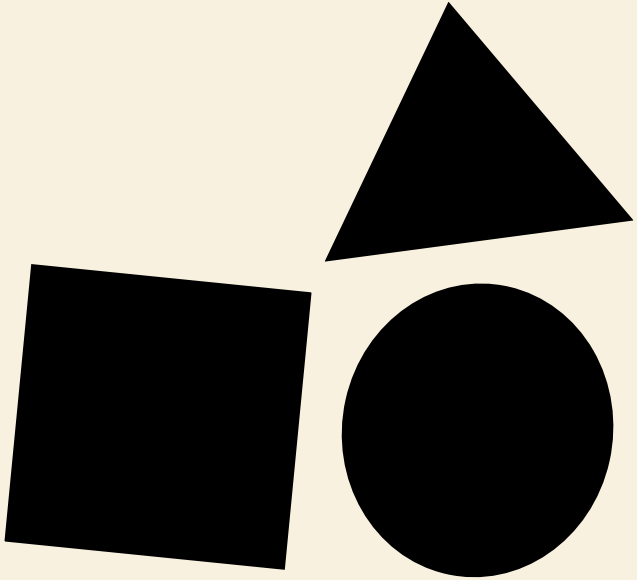
Slice IT !



By Joseph Grados, Ashley Mendez,
Anaya Yorke, and Greg Smith
Georgia Gwinnett College
TAP
Fall 2022| STEC 4800

Pre-Survey

- Please take our pre-survey before we start on the workshop by clicking on the link below: **https://ggc.az1.qualtrics.com/jfe/form/SV_5iEEPaoyKEonOQe**

What Is TAP?

TAP stands for the Technology Ambassadors Program

What do we do in **TAP**:

- Expo (to showcase our game, demoing)
- CREATE
- Workshops
- Improve technical skills
- Improve communication skills

Introducing our team!!

Anaya Yorke🥳
UI/Website Dev
Major: Applied Math Major/Pre-eng Track

Joseph Grados😆
UI/ Lead Software Developer
Major: Software Development

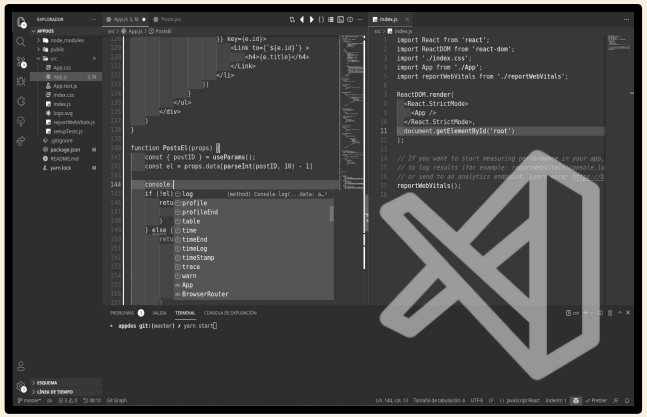
Greg Smith😎
UX / Lead Software Developer

Ashley Mendez🤪
UX

Technology/Software used

- To Make Our Game 🕹️
  - Unity
  - GitHub Collaborative
  - Visual Studios

- To Make Our Website 💻
  - Figma
  - Anima
  - Github Pages
  - Visual Studios Code



Introducing Boolean Expressions!

- A **boolean** is a data type that is used for **conditional statement**.
- Some of the syntax for boolean is &&(for and) ||(for or) and !(for not), and when combine it can change if the boolean statement is logically right or not.
- Examples:
- True or True = True (as no matter the opinion in the end it's true)
- True and False = False(and means that both conditions need to be the same to appropriately match)
- True or !True = True(the or means even that both conditions can be true, and the not sign means it's not true)

Question 1

Which of the following expression “color is green” true?
Hint – three answer will be true

- ❑ Red Sphere
- ❑ Green Sphere
- ❑ Blue Sphere
- ❑ Red Cube
- ❑ Green Cube
- ❑ Red Cylinder
- ❑ Green Cylinder
- ❑ Blue Cylinder

Question 2

Which of the following shapes will make the statement (“shape is cube”) true?

- ❑ Green Cube
- ❑ Red Sphere
- ❑ Blue Cylinder
- ❑ Orange Cube
- ❑ Blue Sphere
- ❑ White Cube
- ❑ Green Sphere

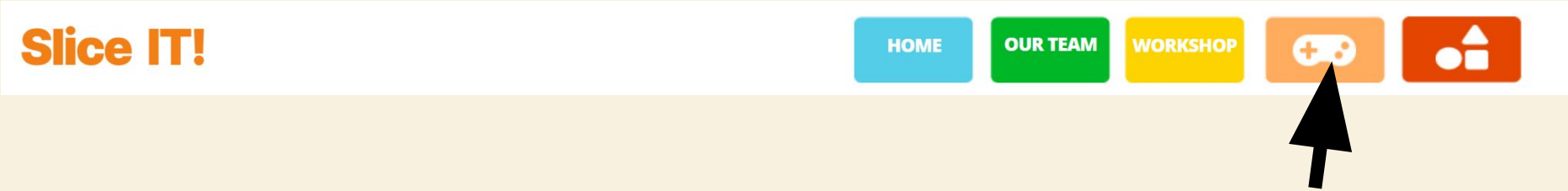
Question 3

Which of the following shapes will make the statement “(color is green) AND (color is blue)”?

- ❑ Green Cylinder
- ❑ Blue Cube
- ❑ White Cylinder
- ❑ Green Sphere
- ❑ Blue Cylinder

Game Time!

