

King Quest

PROJECT SUBMITTED TO ASIAN SCHOOL OF MEDIA STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
AWARD OF DEGREE OF

B. Sc.
in
Game Design & Development

By
Ragiv Chahal
(University Enroll. No: 2018070325)

Under the Supervision of

Prof. Ashish Garg



**ASIAN SCHOOL OF MEDIA STUDIES
NOIDA**

2021

DECLARATION

I, **Ragiv Chahal**, S/O **Lokendra Kumar**, declare that my project entitled "**King Quest**", submitted at School of Animation, Asian School of Media Studies, Film City, Noida, for the award of B. Sc. In Game Design & Development, Shobhit University and Graduate Diploma in Game Design & Development, ASMS, is an original work and no similar work has been done in India anywhere else to the best of my knowledge and belief.

This project has not been previously submitted for any other degree of this or any other University/Institute.



Signature

Ragiv Chahal
raghav.wolf.33@gmail.com
9557566518

B. Sc. In Game Design & Development
School of Animation
Asian School of Media Studies

ACKNOWLEDGEMENT

The completion of the project titled **King Quest**, gives me an opportunity to convey my gratitude to all those who helped to complete this project successfully. I express special thanks:

- To **Prof. Sandeep Marwah**, President, Asian School of Media Studies, who has been a source of perpetual inspiration throughout this project.
- To **Mr. Ashish Garg**, Director and Professor of School of Animation for your valuable guidance, consistent encouragement, advice, timely suggestions and support. I deeply value your guidance.
- To my **friends** for their insightful comments on early drafts and for being my worst critic. You are all the light that shows me the way.

To all the people who have directly or indirectly contributed to the writing of this thesis, but their names have not been mentioned here.

Signature

Ragiv Chahal
raghav.wolf.33@gmail.com
9557566518

B. Sc. In Game Design & Development
School of Animation
Asian School of Media Studies

<u>Content</u>	<u>Page No.</u>
About the project	6
Objectives	7
Unit 1: Pre – production	8-11
● One-Pager	8
● Game Idea and Assets	9
● Game mechanism and Flowchart	10
● Mood Board	11
Unit 2: Production	12-79
● Project Setup	12-13
● Tilemap Setup	14-19
● Level Design Step By Step	20-23
● Making Prefab Brushes	24-28
● Player Setup	29-30
● Player Sprites and Animation	31-32
● Making Player Moveable	33-34
● Setup Player Parameter and Animation	34-46
● Player Scripts (Inspector View)	47-49
● Enemy Setup (Type: Melee Attack)	50-52
● Setup Enemy Animation	53-57
● Enemy Setup (Type: Object Throwble)	58-61
● Enemy Setup (Type: Shooting Enemies)	62-63

● Game collectibles	64
● Traps	64
● Game Camera	65
● Game HUD	66-67
● Main Screen	68-69
● Option Screen	70-71
● All Scripts	72-78
● Development Log	79-81
Conclusion (Learning Outcome)	82-83
Bibliography	84

About the Project

This Project is based on Game Design and Development. In this project, I am designing and developing the game from scratch with the help of different softwares. Using Unity Game engine for building the game and for coding using visual studio with C Sharp language.

This is a 2d, Platform game with the pixel art style. All the pixel art is based on 32 by 32 pixels for this game. For the art I am downloading the royalty free assets from the site and also use pixel software for some art. It's free and has some of the best pixel art options. This game is an Action and Adventure type game.

The Main Protagonist of the game is a Human King. He is a Kind and Powerful, And now he has to fight and take down all the enemies and take back the orb they steal from the human king.

The Audio is Retro and Simple for this game it suits the most. I am downloading the royalty free audio and sfx sound from the sites and for editing sound using Audacity. It's simple and easy to use and also free to use.

The target platform of the game right now is Window. Because It's the default setting of the unity game engine and the most suitable option for the starting phase.

Objectives

- Making the Fully functional 2d game with the pixel art
- The Unity engine has many features for 2d games. I want to explore all the features to understand unity more and make myself comfortable with unity.
- Learning More about Unity game engine Speciality in 2d how it works with the pixel or pixel art.
- Learn more about 2d animation with the sprite sheet and how to use animation events.
- Learn more about pixel perfect cameras with cinemachine.
- Learning more about Tile Maps, Rule Tile and Unity 2d Extra features like Animated Tiles.
- Making an enemy Ai and state machine to learn more about.
- Making a fully functional independent game with almost all the features.

Unit 1: Pre – production (One-Pager)

King Quest

Game Identity: This game is a 2d platform game. where you can defeated all the enemies appear in level.

Design Pillar: Action, Adventure, Platform

Story: There are two Island One have a kind King Human and another has a powerful King Pig. They both have powerful Orb. But one night King Pig Steal the King Human Orb. Now King Human have a duty to take back our Orb.

Feature: Ability and Skill upgrade

Art Style: 2D, Pixel art

Music: Retro, Simple

Platform: Window

Age: 11+

Game Idea

The game idea for this game comes from the 2d android game name “ Dan The Man”. It's a 2d pixel art game with simple controls and a basic UI design. The game logic is simple, you can defeat the enemies and the boss of the stage and proceed to the next stage. All stages have different types of enemies. The game has some upgradeable skill sets.

From this game I have the idea now how my game looks and how the logic and machines work.

Assets

For the Assets I am downloading the pixel frog assets from the itch.io site and modify them when needed.

Treasure Hunters : <https://pixelfrog-assets.itch.io/treasure-hunters>

Kings and Pigs : <https://pixelfrog-assets.itch.io/kings-and-pigs>

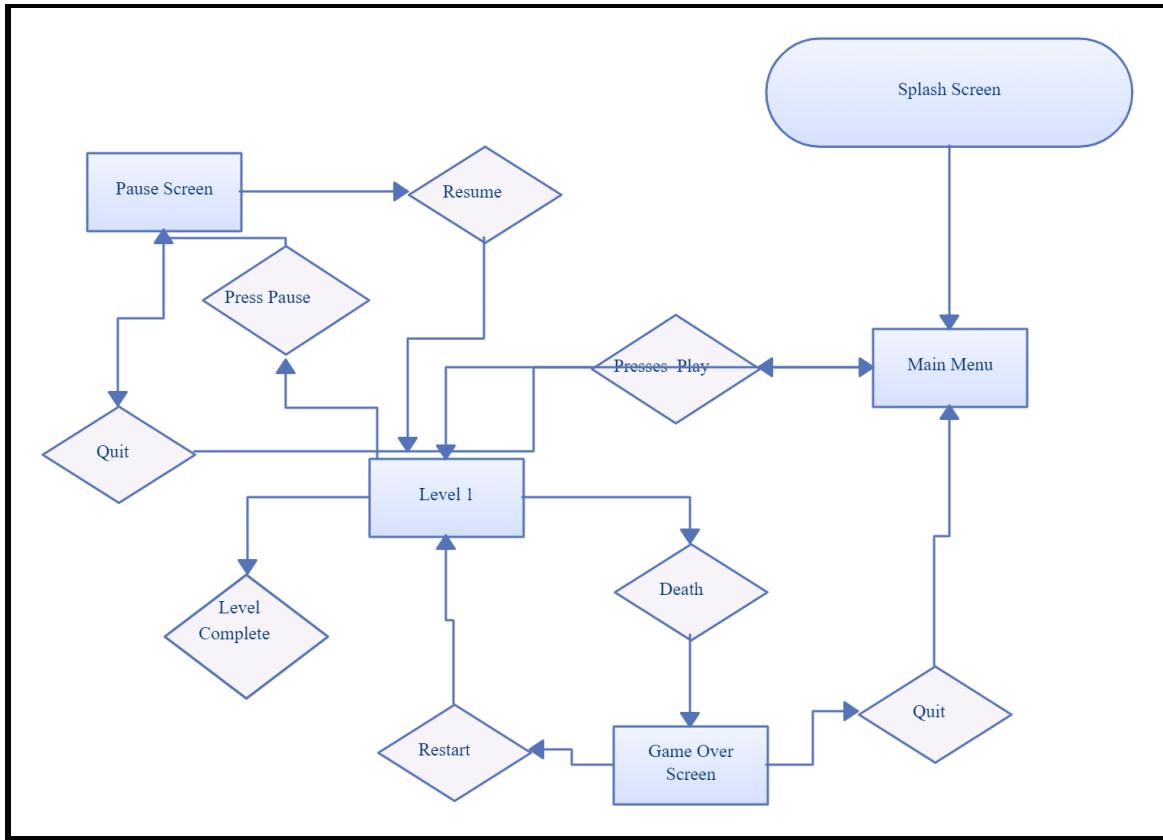
Pixel Adventure: <https://pixelfrog-assets.itch.io/pixel-adventure-1>

For Audio and sfx i am downloading royalty free sound form some sites

<https://www.zapsplat.com/sound-effect-category/game-sounds/>

<https://freesound.org/people/GameAudio/packs/13940/>

Game mechanism or Logic Flow chart



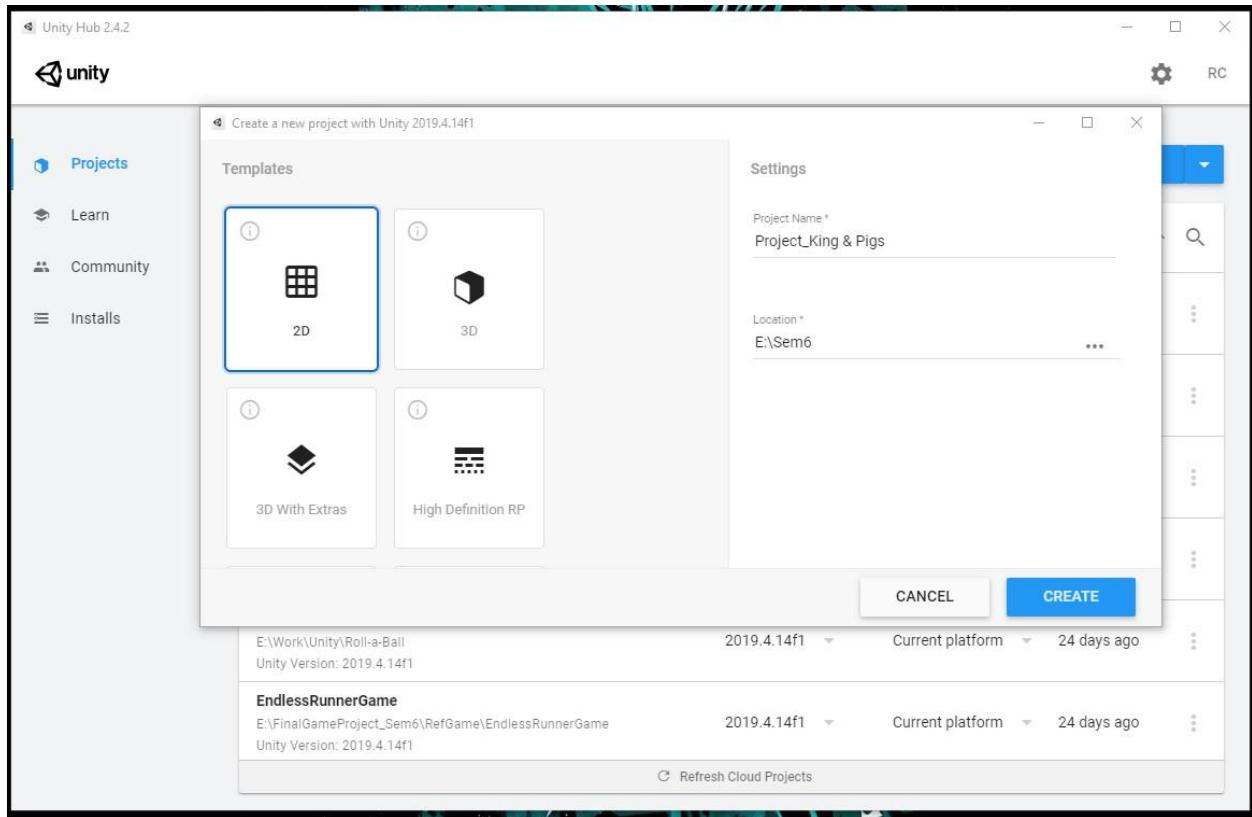
- This is a game logic flow chart. Flow chart is easy to understand and explaining the game mechanism is very useful on the flow chart because visually we understand fast and correct.

Mood Board



Unit 2: Production

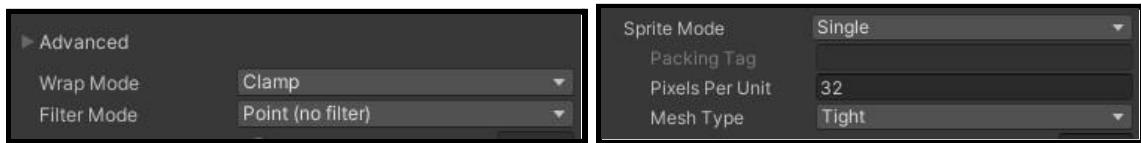
Project Setup



- Open Unity hub, Then click on new and select 2D template and Give a name to a project and select location of the folder.
- Select a 2d template because unity gives us all 2d features and packages.

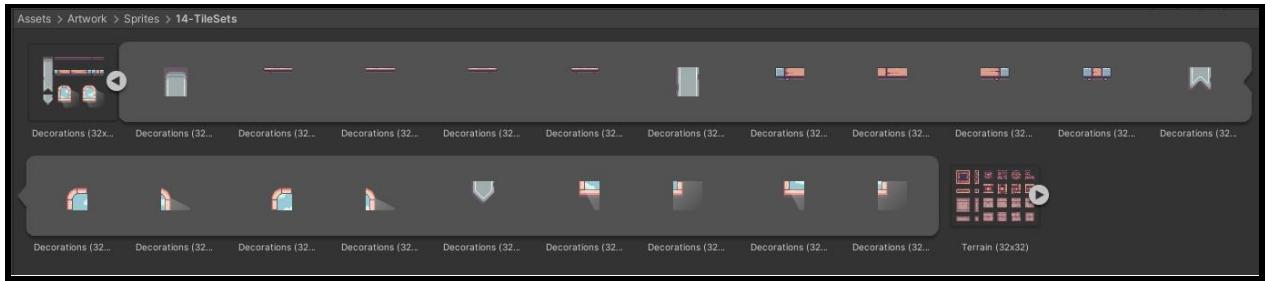


- Creating Separate folders for each different assets like music, Sprites, Scripts.
- Then Import All the Assets in Unity.
- Importing Assets in unity is easy just drag and drop the assets folder system to unity project.
- After Import Assets select all the sprites and change the settings according to the project need.

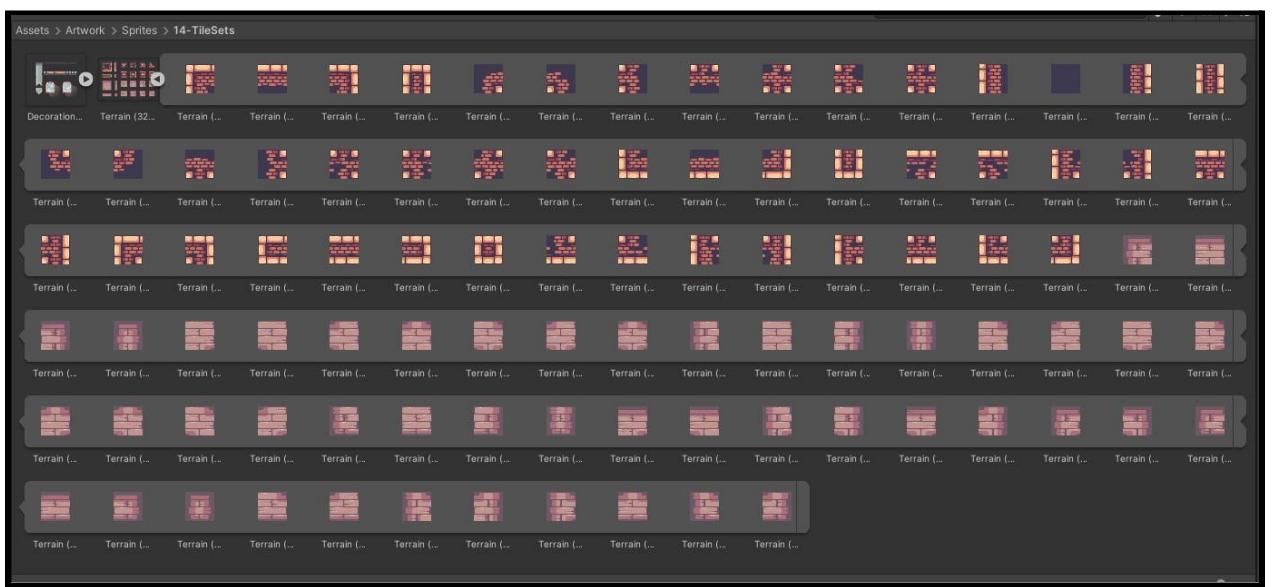


- Change all pixel image filter mode to point and pixel per unity 32 then hit apply.

