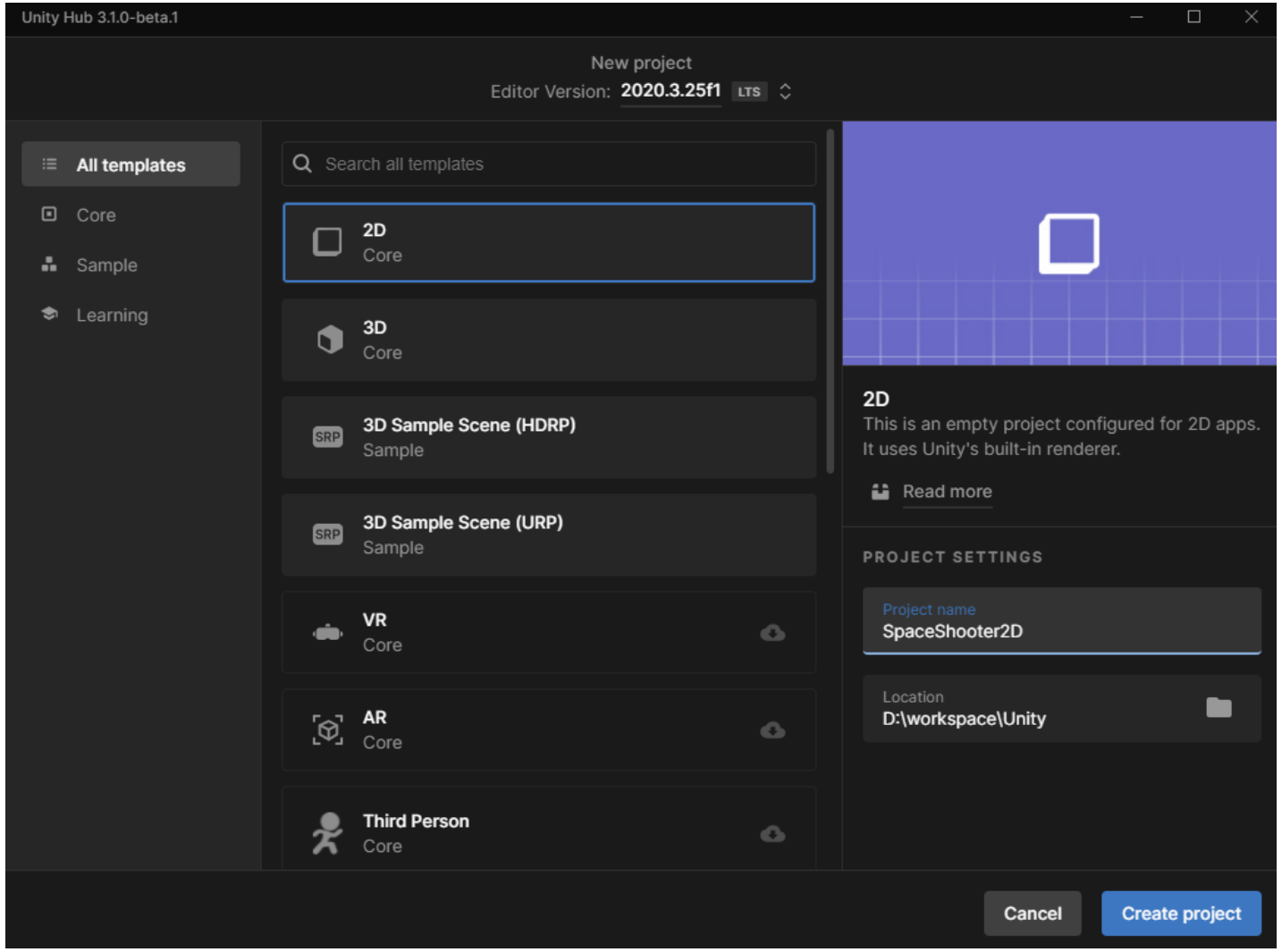
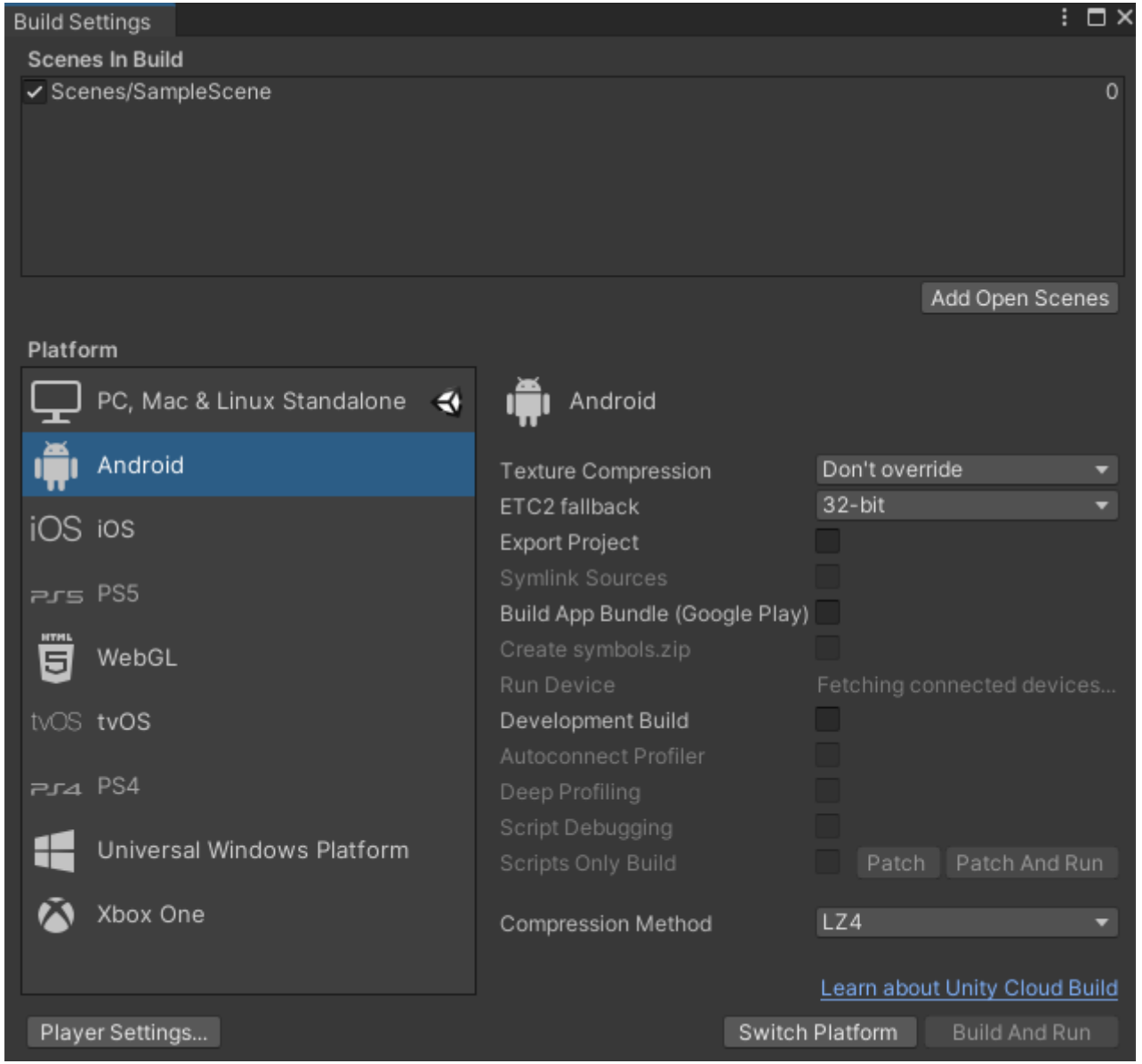
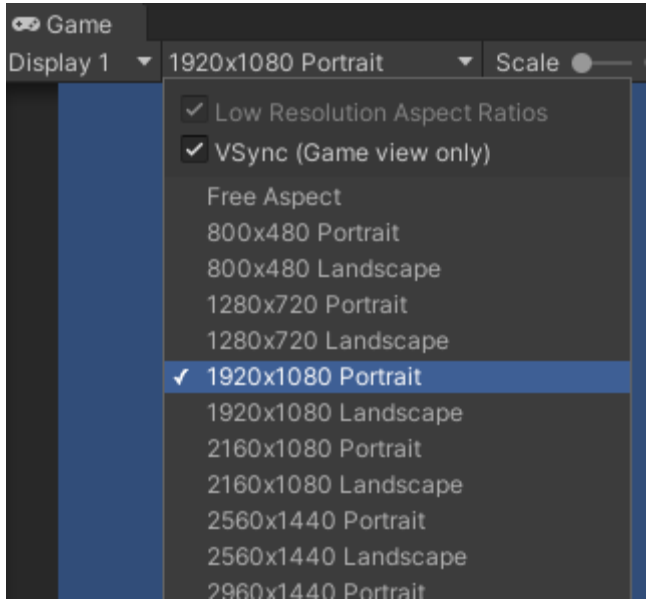
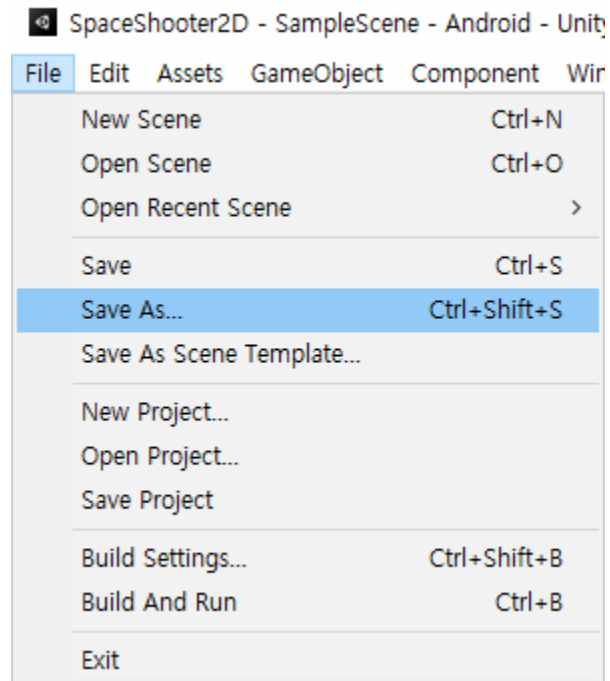
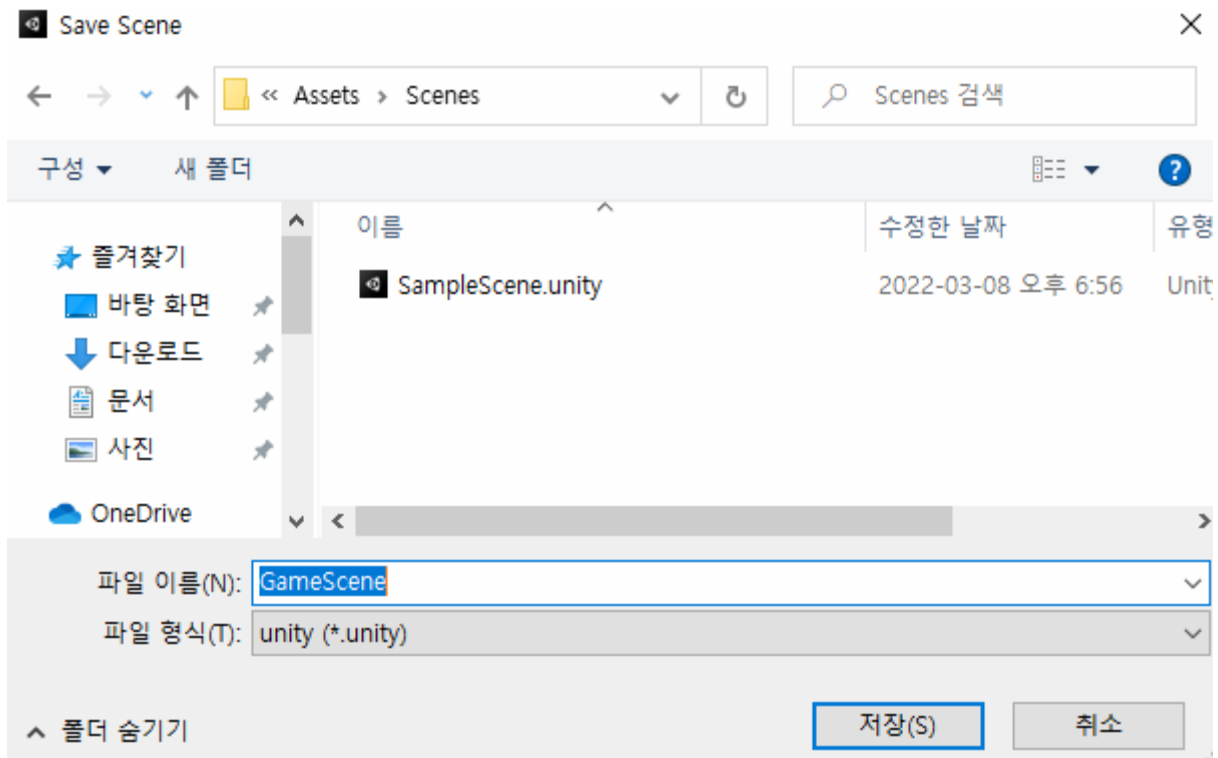
SpaceShooter2D


Unity Hub 3.1.0-beta.1
New project
Editor Version: 2020.3.25f1 LTS
All templates
Core
Sample
Learning
Search all templates
2D
Core
3D
Core
3D Sample Scene (HDRP)
Sample
3D Sample Scene (URP)
Sample
VR
Core
AR
Core
Third Person
Core
2D
This is an empty project configured for 2D apps.
It uses Unity's built-in renderer.
Read more
PROJECT SETTINGS
Project name
SpaceShooter2D
Location
D:\workspace\Unity
Cancel
Create project


Build Settings
Scenes In Build
Scenes/SampleScene
0
Add Open Scenes
Platform
PC, Mac & Linux Standalone
Android
iOS
PS5
WebGL
tvOS
PS4
Universal Windows Platform
Xbox One
Android
Texture Compression
Don't override
ETC2 fallback
32-bit
Export Project
Symlink Sources
Build App Bundle (Google Play)
Create symbols.zip
Run Device
Fetching connected devices...
Development Build
Autoconnect Profiler
Deep Profiling
Script Debugging
Scripts Only Build
Patch
Patch And Run
Compression Method
LZ4
Learn about Unity Cloud Build
Player Settings...
Switch Platform
Build And Run


Game
Display 1
1920x1080 Portrait
Scale
Low Resolution Aspect Ratios
VSync (Game view only)
Free Aspect
800x480 Portrait
800x480 Landscape
1280x720 Portrait
1280x720 Landscape
1920x1080 Portrait
1920x1080 Landscape
2160x1080 Portrait
2160x1080 Landscape
2560x1440 Portrait
2560x1440 Landscape
2960x1440 Portrait


SpaceShooter2D - SampleScene - Android - Unity
File
Edit
Assets
GameObject
Component
Wir
New Scene Ctrl+N
Open Scene Ctrl+O
Open Recent Scene
Save Ctrl+S
Save As... Ctrl+Shift+S
Save As Scene Template...
New Project...
Open Project...
Save Project
Build Settings... Ctrl+Shift+B
Build And Run Ctrl+B
Exit


