unit 1.1 game 1 dev log

part 1/day 1

- ☑ welcome
    - ☑ is a video saying that the course exists
- ☑ watch, then do
    - ☑ is a video saying watch t hen do what the guy does for the tutuorals lol
- ☑ apparently the content will no longer be available as of July 31, 2023 soooooo that's my deadline :D
- ☑ install unity
    - ☑ I'm using unity 2021.3.24f the lts version
    - ☑ with visual studio code editor 2023 edition
    - ☑ already logged in
- ☑ if u complete courses u gain xp????
    - ☑ +10 Unity Essentials xp &
    - ☑ +60 programming xp?????????
    - ☑ u need 500 xp to level up past lvl 1
    - ☑ kinda cringe imo
- ☑ UNIT 1 3d truck driving game where u avoid or crash into boxes
    - ☑ I need this coz camera control and also 3d models not working in gnz

Educator Plan
By the end of this lesson, you will be able to:

- Create a new project through Unity Hub
- Navigate 3D space and the Unity Editor comfortably
- Add and manipulate objects in the scene to position them where you want
- Position a camera in an ideal spot for your game
- Control the layout of Unity Editor to suit your needs

- ☑ download Prototype 1 - Starter Files.zip
    - ☑ does this link even work in the future???
    - ☑ https://connect-prd-cdn.unity.com/20210923/c709e76b-3e93-4140-8675-f694b9f04399/Prototype%201%20-%20Starter%20Files.zip
- ☑ omg they talk so slow, 3x speed it is, tldr make new unity app/folder
- ☑ extract assets from starter files, then import the unity file in the zip

☑ the Project window, in *Assets* > *Scenes* > double-click on the Prototype 1 scene to open

☑ delete sample scene?

☑ roblox camera control to look around XD

  □ whoa everything is so small , I was right my models were too big I'm gnz

□

□

day 2 unity jr programercourse

Man the guy talks slowly,
unit 1.1
important info

your home view if you will. it centers your camera on the object that you have selected.
god I love how smooooooooooooooothe these Phoenix switches are on my keyboard and wish I could just put them on my k3 but no ;-;

the f key enables focus mode, this is like when u double click the middle mouse button in fusion to get to a decent reference frame for your object/file. it is primarily used to lock onto an object in space to move the camera and make it work

- alt left click drag allows you to rotate your camera around the object that you have selected
- alt right-click drag allows you to zoom but idk if I like using it but hay information is information

I can't believe it's an 8-minute video explaining this to the user

unit 1.1-4
in the transform component of an object you can select the menu button and click reset to reset the cords to all default

apparently, in unity, it's commonly accepted knowledge that each unit is one metre.

meaning yes my objects for my gnz project day 2/3 was indeed too big

they also tell us that we can rename the object in the component's window or in the hierarchy window

this happened on accident but you can maximize a screen by doing shift+space
they also want us to name the obstacle of our choice obstacle

a 3.5 min video saying

ctrl+p starts/stops play mode

and play mode does not save your game state
if you click the arrows on the cube on the top right part of the scene thoe arrows will align your camera with the axis that you select.
also the circular tool by the move or pan tool is for rotation

the bottom one is for all types of transformation, scale rotation move

uhhh the project folder area if you don't need to see the assets but just need to see file names/folder system the menu button for the project window has a 1 column layout for efficiency if you don't have the screen real estate for your desired view
also on the top right you can click layout and change the layout by clicking the presets, personally, I don't want to use those for what I'm doing but it might be useful for certain game layouts, troubleshooting or games you are making

you can also save custom layouts

omg, finally I can do 1.2 :D
c# scripting?

just so its easier on other devs and you so it doesn't create problems down the line
DONT RENAME YOUR SCRIPTS
because if u do it might cause problems coz of how its written
just name it what its intended to do
then make more
or merge and fix the bugs

in the tutorial, u can just create a blank script in the project folder area and then drag it to the object but I can't? maybe it's a new unity thing
BUT
you can just create one as a new component
c# = c sharp or .cs
in the line of code which apparently doesn't need a ;
public class PlayerController : MonoBehaviour
so the : kind of means x inherits from y
so, in this case, the 3rd word "PlayerController" inherits from the class "MonoBehaviour"
so ig ill write in shorthand
idrk what the tutorial guy means but the void is a method but sure ill go with it
void Start() is called before the game starts (sort of) or before the first "frame"

void Update() is called every "frame"

ofc // means everything is a comment on that line from that // mark
It's common practice to capitalize certain things and not capitalize others, ex.
code with capital letters means its class
but lowercase letters are components
transform.Translate(0 , 0, 1);

