



2D GAME DEVELOPMENT (LABSHEET)



2D GAME DEVELOPMENT (LABSHEET)

2D GAME DEVELOPMENT DFG40323

Published by :

POLITEKNIK METrO KUALA LUMPUR
No. 2 – 14, Jalan Setiawangsa 10, Taman Setiawangsa
54200 Kuala Lumpur

2D Game Development (Labsheet)

First Published 2022

@ Politeknik METrO Kuala Lumpur

Perpustakaan Negara Malaysia

Cataloguing-in-Publication Data

M. F. Khodari

2D GAME DEVELOPMENT : (LABSHEET) / M. F. Khodari, A. Ali, N.H. Samad.

Mode of Access: Internet

eISBN 978-967-0074-05-4

1. Computer games.
2. Computer games--Programming.
3. Level design (Computer science).
4. Government publications--Malaysia.
5. Electronic books.

I. A. Ali. II. N. H. Samad.

III. Title.

794.8151

All rights reserved. No part of this publication may be reproduced, store in any retrieval system in any form or by any means, electronic, mechanical , photocopying, recording or otherwise without permission of the publisher

Preface

This eBook introduces you to a series of lab sheet to develop a 2D game (Space Shooter). It mainly focuses on the steps by steps guidance on how to develop a 2D game using Unity software.

2D GAME DEVELOPMENT introduces the techniques to develop computer games by using game development tool. The readers will be able to understand the concepts of sprites and object, game interactivity, level design and scripting in the development of game of different genres.

Since this eBook provides an outlook of the overall guidance to develop 2D game, after going through this material, you will find yourself at a moderate level. It will heaps your knowledge from basics to the next levels

Acknowledgement



The highest gratitude to Allah SWT because with His permission, this Introduction to 2D Game Development (Labsheet) was successfully published. This eBook is published as a guide or reference for students who take the 2D Game Development course at Malaysia Polytechnic. In preparing this eBook, various challenges and obstacles need to be faced before being able to produce an eBook. We would like to express our deepest gratitude to our family, the Polytechnic e-Learning Coordinator, and colleagues for their guidance and support in the production of this eBook.

We would also like to thank the following for permission to reproduce copyright photos:

- Unity
- Canva
- freepik.com

We hope that this ebook can be put to good use by all who use it.
Thank you.

M.F. Khodari, A. Ali, N.H. Samad,
July 2022

Table of Contents

Topic 1.0 : Introduction to 2D Game Assets	2
<p>Explains 2D Game development software and its usage, the resources in game development and the different type of 2D game genre available in market</p>	
Topic 2.0 : 2D Assets Creation, Import and Arrangement	13
<p>Covers scene set up, digital images and its usage in games, mizes the digital images into game development software, organize the layers to make up the scene for realistic gameplay</p>	
Topic 3.0 : Level Design	26
<p>Discusses sprite design for 2D game development software and perform level in game</p>	

Table of Contents

Topic 4.0 : Game Interactivity 39

Discusses display event and action in game, organize game mechanics to enable interaction between assets, display conflict and challenge

Topic 5.0 : Game Scripting 64

Manipulates the understanding of scripting in combining the assets

Appendix I

References

Topic 1.0 : Introduction to 2D Game Assets

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 1 : Introduction to 2D Game Assets

Next : LABSHEET 2 - Organizing Unity Files and Project

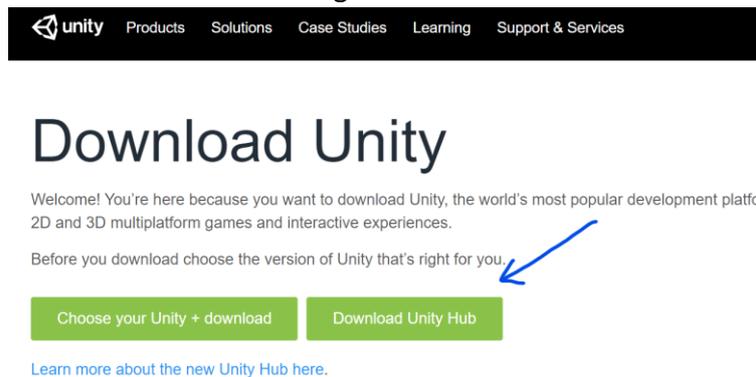
A) Installation 2D Game Development software

1. Installing Unity Hub

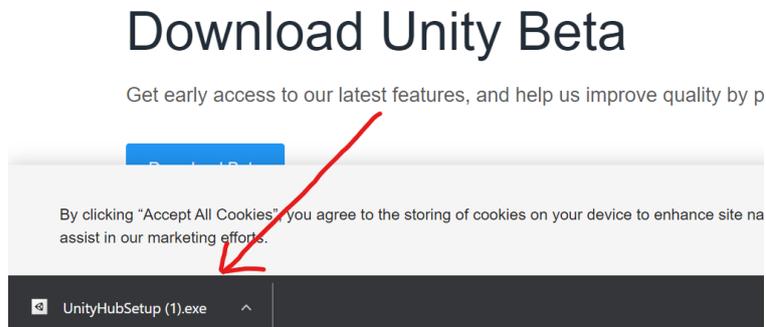
- a. The Unity Hub is a management tool that you can use to manage all of your Unity Projects and installations, To install the Unity Hub for Windows, MacOS, and Linux visit :

<https://unity3d.com/get-unity/download>

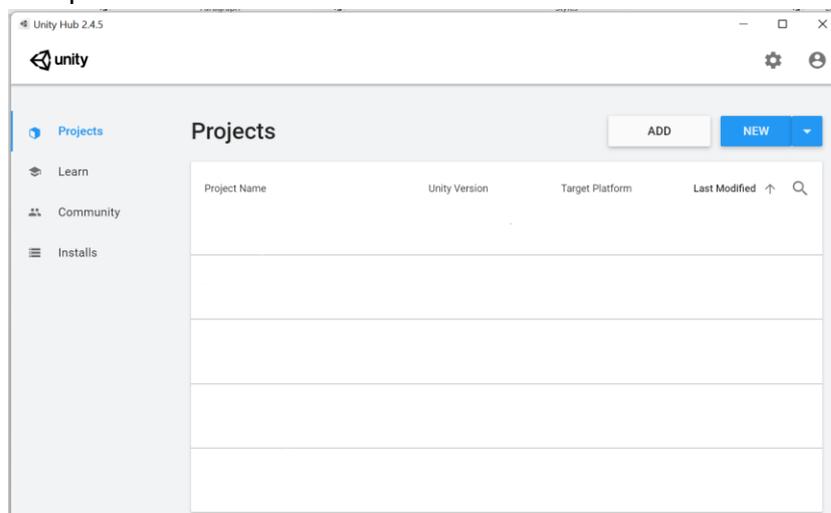
- b. Choose UNITY HUB to begin download installation file



- c. Download file and select the file to begin installation



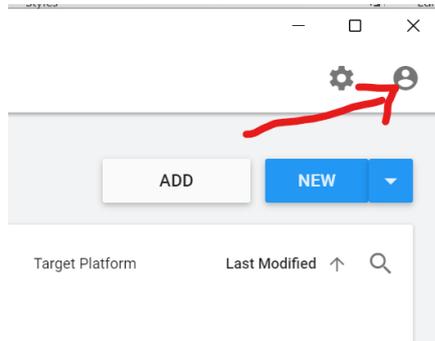
- d. Complete installation and run UNITYHUB



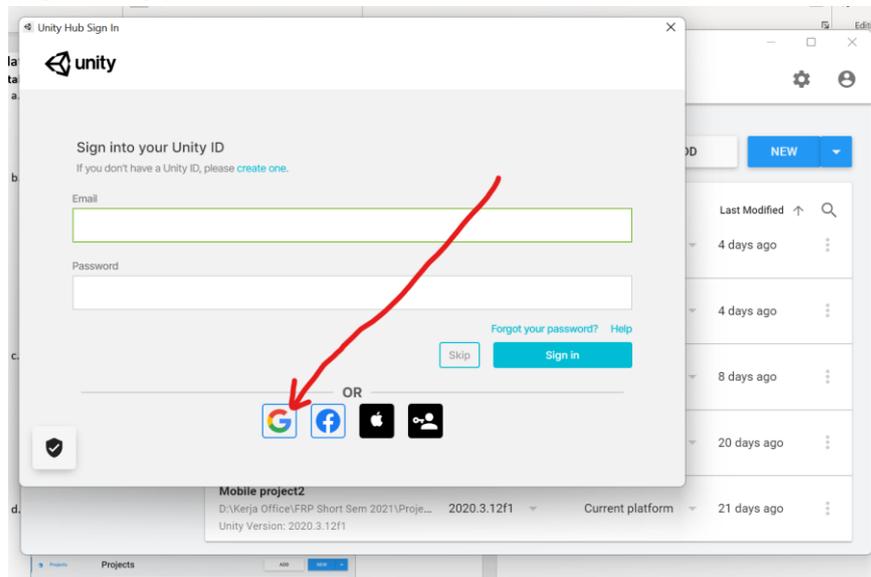
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 1 : Introduction to 2D Game Assets Next : LABSHEET 2 - Organizing Unity Files and Project

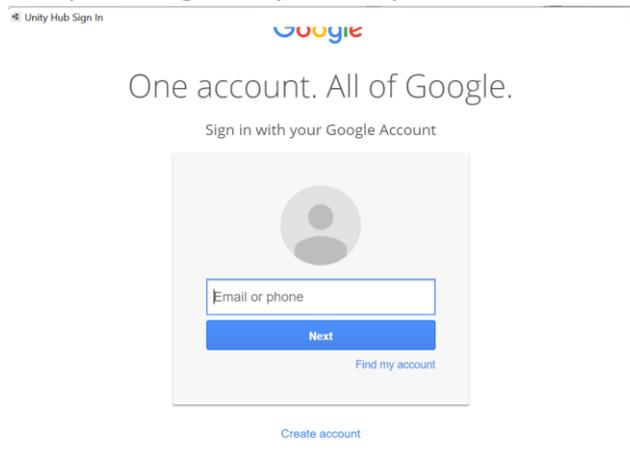
e. Create new UNITY Account by click on icon below :



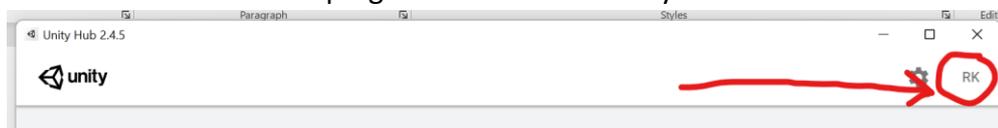
f. Login using your GMAIL account to create new user account



g. Complete sing-in step to complete UNITY account registration



h. Your Avatar at top right connect will show your name:



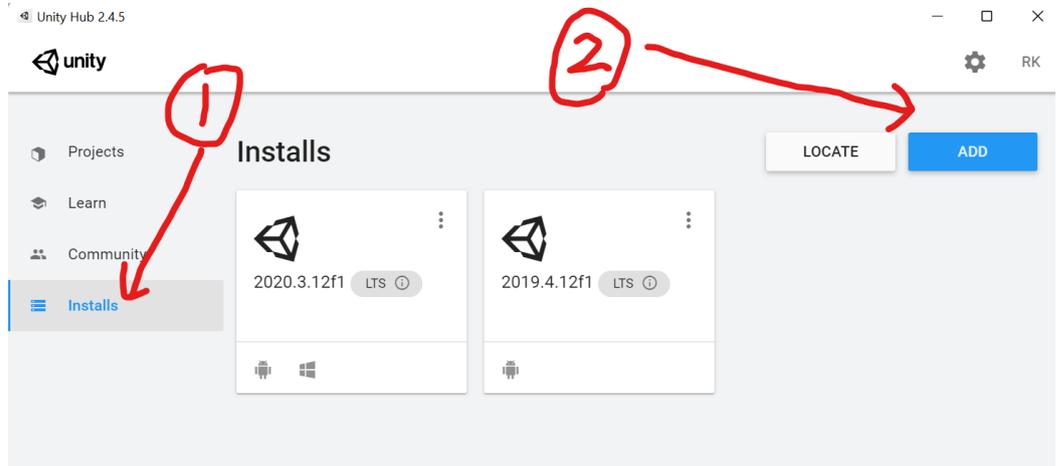
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 1 : Introduction to 2D Game Assets

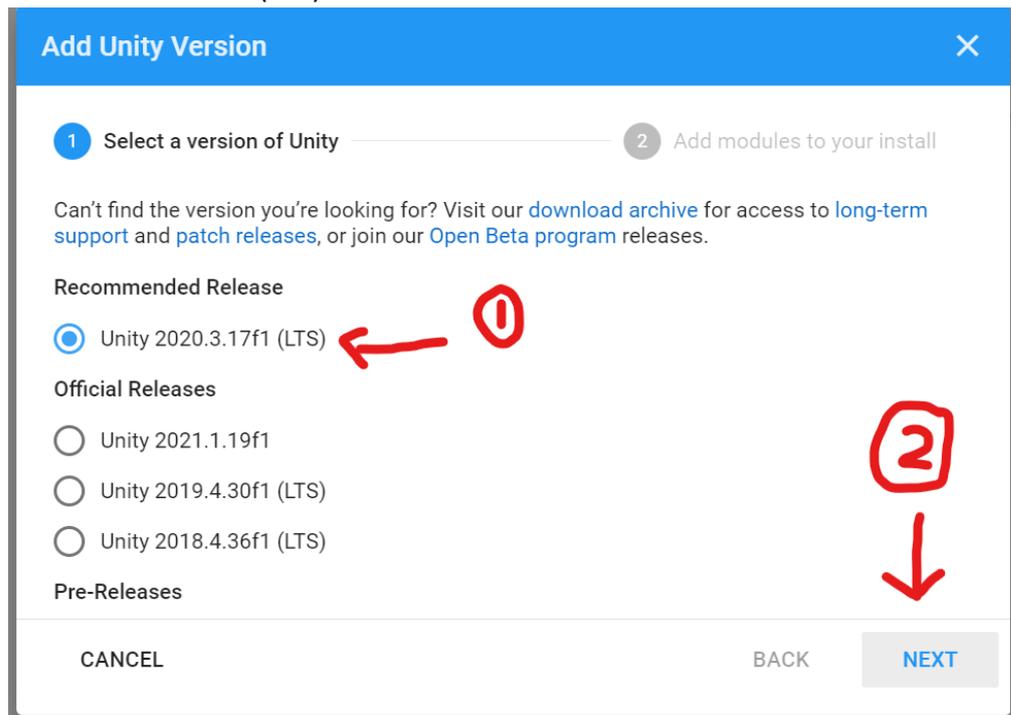
Next : LABSHEET 2 - Organizing Unity Files and Project

2. Installing UNITY ENGINE

- a. Run UNITY HUB and select INSTALL; then Select ADD button to add installation for new UNITY ENGINE VERSION



- b. Select 2020.3.22f1(LTS) and Select NEXT button



- c. Select Microsoft Visual Studio Community 2019 and Windows Build Support; then select NEXT button to proceed installation process.

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 1 : Introduction to 2D Game Assets

Next : LABSHEET 2 - Organizing Unity Files and Project

Add Unity Version ✕

<input checked="" type="checkbox"/>	Microsoft Visual Studio Community 2019	1.4 GB	1.3 GB
Platforms			
> <input type="checkbox"/>	Android Build Support	373.5 MB	1.9 GB
<input type="checkbox"/>	iOS Build Support	364.7 MB	1.6 GB
<input type="checkbox"/>	tvOS Build Support	361.5 MB	1.6 GB
<input type="checkbox"/>	Linux Build Support (IL2CPP)	103.5 MB	435.8 MB
<input type="checkbox"/>	Linux Build Support (Mono)	102.7 MB	428.1 MB
<input type="checkbox"/>	Mac Build Support (Mono)	321.2 MB	1.8 GB
<input type="checkbox"/>	Universal Windows Platform Build Support	289.1 MB	2.1 GB
<input type="checkbox"/>	WebGL Build Support	319.5 MB	1.2 GB
<input checked="" type="checkbox"/>	Windows Build Support (IL2CPP)	82.0 MB	432.7 MB

CANCEL BACK NEXT

d. Agree condition and select DONE button

End User License Agreement ✕

Visual Studio 2019 Community License Terms

Please review and accept the license terms before downloading and installing Microsoft Visual Studio.
<https://go.microsoft.com/fwlink/?linkid=2092534>

I have read and agree with the above terms and conditions

CANCEL DONE

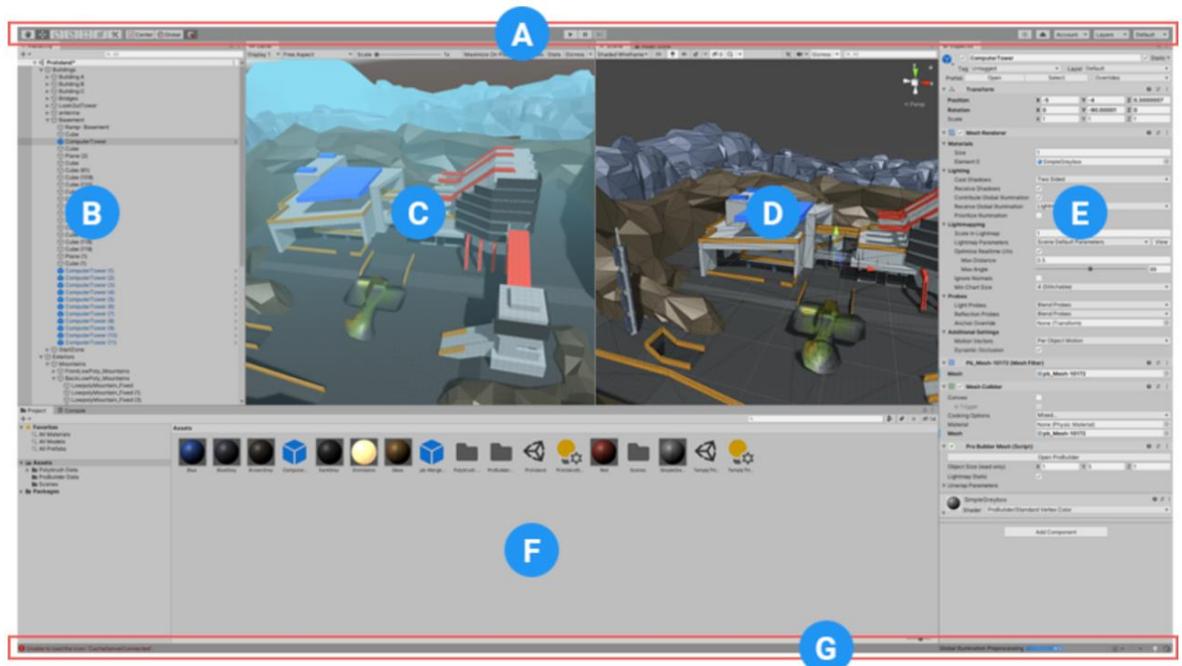
e. Wait until installation complete

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 1 : Introduction to 2D Game Assets

Next : LABSHEET 2 - Organizing Unity Files and Project

B) GAME ENGINE'S INTERFACE



(A) The Toolbar : provides access to the most essential working features. On the left it contains the basic tools for manipulating the Scene view and the GameObjects within it. In the centre are the play, pause and step controls. The buttons to the right give you access to Unity Collaborate, Unity Cloud Services and your Unity Account, followed by a layer visibility menu, and finally the Editor layout menu (which provides some alternate layouts for the Editor windows, and allows you to save your own custom layouts).

(B) The Hierarchy window : is a hierarchical text representation of every GameObject in the Scene. Each item in the Scene has an entry in the hierarchy, so the two windows are inherently linked. The hierarchy reveals the structure of how GameObjects attach to each another.

(C) The Game view : simulates what your final rendered game will look like through your Scene Cameras . When you click the Play button, the simulation begins.

(D) The Scene view : allows you to visually navigate and edit your Scene. The Scene view can show a 3D or 2D perspective, depending on the type of Project you are working on.

(E) The Inspector Window : allows you to view and edit all the properties of the currently selected GameObject. Because different types of GameObjects have different sets of properties, the layout and contents of the Inspector window change each time you select a different GameObject.

(F) The Project window : displays your library of Assets that are available to use in your Project. When you import Assets into your Project, they appear here.

(G) The status bar : provides notifications about various Unity processes, and quick access to related tools and settings.

DFG40323 - 2D GAME DEVELOPMENT

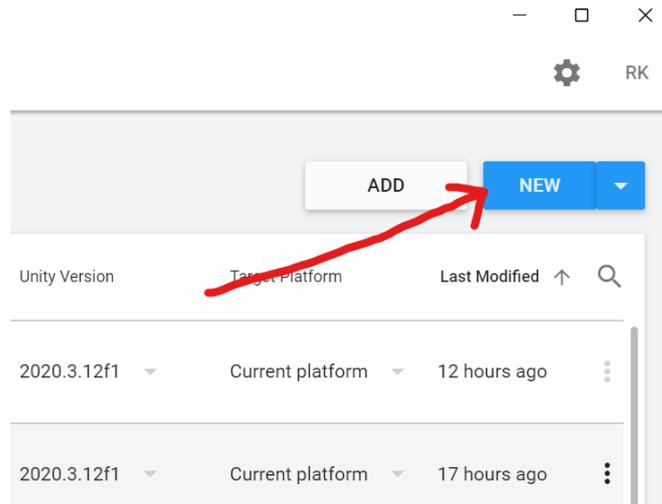
Now : LABSHEET 1 : Introduction to 2D Game Assets

Next : LABSHEET 2 - Organizing Unity Files and Project

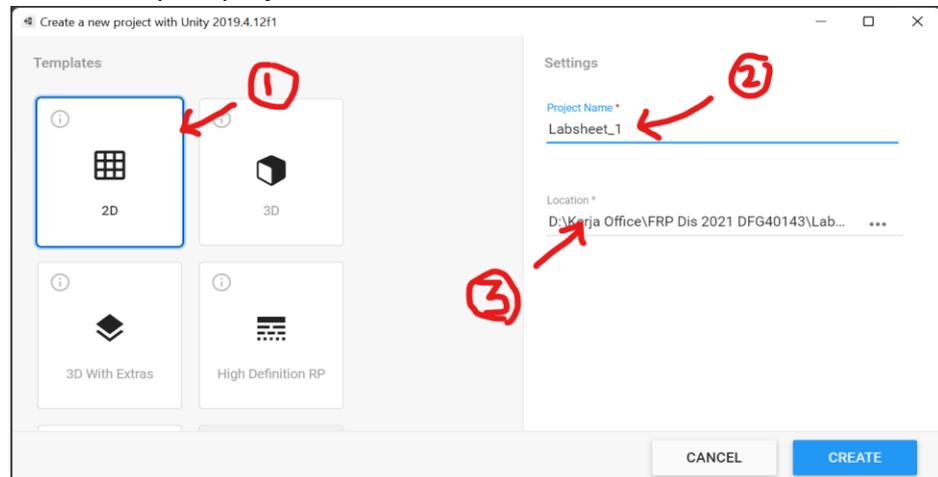
C) RESOURCES IN GAME DEVELOPMENT

1. Creating New Unity Project File

- Login to your CIDOS/ Google classroom portal account, and download “**labsheet1_resource.unitypackage**” from download section.
- Open UNITY HUB to create new project; select NEW button at the top right corner of UNITY HUB window



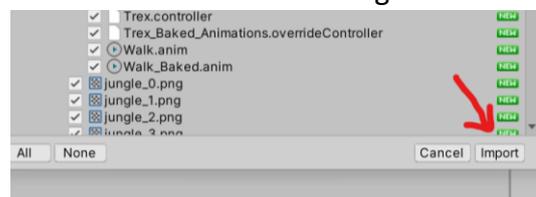
- Next, choose **2D** from template, then name or project as “**labsheet1**” and locate your project file into New Folder called “**Labsheet**”.



- Select CREATE button to create the project

2. Importing unitypackage file into project

- Select and double click (**2X**) at “**labsheet1_resources.unitypackage**” to perform installation new unitypackage file.
- Select IMPORT button to begin installation



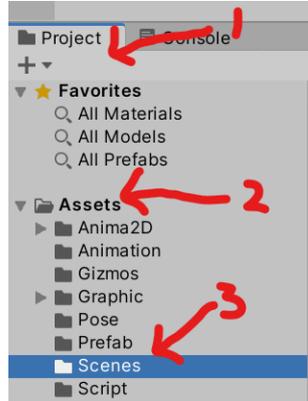
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 1 : Introduction to 2D Game Assets

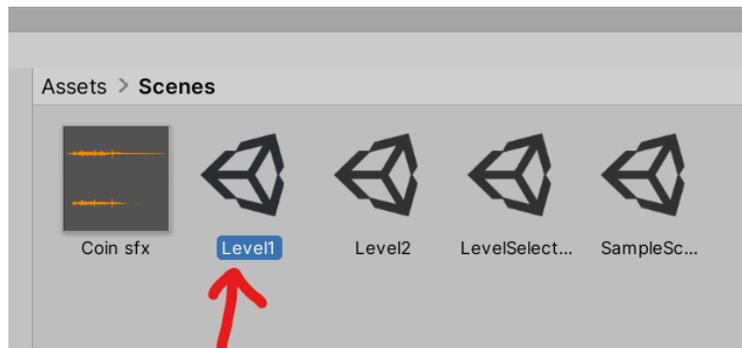
Next : LABSHEET 2 - Organizing Unity Files and Project

3. Opening Game Scene

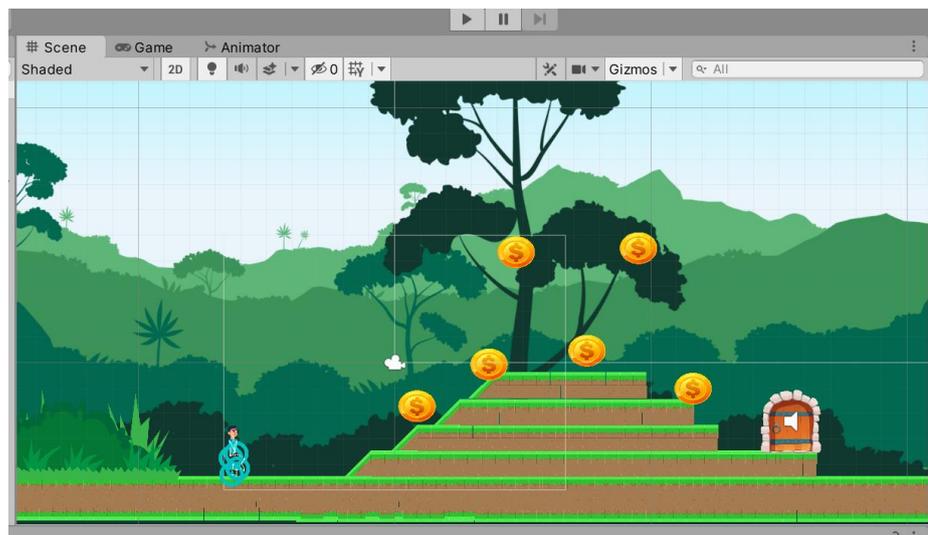
- a. Select **"Scene"** folder in **Project Panel** inside **"Assets"** folder



- b. Select and Double Click (2X) on scene file called **"level 1"** inside **Scene folder**



- c. Your Scene Panel will show as below :



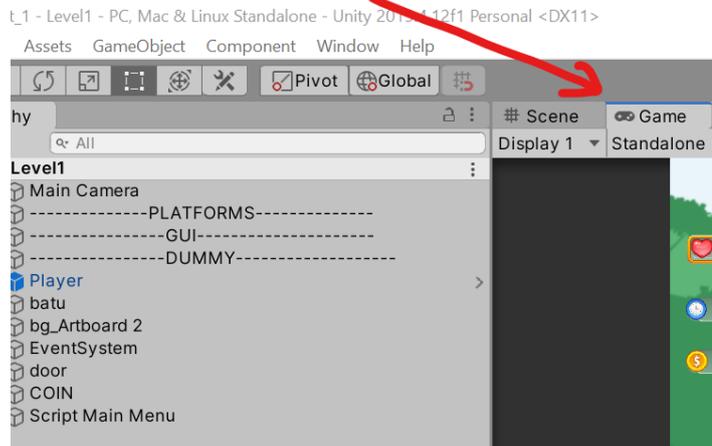
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 1 : Introduction to 2D Game Assets

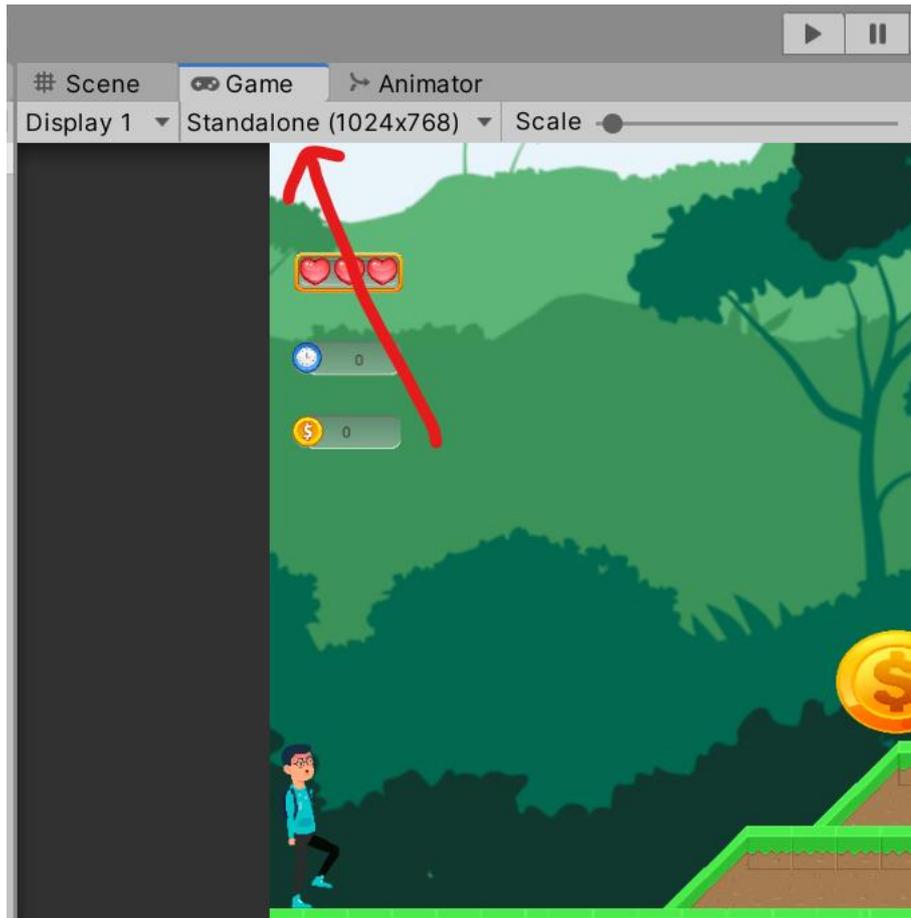
Next : LABSHEET 2 - Organizing Unity Files and Project