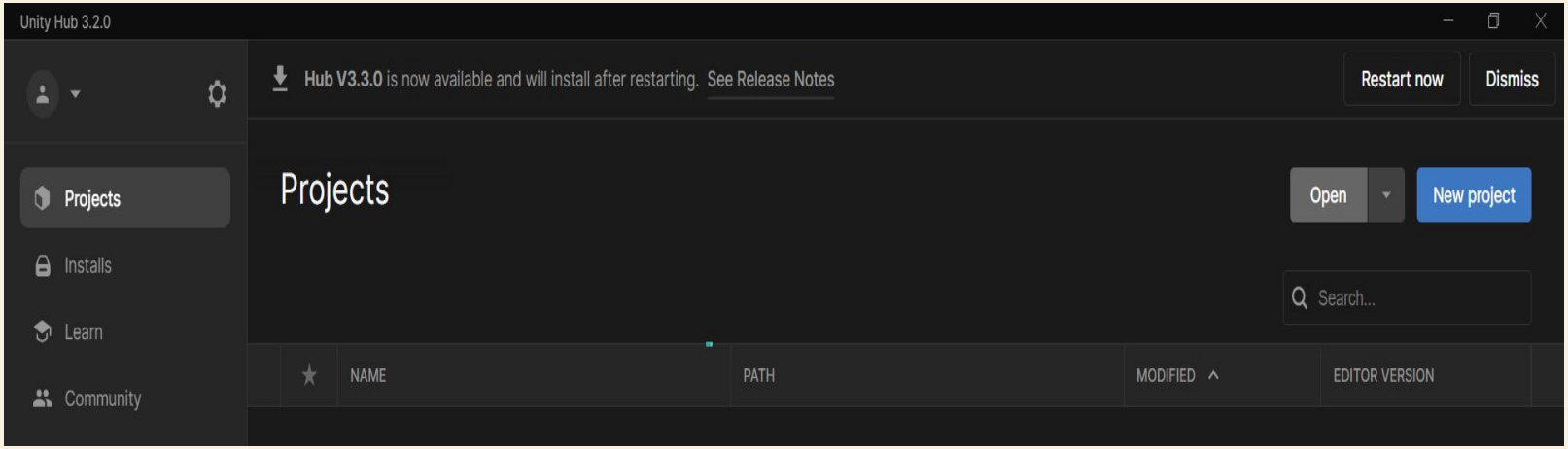
Head over to the game tab to **play**:



Let’s create a level!

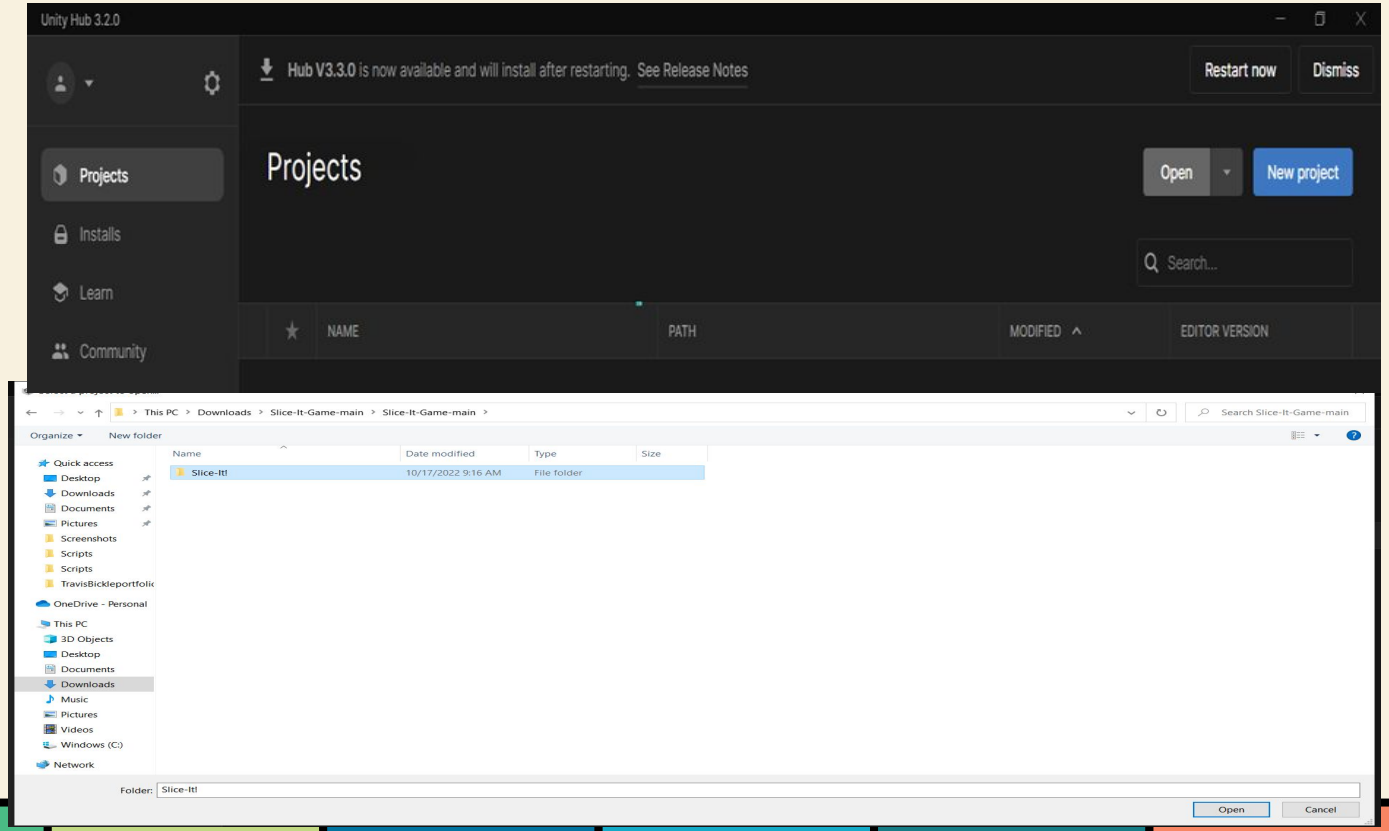
How we made our levels

As we build to game on Unity and can open file on Unity hub.



