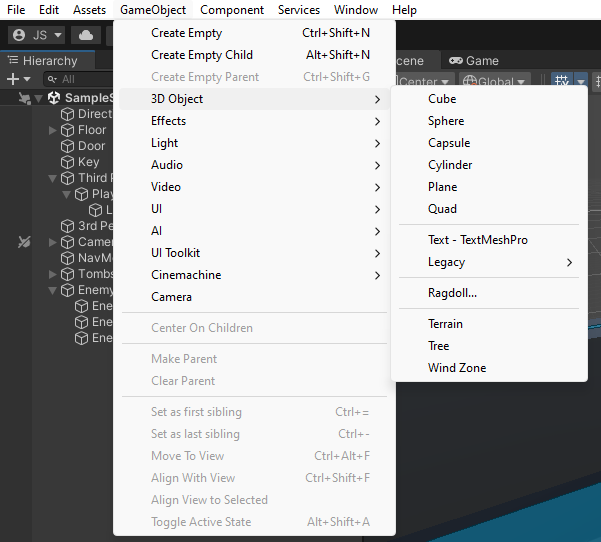
**Game Programming – Tutorials**

**Item Pickup.**

**1.**To start with you will have to go to “GameObject” in the task bar, this will bring out different things you can add to a game. Then you will have to go to 3D Object (as we will be making a 3D Game for this pickup) click on “Cylinder” as this will represent our key in game.

**2.**Then you want to change the name of the object you have created to what it will be in the game “Key”.

**A screenshot of a computer

Description automatically generated3.** Then were going to add a material so we can tell the difference in game. For this you will right click in the assests file in projects then go to create – material.

A screenshot of a computer

Description automatically generatedWith this an inspector tab will pop up and you can set the colour to the item, in this case we will be making a key pickup so the colour will be yellow. Drag and drop the material onto the game object.

**A screenshot of a computer

Description automatically generated4.** Create a script for the new Key game object for this we will right click in the asset folder and click create – C# Script.

This will bring an empty script up.

**A black screen with white text

Description automatically generated5.** create these 3 variables as you will be referring to these later. Maxpickupdistance is about how far away you must be to pick it up, the bool hasKey will determine if the player has the key or not.

A screen shot of a computer program

Description automatically generated

When you’re in the proximity of the key by pressing E on it will destroy the object and turn a valuable in your inventory saying it has key.

A screen shot of a computer program

Description automatically generated

With the raycast variable it will send out invisible laser to find an object that had a tag of “Pickable”. And destroy the object and then in the players inventory it will tick hasKey. Which will be like a pickup but hidden.

**A screenshot of a computer generated image

Description automatically generated6.** with the cylinder add box collider instead of cylinder collider as it is bigger for the player to see.

**A screenshot of a computer

Description automatically generated7.** Add the script to the player and make the max pickup distance the right size for your player.

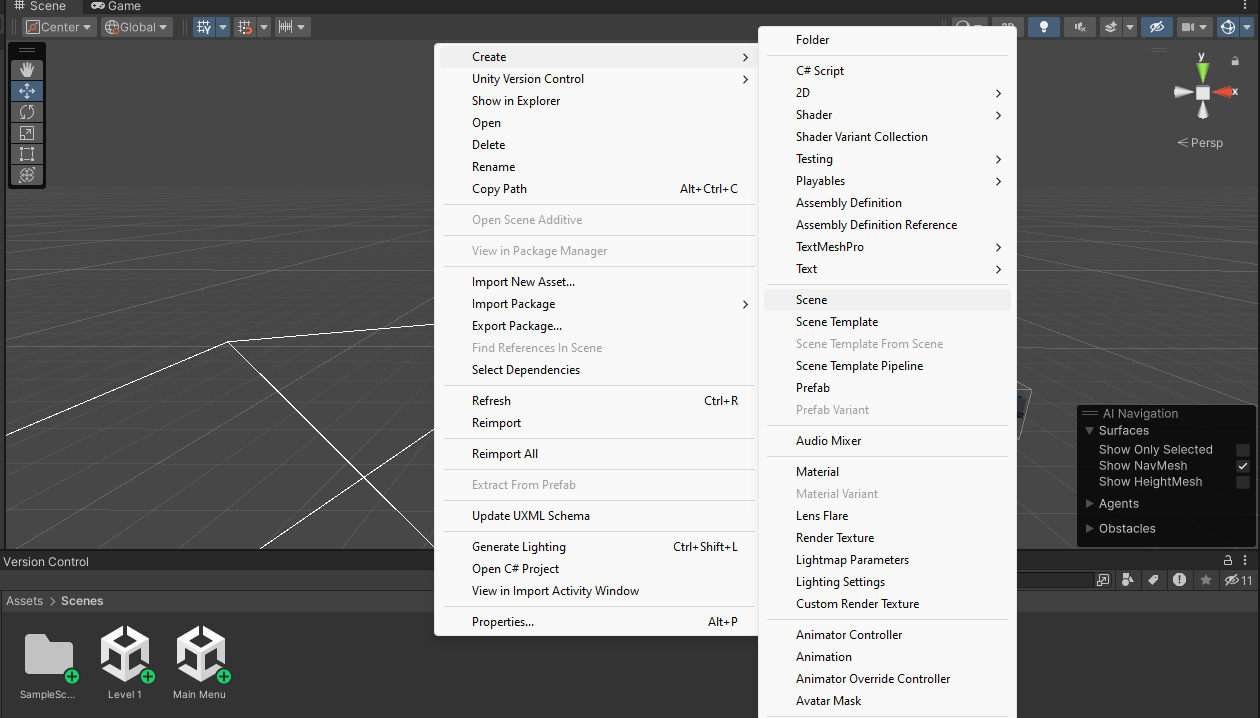
With this code you will be able to destroy the object you pick up by ticking the box.