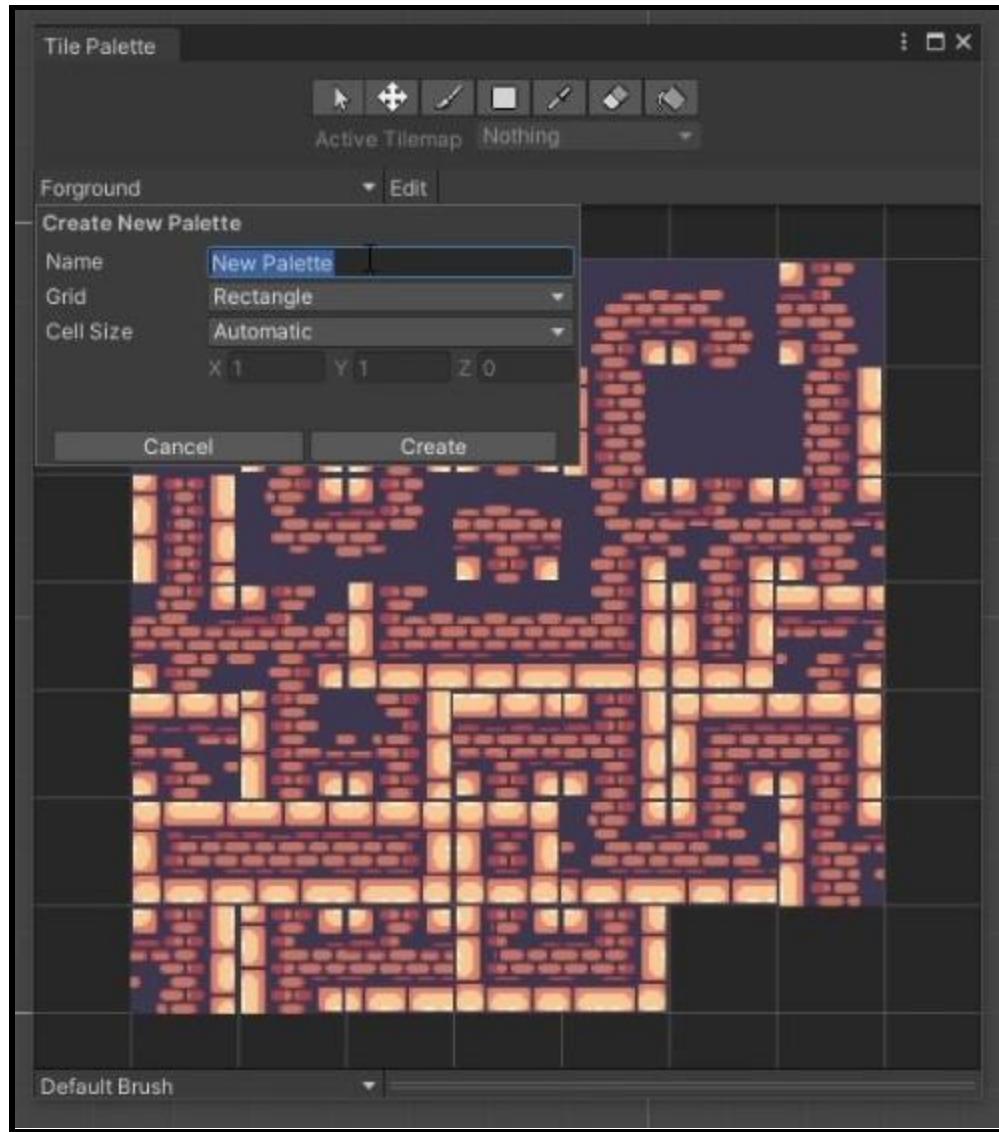
Tilemap Setup



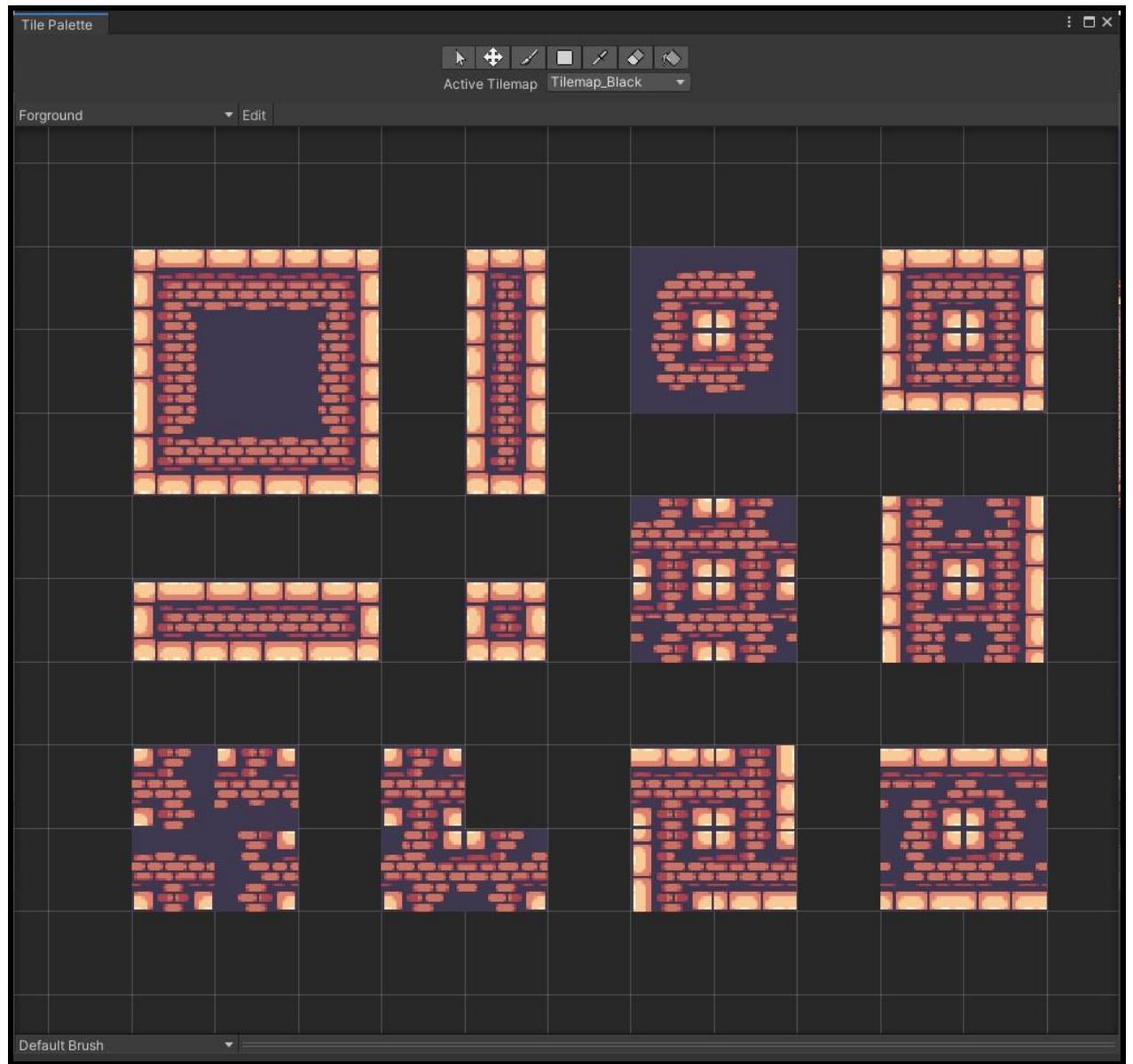
Decorations Tilesets



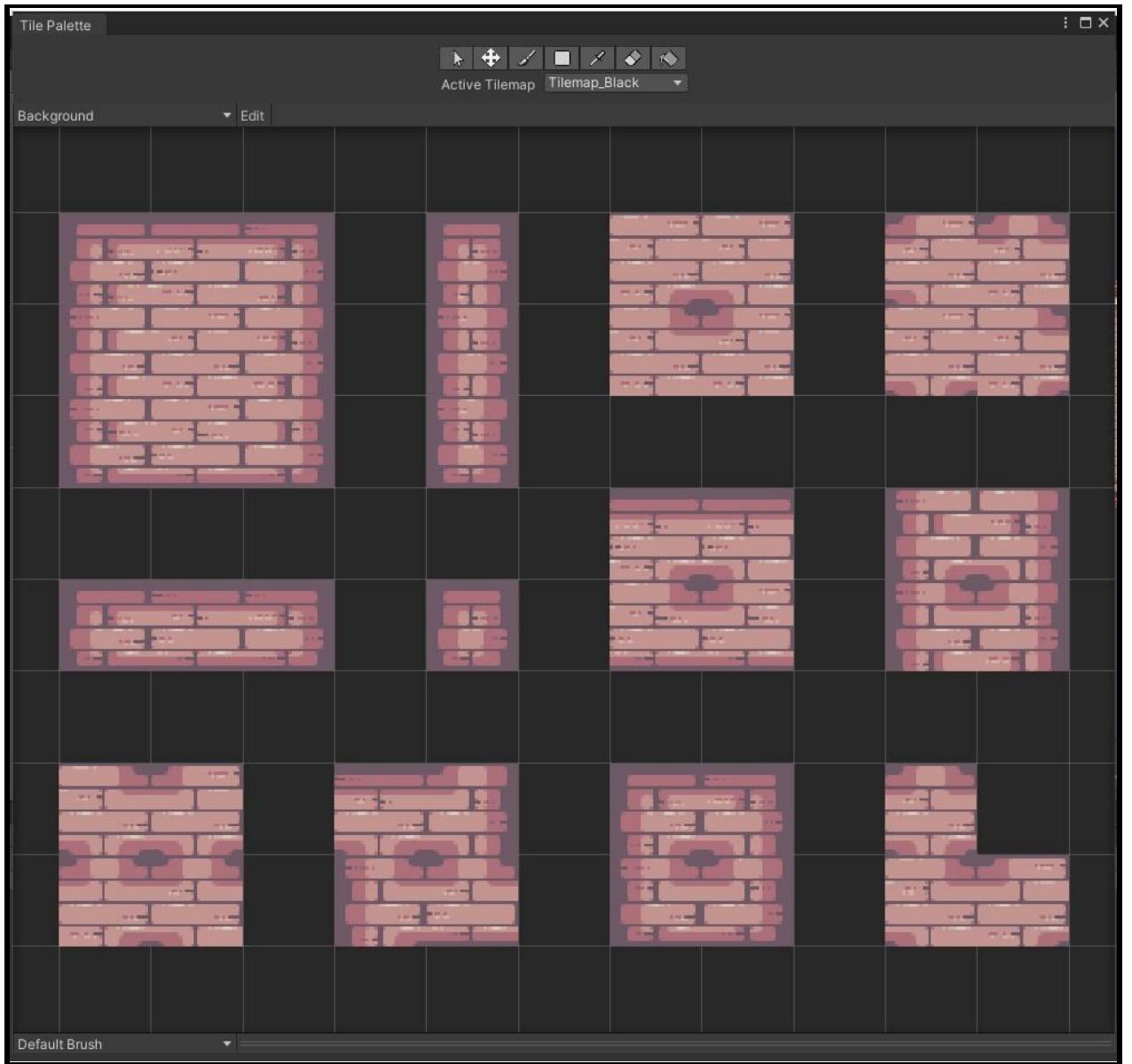
Terrain Tilesets



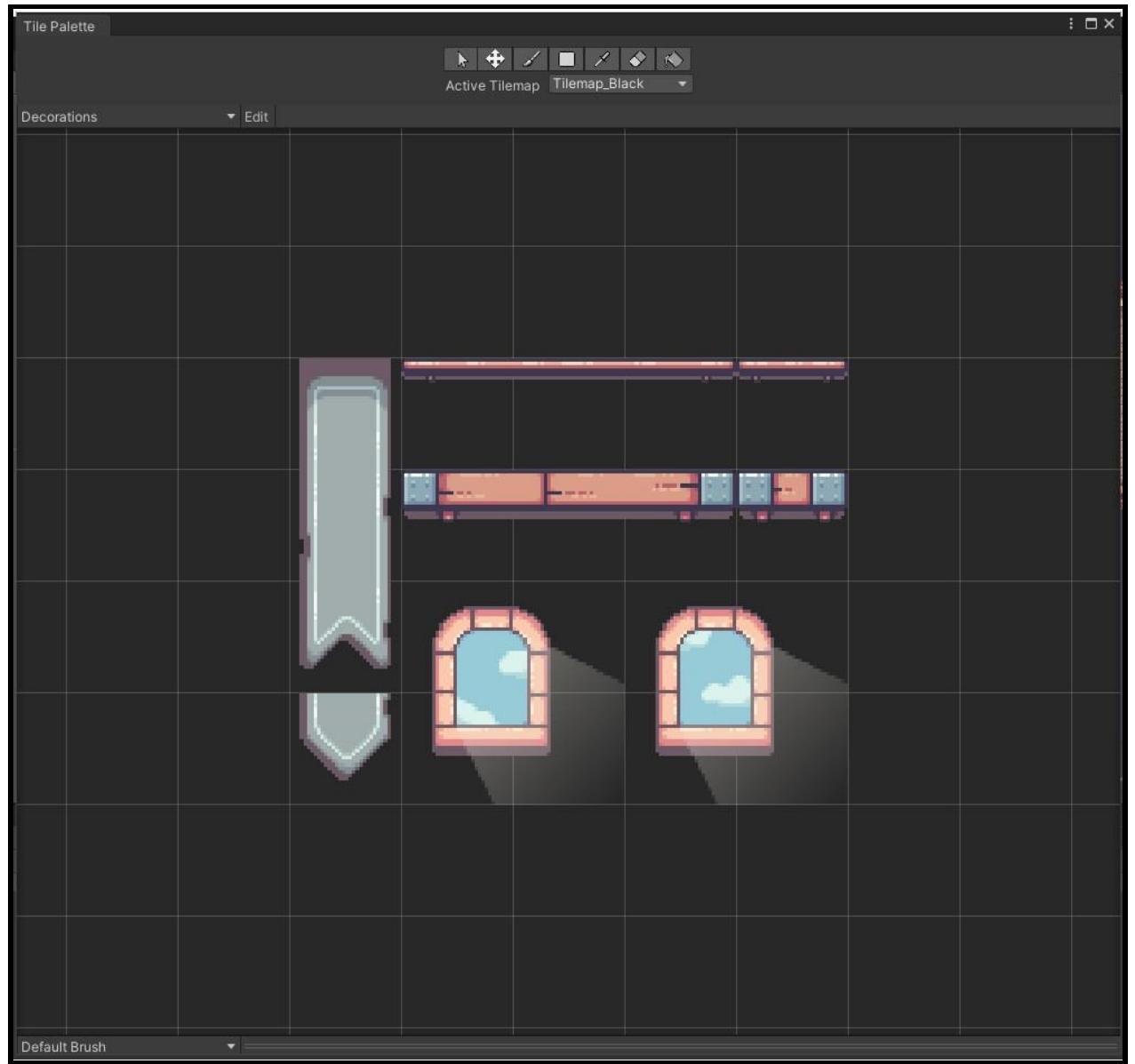
- Creating Palette for tile for use in a game.
- Open TilePalette Window
- Then Click on Create New and give a name to a palette
- Select the folder where you save all the palette. Then click on create.



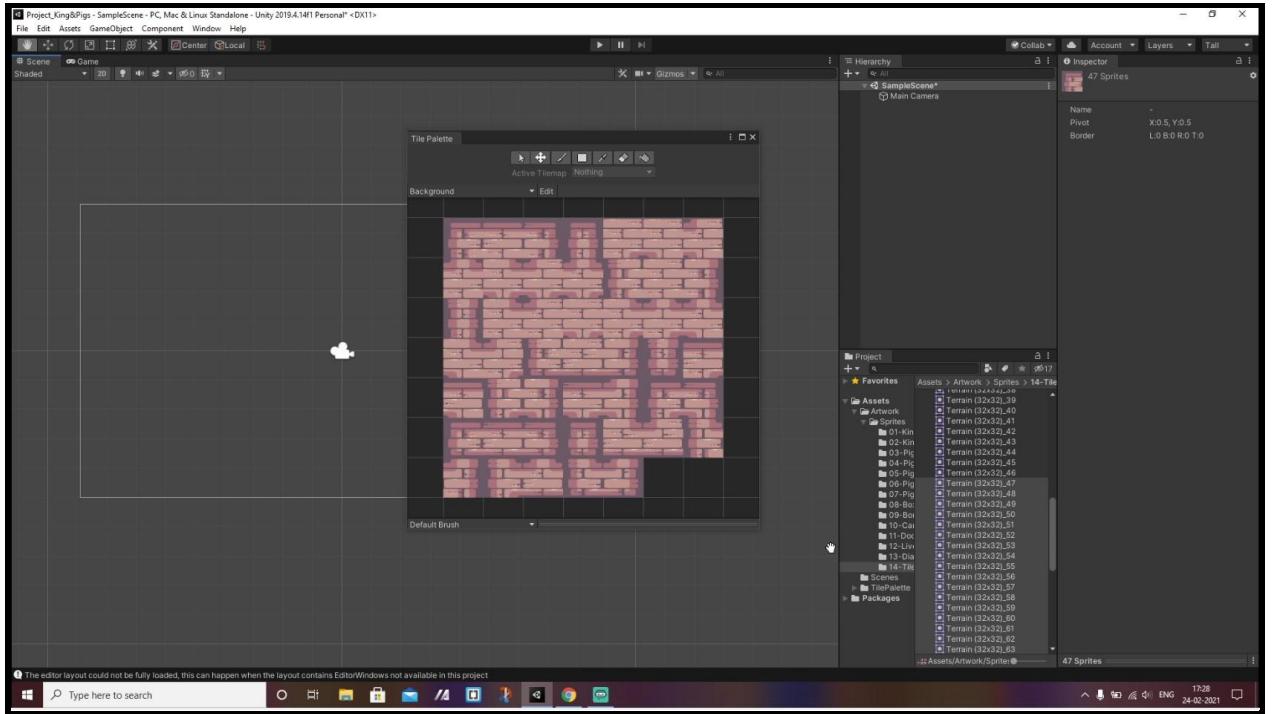
Foreground Tile Palette



Background Tile Palette

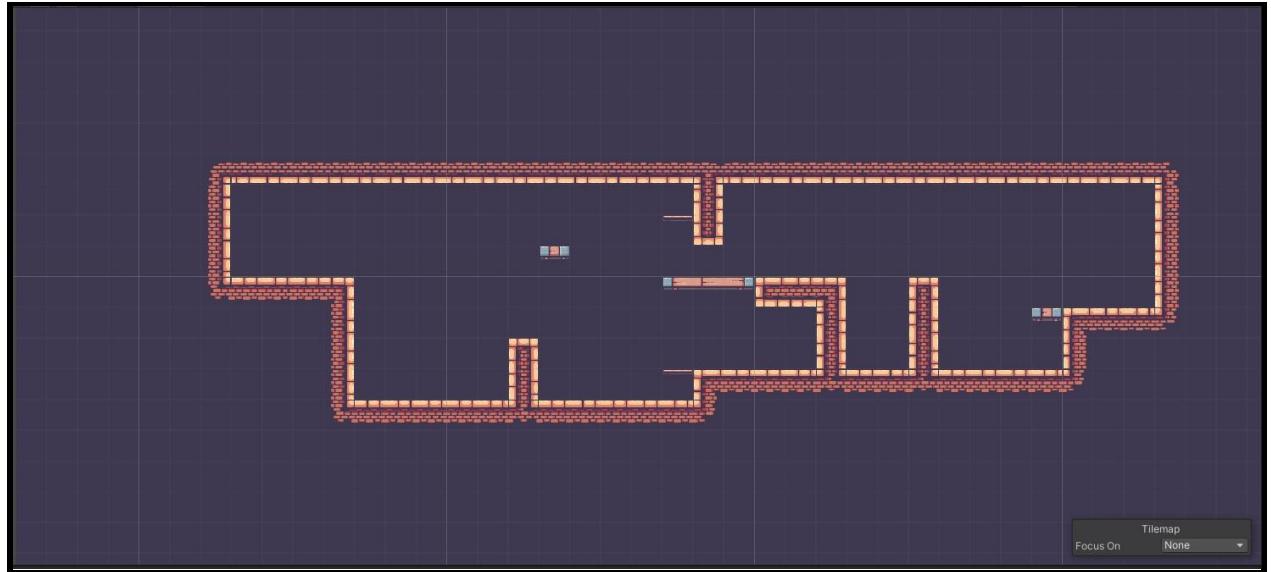


Decorations Tile Palette

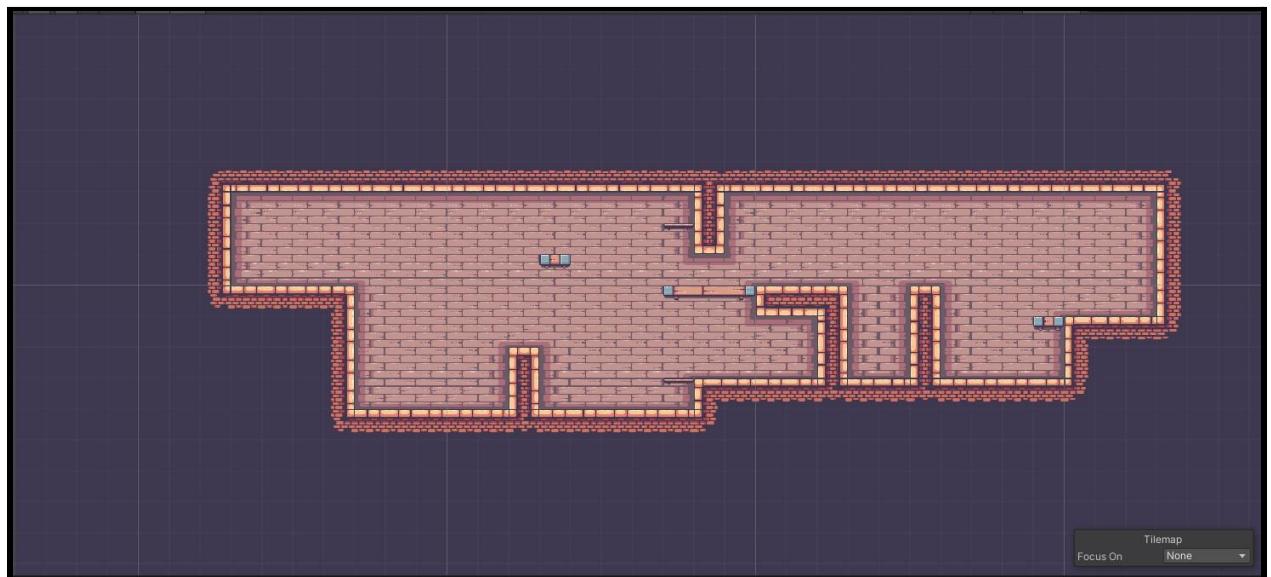


- Import all tiles you want to use in game and then adjust all in the correct way if you want like you see in the upper image.
- It's a bit time consuming work but the result is good.
- When we adjust all the tiles it makes our work easy and we make and place tiles fast and make a level fast.

Level Design Step by Step



Foreground Layer



Background Layer

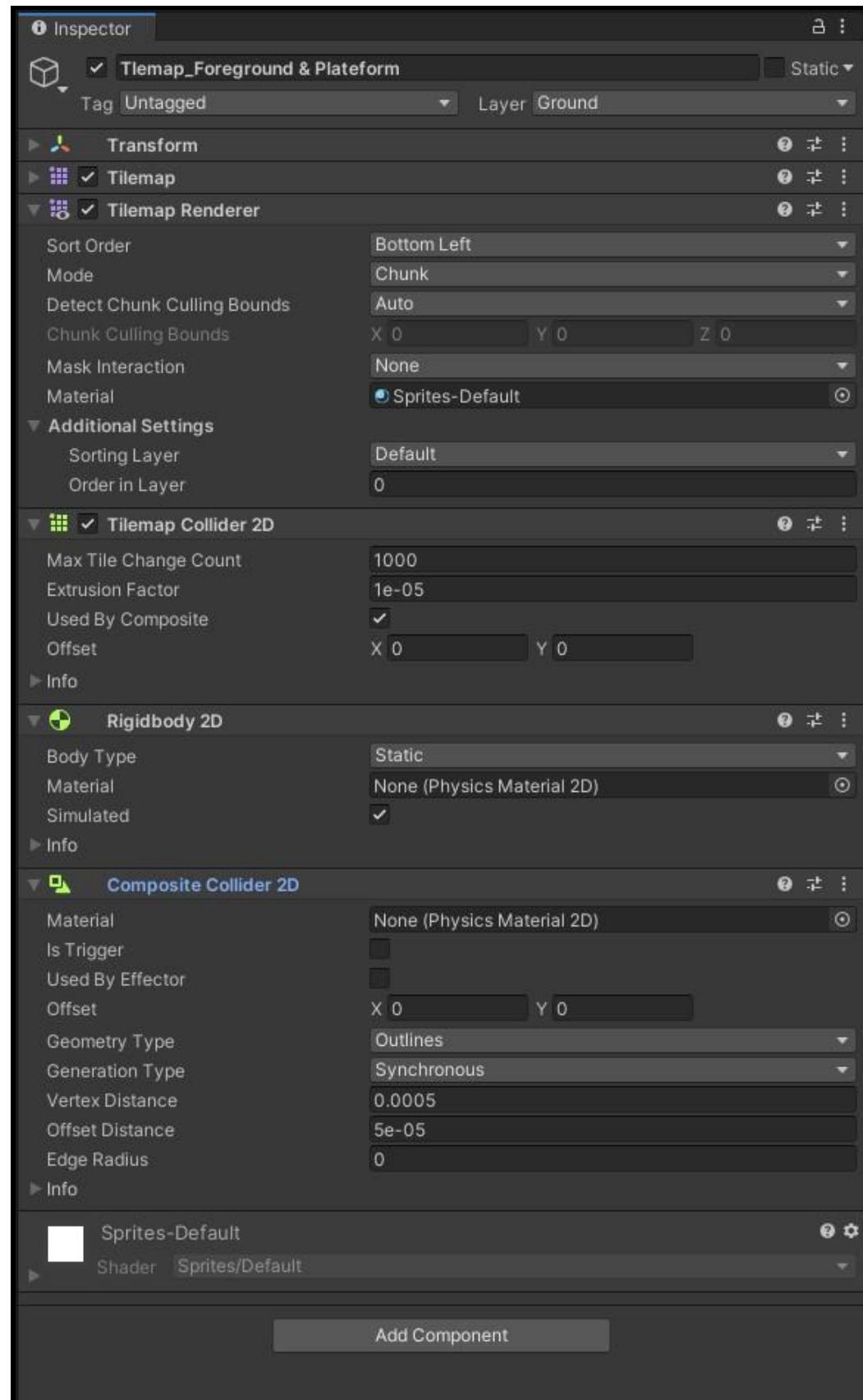


Decoration Layer



Complete Level

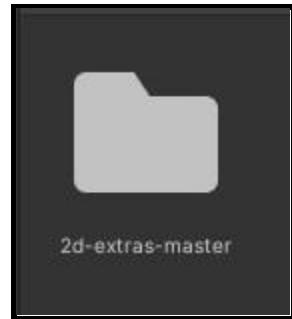
- Creating Level from the help of tile palette
- Use Tile maps to separate each layer of level like Background, Foreground, etc.