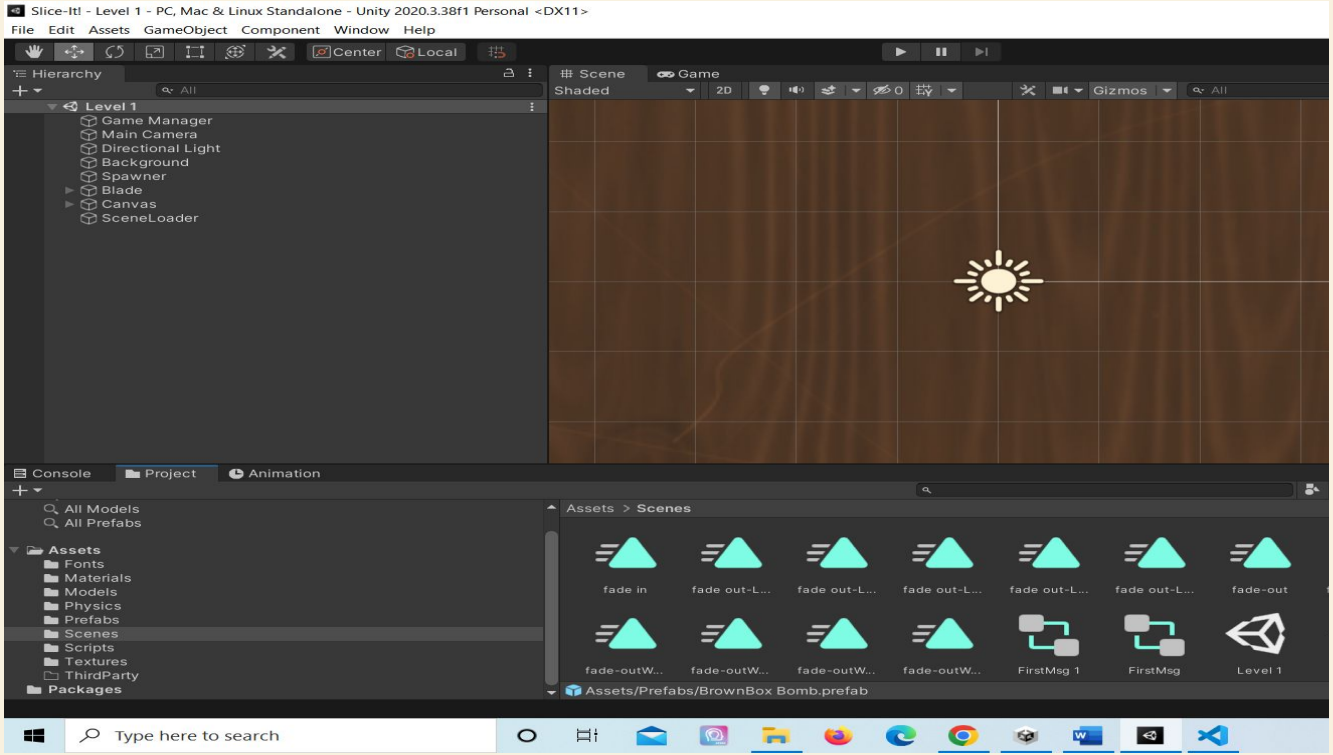
How we made levels

In Unity Hub we hit the open button. Find the Slice It folder, then click it into you reach Slice-It! and then hit open



