.

A separate event function called `FixedUpdate` is called just before each physics update. Since the physics updates and frame updates do not occur with the same frequency, you will get more accurate results from physics code if you place it in the `FixedUpdate` function rather than `Update`.

An example of where the script code should override the effect of an animation (say, to make the character's head look towards a target object in the scene). The `LateUpdate` function can be used for these kinds of situations.

- `FixedUpdate()` vs. `Update()` vs. `LateUpdate()`:

- **Update()** is called on every Frame, regardless of time passed since last frame. Good for Movement, InputControl etc. (most of the time you'll use `Update`). Usually you will use `Time.deltaTime` to take passed time into account (e.g. for `GameObject` Translations)
- **LateUpdate()** is called after all `Update()` methods are processed. So e.g. for the camera that follows your character it's good to `Update` after it has moved
- **FixedUpdate()** is called by the physics engine in fixed intervals (that can be set in the Options). This is basically good for all Physics related functions (trigger, collisions etc.)

I did some research on this matter and these helped me a lot (<https://docs.unity3d.com/Manual/ExecutionOrder.html>) and (<https://answers.unity.com/questions/153008/what-is-the-difference-between-update-lateupdate-f.html>), it's always good to a good variety of research.

-`Awake()` vs. `Start()`: `Awake` is called after all objects are initialized so you can safely speak to other objects or query them using eg: `GameObject.FindWithTag`. The difference between `Awake` and `Start` is that `Start` is only called if the script instance is enabled. This allows you to delay any initialization code, until it is really needed. `Awake` is always called before any `Start` functions.

-Object Pooling: Basically Object pooling is where you pre-instantiate all the objects you'll need at any specific moment before gameplay. For instance, during a loading screen. Instead of creating new objects and destroying old ones during gameplay, your game reuses objects from a "pool", or when a Player shoots bullets a specific object is activated and when it reaches out of bounds it's deactivate.

I used Object pooling on my prototype 2 game making my food prefab be activated and deactivated accordingly.

I opened my Prototype 1 game again, i went to Assets and exported the package in order to make a backup form my game because i'm about to change it's code, so in case anything goes wrong i can go back and start over. I then went to my PlayerController script and "private Rigidbody playerRb;" and typed, in the void Start which i had to create because i deleted it previously, "playerRb = GetComponent< Rigidbody >();", after that i replaced the name of the "speed" variable to "horsePower", and increased the number from 20.0f to 20000.0f. Later in void FixedUpdate i replaced the old code to move the vehicle forward with:

"playerRb.AddRelativeForce(Vector3.forward * horsePower * forwardInput);" at first i tried to use verticalInput but for some reason Visual Studio didn't let me use it, it was always with a red wobbly underline, so i though that maybe forwardInput will work and it does.

Later i realized that my forwardInput was named like that instead of verticalInput like the guy from the tutorials, but that's alright the name is just different and i won't change it since it's not necessary.

After that i went and tested my game to see how it works, and it moves slowly at first but then ramps up speed but just when it gets fast, the car flips, which is not good. Then after doing some tests in play mode with the global and local rotation, my vehicle seems to move just fine, whether its local or global rotation, which is weird since i practically didn't do anything differently to my code or game, but it still works fine so it's alright.

I learned that i can easily test and object's direction and position, in play mode by changing the the tool handle Rotation and Position buttons, above the Scene, Game and Assets Store tabs, also next to the tools button on the top left hand corner. It's a useful thing to know when you want to test your game and it's objects, and thus adjust my code because of the tests you do.

After all that i later, went and added wheel colliders to the wheels of my vehicle and edit their radius and center position, then disable any other colliders on the wheels, i made sure everything was fine and dandy. I then create a new empty child object for the vehicle called "Center Of Mass", then i made some new additions to my PlayerController script, which are: adding "[SerializeField] GameObject centerOfMass;" so i can place the Center Of Mass empty object into the component that the vehicle has, and "playerRb.centerOfMass = centerOfMass.transform.position;" supposed to add a new center of mass to my vehicle. Later i dragged the Center of Mass empty object to the new slot that the vehicle has in the inspector window, called "Center Of Mass", then i adjusted the empty Center of Mass object position.