D) GAME RESOLUTION SETTING

1. Open previous game project (C.1)
2. Select Game View



3. From DISPLAY 1 select STANDALONE (1024X768)

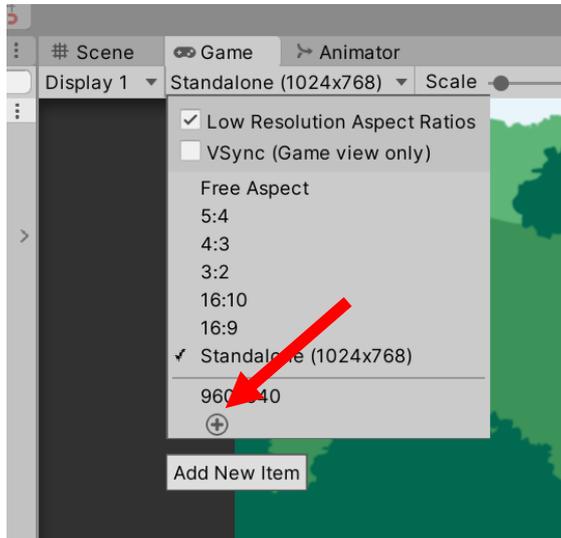


DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 1 : Introduction to 2D Game Assets

Next : LABSHEET 2 - Organizing Unity Files and Project

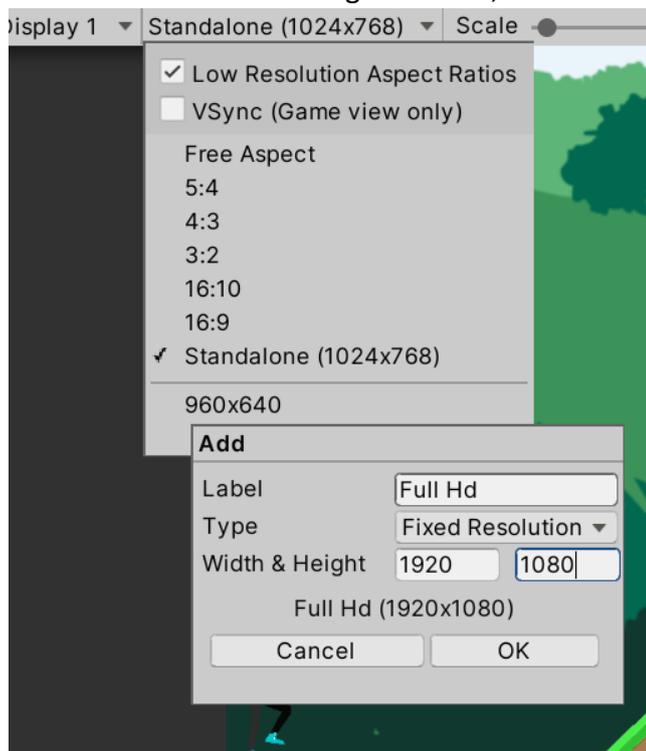
4. Select (+) button to add new Resolution



5. Insert New value as below :

a. LABEL = Full HD

b. Width & Height = 1920 ; 1080

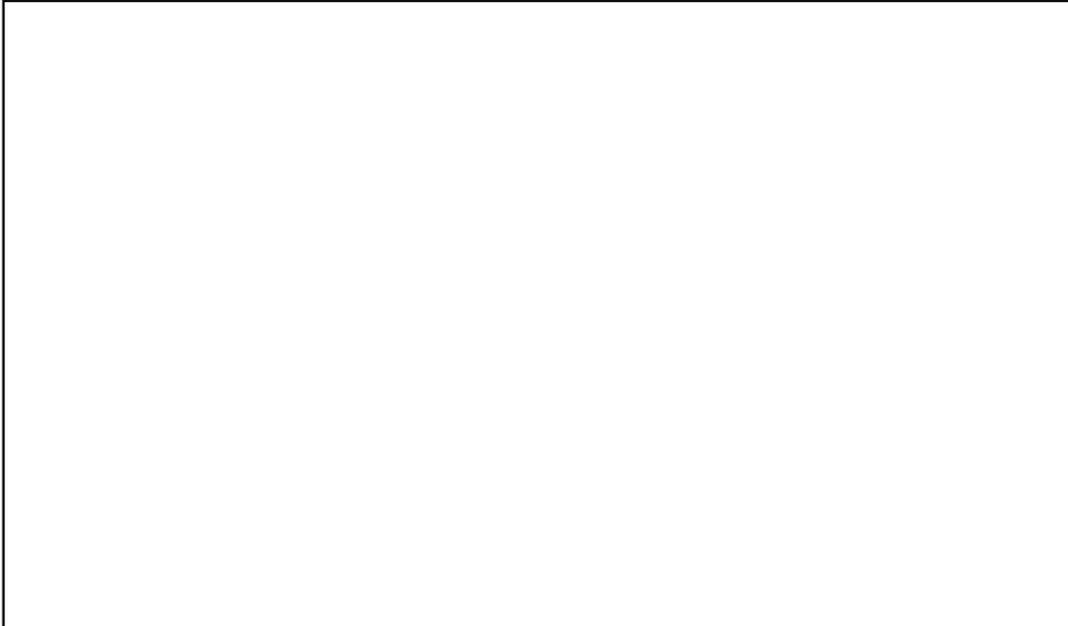


6. Select OK button

EXERCISE 1
DFG40323 2D GAME DEVELOPMENT

Fill the blank with **CORRECT** image from unity project (labsheet_1) : (use snapping tool)

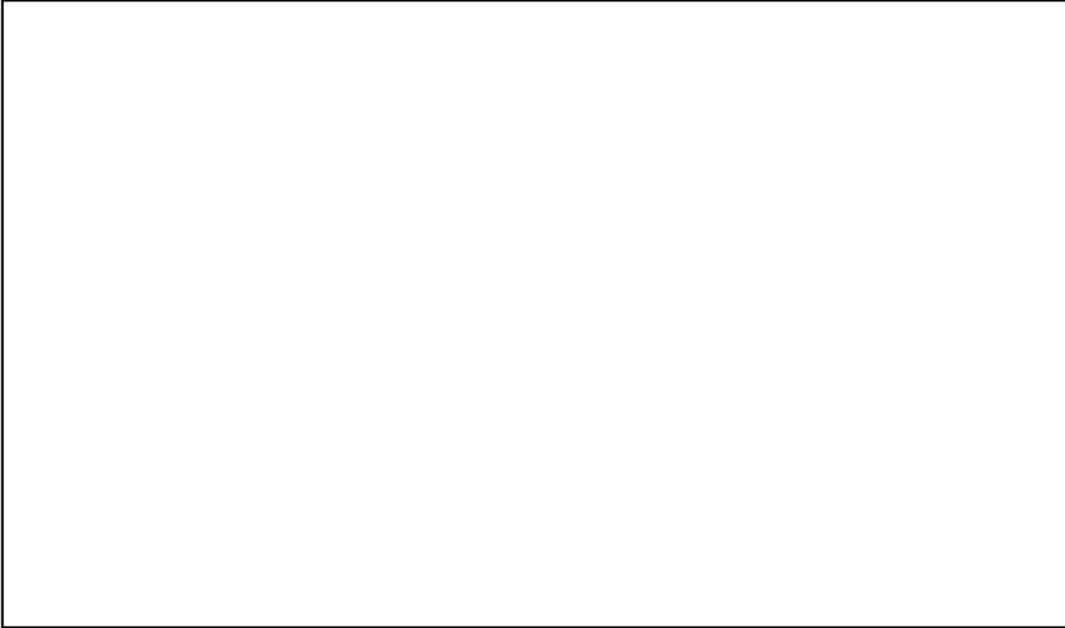
1. **GAMEOBJECT** is the most important concept in the Unity Editor. Every object in your game is a GameObject, from characters and collectible items to lights, cameras and special effects.



2. **SCENE** are where you work with content in Unity. They are assets that contain all or part of a game or application.



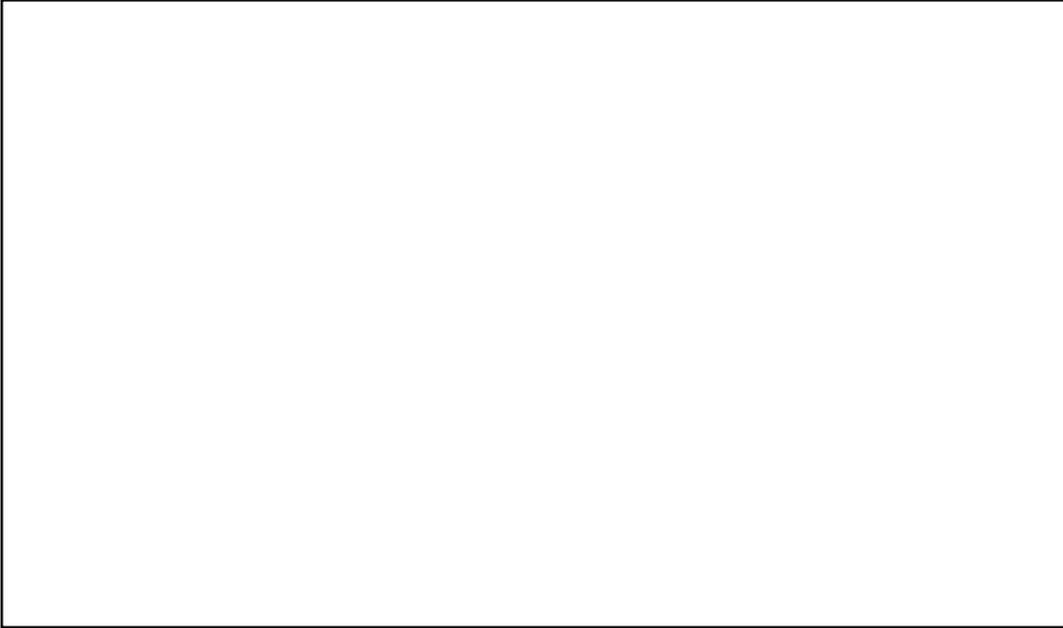
3. **Audio** can be it background music or sound effects. It can import most standard audio file formats and has sophisticated features for playing sounds in 3D space, optionally with effects like echo and filtering applied.



4. **Font** used to display the text. Text can be used to provide captions or labels for other GUI controls or to display instructions or other text.



5. **Script** is an essential ingredient in all applications you make in Unity. Most applications need scripts to respond to input from the player and to arrange for events in the gameplay to happen when they should. Beyond that, scripts can be used to create graphical effects, control the physical behaviour of objects or even implement a custom AI system for characters in the game.



Topic 2.0 : 2D Assets Creation, Import and Arrangement

DFG40323 - 2D GAME DEVELOPMENT

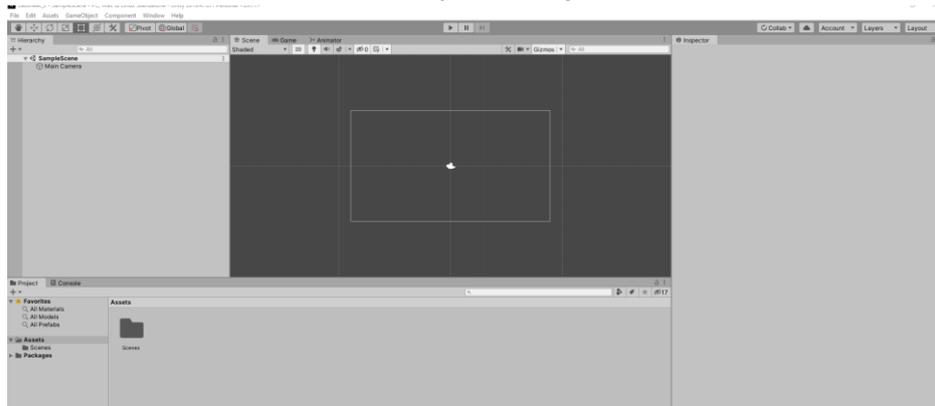
Now : LABSHEET 2 : Present animation with 2D assets for realistic gameplay

Next : LABSHEET 3 – Level Design

A) Setup new game project

1. Setting up game development environment

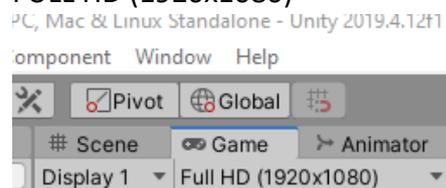
- Open UNITY HUB to create new project.
- Next, choose 2D from template, then name or project as “Labsheet2-game2D” and locate your project file into New Folder called “Labsheet”.
- Wait until your project are complete to be created.
- Below is default interface of Unity new Project



- Login into your CIDOS/ google classroom account and download “Labsheet2_Part2_Resources.unitypackage”
- Double click on the downloaded file to install into your project.
- Click “Import” button



- As result, a folder called Graphics will exist into your Assets folder.
- Access to GAME VIEW tab, and make sure your Display1 setting is set to FULL HD (1920x1080)



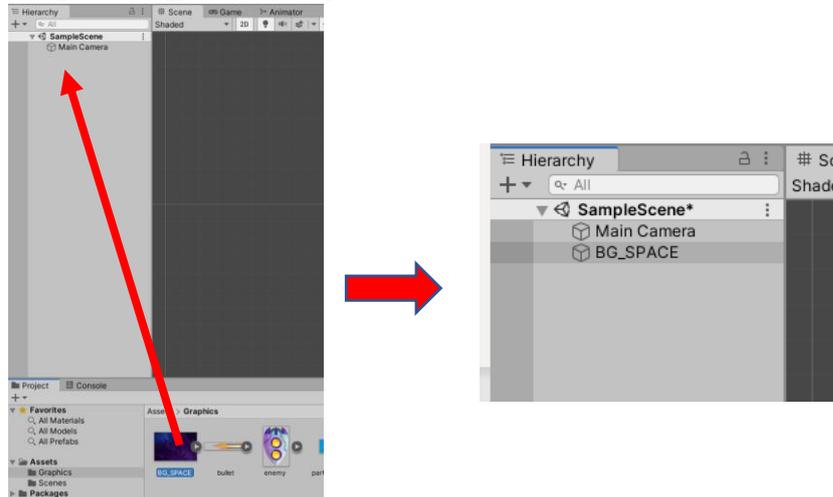
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 2 : Present animation with 2D assets for realistic gameplay

Next : LABSHEET 3 – Level Design

2. Creating your first game scene

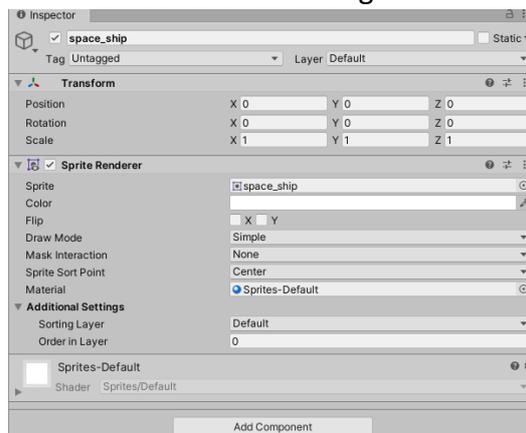
- Firstly, we need to add background into your game scene.
- Drag the BG_SPACE image from the GRAPHICS folder to the hierarchy tab.



- Now, drag the SPACE_SHIP to the hierarchy tab (Refer step b)
- Your result will show as below



- Select the SPACE_SHIP in the hierarchy tab. You can see that the inspector tab is now full of interesting information about the SPACE_SHIP.



- From Inspector Panel, change the value for SPACE_SHIP information as below:

Position : X=0;Y=-4;Z=0
Rotation : X=0;Y=0;Z=0
Scale : X=0.3;Y=0.3;Z=0.3

DFG40323 - 2D GAME DEVELOPMENT

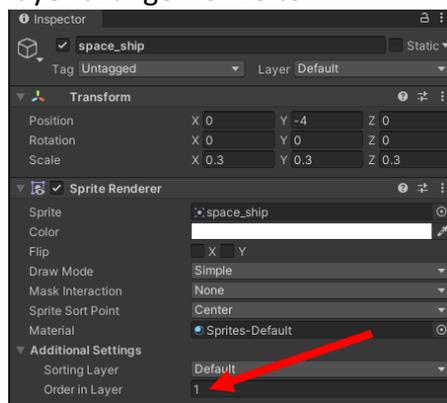
Now : LABSHEET 2 : Present animation with 2D assets for realistic gameplay

Next : LABSHEET 3 – Level Design

g. The result will show as below

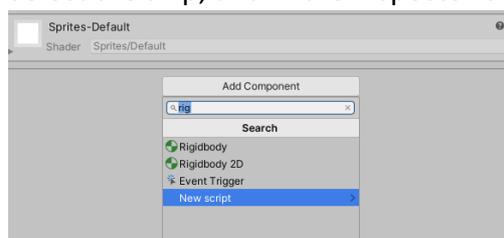


h. If you SPACE_SHIP is hidden (not appear on the screen after drag and drop), click SPACE_SHIP at Hierarchy tab. On the Inspector tab, order in layer change from 0 to 1.

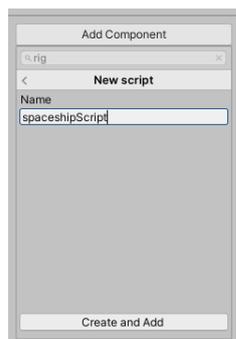


3. Adding movement script

a. Select the ship, and in the inspector click Add Component → New script



b. Set the name to “spaceshipScript” → Create and Add

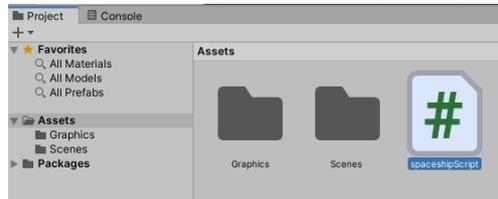


DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 2 : Present animation with 2D assets for realistic gameplay

Next : LABSHEET 3 – Level Design

- c. Find the “spaceshipScript” in the project tab and place it inside the Scripts folder (right click inside Assets area Create > Folder > rename to Scripts). Then double click on the script.

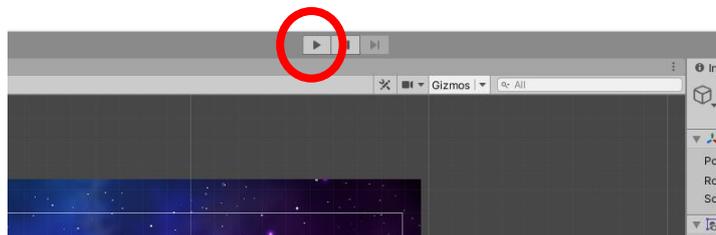


1. Make sure you already install MICROSOFT VISUAL STUDIO COMMUNITY 2019 into your machine (Refer LABSHEET 1: Introduction to 2D Game Assets; Step A.2.c)
2. Make sure you already register and login into MICROSOFT ACCOUNT

- d. Replace the default code by this (refer line 19) > Ctrl + S to save to script > close script window by click 

```
spaceshipScript.cs
Assembly-CSharp
spaceshipScript
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5
6 public class spaceshipScript : MonoBehaviour
7 {
8
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
18
19        transform.Translate(Input.GetAxis("Horizontal") * 100f * Time.deltaTime, 0f, 0f);
20
21    }
22
23 }
```

- e. Play your Scene by hit Play Button



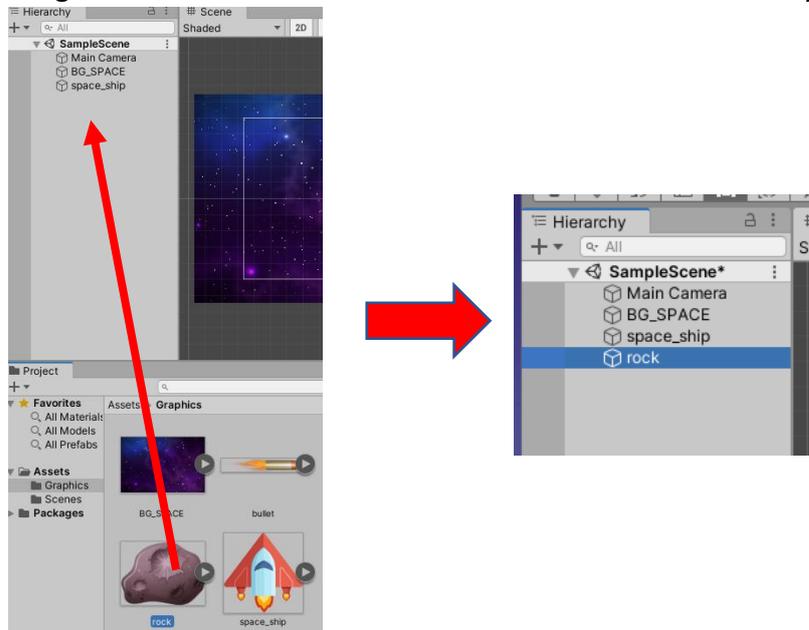
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 2 : Present animation with 2D assets for realistic gameplay

Next : LABSHEET 3 – Level Design

4. Adding gameplay element (Obstacle)

- First let's start by creating some obstacles.
- Drag the "ROCK" from the GRAPHICS folder into the hierarchy tab



- Next, add a script component to the **ROCK** called "rockScript". (Refer step A.3.a)
- Add "rockScript" with this line by double-click to modify the script (Refer step A.3.b)

```
rockScript.cs
Miscellaneous Files
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class rockScript : MonoBehaviour
6 {
7
8     private Rigidbody2D rb;
9
10    void Awake()
11    {
12        //To attach component Rigidbody2D into Rock
13        rb = gameObject.AddComponent<Rigidbody2D>() as Rigidbody2D;
14        //To set Kinematic function on Rock
15        rb.bodyType = RigidbodyType2D.Kinematic;
16    }
17
18    // Start is called before the first frame update
19    void Start()
20    {
21        //To make the rock rotate randomly
22        rb.angularVelocity = Random.Range(-200, 200);
23    }
24
25    // Update is called once per frame
26    void Update()
27    {
28        //To make the rock falling down with speed was -5
29        rb.velocity = new Vector2(0.0f, -5.0f);
30    }
31
32 }
```

- Modify your script as below.

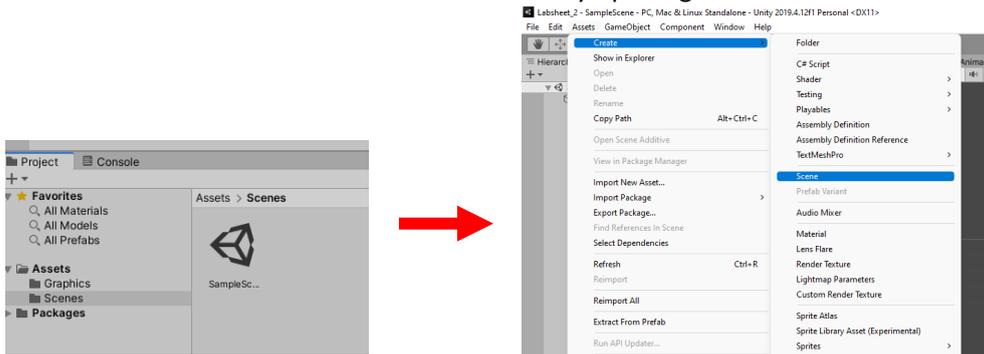
DEPARTMENT OF INFORMATION TECHNOLOGY AND COMMUNICATION

Practical Task (s) No: 1 / 2 / 3 / 4

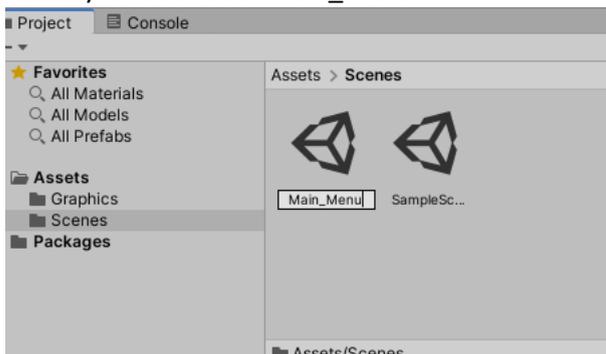
CLO 1: Display 2D assets, game interactivity, level design and scripting skills of game development tools to maintain coherency in game. (P4, PLO3)

Instruction: Manipulating Unity Assets to build your first game scene interface

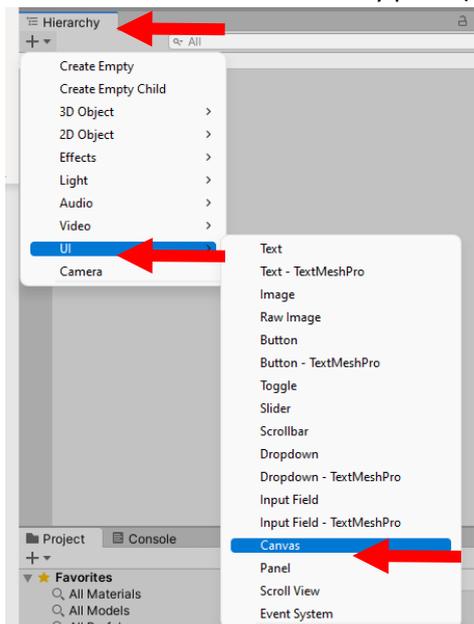
1. Create new scene inside Scene Folder in Assets by opening Scene Folder inside Assets Folder



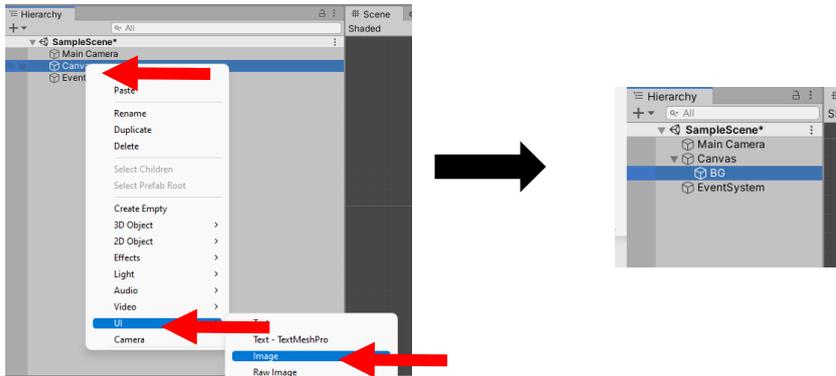
2. Then access menu bar to create new scene (Menu bar → Assets → Create → Scene)
3. Name your Scene as “Main_Menu”



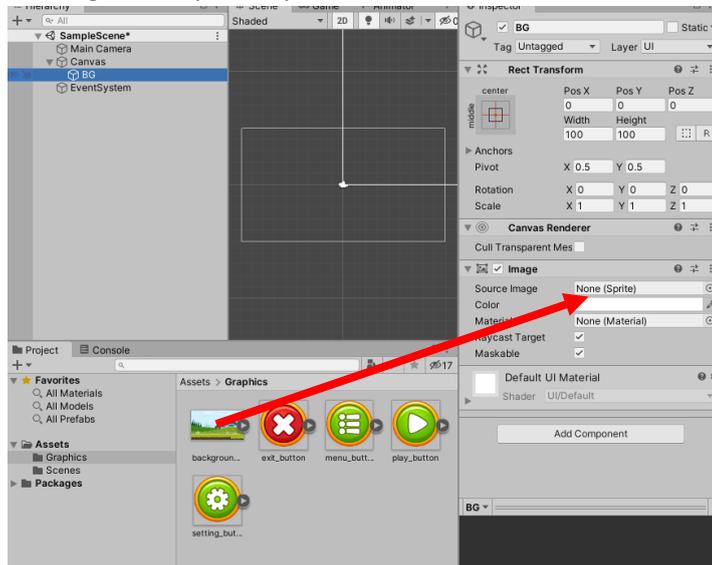
4. Create a canvas in the hierarchy pane (+ button → UI → Canvas).



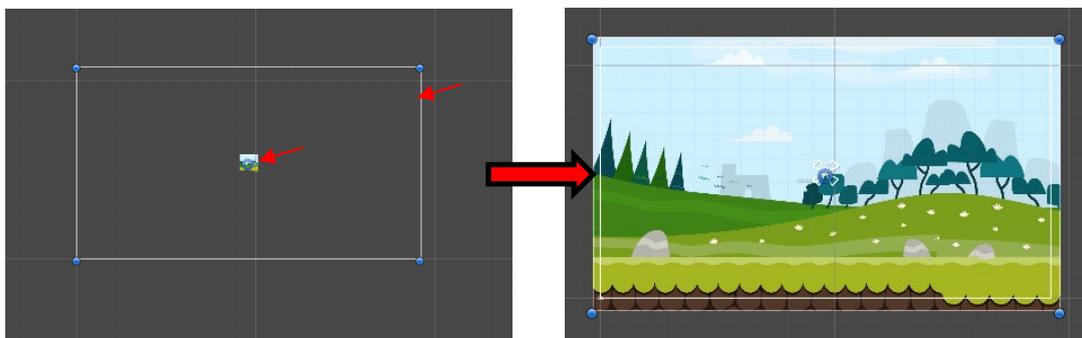
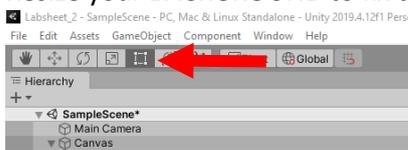
- Under the canvas create an image to insert background image.
(**Right click** at canvas → UI → Image) Rename the image as **BG**.