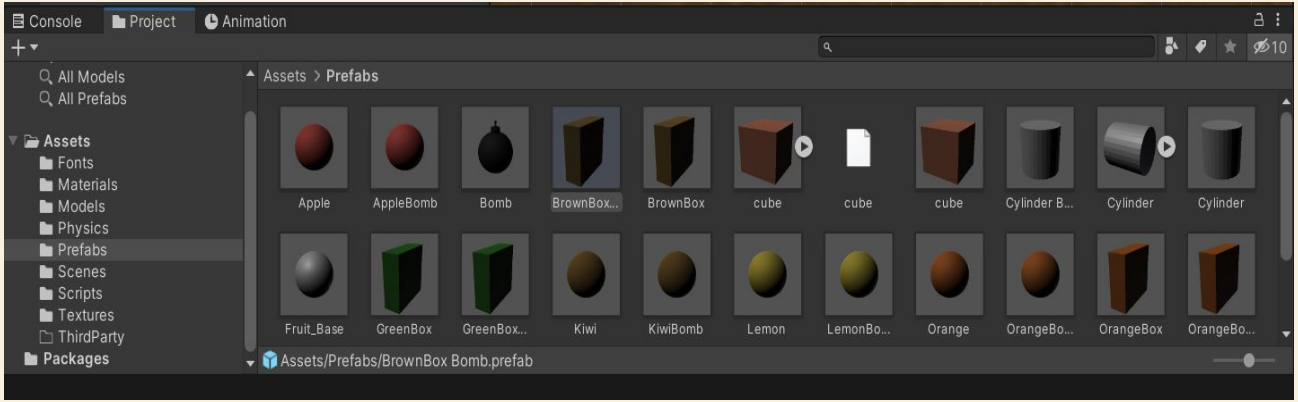
How we made levels

Check all of the scripts, prefabs, and there should be one scene named Level 1



How we made levels

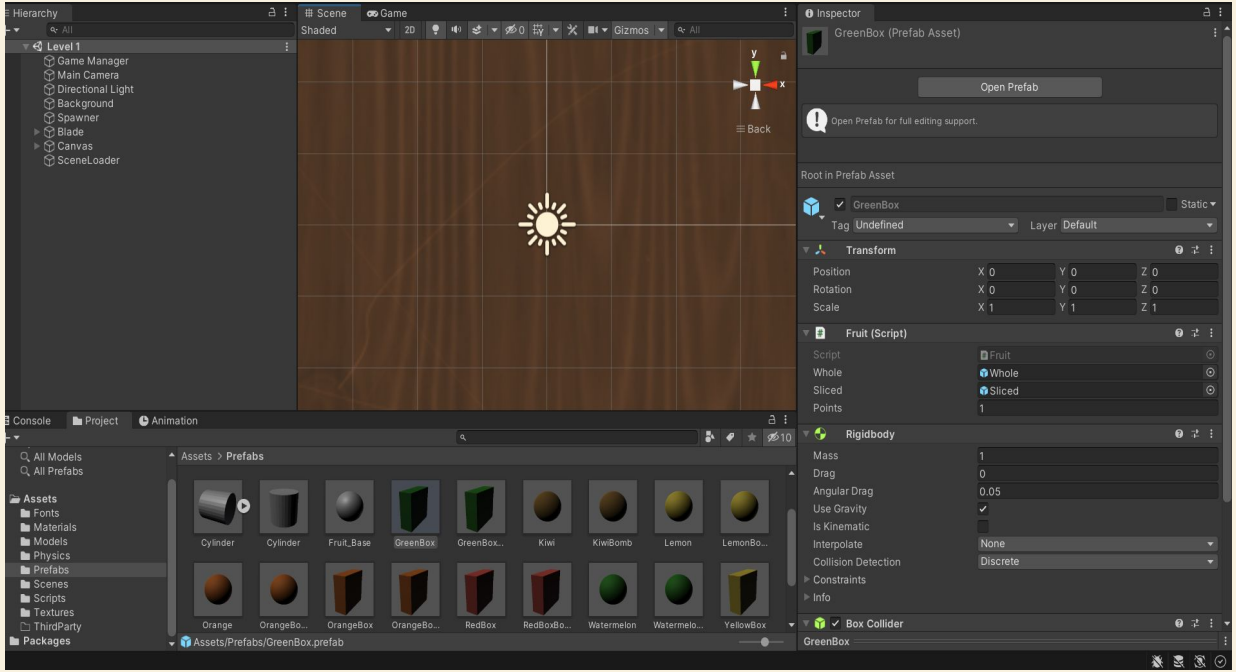
All our prefabs for the game



How we made levels

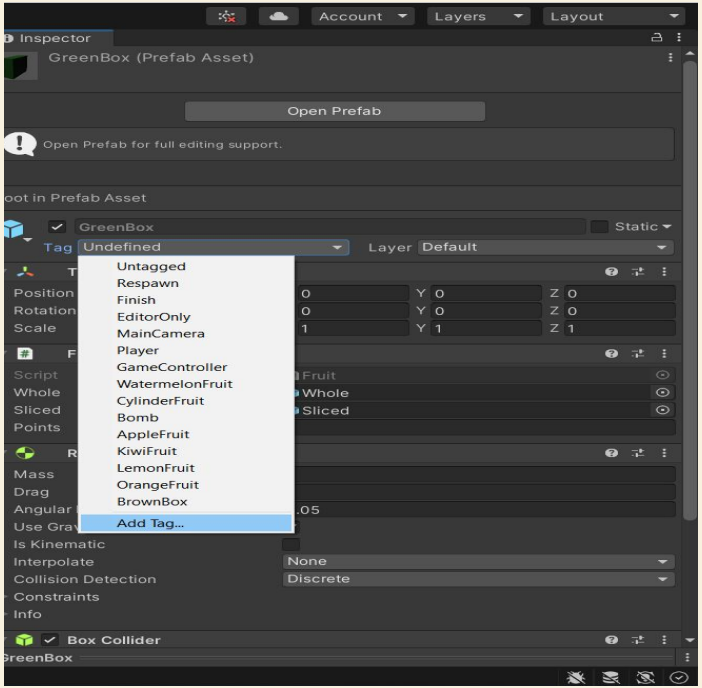
Example:

Clicked on the “GreenBox” prefab



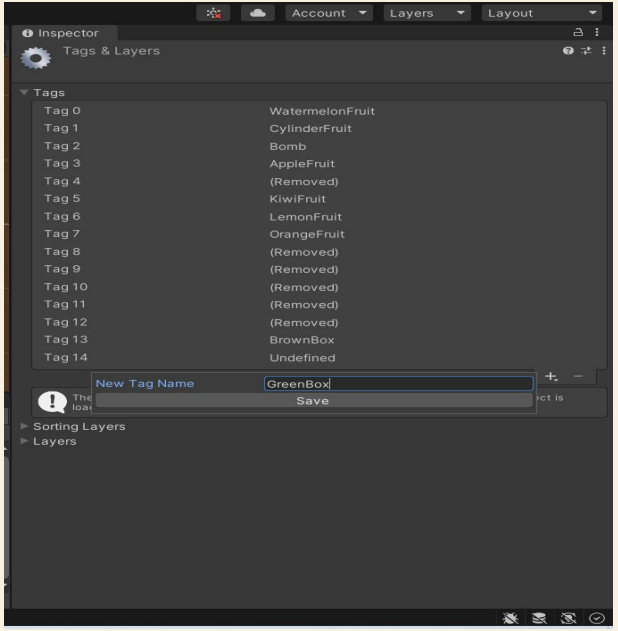
How we made levels

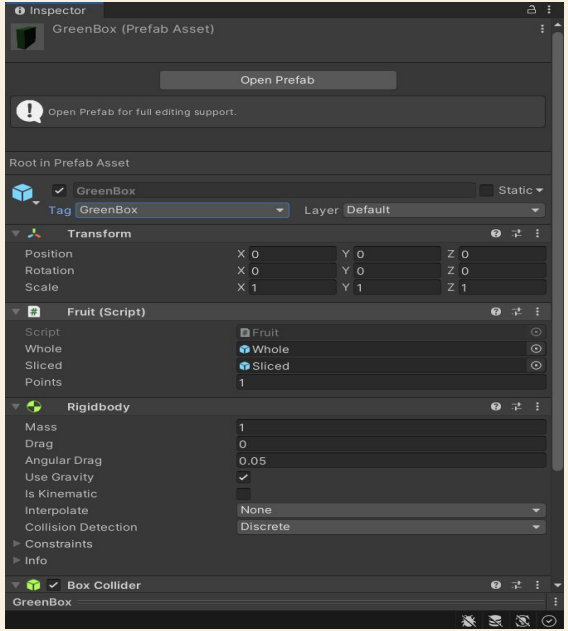
On the tag icon, clicked the add tag sign.



How we made levels

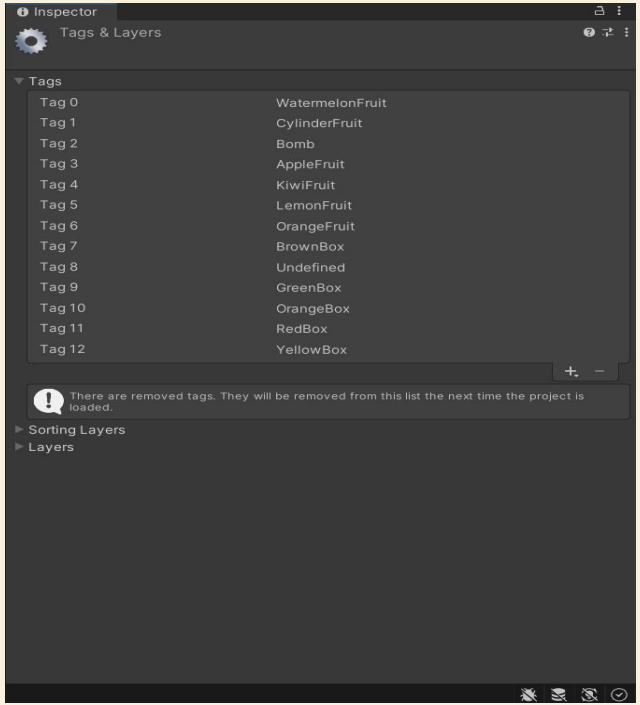
Name this new tag “GreenBox” and go back the it’s prefab and put the tag in the asset.