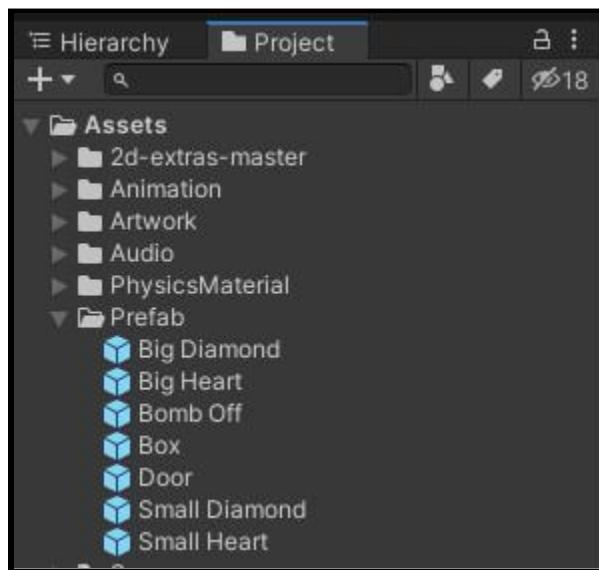
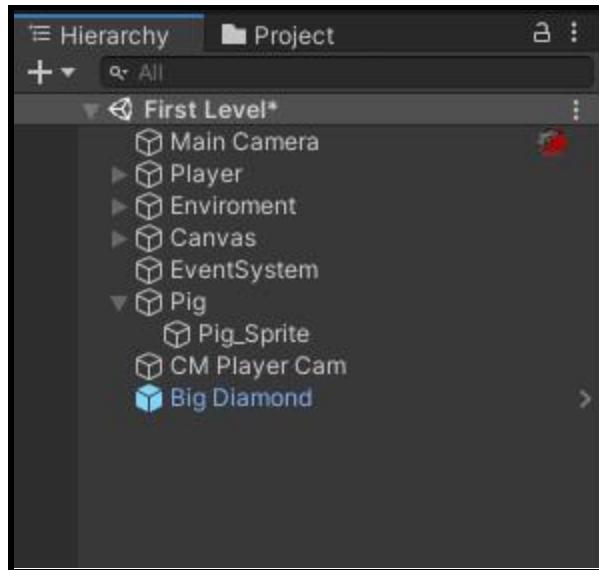


- Add Tilemap Collider 2d In the Ground Layer Tiles and also add Composite 2d After completing the level Design.
- After Adding Collider 2d in the Layer you see the green line around the existing layer like you see the upper image.
- After adding Tilemap collider and composite 2d.
- Click on the composite in Tile Maps collider 2d.
- Add Rigidbody to the layer and Set body type to Static.

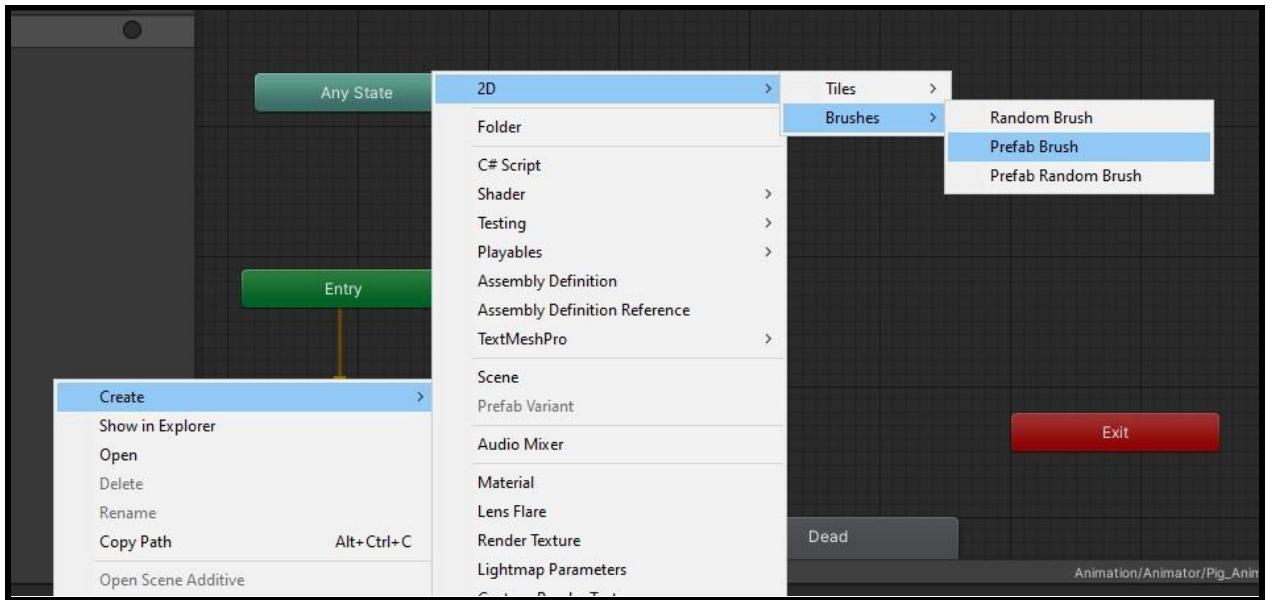
Making Prefab Brushes



- Now we are making prefab brushes for pick-up objects because it's simple to place and simple to use.
- First Download Unity 2D extra from unity github site
- Then drag and drop extract folder of unity 2d extra in unity project.



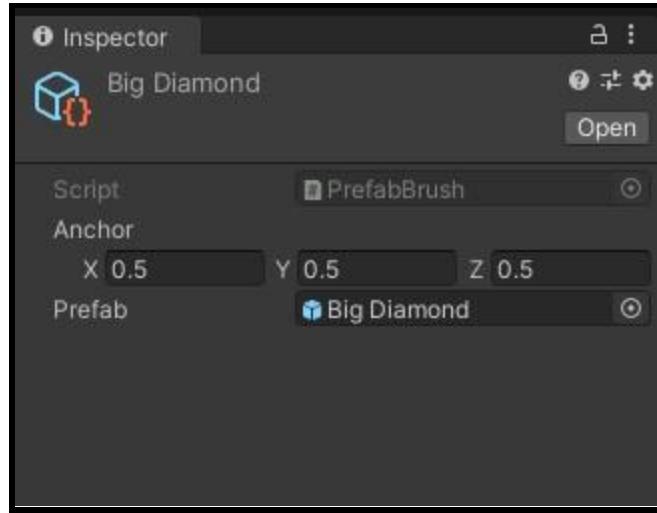
- Next Making a prefab Object
- Drag and drop any game object from Hierarchy Window to a Project Window.
- Icon Turn into a blue colour.



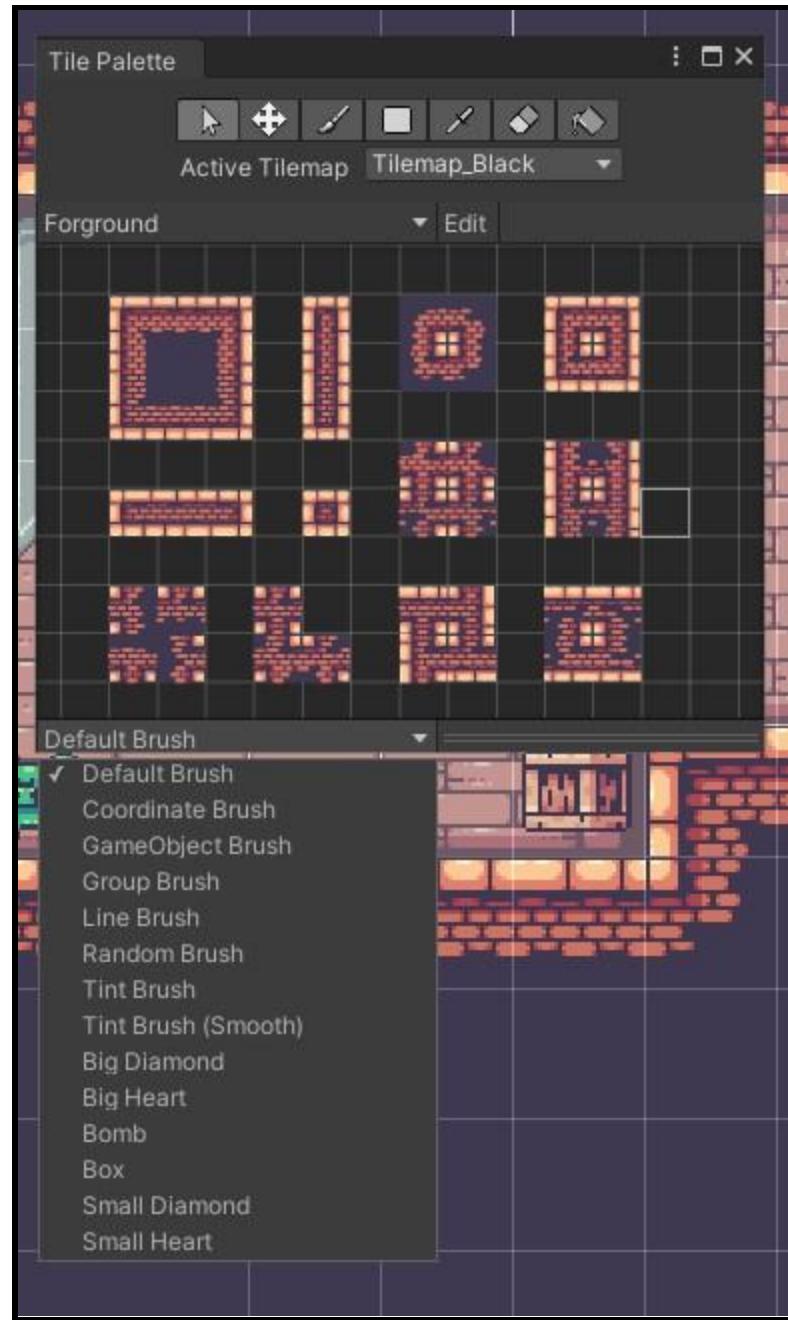
- Now Right click on the unity project window then
2D>Brushes>Prefab Brushes.



- Now make a Prefab for all the Collectables and a decoration object.
- Players can interact with all these objects.
- In these Prefab we can also add the scripts, Collider and all things we do with the normal objects.

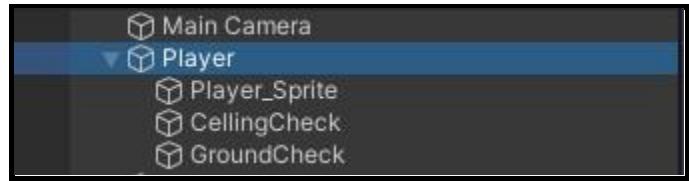


- After making prefab brushes and prefabs, Now drag and drop the prefab into the Scriptable Prefab Brush.
- You can make as many as you want.
- Prefab Brushes are very useful, it makes it easier to place all the coins, diamonds, etc.
- With the Prefab Brushes it makes it easier to place all the game objects in the scene (level).
- We can also change the game object position in the scene like we do other game objects.



- Now Go to Tile Palette and change Default Brush to your custom brush (Big Diamond).
- And Place it on the level.

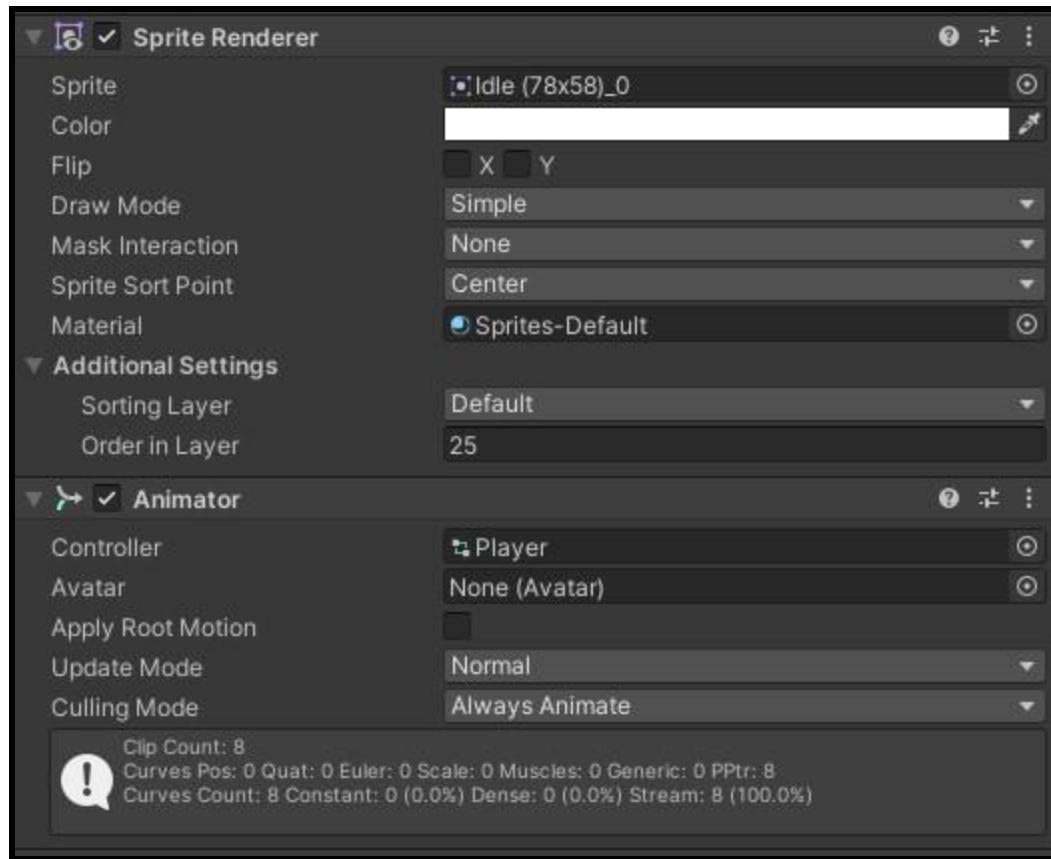
Player Setup



- Create a empty child in Hierarchy window and name it Player

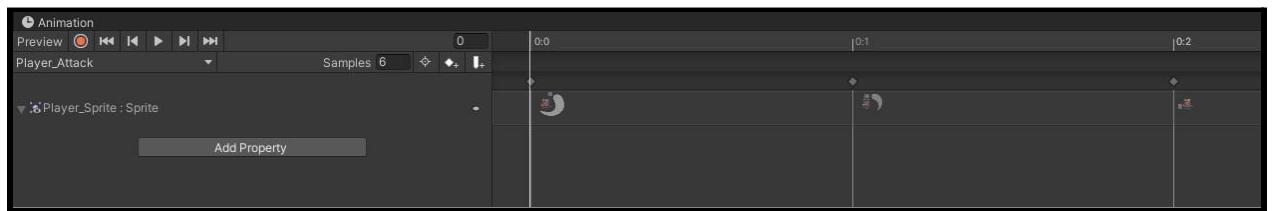


- Then drag and drop the Player sprite in the scene view and change the name Player Sprite.
- After completing this step Now make the sprite Child of the main player game object.
- Player>Player_Sprite (Like you see in the Upper Image)

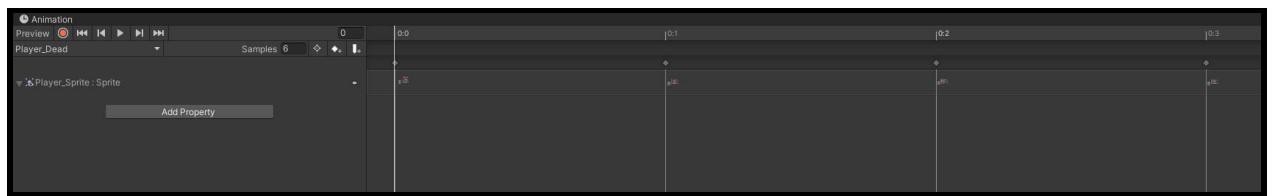


- If you drag multiple sprites at the same time it will make an Animation and also make an animator.
- When it makes animation an animator also chooses a path where we save all the player animation.
- Now Making all the Animations of the player.

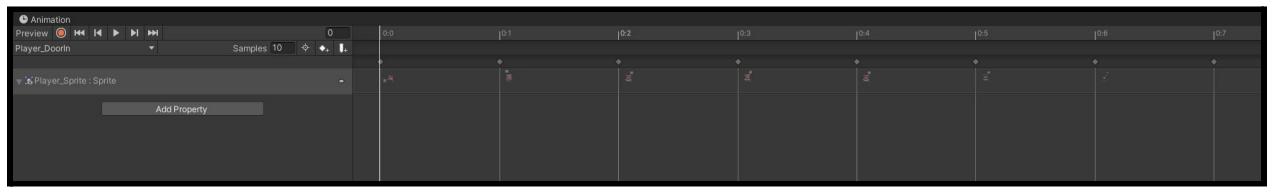
Player Sprites and Animations



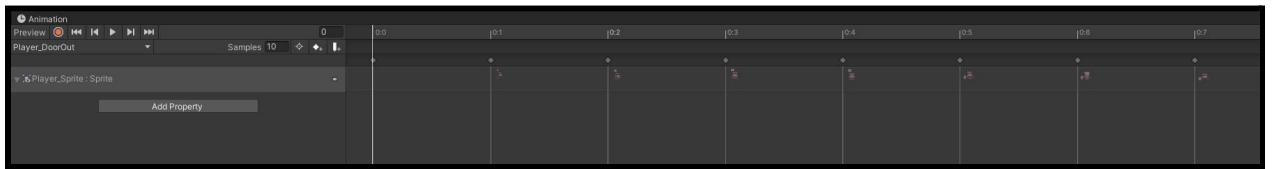
Player Attack



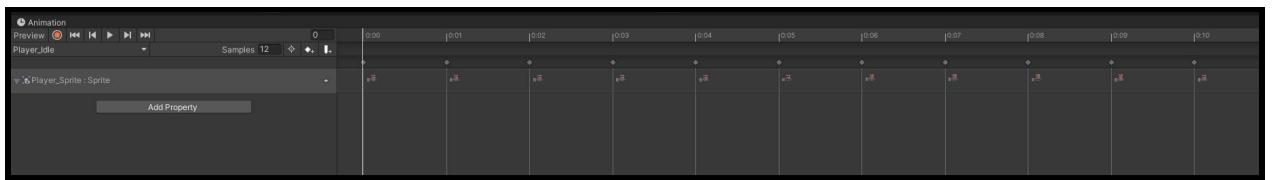
Player Dead



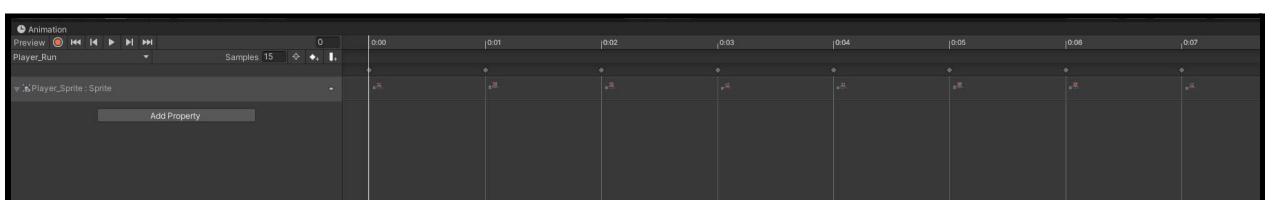
Player DoorIn



Player DoorOut



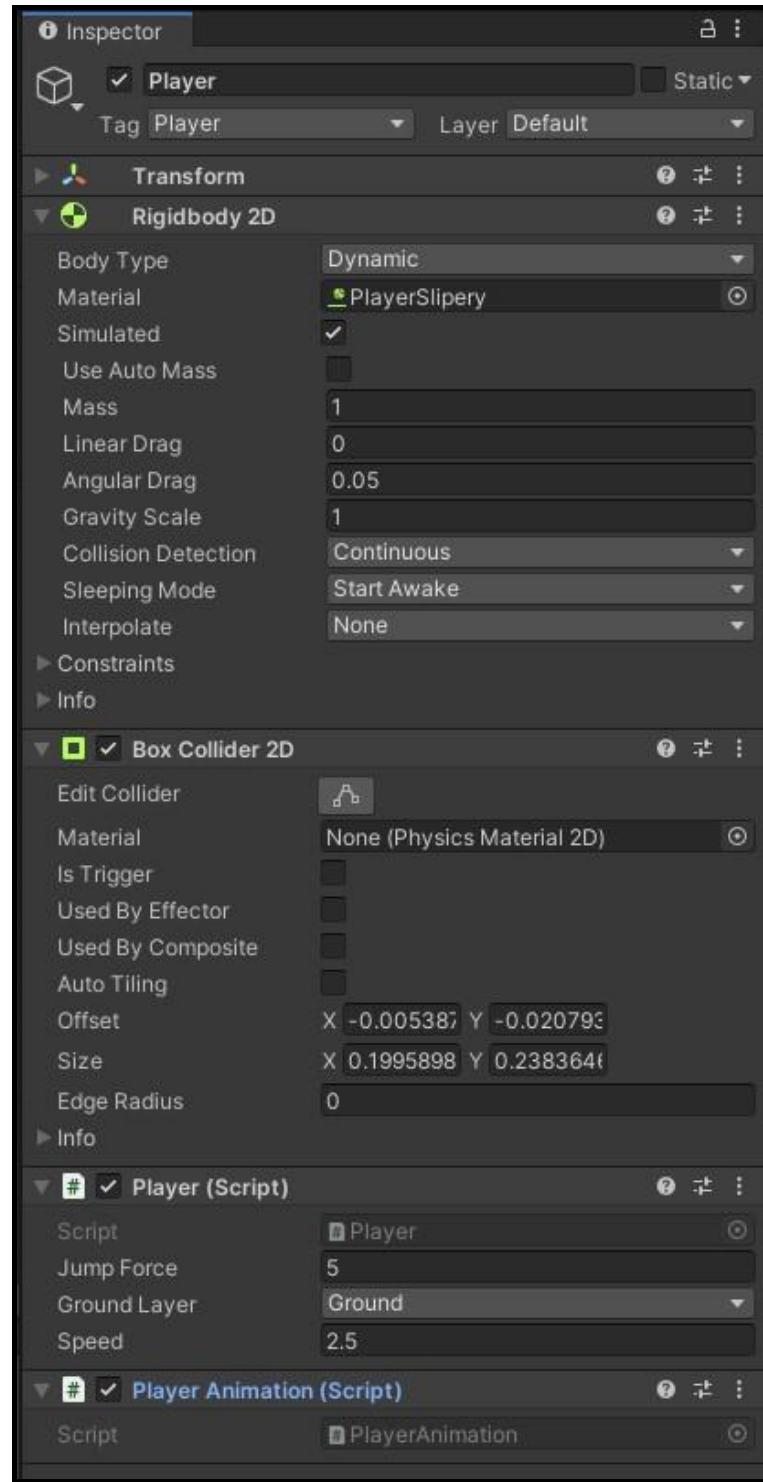
Player Idle



Player Run

- This is the process of making animation from the sprite sheets.