To upgrade this code:

-you can add your own images of the pickup you will have.

-you could make an inventory of all the items your character holds

**Main menu (UI)**

1. To start with go into Asset > Scenes. This is where all your level and scenes for the game will be. For this we will make a new scene for the main menu to run off.

For this right mouse click then Create> Scene. Rename that scene “Main menu”.

1. A screenshot of a phone

   Description automatically generatedGo into that new scene that you’ve just made. And press 2D. this will make it easier.
2. Add the scene to the Build Settings. This can be found in File > build Settings…
3. A screenshot of a computer

   Description automatically generatedAdd all your scenes and levels you have so far by dragging them into the scene in build area.

Making sure that the main menu is at the top (0).

Then close that tab. (Do not build as it will build your game finished.)

1. A screenshot of a computer

   Description automatically generatedTo create a button for the main menu, you will have to right mouse click > UI > Button. Then rename it “Play”.

Go into the play button, there will be a text (TMP) child to the button this will be where we change the text and size of the button, for this we will name it “PLAY”, change the font size to **bold.** And move to the centre of the screen.A screenshot of a computer

Description automatically generated

You can move the button to anyone of these locations on the screen.

1. Make a new script for the scene and call it “Main menu”.

A screen shot of a computer program

Description automatically generated

Create a new public void as PlayGame, from this we want level 1 to load and play from this void. By creating a scene manager, we do this so the scene manager searches our scenes in build and can load up our next level.

For this I added LoadSceneAsync(1);

A black rectangular object with a black border

Description automatically generatedMeaning that I want it to find number 1 in the build scene and change to that when activated. (You can also add the name of the scene, but I find the scene build index to be easier.)

1. Add the script to the main camera just so it’s not lost.
2. A screenshot of a computer

   Description automatically generatedGo into the play button to “button tab” there you will find “On click” for this press the + to add to the empty list.

When added there will be nothing in it to make it work so for this you will drag main camera that holds the script in it.

A screenshot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generatedTo add a function, you will have to go onto No Function > MainMenu > PlayGame.

1. Do step 4 again but rename it “Options” and another one celled “Quit”.
2. A screenshot of a computer

   Description automatically generatedTo create a background, right mouse button > UI > Panel. Rename it “Background”. In the hierarchy create and empty game object and call it “Main menu” move Play, Quit and Options into it, making that a parent. Add that to the canvas now with the new background panel add that to the canvas and put it on top of the main menu (Don’t add it to the list with the play, quit, and options.)
3. With the background change the colour and make the alpha value of the colour stronger(A).
4. Now create an empty object and call it “Options Panel” inside this we will create a panel this will be for the background of our options. Change the colour of the panel background to something lighter.

A screenshot of a computer program

Description automatically generated

1. To make the options button work you must go into the options inspector and on “Button” drag the Options Panel in and change the “no function” to Game object > SetActive (bool). And tick the box.

A screenshot of a computer

Description automatically generatedThat will get you into the options part of the menu, so you will have to create a way out. For this you will drag the main menu. This will have your play, options and quit in it. And the No function > Game object > SetActive (bool). And leave this unticked.

1. We will make a cancel button it the options menu so we can leave without having to restart the game. For this we will do step 4 again but call it “Cancel”

A screen shot of a game

Description automatically generated

With all the buttons and background on it will look something like this.

You want to hide the cancel button to start with.