Save Scene
« Assets › Scenes
Scenes 검색
구성
새 폴더
즐겨찾기
바탕 화면
다운로드
문서
사진
OneDrive
이름
수정한 날짜
유형
SampleScene.unity
2022-03-08 오후 6:56
Unit
파일 이름(N):
GameScene
파일 형식(T):
unity (*.unity)
폴더 숨기기
저장(S)
취소

Unity Asset Store

종스크롤 2D 슈팅

3D 2D Add-Ons 음향 Unity 필수에셋 템플릿 도구 시각 효과 Sale

평점 만점의 리뷰가 11,000개 이상 | 8만 5천명 이상의 리뷰 | 10만명 이

1-1 / 1개 결과 **종스크롤 2D 슈팅** 에 대한 결과

정렬 기준 인기도 결과 보기 24

종스크롤 2D 슈팅 ×

GOLDMETAL

종스크롤 2D 슈팅 에셋 팩

(평가가 충분하지 않습니다) | (114)

FREE 내 에셋에 추가하기

상세 검색

모든 카테고리

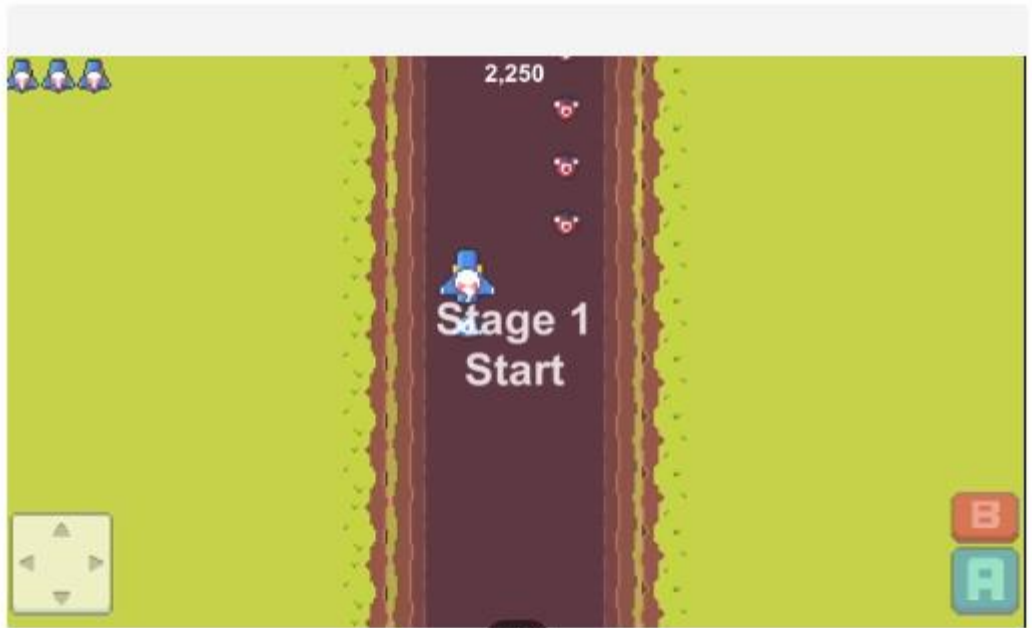
2D (1개)

가격

Unity 버전

퍼블리셔

평점



종스크롤 2D 슈팅 에셋 팩

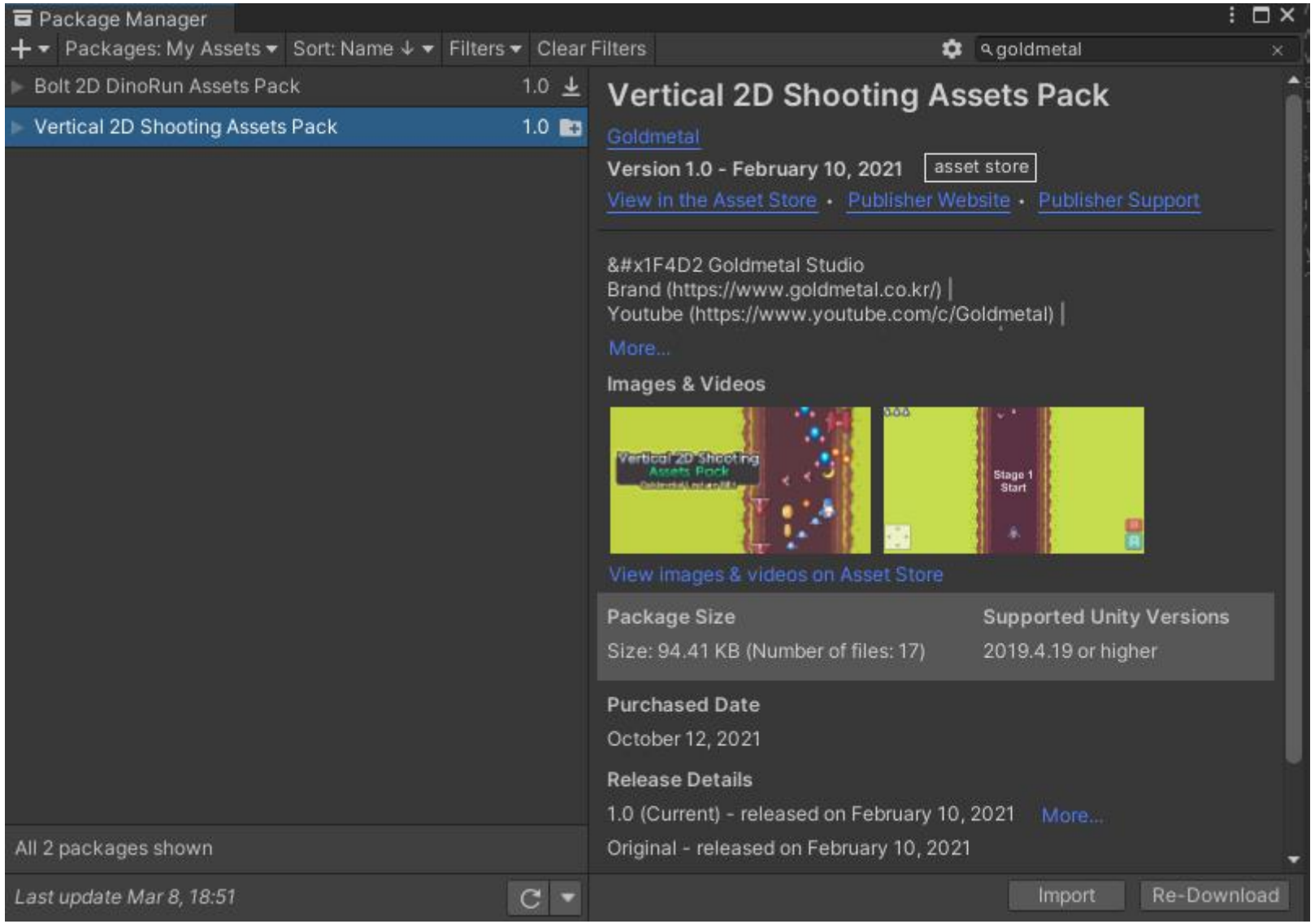
G Goldmetal (평가가 충분하지 않습니다) | ♥ (114)

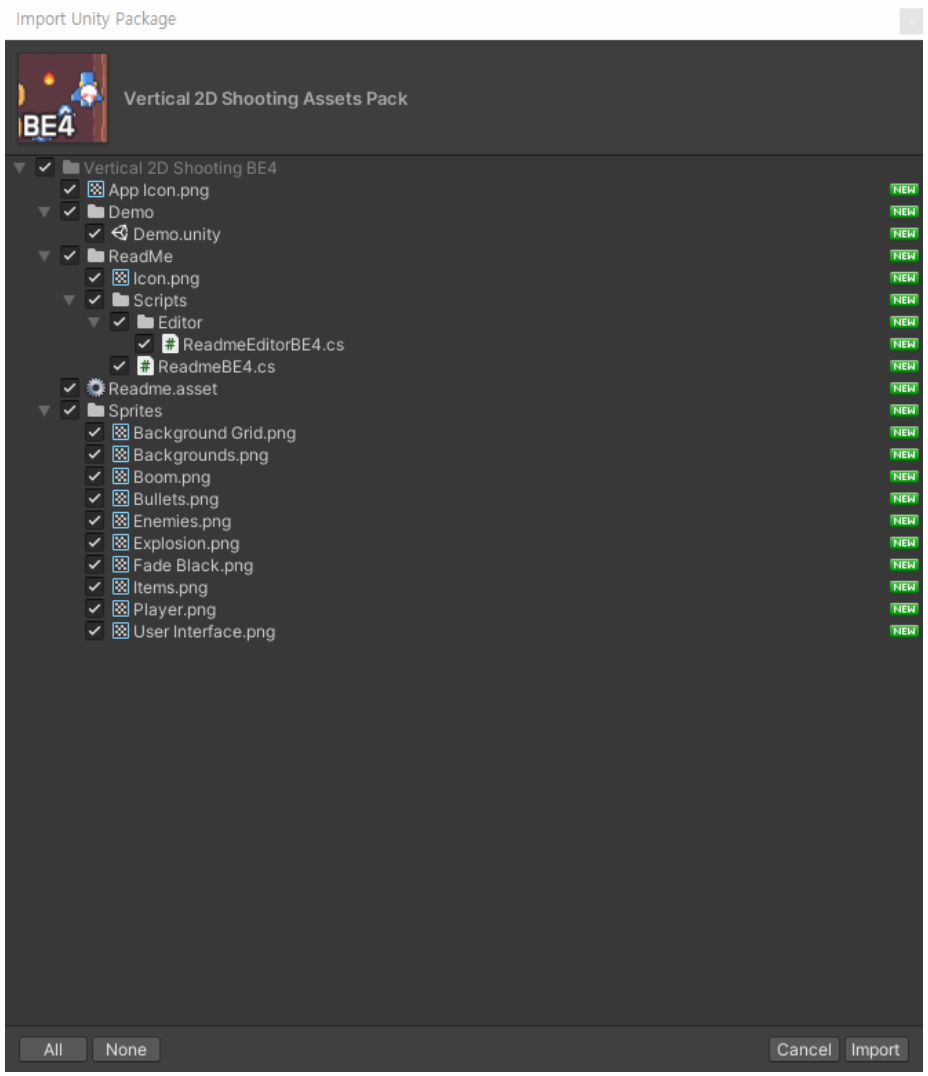
FREE

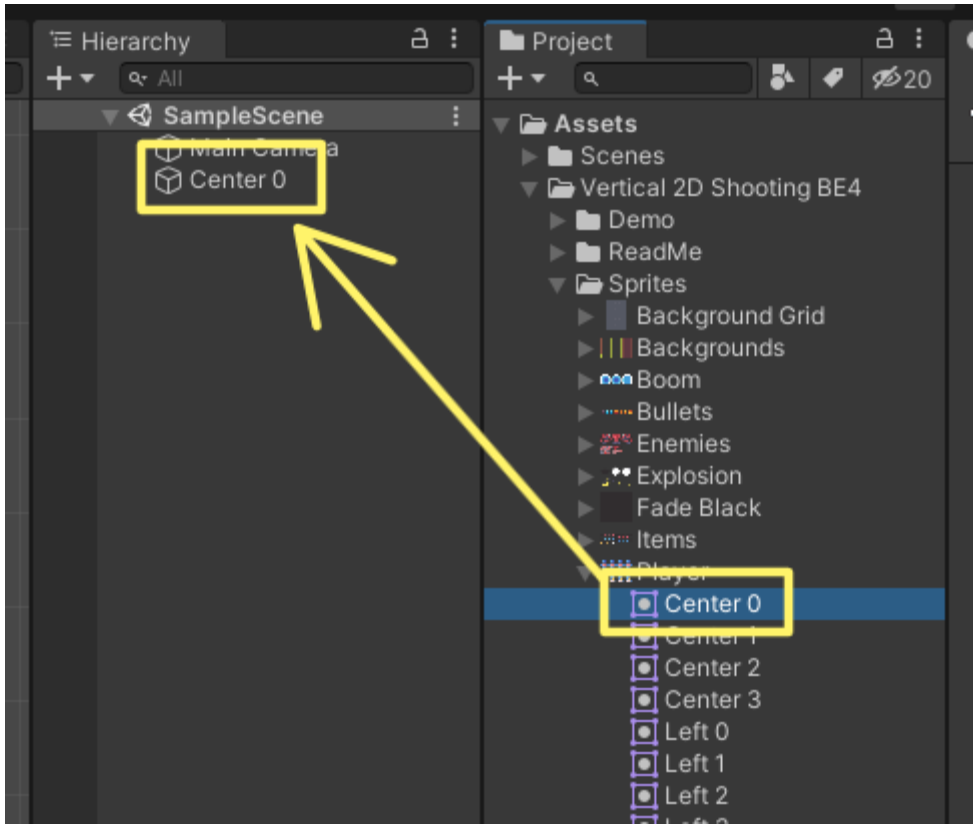
Taxes/VAT calculated at checkout

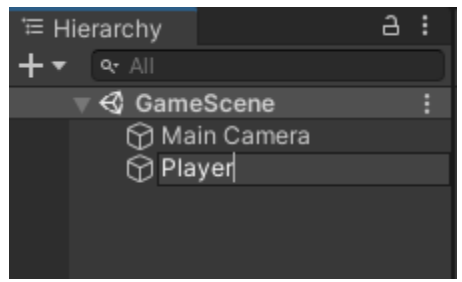
License agreement	Standard Unity Asset Store EULA
파일 크기	94.4 KB
최신 버전	1.0
최신 릴리스 날짜	2021년 2월 10일
지원되는 Unity 버전	2019.4.19 이상
지원	웹사이트 방문