Then i did some tests but i was having some problems with the vehicle always flipping on it's back, i tried to solve this issue for hours, i tried to move the Center Of Mass object below, forward, back, above and anywhere i can think of, but nothing worked, not even the distance mattered. Until i saw the positioned of the car in the tutorial had the z axis on it's position to be 0 while mine was -3.36, so maybe i though it has something to do with the positioned of my car itself. I moved the car so now it's positioned is X-0, Y-0.02, Z-0, and when i tested my car it no longer flipped backwards, which was a great relief, i was happy that i finally fixed the problem.

But my vehicle now couldn't move so i had to adjust the horse power and the position of the Center Of Mass, to be slightly higher. After some adjustments everything seemed to work out fine, which is good, my car's horse power is now 15000, with a mass of 1000, starts at the position of X-0, Y-0.04, Z-0 and turn speed of 45, now i can move on to the next step.

Now i will add a speedometer display in my Prototype 1 game. To do this i have to add a TextMeshPro object to my scene, much like what i did with projects before, so in the hierarchy i right-click in the empty space go to UI > Text - TextMeshPro and import it, inside the Canvas is the Test (TMP) child object which i renamed it to "speedometer" and moved it on the scene to be in the top left corner. I also changed it's anchor, in the inspector window, to be also top left and the text in the inspector will be "Speed: ".

Later in the PlayerController.cs, i added, at top of my script, "using TMPro;" so now I can find the TextMeshPro library, i then created "[SerializeField] TextMeshProUGUI speedometerText;", then in Unity i dragged the speedometer to the Speedometer Text in the Vehicle's Player Controller component. After that i added in the script a new float variables for your speed which is "[SerializeField] float speed;", then in the void FixedUpdate() i added "speed = Mathf.RoundToInt(playerRb.velocity.magnitude * 3.6f); and "speedometerText.SetText("Speed: " + speed + "km/h");", i used 3.6 because i prefer km rather then miles. I then tested my game and the text was a very small, so i re-looked at the tutorial video i was watching, and realized that my aspect ratio was set to 1920x1080, instead of the Free Aspect that the guy from the tutorial has, so i change my aspect ratio and now when i play my game the text is now big and visible which is great.

I then added a RPM display to my game, what i did first was make a new TextMeshPro by right-clicking on canvas and going to UI > Text - TextMeshPro, rename it to "RPM" and do pretty much what i also did with the "speedometer" TextMeshPro object, but on the text i typed "RPM: 0" and made the speedometer and RPM objects wrapping to disabled, so that the numbers are to the left of it instead of below it.

I repositioned the "RPM: 0" on he scene to be below the "Speed: 0", then i went to my PlayrController script and added "[SerializeField] TextMeshProUGUI rpmText;", of which i dragged the RPM text object to the new empty slot called Rpm Text. I also added in my code, [SerializeField] float rpm, rpm = Mathf.Round((speed % 30) * 40) and rpmText.SetText("RPM: " + rpm). I tested my game and it works perfectly, although i don't like that the RPM on the scene keeps going up and then goes back to zero and starts over, but the man in the

tutorial said that it's suppose to do that. I did more tests manly to try two different codes, `rpm = Mathf.Round((speed % 30) * 40)` and `rpm = (speed % 30) * 40;` after i tried both methods i realized no noticeable difference happens so i decided to just use the `rpm = Mathf.Round((speed % 30) * 40)` code because i prefer this one.

After that i went to my PlayerController C# script again to prevent my Vehicle from driving in mid-air, to do this i added [SerializeField] List<WheelCollider> allWheels and [SerializeField] int wheelsOnGround. After that i increased the size of the All Wheels to 4 so i can add my wheels from the Vehicle to that slot, then i dragged all four Wheels child objects from the Vehicle, to the slots that i made in the Player Controller component.