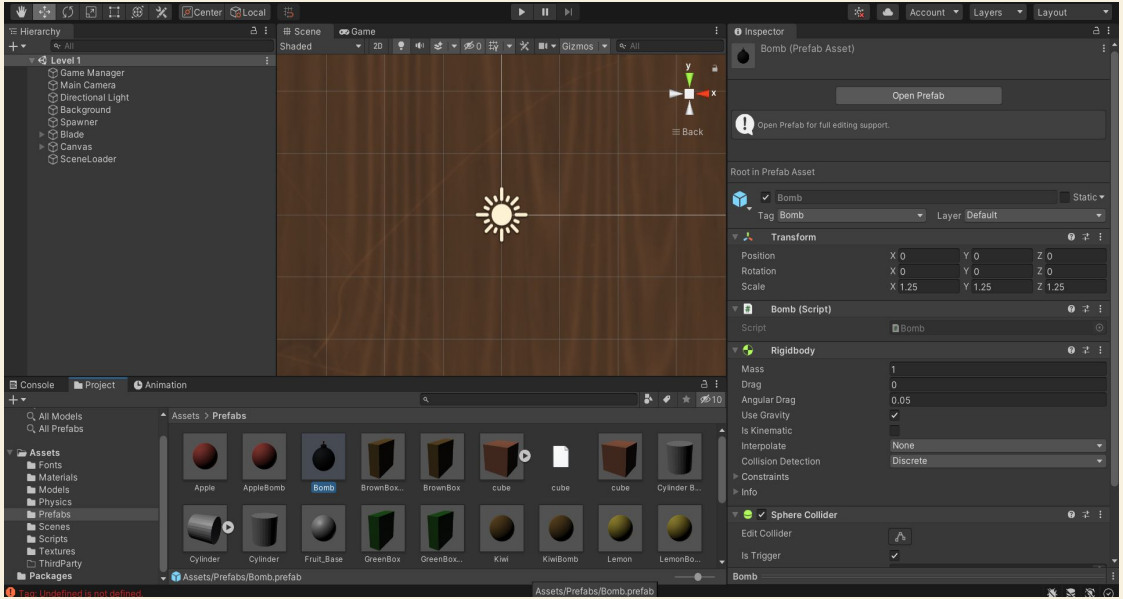
How we made levels

Now repeat this process to all the other prefabs, except the bombs. For reference it's was helpful to name the tag after the color and shape of the prefab



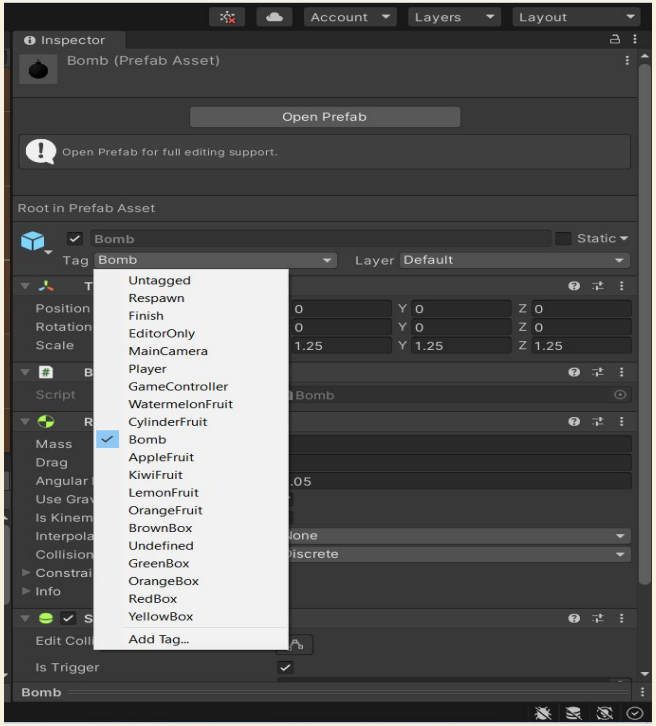
How we made levels

Then we clicked on the bomb prefab



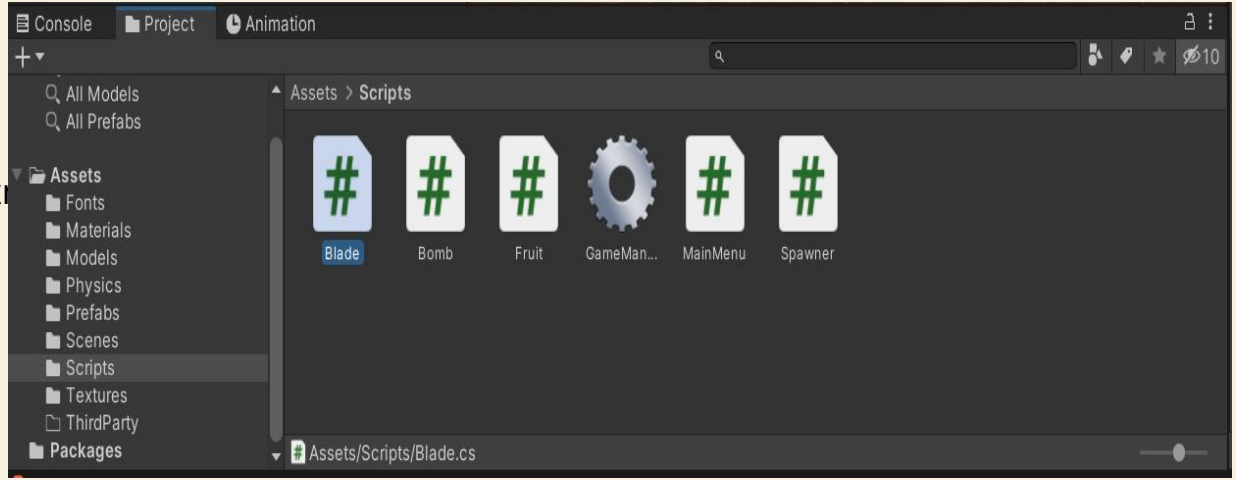
How we made levels

As same as before clicked the tag icon and the plus sign and type “Bomb” as the new tag.



How we made levels

Then we went to the scripts folder, and clicked the blade scr



How we made levels

Line 19 it’s where we first start to create some code.