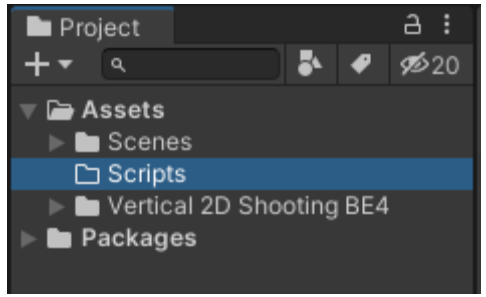
내 에셋에 추가하기 | View Full Details

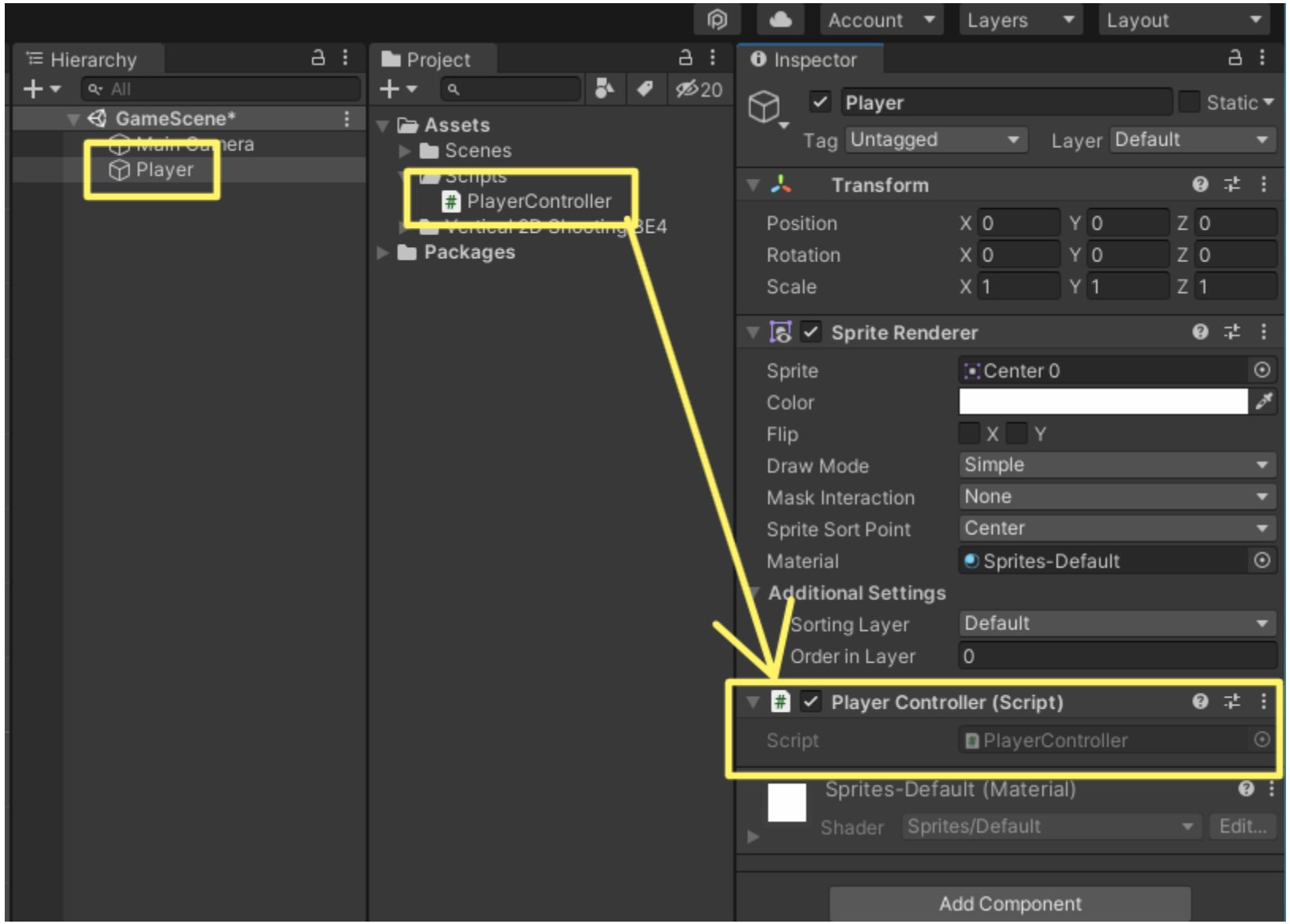

Package Manager
Packages: My Assets
Sort: Name
Filters
Clear Filters
goldmetal
Bolt 2D DinoRun Assets Pack 1.0
Vertical 2D Shooting Assets Pack 1.0
All 2 packages shown
Last update Mar 8, 18:51
Vertical 2D Shooting Assets Pack
Goldmetal
Version 1.0 - February 10, 2021
asset store
View in the Asset Store · Publisher Website · Publisher Support
&#x1F4D2 Goldmetal Studio
Brand (https://www.goldmetal.co.kr/)
Youtube (https://www.youtube.com/c/Goldmetal)
More...
Images & Videos
Vertical 2D Shooting Assets Pack
Stage 1 Start
View images & videos on Asset Store
Package Size
Size: 94.41 KB (Number of files: 17)
Supported Unity Versions
2019.4.19 or higher
Purchased Date
October 12, 2021
Release Details
1.0 (Current) - released on February 10, 2021 More...
Original - released on February 10, 2021
Import
Re-Download

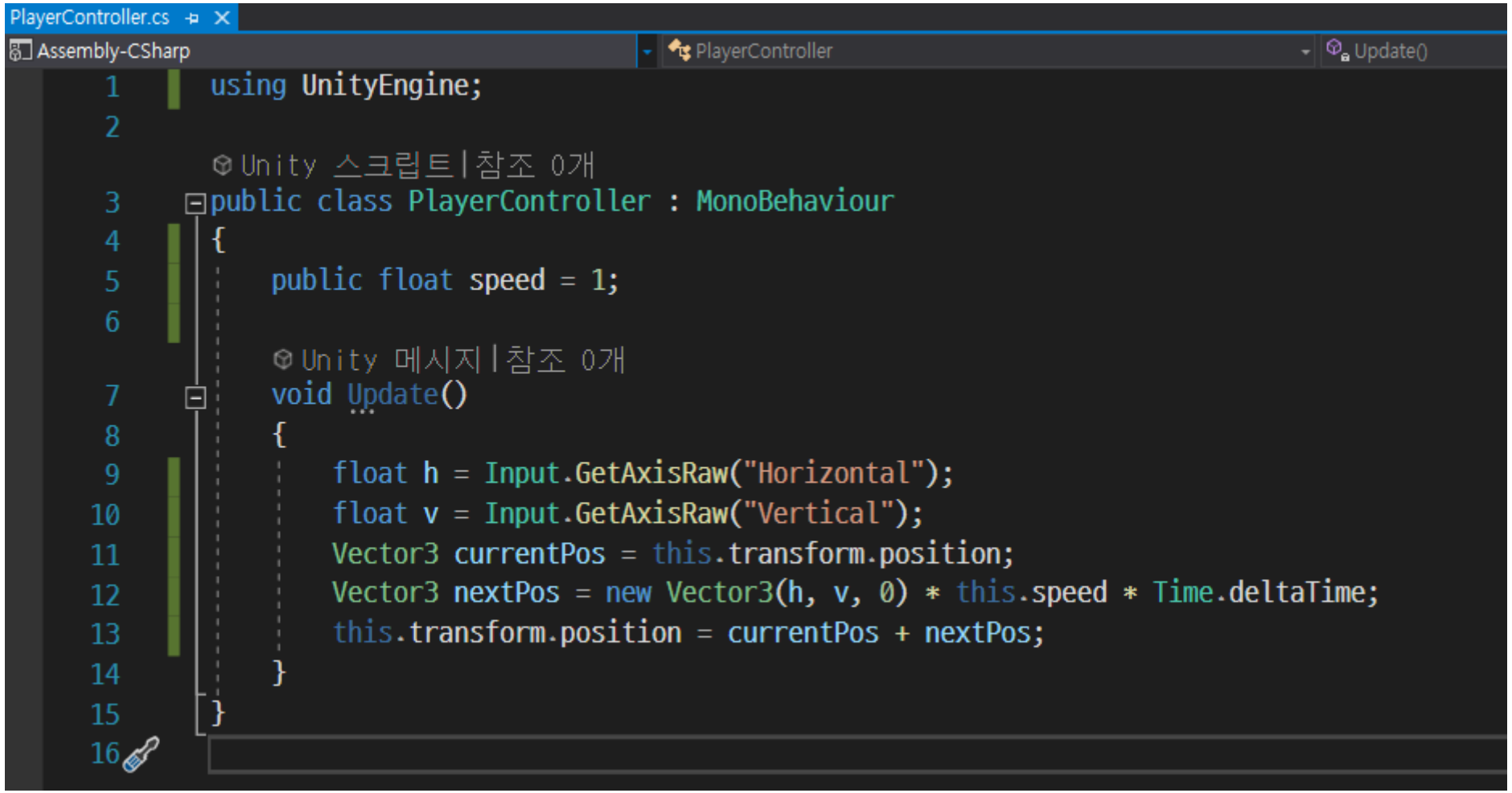

Import Unity Package
Vertical 2D Shooting Assets Pack
Vertical 2D Shooting BE4
App Icon.png
Demo
Demo.unity
ReadMe
Icon.png
Scripts
Editor
ReadmeEditorBE4.cs
ReadmeBE4.cs
Readme.asset
Sprites
Background Grid.png
Backgrounds.png
Boom.png
Bullets.png
Enemies.png
Explosion.png
Fade Black.png
Items.png
Player.png
User Interface.png
NEW
All
None
Cancel
Import


Hierarchy
All
SampleScene
Center 0
Project
20
Assets
Scenes
Vertical 2D Shooting BE4
Demo
ReadMe
Sprites
Background Grid
Backgrounds
Boom
Bullets
Enemies
Explosion
Fade Black
Items
Center 0
Center 2
Center 3
Left 0
Left 1
Left 2

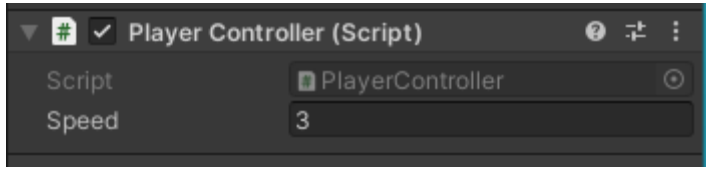

Hierarchy
All
GameScene
Main Camera
Player

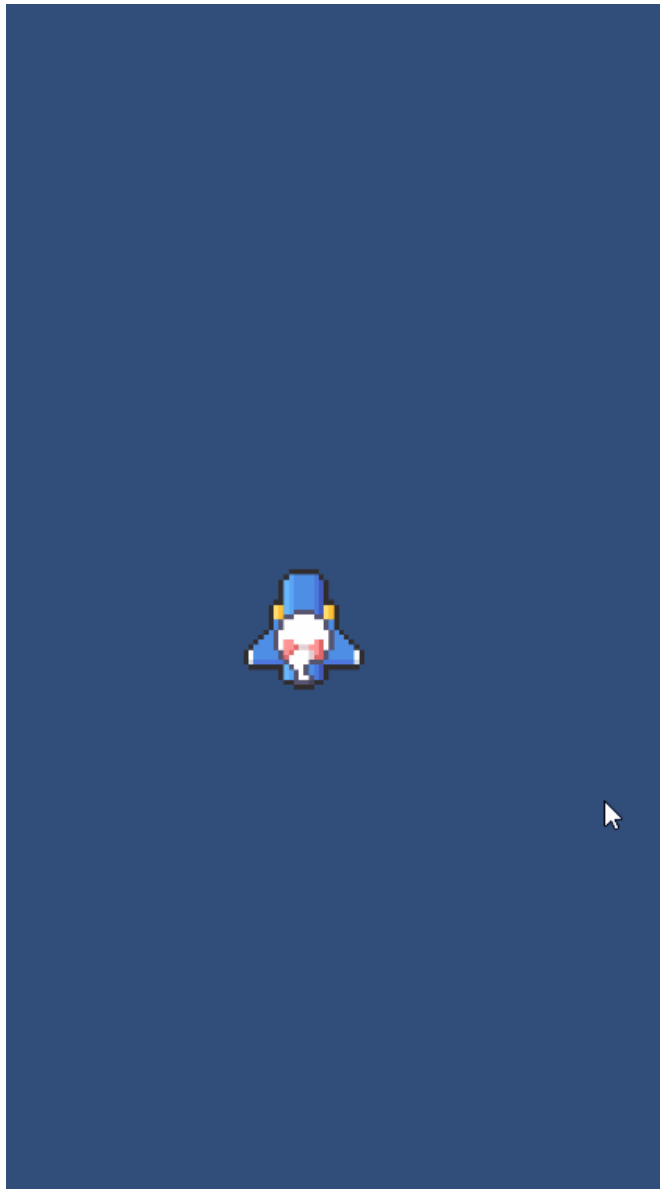

Project
20
Assets
Scenes
Scripts
Vertical 2D Shooting BE4
Packages


Account
Layers
Layout
Hierarchy
All
GameScene*
Player
Project
Assets
Scenes
PlayerController
Packages
Inspector
Player
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
Sprite Renderer
Sprite
Center 0
Color
Flip
Draw Mode
Simple
Mask Interaction
None
Sprite Sort Point
Center
Material
Sprites-Default
Additional Settings
Sorting Layer
Default
Order in Layer
Player Controller (Script)
Script
PlayerController
Sprites-Default (Material)
Shader
Sprites/Default
Edit...
Add Component

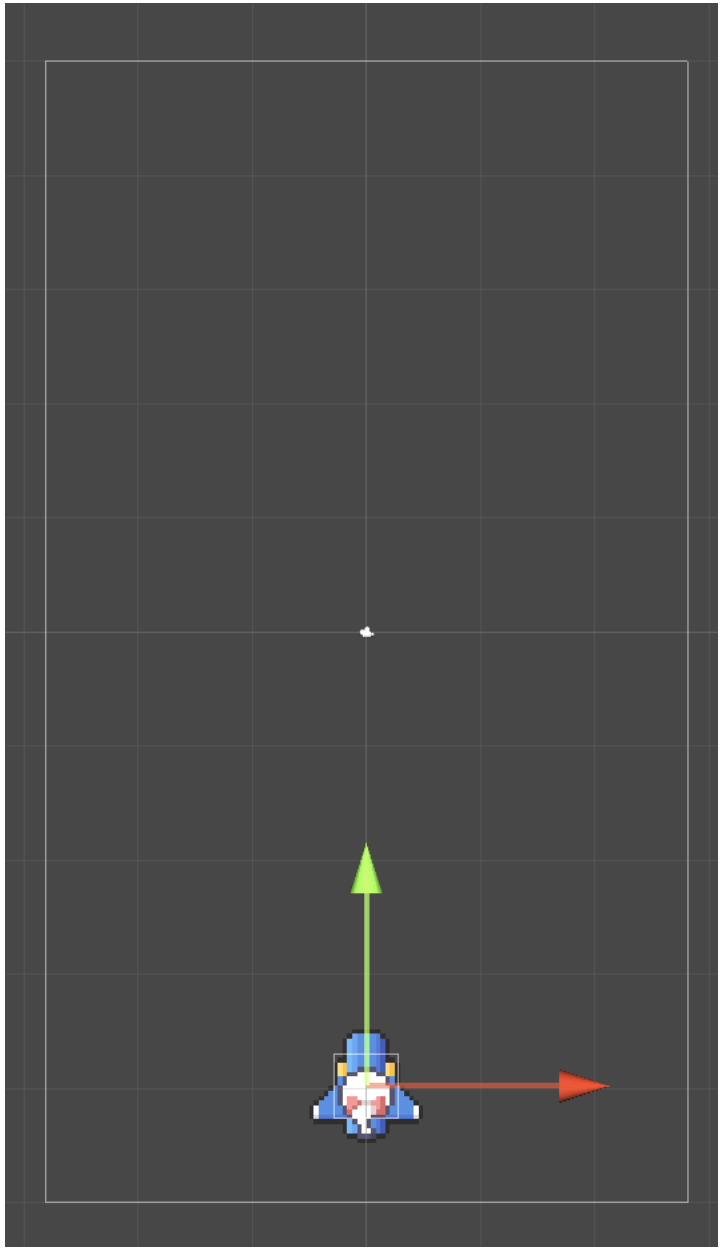



```
Vector3 nextPos = new Vector3(h, v, 0).normalized * this.speed * Time.deltaTime;
```