Later i made a new method called "bool IsOnGround(){}" and inside the curly brackets is the code i added so that my game can know if all 4 wheels are on the ground or not, by counting them. If one wheel or more are not on the ground i (the player) will not be able to control the car, you won't be able to accelerate or stop the car until all 4 wheels are on the ground. To do this i had to add a if statement, inside the void FixedUpdate, called, if (IsOnGround()) and inside this if statement is the code to accelerate and rotate the care, and the speed and rpm display, not the input code. Thus this code in the if statement can connect with the bool IsOnGround method code. I did a test and everything works, but one this i still have a problems is that my Vehicle can still flip to the sides if you turn left then right, or vice versa, really fast and strong, making the car flip. I think i leave it just has it is because i don't won't to break my game, it works fine just as it is.

-Searching on Unity Answers, Forum, Scripting API: I learned through videos i watched on this lesson 6.2 and tutorials on how to search stuff on the internet so i can solve problems and find solutions to stuff want to do in my game, On unity specific, i can search on the Scripting API which is basically the official information on specific codes, variables, methods and such. I can on unity forums and Answers where i can get information on people having a problem and asking for help, i can learn something from this, which is useful.

-Troubleshooting to resolve bugs: Basically to fix a bug you have to,

Step 1 – Replicate the bug yourself, or at least identify the problem and use that, that you have found;

Step 2 – Make sure you truly understand the problem and maybe establish a theory of probable cause;

Step 3 – Fix the bug or test the theory you have created;

Step 4 – Prove the fix;

Step 5 – Don't test your fix;

Step 6 – Closing the loop.

-AddRelativeForce, Center of Mass, RoundToInt: AddRelativeForce adds a force to the rigidbody relative to its coordinate system, force can be applied only to an active rigidbody. If a GameObject is inactive, AddRelativeForce has no effect and it wakes up the Rigidbody by default. If the force size is zero then the Rigidbody will not be woken up. It can be either 2D or 3D, in my case on my game i used 3D to make my car accelerate fast, always going faster and faster.

In terms of Center of Mass, if you don't set the center of mass from a script it will be calculated automatically from all colliders attached to the rigidbody. After a custom center of mass set, it will no longer be recomputed automatically on modifications such as adding or

removing colliders, translating them, scaling etc. To revert back to the automatically computed center of mass, use Rigidbody.ResetCenterOfMass, i used this in my game so my Vehicle can be more balanced and not flip when you are driving it.

RoundToInt or better Mathf.RoundToInt, returns f rounded to the nearest integer, it basically makes non whole numbers whole, if a number is, 0.9356 with Mathf.RoundToInt it will round up to 1, making it a whole number. If a number ends in .5 so it is halfway between two integers, one of which is even and the other odd, the even number is returned.

-Modulus/Remainder (%) operator: The percent symbol is a special kind of operator in C#, and in programming in general, called the modulus operator, or also known as the remainder operator. The (%) actually means that you are dividing something in your code, i used this to, so that my speed is divided by 30 so then it will actually give me the remainder value.

-Looping through lists: I made a list that my game can loop through to check if my car has all 4 wheels on the ground, i added this, [SerializeField] List<WheelCollider> allWheels, to my code in order for a list to appear in Unity.

-Custom methods with bool return: I made a new bool, which is called "bool IsOnGround()" so that the code that accelerates the Vehicle, displays the speed and rpm, turn on and off because of the custom bool method that returns that i created.

After all that i went to my Unity Hub and went to Installs, clicked on the 3 dots on my 2019.2.3f1 and added modules. I selected WebGL Build Support, and both Mac or Windows build support, then i clicked Done and waited for the installation to complete.

Later, after the installment is completed i opened my prototype 4 game, i went to Files > Builds Settings and a new build settings window appeared. I clicked on, Add Open Scenes and the most recent scene appeared, which is good, then i clicked on Player settings and i looked at the options i could change but i just did a little bit of change like changing the games name to Sumo Balls. On the Resolution and Presentation like making the Resolution - Fullscreen Mode: Windowed and making it a resizeable window, i looked at other options i had but that pretty much all i did.

I tried to make the Display Resolution Dialog, enable but a yellow warning sign appeared saying " The Display Resolution Dialog has been deprecated and will be removed in a future version", so i decided to make it disabled instead.