A screenshot of a computer

Description automatically generatedA screenshot of a menu

Description automatically generated

A black background with white ovals with words

Description automatically generatedFor this go into the Hierarchy and into the option panel in the inspector you will untick that box and this will make the cancel button disappear from the scene.

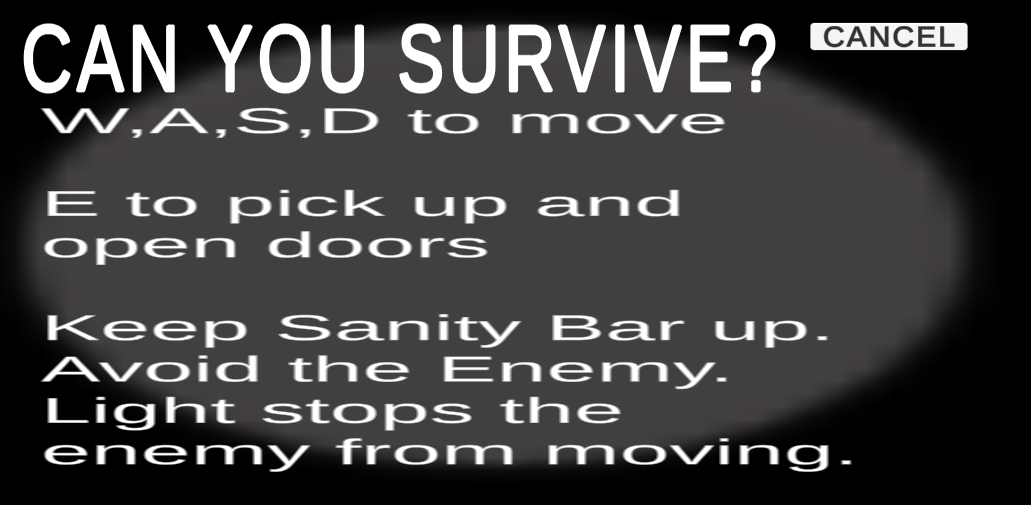
1. Now when you play the option button will lead you to where the options screen will be by pressing cancel it takes you back to the main menu.

By pressing play will take you to level 1 and the start of the game.

1. A screen shot of a computer

   Description automatically generated To make the Quit button work we need the script back to add the new quit option to it. For this we will add a new public void and call it QuitGame. We will have the application in it to quit.
2. A screenshot of a computer

   Description automatically generatedIn quit in the hierarchy go to button add in the script again from the main camera. No function > MainMenu > QuitGame.
3. For this to work you need to build the game and then you quit.

To update this:

* Change options of the game to how to play. By just creating a new text and a cancel button that will lead back to the main menu.

**Timer**

1. Create a canvas by right mouse click and go to canvas. Rename it timer, and in that create a Text (TMP).
2. Create a new script and call it Timer, within this script under using UnityEngine; on line 3 add using TMPro; this will be for the text for the timer clock that will be counting down.
3. Deleting void start up and void update, on a new line we will add.

A screen shot of a computer code

Description automatically generated

And by creating a new void update we will add.

A computer screen with white text

Description automatically generated

A number on a black background

Description automatically generatedWith this code the timer will count and have more decimal points then we need.

1. So, for this to count down and have no decimal points we will need this code.

A screen shot of a computer code

Description automatically generatedbasically, what it means if the seconds can only reach up to 60 and then change to minutes. But having the set up as 00:00 01:00 meaning that it will just count going up or down. so having it a timeLeft -= Time.deltaTime meaning that whatever the time will be set to it will go down to 00:00 and stop.

But what I added to it to make it more effective in the game was when it reached 00:00 it would change to red.

A computer screen shot of a program code

Description automatically generated

A computer screen shot of a program

Description automatically generated

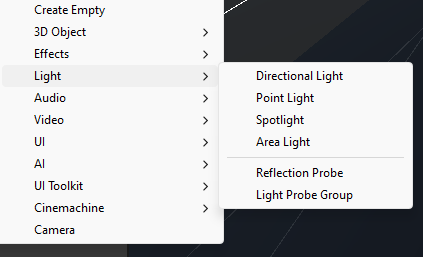
So, the whole code should read and look like so. By looking at the whole code here there is a small error that will make the time count down faster then we wanted. To fix this problem we just must get rid of one of the lines that say:

timeLeft -= Time.delaTime;

having this twice in the code will make it not match real time when counting down.

To update this code, you can have the timer start when you player interacts with an object add more urgency to the player.

**Flashlight**

1. Make a new game object (Cube) and shape it to what you want your flashlight to look like, then right click select Light and go to spotlight.
2. A computer screen shot of a computer

   Description automatically generatedMove the spotlight to the end of the flashlight.
3. Create a new script and call it Flashlight. And add it to the spotlight.