Player Controller (Script)
Script
PlayerController
Speed
3

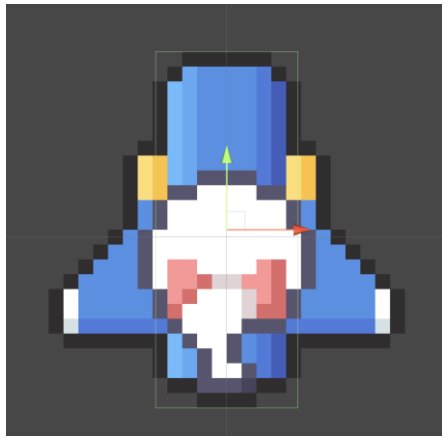


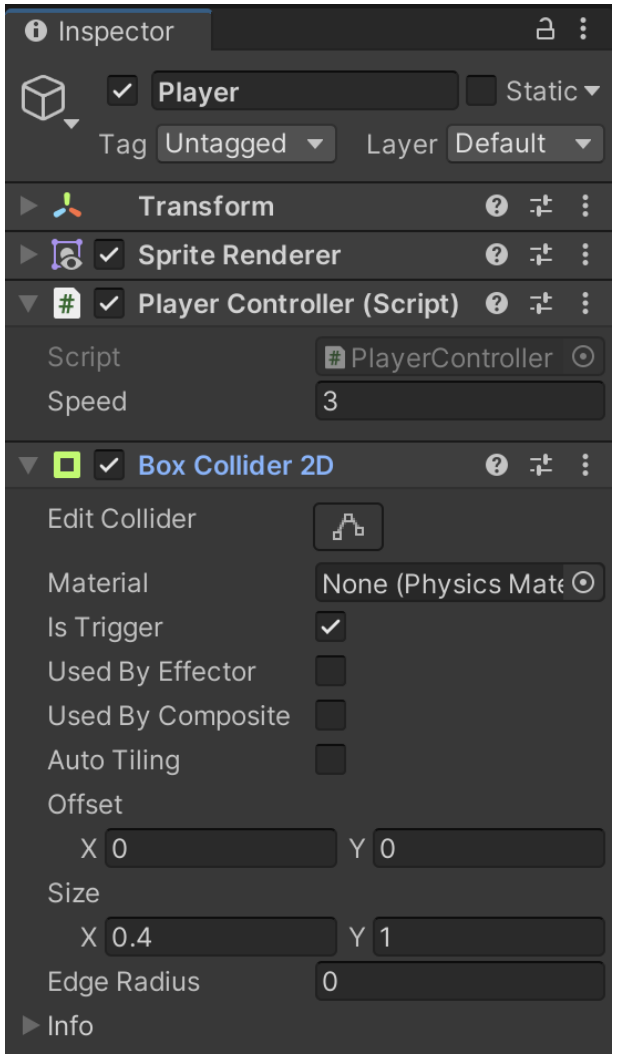
경계 설정



BoxCollider2D컴포넌트를 부착 하고 size를 조절합니다

Is Trigger를 체크 합니다

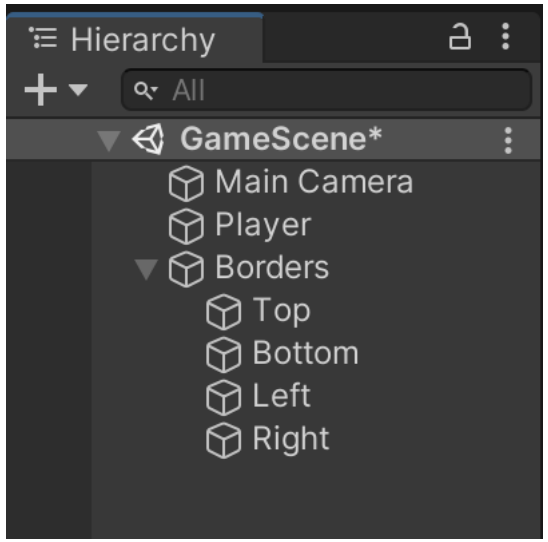


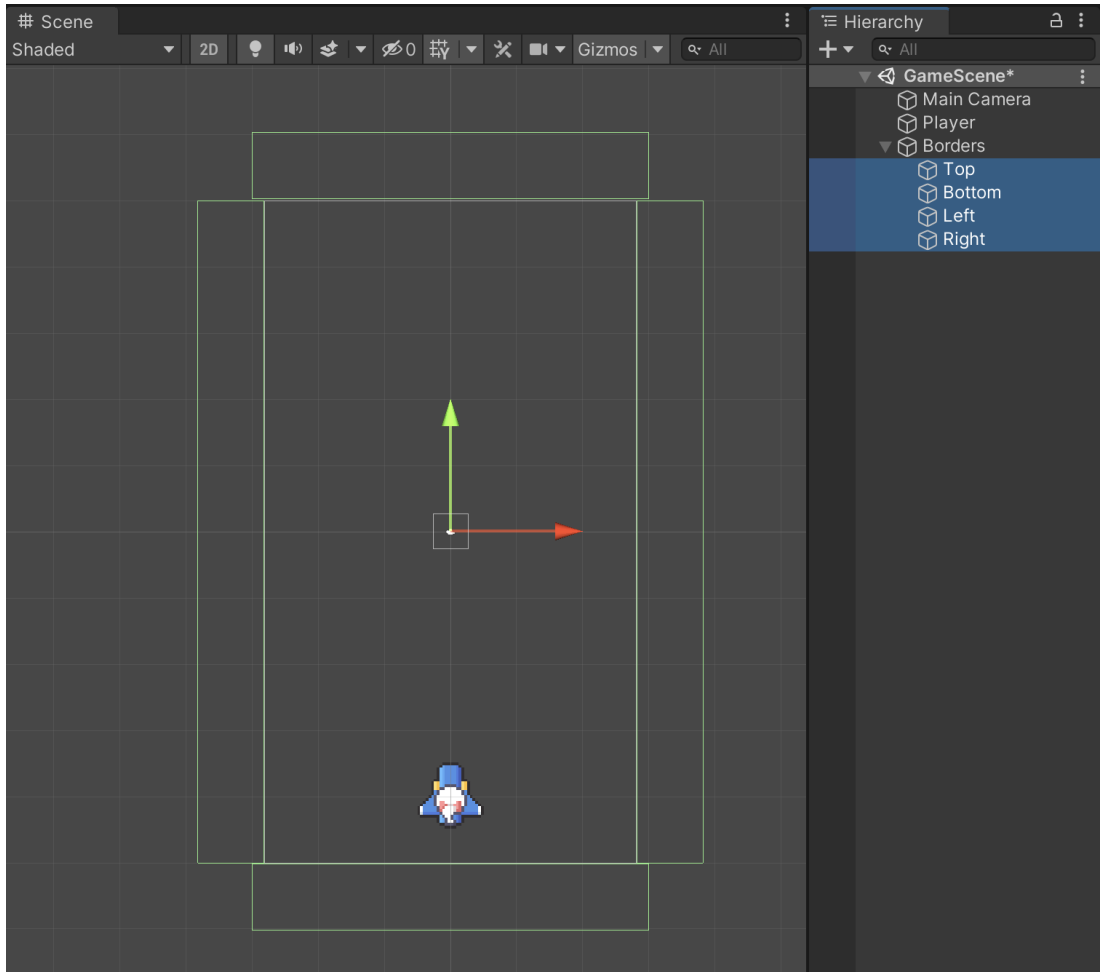


빈 오브젝트(Border)를 만들고

자식으로 4개의 빈오브젝트 (Top, Bottom, Left, Right)를 생성합니다

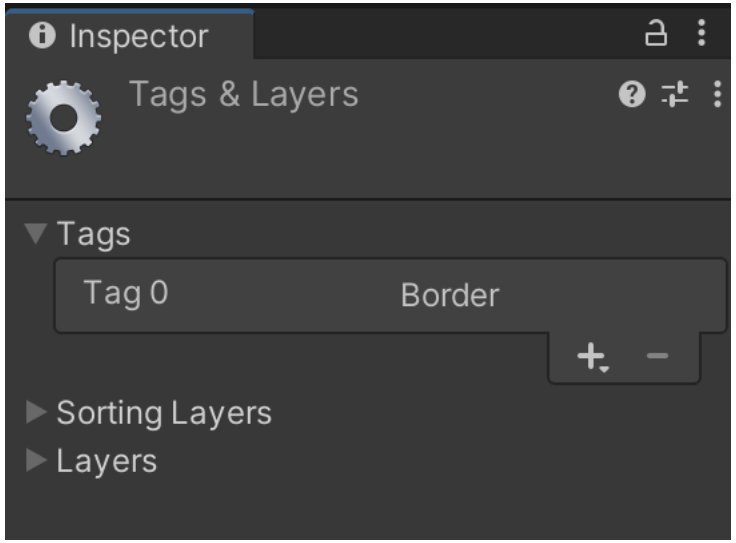
Top, Bottom, Left, Right오브젝트에 BoxCollider2D 컴포넌트를 부착 하고 size와 위치를 설정합니다

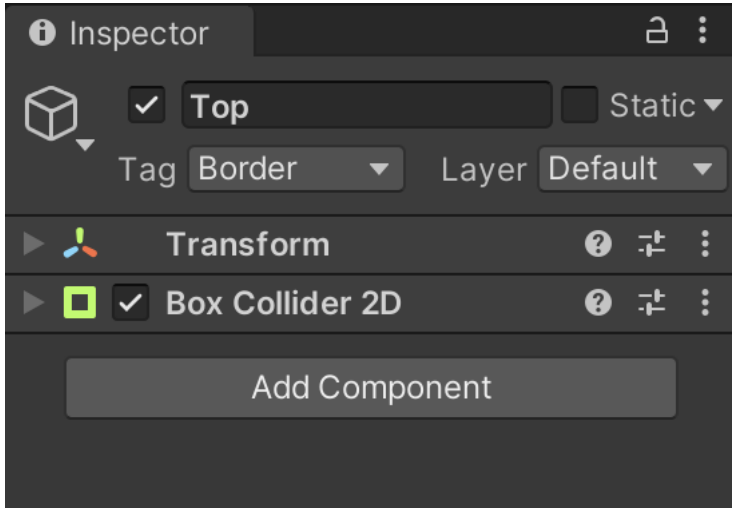




Border태그를 추가 합니다

Top, Bottom, Right, Left의 Tag를 Border로 설정 합니다





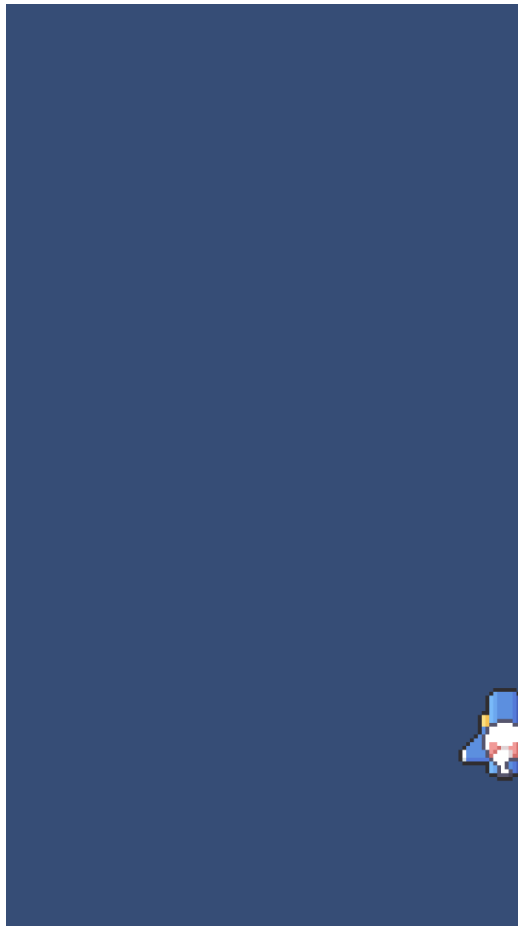
```
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    public float speed = 1;
    private bool isTouchTop;
    private bool isTouchBottom;
    private bool isTouchLeft;
    private bool isTouchRight;

    void Update()
    {
        float h = Input.GetAxisRaw("Horizontal");

        if ((this.isTouchRight && h == 1) || (this.isTouchLeft && h == -1))
        {
            h = 0;
        }

        float v = Input.GetAxisRaw("Vertical");

        if ((this.isTouchTop && v == 1) || (this.isTouchBottom && v == -1))
        {
            v = 0;
        }

        Vector3 currentPos = this.transform.position;
        Vector3 nextPos = new Vector3(h, v, 0).normalized * this.speed * Time.deltaTime;
        this.transform.position = currentPos + nextPos;
    }

```

```
32      private void OnTriggerEnter2D(Collider2D collision)
33      {
34
35          if (collision.gameObject.tag == "Border")
36          {
37              switch (collision.gameObject.name)
38              {
39                  case "Top":
40                      this.isTouchTop = true;
41                      break;
42
43                  case "Bottom":
44                      this.isTouchBottom = true;
45                      break;
46
47                  case "Left":
48                      this.isTouchLeft = true;
49                      break;
50
51                  case "Right":
52                      this.isTouchRight = true;
53                      break;
54              }
55          }
56      }
57
```

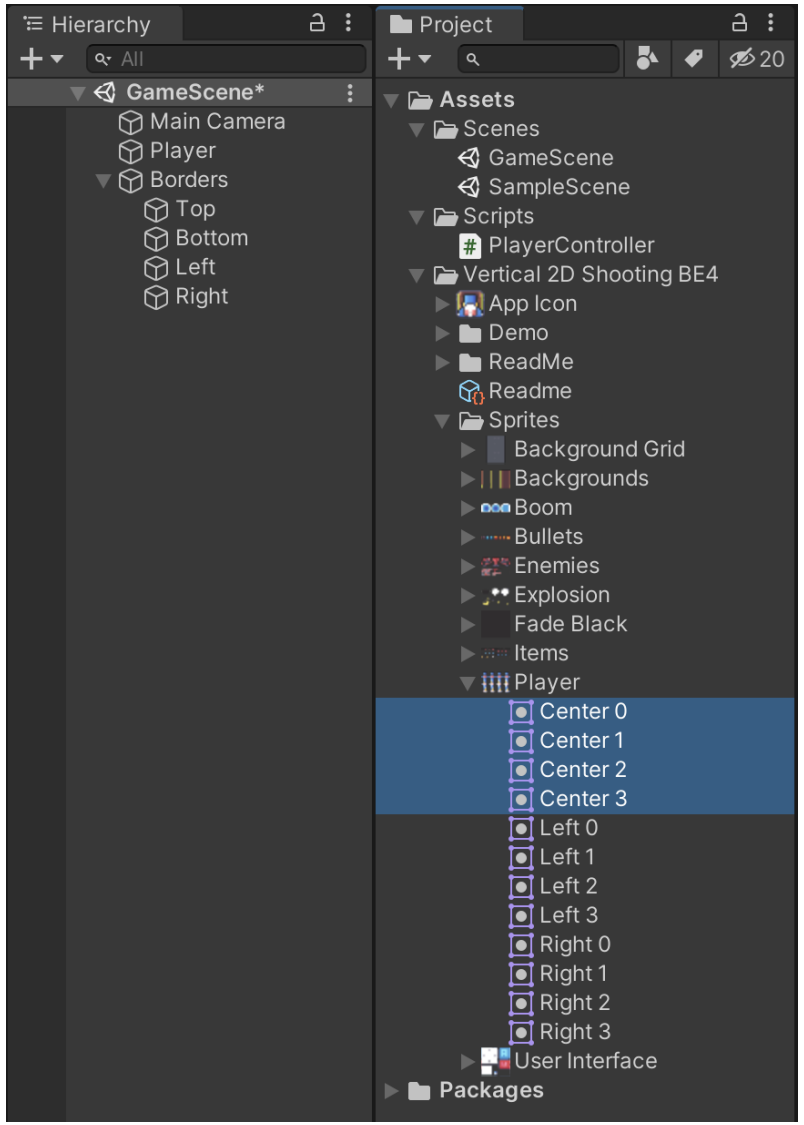
```
58          private void OnTriggerExit2D(Collider2D collision)
59          {
60              if (collision.gameObject.tag == "Border")
61              {
62                  switch (collision.gameObject.name)
63                  {
64                      case "Top":
65                          this.isTouchTop = false;
66                          break;
67
68                      case "Bottom":
69                          this.isTouchBottom = false;
70                          break;
71
72                      case "Left":
73                          this.isTouchLeft = false;
74                          break;
75
76                      case "Right":
77                          this.isTouchRight = false;
78                          break;
79                  }
80              }
81          }
82      }
83
```