I then went and build my game and placed it in my Builds folder i created, after some time has passed, i tested and played my game inside the Builds folder, and one thing i noticed was that a small window displaying configurations, the resolution and graphics options and input controls, i think my game doesn't start with that because i did enabled the Display Resolution Dialog but i'm not sure. I played my game

and the screen itself was shorter the usual, it went back to normal when i clicked on the fullscreen button on the top right. After the test i did, everything was functioning well and everything was fine, which is great.

Later i had to make a game for WebGL and instead of the same game i used for mac and PC build, i wanted to use the prototype 3 game for a WebGL build. First i went to my game, went to files and click on Build Settings, went to Player setting and only changed the name to "Runner" and then clicked on the Switch platform, after that i clicked on Build and placed the build on a Build WebGL folder i created for this.

After that i went to my Build WebGL folder and clicked on the new index with a chrome icon, but my google chrome could't play the game so i opened it with Microsoft edge and it works with that, i could play my game and it works fine, although you don't see all of the Player because i think the screen is a bit small but that's alright. After that i compressed zip my Build WebGL folder so i could send it to someone if i wanted to.

After i did all the lessons i wanted to watch the Tutorial - Entity Component System, so i could get a better understanding on what it is and how that will help me in my future programming life. I also used this site to help me understand ECS:

([https://learn.unity.com/tutorial/entity-component-system?
courseId=5cf96c41edbc2a2ca6e8810f&projectId=5d092adcedbc2a0e5c02d26f#5c7f8528edbc2a002053b676](https://learn.unity.com/tutorial/entity-component-system?courseId=5cf96c41edbc2a2ca6e8810f&projectId=5d092adcedbc2a0e5c02d26f#5c7f8528edbc2a002053b676)).

ECS also known as "Entity Component System" has a lot of advantages in using it, like: is extremely performant code, easy to read code, easy to reuse code, can use Burst compiler and integrates well with a C# job system. ECS essentially is a new way to write code in Unity, ECS uses entities, components and systems, it splits your game into 3 parts. Entities are used to group together components, they're much like traditional gameobjects but much lighter, making your game run faster. Components are just containers for data unlike traditional MonoBehaviours, these components don't have any logic. Systems define components based behavior, this means that the system is responsible for operating on all entities with a specific set of components, these are the only objects that actually contain any logic. All this is much better than the traditional way of coding which is one by using GameObjects and MonoBehaviours, this is a not so optimized and performance based way of coding and ECS is better, in fact I have read, in the Unity Wiki, that the Entity Component System (ECS) is a new official framework for writing highly optimized, data oriented code in Unity. Unity claims that using these new packages together can lead to as much as x100 speed improvements over the existing "classic" GameObject and MonoBehaviour architecture.

There are two types of ECS, one is pure ECS, which is completely separate from the old way of coding, with pure ECS we stop using GameObjects and MonoBehaviours all together. The other one is hybrid ECS, these types make it easier for people to do the transition from the old system to ECS, this type won't unlock the full performance benefits of pure ECS but it's a great way to start converting your scripts to ECS and learning this new way of thinking.

Why not use the "classic" or "traditional" way of coding? You should use ECS instead because the problems of the "classic" way of coding is that it's largely dependent on reference types, more problems it has are that the it uses are scattered, loading memory to cache is very slow, a bunch of extra data is provided that is unnecessary and processing is done one at a time. The Job System Overview helps your game to perform better, what it can do is separate data from function, it can multi-core processing and it can save multi-threading. Not only that but there are many more benefits to using the Job system like solving multithreading problems, job system manages multithreading for you and focus entirely on your game specific code, and probably much more.

The Job system and the Entity Component System are not the same thing, they are separate, although they do work very well together. After seeing the example from Tutorial - Entity Component System, I learned that the job system itself can improve performance in your game by around 50% - 100%, which is great, but the ECS can improve your performance even more, and this can be even greater if you add Burst Compiler with ECS, tremendously increasing optimization and performance. The only small downside for using job system or ECS is that it may be a little hard to use at first and that you have to use different code to use it. But it's still great you can use it for any game you want, or on anything, the sky is the limit.