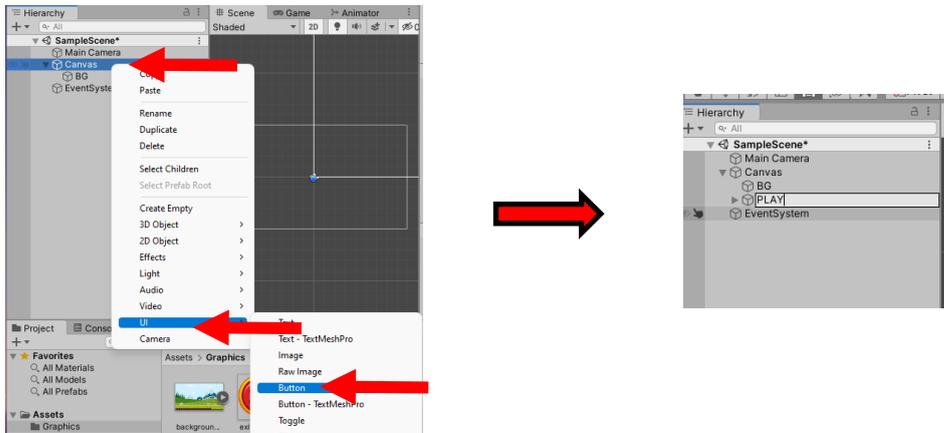
- Change the image by **dragging and dropping** “background” inside Graphics folder into **Source Image** at inspector panel as shown below



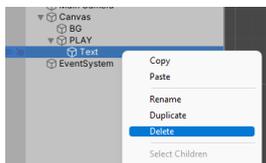
- Double Click at Canvas from Hierarchy Panel to show your screen
- Resize your **BACKGROUND** to fix the whole canvas using **RECT TOOL**



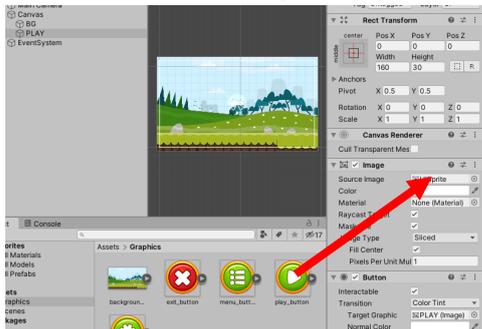
9. Then create a button under the canvas. (**Right click** at canvas → UI → Button). Rename it as Play.



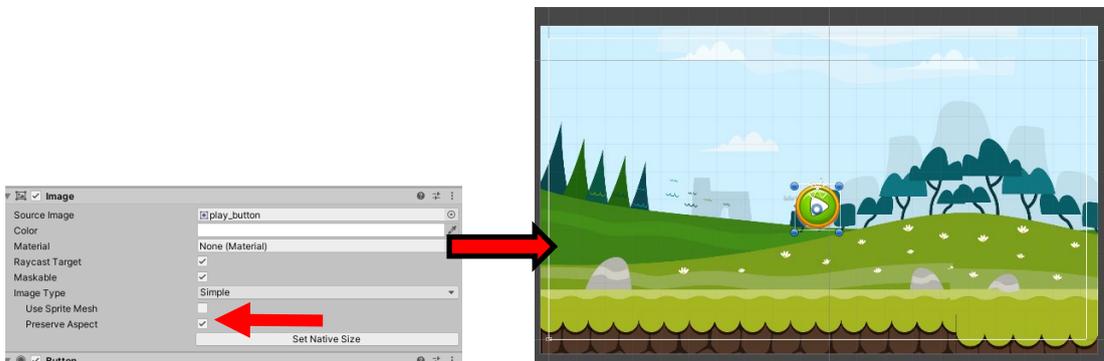
10. Drop down at PLAY button and delete Text item.



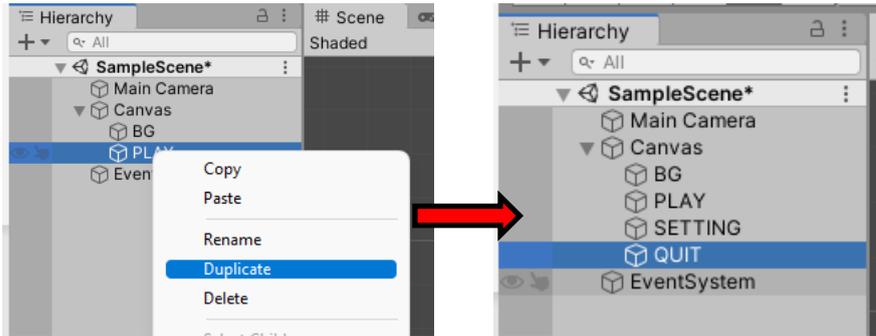
11. **Select PLAY** button inside Hierarchy panel, and drag the Green Play button from Graphic Folder into Source Image at Inspector Panel.



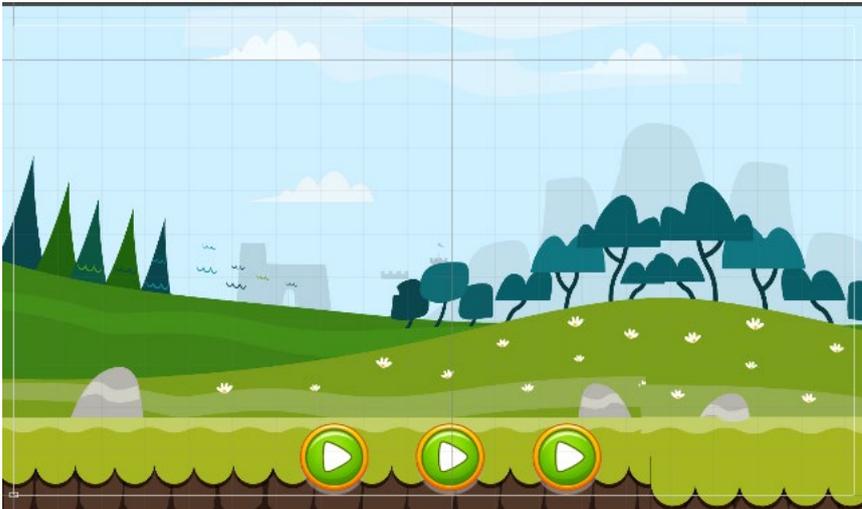
12. Tick **PRESERVE ASPECT** at Inspector Panel as below and **resize your button** using RECT TOOL at menu bar:



13. Duplicate PLAY button at Hierarchy panel by (Right Click -> Duplicate) ; rename as SETTING and QUIT

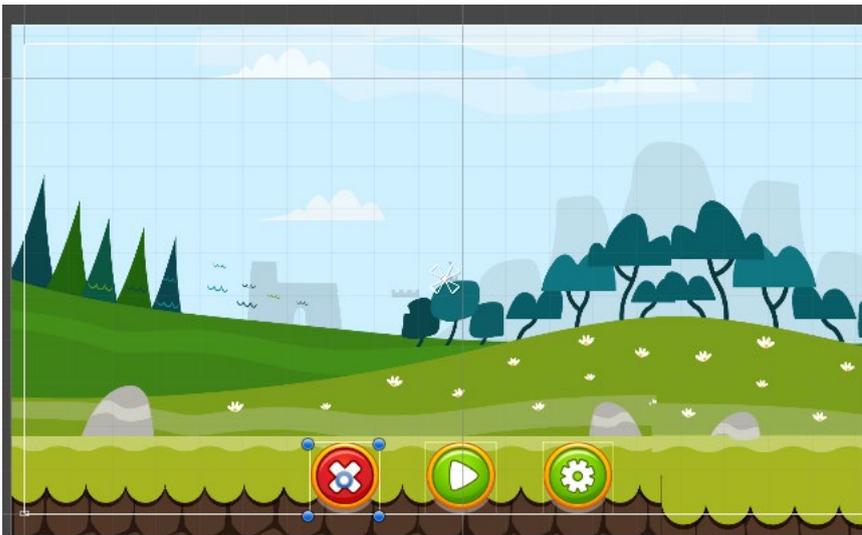


14. Arrange your buttons as shown below



15. Next, change Image Source for SETTING and EXIT button by repeating step at **10**

16. **Result** as shown below :



17. Now go to Cooltext Graphics Generator (<https://cooltext.com/>) and generate the wording of **MY GAME** using different graphics. You also can design using Adobe Photoshop or Adone Illustrator.
18. Import or Paste the graphic into the asset folder. (Refer step Labsheet 2 → A.1.e)
19. Create new Image into your Hierarchy Panel (Refer step 4), name as LOGO.
20. Change the image by **dragging and dropping** "Title Graphic" from **Step 16** into Source Image at inspector panel (Refer step 5)
21. Arrange your design as your own style.
22. Submit your Practical Task 1 at CIDOS.
23. PASTE your OUTPUT below

Name: _____

Registration Num.: _____

Code & Course: DFG40143 - 2D GAME DEVELOPMENT

Programme/Class: _____

Practical Task (s) No: 1 / 2 / 3 / 4

MARKS: /20

CLO 1: Display 2D assets, game interactivity, level design and scripting skills of game development tools to maintain coherency in game. (P4, PLO3)

1. PASTE your OUTPUT below :

MOHD FAKHURRAZI BIN KHODARI

Pensyarah

Jabatan Teknologi Maklumat dan Komunikasi

Politeknik METRO Kuala Lumpur

DFG40323 - 2D GAME DEVELOPMENT EXERCISE 2

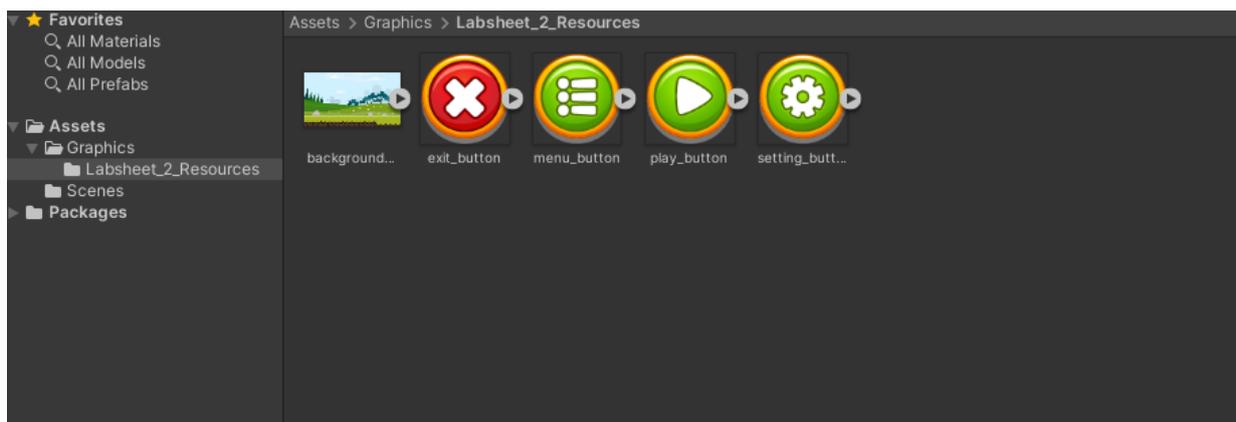
Code & Course: DFG40143 2D GAME DEVELOPMENT

- 1) Write down the path of your saved Labsheet2 project below:
- 2) How many folders exist in Assets Folder and list down all of them?

Scene and Graphics

.....

- 3) Snap and paste the content of Graphics folder inside Assets in Unity Project.



DFG40323 – 2D GAME DEVELOPMENT EXERCISE 2B

Code & Course: DFG40143 2D GAME DEVELOPMENT

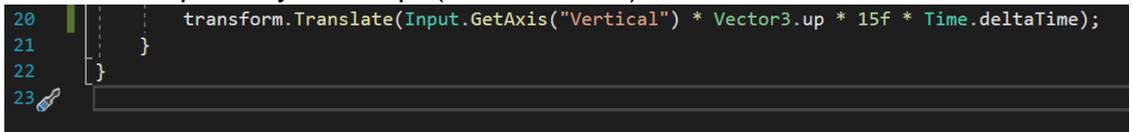
- 1) What is the function of Position in Inspector panel? (Refer step A.2.e)

.....
.....

- 2) Change the script at A.3.d to slow down the movement of the ship. Write down your script below:

.....
.....

- 3) Add this script into your script (AT LINE #20) and describe the result:



```
20         transform.Translate(Input.GetAxis("Vertical") * Vector3.up * 15f * Time.deltaTime);  
21     }  
22 }  
23
```

- 4) What is the final result of this labsheet?

.....
.....

Topic 3.0 : Level Design

DFG40323 - 2D GAME DEVELOPMENT

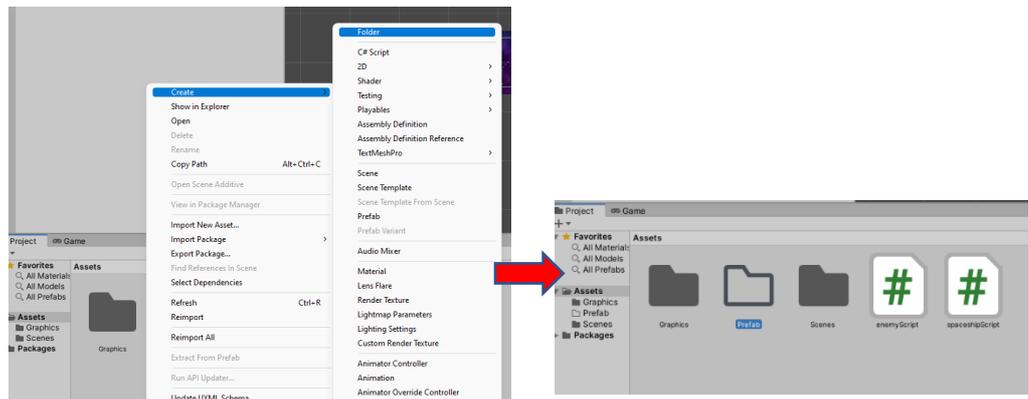
Now : LABSHEET 3 : Level Design (Part 1)

Next : Practical Task 2

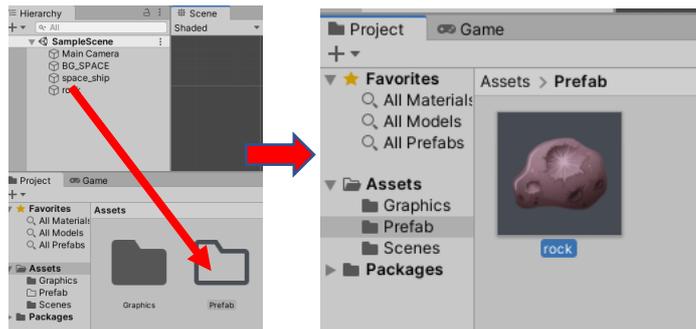
A) Creating basic level design

1. Generating enemies

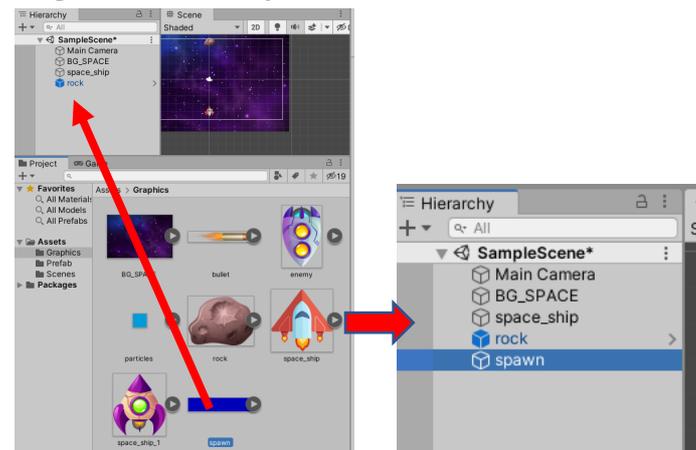
- a. Open your previous project.
- b. Create new folder inside Assets Folder and name it as "Prefab" by right click inside Assets Folder → Create → Folder.



- c. Next, drag the ROCK into the Prefabs folder.



- d. Then, delete the ROCK from the hierarchy by right click → delete.
- e. Drag the SPAWN object from the GRAPHICS folder into the hierarchy tab.

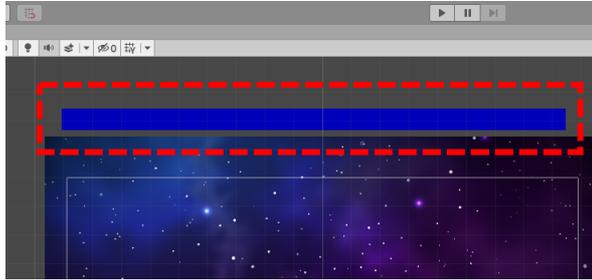


DFG40323 - 2D GAME DEVELOPMENT

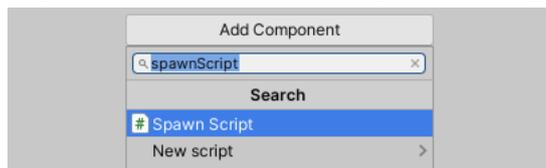
Now : LABSHEET 3 : Level Design (Part 1)

Next : Practical Task 2

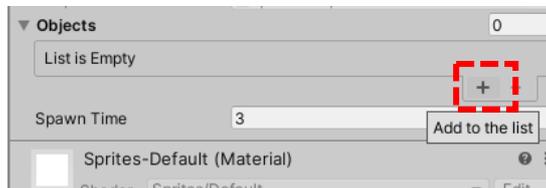
- f. Resize and reposition SPAWN object just above the background using RECT TOOL. See the result below:



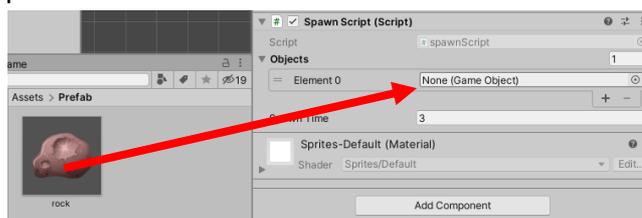
- g. Now, let's try to import **script** from outside into project.
h. Download script called "spawnScript" from CIDOS.
i. Import the script into your project (**Refer Labsheet 1 to import object into project folder**).
j. Attach SPAWN (from hierarchy panel) to the spawnScript (Select SPAWN → Inspector Panel → Add Component → type "spawnScript").



- k. Click at (+) button to add new Object to SPAWN.

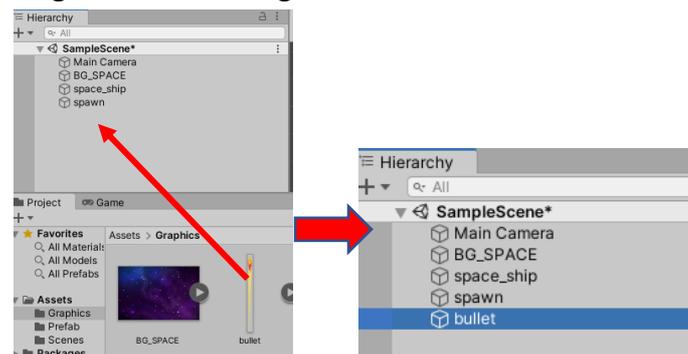


- l. Drag ROCK from PREFAB folder into [None (Game Object)] inside inspector panel. See below:



2. Generating Weapon

- a. Drag the BULLET image from the GRAPHICS folder to the hierarchy tab.

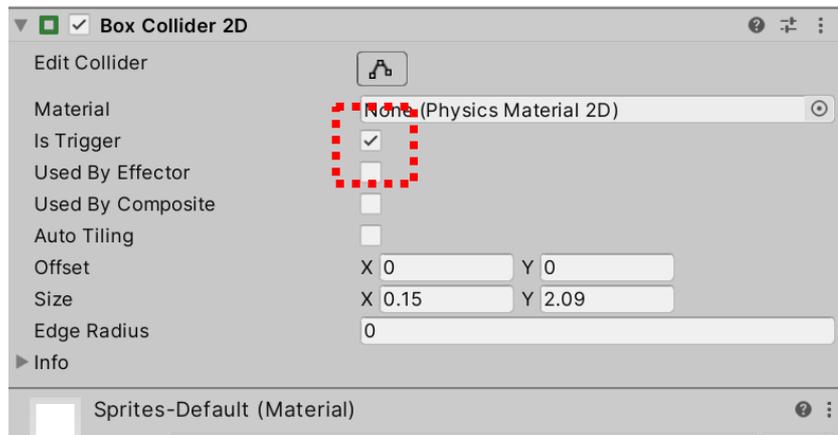


DFG40323 - 2D GAME DEVELOPMENT

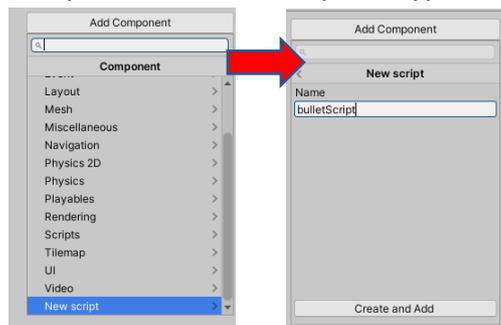
Now : LABSHEET 3 : Level Design (Part 1)

Next : Practical Task 2

- b. Add a “Box Collider 2D” component to it and make sure you have TICK at “is Trigger” on Inspector panel.



- c. Create new script on the BULLET (Select BULLET on Hierarchy → Add Component → New Script → Type “bulletScript” → Create and Add).`o



- d. Find “bulletScript” inside Assets folder and double click to modify.

```
Unity Script | 0 references
5 public class bulletScript : MonoBehaviour
6 {
7     private Rigidbody2D rb;
8     void Start()
9     {
10    }
11 }
12 private void Awake()
13 {
14     rb = gameObject.AddComponent<Rigidbody2D>() as Rigidbody2D;
15     rb.bodyType = RigidbodyType2D.Kinematic;
16 }
17
18 // Update is called once per frame
19 void Update()
20 {
21 }
22     rb.velocity = new Vector2(0.0f, 5.0f);
23 }
24 }
```

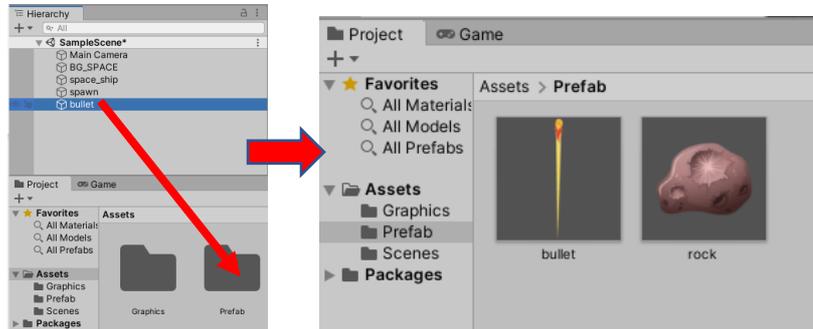
- e. Make sure you add new line script at (line 7,line 12, line 14, line 15, line 22).

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 3 : Level Design (Part 1)

Next : Practical Task 2

- f. Drag and drop the BULLET from the hierarchy tab to the Prefabs folder.



- g. Then delete the bullet from the scene.

3. Fire Bullet

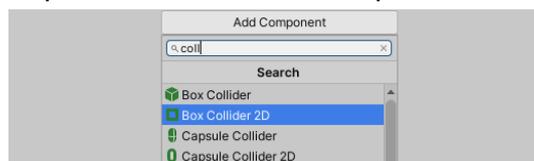
- a. To make the spaceship fire the bullets, we will need to make some changes to our "spaceshipScript".
- b. Find "spaceshipScript" inside Assets folder. Open it and edit it as below:

```
6 public class spaceshipScript : MonoBehaviour
7 {
8
9     public GameObject bullet;
10    // Start is called before the first frame update
11    void Start()
12    {
13    }
14
15    // Update is called once per frame
16    void Update()
17    {
18
19        transform.Translate(Input.GetAxis("Horizontal") * 50f * Time.deltaTime, 0f, 0f);
20        transform.Translate(Input.GetAxis("Vertical") * Vector3.up * 8f * Time.deltaTime);
21
22
23        if (Input.GetMouseButtonDown(0))
24        {
25            Instantiate(bullet, transform.position, Quaternion.identity);
26        }
27    }
28 }
29
30
```

- c. Make sure you add new line script at (line 24 to line 27).
- d. Play your scene.

4. Destroying Enemy

- a. The spaceship, the enemies, and the bullets should collide with each other.
- b. Add **Box Collider 2D** component to the **SPACESHIP** (Select on spaceship → Inspector Panel → Add Component → Type "Box Collider 2D").

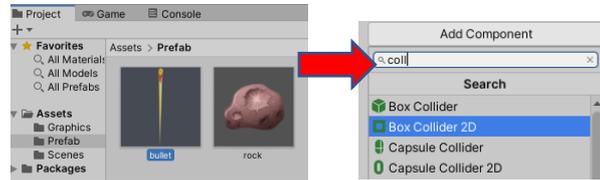


DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 3 : Level Design (Part 1)

Next : Practical Task 2

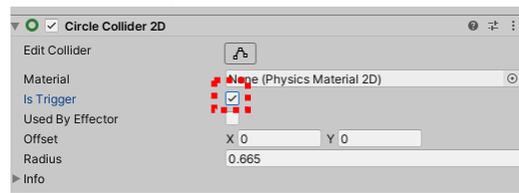
- c. Add **BOX Collider 2D** to the **BULLET** from Prefab folder (Select BULLET → Inspector Panel → Add Component → Type “Box Collider 2D”).



- d. Then, Add **CIRCLE Collider 2D** to the **ROCK** from Prefab folder (Select BULLET → Inspector Panel → Add Component → Type “Circle Collider 2D”).



- e. Then select the **ROCK** prefab and check the "Is Trigger" option in the box collider component.



- f. Lastly, modify your “rockScript” and add these line to it.

```
0 references
27 void OnTriggerEnter2D(Collider2D other)
28 {
29     // Name of the object that collided with the enemy
30     var name = other.gameObject.name;
31
32     // If the enemy collided with a bullet
33     if (name == "bullet(Clone)")
34     {
35         // Destroy itself (the enemy) and the bullet
36         Destroy(gameObject);
37         Destroy(other.gameObject);
38     }
39
40     // If the enemy collided with the spaceship
41     if (name == "space_ship")
42     {
43         // Destroy itself (the enemy) to keep things simple
44         Destroy(gameObject);
45     }
46 }
47
48
```

- g. Done

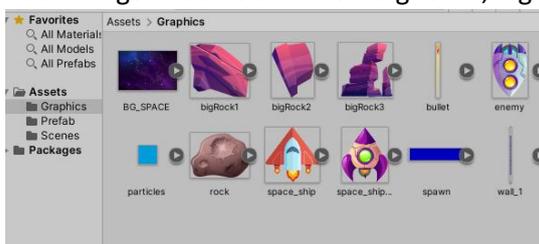
DEPARTMENT OF INFORMATION TECHNOLOGY AND COMMUNICATION

Practical Task (s) No: 1 / 2 / 3 / 4

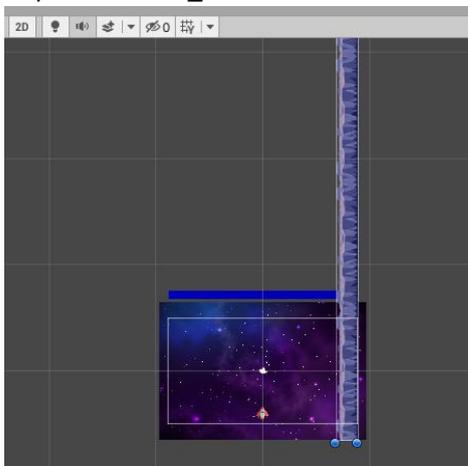
CLO 1: Display 2D assets, game interactivity, level design and scripting skills of game development tools to maintain coherency in game. (P4, PLO3)

Instruction: Manipulating Unity Assets to build your first game scene Assets and Prop

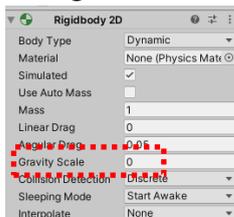
1. Login to your CIDOS account and download “Practical_Task_2_Resources”
2. Open your previous project
3. Double Click to perform installation into your current project
4. You will get 4 additional files: bigRock1, bigRock2, bigRock3 and wall_1



5. Drag wall_1 into hierarchy panel
6. Change value on “Order in Layer” to 4 (Select wall_1 on hierarchy → Inspector Panel → “Order in Layer” → 4)
7. Reposition wall_1 to show result as below



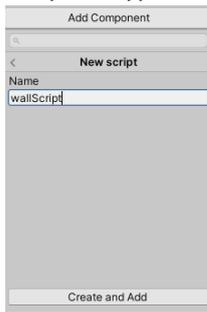
8. Select wall_1 and add BOX COLLIDER 2D into it (Add Component → Type Box Collider 2D)
9. Select on space_ship and add Rigidbody2D on it (Add Component → Type Rigidbody 2D)
10. Change the value to Gravity Scale to 0



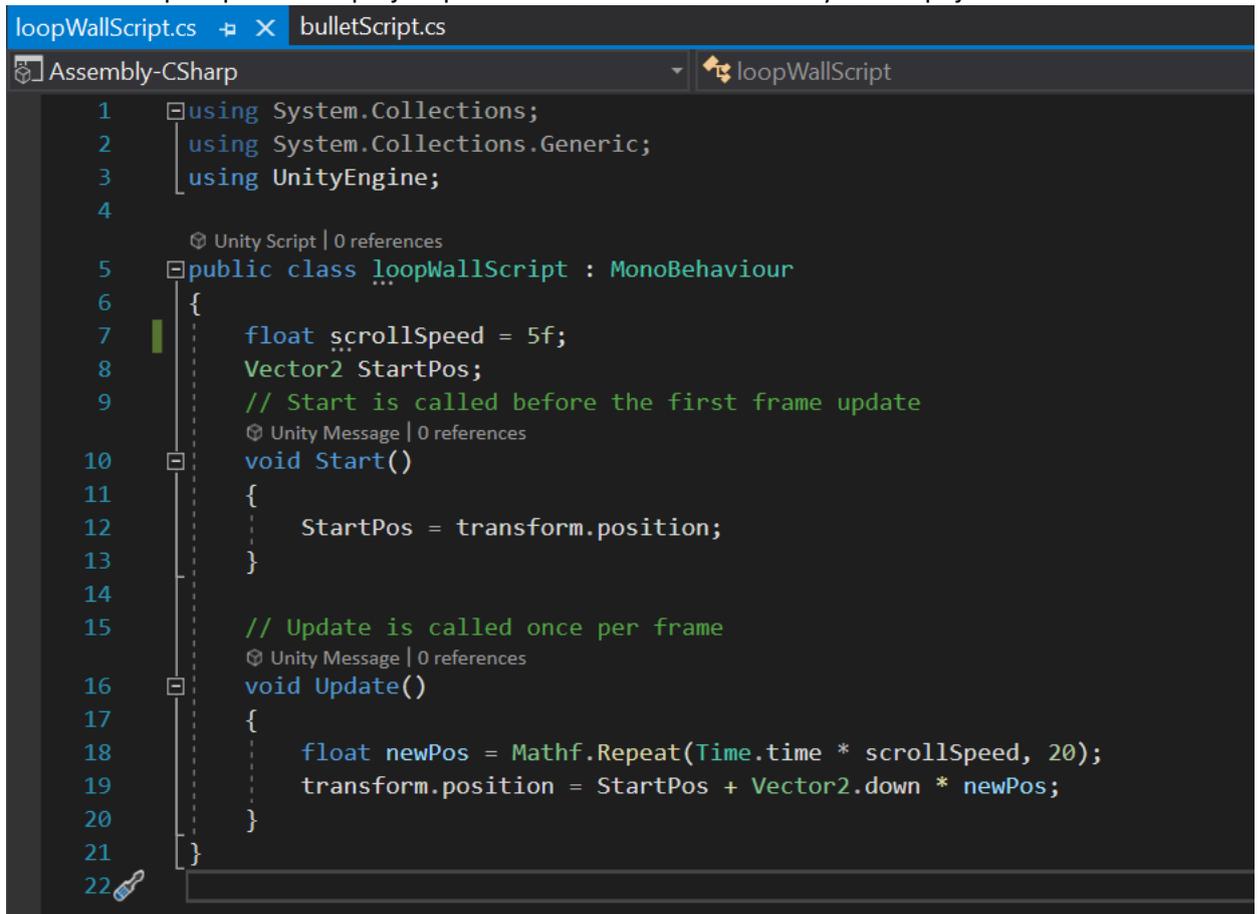
11. Do the same step to make the same wall at left side as shown below:



12. Then, select at the left wall, attach new script from inspector panel (Add Component → New Script → Type “loopWallScript” → Create and Add)



13. Find “loopScriptWall” at project panel and double click and edit your script just below:



```

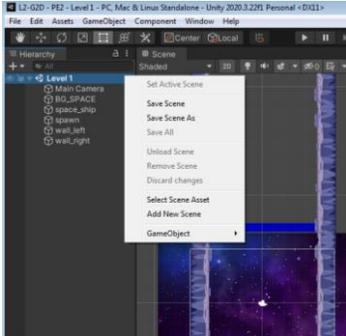
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  Unity Script | 0 references
6  public class loopWallScript : MonoBehaviour
7  {
8      float scrollSpeed = 5f;
9      Vector2 StartPos;
10     // Start is called before the first frame update
11     Unity Message | 0 references
12     void Start()
13     {
14         StartPos = transform.position;
15     }
16     // Update is called once per frame
17     Unity Message | 0 references
18     void Update()
19     {
20         float newPos = Mathf.Repeat(Time.time * scrollSpeed, 20);
21         transform.position = StartPos + Vector2.down * newPos;
22     }

```

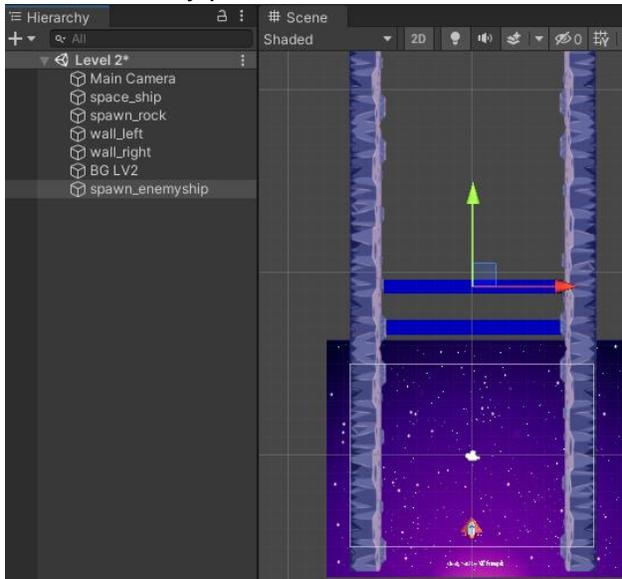
14. Attach “loopWallScript” into right wall as well (Click on the right wall object at hierarchy tab → drag loopWallScript at Assets folder → drop on to Add Component button at Inspector panel)
15. Done

DFG40323 2D Game Development Practical Exercise 2

1. Right click at Level 1 and click 'Save Scene As', name as Level 2.

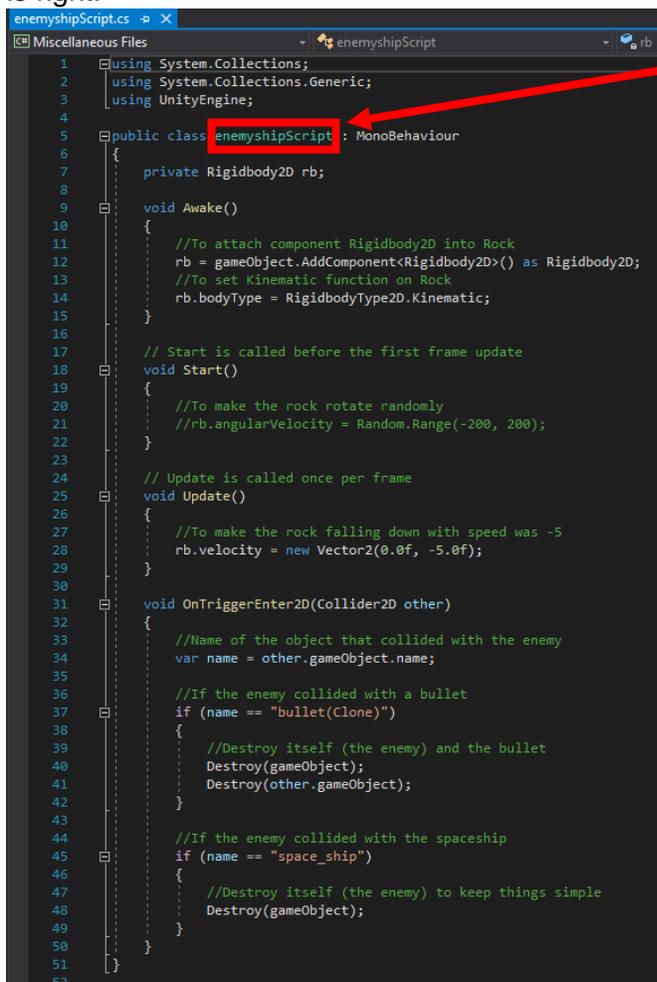


2. Drag new space background to folder Graphics & then drag to Hierarchy panel.
3. Delete the previous background by right click at it & delete.
4. Rearrange the new background using move tool (W) and rect tool (T).
5. Right click spawn at Hierarchy panel, and rename as spawn_rock.
6. Right click spawn at Hierarchy panel, and click Dupliate.
7. Rename the new spawn as spawn_enemyship.
8. Click spawn_enemyship at Hierarchy panel, by using move tool, move it above spawn_rock.
9. Your Hierarchy panel & Scene tab will be like this:



10. Go to folder Graphics, drag enemy ship  to Hierarchy panel. Then drag it back to folder Prefab.
11. Click enemyship at folder Prefab → at the Inspector panel, click Add Component at the bottom.
12. Type New Script → name as enemyshipScript → click 'Create and Add' button.

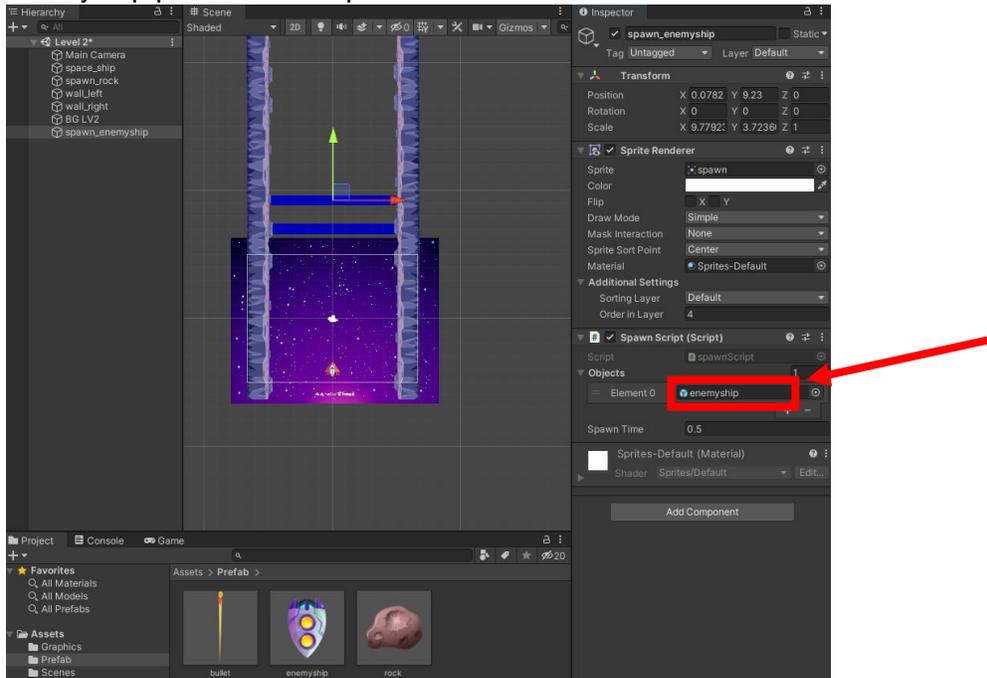
13. Open enemyshipScript, and copy the code from rockScript. make sure the gameobject name is right:



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class enemyshipScript : MonoBehaviour
6 {
7     private Rigidbody2D rb;
8
9     void Awake()
10    {
11        //To attach component Rigidbody2D into Rock
12        rb = gameObject.AddComponent<Rigidbody2D>() as Rigidbody2D;
13        //To set Kinematic function on Rock
14        rb.bodyType = RigidbodyType2D.Kinematic;
15    }
16
17    // Start is called before the first frame update
18    void Start()
19    {
20        //To make the rock rotate randomly
21        //rb.angularVelocity = Random.Range(-200, 200);
22    }
23
24    // Update is called once per frame
25    void Update()
26    {
27        //To make the rock falling down with speed was -5
28        rb.velocity = new Vector2(0.0f, -5.0f);
29    }
30
31    void OnTriggerEnter2D(Collider2D other)
32    {
33        //Name of the object that collided with the enemy
34        var name = other.gameObject.name;
35
36        //If the enemy collided with a bullet
37        if (name == "bullet(Clone)")
38        {
39            //Destroy itself (the enemy) and the bullet
40            Destroy(gameObject);
41            Destroy(other.gameObject);
42        }
43
44        //If the enemy collided with the spaceship
45        if (name == "space_ship")
46        {
47            //Destroy itself (the enemy) to keep things simple
48            Destroy(gameObject);
49        }
50    }
51 }
52
```

14. The enemyship will not rotate randomly, so put the comment on the code by using `//` in front of the line.

15. Click spawn_enemyship at Hierarchy panel and click enemyship enemyship prefab. Drag enemyship prefab and drop at Element 0.



16. Set spawn time spawan_enemyship to 3 and spawn_rock to 0.5.
17. Your game tab will be like this:



DEPARTMENT OF INFORMATION TECHNOLOGY AND COMMUNICATION

Practical Exercise (s) No: 1 / 2 / 3 / 4

MARKS: /20

Name: _____

Registration No.: _____

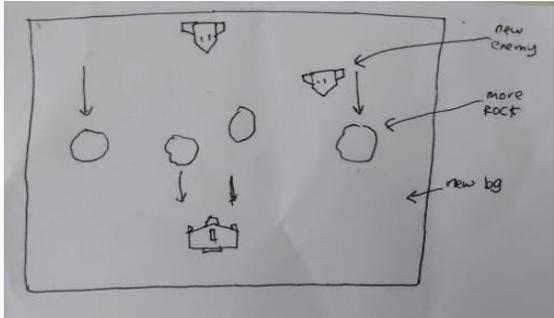
Code & Course: DFG40143 - 2D GAME DEVELOPMENT Class: _____

CLO 1: Display 2D assets, game interactivity, level design and scripting skills of game development tools to maintain coherency in game. (P4, PLO3)

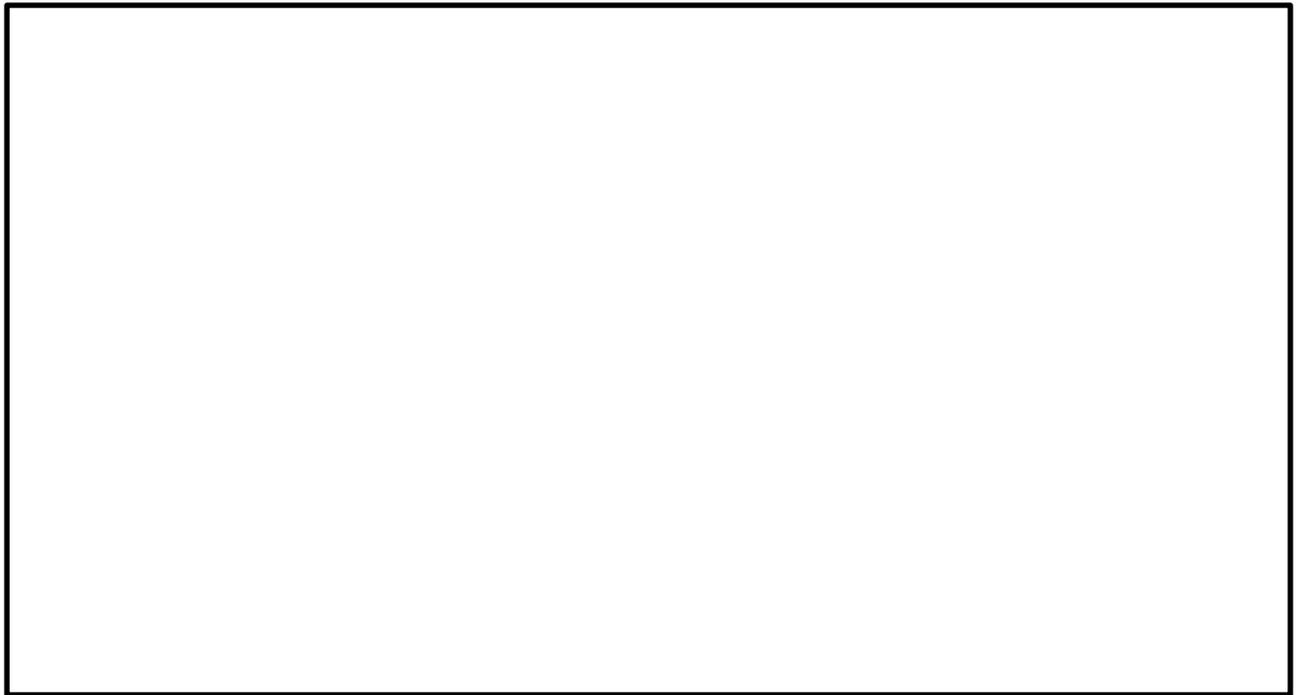
A) CREATE NEW LEVEL SCENE

Question:

Your company are working on action 2D game project. Your supervisor gives a task for you to develop new level scene (Level 2) for your current game project. **DUPLICATE** your LEVEL 1 scene (**Right Click on Scene → Duplicate → Rename as "Level 2"**), and modify your Level 1 scene to meet the Level 2 requirement shown as below:

<u>Level 1 Requirement</u>	<u>Level 2 Requirement</u>
 <ul style="list-style-type: none">- Rock coming infrequently- Player can shoot rock	 <ul style="list-style-type: none">- Rock coming more frequently- New background- New enemy coming (use new spaceship) but infrequently

Paste your Level 2 Scene which you develop below:



Topic 4.0 : Game Interactivity

DFG40323 - 2D GAME DEVELOPMENT

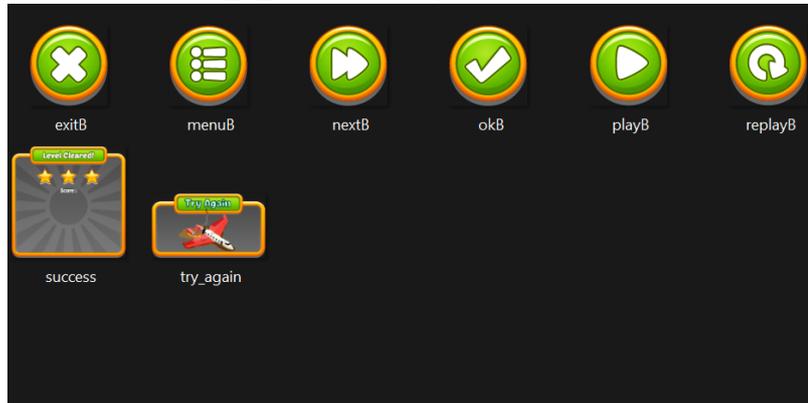
Now : LABSHEET 4 : Interactivity (Part 1)

Next : Practical Task 3 : Interactivity (Part 2)

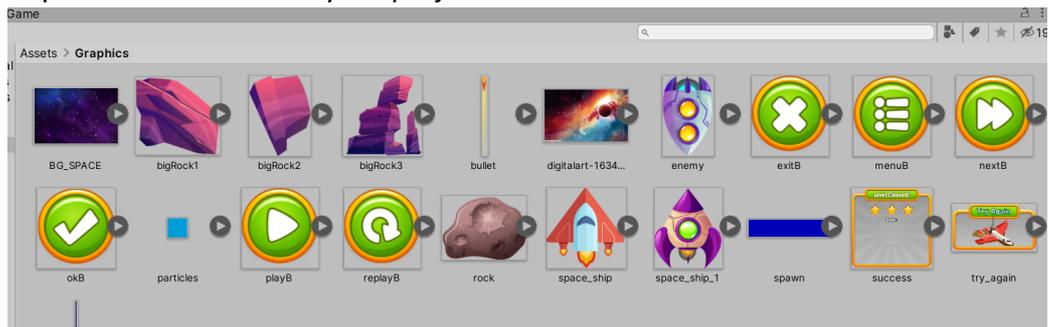
A) Creating Pop-up Panel

1. Importing new graphic assets.

- Download Labsheet_4_Resources from CIDOS/ google classroom.
- Extract Labsheet_4_Resources.zip.



- Open your previous project (Practical Exercise 2).
- Import the content into your project Asset.



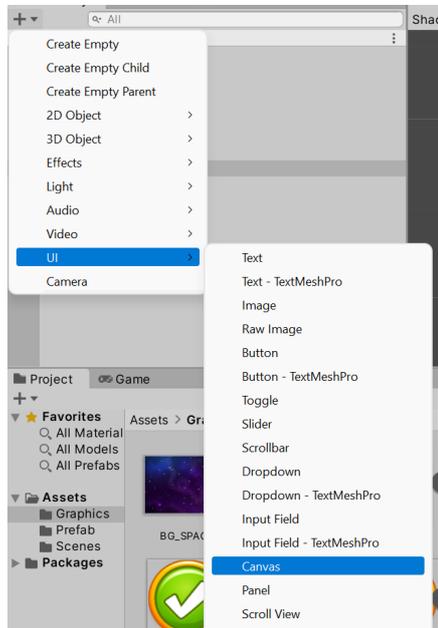
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 4 : Interactivity (Part 1)

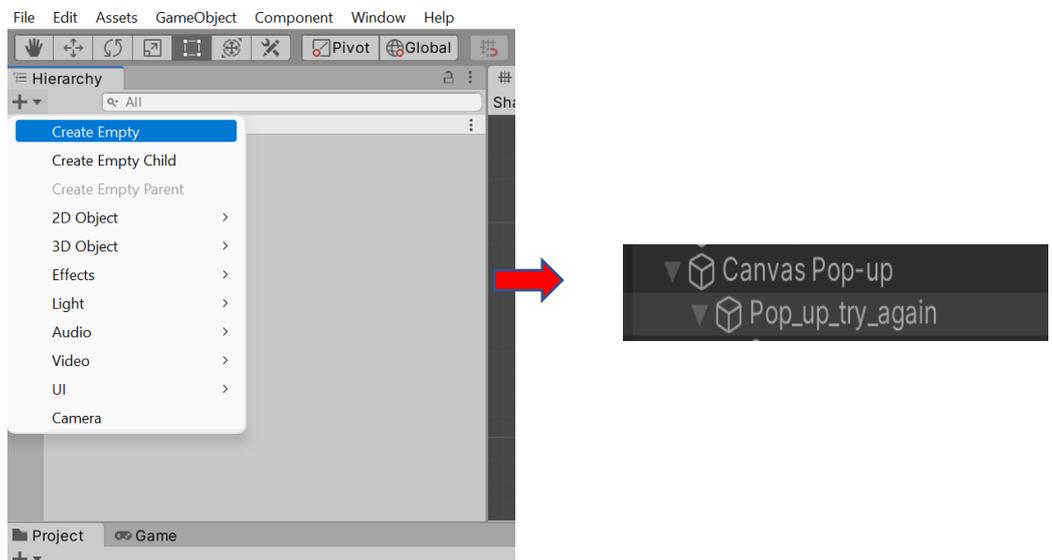
Next : Practical Task 3 : Interactivity (Part 2)

2. Preparing pop-up

- a. Create new Canvas and name it as “Canvas Pop-up”; (Hierarchy → (+) → UI → Canvas → type “Canvas Pop-up”).



- b. Next, create new emptyClip inside “Canvas Pop-up” (Select Canvas Pop-up → (+) → Create Empty → type “Pop_up_try_again”).

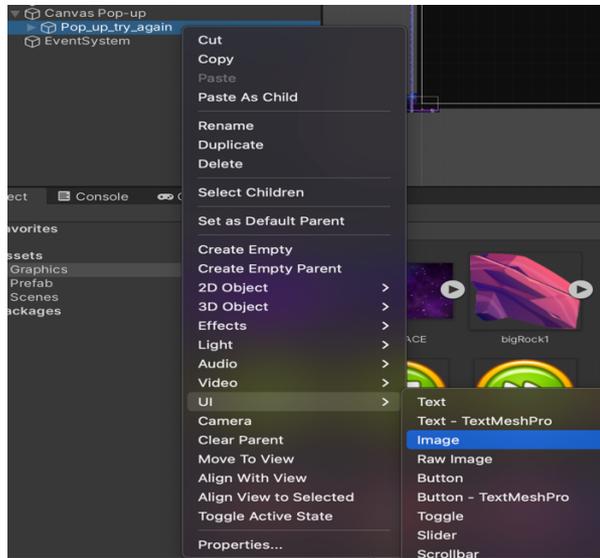


DFG40323 - 2D GAME DEVELOPMENT

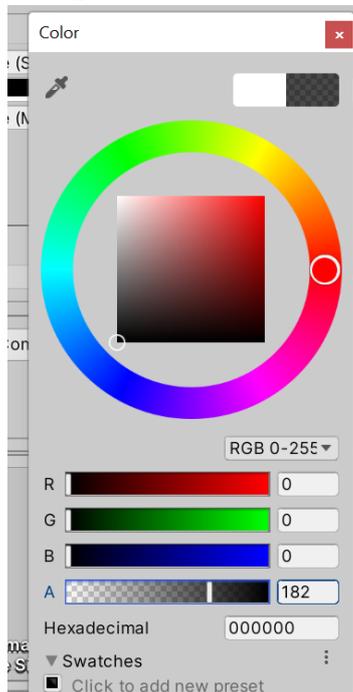
Now : LABSHEET 4 : Interactivity (Part 1)

Next : Practical Task 3 : Interactivity (Part 2)

- c. Then, create new Image inside “Pop_up_try_again”;
(Select Pop_up_try_again → Right Click → UI → Image → Name it as “Black_Background”)



- d. Change the colour setting as below (Inspector Panel → Color → Change value).

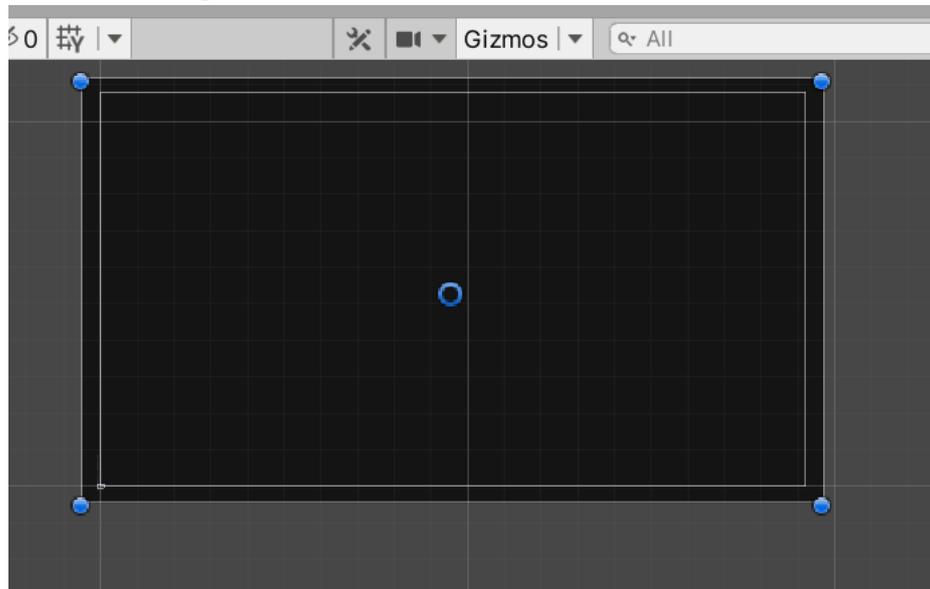


DFG40323 - 2D GAME DEVELOPMENT

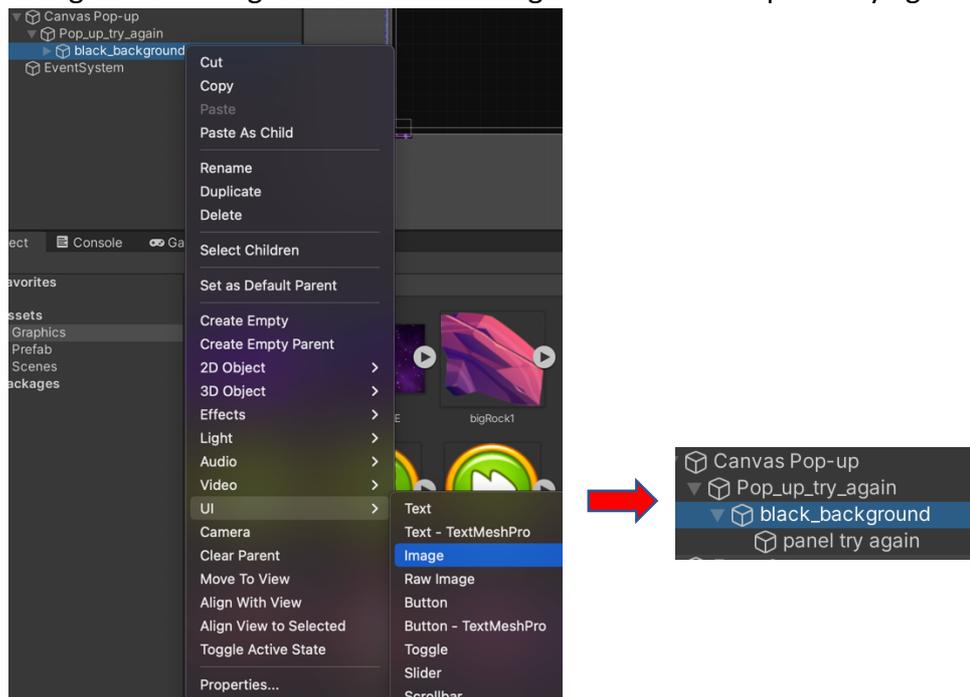
Now : LABSHEET 4 : Interactivity (Part 1)

Next : Practical Task 3 : Interactivity (Part 2)

- e. Resize the Image to cover all the canvas.



- f. Next, insert new Image inside “black_background” (Select “black_background” → Right Click → UI → Image → Name is as “panel try again”).

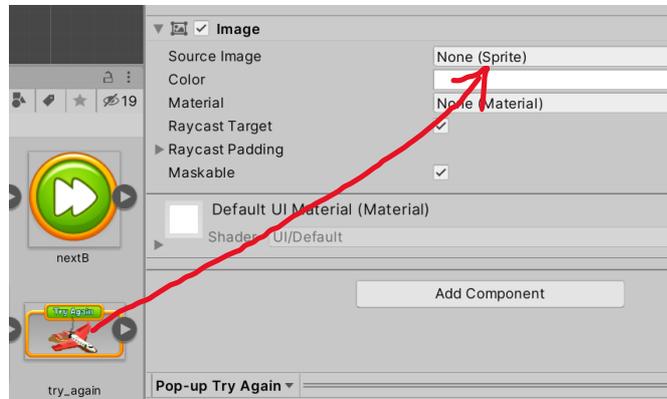


DFG40323 - 2D GAME DEVELOPMENT

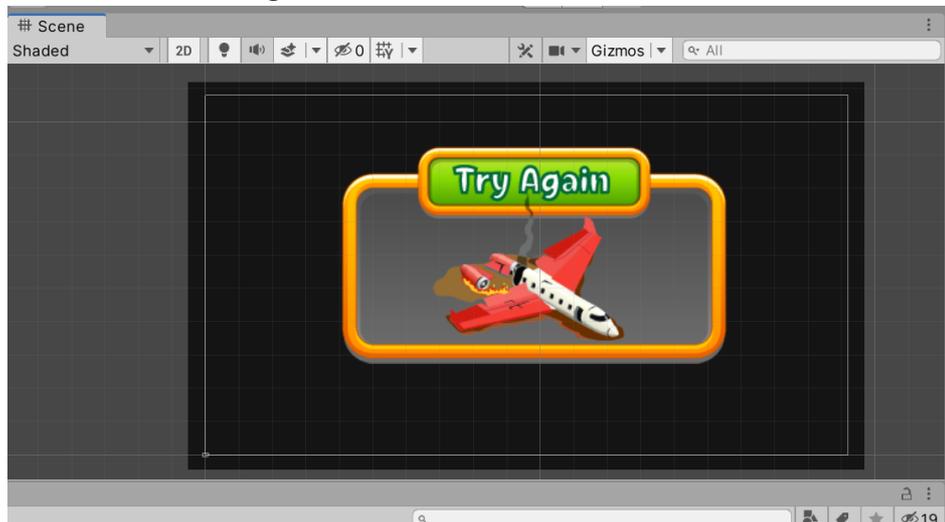
Now : LABSHEET 4 : Interactivity (Part 1)

Next : Practical Task 3 : Interactivity (Part 2)

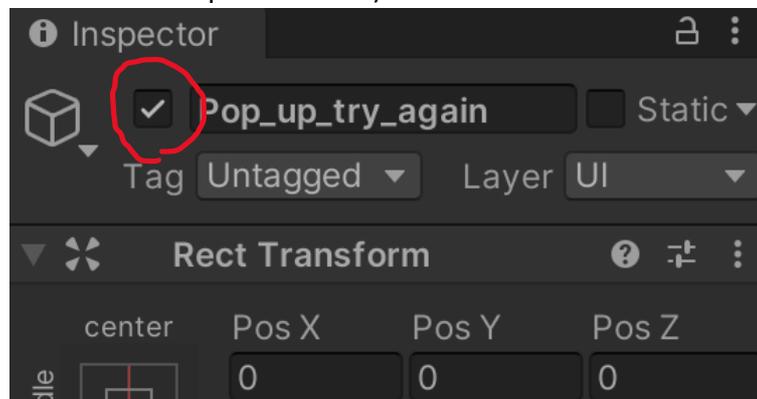
- g. Change the image source for “panel try again” to “try_again” from Asset folder (Drag “try_again” into “Source Image”).



- h. Now, resize the image as below:



- i. Next, disable “Pop_up_try_again” from Inspector Panel (Untick radio button from Inspector Panel).



DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 4 : Interactivity (Part 1)

Next : Practical Task 3 : Interactivity (Part 2)

3. Creating interactivity

- Open "spaceshipScript" to modify.
- Declare new GameObject variable called "Pop_up_try_again".

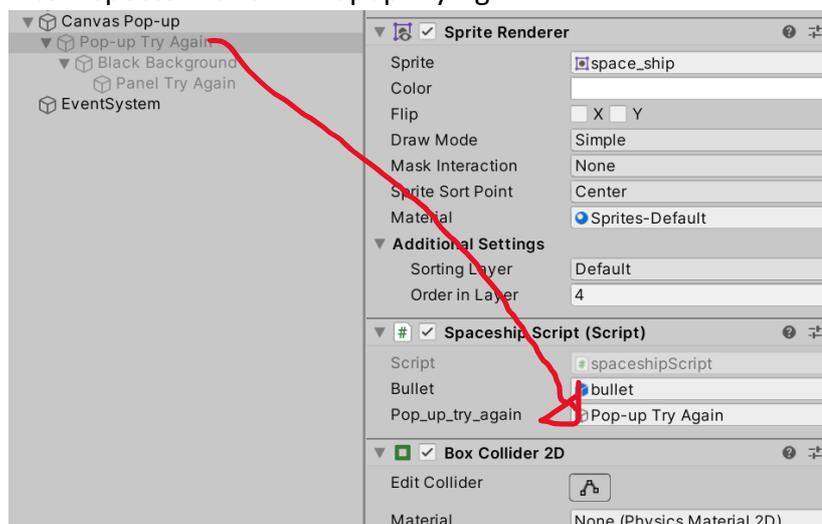
```
Unity Script | 0 references
public class spaceshipScript : MonoBehaviour
{
    public GameObject bullet;
    public GameObject Pop_up_try_again;
    // Start is called before the first frame update
    void Start()
}
```

- Add new script below at line 31 (or AFTER method update()).