Making Player Moveable

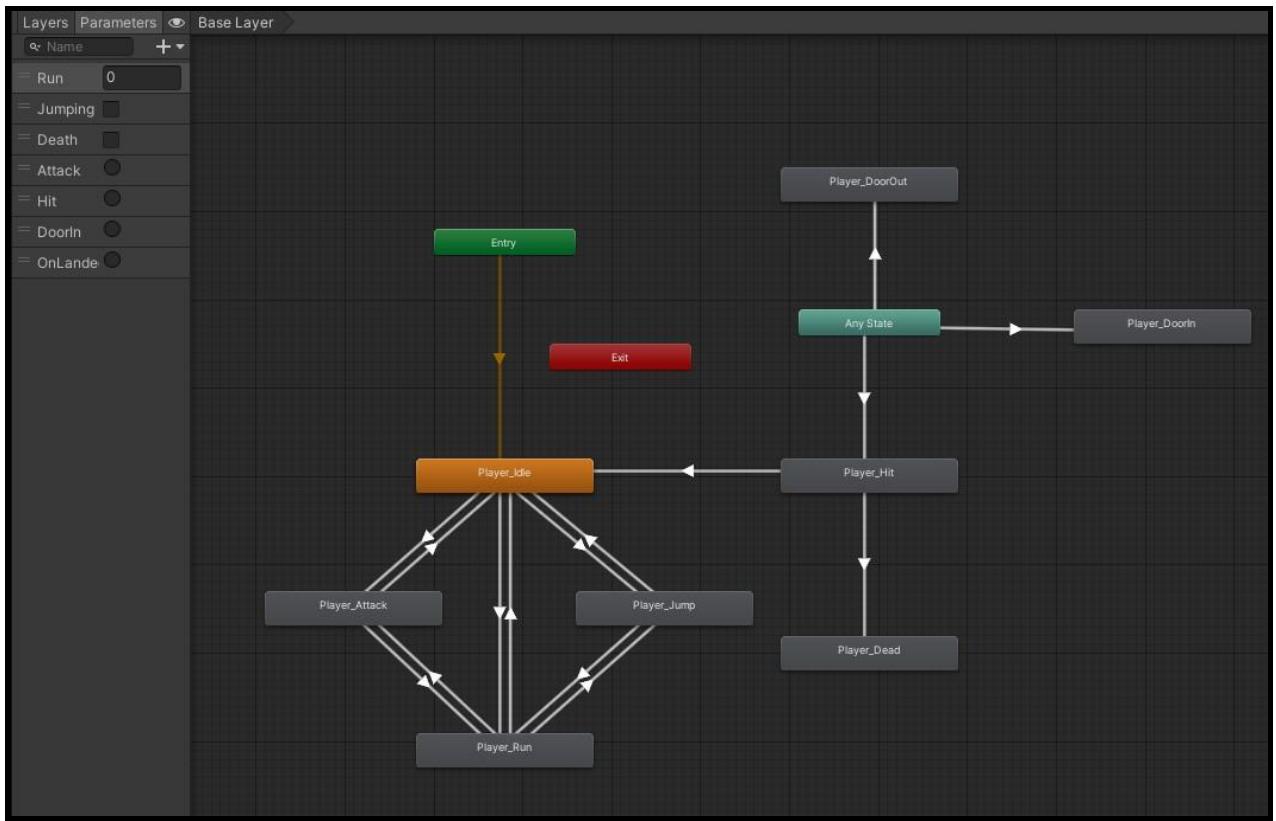


- Now Making player Moveable
- First Add Box collider 2D to the player
- Second Add Rigidbody 2D in the player
- Remember After Add rigidbody in the player Go to Constraints>Freeze Position > Z (Make it Tick)
- Third Add Scripts.

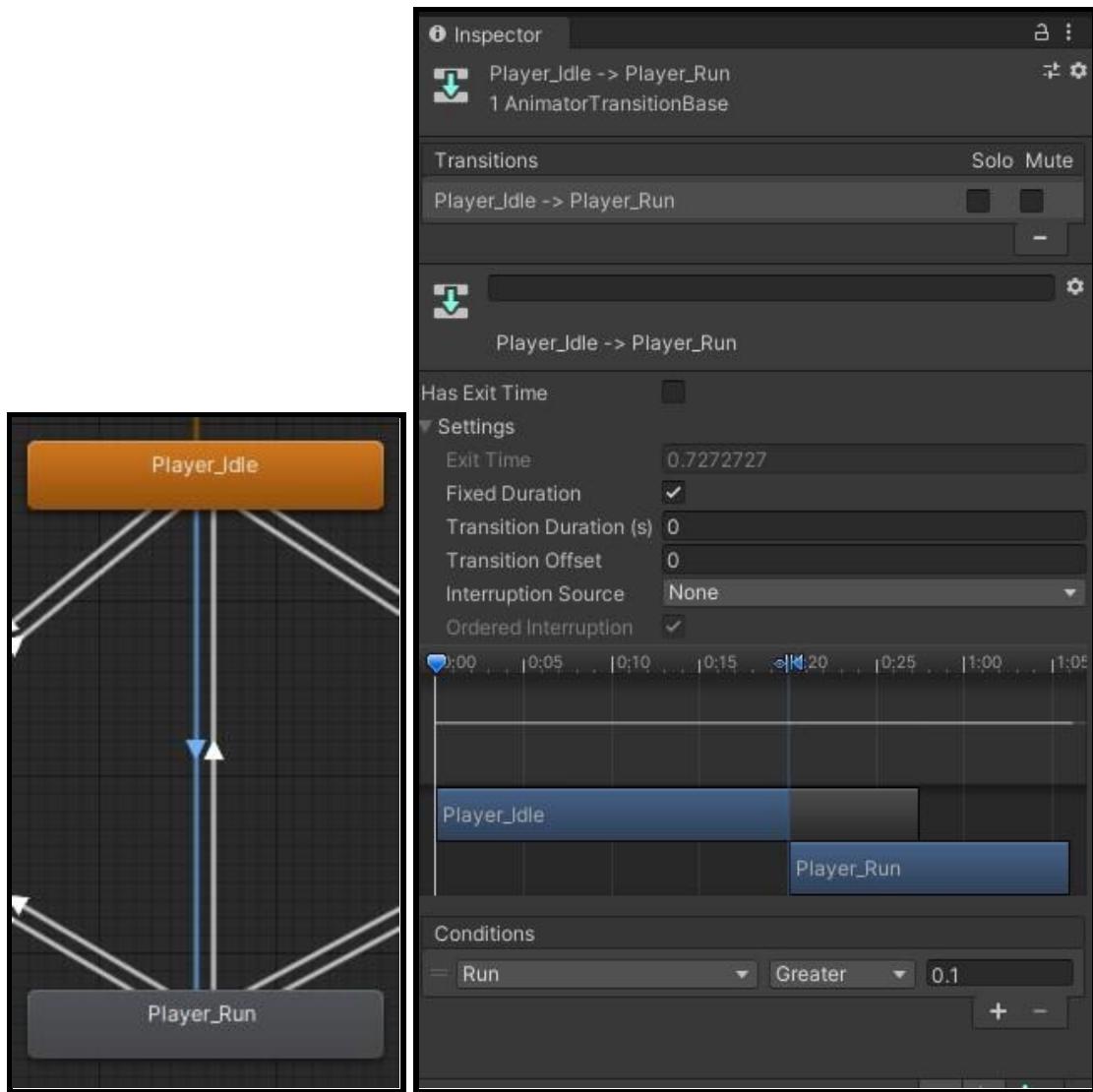
Setup Player Parameter and Animation



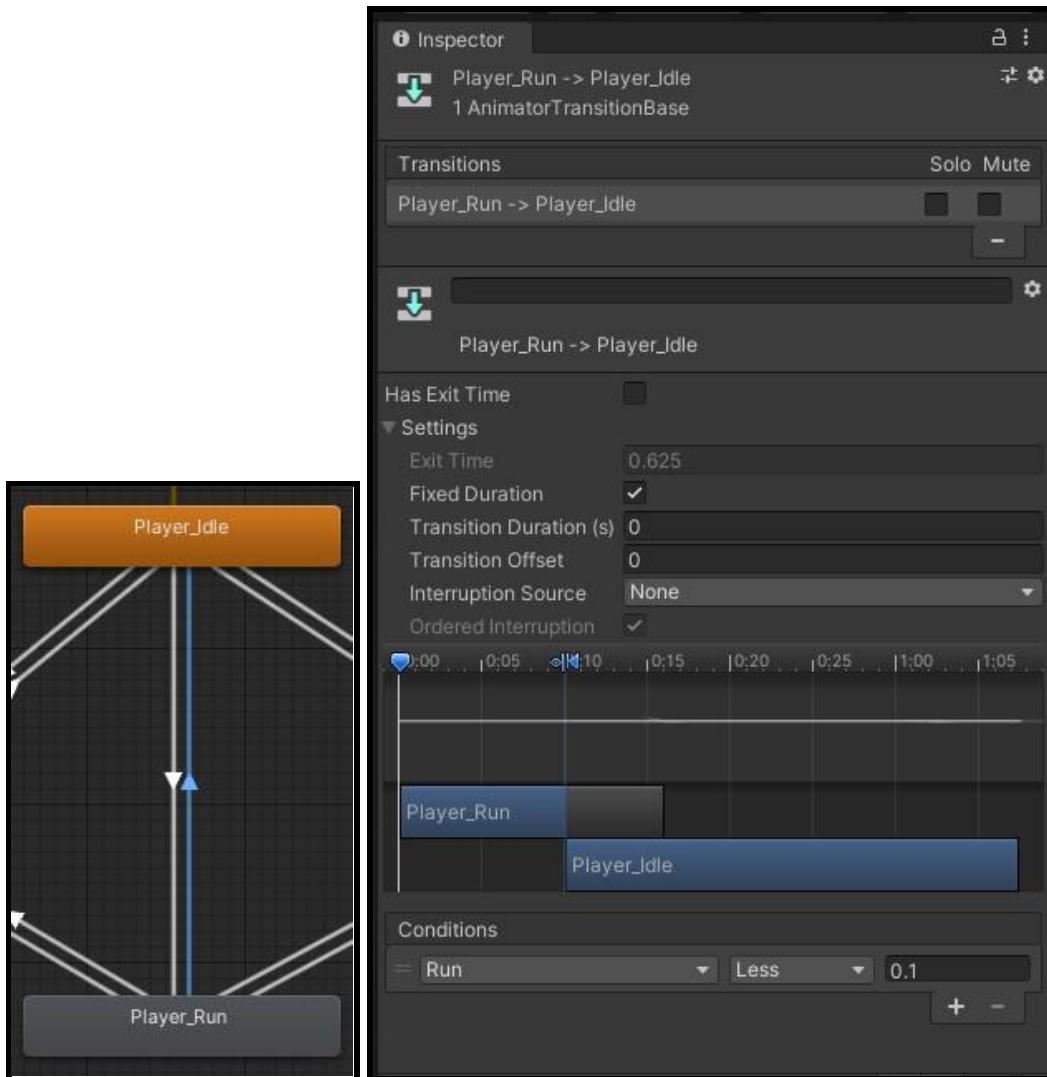
- After completing previous steps, Now setup the player Animation in the animator window and make some Parameters.
- Parameter helps us to run an animation we can call parameter from the scripts.
- Without Parameters we can not trigger animation correctly like Run Animation, Idle, Jump etc.



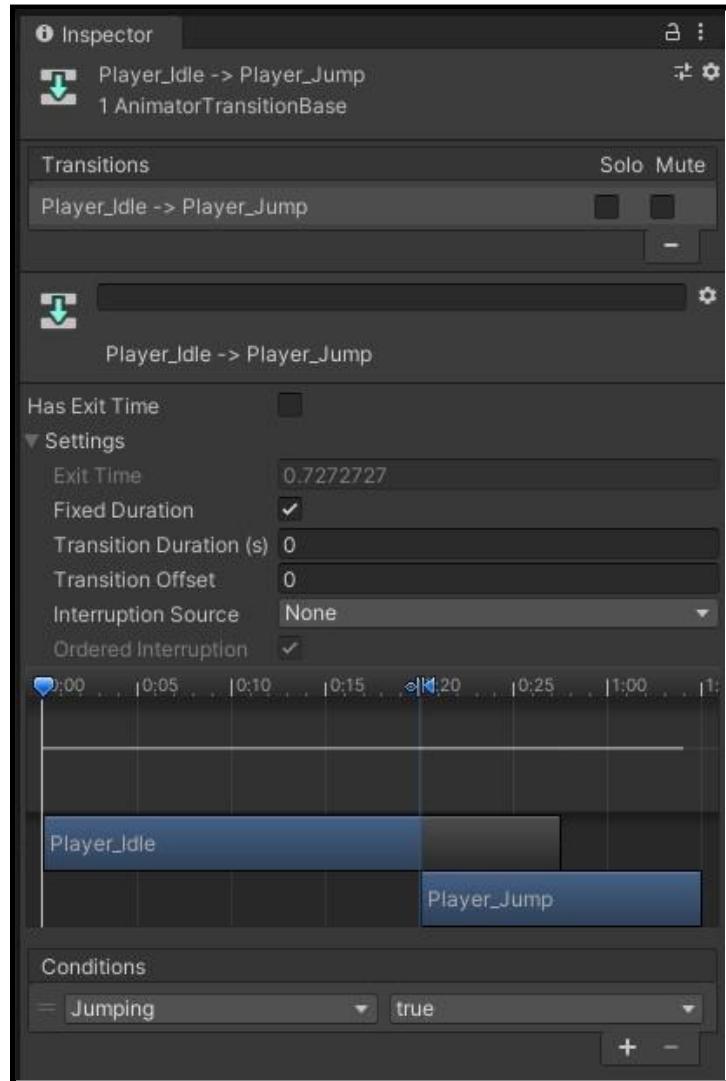
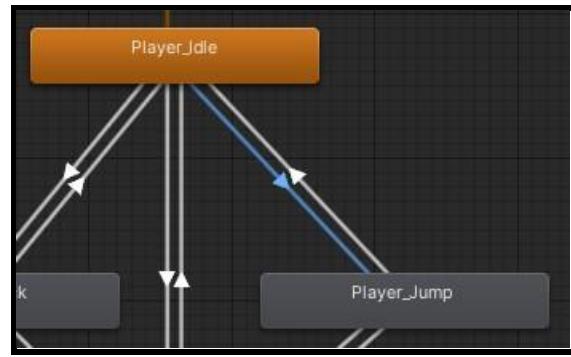
- This is the Animator View where we can make a logic of every animation Example :- In the start your player in Idle animation then you click the left button player animation change to Run and so on like jump animation and attack animation.



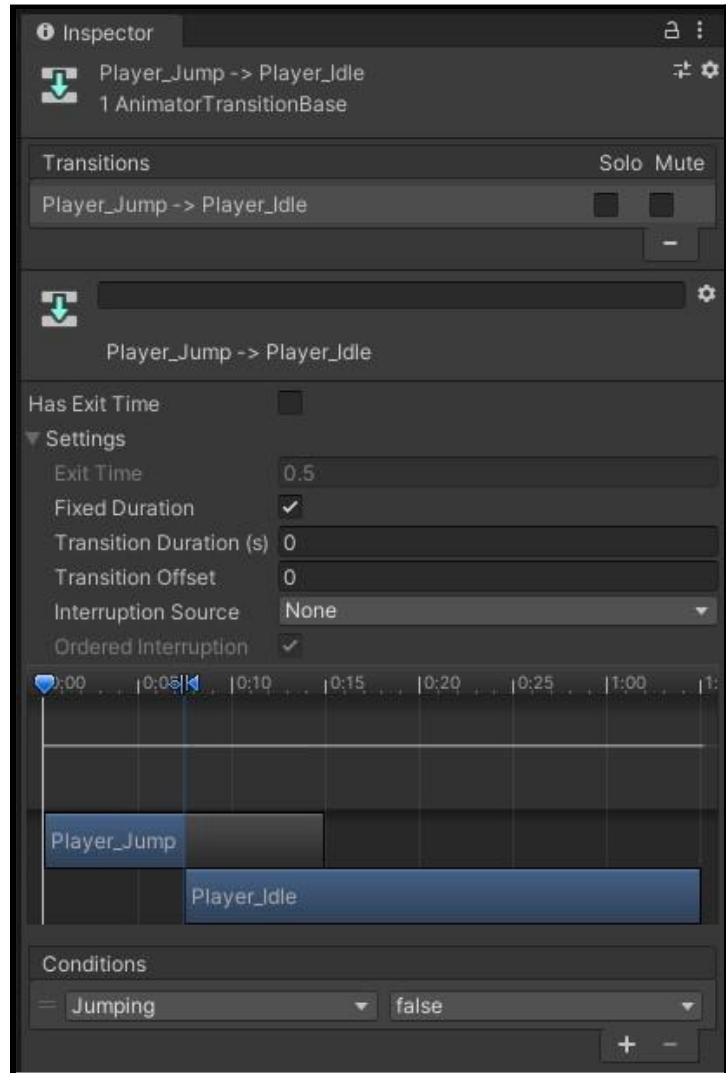
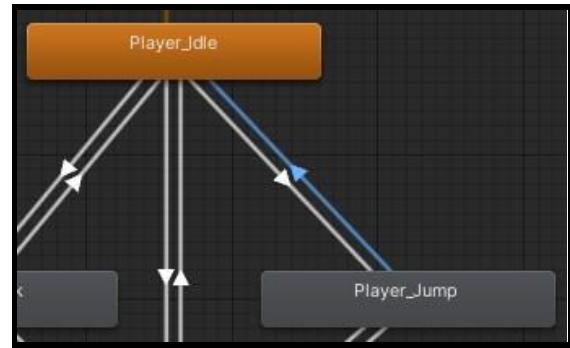
- Player Idle to Run Condition.



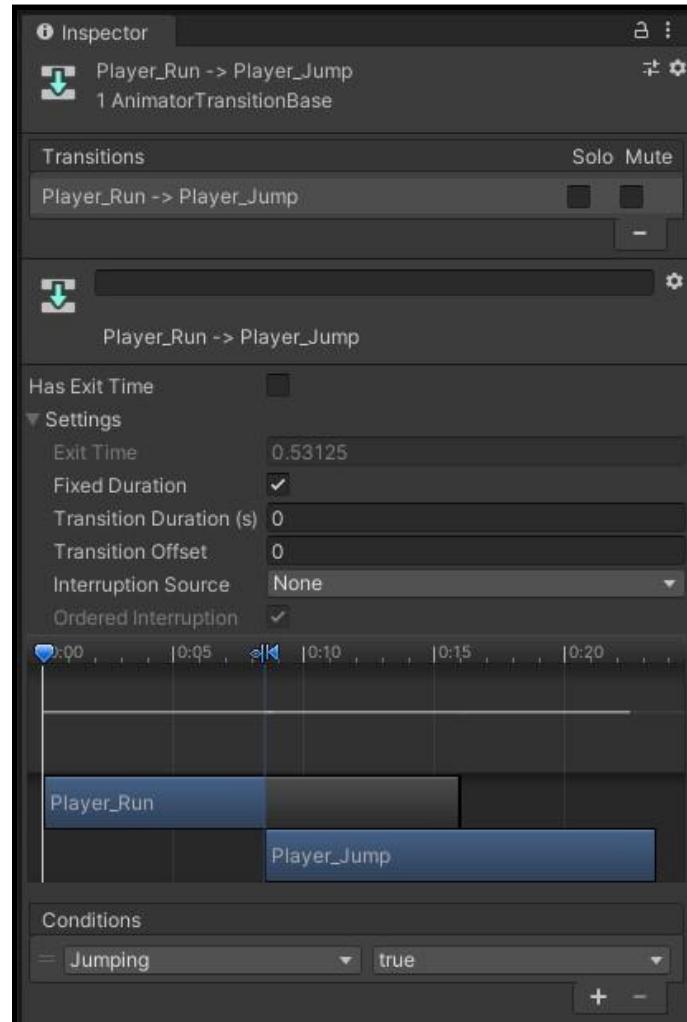
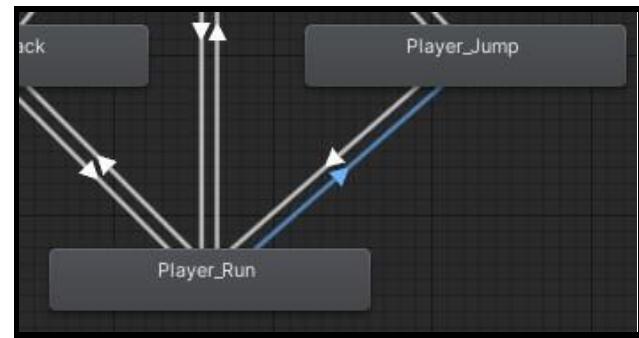
- Player Run to Idle Condition



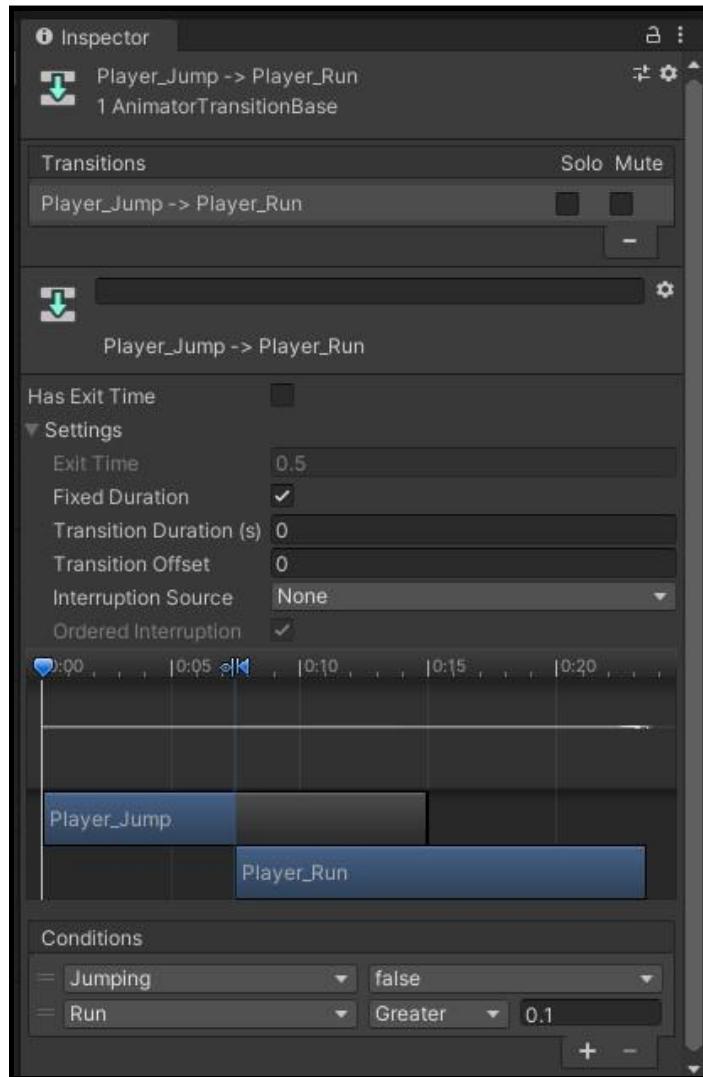
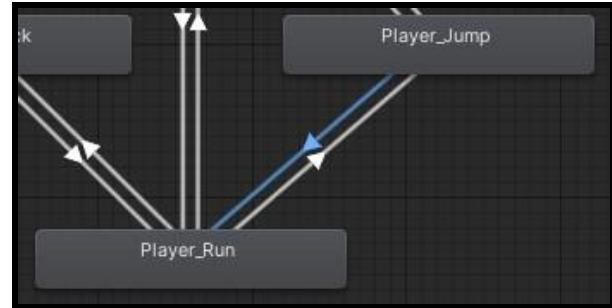
- Player Idle to Jump Condition