실행 후 결과를 확인 합니다

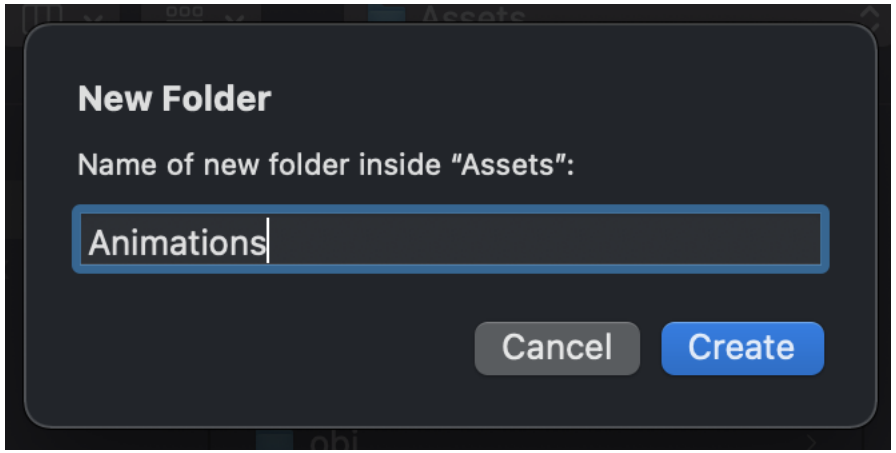
키보드를 이용해 Player를 움직이고 좌우상하 Border를 넘어 가지 않는지 확인합니다



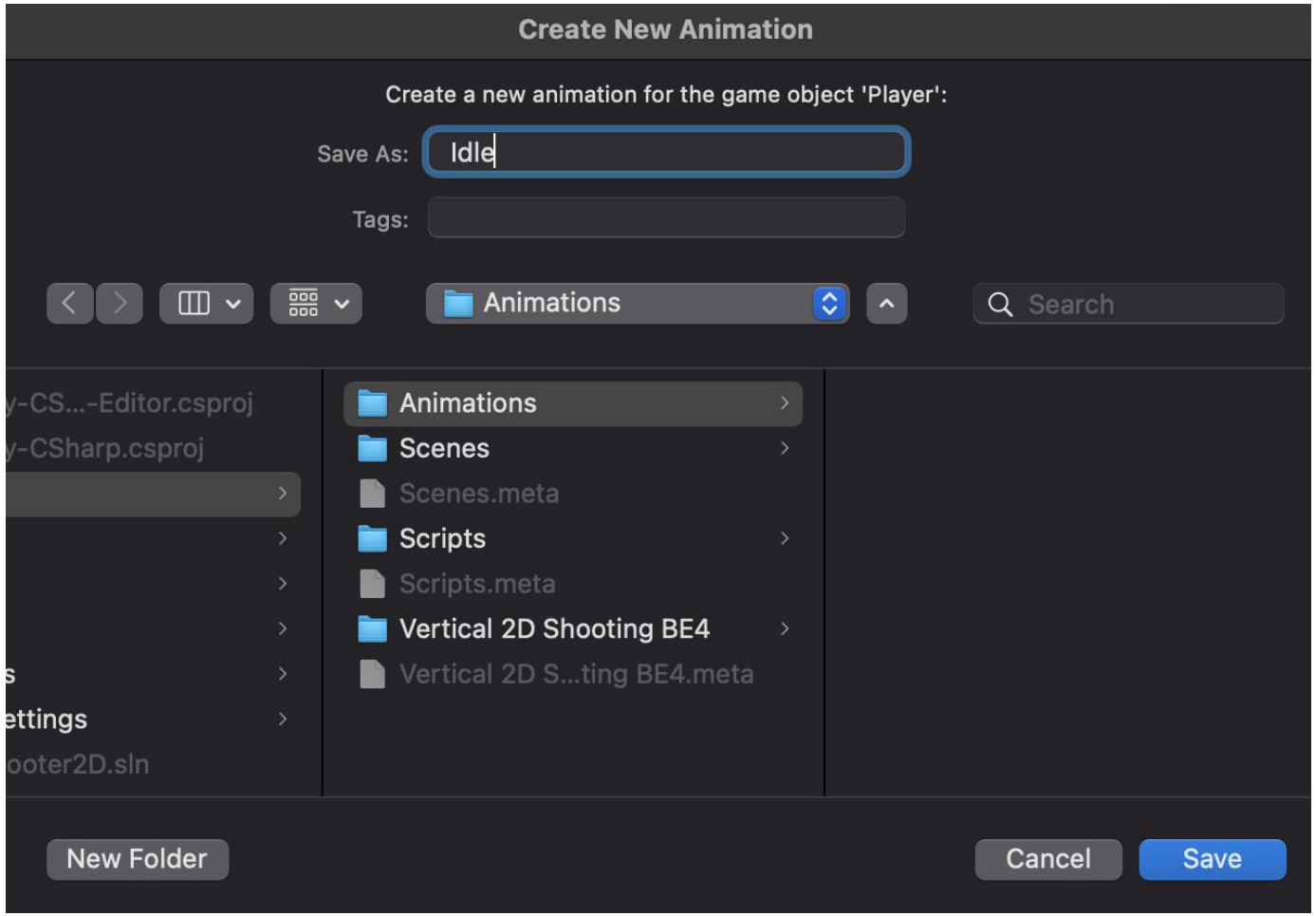
Center0 ~ Center4 스프라이트를 선택해 player에게 드레그&드롭 합니다