```
Unity Message | 0 references
void OnTriggerEnter2D(Collider2D other)
{
    var name = other.gameObject.name;
    Debug.Log(name);
    if (name == "rock(Clone)")
    {
        // Destroy itself (the enemy) to keep things simple
        Destroy(gameObject);

        Pop_up_try_again.SetActive(true);
        Time.timeScale=0;
    }
}
```

- Select "SpaceShip" at hierarchy panel → drag "Pop-up Try Again" inside → into inspector Panel → Popup Try Again.



- Save your script and Play.

DEPARTMENT OF INFORMATION TECHNOLOGY AND COMMUNICATION

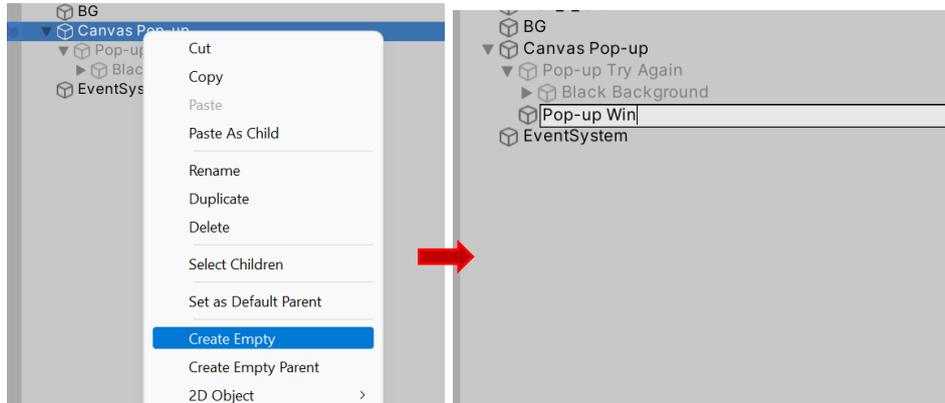
Practical Task (s) No: 1 / 2 / 3 / 4

CLO 1: Display 2D assets, game interactivity, level design and scripting skills of game development tools to maintain coherency in game. (P4, PLO3)

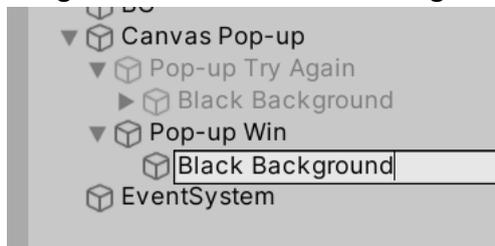
A) Navigatiion Button

1. Setting up Win Pop Up

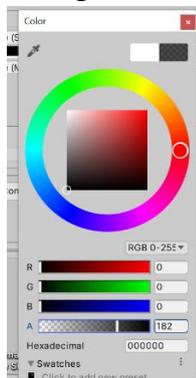
- a. Navigate to the Canvas Pop Up and create new Empty inside it (Select Canvas Pop up → Right Click → New Empty → Name it as “ Pop up Win”



- b. Next, insert new image inside “Pop-up Win” (Select “Pop-up Win” → UI → Image → Name it as “Black Background”

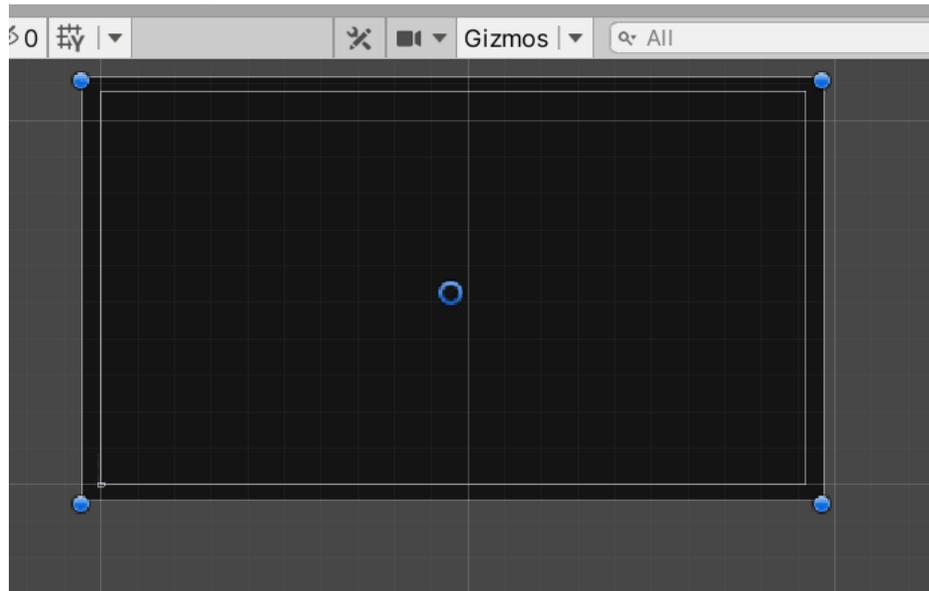


- c. Change the color setting as below (Inspector Panel → Color → Change value) :

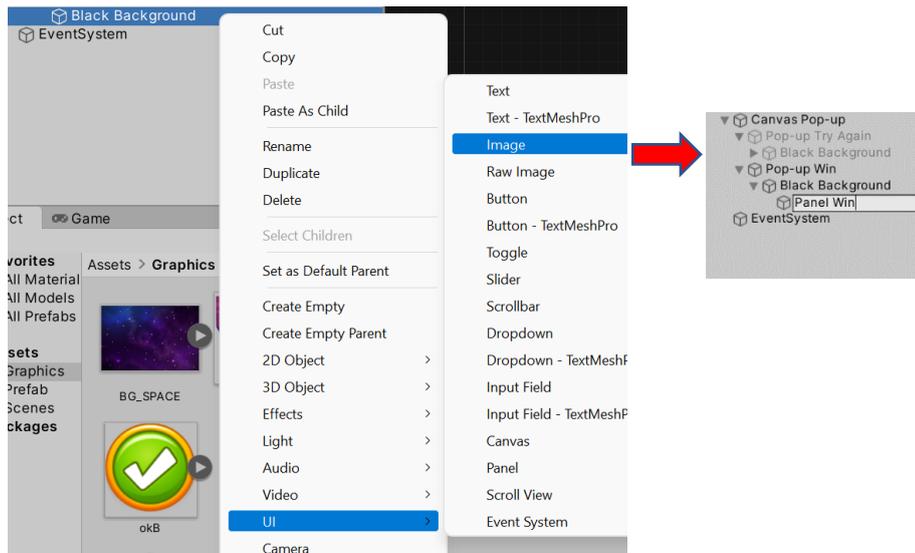


Color : Black
A : 182

d. Resize the Image to cover all the canvas

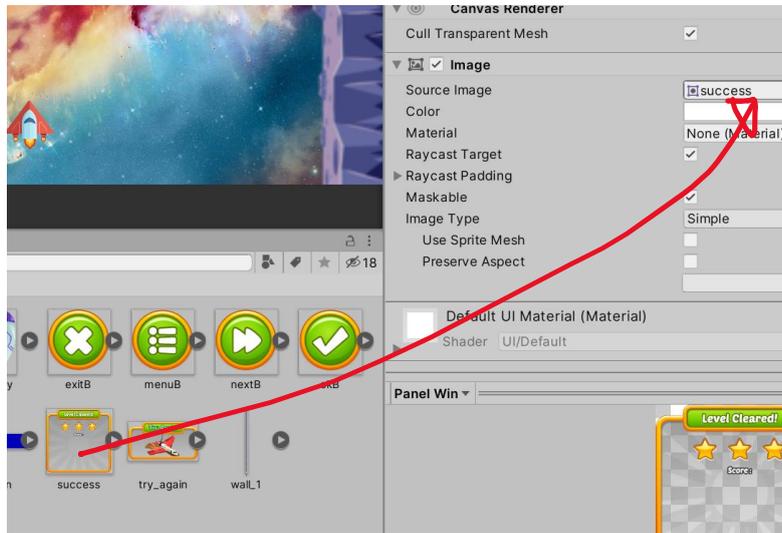


e. Next, insert new Image inside “Black Background”, (Select “Black Background” → Right Click → UI → Image → Name is as “Panel Win”



Now : Practical Task 3

- f. Change the image source for “Panel Win” to “success” from Asset Folder (Drag “success” into Source Image)

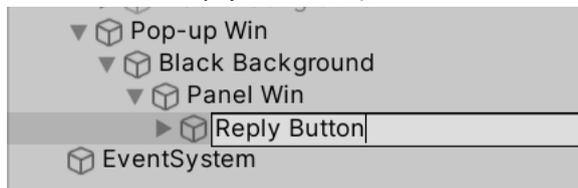


- g. Now, resize the image as below :

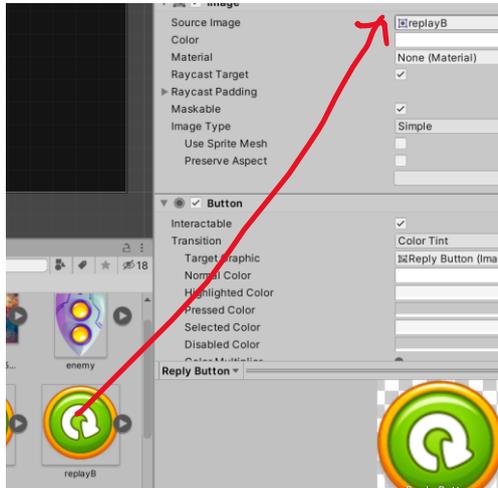


2. Creating Navigation Button

- a. Insert new UI Button inside “Panel Win” (Select “Panel Win” → UI → Button → Name is as “Reply Button”)



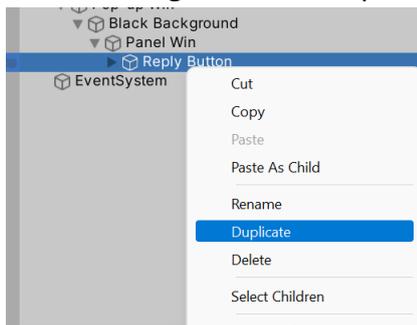
- b. Change the image source for “Reply Button” to “ReplyB” from Asset Folder (Drag “ReplyB” into Source Image)



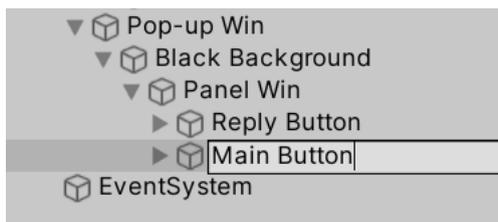
- c. Resize and reposition your button as below :



- d. Navigate to Hierarchy Panel and select the button and duplicate it.(Select Reply Button → Right Click → Duplicate)



- e. Rename the button as “Main Button”

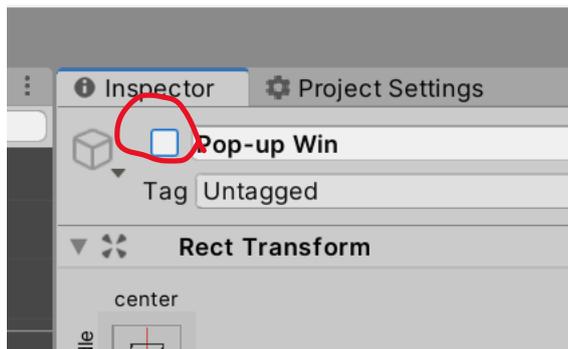


f. Change the image source for “Main Button” to “Menu B” from Asset Folder (Drag “Menu B” into Source Image)

g. Resize and reposition your button as below :



h. Select “Pop Win” from hierarchy panel and disable (UNTICK) it at Inspector Panel



i. Save your project

Name: _____

Registration Num.: _____

Code & Course: DFG40323 - 2D GAME DEVELOPMENT

Programme/Class: _____

Practical Task (s) No: 1 / 2 / 3 / 4

MARKS: /20

CLO 1: Display 2D assets, game interactivity, level design and scripting skills of game development tools to maintain coherency in game. (P4, PLO3)

1. Insert same buttons for “Panel Try Again” , the buttons are
 - a. Menu Button
 - b. Reply Button

2. Paste your final output for “Panel Try Again” below:

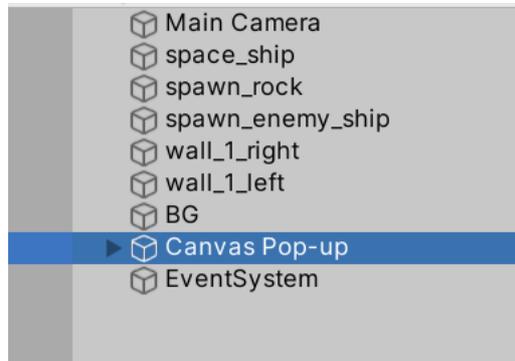
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 4 : Interactivity (part 3)

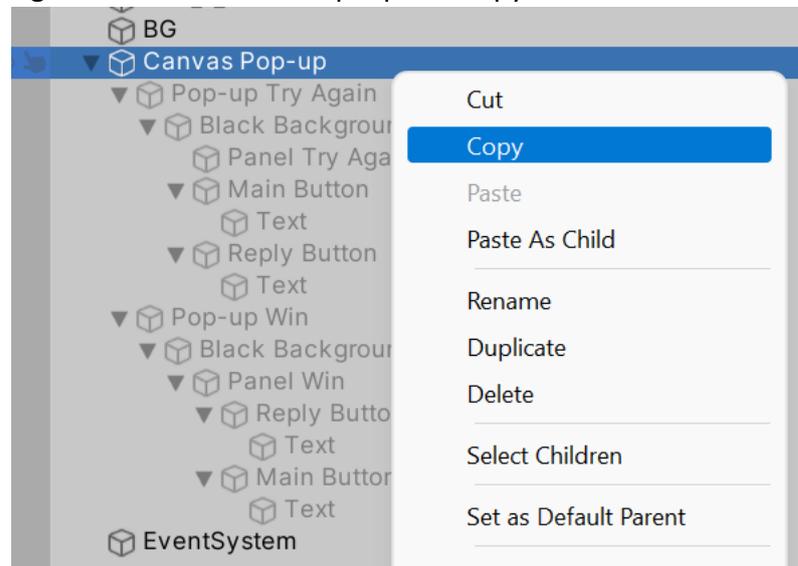
Next : LABSHEET 4 : Interactivity (part 4)

A) Duplicating Game Object across Scene

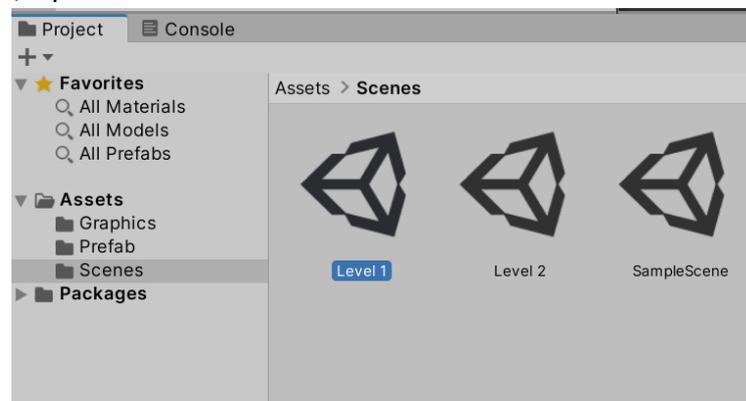
- Open your previous project.
- Open scene **Level 2** and Select “Canvas Pop-Up”



- Right Click on “Canvas Pop-Up” → Copy



- Next, Open Scene “Level 1” from Assets Folder

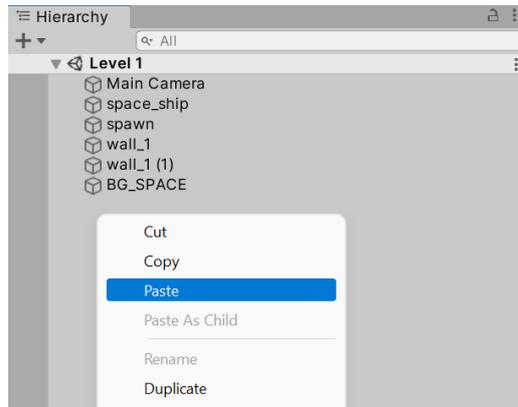


- Then, navigate to Hierarchy panel → Right Click → Paste

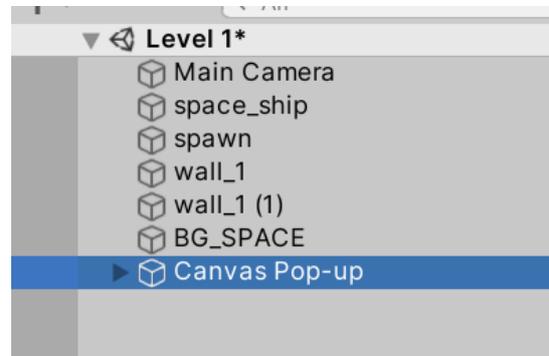
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 4 : Interactivity (part 3)

Next : LABSHEET 4 : Interactivity (part 4)



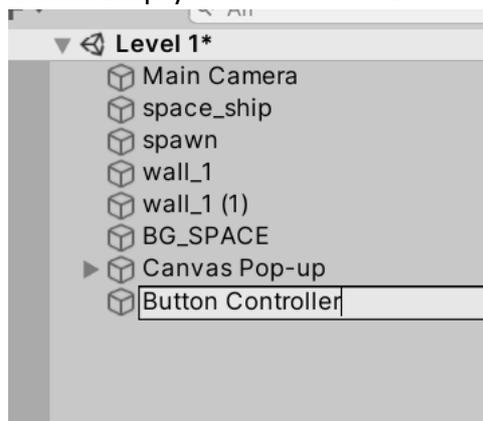
- f. You will have same “Canvas Pop-Up” in Scene 1 Hierarchy Panel



- g. Save your project

2. Scripting Navigate Button

- a. Create new Empty inside Hierarchy Panel (Hierarchy Panel → Right Click → Create Empty → Name it as “Button Controller”

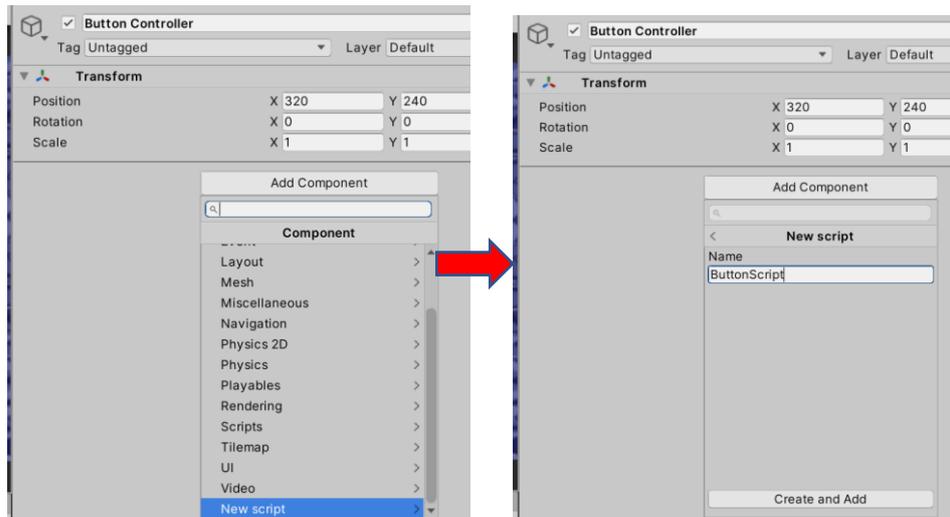


- b. Select “Button Controller” and add new Script on it (Button Controller → Inspector Panel → Add Component → New Script → Type “ButtonScript” → Create and Add)

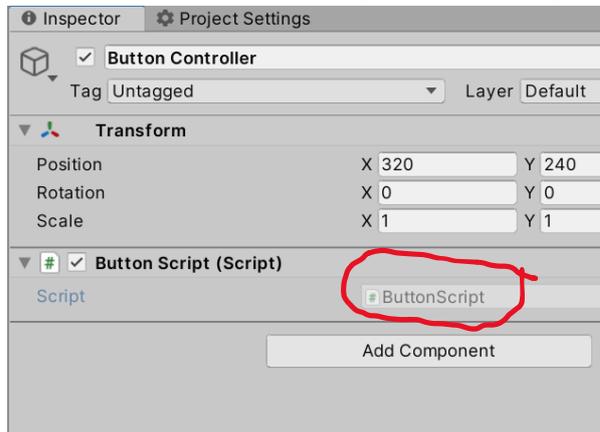
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 4 : Interactivity (part 3)

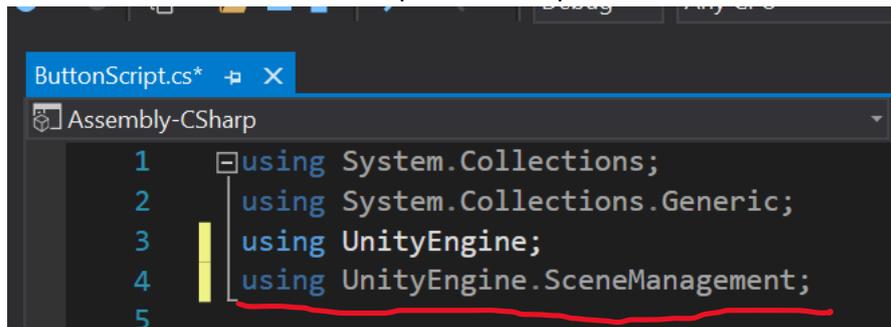
Next : LABSHEET 4 : Interactivity (part 4)



c. Double click on "ButtonScript" to edit script



d. Add new "NAMESPACE" at top line of script



DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 4 : Interactivity (part 3)

Next : LABSHEET 4 : Interactivity (part 4)

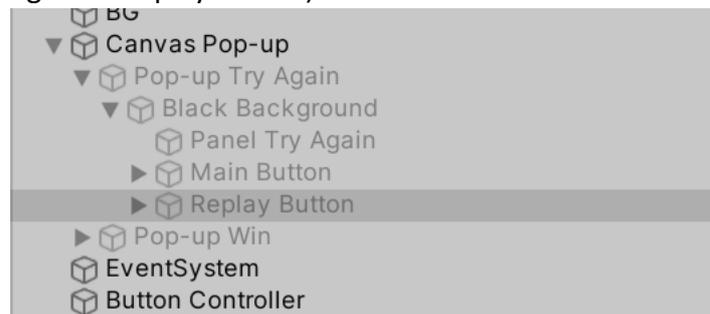
- e. Create new "METHOD" called ReplayMethod inside script (after method Update)

```
12
13 // Update is called once per fra
   Unity Message | 0 references
14 void Update()
15 {
16
17 }
   0 references
18 public void ReplayMethod()
19 {
20
21 }
22
23
```

- f. Type line below inside ReplayMethod()