Blade.cs

C: > Users > Joseph > Downloads > Slice-It-Game-main (Workshop copy) > Slice-It-Game-main > Slice-It! > Assets > Scripts > Blade.cs

```
1  using UnityEngine;
2  using UnityEngine.UI;
3  using UnityEngine.SceneManagement;
4
5  public class Blade : MonoBehaviour
6  {
7      public Vector3 direction { get; private set; }
8
9      private Camera mainCamera;
10
11     private Collider sliceCollider;
12     private TrailRenderer sliceTrail;
13
14     public float sliceForce = 5f;
15     public float minSliceVelocity = 0.01f;
16
17     private bool slicing;
18     //Can edit
19
20     //All shapes (fruits)
21
22     private Text WinMsg;
23     private Text LoseMsg;
24
```

How we made levels

```
18    //Can edit
19    public int GreenBoxCount = 0;
```

Create a “public int” object within the scripts and put a prefabs name with count added. Then we put an equal sign with a o next to it, finally put a semicolon to make it a statement. The photo is an example for reference.

How we made levels

Do the same of all other prefabs, and remember to use the same objects and statements to avoid confusion.

```
//Can edit
public int GreenBoxCount = 0;
public int WatermelonCount = 0;
public int BrownBoxCount = 0;
public int KiwiCount = 0;
public int LemonCount = 0;
public int AppleCount = 0;
public int OrangeCount = 0;
public int OrangeBoxCount = 0;
public int RedBoxCount = 0;
public int YellowBoxCount = 0;
public int CylinderCount = 0;
```

How we made levels

In the blade script go to line 98. Where we are going to type “private void OnTriggerEnter(Collider other){}”.

```
97  // Can edit
98      private void OnTriggerEnter(Collider other)
99      {
100
101
102     }
```

How made levels

Make space between the method brackets. Then take one of your tags inside this statement. Which is “if (other.gameObject.tag == “GreenBox”){}.

```
if (other.gameObject.tag == "GreenBox"){}
```

How we made levels

Inside of the if statement bracket put in the count object in the beginning of code by an add increment of one. Example: shown in the photo.

```
if (other.gameObject.tag == "GreenBox"){
    GreenBoxCount++;
}
```

How we made level

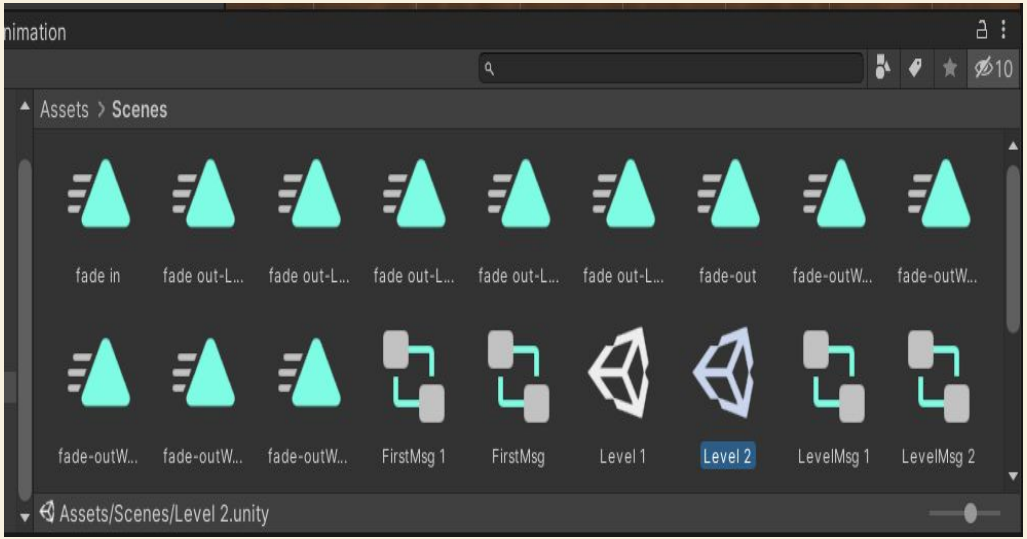
Do the same to all other prefab counts.

```
if (other.gameObject.tag == "GreenBox"){
    GreenBoxCount++;
}
if(other.gameObject.tag == "WatermelonFruit"){
    WatermelonCount++;
}
if(other.gameObject.tag == "AppleFruit"){
    AppleCount++;
}
if(other.gameObject.tag == "CylinderFruit"){
    CylinderCount++;
}
if(other.gameObject.tag == "BrownBoxFruit"){
    BrownBoxCount++;
}

if(other.gameObject.tag == "KiwiFruit"){
    KiwiCount++;
}
if(other.gameObject.tag == "LemonFruit"){
    LemonCount++;
}
if(other.gameObject.tag == "OrangeFruit"){
    OrangeCount++;
}
if(other.gameObject.tag == "OrangeBoxFruit"){
    OrangeBoxCount++;
}
if(other.gameObject.tag == "RedBoxFruit"){
    RedBoxCount++;
}
if(other.gameObject.tag == "YellowBoxFruit"){
    YellowBoxCount++;
}
```

How we made levels

We went back to the game main and go to scenes. Then click on level 1 and press CTRL + D to duplicate the scene. That's the easy way to make multiple scene with the same format in the Level 1 scene.



How we made levels

Go back to the blade script and where going to make the condition to transfer different level if the player met the required task.

```
//Level One
if(BoxCount + AppleCount + WatermelonCount>24){
    WinMsg.text = "You Win!";

}
```

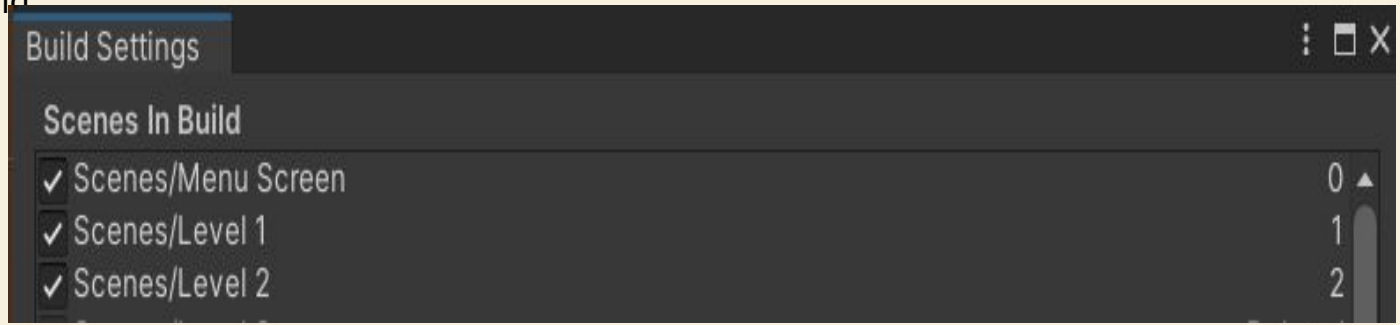
How we made levels

Add UnityEngine.SceneManagement just above the public class method.

```
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
```

How we made levels

Go back to the unity game menu to click on file, then click on build setting. There you will see the index or order of the scene the game is processing.