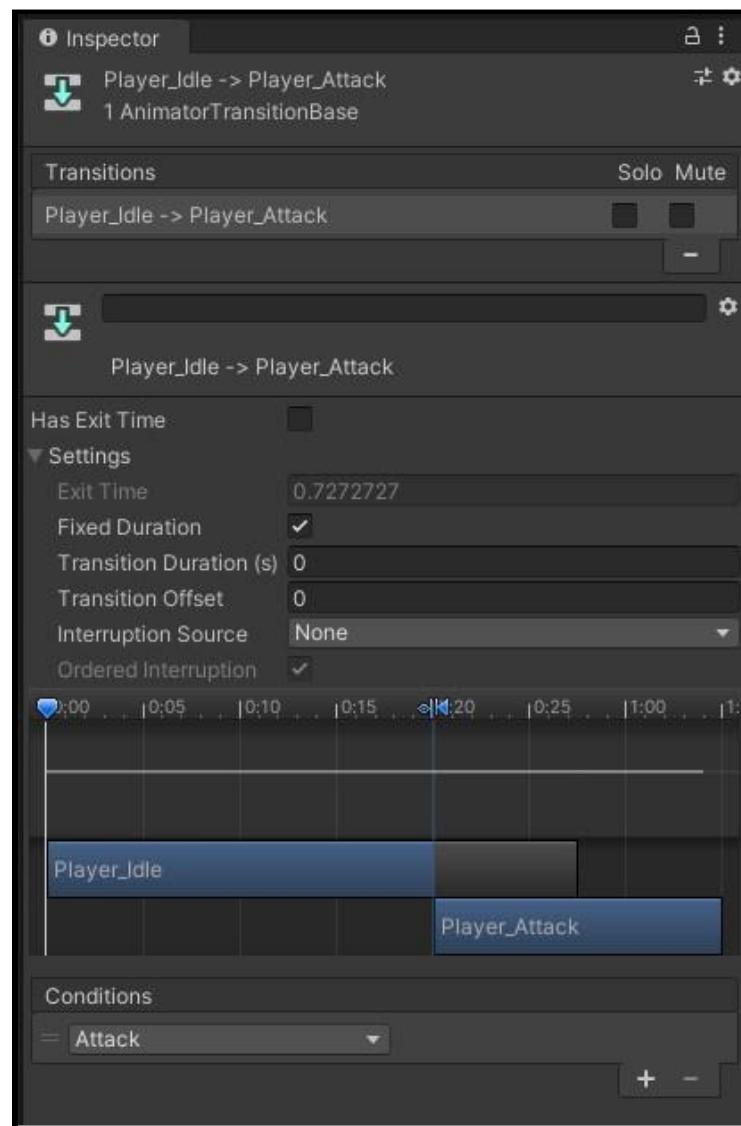
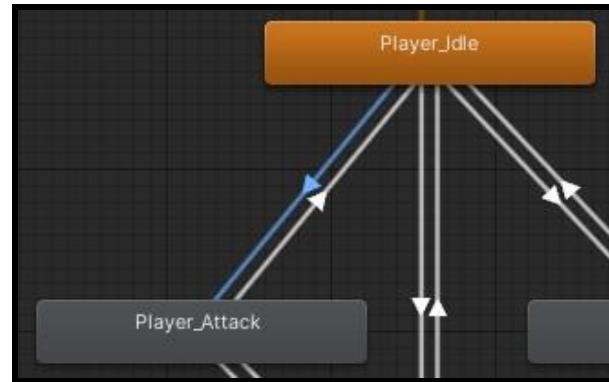
- Player Jump to Idle Condition



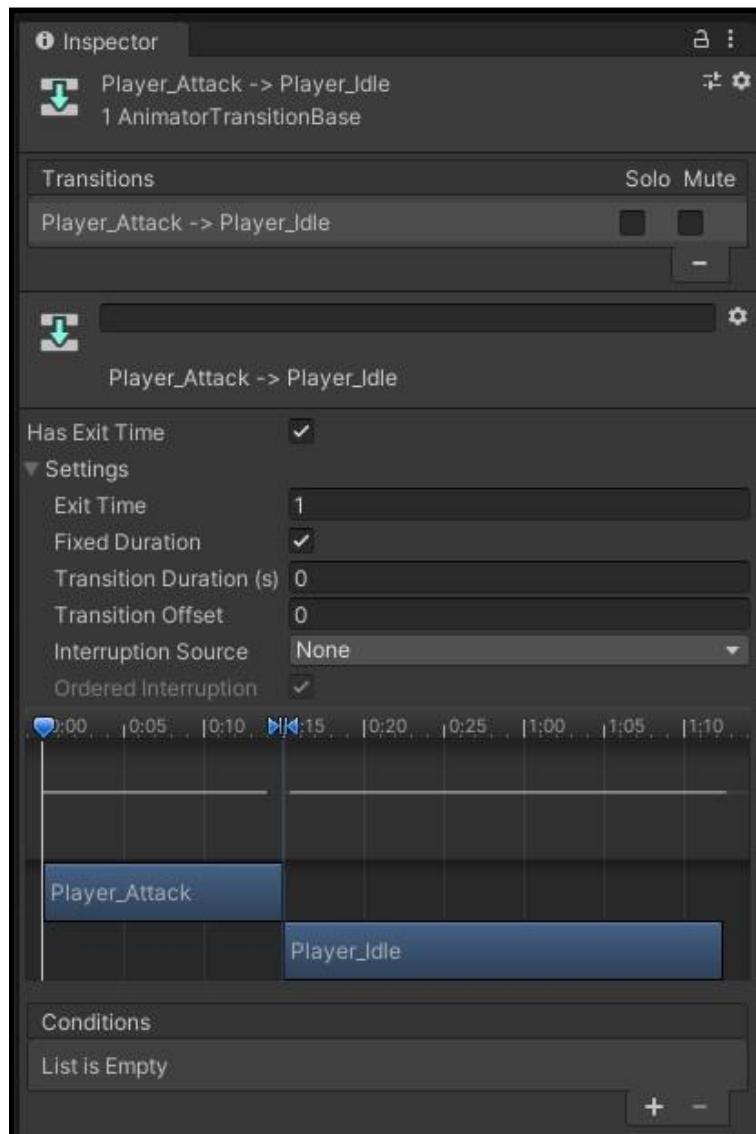
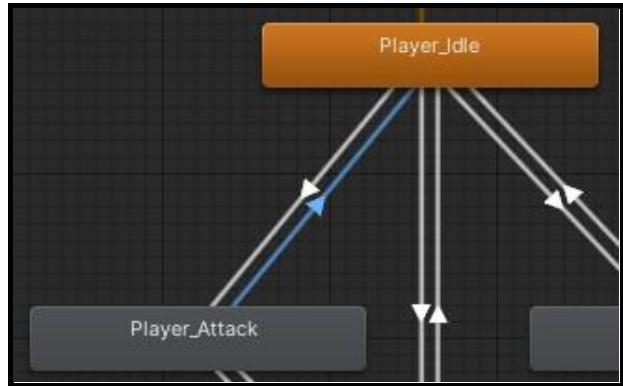
- Player Run to Jump Condition



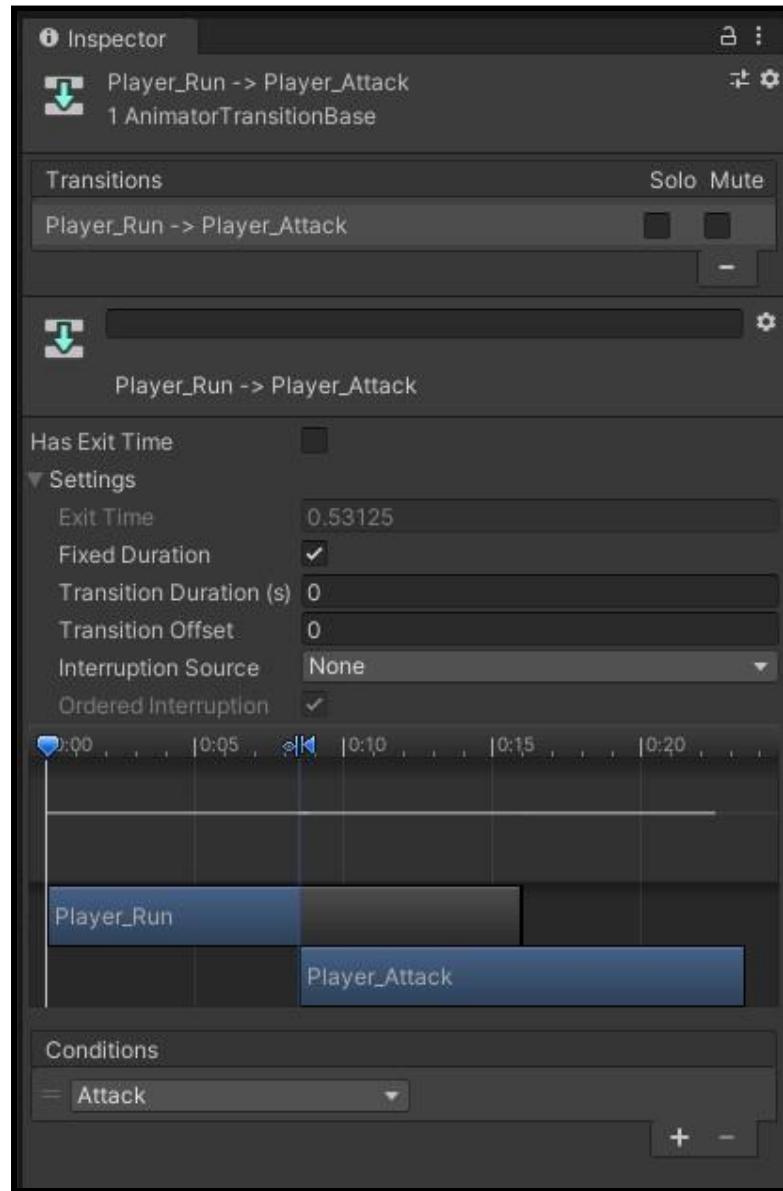
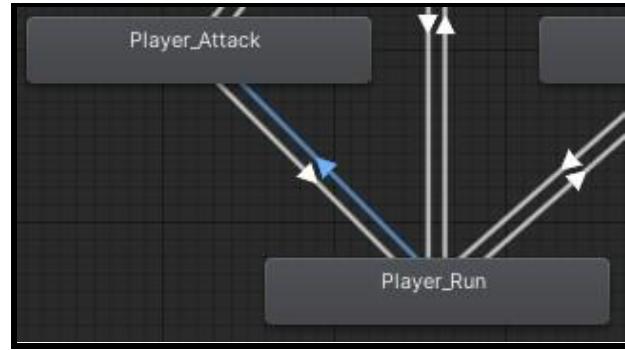
- Player Jump to Run Condition



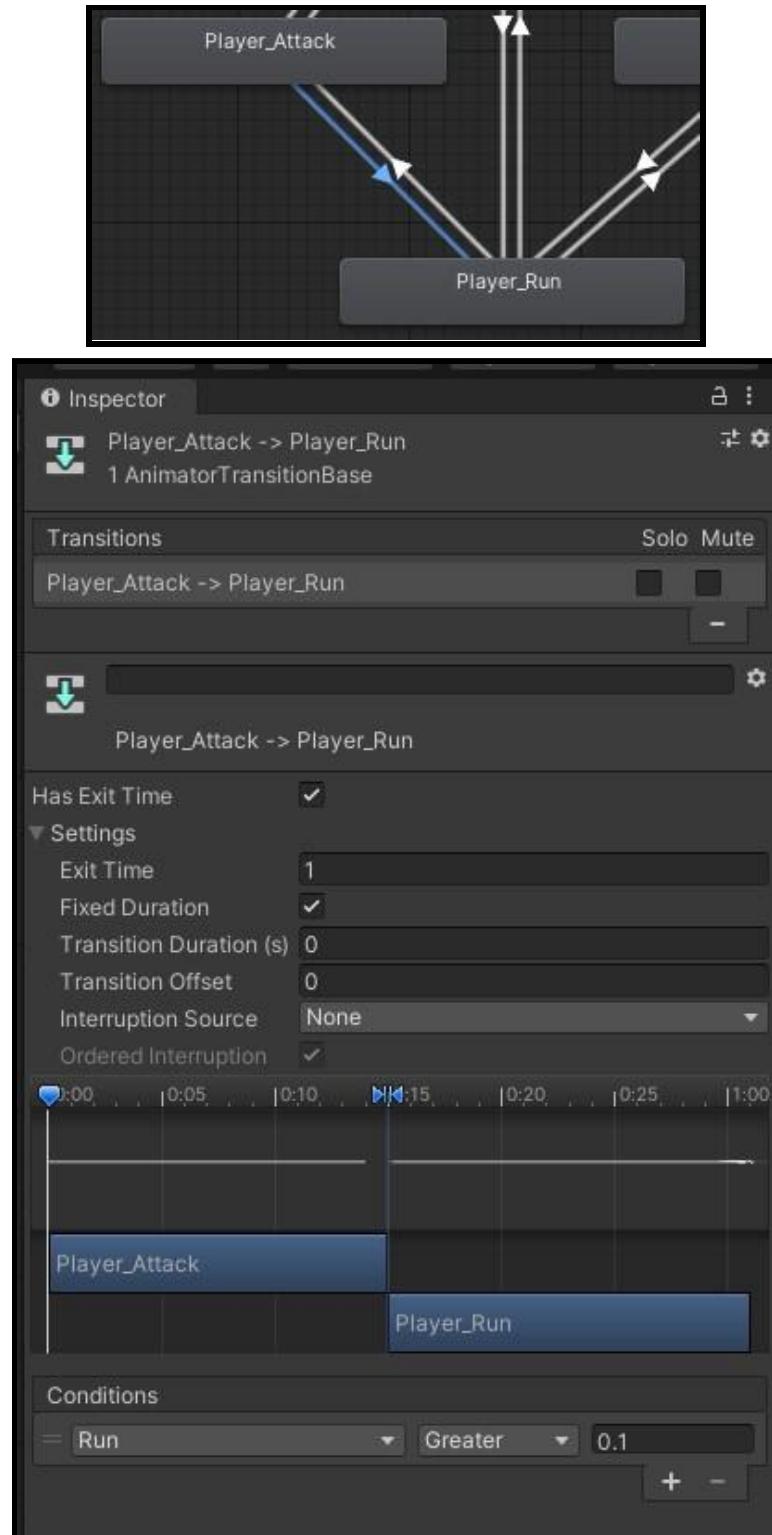
- Player Idle to Attack Condition



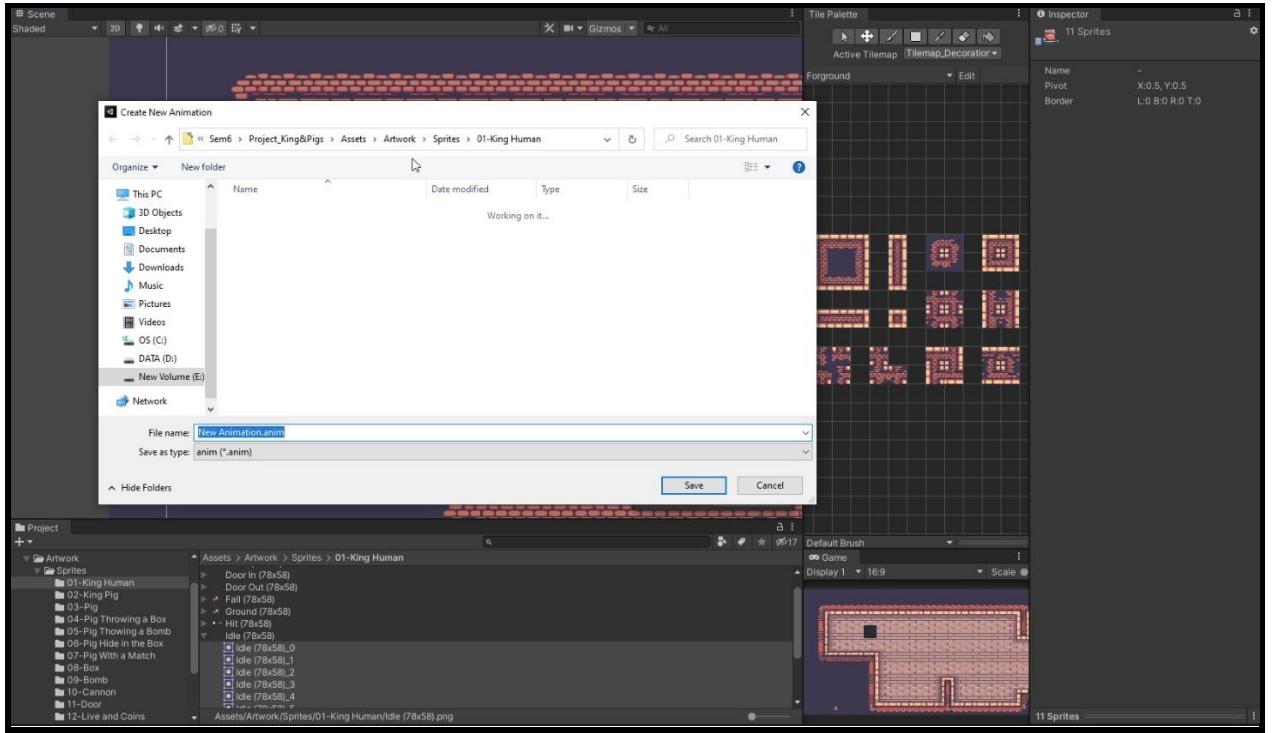
- Player Attack to Idle Condition



- Player Run to Attack Condition

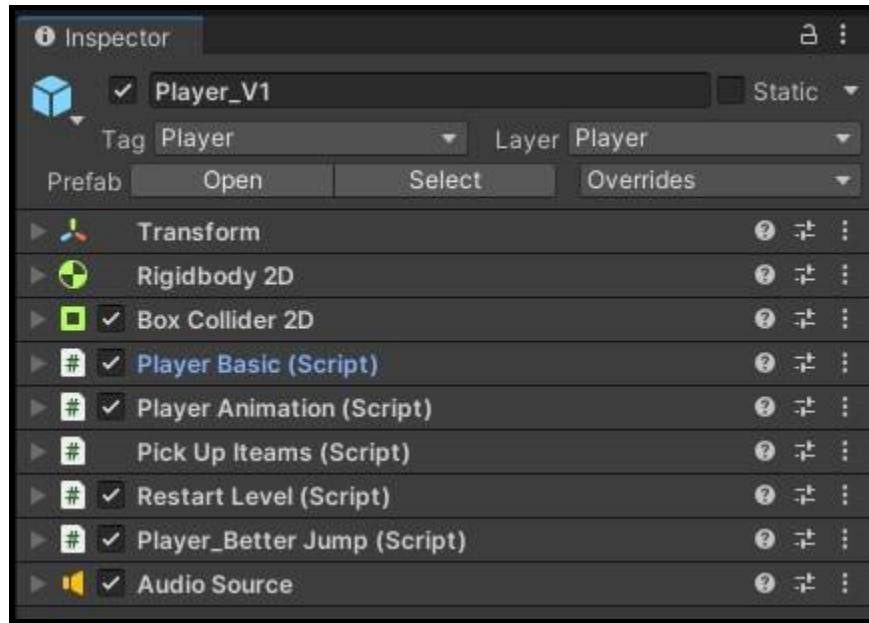


- Player Attack to Run Condition



- Importe all idle sprite to scene view
- It's automatic create a animation and render a sprite in scene view
- Give Name an animation and give a path where animation and animator are saved for a character.
- Give a player Tag called it player.

Player Scripts (Inspector View)



- In This Player Component, attached six different scripts.
- Player Basic Script controls all the movement, physics and attack system.
- PlayerAnimation Script Controls all the player animator and animations.
- PickUpIteams Script takes care of all the collectable Items like: gold coin, silver coin, Health pickup, etc.
- Restart Level takes care of restart level when the player dies.
- Player_Better Jump Script Control player jump.