In the script we will start off by adding

Light m\_Light;

And in void start we will add

m\_Light = GetComponent<Light>():

And in update we will add

If (Input.getKeyDown(KeyCode.F))

{

M\_Light.enabled = !m\_Light.eneabled;

}

This is the function for us to turn the flashlight on and off by pressing F key. Now that that function works, we will add the brightness to the flashlight to act like it is losing battery power over time.

1. For the brightness we will add under Light m\_Light at the top

public float maxBrightness;

public float minBrightness;

public bool drainOverTime;

public float drainRate;

with these we will be able to change how we want the batteries to drain.

In void update we will be adding

M\_Light.intensity = Mathf.Clamp(m\_Light.intensity, minBrightness, maxBrightness);

If (drainOverTime == true && m\_Light.enabled == true)

{

If (m\_Light.intensity > minBrightness)

{

M\_Light.intensity -= Time.deltaTime \* (drainRate / 1000);

}

}

1. Now to change the batteries.

We will create a new void in the script.

Public void ReplaceBattery(float amount)

{

m-Light.intensity += amount;

}

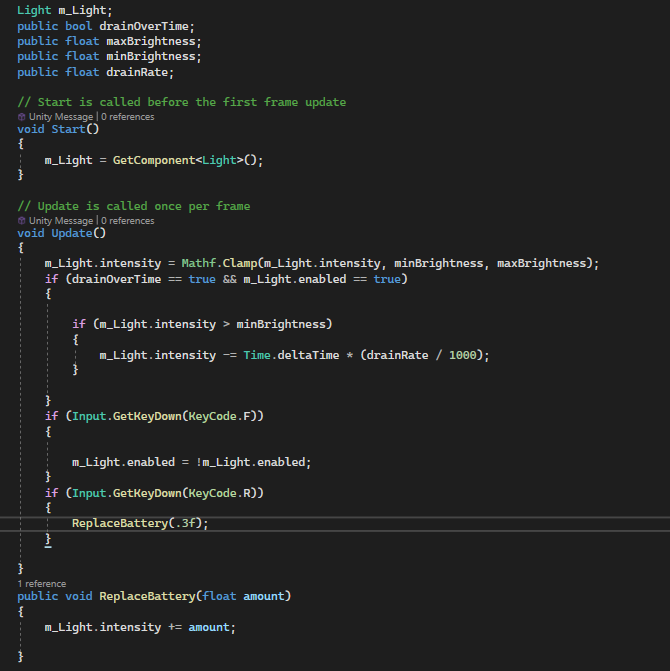
And in the void update we will add.

If (Input.GetKeyDown(KeyCode.R))

{

ReplaceBattery(.3f);

}

The code should read like this. So, you’re saying when you press F the flashlight will turn on and off and when you press R it will change the batteries, over time the brightness will drain if not changed.

1. In the inspector on spotlight, we will change the max brightness to 1 and the min brightness to 0.1 then drain rate to 5 and have drain over time ticked. We will change the render mode to important and mode to mixed, and shadow type to soft shadows. This is about when the light hits an object it won’t have harsh edges and will be able to show up on the object.

To upgrade this, you can:

-Change the colour of the light.

-Instead of having all the batteries you will need making the player collect them so that it can only be changed when you have them in hand.

Also, you can show what percentage the flashlight is at and make in countdown to 0 and making the light turn off. For this I added some more code

You will have to have a using TMPro;

So, you can add text to the canvas where it will show your percentage.

Also add:

Public TMP\_Text Text;

Public TMP\_Text batteryText;

Public float Lifetime = 100;

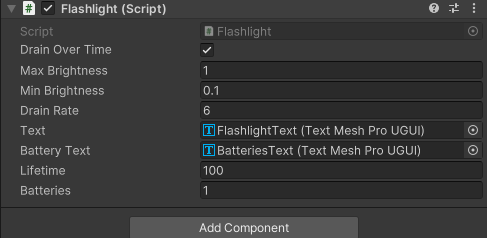
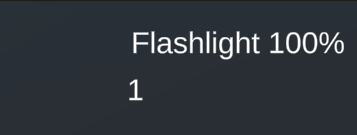
Public float batteries = 1.

This is saying that your flashlight will have a lifetime of 100 and you will start with one battery.

Then in void Update we will add:

Text.text = “Flashlight “ + Lifetime.ToString(“0”) + “%”;

batteryText.text = batteries.ToString();

now create a UI > Text. Drag under the flashlight and call this a “FlashlightText” and do the same again and call this one “BatteriesText” drag the flashlight text into the inspectors script under Text and the batteries one into Battery Text.

Now it will show on the game.