How we made Levels

Under the box count increment codes we will type in a condition or rather a task to make to advance to the next level. We're use an example from the photo. "SceneManager.LoadScene(SceneManager.GetActiveScene().build index + 1;" will transfer the games into the next scene which should be level 2.

```
//Level One
if(BoxCount + AppleCount + WatermelonCount>24){
    WinMsg.text = "You Win!";
    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
}
```

How we made levels

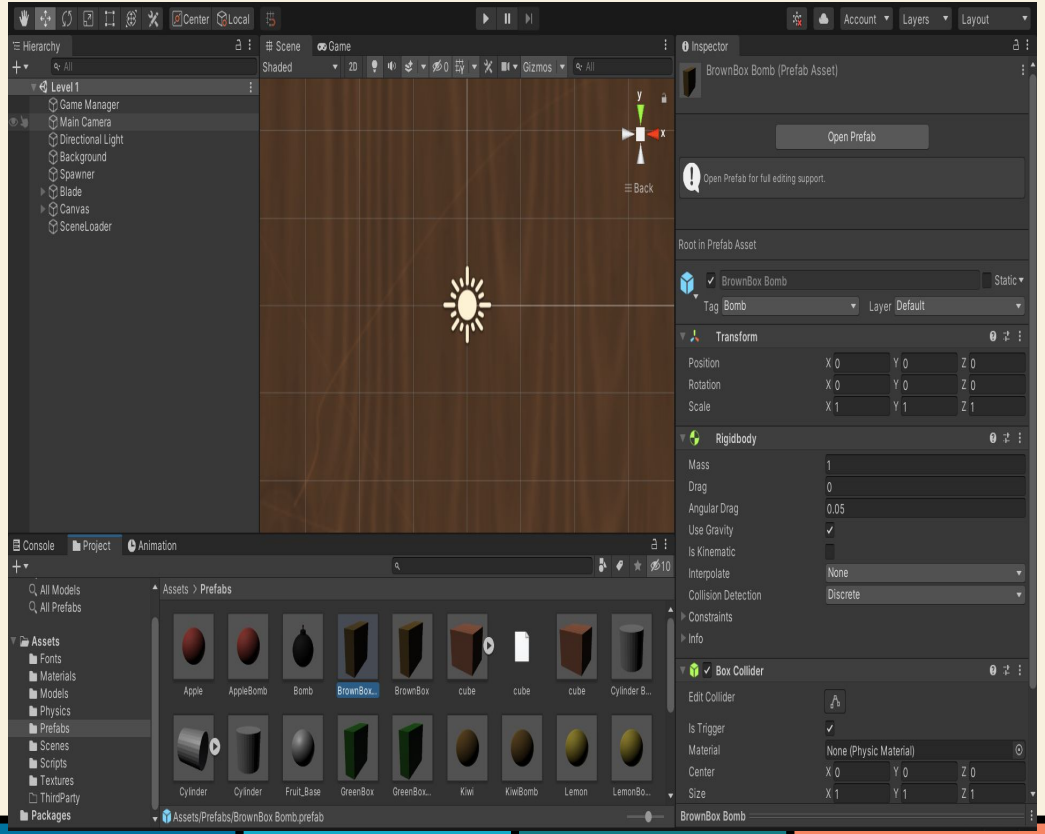
Below the if statement for level one make an else if statement where a player hits a bomb. Which should revert all prefab counts to zero.

```
else if(other.gameObject.tag == "Bomb"){
    BoxCount = 0;
    WatermelonCount = 0;
    OrangeBoxCount = 0;
    OrangeCount= 0 ;
    BrownBoxCount =0;
    KiwiCount = 0;
    RedBoxCount = 0;
    YellowBoxCount = 0;
    CylinderCount = 0;
    AppleCount = 0;
    LemonCount = 0;

}
```

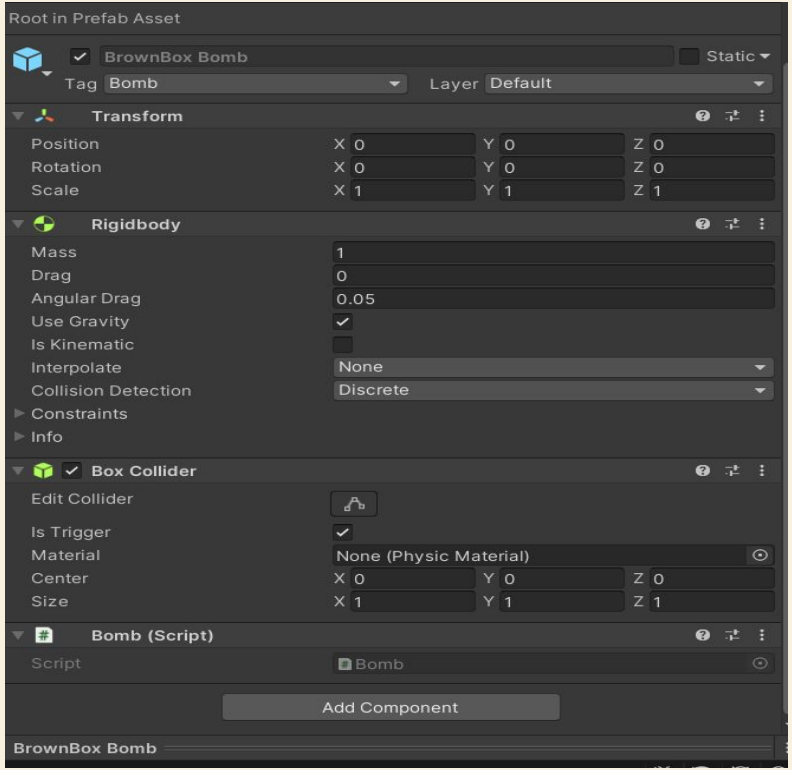
How we made levels

To make fruits as bomb first hit CTRL + D to duplicate them. Then click on the tag icon and put a bomb tag to them.