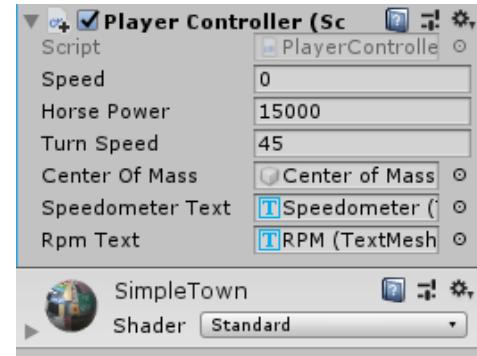
The Unity C# Job System lets you write simple and safe multithreaded code that interacts with the Unity Engine for enhanced game performance. You can use the C# Job System with the Unity Entity Component System (ECS), which is an architecture that makes it simple to

create efficient machine code for all platforms.

Burst is a compiler that translates from IL/.NET bytecode to highly optimized native code using LLVM. It is released as a Unity package and integrated into Unity using the Unity Package Manager.

Another site i read so i can understand job system and Burst Compiler,

(<https://docs.unity3d.com/Packages/com.unity.burst@0.2/manual/index.html>)



On the Vehicle GameObject and on the Player Controller component, the slot which i filled because of the code i added in my script. Especially the Speedometer and Rpm Text.

```
FollowPlayer.cs PlayerController.cs X
Assembly-CSharp PlayerController
9 [SerializeField] float speed;
10 [SerializeField] float horsePower = 15000.0f;
11 [SerializeField] float turnSpeed = 45.0f;
12 [SerializeField] float rpm; —
13 private float horizontalInput;
14 private float forwardInput;
15 private Rigidbody playerRb;
16 [SerializeField] GameObject centerOfMass;
17 [SerializeField] TextMeshProUGUI speedometerText;
18 [SerializeField] TextMeshProUGUI rpmText; —
19
20 void Start()
21 {
22     playerRb = GetComponent<Rigidbody>();
23     playerRb.centerOfMass = centerOfMass.transform.position;
24 }
25
26 // Update is called once per frame
27 void FixedUpdate()
28 {
29     // This is where we get player input
30     horizontalInput = Input.GetAxis("Horizontal");
31     forwardInput = Input.GetAxis("Vertical");
32
33     //Move the vehicle forward
34     playerRb.AddRelativeForce(Vector3.forward * horsePower * forwardInput);
35     // We turn the vehicle
36     transform.Rotate(Vector3.up, Time.deltaTime * turnSpeed * horizontalInput);
37
38     speed = Mathf.RoundToInt(playerRb.velocity.magnitude * 3.6f);
39     speedometerText.SetText("Speed: " + speed + "km/h");
40
41 rpm = Mathf.Round((speed % 30) * 40); —
42 rpmText.SetText("RPM: " + rpm); —
```

What I added to my script (showed in the red lines) in order for the RPM to be displayed on the scene
of my game.





How my display works when i'm in play mode.

Unit 7 Creator Kit: Beginner Code:

I went to the "Tutorial - Creator Kit: Beginner Code", which is Unit 7, by clicking the link that was provided by the CRC VLE and start doing it's lessons so i can improve my coding skills and understand C# better. I first went to "Tutorial - Get started with Creator Kit: Beginner Code" so i can get a better understanding in what i'll be doing. I read the points in order and the lesson told me to update my Unity hub to be 2019.2 or above, which thankfully i already have it updated to 2019.2.3. The tutorial then told me to download the Project for Beginner Code: which i did in the unity hub of which i took some time but i did downloaded it, i also read about a way to save your game if you cannot save your Unity Project, which is good to know and i also did use this to save my game when i tested the game.