Animations폴더를 생성 합니다



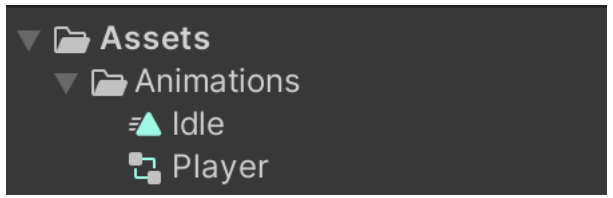
Idle이름으로 저장합니다

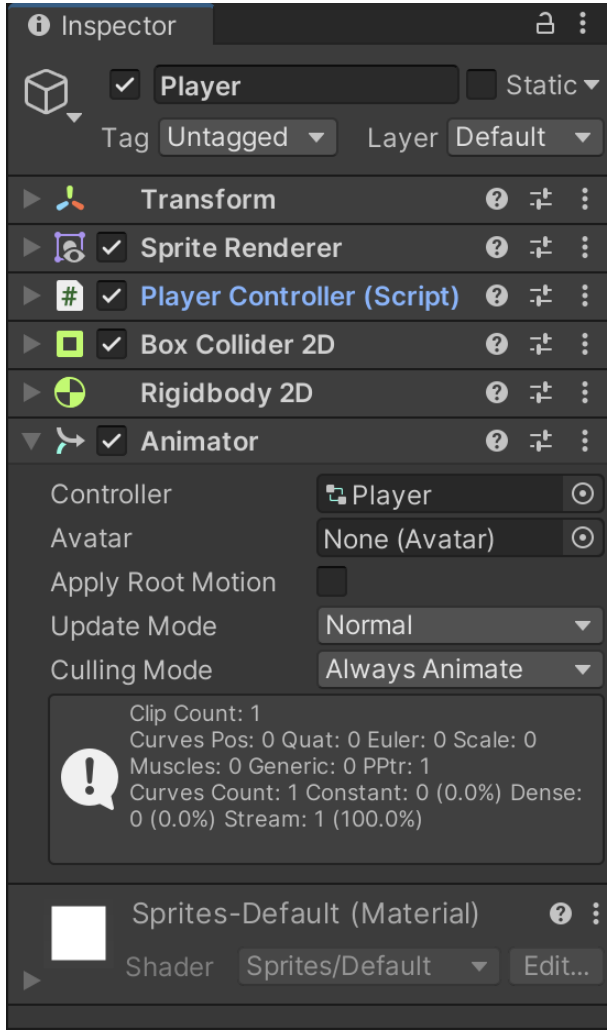


Player에 Animator컴포넌트가 부착되었습니다

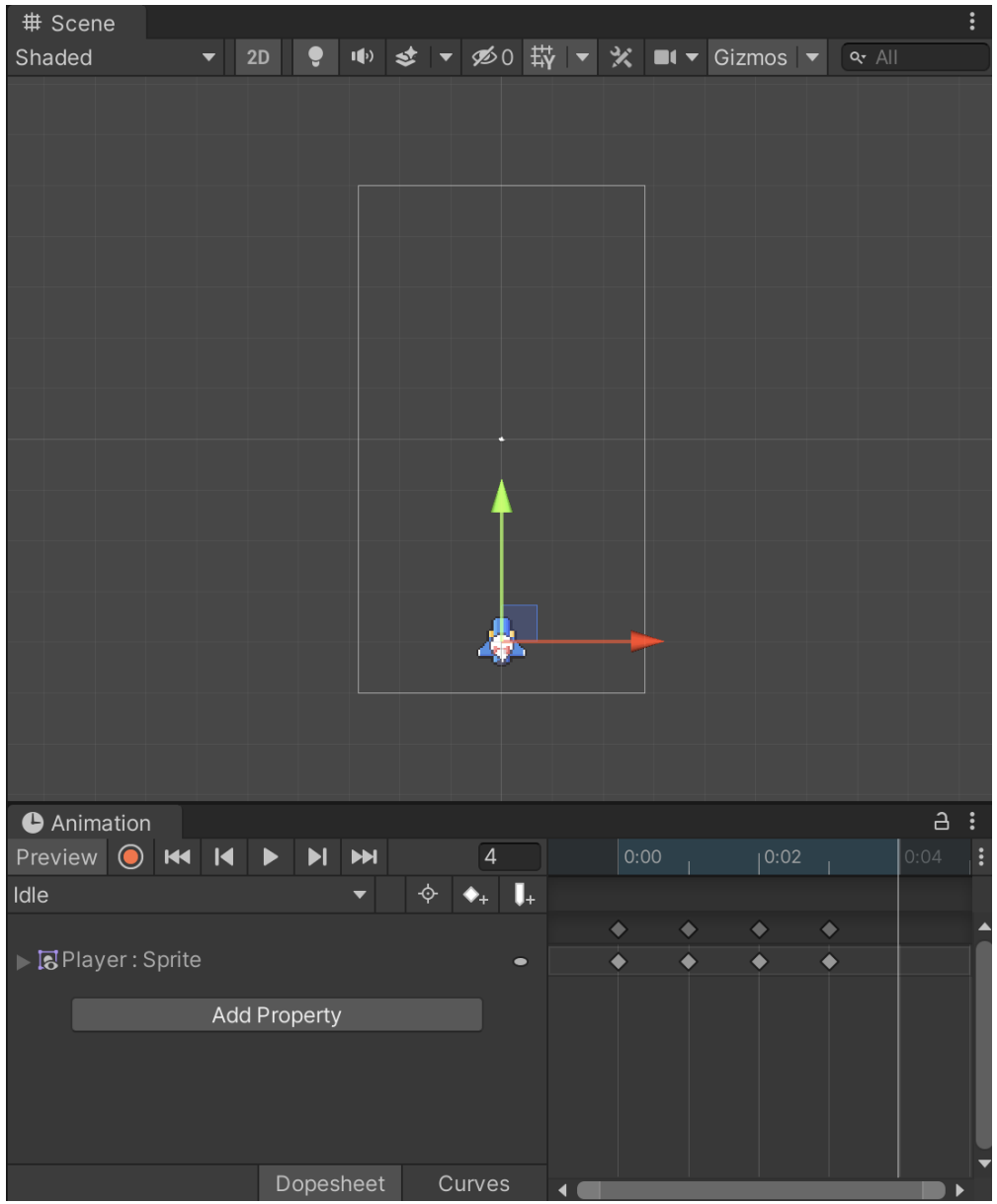
Idle 애니메이션 클립이 생성되었습니다

Player 애니메이션 컨트롤러가 생성되었습니다

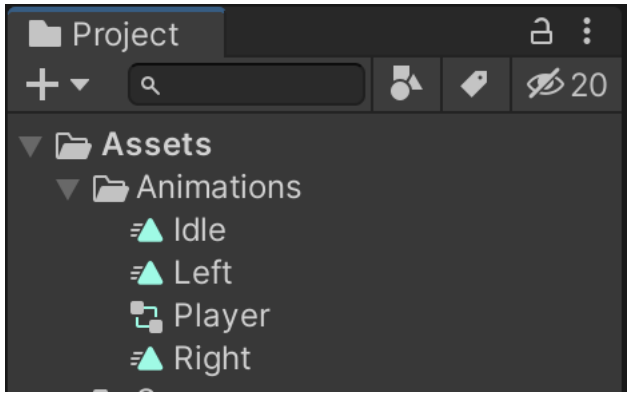




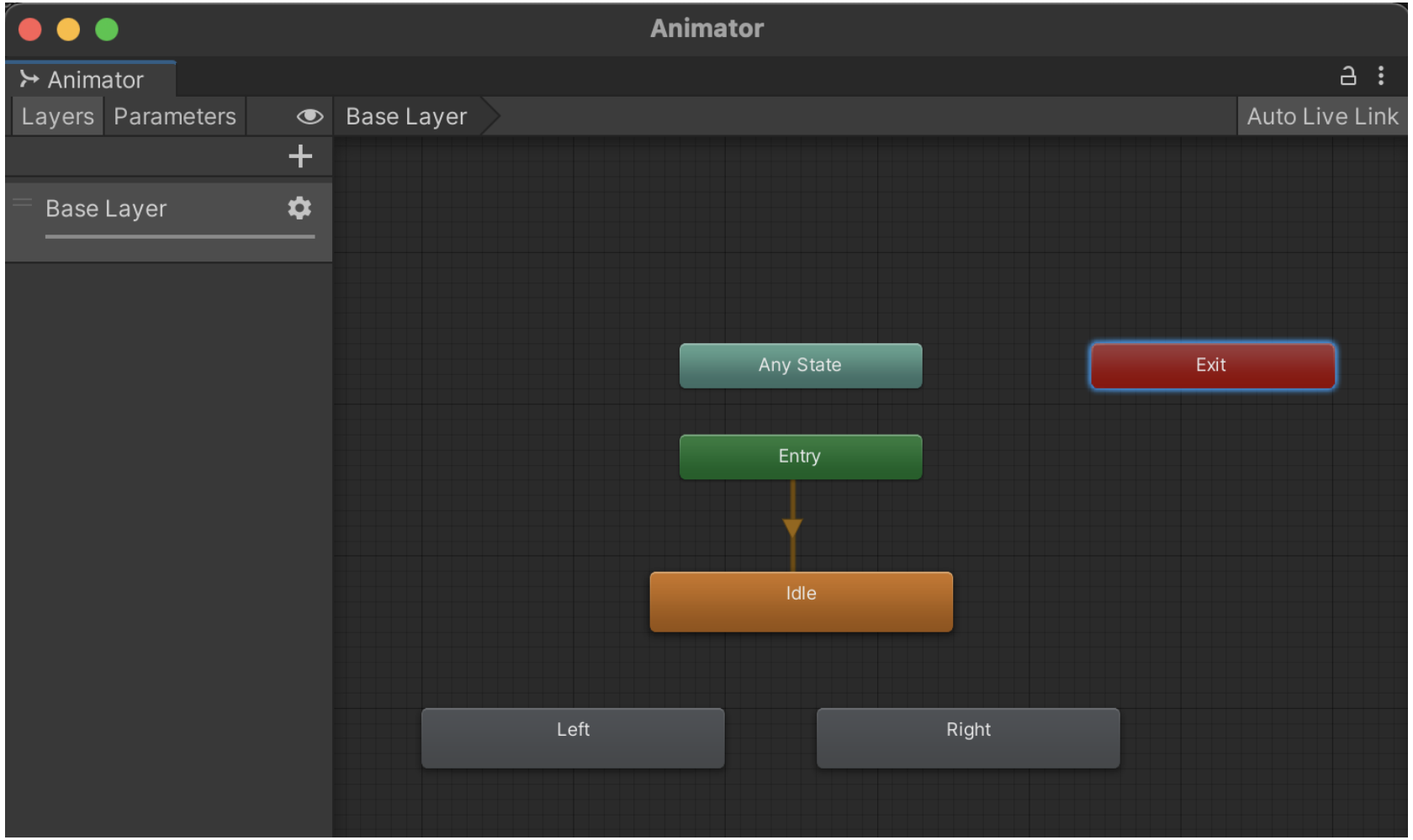
Idle애니메이션을 확인 합니다



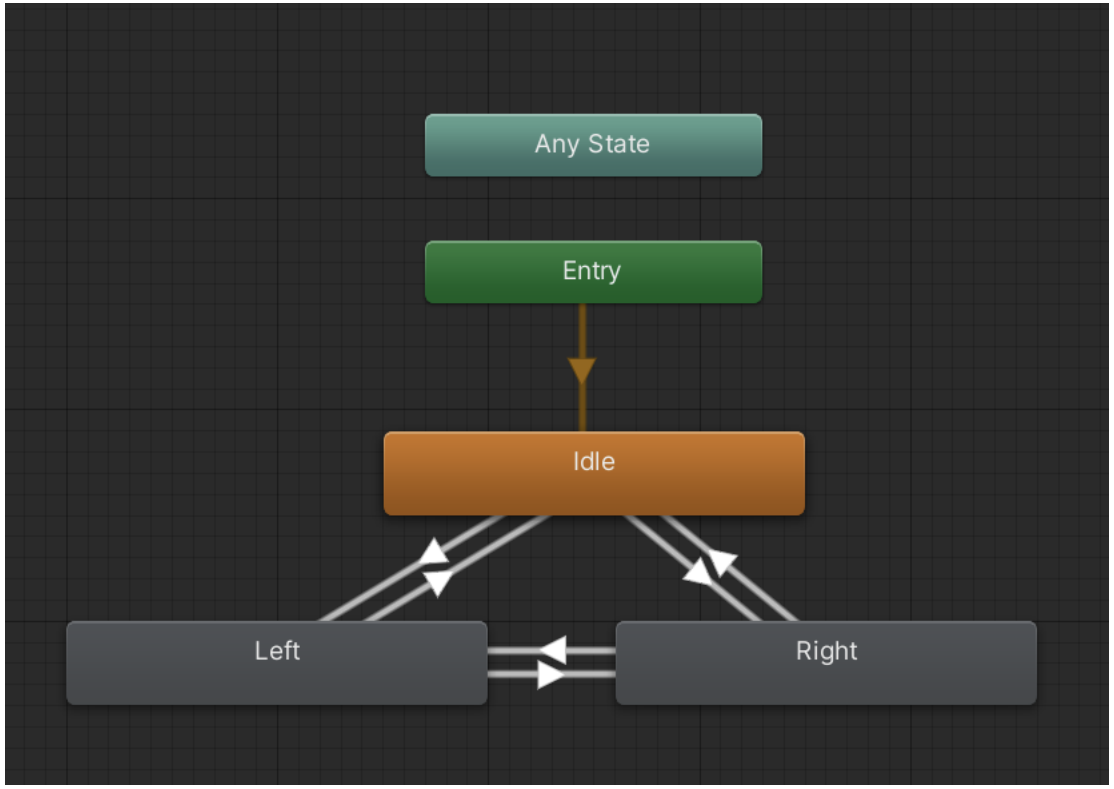
동일한 방법으로 Left, Right도 만들어 줍니다



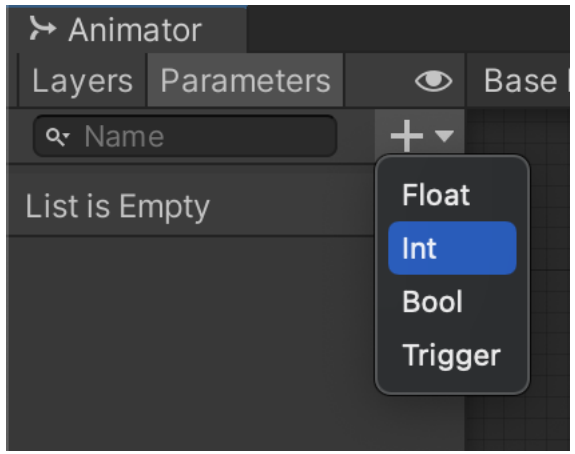
Player애니메이션 컨트롤러를 열어 줍니다

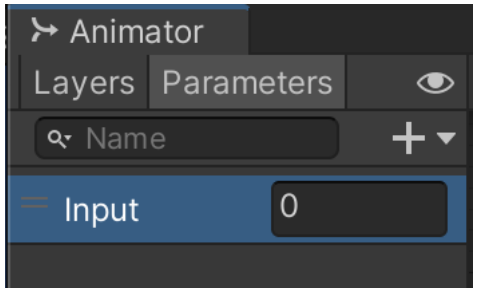


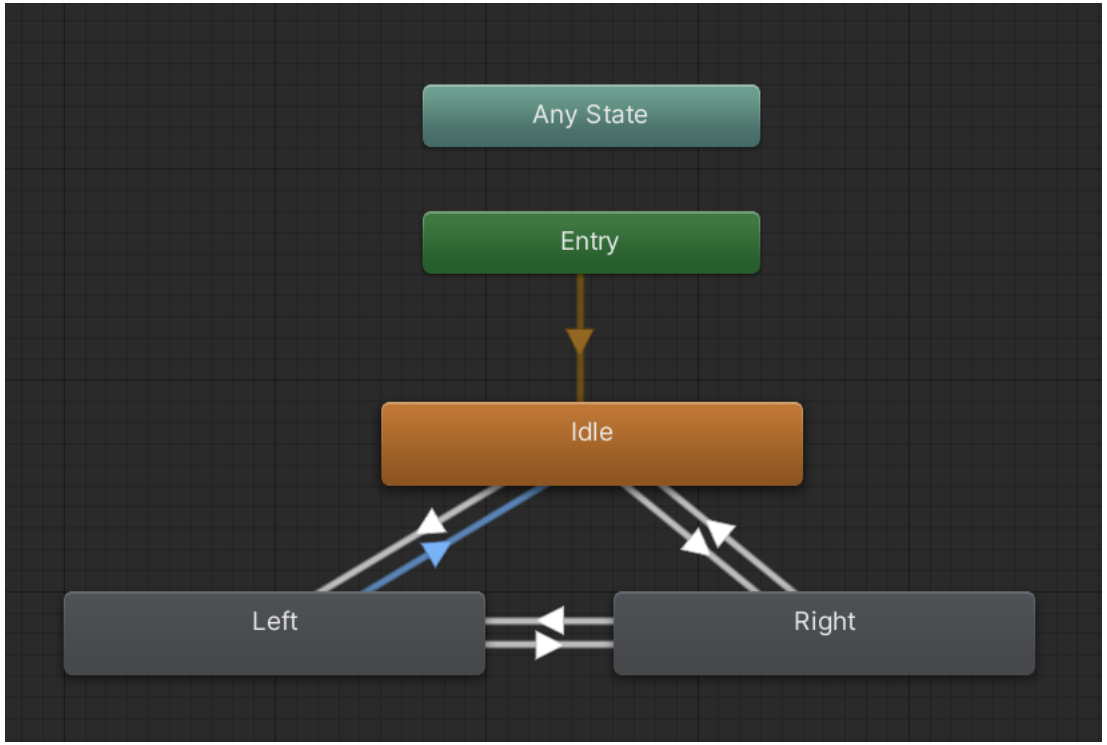
다음과 같이 Transition을 연결합니다

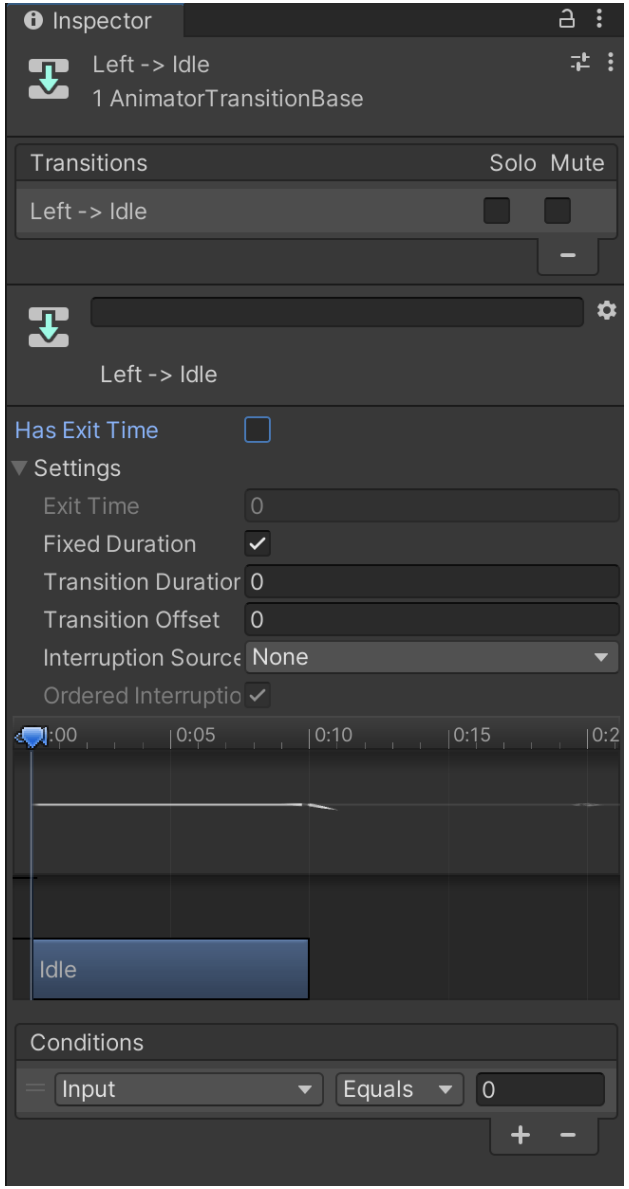


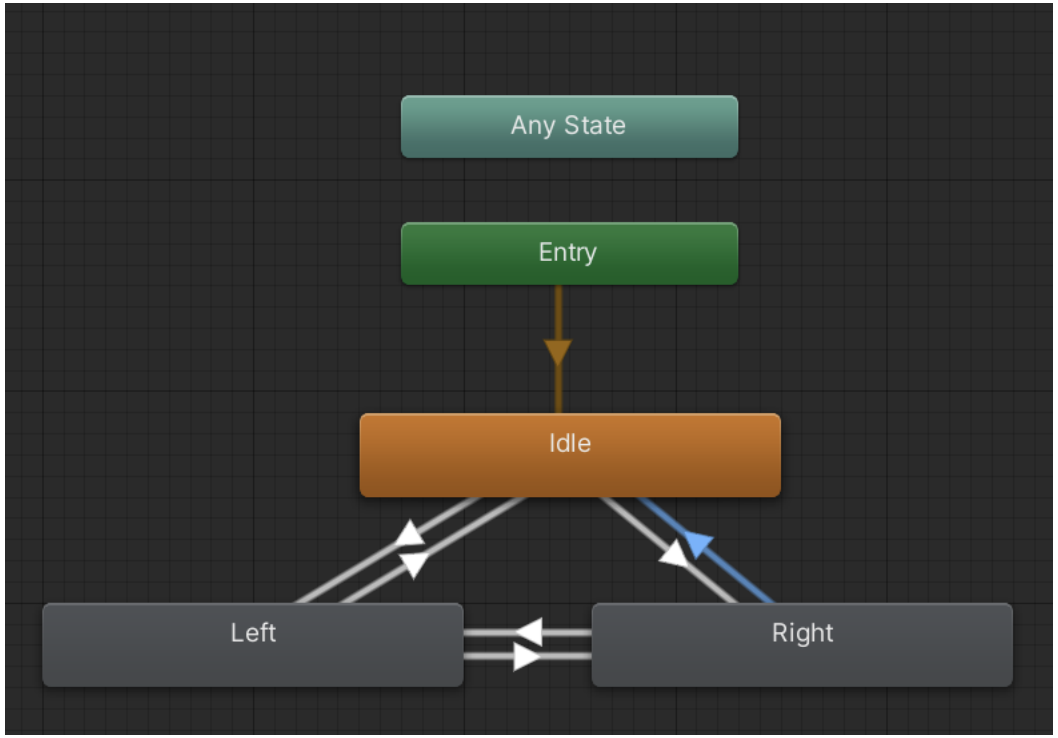
매개변수는 Input.GetAxisRaw값을 사용합니다

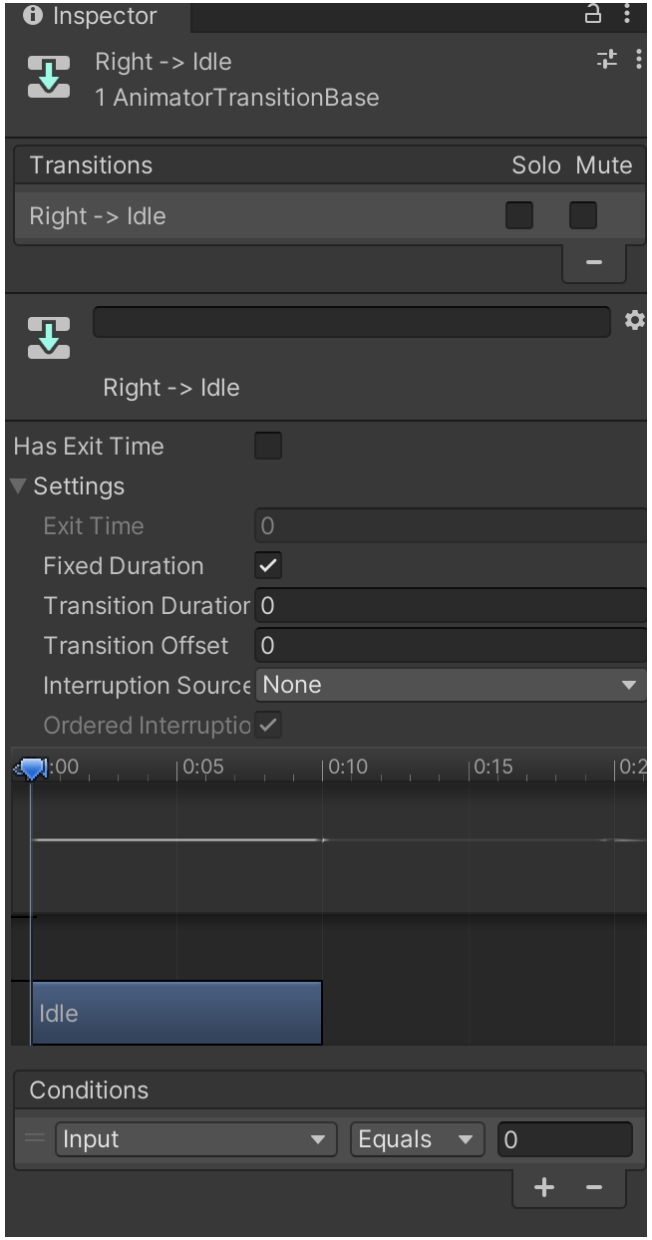


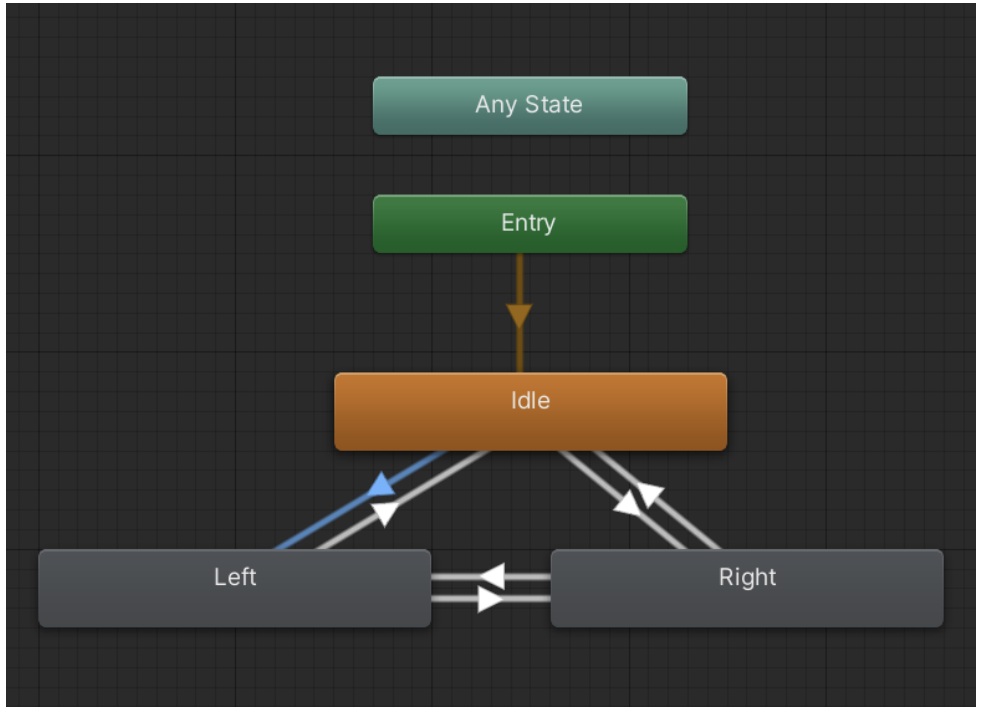