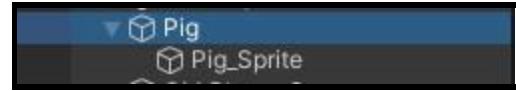




Enemy Setup (Type: Melee Attack)



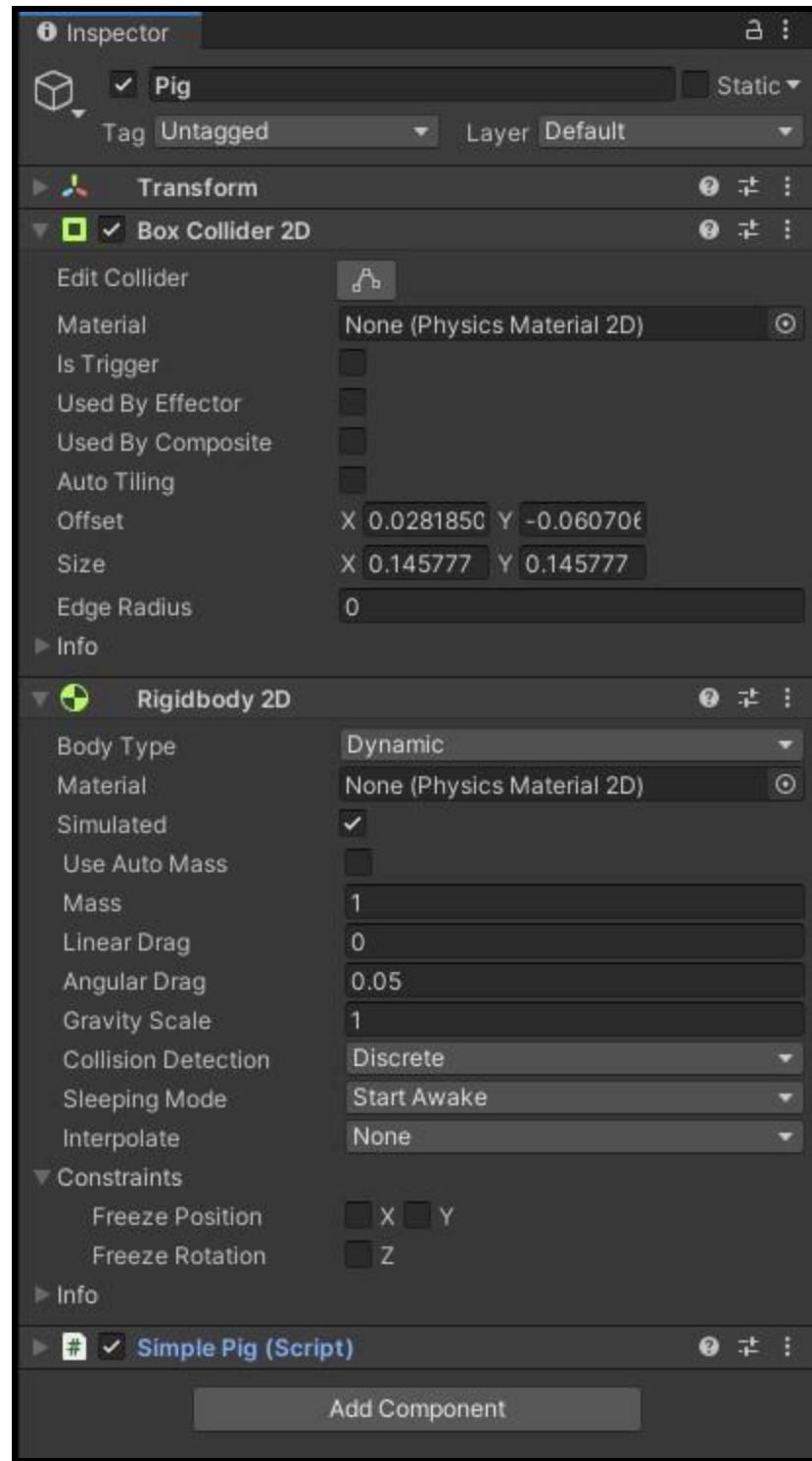
- These are three enemies, they all share the same script and method but all three have unique attacks and different ranges.
- Melee attack type enemies can patrol, If they see the player in their range, they can chase or if a player in attack range they can attack on the player.
- They have three states: Idle state, Patrol state, Chase state.



- Create an empty child like we do in player setup.

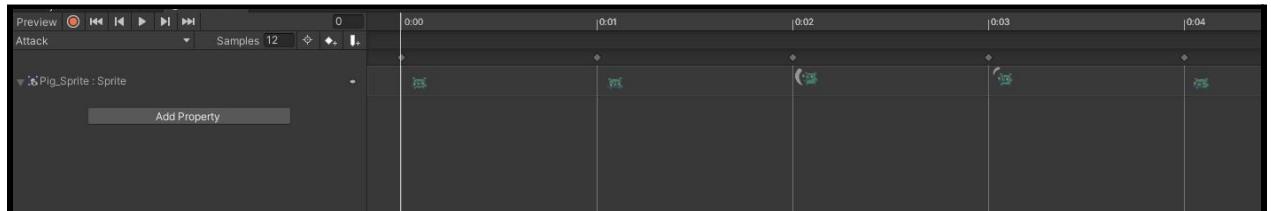


- Then drag and drop the Enemy sprite in the scene view and change the name Enemy Sprite.
- After completing this step Now make the sprite Child of the main enemy game object.
- Enemy>Enemy_Sprite (Like you see in the Upper Image)

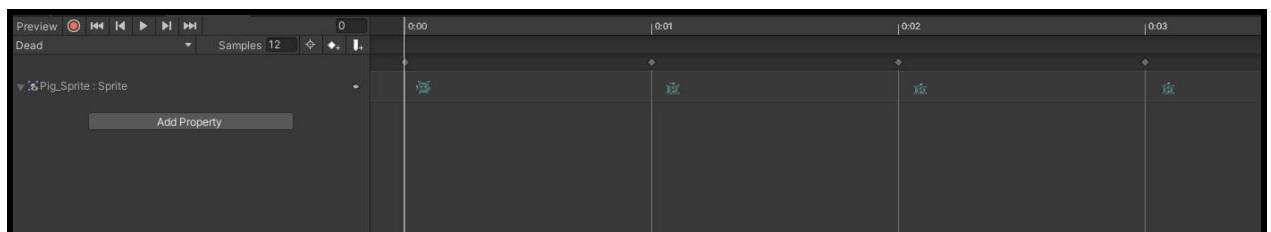


- Add Box Collider 2D and Rigidbody 2D.
- Also Tick on the Freeze Rotation Z.

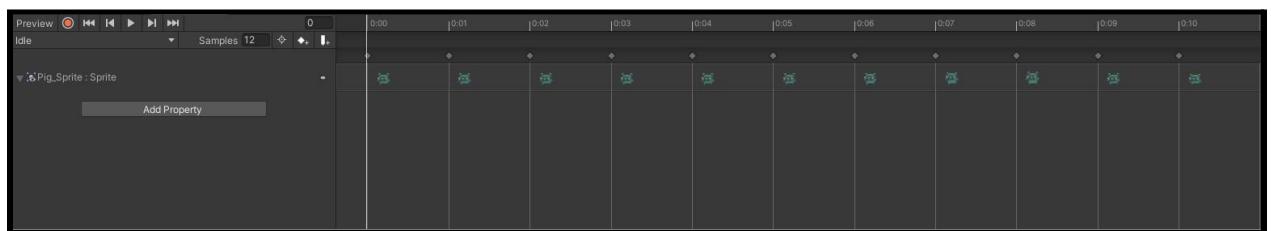
Setup Enemy Animation



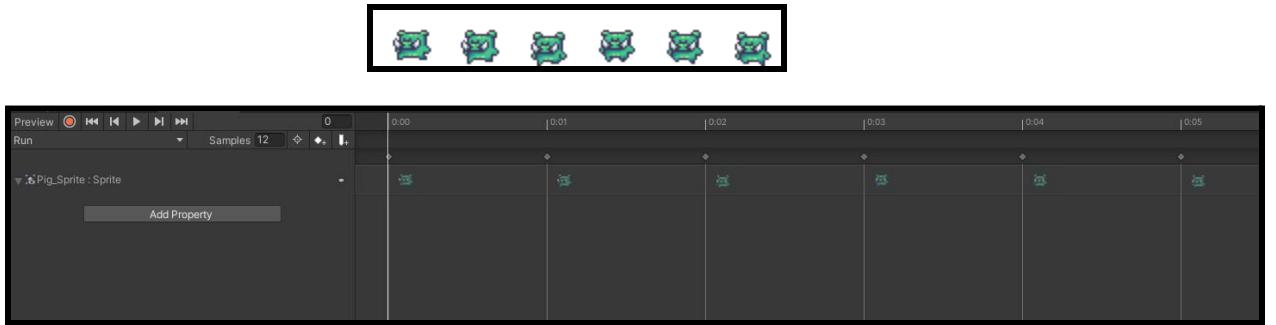
Enemy Attack



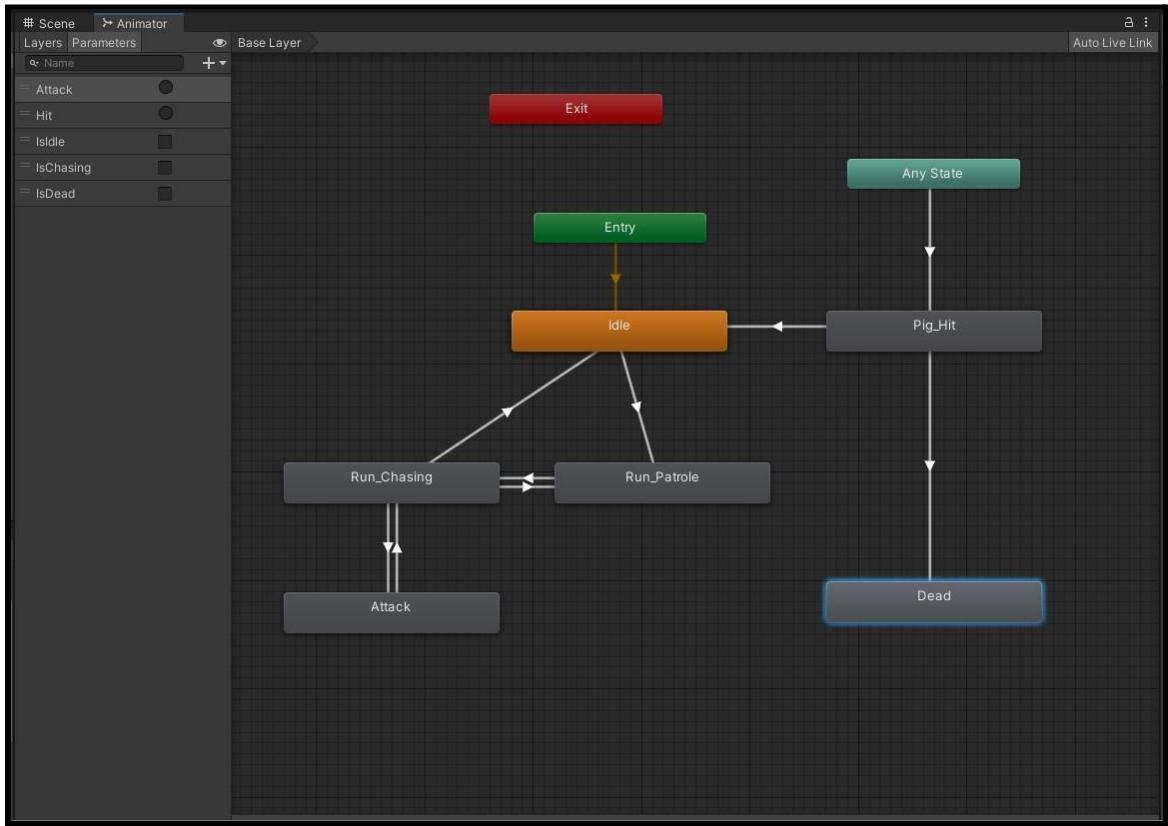
Enemy Dead



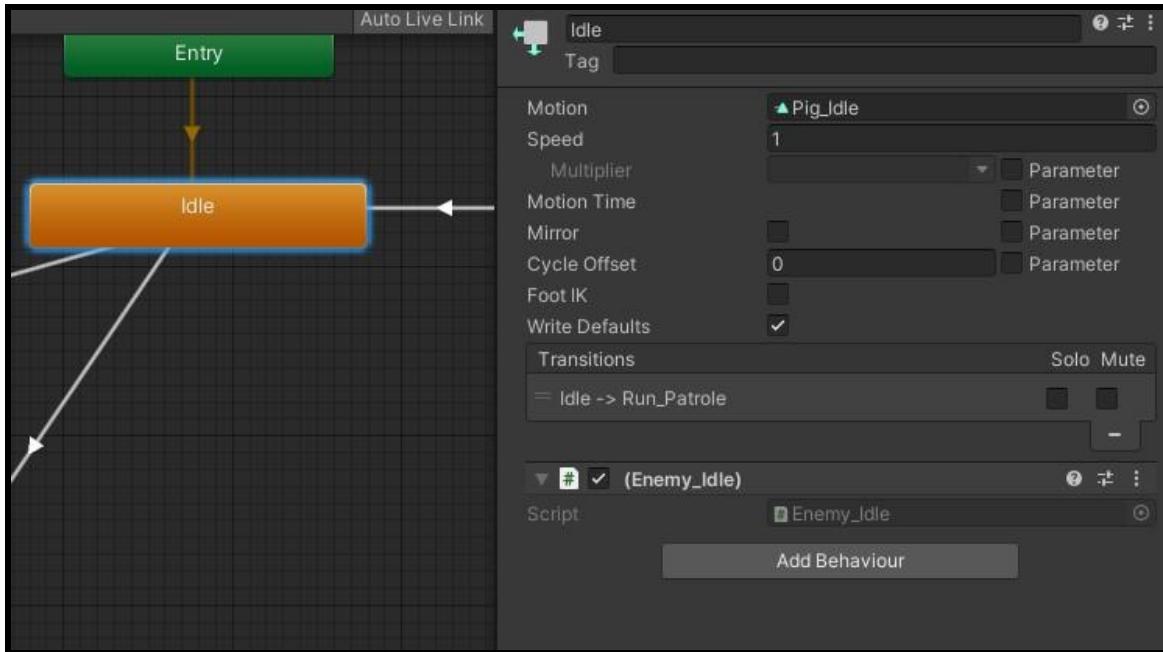
Enemy Idle



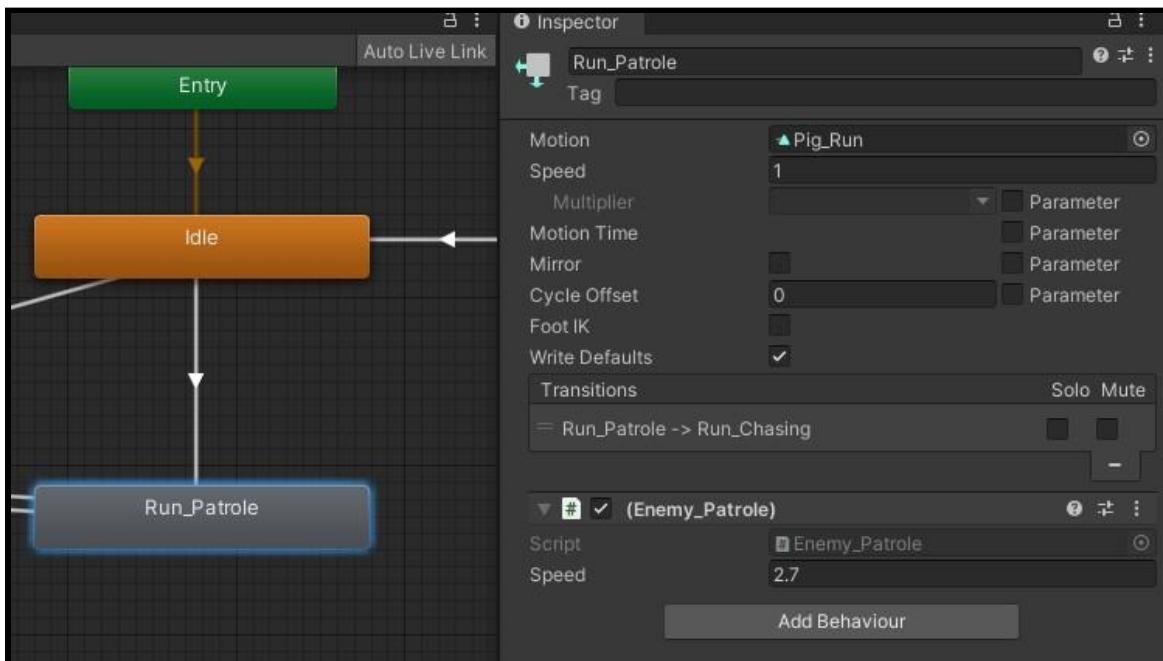
Enemy Run

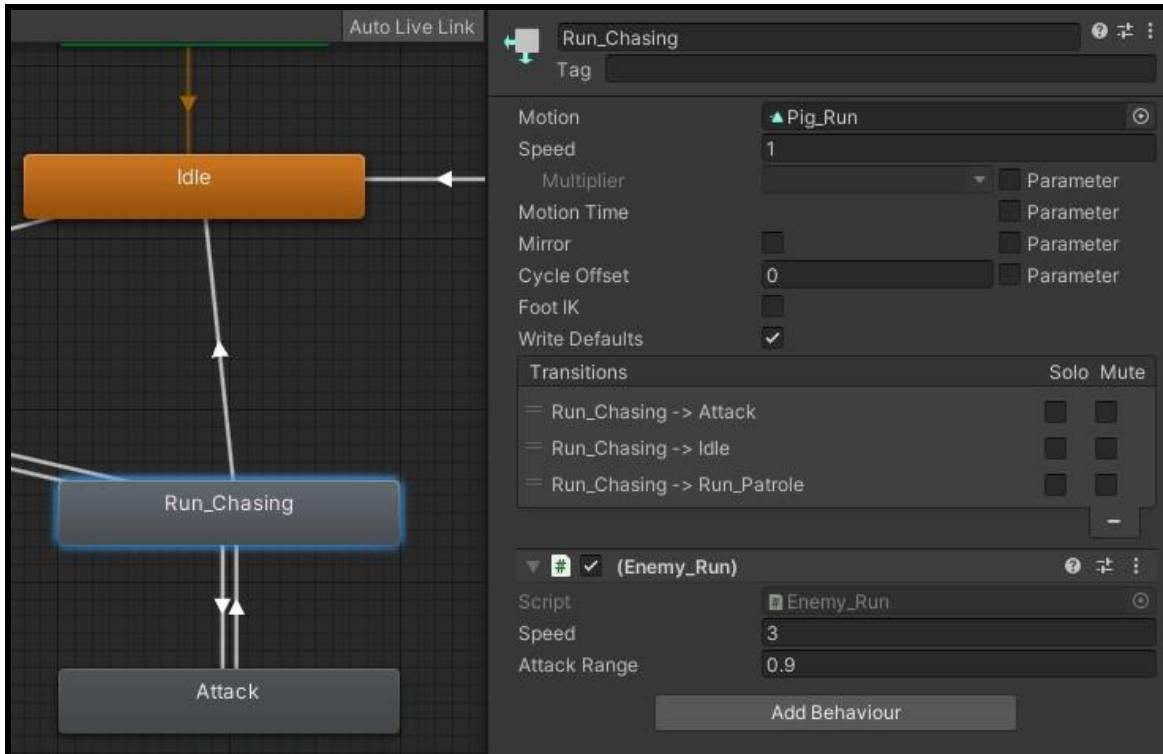


- For Enemy Ai, I am using the state machine method. In this method enemies switch one state to another like Patrole state to Chasing state.
- For the State machine, I am using the Unity Animation system and making a simple and accurate enemy Ai.
- My Melee Attack Enemies have Idle, Patrol, Chasing And Attack state.



- Making State machine with animation it's very easy. Click on the animation and now see the inspector click on the Add behaviour Button in the inspector then select script name the script and done.
- Now we can add any behaviour to that animation that we want.





```

34     if (Vector2.Distance(player.position, rb.position) <= attackRange)
35     {
36         animator.SetTrigger("Attack");
37     }
38 }
39
40 // OnStateExit is called when a transition ends and the state machine finishes evaluating this state
41 [Unity Message | 2 references]
42 override public void OnStateExit(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
43 {
44     animator.ResetTrigger("Attack");

```

- For Attack We can Trigger attack animation from the chasing state script.
- We can change to use different speeds for both condition patrol or chasing.

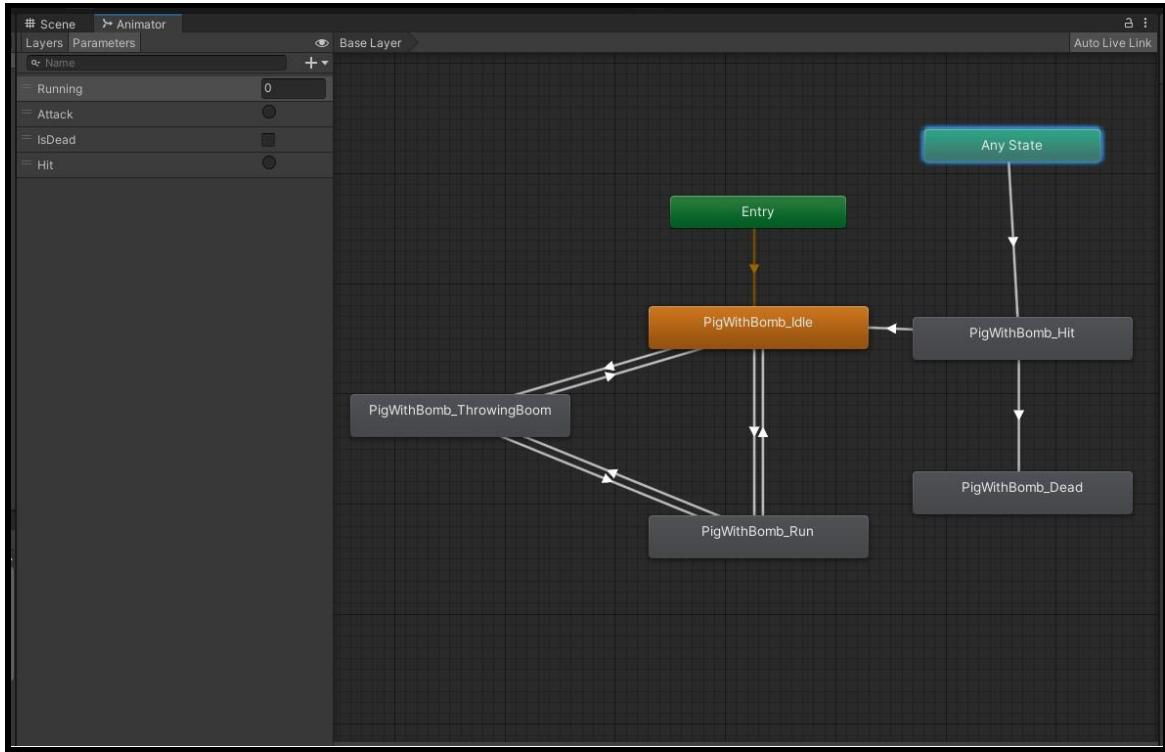


- All Three melee enemies have the same method and scripts. But all have different animations and attacks.