Then the tutorial, from 5. to 7. just reviews some basic things that i have already gone thru, like key concepts, the tool bar and the interface. After i read that part of the tutorial i tried out the game, i looked at the game in unity and it's quite big it has a lot of content in it. I pressed play button on the toolbar and tried the game, with the home computer i have now, the FPS(frames per second) isn't very high, so the game will not run very smoothly. But it is playable, i moved my Player and i learned that the Character moves a bit to fast for my liking, it's good in one way as you can traverse the area fast but it makes the Player hard to control if that makes sense, the graphics are nice and so is the art style, very simple and nice, the controls itself are very simple, you just click the left-mouse button on an are you want to go or an enemy you want to attack, you also have a few shortcut which is appreciated, like pressing "I" for the inventory instead of clicking the

backpack image. I then wanted to save my game, but for some reason i couldn't, so i tried to do what the tutorial said if you encountered this problem, and it works i saved my game in a file i created called "Creator Kit: Beginner Code", not only that i also Collabed this game, just in case and to also be able to play it in my college.

Later, I then went to the Tutorial - Get started with code in Unity, here i learned more about what is programming, how computers understand instructions and how to use that to make a game. Then it told me to open the SpawnerSample script file, so i went to Unity and looked for the PotionSpawner GameObject, in the hierarchy. I found it, it was in the LevelDesign Empty GamObject, which was easy to find as i thought, i went to the PotionSpawner inspector window and clicked on it's script field of the component (which is greyed out), i clicked on the name of the script (SpawnerSample). This highlighted the script file in the Project window, i then double-clicked the SpawnerSample script and looked at it's code, which is how the health potions spawn, their position and rotation. With this lesson i learned how to find a script i wanted by clicking in the greyed out name in the script component, which is good to know.

I then learned in the next tutorial that Variables are basically, how you store data that can change also data stored in a variable can be a integer or a string and variables are useful as a way to saving time, writing cleaner code and if you name them clearly, making your program easier to understand.

Another thing i had to do was consider: the purpose of this script in the game and the word 'angle', the lesson asked me, "What could this instruction be telling the computer to do?" and i personally think that the purpose of this script in the game is to spawn the health potions GameObject at the start of the game so that Payer can get some help in the beginning of the game. The word "angle" serves as the name of the variable it's using, this helps identify what the variable is used for.

Then later, the tutorial asks me "In each line, what is different from the first time the variable was assigned a value?", about the two next lines where the int variable angle is assigned a different value. The difference is the "int" no longer exists and the value of the variable is different for each one. The reason the "int" no longer exists is because the computer already knows that the variable exists and how much memory space it needs and since instructions in a script are sequential, the changed value for the variable will only be used in instructions after it has been assigned.

After that i went to the next tutorial, the Tutorial - Write your own instruction lesson, i had to find the instruction which tells the computer where the potions should be created, which is repeated 3 times in the code. I found the code and it's the "spawnPosition = transform.position + direction * 2;" this code makes the potion far from each other, without it the health potions would spawn inside of each others. I then had to add my own new variable, called "radius", it's value needs to be 5, have a semicolon (;) and should be an int type, so that it can store whole number values. i typed "int radius = 5;" on top of the angle variable and replaced the hard coded 2 number in the "spawnPosition = transform.position + direction * 2;" codes to be "spawnPosition = transform.position + direction * radius;" this way he code is better and more clean. i saved and tested it and the separation of the potions are bigger.

Next lesson i made the SpawnSample C# script spawn +1 more health potion so there will be 4 potions instead of 3. I did this by basically copying 4 lines of code that make a potion spawn and changing the value of the angle to 135. The code was this:

```
angle = 135;  
direction = Quaternion.Euler(0, angle, 0) * Vector3.right;  
spawnPosition = transform.position + direction * radius;  
Instantiate(ObjectToSpawn, spawnPosition, Quaternion.identity);
```

Later i learned about Functions and what they do. What they do is they help make you avoid writing lots of repetitive code, functions are sets of instructions, so when the compiler reaches a function, it will tell your computer to execute all the instructions in that set before continuing with the rest of the script.

I then learned about what's inside functions in scripts, function declaration and function body. Function declaration is similar to a variable declaration, it includes: the type of function, the name and parameters. Function body contains the instructions that are bundled under the function name. The computer will execute these instructions when it encounters the function name in another instruction, it's simple it starts with an opening curly brace, then the instructions and lastly a closing curly brace, and also each single instruction within the group ends with a semicolon.