```
18 }
   0 references
19 public void ReplayMethod()
20 {
21     SceneManager.LoadScene(0);
22     Time.timeScale = 1;
23 }
24
```

- g. Next, navigate to REPLAY button inside Pop Up Try Again, (Hierarchy Panel → Canvas Pop Up → Pop up Try Again → Black Background → Panel Try again → Replay Button)

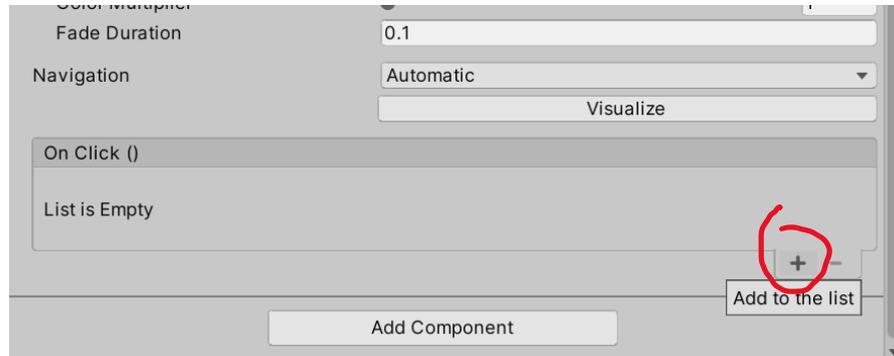


DFG40323 - 2D GAME DEVELOPMENT

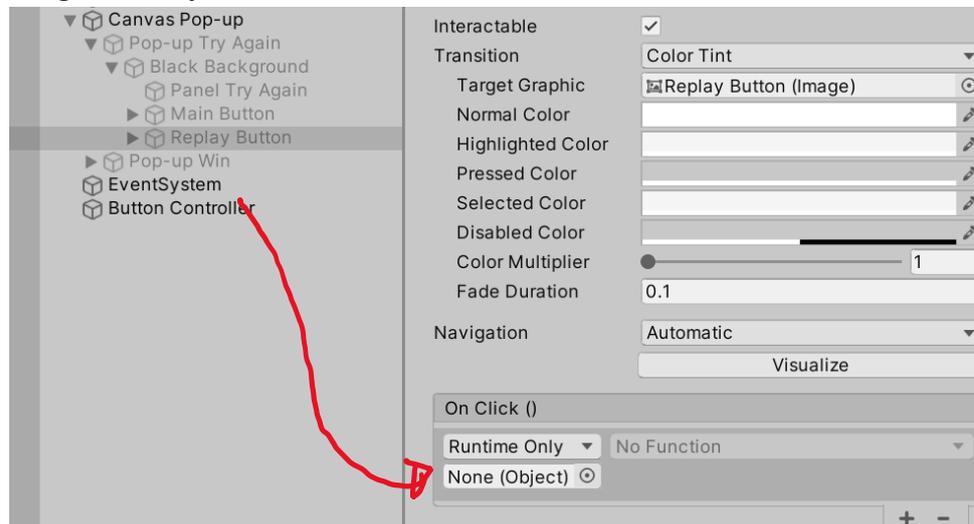
Now : LABSHEET 4 : Interactivity (part 3)

Next : LABSHEET 4 : Interactivity (part 4)

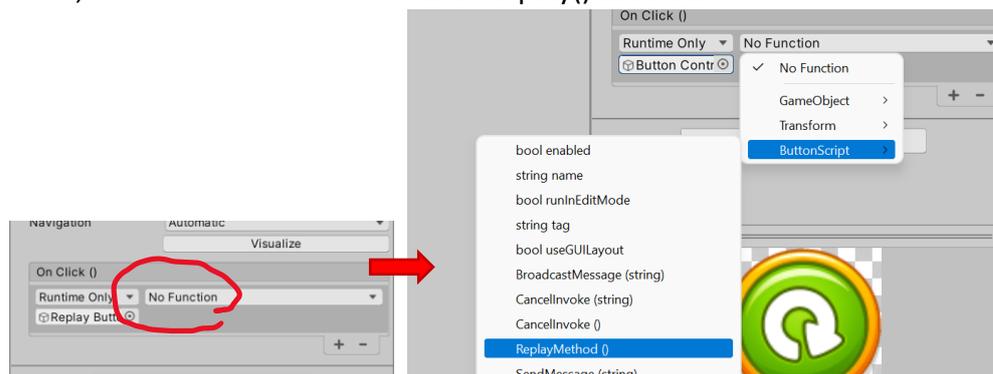
- h. Then, navigate to Inspector Panel and click (+) button inside OnClick() area



- i. Drag GameObject "Button Controller" into "NONE" area



- j. Then, click on "Function" and select Replay() method

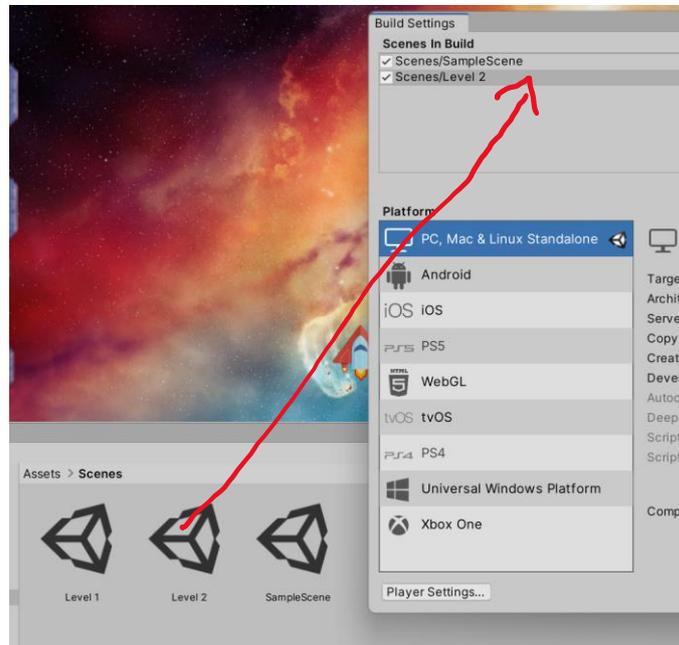


DFG40323 - 2D GAME DEVELOPMENT

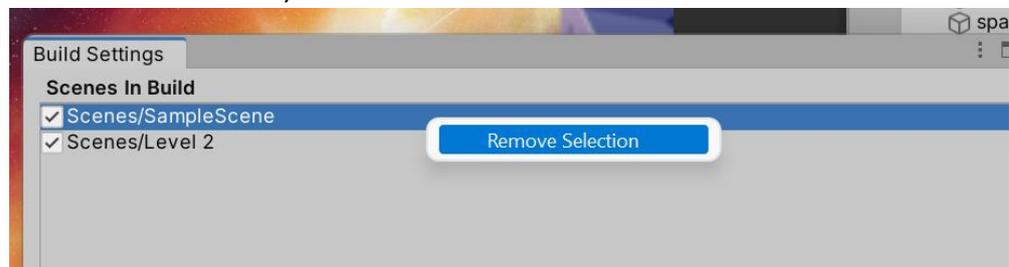
Now : LABSHEET 4 : Interactivity (part 3)

Next : LABSHEET 4 : Interactivity (part 4)

- k. Final step, navigate to Build Setting, (File → Build Setting), and drag Level 2 Scene inside “Scene in build” area.



- l. You can delete “SampleScene” which already inside that area. (Right Click → Remove Selection)



- m. Play and test your “Replay Button”

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 4 : Interactivity (part 3)

Next : LABSHEET 4 : Interactivity (part 4)

Name: _____

Registration No.: _____

Code & Course: DFG40143 2D GAME DEVELOPMENT

Programme/Class: _____

EXERCISE

1) Attach "Replay()" into Replay Button inside Level 1 scene

.....

2) Make sure Replay button on Level 1 scene is function

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 4 : Interactivity (part 4)

Next : LABSHEET 5 : Game Scripting (part 1)

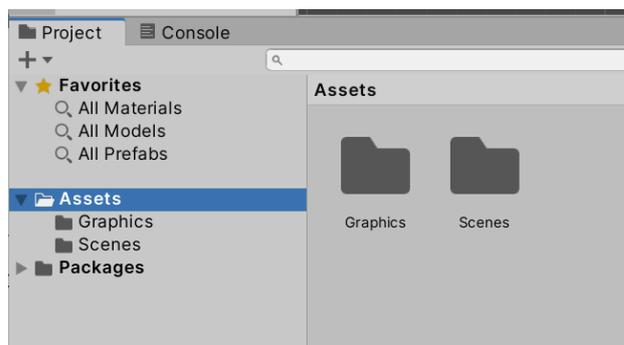
A) Combining 2 project into 1 master project

1. Exporting new package

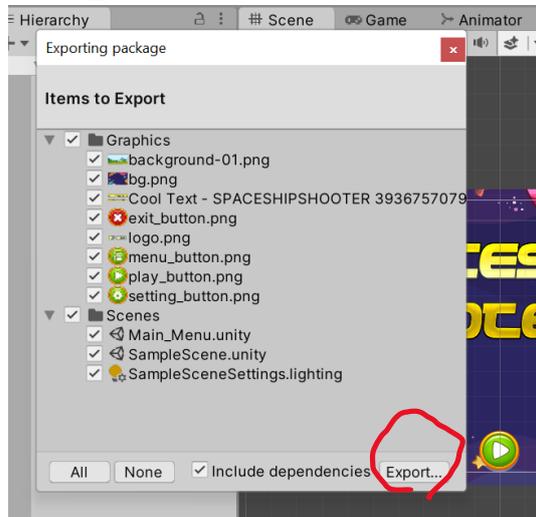
- a. Open your "Practical Exercise 1" project which is contain "Main Menu scene". Example below :



- b. Select at Asset Folder



- c. Navigate to Asset → Export Package; select "ALL" and click Export; Save as "Main_Menu" into any location in your pc



- d. Close your current project

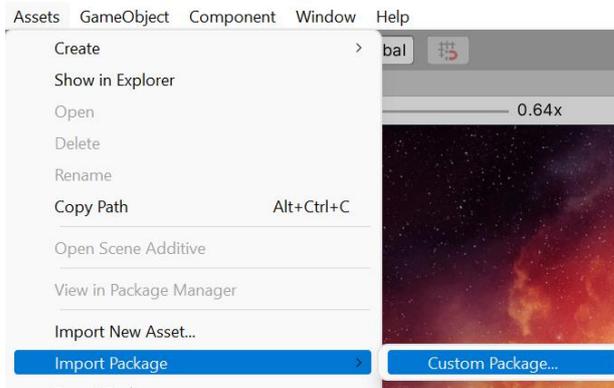
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 4 : Interactivity (part 4)

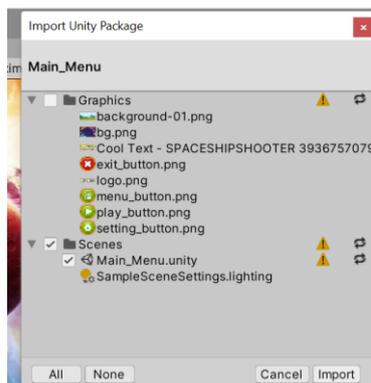
Next : LABSHEET 5 : Game Scripting (part 1)

2. Importing “Main_Menu” Package into new project

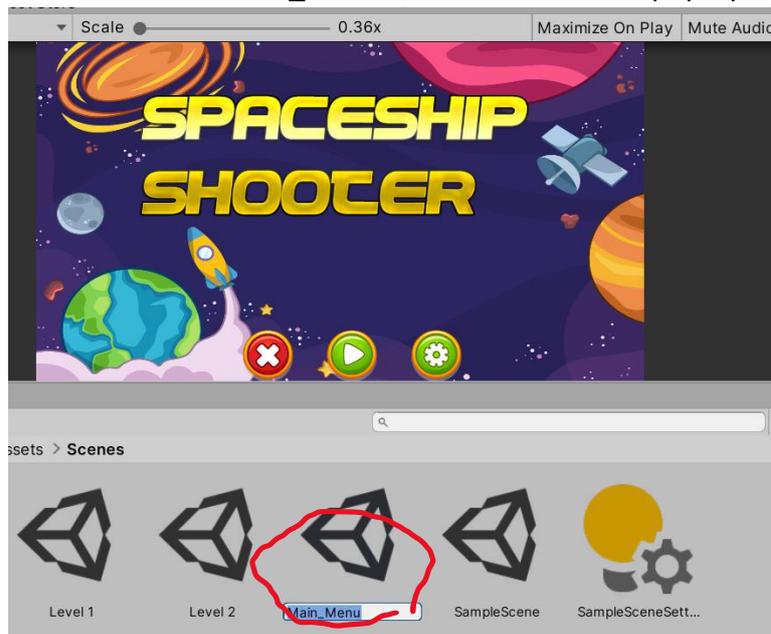
- Open your previous project (Labsheet 4 : Interactivity (part 3))
- Then, Navigate to (Assets → Import Package → Select “Main_Menu.package”)



- Click Import button to perform import



- Navigate to Scene folder, make sure which one is your main menu scene, and rename to “Main_Menu” ; Click “RELOAD” if pop-up appear.



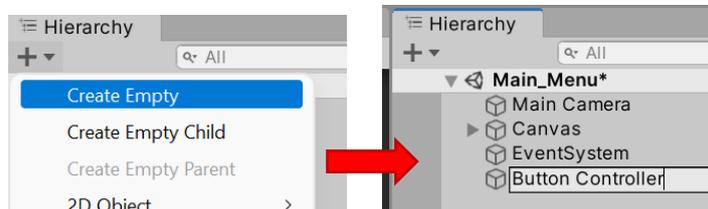
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 4 : Interactivity (part 4)

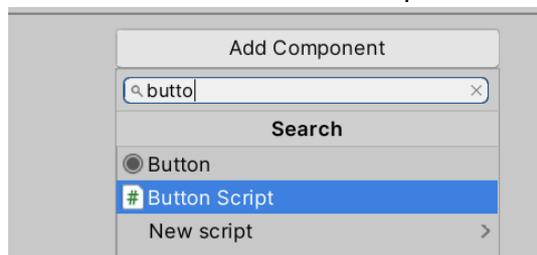
Next : LABSHEET 5 : Game Scripting (part 1)

3. Scripting Navigate Button

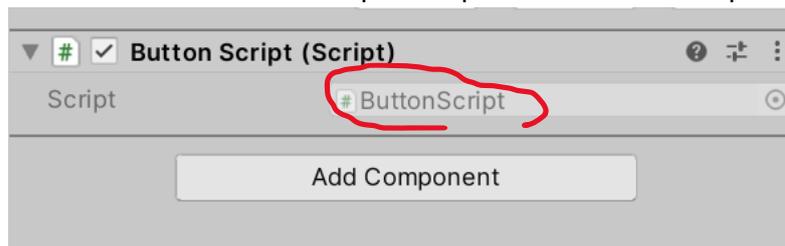
- a. Still in Main_Menu Scene, Create new Empty in hierarchy panel and rename as "Button Controller"



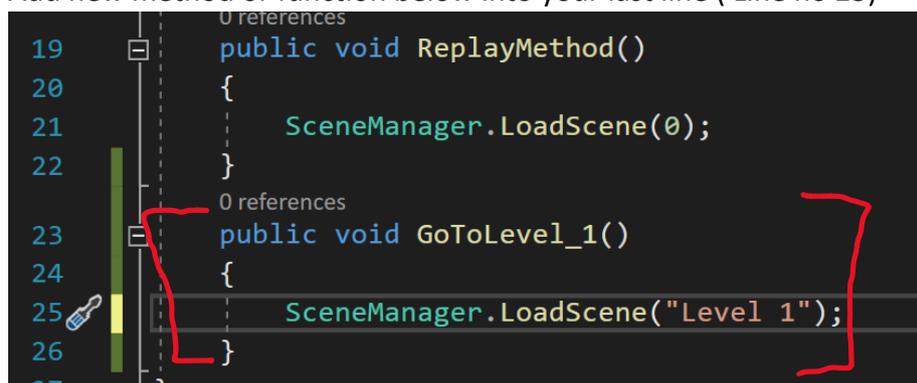
- b. Attach "ButtonScript" script into gameobject "Button Controller", (Select Button Controller → Add Component → Type "ButtonScript")



- c. Double Click at "ButtonScript" to open and edit the script



- d. Add new method or function below into your last line (Line no 23)



- e. Make sure "Main_Menu" is same name with your scene name inside Scene Folder

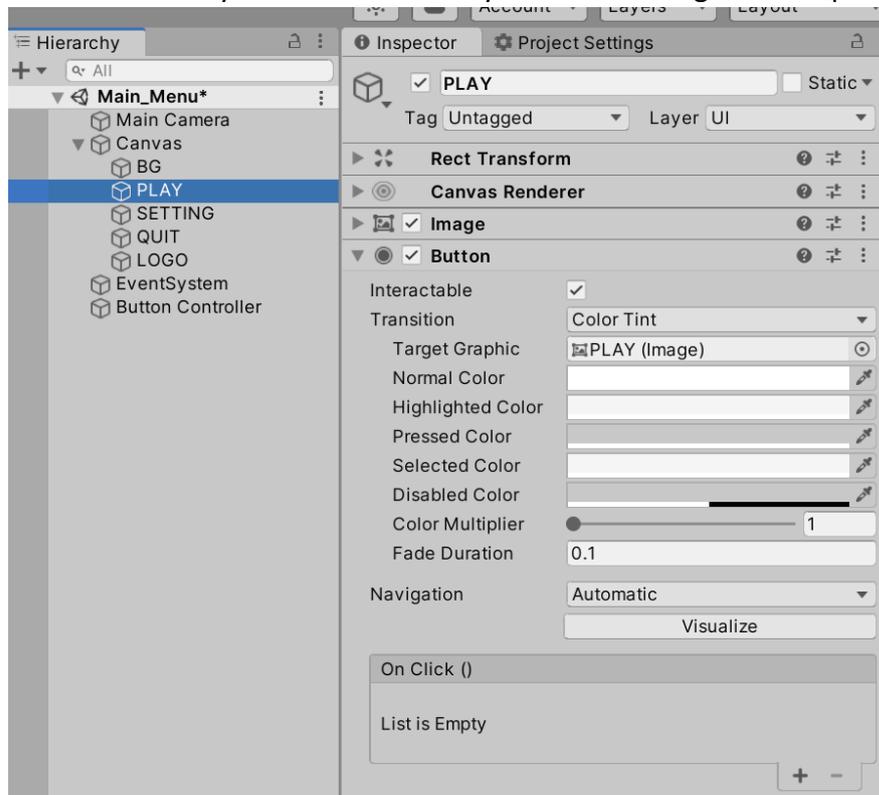


DFG40323 - 2D GAME DEVELOPMENT

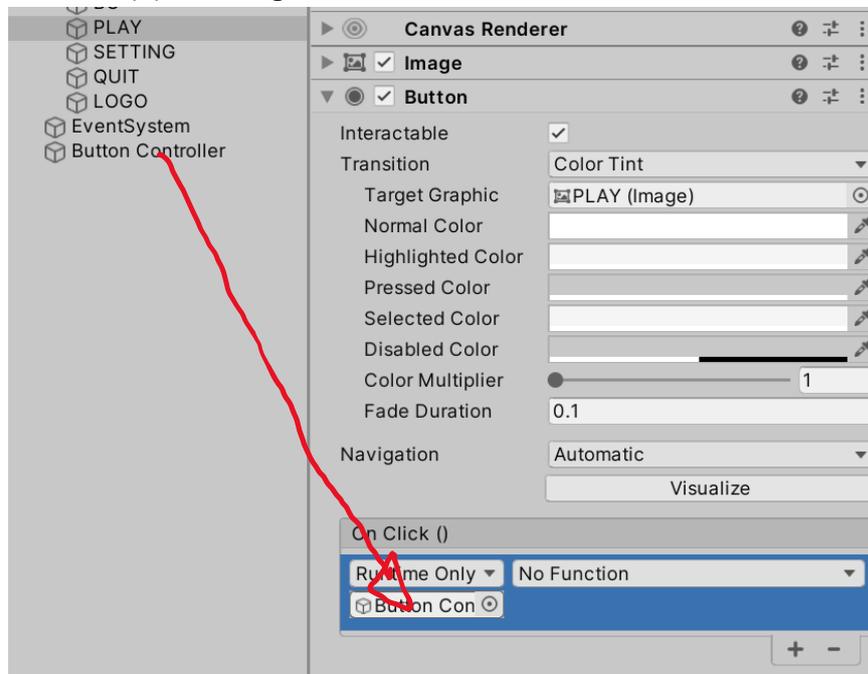
Now : LABSHEET 4 : Interactivity (part 4)

Next : LABSHEET 5 : Game Scripting (part 1)

- f. Select at the Play button at Hierarchy Panel and navigate to Inspector Panel



- g. Select at (+) and drag "Button Controller" into "NONE" Area

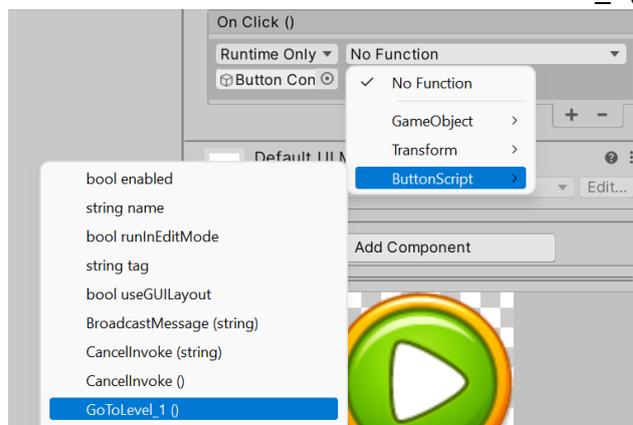


DFG40323 - 2D GAME DEVELOPMENT

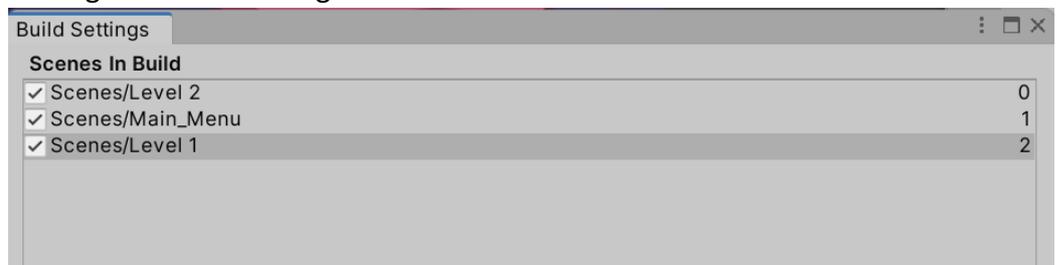
Now : LABSHEET 4 : Interactivity (part 4)

Next : LABSHEET 5 : Game Scripting (part 1)

- h. Select “No Function” and choose “GoToLevel_1()” method or function



- i. Add “Main_Menu” and “Level 1” Scene inside Build in Scene from Build Setting and Close setting.



- j. Save and test your project.

Topic 5.0 : Game Scripting

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 1)

Next : LABSHEET 5 : Game Scripting (part 2)

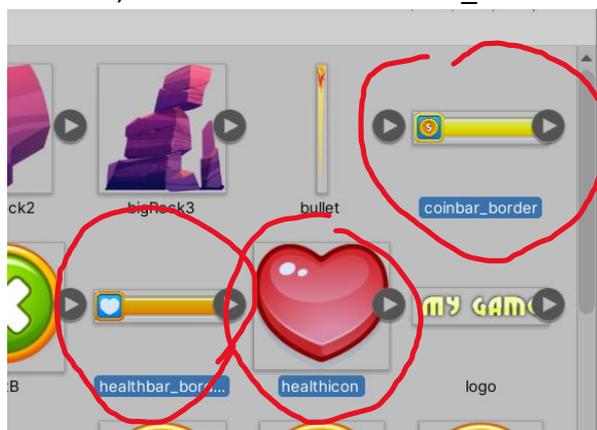
A) Setting up Health bar

1. Importing health bar graphic

- Open your previous project (Labsheet 4 Part 4)
- Download and Import Labsheet 5 part 1 Resources from CIDOS into Asset/Graphics

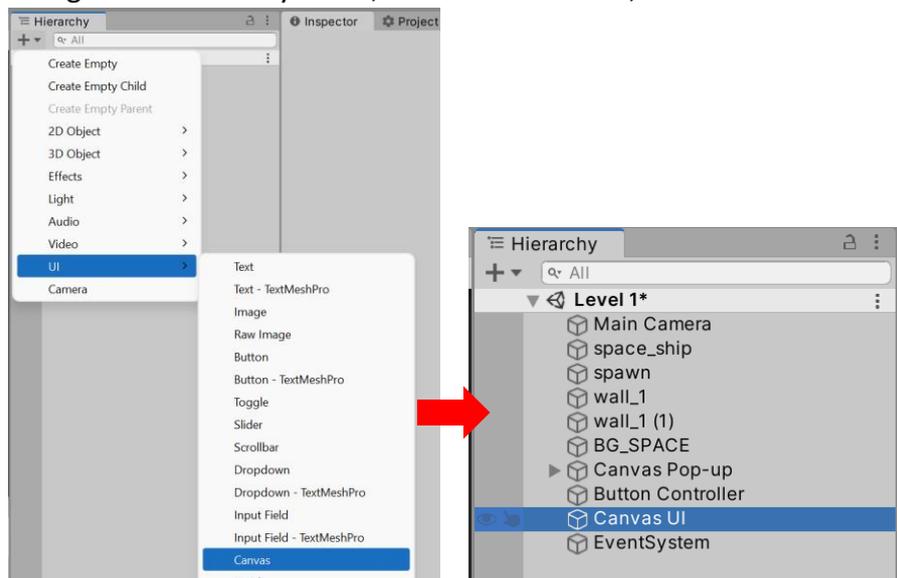


- Navigate to Asset → Graphics; make sure you have 3 graphics which are coinicon, healthicon and healthbar_border as below :



2. Creating health bar inside Hierarchy Panel

- Navigate to Hierarchy Panel, create new canvas, and name it as "Canvas UI"

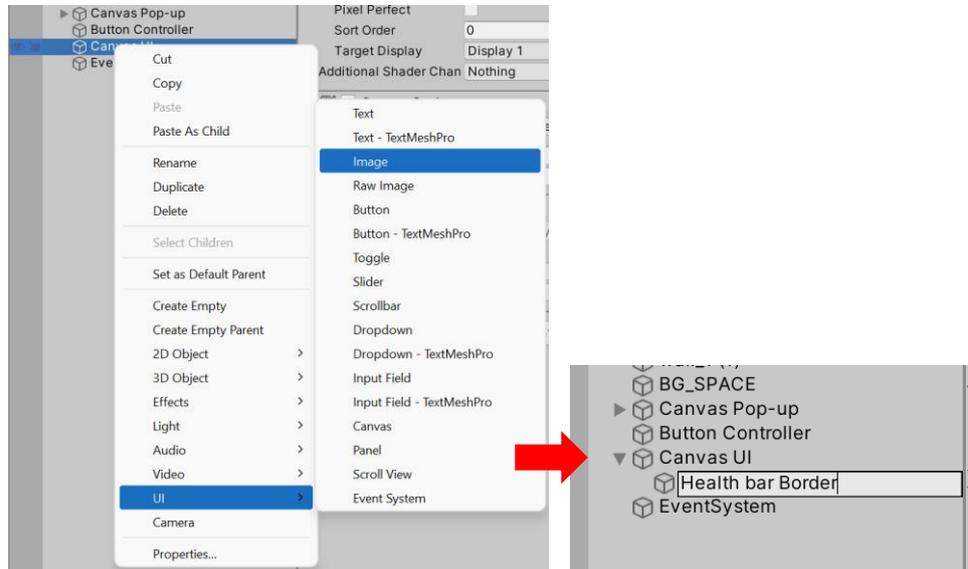


DFG40323 - 2D GAME DEVELOPMENT

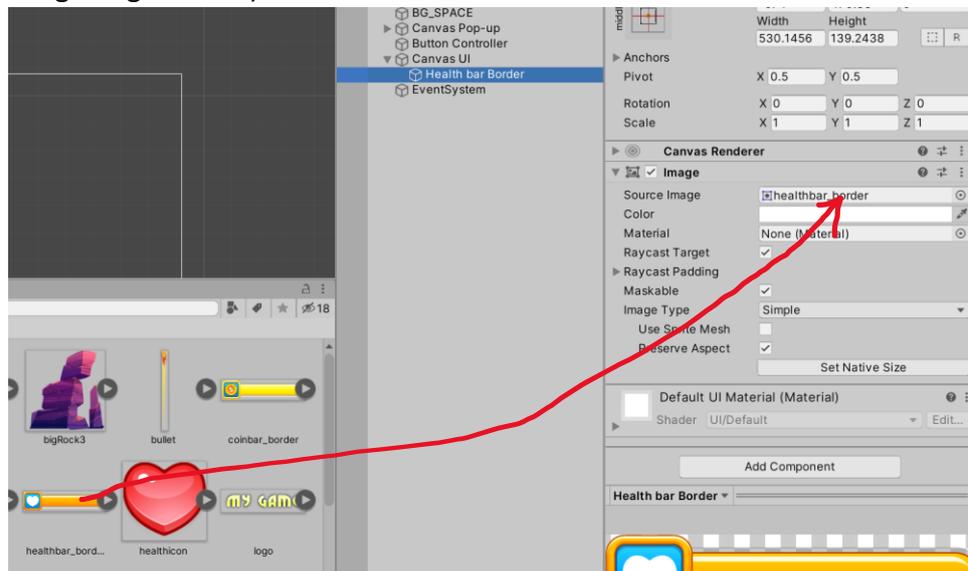
Now : LABSHEET 5 : Game Scripting (part 1)

Next : LABSHEET 5 : Game Scripting (part 2)

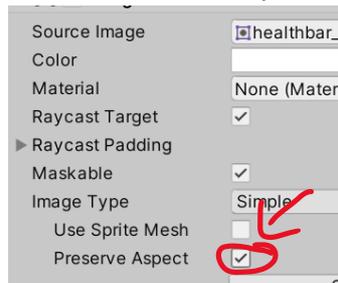
- b. Then, create new Image inside “Canvas UI” ; (Select Canvas UI → Right Click → UI → Image → Name as “Health bar border”



- c. Change image source for “Health bar Border” as healthbar_border, (Select “Health bar border → Inspector Panel → Image Source → Image Source → Drag Image into it)



- d. Select “Preserve Aspect” and resize your “Heathbar Border”

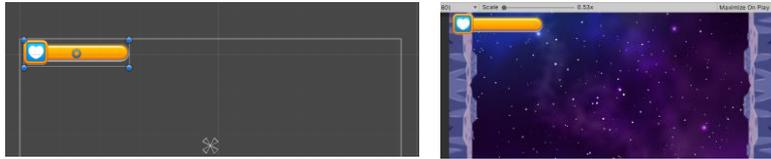


DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 1)

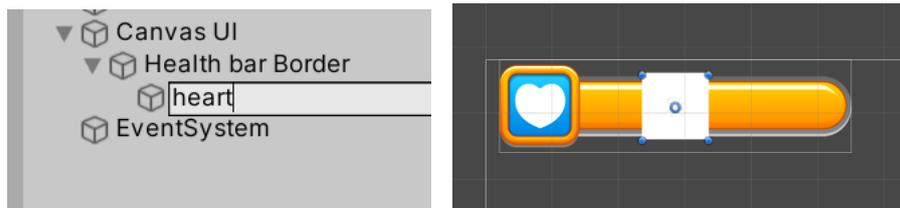
Next : LABSHEET 5 : Game Scripting (part 2)

e. Reposition your bar as below :

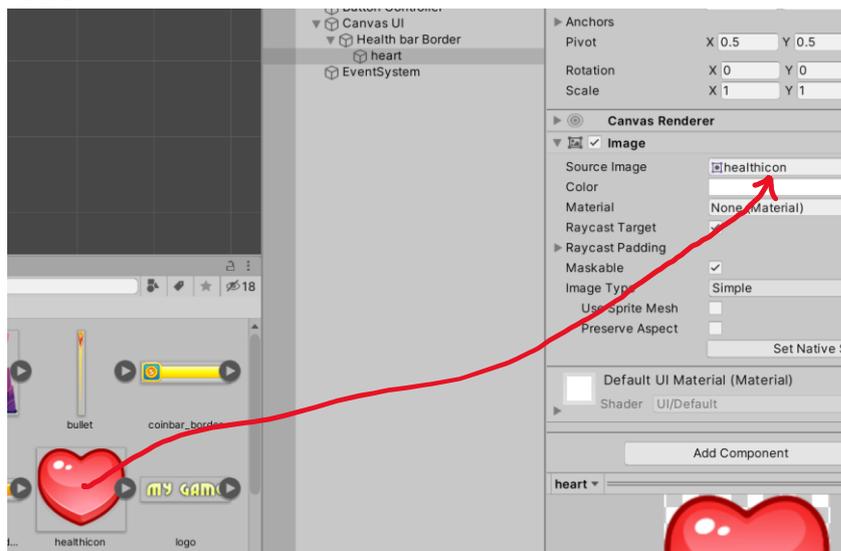


3. Adding health icon

a. Insert new Image inside “Healthbar border” , and rename as “heart” show as below:



b. Change image source for “heart’ to “healthicon” from Assets/Graphic folder



c. Duplicate and reposition “heart” and rename as “heart1”, “heart2” and “heart3” shown as below :



d. Save your work

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 1)

Next : LABSHEET 5 : Game Scripting (part 2)

EXERCISE

1. Complete your project as output below for every scene you have :



DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 1)

Next : LABSHEET 5 : Game Scripting (part 2)

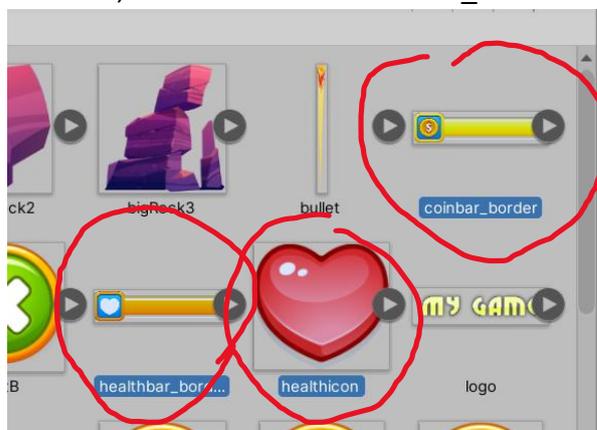
A) Setting up Health bar

1. Importing health bar graphic

- Open your previous project (Labsheet 4 Part 4)
- Download and Import Labsheet 5 part 1 Resources from CIDOS into Asset/Graphics

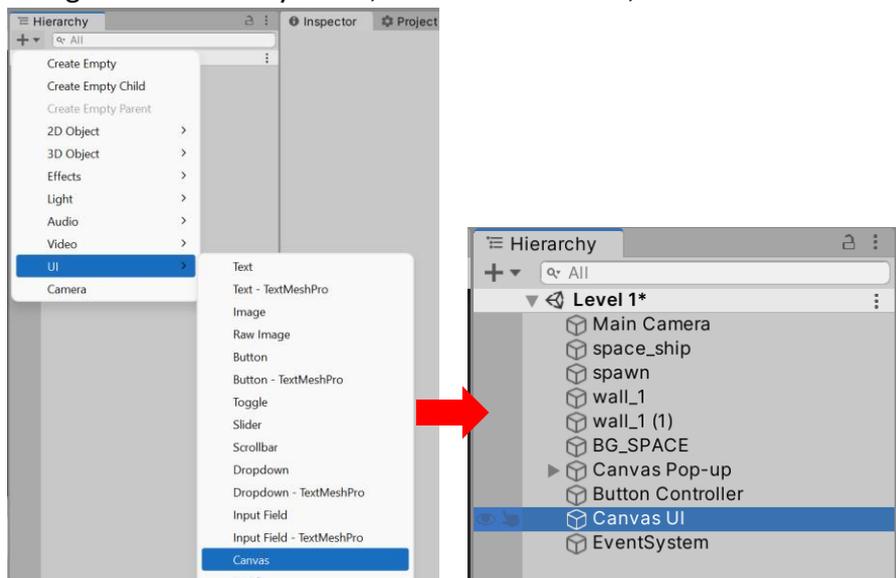


- Navigate to Asset → Graphics; make sure you have 3 graphics which are coinicon, healthicon and healthbar_border as below :



2. Creating health bar inside Hierarchy Panel

- Navigate to Hierarchy Panel, create new canvas, and name it as "Canvas UI"

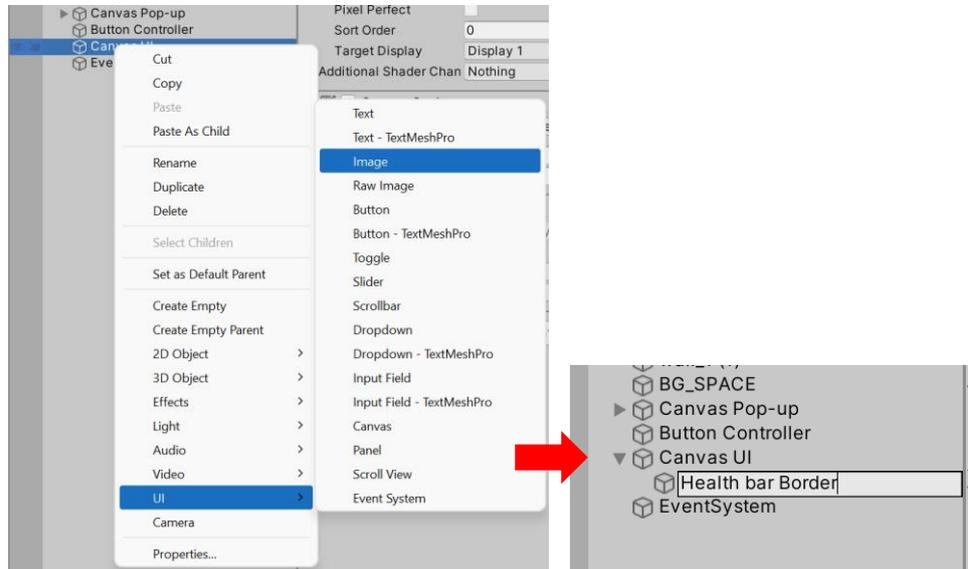


DFG40323 - 2D GAME DEVELOPMENT

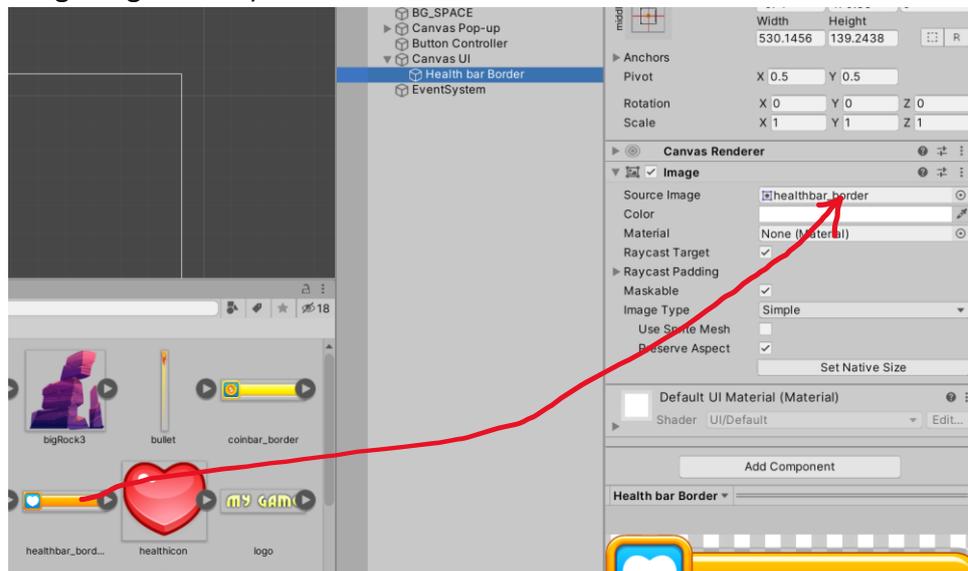
Now : LABSHEET 5 : Game Scripting (part 1)

Next : LABSHEET 5 : Game Scripting (part 2)

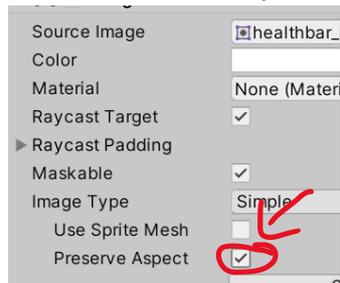
- b. Then, create new Image inside “Canvas UI” ; (Select Canvas UI → Right Click → UI → Image → Name as “Health bar border”



- c. Change image source for “Health bar Border” as healthbar_border, (Select “Health bar border → Inspector Panel → Image Source → Image Source → Drag Image into it)



- d. Select “Preserve Aspect” and resize your “Heathbar Border”

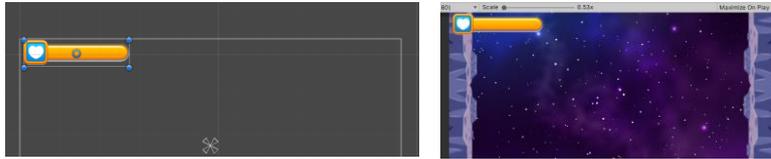


DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 1)

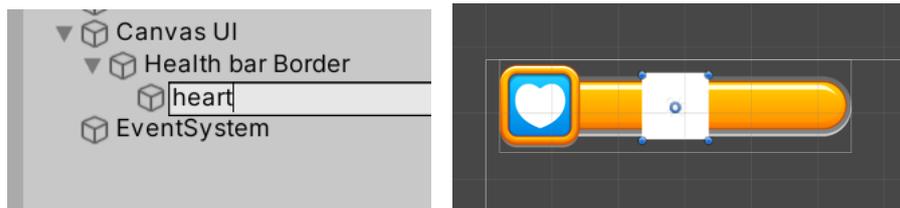
Next : LABSHEET 5 : Game Scripting (part 2)

e. Reposition your bar as below :

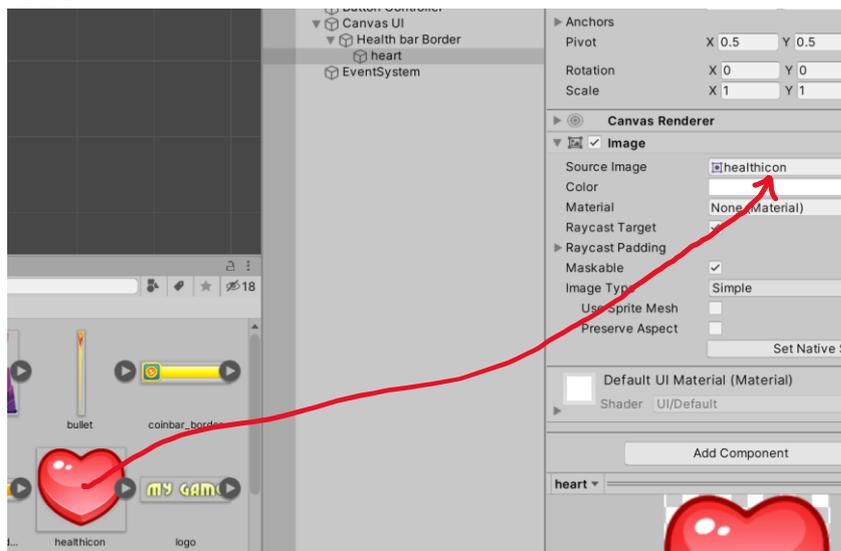


3. Adding health icon

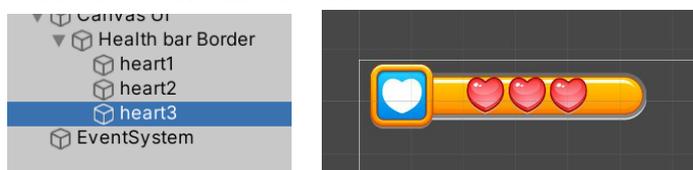
a. Insert new Image inside “Healthbar border” , and rename as “heart” show as below:



b. Change image source for “heart’ to “healthicon” from Assets/Graphic folder



c. Duplicate and reposition “heart” and rename as “heart1”, “heart2” and “heart3” shown as below :



d. Save your work

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 1)

Next : LABSHEET 5 : Game Scripting (part 2)

EXERCISE

1. Complete your project as output below for every scene you have :



DFG40323 - 2D GAME DEVELOPMENT

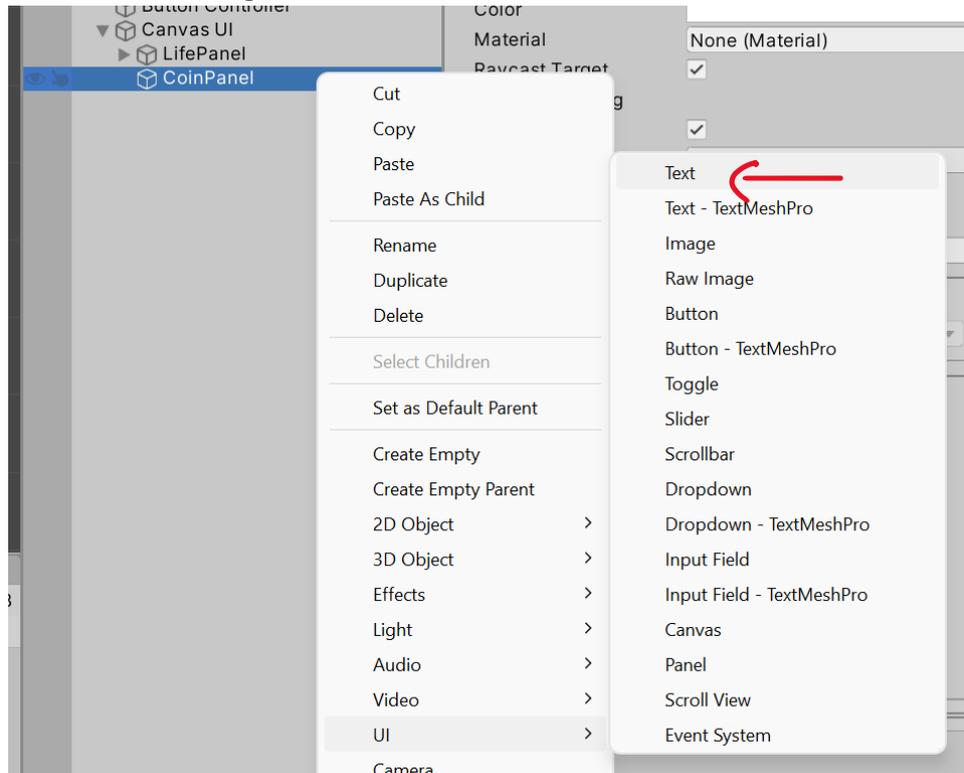
Now : LABSHEET 5 : Game Scripting (part 3)

Next : Practical Task 4: Game Scripting (part 4)

A) Preparing Script for Coin

1. Textfield for coin

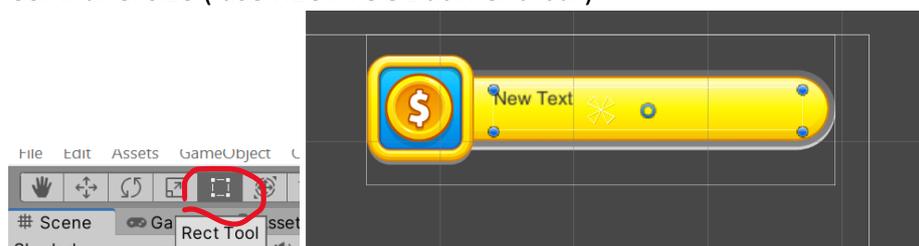
- Open your previous project and open your scene file (either level 1 or 2).
- Add new UI Text inside Canvas UI, (Navigate Hierarchy → Canvas UI → CoinPanel → Right Click → UI → Text



- Rename your Text as “Coin_sum” as below:



- Navigate to Inspector Panel and resize your text size to fit with your CoinPanel size (use RECT TOOL at menu bar)

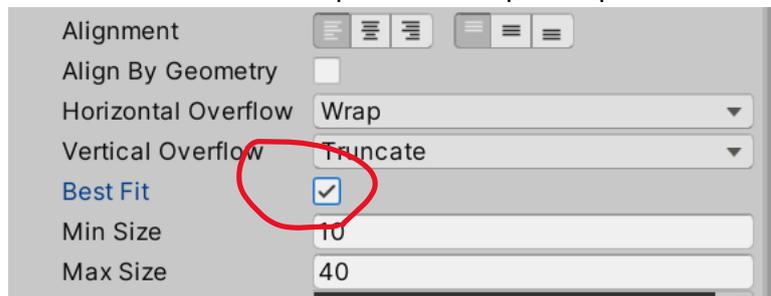


DFG40323 - 2D GAME DEVELOPMENT

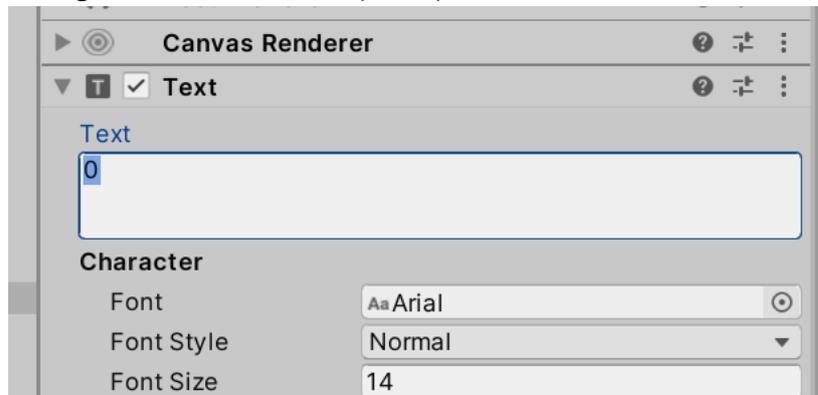
Now : LABSHEET 5 : Game Scripting (part 3)

Next : Practical Task 4: Game Scripting (part 4)

- e. Next Tick “BEST FIT” component in inspector panel

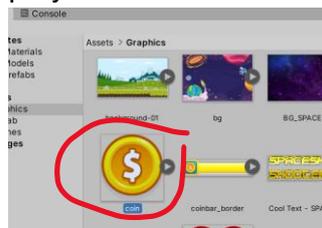


- f. Change “New Text” to “0” (ZERO)

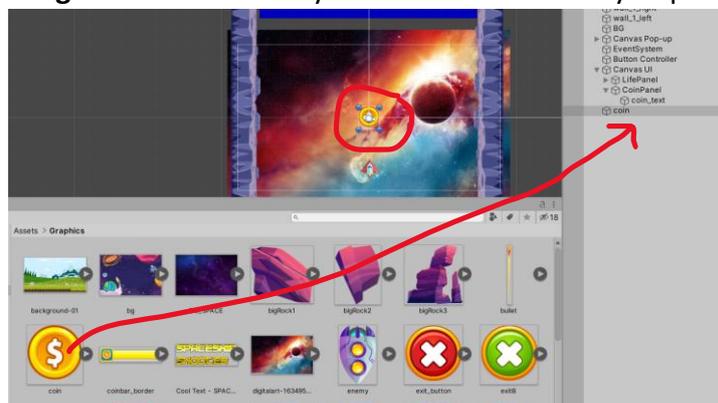


2. Preparing Coin drop

- a. **Import** New Labsheet 5 Part 3 Resources from CIDOS and import into your project



- b. **Drag** coin into Hierarchy Panel and resize it as you prefer

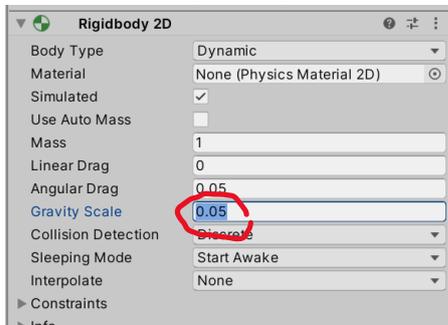


DFG40323 - 2D GAME DEVELOPMENT

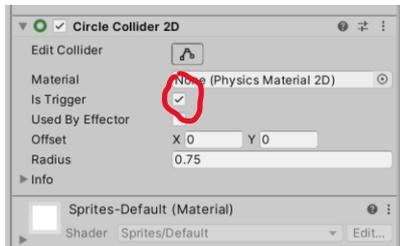
Now : LABSHEET 5 : Game Scripting (part 3)

Next : Practical Task 4: Game Scripting (part 4)

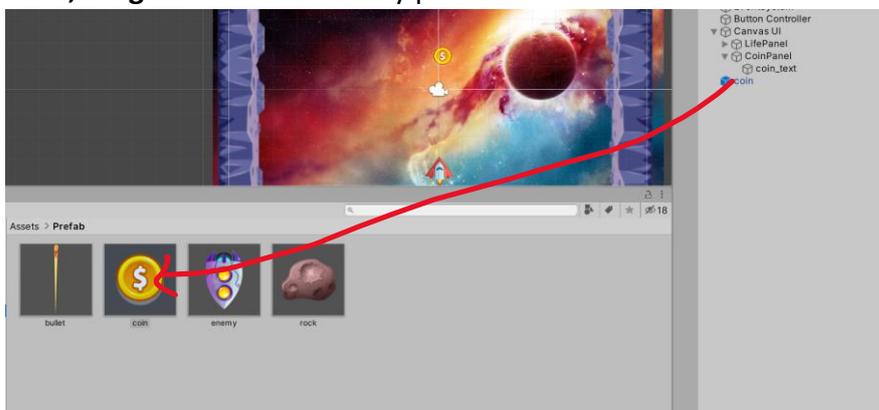
- c. Select at the coin, add **Rigidbody2D** and **Circle Collider2D** component into it
- d. Change setting for Rigidbody2D as below:



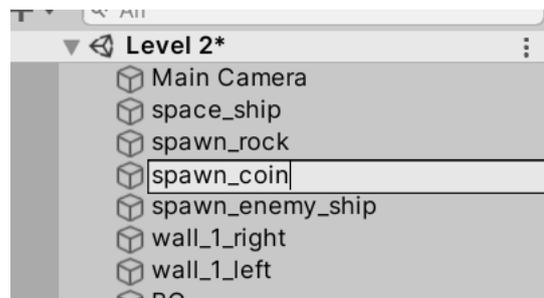
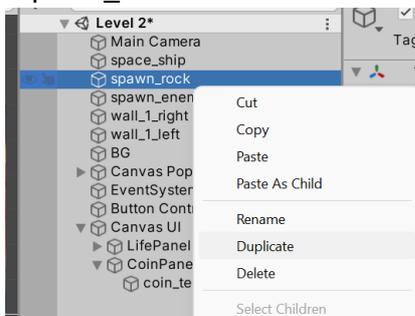
- e. And tick **ISTRIGGER** on Circle Collider 2D



- f. Next, drag coin from hierarchy panel into PREFAB folder



- g. Then delete coin in Hierarchy panel
- h. Next, duplicate "spawn_rock" from hierarchy panel and rename as "spawn_coin"

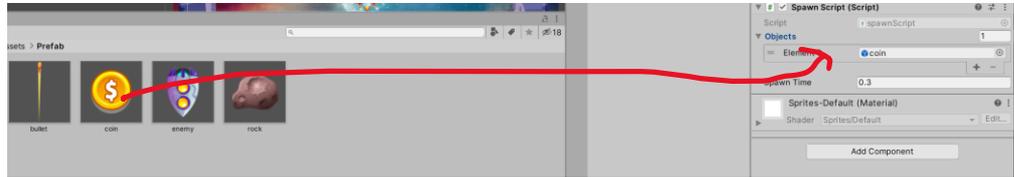


DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 3)

Next : Practical Task 4: Game Scripting (part 4)

- i. **Navigate to** hierarchy panel, change Object Elemen 0 to “coin” from Prefab folder (drag and drop)



3. Scripting Spaceship and Coin trigger

- a. Open “SpaceshipScript.cs”
- b. Add this line into your script:

```
38     if (name == "rock(Clone)")
39     {
40         //heart1.SetActive(false);
41         life--; //life deducted by 1
42         Debug.Log("nilai semasa life : " + life);
43
44         if(life == 2)
45         {
46             heart3.SetActive(false);
47         }
48         if( life == 1)
49         {
50             heart2.SetActive(false);
51         }
52         if (life == 0)
53         {
54             Time.timeScale = 0;
55             Pop_up_try_again.SetActive(true); //paparkan popup
56             //heart1.SetActive(false);
57         }
58         // Destroy itself (the enemy) to keep things simple
59         // Destroy(gameObject);
60     }
61     if (name == "coin(Clone)")
62     {
63         Destroy(other.gameObject);
64     }
```

- c. Save your script and test your project.

DEPARTMENT OF INFORMATION TECHNOLOGY AND COMMUNICATION

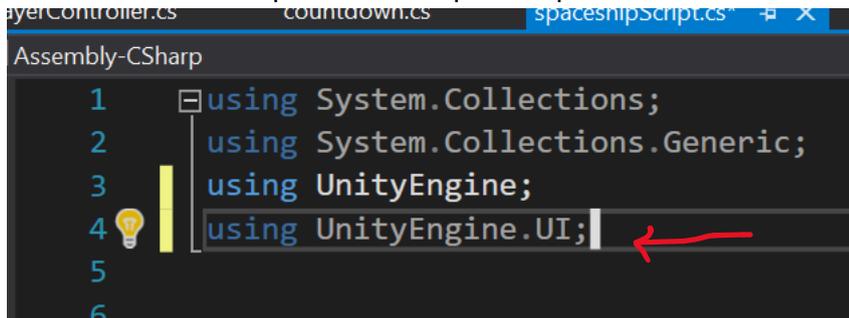
Practical Task (s) No: 1 / 2 / 3 / 4

CLO 1: Display 2D assets, game interactivity, level design and scripting skills of game development tools to maintain coherency in game. (P4, PLO3)

A) Displaying Score

1. Modify script

- a. Open your previous project (Labsheet 5 Part 3)
- b. Open “spaceshipScript.cs” to edit
- c. Add new line namespace at the top of script as below:

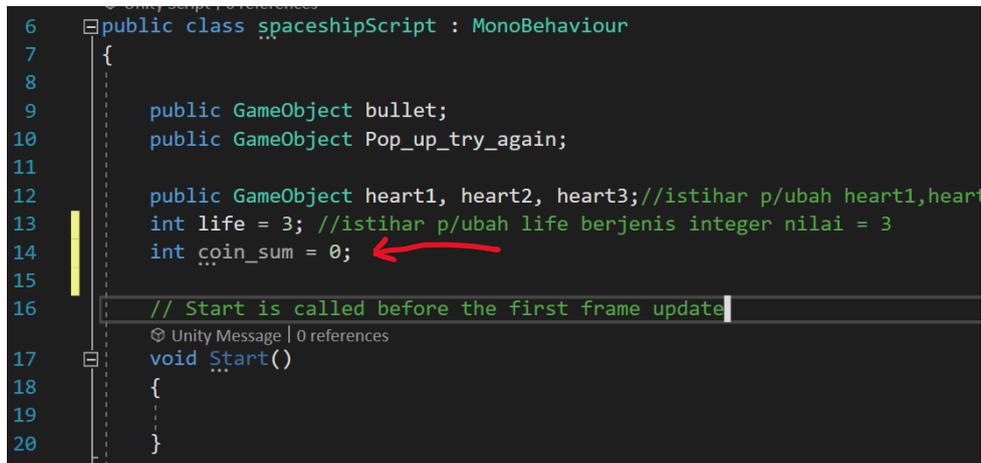


```

Assembly-CSharp
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6

```

- d. Register new variable called “coin_sum” with ZERO as default value at the top of class as below:

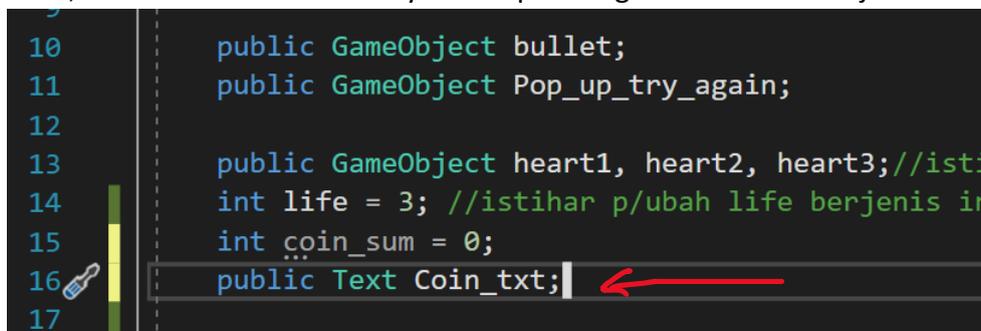


```

6  public class spaceshipScript : MonoBehaviour
7  {
8
9      public GameObject bullet;
10     public GameObject Pop_up_try_again;
11
12     public GameObject heart1, heart2, heart3; //istihar p/ubah heart1,heart
13     int life = 3; //istihar p/ubah life berjenis integer nilai = 3
14     int coin_sum = 0;
15
16     // Start is called before the first frame update
17     void Start()
18     {
19
20     }

```

- e. Next, add new line below into your script to register new Text Object.



```

10     public GameObject bullet;
11     public GameObject Pop_up_try_again;
12
13     public GameObject heart1, heart2, heart3; //isti
14     int life = 3; //istihar p/ubah life berjenis in
15     int coin_sum = 0;
16     public Text Coin_txt;
17

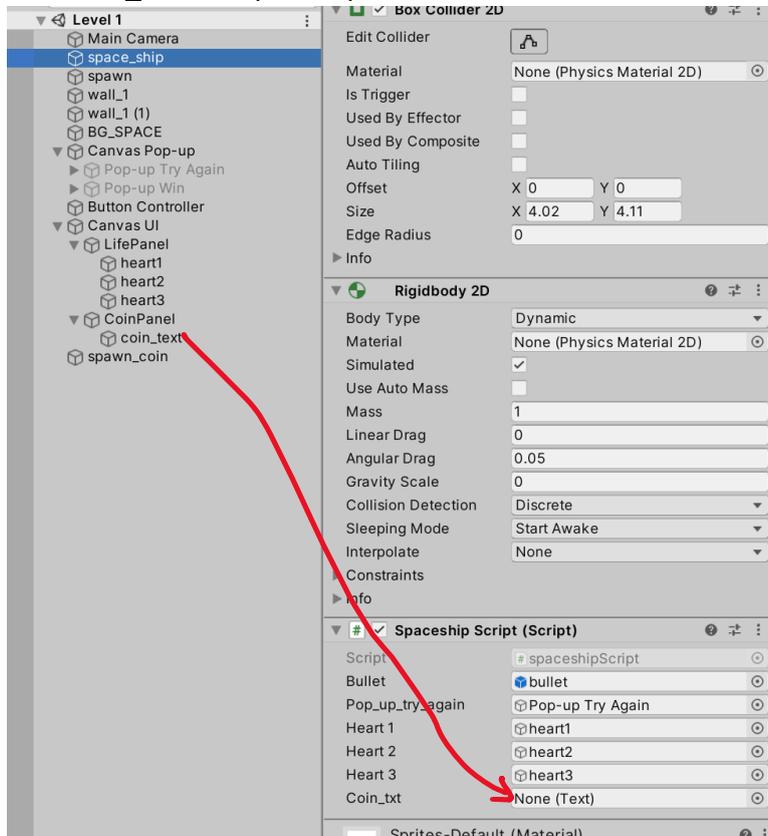
```

f. Next, add these lines into your script file and save your script:

```

Unity Message | 0 references
void OnTriggerEnter2D(Collider2D other)
{
    var name = other.gameObject.name;
    Debug.Log(name);
    if (name == "rock(Clone)")...
    if (name == "coin(Clone")//jika spaceship berlanggar dgn
    {
        Destroy(other.gameObject);// coin akan hilang
        coin_sum++;
        Coin_txt.text = coin_sum.ToString();
    }
}
    
