

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 smooooooooooooooooothe 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 those 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 vroooooomm 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 colliable object and has different ways of interacting with the player/ a given object

mass is done in kilos :D