I then begin to read about "function call", which is made up of the function name, a set of parentheses containing two values and a semicolon, to mark the end of the instruction. Function calls tell the computer to execute all the instructions contained in the function before continuing to work through the script sequentially. I also learned about how to Adjust the behaviour of my function when it is called, which is called passing a value, and identify the function that already exists in your script, which is "void Start()"

After that the lesson told me to write my own spawn function, so i did it, i made a separate function below the void Start() called "void SpawnPotions(int angle)" and it's body was:

```
"Vector3 direction = Quaternion.Euler(0, angle, 0) * Vector3.right;  
spawnPosition = transform.position + direction * radius;  
Instantiate(ObjectToSpawn, spawnPosition, Quaternion.identity);"
```

Then then was a error that was expected from the lessons, and so it told me to fix it. I also learned why the errors appeared, it was because of scope which are the curly braces in my scripts, so to fix the problems in the scripts i had to Re-declare the variables in my function, so i added "int radius = 5;" to my void SpawnPotions. Later the tutorial told me to replace the spawning instructions with function calls so i tried to do that and i added four of them along with the instructions, i forgot to delete them, and tested my game, what appeared was 6 or 8 health potions near the Player. So i went back to the script and deleted the instructions and try a few things out to fix my game but i still had error, then i re-looked at the the tutorial and realized that i forgot to add a "Vector 3" to one of my lines of code, specifically the re-declaring variables. After placing just the "int radius = 5;" and the function calls in the void SpawnPotions(int angle), and just the "Vector3 spawnPosition = transform.position;" and the four SpawnPotions(); with 4 different variables in the void Start(), i tested my game and everything finally works, there are no more error or problems.

After all that i went to the next tutorial the: Tutorial - Introduction to classes, here i learned what are classes and how to use them. Classes basically organize your code and act as blueprints, you can also create subclasses which inherits common features with their own specific characteristics. I learned what are dot operators and what you need to do to make one.

Next i have to create a class declaration, first i typed " public class LootAngle{}" below the closing curly brace of the SpawnerSample class, then i added two integers which are " int angle;" and "int step;" after that i added a function called int NextAngle(), with some code inside it. I learned some new things like the constructor, which is a thing that classes can do, a constructor is an instance method that usually has the same name as the class, and can be used to set the values of the members of an object, either to default or to user-defined values.

Later i had to add a code that is "LootAngle myLootAngle = new LootAngle();" and the tutorial said i had to place it on top of the Start function (above the first call to SpawnPotion), but when i create parameters for the constructor in my public class LootAngle, an error appeared on LootAngle() on top of the void Start. After a little thinking and reading the tutorial i realized that the "LootAngle myLootAngle = new LootAngle%;" was meant to be placed at my LootAngle class. Then i added a 45 value to the new LootAngle(), i then replaced all of the SpawnPotions() to SpawnPotions(myLootAngle.NextAngle()); i accidently forgot that my SpawnPotions had an s in the end so when i looked at the console in Unity i had more error then what the tutorial had so after that fix, i had the correct error and the tutorial said that we would fix that in the next lesson.

I then continued to the next lesson where i learned about the difference between public and private keywords. Classes are private by default but the "public" keyword lets you access a class, function or variable from another class and exposes it to the Unity editor, allowing you to adjust your gameplay without changing your script.

Then i added the "public" keyword to the left of my "LootAngle(int increment)" and "int NextAngle()", with this the errors in the console will disappear in the Unity editor.

Later i had to make a 3D sphere and rename it to "MoneySpawner" and place it near the PotionSpawner, then add the SpawnerSample script to it, then i went to the prefabs tutorial in unity and drag the MoneyLoot prefab to the SpawnerSample component that i just added to the MoneySpawner object, i save my Unity editor then i tested the game and it works, when i start the game 4 moneys will spawn in the game that i can collect, the moneys are separated from each other.

I then learned about derivation and inheritance, inheritance is the ability to create a class that inherits attributes and behaviours from an existing class. The newly created class is the derived (or child) class and the existing class is the base (or parent) class. I also learned about what does the MonoBehaviour class contain? And it contains the variable transform, which contains the Transform component and the

function GetComponent, which enables you to access a component on a GameObject, which is good to have since it does most of the work for you and you don't need to write extra stuff in your script.