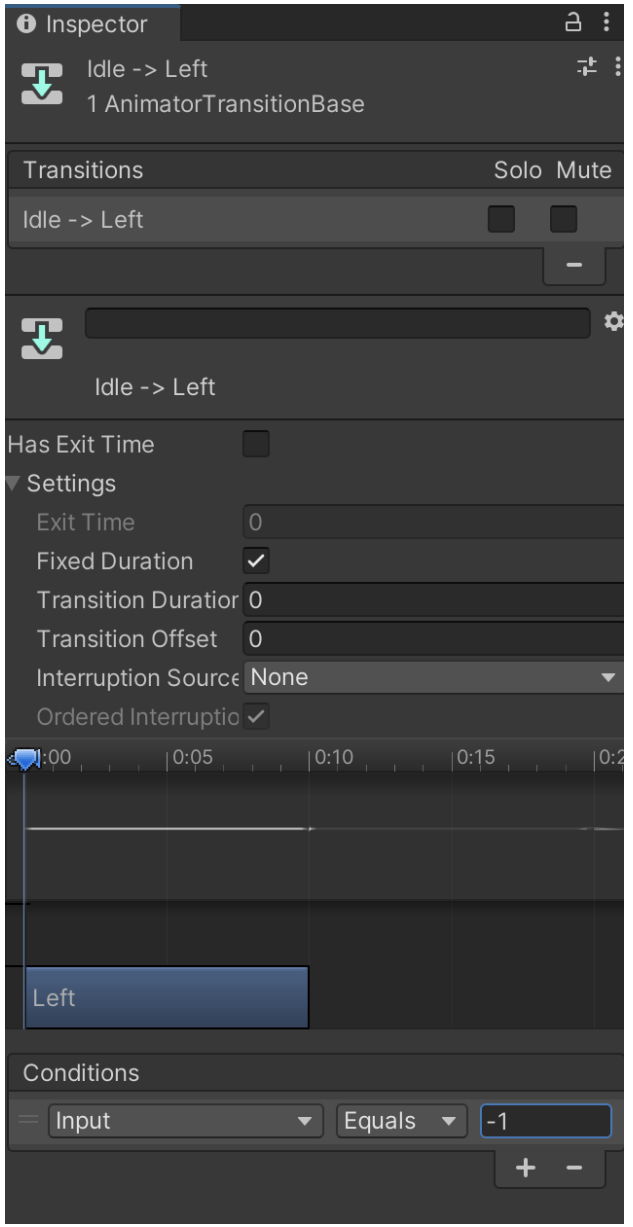

Any State
Entry
Idle
Left
Right

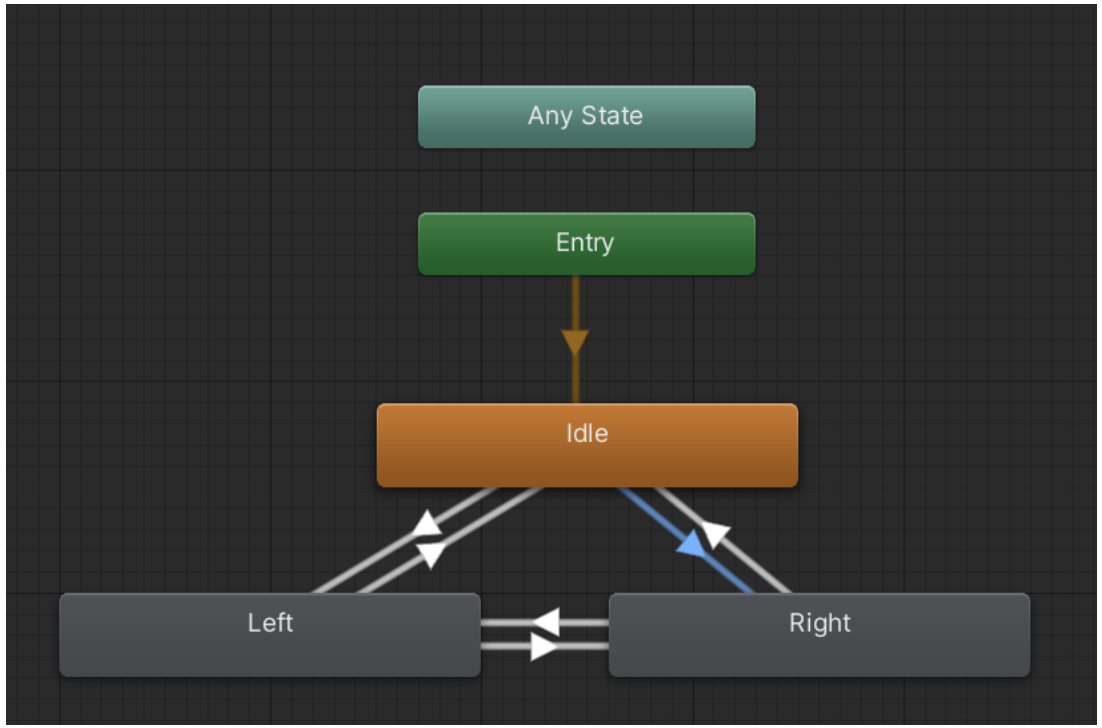

Inspector
Left -> Idle
1 AnimatorTransitionBase
Transitions
Solo
Mute
Left -> Idle
Left -> Idle
Has Exit Time
Settings
Exit Time
0
Fixed Duration
Transition Duratior
0
Transition Offset
0
Interruption Source
None
Ordered Interruptio
:00
0:05
0:10
0:15
0:2
Idle
Conditions
Input
Equals
0

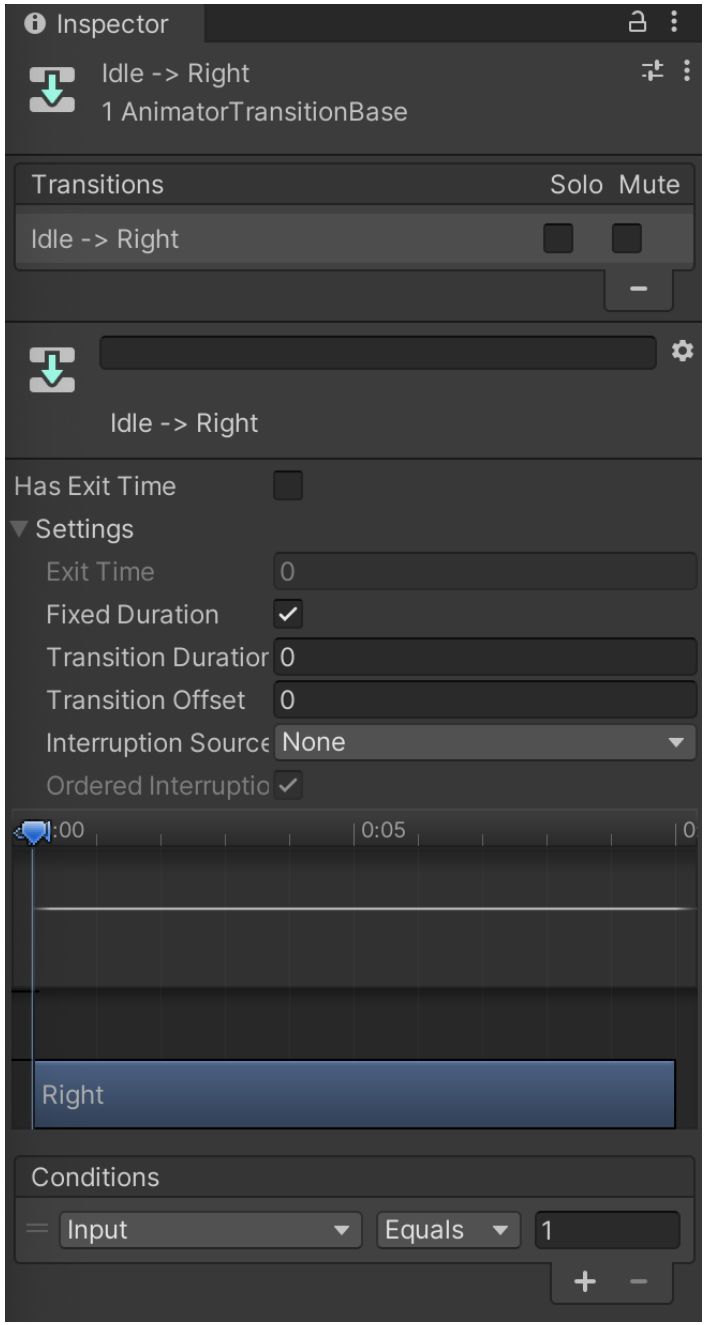

Any State
Entry
Idle
Left
Right

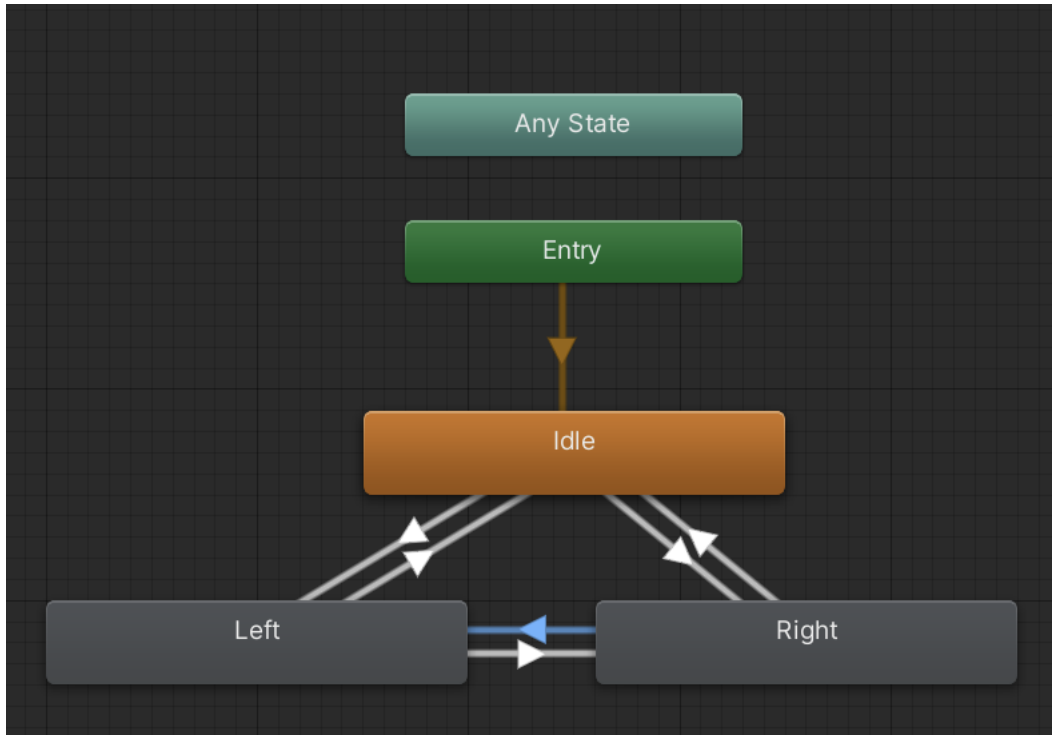

Inspector
Right -> Idle
1 AnimatorTransitionBase
Transitions
Solo
Mute
Right -> Idle
Right -> Idle
Has Exit Time
Settings
Exit Time
0
Fixed Duration
Transition Duration
0
Transition Offset
0
Interruption Source
None
Ordered Interruptio
:00
0:05
0:10
0:15
0:2
Idle
Conditions
Input
Equals
0