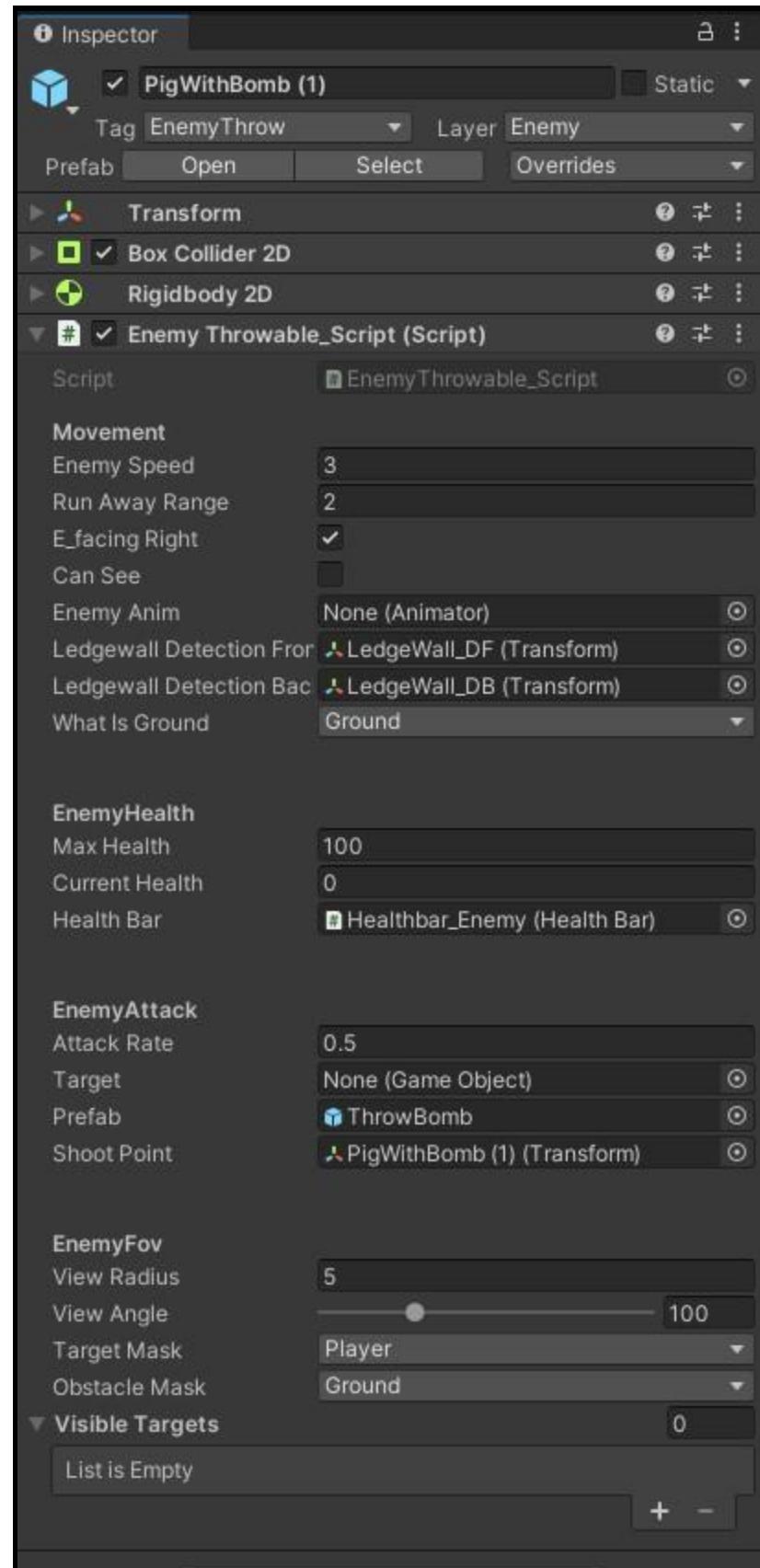
Enemy Setup (Type: Object Thrower)

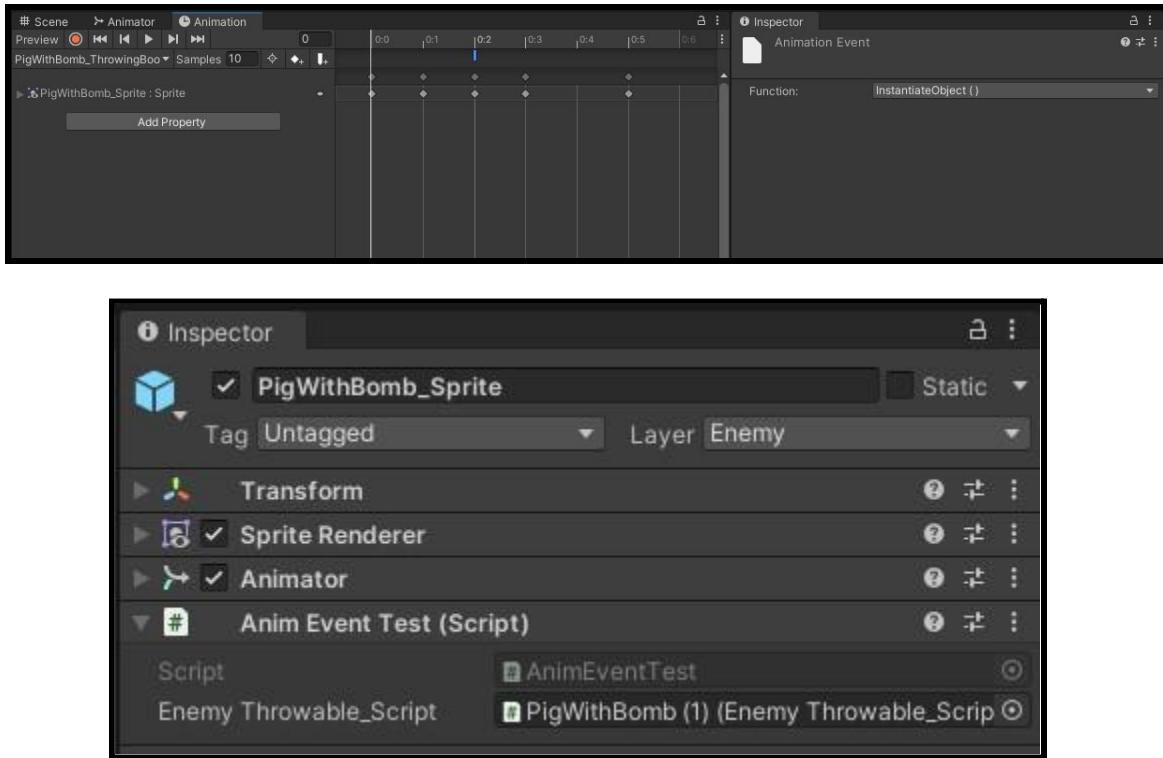


- Sprite and animation set are same in all the enemies and player
- In this type of enemies, I used a different method for Ai. Using script only, not an animation state machine.
- This Type of Enemy can Patrol, Attack when they see the player and run away when the player is very near.
- To throw a bomb at a certain frame, I am using an Animation Event.
- In this method we can spawn a bomb at a correct frame and it looks good and original.



- Parameters and Animation of the throwable enemies.
- Connect the animation according to the logic or enemy behavior.
- Behaviour means how the enemy interacts with the game world and how it reacts when seeing (or detect) the player.



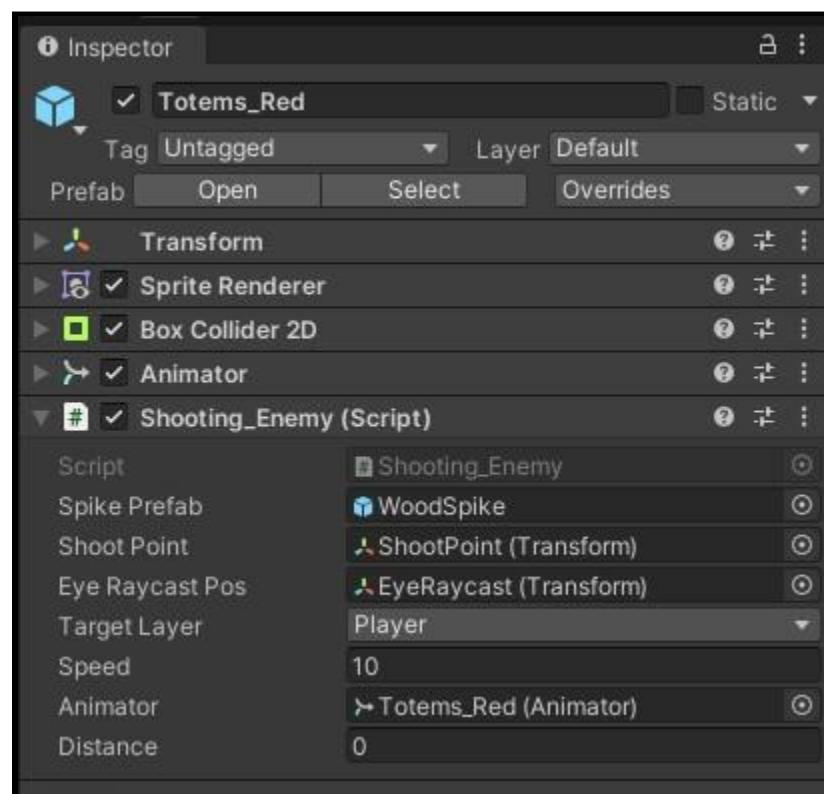
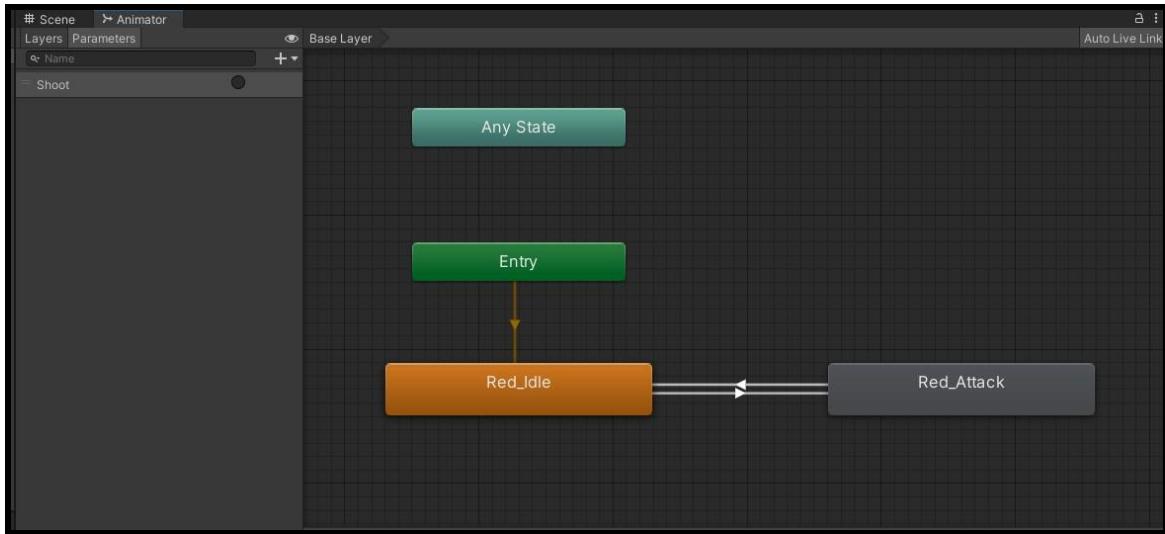


- After adding an event on any frame then click on it, it turns blue then see the inspector it shows all the public functions of the script attract the game object.
- Animation Event system is very useful for doing things according to the animation. we can control all the action on the specific frame like: How player takes damage, we can add the animation event on that specific frame.

Enemy Setup (Type: Shooting Enemies)



- These types of enemies can detect the player and start shooting.
- Sprite and animation use the same method as previous enemies.
- For spawn bullets also use the same method (Animation event).
- This type of enemies have only two types of animation: Idle animation and shoot animation.



Game collectibles



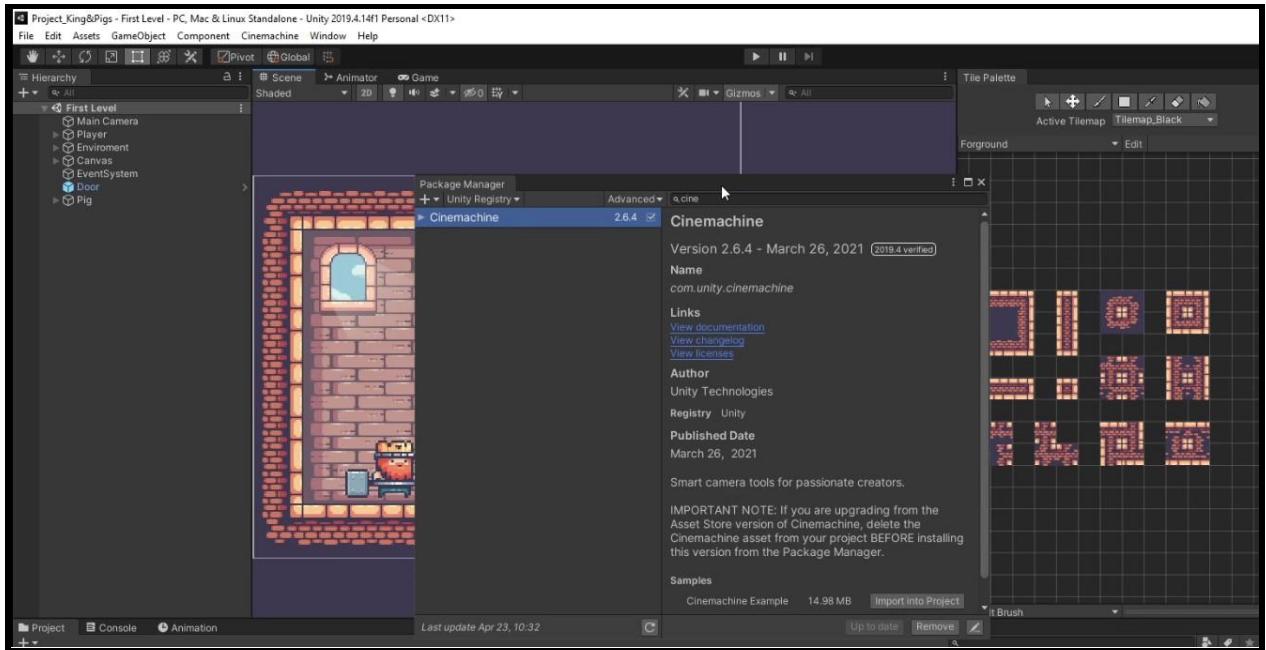
- All Collectables have a rotation animation, using a sprite sheet. and have a hit effect when the player hits any of these its play hit effect animation.

Traps



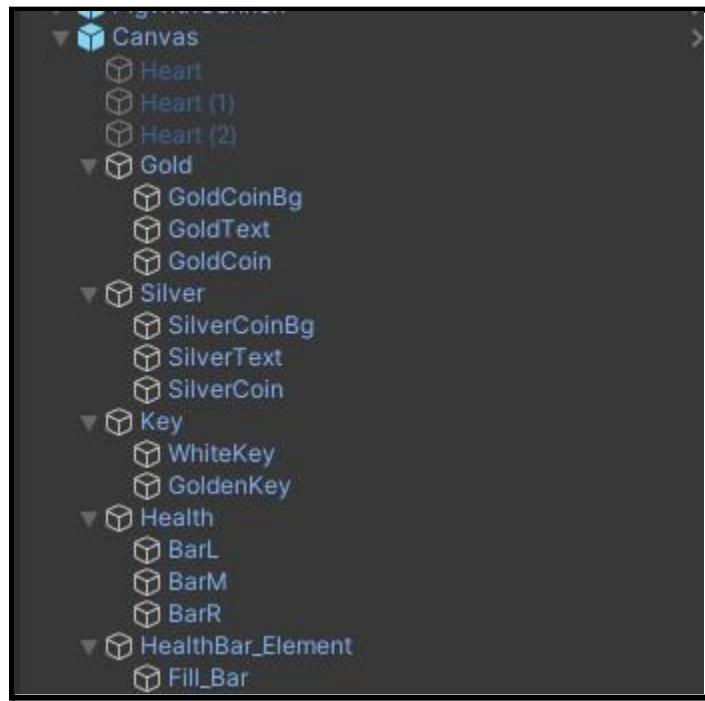
- These are all the traps I am using in this project or game.
- All traps used in different levels and locations.
- All traps are different from each other, some do damage and some do fall when the player collides with the objects like: Falling platform when we touch the platform its fall to the ground and player dead.

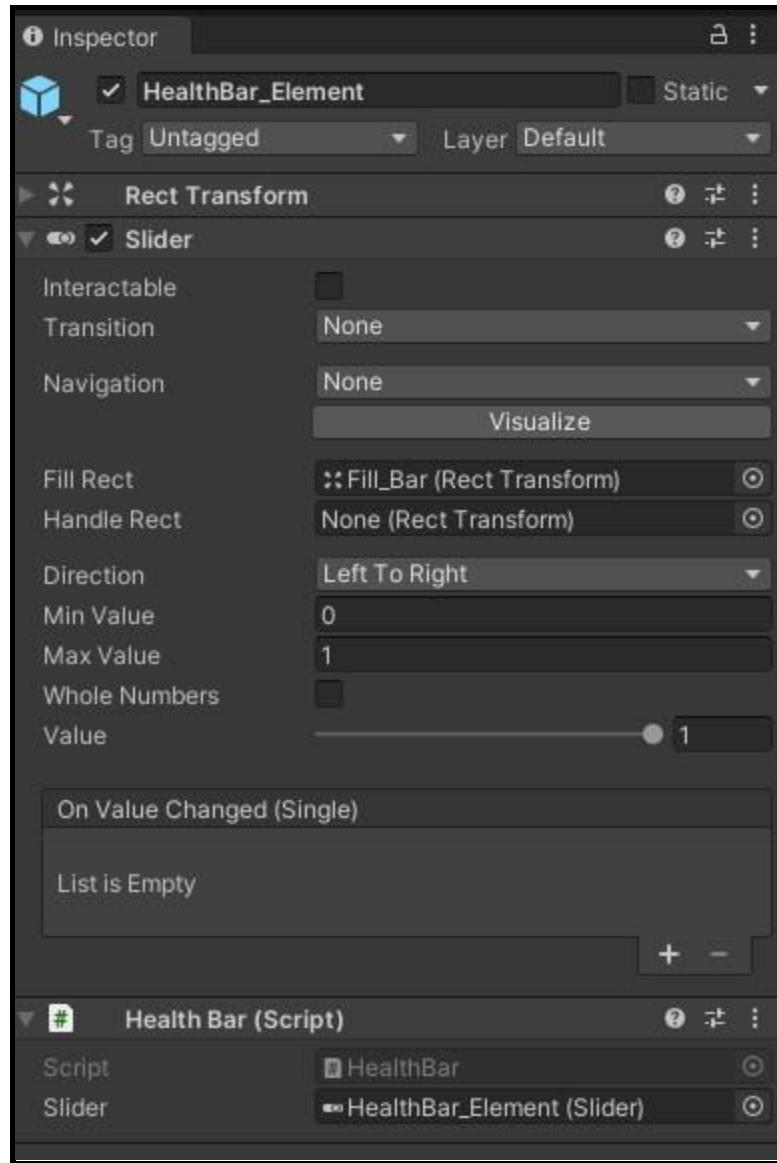
Game Camera



- Installing Cinemachine for game Camera.
- Cinemachine is simple to use, so we can use and follow the character without coding.
- First Go to window>Package Manager
- After the Package Manager window opens, search for cinemachine and click on install.
- After install you see Cinemachine in the top menu click on it then Free 2D camera.

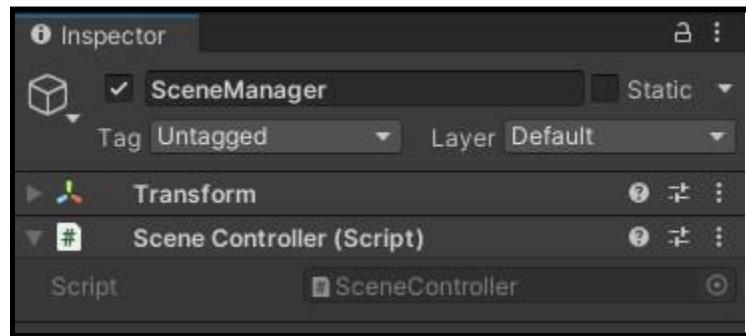
Game HUD



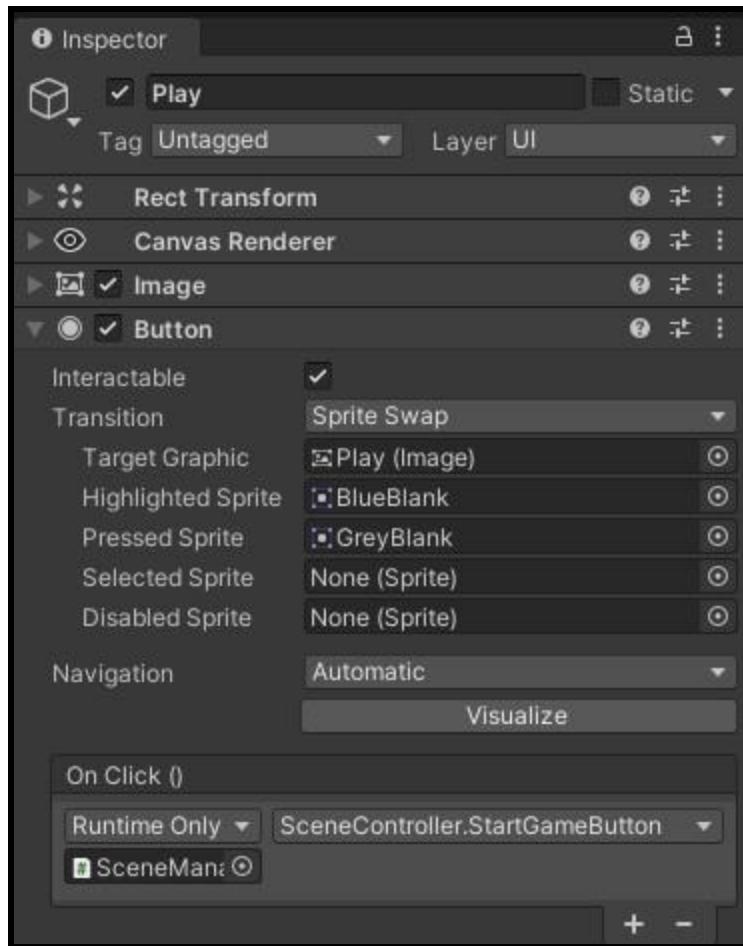


- Script only used in the health element to control the slider value.
- With The script we can access the slider value element and set the max and min value for the slider.