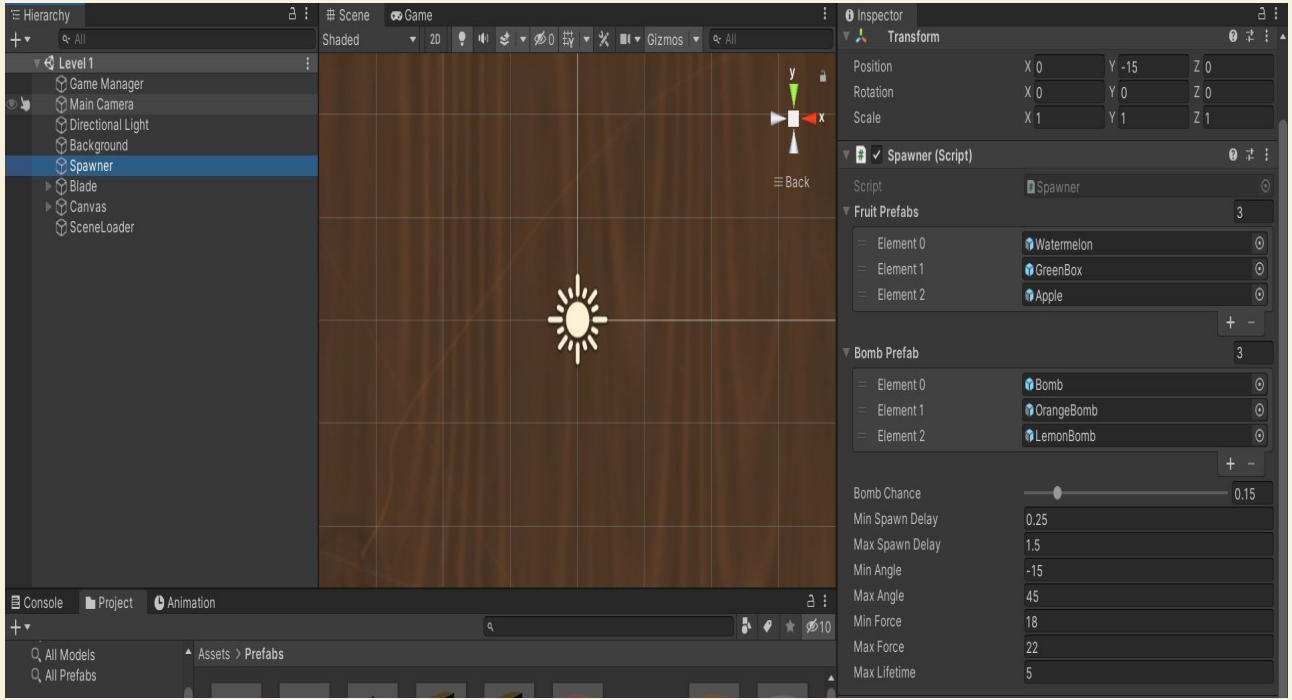
How we made levels

Remove the fruit scripts from them and instead replace it with a bomb scripts which will make it act like a bomb.



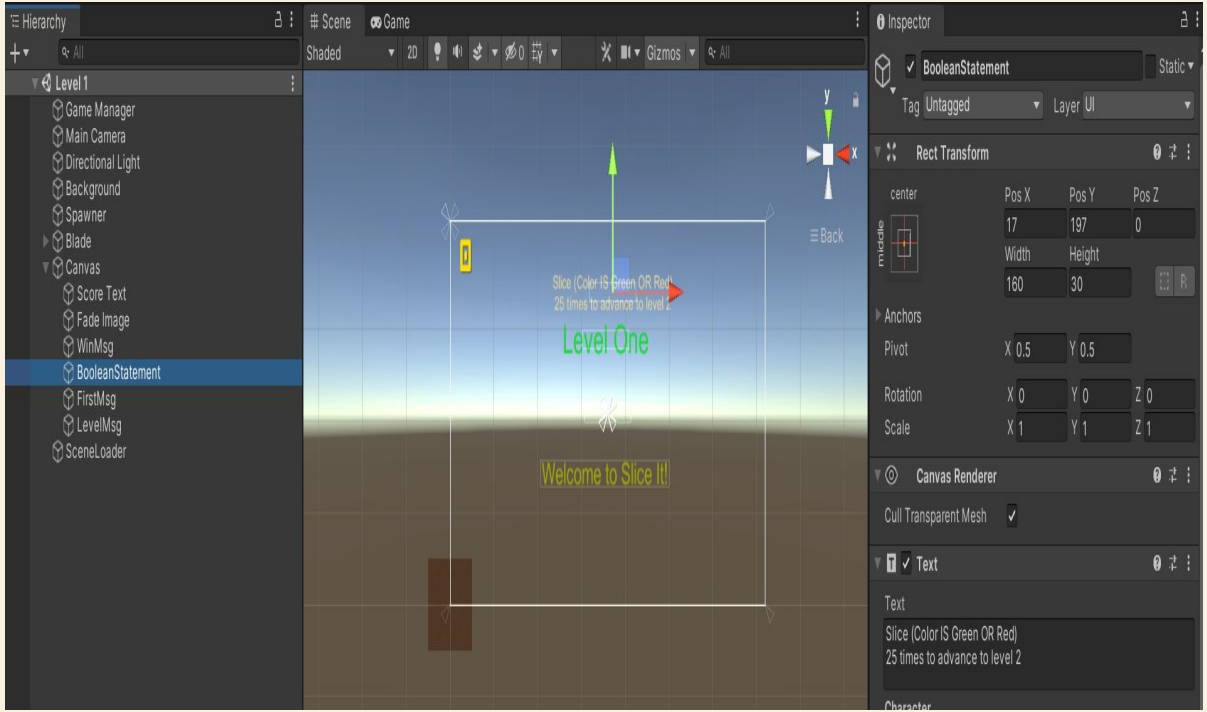
How we made levels

Click on the spawner object and clicked on the prefabs folder. Then drag and dropped the appropriate prefabs object in the fruit array and the bomb array. There you can manuiplate the chances of bombs for difficulty.



How we made levels

Click the canvas object and duplicate either the “Firstmsg” or “Levelmsg” to made a boolean statement message. This will help not only you but other player on what to do in the game.



It is Done!

You've create a level in Unity.

Since Unity is a complex application there will be times where it can get frustrating, but remember with resilience and dedication anyone can make a good game.

Any questions?

We would love to hear your feedback!

- Please if you have the time go do our post survey, as it gives us valuable information on how we did, do this by clicking on the link below: **https://ggc.az1.qualtrics.com/jfe/form/SV_8IAyAgqFH1SnyJ0**