```

g. Last step, open your scene and navigate to hierarchy panel; drag “coin_text” into Coin_txt at inspector panel as below:



h. Save your project and test it.

Now: Practical Task 4

Name: _____

Registration Num.: _____

Code & Course: DFG40323 - 2D GAME DEVELOPMENT

Programme/Class: _____

Practical Task (s) No: 1 / 2 / 3 / 4

MARKS: /20

CLO 1: Display 2D assets, game interactivity, level design and scripting skills of game development tools to maintain coherency in game. (P4, PLO3)

1. Make sure all scene (Level 1 and Level 2) will show total coin collected by user.
2. Paste your final output below for BOTH scene:

DFG40323 - 2D GAME DEVELOPMENT

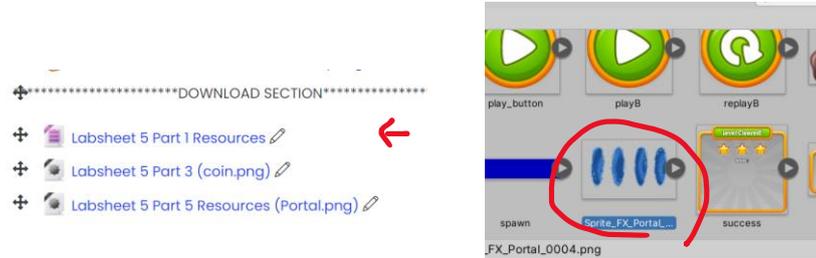
Now : LABSHEET 5 : Game Scripting (part 5)

Next : Labsheet 5: Game Scripting (part 6)

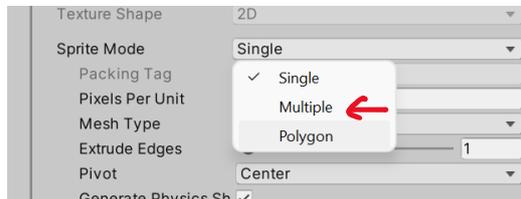
A) Preparing Portal

1. Setting up Portal Animation

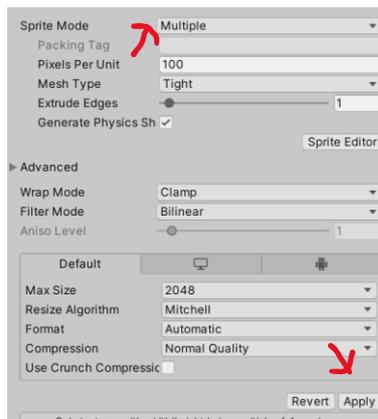
- Open your previous project (Practical Task 4) → **LEVEL 1 SCENE**
- Download Labsheet 5 Part 5 Resources and import into your project



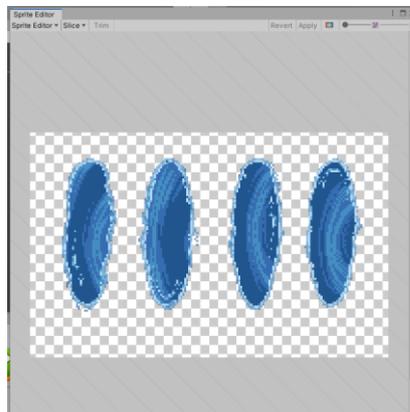
- Click at the graphic → Inspector Panel, and setting up SPRITE MODE from SINGLE to MULTIPLE as below :



- Press APPLY



- Next, Click SPRITE EDITOR and window will pop up as below:

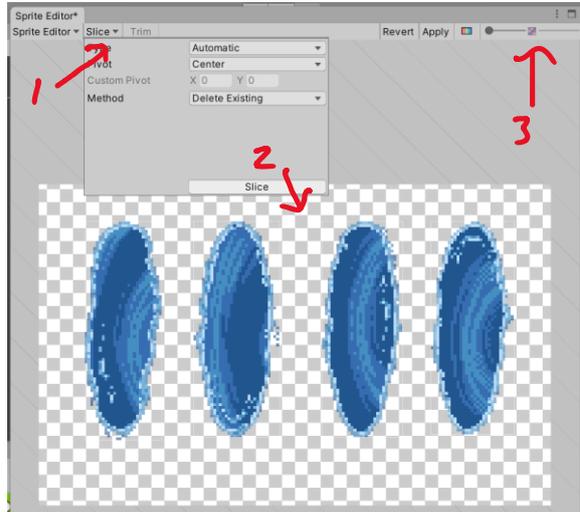


DFG40323 - 2D GAME DEVELOPMENT

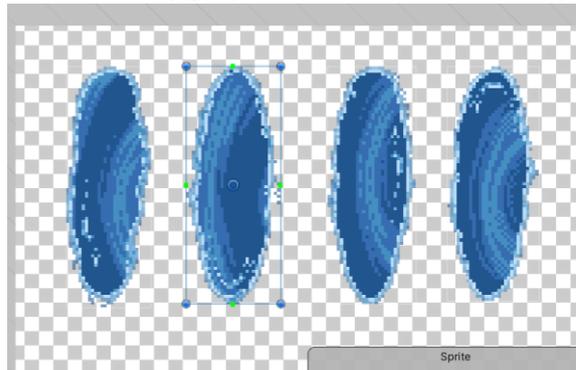
Now : LABSHEET 5 : Game Scripting (part 5)

Next : Labsheet 5: Game Scripting (part 6)

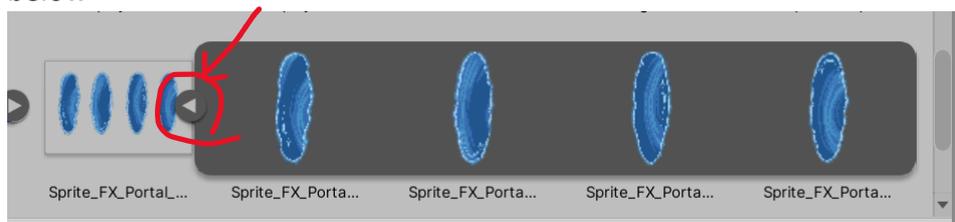
f. Then, click "Slice" → click Slice → Click APPLY



g. Click at every graphic, and make sure every graphic are break apart



h. Close the window, next navigate to Graphic folder → select at the ARROW button beside the portal graphic and make sure the graphic will produce as below

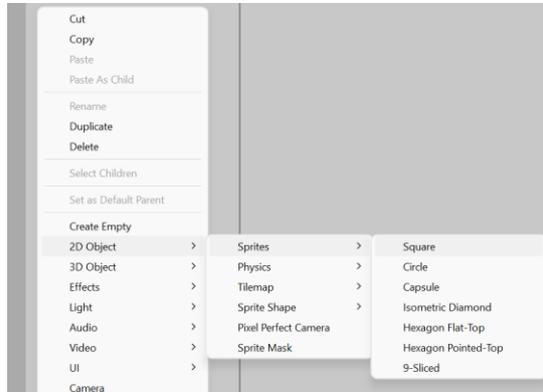


DFG40323 - 2D GAME DEVELOPMENT

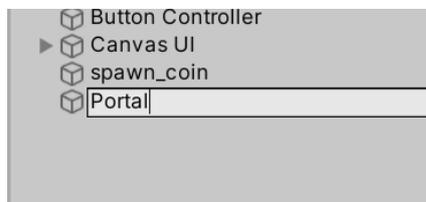
Now : LABSHEET 5 : Game Scripting (part 5)

Next : Labsheet 5: Game Scripting (part 6)

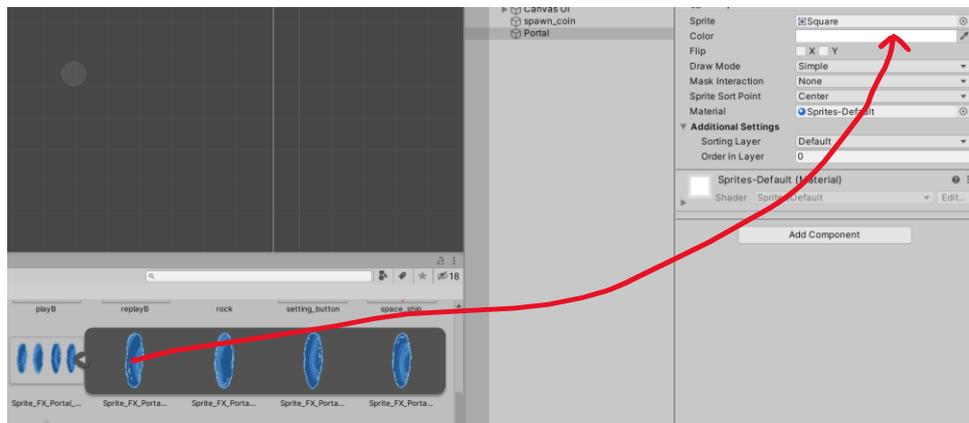
- i. Next, Create new 2D SPRITE at Hierarchy Panel (navigate to Hierarchy Panel → 2D Object → Sprites → Square



- j. Rename as "Portal"



- k. Navigate to Inspector Panel → Drag "Portal's First Slice" into Sprite area as below :



- l. Resize your "portal" as below

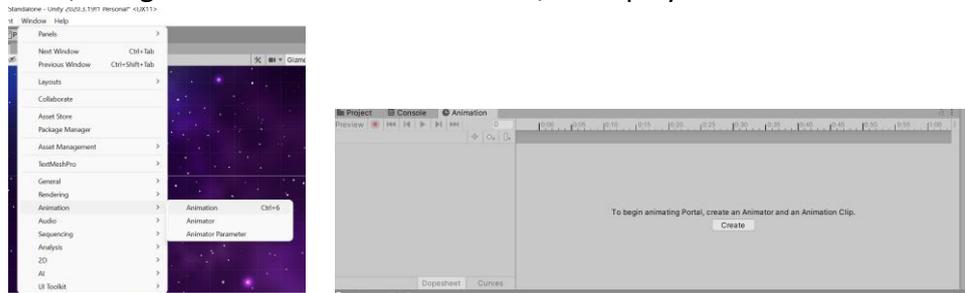


DFG40323 - 2D GAME DEVELOPMENT

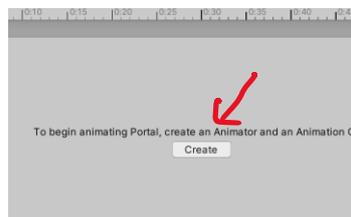
Now : LABSHEET 5 : Game Scripting (part 5)

Next : Labsheet 5: Game Scripting (part 6)

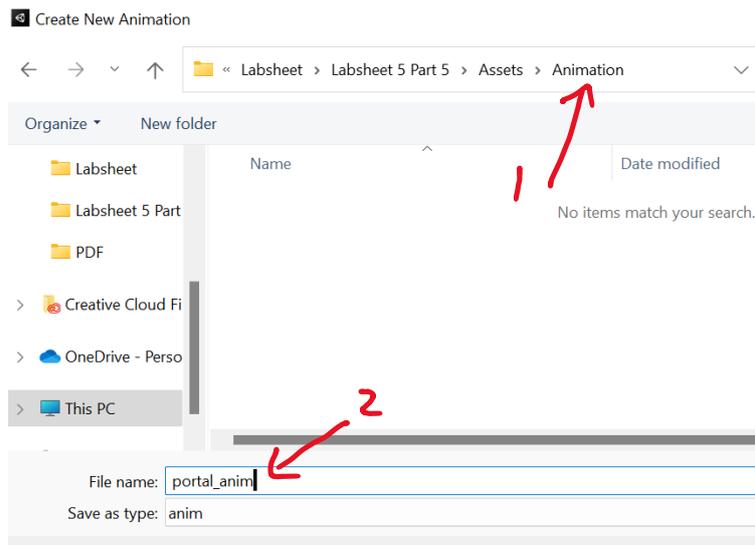
m. Next, navigate to Window → Animation , to display animation window



n. Next, Click at “portal” at hierarchy panel → Animation windows → Click “CREATE”



o. Save as “portal_anim” (save inside Animation folder)



p. Next it will end up like below

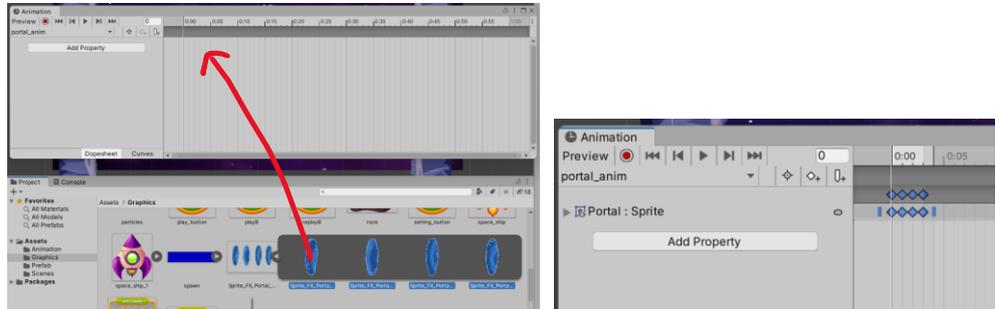


DFG40323 - 2D GAME DEVELOPMENT

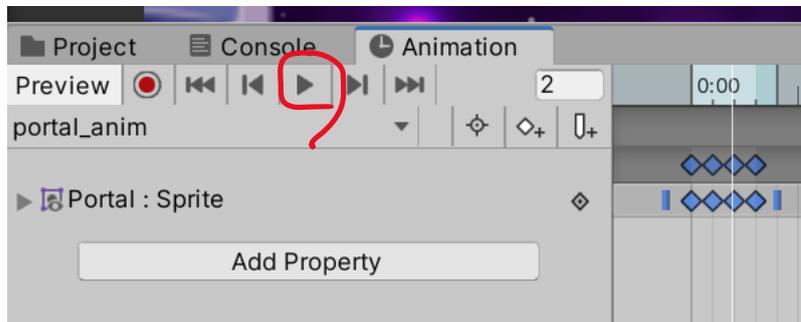
Now : LABSHEET 5 : Game Scripting (part 5)

Next : Labsheet 5: Game Scripting (part 6)

- q. Next, drag 4 (FOUR) clip portal below into animation timeline as below (Select All graphics and drag):



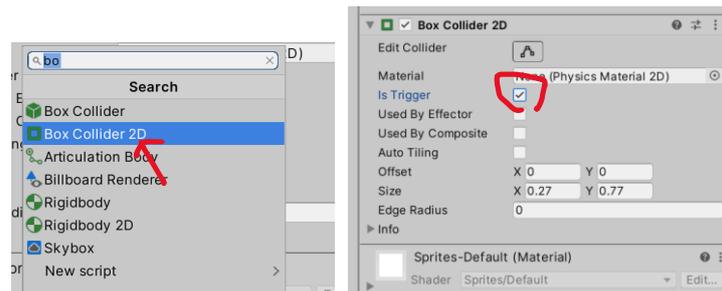
- r. Preview your animation by click on the PLAY button below



- s. Reposition your "PORTAL" as below :



- t. Add "BOX COLLIDER2D into the "PORTAL" and tick "ISTRIGGER" item on it



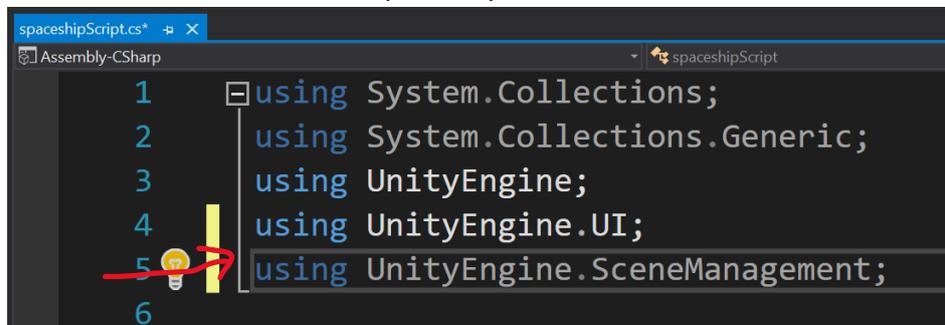
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 5)

Next : Labsheet 5: Game Scripting (part 6)

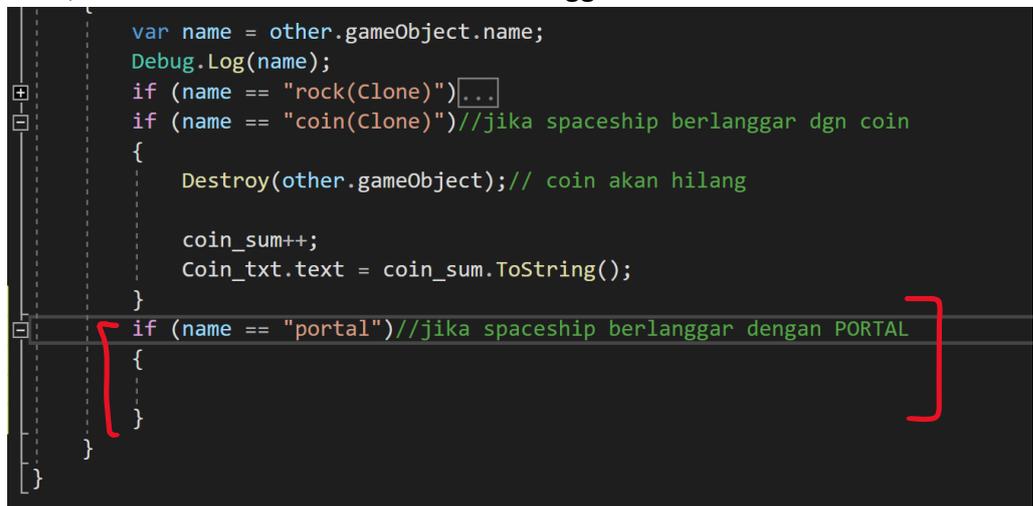
2. Scripting "Portal"

- Open "spaceshipScript.cs" to modify script
- Add new line below at the top of script :



```
spaceshipScript.cs* x
Assembly-CSharp spaceshipScript
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using UnityEngine.SceneManagement;
6
```

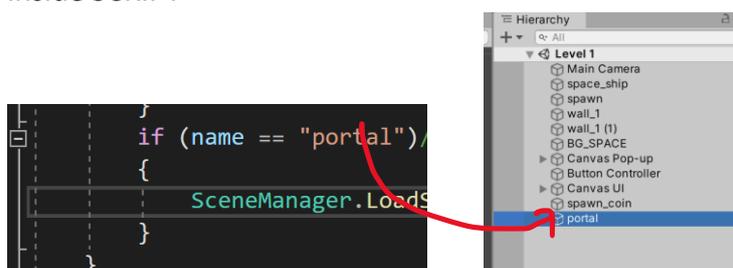
- Next, add these "IF" condition inside OnTriggerEnter2D



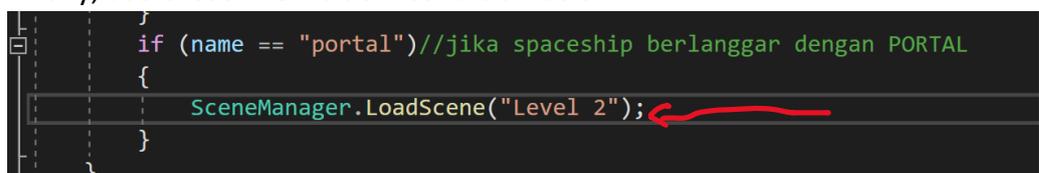
```
var name = other.gameObject.name;
Debug.Log(name);
if (name == "rock(Clone)")
if (name == "coin(Clone")//jika spaceship berlanggar dgn coin
{
    Destroy(other.gameObject);// coin akan hilang

    coin_sum++;
    Coin_txt.text = coin_sum.ToString();
}
if (name == "portal")//jika spaceship berlanggar dengan PORTAL
{
}
```

- Make sure, the name of your "portal" inside HIERARCHY same as name inside SCRIPT



- Finally, add these line inside If condition below :



```
if (name == "portal")//jika spaceship berlanggar dengan PORTAL
{
    SceneManager.LoadScene("Level 2");
}
```

DFG40323 - 2D GAME DEVELOPMENT

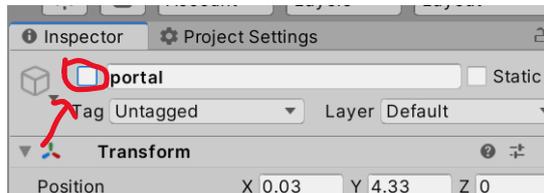
Now : LABSHEET 5 : Game Scripting (part 5)

Next : Labsheet 5: Game Scripting (part 6)

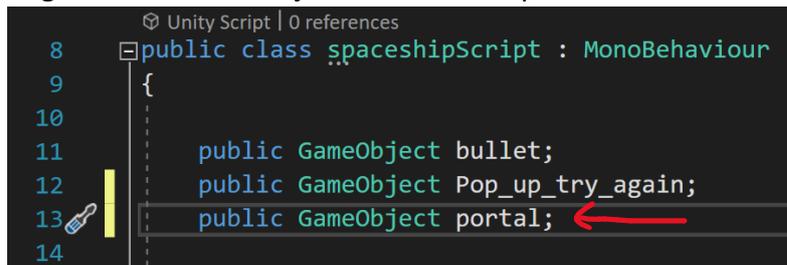
- f. Make sure, name of SCENE inside script, same as the scene name for your LEVEL 2.



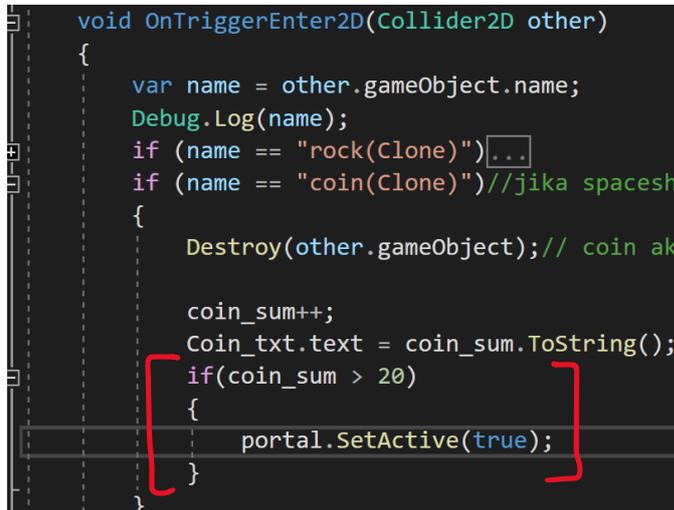
- g. Save and test your project, make sure that your script is working. (Level 2 scene will load after spaceship colliding the portal)
h. Next, disable your "PORTAL" at hierarchy panel



- i. Register new GameObject variable for "portal"



- j. Add below IF condition below inside your script

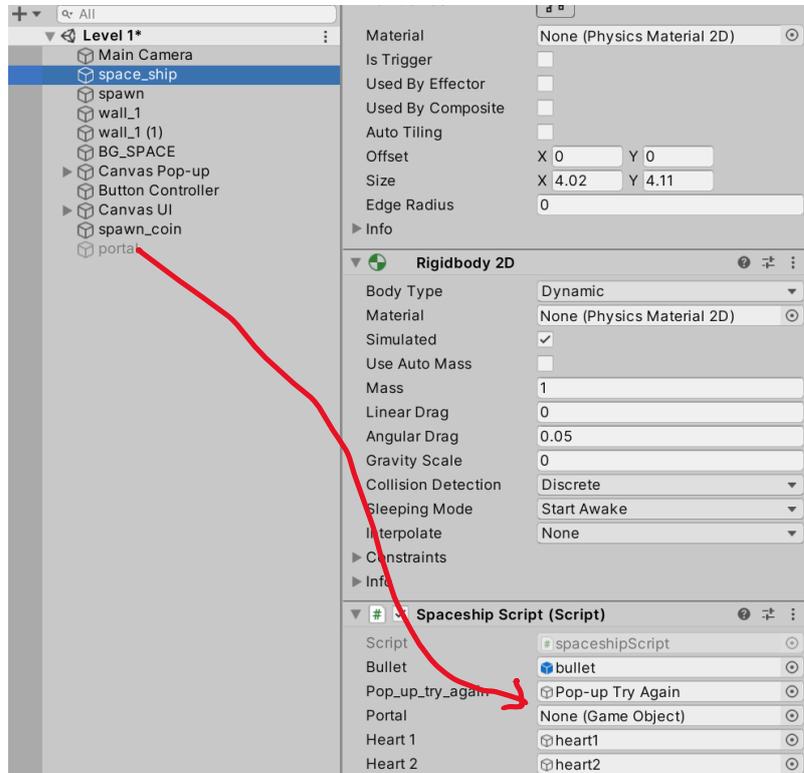


DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 5)

Next : Labsheet 5: Game Scripting (part 6)

- k. Drag “portal” from hierarchy Panel into “spaceshipScript” at Inspector Panel



- l. Save your project and run

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 6)

A) Preparing Portal

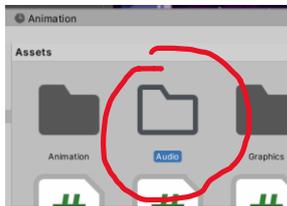
1. Importing Audio File

- a. Open your previous project (Practical Task 5) → **LEVEL 1 SCENE**
- b. Download Labsheet 5 Part 6 Resources and extract it

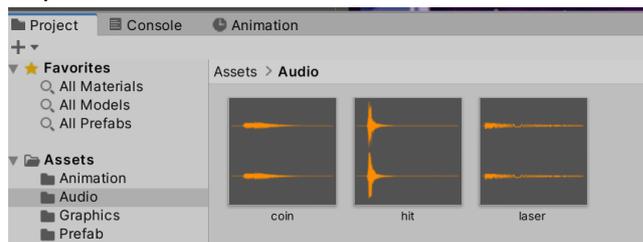
*****DOWNLOAD SECTION*****

-  Labsheet 5 Part 1 Resources
-  Labsheet 5 Part 3 (coin.png)
-  Labsheet 5 Part 5 Resources (Portal.png)
-  Labsheet 5 Part 6 Resources ←

- c. Create new folder insider ASSET folder called Audio (Right Click → Create → Folder → Rename as “Audio”

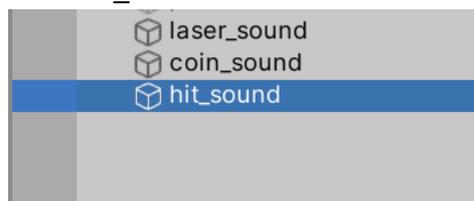


- d. Import downloaded file inside Audio Folder inside Asset

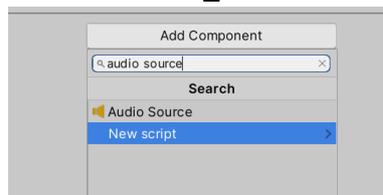


2. Setting up Audio file

- a. Create THREE (3) newEmpty and name it as “laser_sound”, “coin_sound” and “hit_sound”



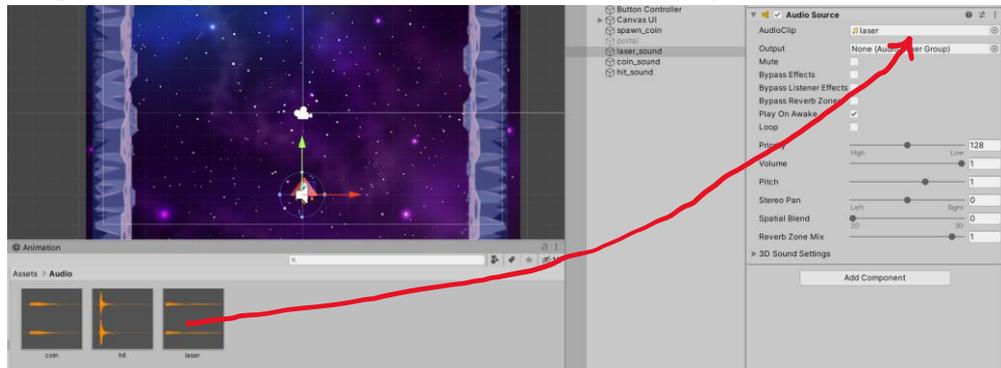
- b. Select at “laser_sound” and add new component called “Audio Source”



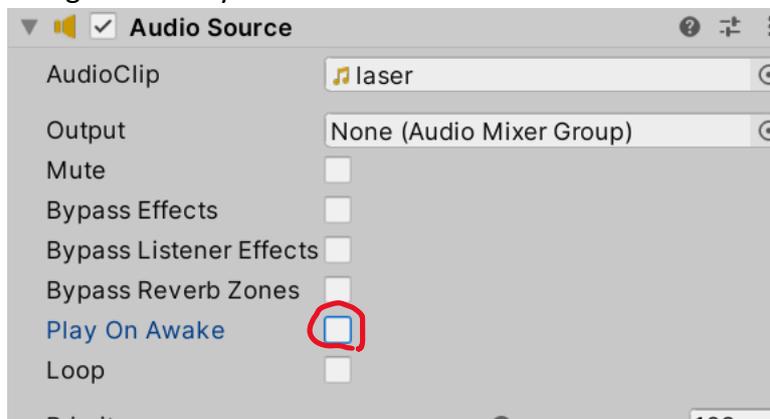
DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 6)

- c. Drag "laser.mp3" from Audio Folder inside "Audio Clip" area as below:



- d. Navigate to "Play On Awake" and UNTICK it.



- e. Do the same step (step b,c and d) to the "coin_sound" and "hit_sound"

3. Scripting audio file

- a. Select "spaceShipScript.cs" and open it
b. Register new AudioSource Variable for "laser, coin and hit" as below :

```
public class spaceshipScript : MonoBehaviour
{
    public GameObject bullet;
    public GameObject Pop_up_try_again;
    public GameObject portal; //istihar portal gameobject
    public AudioSource laser, coin, hit; ←
```

- c. Navigate to line below and add this script :

```
35
36     if (Input.GetMouseButtonDown(0))
37     {
38         Instantiate(bullet, transform.position, Quaternion.identity);
39         laser.Play(); ←
40     }
41
42 }
```

DFG40323 - 2D GAME DEVELOPMENT

Now : LABSHEET 5 : Game Scripting (part 6)

- d. Next, navigate to line below and add this script to Play HIT audio file :

```
Debug.Log(name);
if (name == "rock(Clone)")
{
    //heart1.SetActive(false);
    life--; //life deducted by 1
    hit.Play();
}

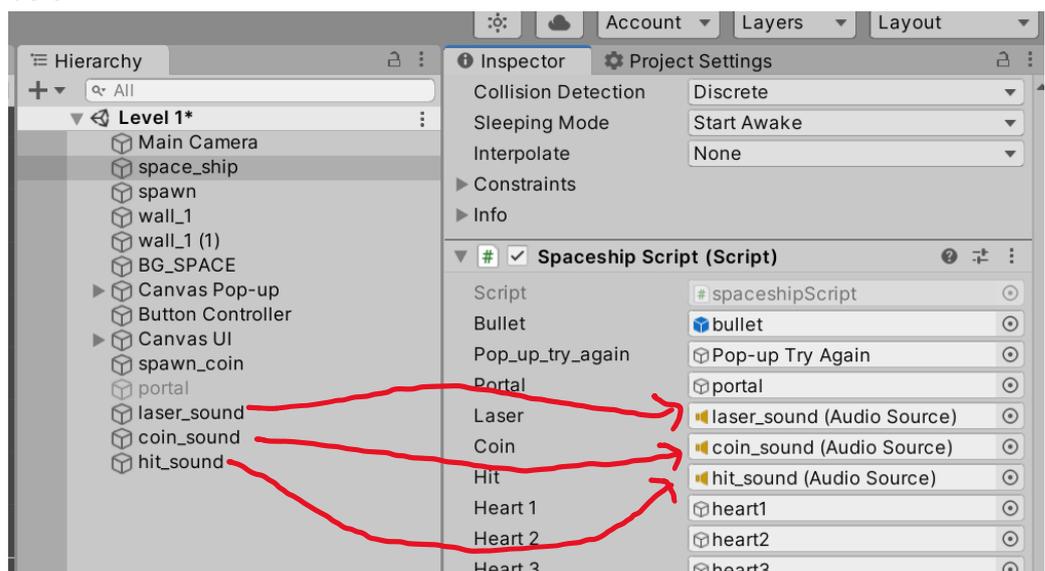
if(life == 2)
{
```

- e. And last, this script to play COIN audio file :

```
69 }
70 if (name == "coin(Clone")//jika spaceship berlanggar dgn coin
71 {
72     Destroy(other.gameObject);// coin akan hilang
73     coin.Play();
74     coin_sum++; //menambahkan nilai coin
75     Coin_txt.text = coin_sum.ToString();//memaparkan jumlah coint panel
76     if(coin_sum >= 5)// coin lebih besar dari 20
77     {
```

- f. Save your Script.

- g. Next, navigate to Hierarchy Panel, select SpaceShip and Drag "laser_sound", "coin_sound" and "hit_sound" into inspector Panel as below :



Exercise:

- Do the same step for Game Scene LEVEL 2
- Download any MUSIC EFFECT and add it into your project, modify your project to play the music at the start of the scene game

Appendix

References & Resources

REFERENCES

Franz, L. (2020). 2D Game Development with Unity 1st Edition. CRC Press. (ASIN: B08L5L7TC3)

Halpern, J. (2019). Developing 2D games with Unity: independent game programming with C#. NY, NY: Apress. (ISBN: 13:978-1484237717)

Jarred, C. & Louis, S. (2020). MonoGame Mastery: Build a Multi Platform 2D Game and Reusable Game Engine 1st ed. Apress. (ISBN: 1484263081)

Patrick, F. (2020). A Beginner's Guide to 2D Shooter Games with Unity: Create a Simple 2D Shooter Game and Learn to Code in C# in the Process. Independently published. (ISBN: 979-8555881885)

Tyers, B. G. (2018). Beginning GameMaker Studio 2: Master The Basics of GML Programming. Independently published. (ISBN: 13: 978-1790577156)

Special Resources to Deliver The Course :

- <https://tinyurl.com/2DGameeBookResources>. Resources



e ISBN 978-967-0074-05-4



9 789670 074054

2D GAME DEVELOPMENT