Main Screen

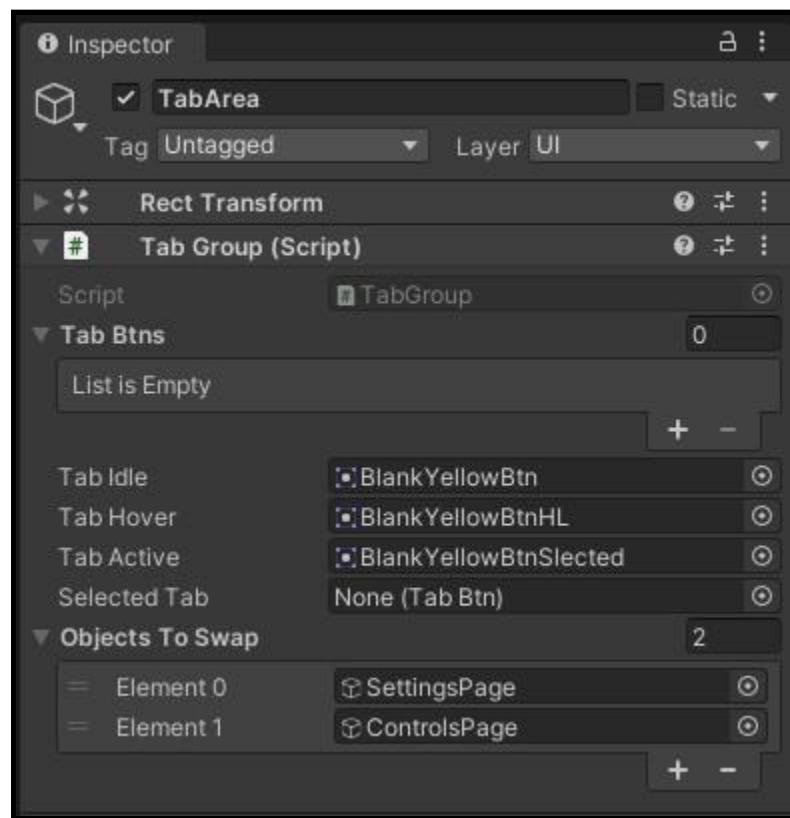
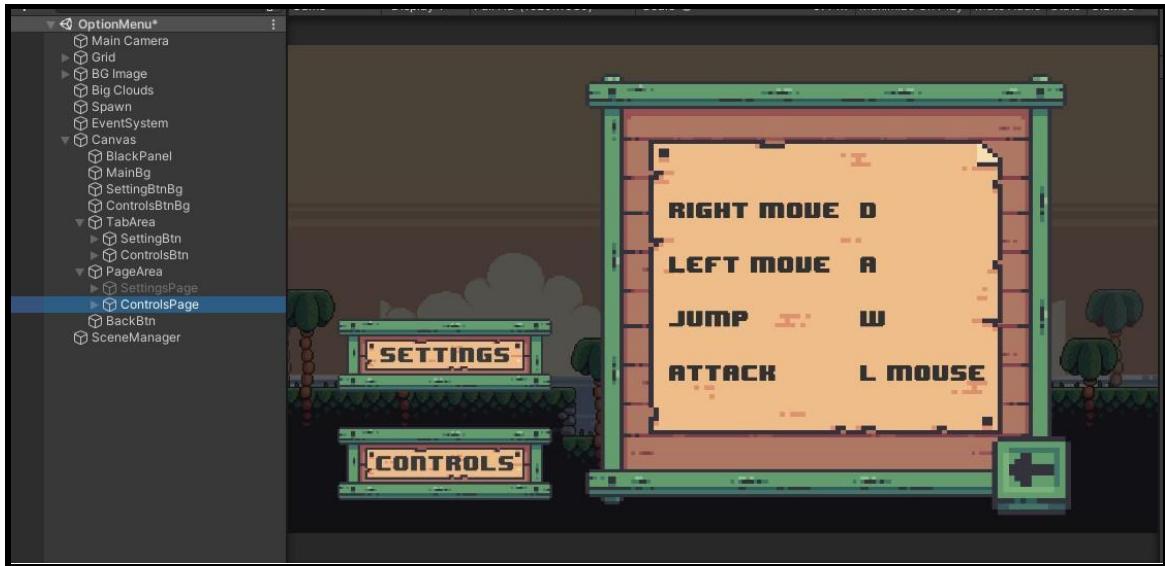


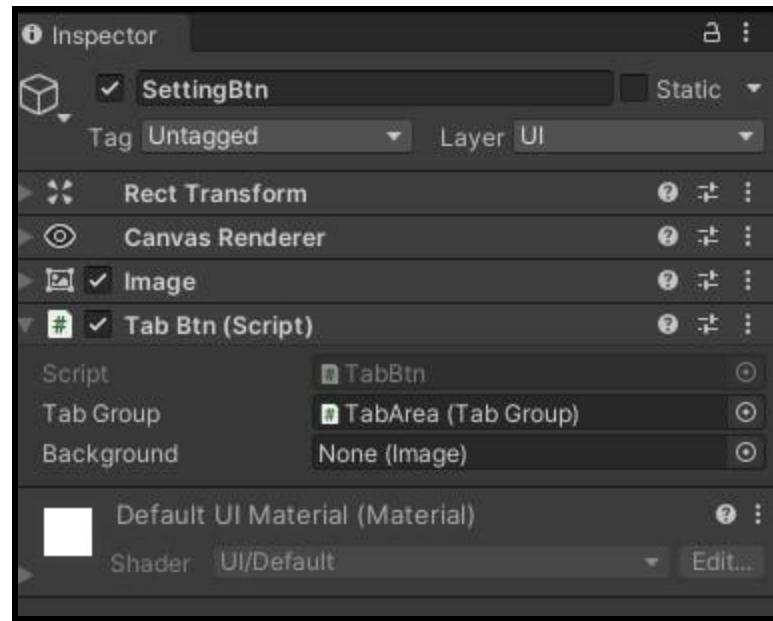
- Scene Manager has all the script and manages all the scene management through the script but its not working until not attach with the buttons.



- On click event add an event click on the add icon under the tab.
- then drag and drop the game object attached with the scene manager script and choose the function.
- All the public functions show in this tab.

Option Screen

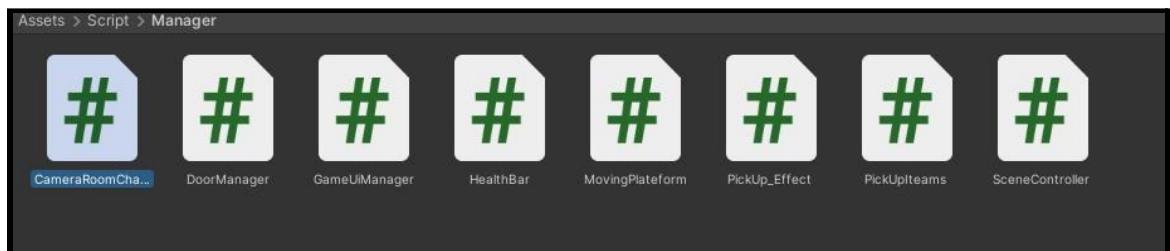


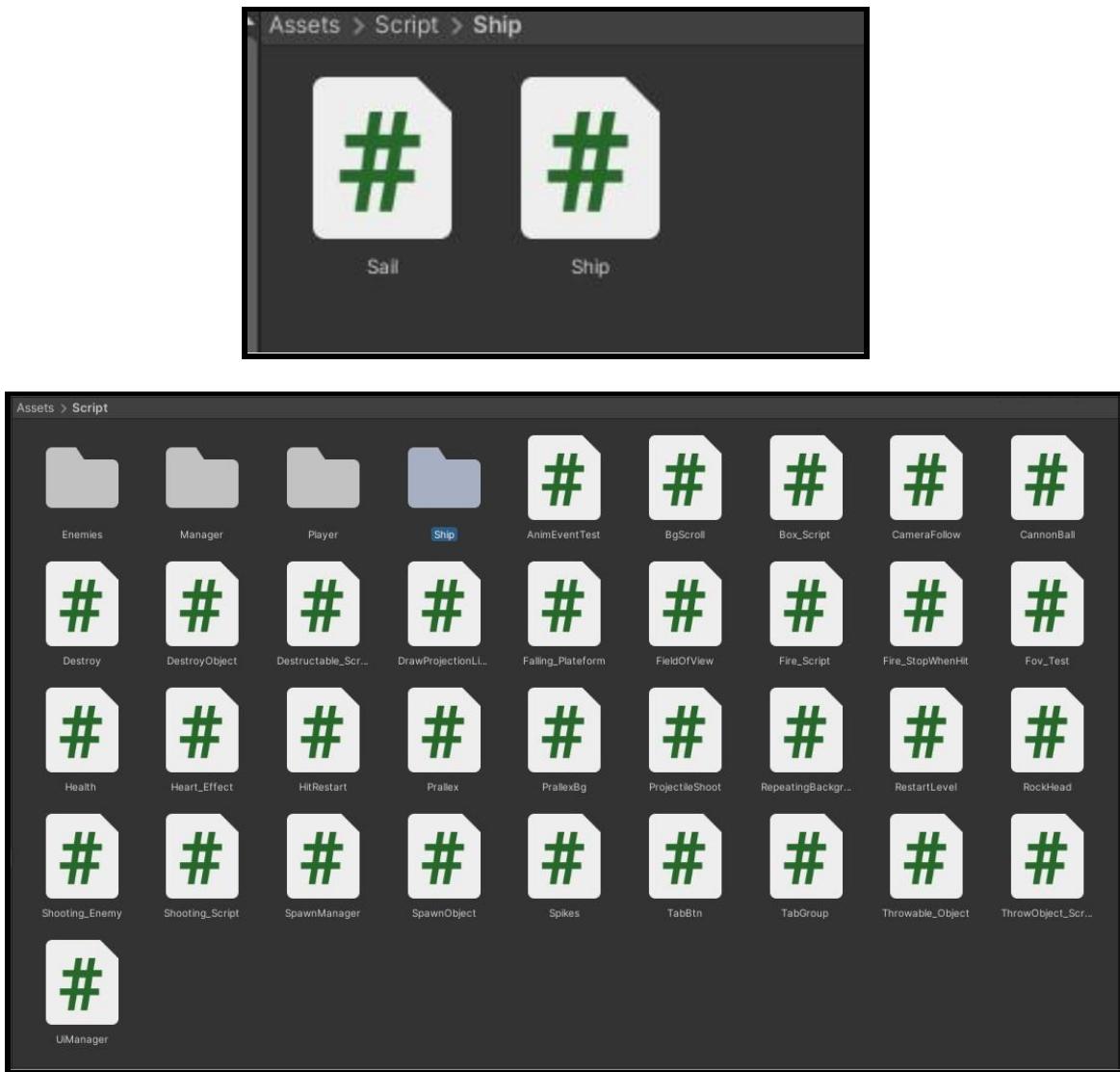


- For this I am using the tab system.
- It's a like Button and a page system with this we make many pages and the buttons for all of the tabs.

All Scripts

In This project I am making many scripts. It's not possible to screenshot all the script and paste here. But I will give you an idea.





- All Scripts (Unity Project Window View)

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  @UnityScript | 26 references
7  public class PlayerBasic : MonoBehaviour
8  {
9      [Header("Movement")]
10     [SerializeField] private float speed;
11     [SerializeField] private float jumpForce;
12     [SerializeField] private float groundCheckerRadius;
13     [SerializeField] private Transform groundChecker;
14     [SerializeField] private LayerMask whatIsGround;
15     private bool playerIsOnGrounded = false;
16     private bool resetJump = false;
17     public bool p_FacingRight = true;
18     private Rigidbody2D rb2d;
19
20     [Space(20)]
21     [Header("Attack")]
22     [SerializeField] private float attackRange;
23     [SerializeField] private float attackRate;
24     [SerializeField] private Transform attackPoint;
25     [SerializeField] private LayerMask enemyLayers;
26     private float nextAttackTime = 0f;
27
28     [Space(20)]
29     [Header("Health")]
30     public int maxHealth = 100;
31     public int currentHealth;
32     public HealthBar healthBar;
33
34
35
36
37     private float flameTimer = 1;
38     private float currentTimer;
39
40
41     [Space(20)]
42     [Header("Script Reference")]
43     [SerializeField] private PlayerAnimation playerAnim;
44     [SerializeField] private GameUiManager gameUiManager;
45     [SerializeField] private EnemyMelee_Script enemyMelee_Script;
46     [SerializeField] private PigWithMatch_Script pigWithMatch_Script;
47     [SerializeField] private EnemyThrowable_Script enemyThrowable_Script;
48     [SerializeField] private Fire_StopWhenHit fire_StopWhenHit;
49     [SerializeField] private Box_Script box_Script;
50     [SerializeField] private Enemy enemy;
51

```

- Player Script (Variables)

```
52     [+] @ Unity Message | 0 references
53     void Start()...
54
55     [+] @ Unity Message | 0 references
56     void Update()...
57
58     [+] 1 reference
59     void Movement()...
60
61     [+] 4 references
62     public bool isGrounded()...
63
64     [+] 2 references
65     private void Flip()...
66
67
68     //When attack key down player do attack
69     [+] 1 reference
70     public void PlayerAttack()...
71
72
73     [+] 1 reference
74     public void AttackOnTraps()...
75
76
77     [+] 12 references
78     public void TakeDamage(int damage)...
79
80
81     [+] 1 reference
82     public void HappenWhenDoorIn()...
83
84
85     [+] @ Unity Message | 0 references
86     private void OnTriggerEnter2D(Collider2D collision)...
87
88
89     [+] @ Unity Message | 0 references
90     private void OnTriggerStay2D(Collider2D collision)...
91
92
93     [+] 1 reference
94     IEnumerator ResetJump()...
95
96
97     // To see selected Gizmo in the scene view and adjust the radius or value of selected object
98     [+] @ Unity Message | 0 references
99     public void OnDrawGizmosSelected()...
100
101
102     [+] @ Unity Message | 0 references
103     public void OnDrawGizmos()...
104
105 }
```

- Player Script (Functions)

```
1  @using System.Collections;
2  [System.Collections.Generic];
3  using UnityEngine;
4
5  // Unity Script | 11 references
6  public class Enemy : MonoBehaviour
7  {
8      [SerializeField] private Transform player;
9      [SerializeField] public Transform ledgewallDetection;
10     [SerializeField] public LayerMask whatIsGround;
11     [SerializeField] private LayerMask playerLayer;
12     [SerializeField] private float raycastRange = 5f;
13     [SerializeField] private float chasingRange;
14     [SerializeField] private float attackRange;
15
16     [SerializeField] private ParticleSystem attackEffect;
17
18     [SerializeField] private int maxHealth = 100;
19     [SerializeField] private int currentHealth;
20
21     [SerializeField] private Animator anim;
22
23     private bool isPlayerHere = false;
24     private bool e_facingRight = true;
25
26     // Unity Message | 0 references
27     private void Awake()...
28
29
30
31     1 reference
32     public void PatrolState()...
33
34
35
36     1 reference
37     public void ChaseState()...
38
39
40
41     1 reference
42     public void IdleState()...
43
44
45
46     2 references
47     public void LookAtPlayer()...
48
49
50
51     4 references
52     public void Flip()...
53
54
55
56     1 reference
57     public void EnemyTakeDamage(int damage)...
58
59
60
61     0 references
62     public void AttackEffect()...
63
64
65
66     1 reference
67     private void Die()...
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148 }
```

- Enemy Script (Main script)

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  @UnityScript | 0 references
6  public class Enemy_Idle : StateMachineBehaviour
7  {
8      Enemy enemy;
9
10     // OnStateEnter is called when a transition starts and the state machine starts to evaluate this state
11     @UnityMessage | 2 references
12     override public void OnStateEnter(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)...
13
14
15     // OnStateUpdate is called on each Update frame between OnStateEnter and OnStateExit callbacks
16     @UnityMessage | 2 references
17     override public void OnStateUpdate(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)...
18
19
20     // OnStateExit is called when a transition ends and the state machine finishes evaluating this state
21     @UnityMessage | 2 references
22     override public void OnStateExit(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)...
23
24
25 }
26
27

```

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  @UnityScript | 0 references
6  public class Enemy_Patrole : StateMachineBehaviour
7  {
8      [SerializeField] private float speed = 2.7f;
9
10     Enemy enemy;
11
12     //OnStateEnter is called when a transition starts and the state machine starts to evaluate this state
13     @UnityMessage | 2 references
14     override public void OnStateEnter(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)...
15
16
17     //OnStateUpdate is called on each Update frame between OnStateEnter and OnStateExit callbacks
18     @UnityMessage | 2 references
19     override public void OnStateUpdate(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)...
20
21
22     //OnStateExit is called when a transition ends and the state machine finishes evaluating this state
23     @UnityMessage | 2 references
24     override public void OnStateExit(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)...
25
26
27 }
28
29
30
31

```

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  @UnityScript | 0 references
6  public class Enemy_Run : StateMachineBehaviour
7  {
8      [SerializeField] private float speed = 3;
9      [SerializeField] private float attackRange = 1.3f;
10
11     Transform player;
12     Rigidbody2D rb;
13
14     Enemy enemy;
15
16     // OnStateEnter is called when a transition starts and the state machine starts to evaluate this state
17     @UnityMessage | 2 references
18     override public void OnStateEnter(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)...
19
20
21     // OnStateUpdate is called on each Update frame between OnStateEnter and OnStateExit callbacks
22     @UnityMessage | 2 references
23     override public void OnStateUpdate(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)...
24
25
26
27     // OnStateExit is called when a transition ends and the state machine finishes evaluating this state
28     @UnityMessage | 2 references
29     override public void OnStateExit(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)...
30
31
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

- Enemy All State Machine Scripts

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class PickUpItems : MonoBehaviour
7  {
8      [SerializeField] private int totalGold;
9      [SerializeField] private int totalSilver;
10
11     public int key;
12
13     public Text goldCountText;
14     public Text silverCountText;
15     public HealthBar healthBar;
16
17     public GameObject keyPhoto;
18
19     [Space(20)]
20     [Header("Audio")]
21     [SerializeField] AudioSource playerAudioScourse;
22     [SerializeField] AudioClip coins;
23     [SerializeField] AudioClip gems;
24
25     Heart_Effect heart_Effect;
26     PlayerBasic playerBasic;           //player script
27
28     private void Awake()...
29
30     public void OnTriggerEnter2D(Collider2D pickUpItems)
31     {
32         if (pickUpItems.gameObject.CompareTag("Coin"))...
33
34         if (pickUpItems.gameObject.CompareTag("Silver"))...
35
36             if (pickUpItems.gameObject.CompareTag("Diamond"))...
37
38             if (pickUpItems.gameObject.CompareTag("Skull"))...
39
40             if (pickUpItems.gameObject.CompareTag("Key"))...
41
42             //Regain Health when player health is lower than the current health
43             if (pickUpItems.gameObject.CompareTag("HealthPickUp"))...
44
45         }
46
47     }
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98

```

- All Pickup Items Script

Development Log

13 March, 2021 - 26 March, 2021

- Decide Game Concept
- Decide game Name
- Decide Gameplay
- Collect the Reference Images
- Making One-Pager
- Making Game Assets List
- Game Poster Concept

27 March, 2021 - 7 April, 2021

- Making Game Logic Flow Chart
- Makes the game prototype with basic sprite
- Player Basic Movement

8 April, 2021 - 25 April, 2021

- Game Logo
- Game Icon Concept
- Making Main menu, Setting menu and Pause menu Prototype
- Making all the scenes functionality
- Adding HUD and score system in prototype game
- Start Making the GDD

24 April, 2021 - 28 April, 2021

- Game Logo Final
- Game Poster Concept
- Importing All assets
- Arrange all the assets in separate folder according categories

29 April, 2021

- Setup the unity project according to the game

1 May, 2021

- Making Player All Animation
- Setup player animation

3 May, 2021 - 6 May, 2021

- Working on player script
 - Player Movement
 - Player Jump
 - Player Attack

8 May, 2021 - 11 May, 2021

- Player Animation now in action (Idle, Run, Jump, Attack)
- Fix the jump animation

15 May, 2021 - 25 May, 2021

- Design starting 5 levels

26 May, 2021 - 28 May, 2021

- Adding game HUD
- Adding Player Health
- Adding coins and other pick up objects

1 June, 2021- 10 June, 2021

- Adding 3 types of Enemy in the game
- Setup enemy animation (Idle, Run, Attack, Hit, Dead)
- Making Enemy Ai
- Making enemy Field of view Script
- Adding Player Hit Animation

12 June, 2021 - 20 June, 2021

- Adding 2 new enemies with animation
- Makin script for them
- Adding Hit and Dead system for player and enemies
- Adding Game Ui
- Adding some new pickup

22 June, 2021 - 30 June, 2021

- Fix Enemy Ai
- Adding traps in the game and their animation
- Adding new environment in the game
- Make next 2 level
- Adding Moving Platform
- Adding Enemy health
- Update UI

2 July, 2021 - 10 July, 2021

- Adding 4 new enemies and their animation
- Adding water and ship in the game
- Makes 3 more level with new tiles map
- Update Enemy Ai for new enemies
- Update Player script

11 July, 2021 - 18 July, 2021

- Check all the levels
- Adding Cinemachine camera
- Makes Final adjustment
- Finish the game

Conclusion (Learning Outcome)

In the journey of this project I am learning many new things and new tricks to solve the problem. This project takes a long time but it's worth it, because it teaches me a lot of things. I have completed all the objectives that I set for this project and learn more than that.

- I am learning how to work with the state machine to make the enemy Ai or Ncp in the game. State machine is very useful for making these things. because we can switch one state to another state and make a complex enemy AI or ncp behaviour for all conditions. For example: How to react to Ncp when a player is hurt or a player needs healing.
- Learning how to make an effect with the sprite sheet and how to spawn it with the help of a particle system. In this project I am learning how to make a sprite sheet particle system.
- In this project I am learning how to manage a big game project. when we work alone on that project like indie developers.
- New thing that I learned through this whole process is making a complex UI system and Tab system for the game. Tab system is very useful when we make a different skill set for the player. We set one tab each of the skills. When the player clicks boost skill it shows all the boost skill page, when he clicks the power skill tab it shows all the power skill only. It's make a Ui more cleaner and understandable.
- All the objectives that I set from the starting of the project. I completed all of it and also learned more than that.

- I am learning how to do coding in a simple and effective way. making an enumerator that is very useful to making a state machine through the script. then you don't need an animation state machine system.

After completing this project I am learning all the important aspects of game design and development. In this time my knowledge increases and makes me a better indie game developer. Now I am ready to take the next step in this field and make better games for the user.

Bibliography

Player Assets, Castle location:

Kings and Pigs : <https://pixelfrog-assets.itch.io/kings-and-pigs>

Island Location and shooting enemies:

Treasure Hunters : <https://pixelfrog-assets.itch.io/treasure-hunters>

Traps and other objects in the game:

Pixel Adventure: <https://pixelfrog-assets.itch.io/pixel-adventure-1>

The assets I downloaded are from Pixel Frog. I modify some assets or make new ones when I need to.

For Audio and sfx, I am downloading royalty free sound form some sites

<https://www.zapsplat.com/sound-effect-category/game-sounds/>

<https://freesound.org/people/GameAudio/packs/13940/>

Script Reference

Github

Youtube

- Brackeys
- Code Monkey
- Game Dev Guide
- Mix and Jam
- Blackthornprod