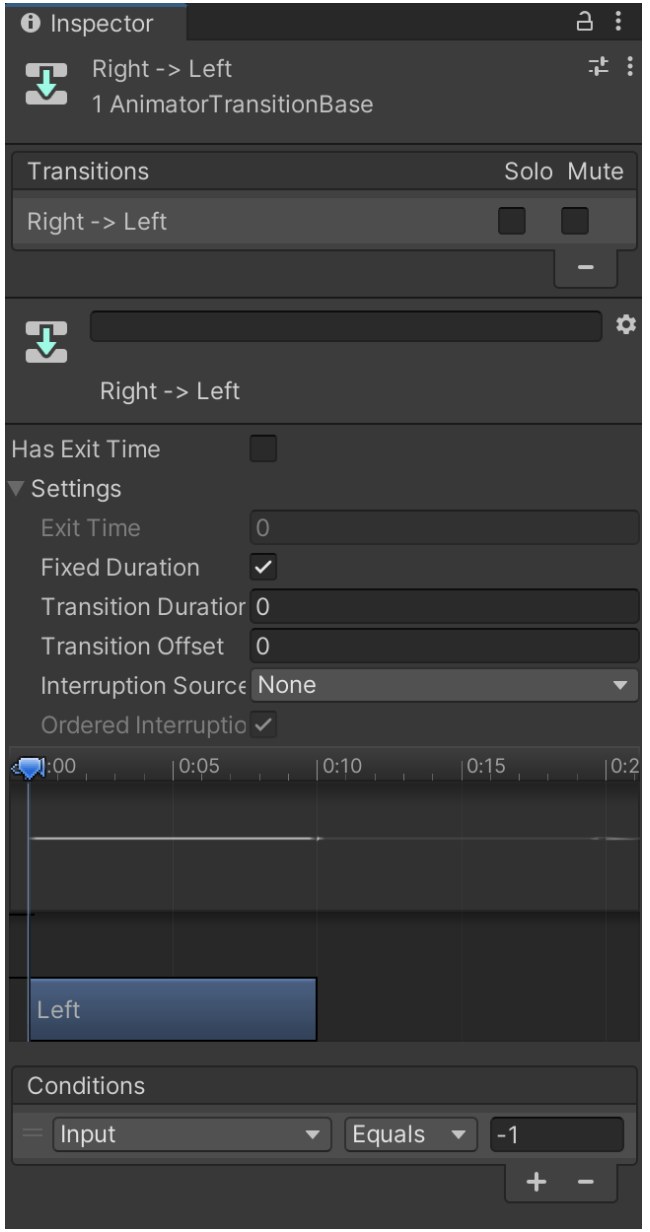

Any State
Entry
Idle
Left
Right

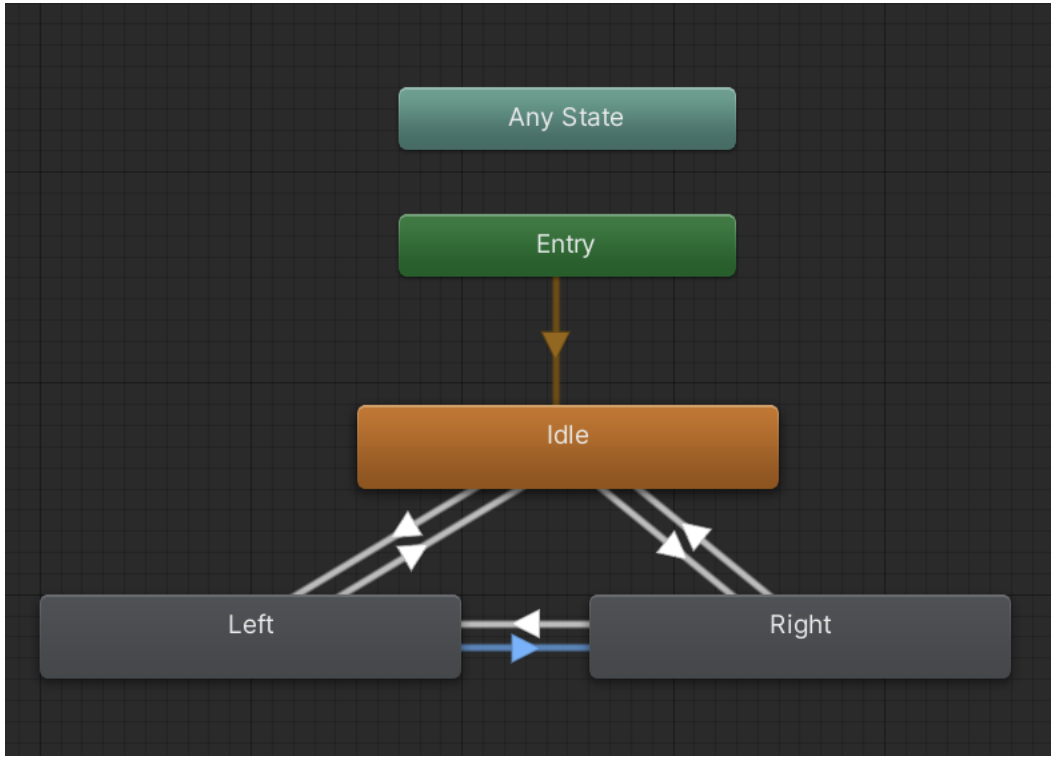

Inspector
Idle -> Left
1 AnimatorTransitionBase
Transitions
Solo
Mute
Idle -> Left
Idle -> Left
Has Exit Time
Settings
Exit Time
0
Fixed Duration
Transition Duration
0
Transition Offset
0
Interruption Source
None
Ordered Interruptio
:00
0:05
0:10
0:15
Left
Conditions
Input
Equals
-1

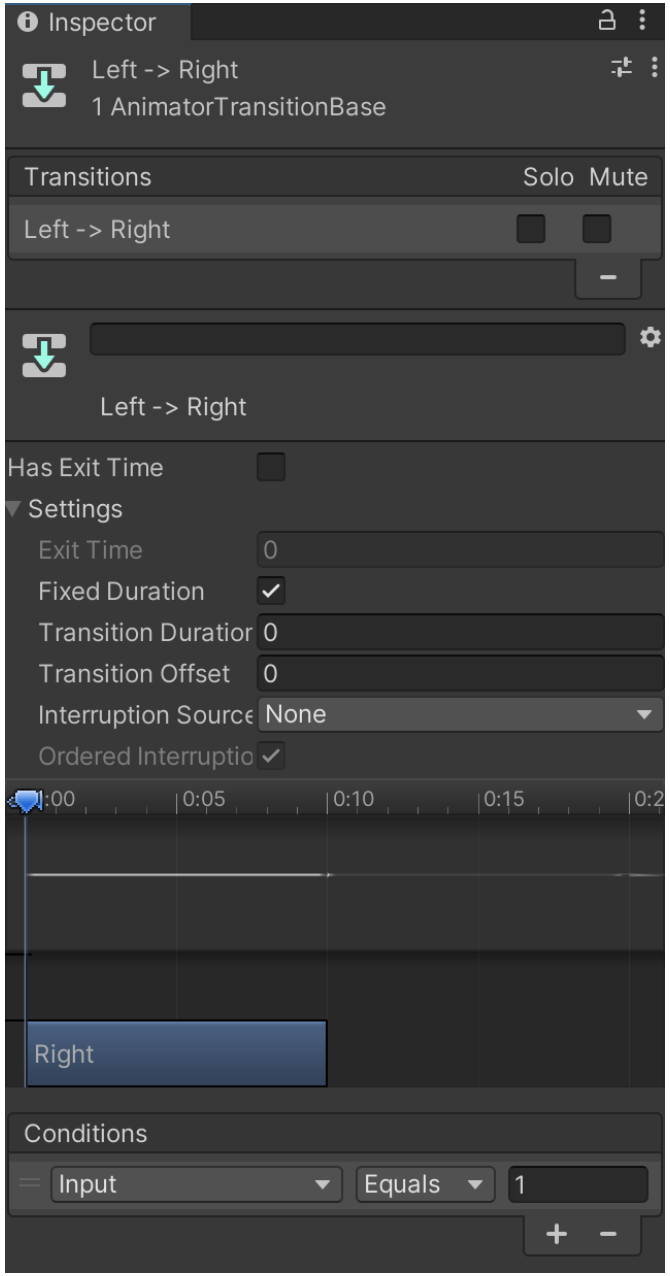

Any State
Entry
Idle
Left
Right


Inspector
Idle -> Right
1 AnimatorTransitionBase
Transitions
Solo
Mute
Idle -> Right
Idle -> Right
Has Exit Time
Settings
Exit Time
0
Fixed Duration
Transition Duration
0
Transition Offset
0
Interruption Source
None
Ordered Interruptio
:00
0:05
0
Right
Conditions
Input
Equals
1


Any State
Entry
Idle
Left
Right


Inspector
Right -> Left
1 AnimatorTransitionBase
Transitions
Solo
Mute
Right -> Left
Right -> Left
Has Exit Time
Settings
Exit Time
0
Fixed Duration
Transition Duration
0
Transition Offset
0
Interruption Source
None
Ordered Interruptio
:00
0:05
0:10
0:15
0:2
Left
Conditions
Input
Equals
-1

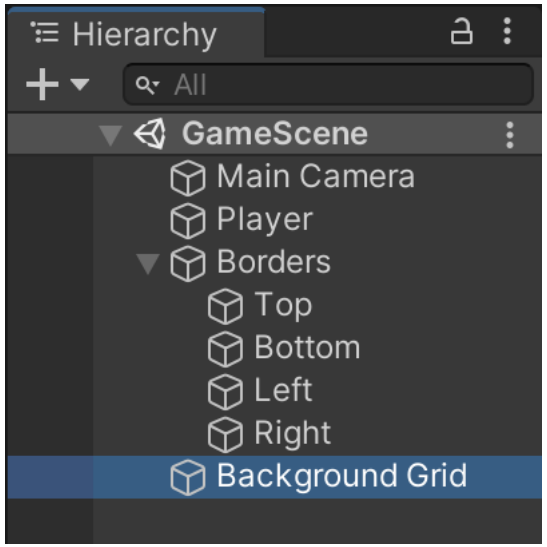

Any State
Entry
Idle
Left
Right

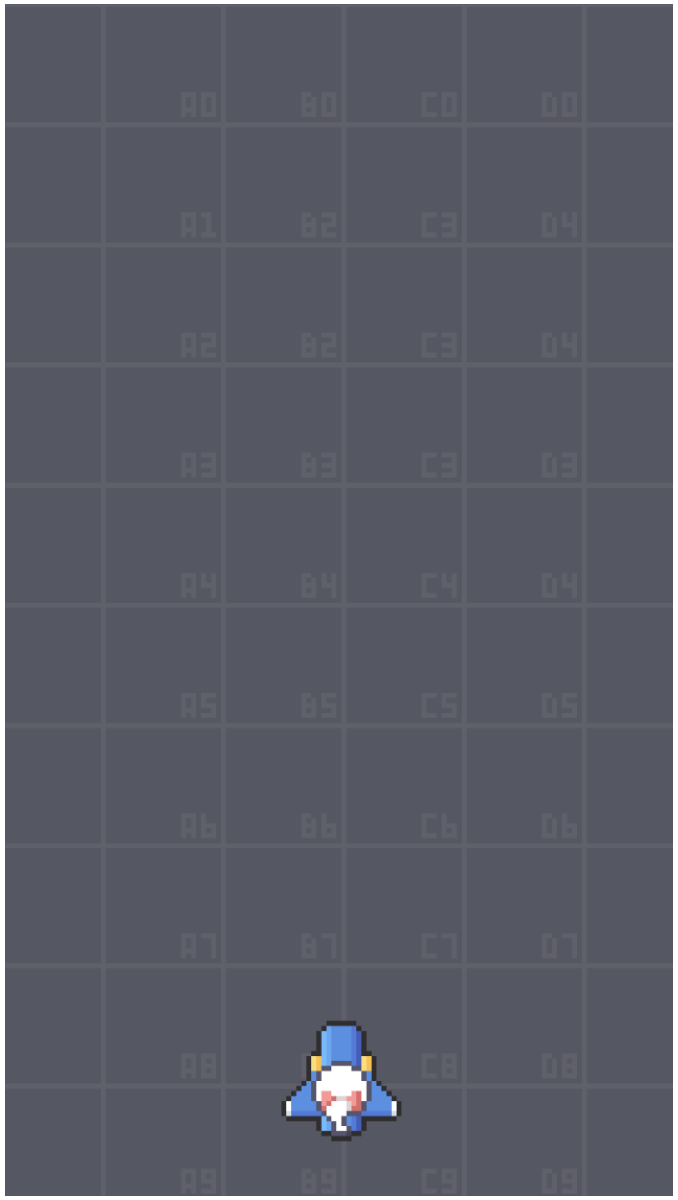

Inspector
Left -> Right
1 AnimatorTransitionBase
Transitions
Solo
Mute
Left -> Right
Left -> Right
Has Exit Time
Settings
Exit Time
0
Fixed Duration
Transition Duratior
0
Transition Offset
0
Interruption Source
None
Ordered Interruptio
:00
0:05
0:10
0:15
0:2
Right
Conditions
Input
Equals
1

```
public class PlayerController : MonoBehaviour
{
    public float speed = 1;
    private bool isTouchTop;
    private bool isTouchBottom;
    private bool isTouchLeft;
    private bool isTouchRight;

    Animator anim;

    private void Start()
    {
        this.anim = this.GetComponent<Animator>();
    }
```

```
void Update()
{
    float h = Input.GetAxisRaw("Horizontal");

    if ((this.isTouchRight && h == 1) || (this.isTouchLeft && h == -1))
    {
        h = 0;
    }

    float v = Input.GetAxisRaw("Vertical");

    if ((this.isTouchTop && v == 1) || (this.isTouchBottom && v == -1))
    {
        v = 0;
    }

    Vector3 currentPos = this.transform.position;
    Vector3 nextPos = new Vector3(h, v, 0).normalized * this.speed * Time.deltaTime;
    this.transform.position = currentPos + nextPos;

    if (Input.GetButtonDown("Horizontal") || Input.GetButtonUp("Horizontal"))
    {
        this.anim.SetInteger("Input", (int)h);
    }
}
```

배경 텍스쳐를 씬에 가져다 놓고 실행후 결과를 확인 합니다





https://www.youtube.com/watch?v=ETYzjbnLixY&list=PLO-mt5Iu5TeYtWvM9eN-xnwRbyUAMWd3b

https://github.com/smilejsu82/learn-shooting-game

https://assetstore.unity.com/?q=%EC%A2%85%EC%8A%A4%ED%81%AC%EB%A1%A4%202D%20%EC%8A%88%ED%8C%85&orderBy=1