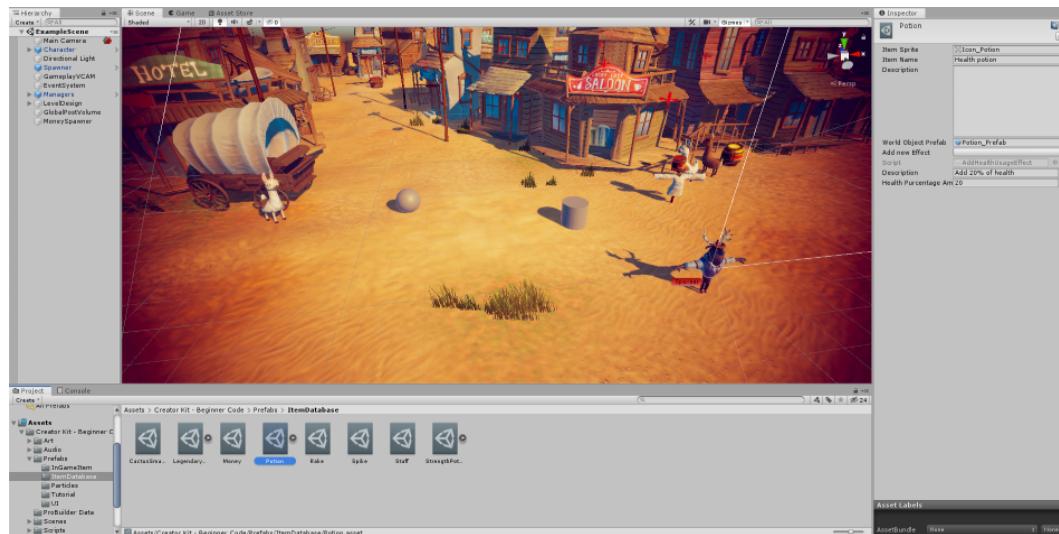
After that i went to the next lesson and was asked to customise the health potions. First i went to the top menu, went to Beginner Code > Create Item Effect, i named it to "AddHealthEffect" and went to where the new script that i created was and doubled-clicked it.

I then learned what each thing in the script does, then i went back to the Unity editor, went to Assets/Creator Kit - Beginner Code/Prefabs /ItemDatabase and selected the Potion Prefab. Later in the Inspector, i used the Add New Effect drop-down menu to select the AddHealthEffect, then in the Description field, add a description of the effect which is "Gives 10 HP", then i saved Unity.

Then do some changes to the new script, basically just added "public int HealthAmount;" and replace the 10 value in "user.Stats.ChangeHealth(10);" with HealthAmount. After that a new Health Amount field appeared in the inspector window of the Potion Prefab, i typed the value of 10, because the lesson wanted me to, saved the game, tested the game, got damaged to 5 HP and used the potion and it worked, it healed my health to 15 HP, this will make the gameplay more easier since you can heal more now then before. With that i completed the main tutorial of Creator Kit: Beginner Code.


```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using CreatorKitCode;
5
6  public class AddHealthEffect : UsableItem.UsableEffect
7  {
8      public override bool Use(CharacterData user)
9      {
10         return false;
11     }
12 }
```

What was inside my AddHealthEffect C# script that i created on Beginner Code
> Create Item Effect.



The item database that i used and the PotionPrefab, of which i added the AddHealthEffect script that i made.

The screenshot shows a code editor window with the tab bar at the top labeled "AddHealthEffect.cs" and "X". Below the tab bar is a toolbar with icons for file operations. The main area displays a C# script named "AddHealthEffect.cs" under the assembly "Assembly-CSharp". The code is as follows:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using CreatorKitCode;
5
6  public class AddHealthEffect : UsableItem.UsableEffect
7  {
8      public int HealthAmount;
9
10     public override bool Use(CharacterData user)
11     {
12         user.Stats.ChangeHealth(HealthAmount);
13         return true;
14     }
15 }
```

The additions and changes i did to the new AddhealthEffect script so that the potions give the player 10 HP.