so you write code in this way

```
component.Method(arguments);
```

literally make it go vrooooomm and move forwards
it uses x y z and its 1 unit per frame

okok so now

4.Use a Vector3 to move forward

```
        transform.Translate(Vector3.forward);
```

is literally the same thing
but how do we make it so that its more than just along the z axis???
idk its just shorthand for that
we use the Vector3 library ig

we can modify that code to be like 20m/s with this

```
        transform.Translate(Vector3.forward * Time.deltaTime * 20);
```

coz delta means change in y
so delta time means change in time
so this means using the vector3 library we will move forward by 1 unit * 1 unit of time (second) * 20
or 1 unit per second * 20
or 20 units per second
or 20u/s
and since each unit is 1m
its 20m/s

OKOKOK
so now **RigidBody**'s
they are component that means the object its assigned to are a physics object, we can give it gravity and other things with this
sphere colider and mesh colider make it a coliable object and has different ways of interacting with the player/ a given object
mass is done in kilos :D

New Page

```
public float speed = 5.0f;
```

the word public can be changed to private, this denotes the accessability of the variable/class/other

```
private float speed = 5.0f;
```

also this is done to initialize the default value of the variable in the script's component tab
they are also known as access modifiers :D

when we do 5 as a variable, its only valid as an intiger, a whole number, so if we want to do fractional speeds we need to make it a floating point, a decimal number, so we could do 5.0, but its not understandable because its a decimal number

so since its not an intiger we need to use float to do decimals coz its a floating point number
so then can do 5.0f to turn 5.0 into a readable understandable number for the game
:D
what happens when we do dword :D it doesn't tell us ;-;

anyways next line of code
this is similar to the public float speed = thing BUT
it doesn't give us the number, or object its just a name

normally this doesn't work unless you assign a value in some programming languages but with c# if the script is assigned to an object ex. the camera you can drag an object into the player variable to reference what the player variable is, so ex we want the tank to be the player, (later the player model)

we can just put this code in the first level of the script in the public class and it will allow us to assign an object to the variable. whoa python is hard lmfao

```
    public GameObject Player;
```

ok so next line

in the void update() area

```
        transform.position = Player.transform.position;
```

so basically this transforming the position of the camera to be exactly the same as the Players, including its transformed(updated) position

so since we want to offset that
we can add
`+ new Vector3(0, 5, -7)`
we need to use a new vector3 variable (xyz)
but since we need to be able to change that

I messed with the code more and got the ideal maybe ggz angels

image.png

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

public class FollowPlayer : MonoBehaviour
{

    public GameObject Player;
    private Vector3 offset;
    public float camera_x_offset = 0.0f;
    public float camera_y_offset = 5.0f;
    public float camera_z_offset = -7.0f;

    // Camera rotation offsets
    public float camera_x_rotation_offset = 0.0f;
    public float camera_y_rotation_offset = 0.0f;
    public float camera_z_rotation_offset = 0.0f;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        offset = new Vector3(camera_x_offset, camera_y_offset, camera_z_offset);
        transform.position = Player.transform.position + offset;

        // Apply the rotation offsets
        Quaternion rotationOffset = Quaternion.Euler(camera_x_rotation_offset,
camera_y_rotation_offset, camera_z_rotation_offset);
        transform.rotation = Player.transform.rotation * rotationOffset;
    }
}
```

this still follows the player when they move away and jump but hayyy its what I want later on when I deny up/down / in / out movement

okok so important thing
to midiage or remove studder we could make a smoother with another script
OR fix the problem at the core sortof
do void LateUpdate() instead of Update()
this will make the camera's movement update be after the players position updates :D
BRO
someone should've told me sooner you can make the playmode color change wayyy more noticable

```
    From the top menu, go to Edit > Preferences (Windows) or Unity > Preferences (Mac)
    In the left menu, choose Colors, then edit the “Playmode tint” color to have a slight
color
    Play your project to test it, then close your preferences
```

programming/unity essentials xp +5 each

temp

```
using System.Diagnostics;
using System.Collections;
using UnityEngine;

[DebuggerDisplay("{" + nameof(GetDebuggerDisplay) + "(),nq}")]
public class PlayerController : MonoBehaviour
{
    public float speed = 5.0f;
    // Start is called before the first frame update
    void Start()
    {

}

    // Update is called once per frame
    void Update()
    {
        //move the vheicule forward
        transform.Translate(Vector3.forward * Time.deltaTime * speed);
    }

    private string GetDebuggerDisplay()
    {
        return ToString();
    }
}
```