

Unity3D Lecture Notes (Part2)

prefab로 만들어서 재활용 가능

toggle하기 쉬워

framework -- partially implemented application

white box framework → Unity는 이렇게 만들

black box framework

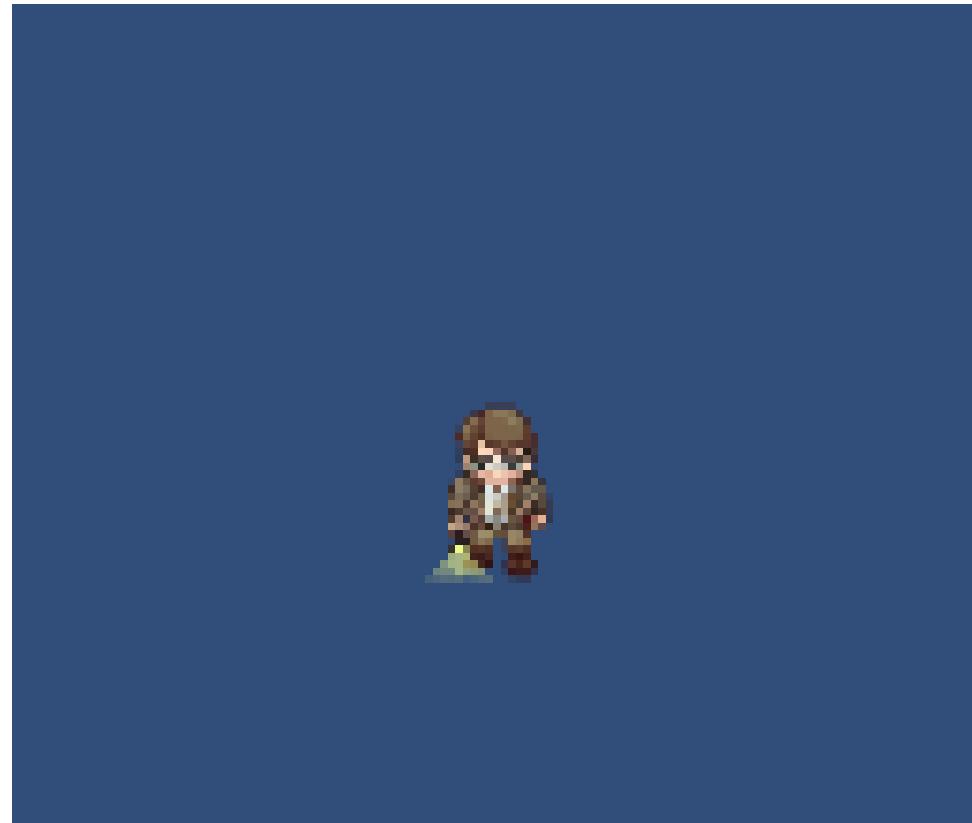
Unity engine → framework

2D Animation

- Animated sprite
 - **Animator Component**를 가짐
 - Animator Component에서는,
 - Controller 속성은 **Animator Controller** 에셋이 설정되어야 함.
 - **Animator Controller** = a finite state machine, where each state is an animation (or **animation clip**)

Project 9: Understand a 2D Animation Example: Project

- See “Lab10_Egg_Shooting_borrillis-blog-projects Project”



File Edit Assets GameObject Component Window Help



Pivot Global



Account

Layers

Layc

Hierarchy

Create All
Main*
Main Camera
professor
another professor

Scene

Game

Animator

Layers Parameters

Base Layer

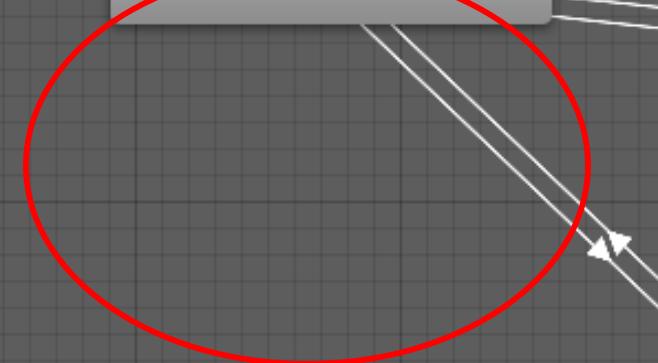
Auto Live Link

Direction

0

professor_walk_west

Animated sprite
에는 animator
controller가
controller 속성
에 설정되어야
함.



professorAnimatorController.controller

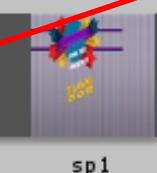
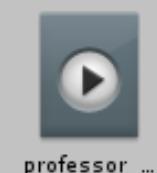
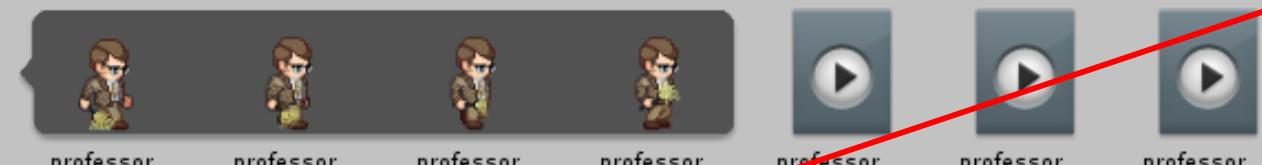
Project Console

Create

Favorites
All Materials
All Models
All Prefabs
All Scripts

Assets

Scenes



Inspector

professor
Tag Untagged Layer Default

Transform

Position X 1.9063 Y -1.9961 Z 0
Rotation X 0 Y 0 Z 0
Scale X 5 Y 5 Z 5

Sprite Renderer

Sprite professor_walk_west
Color
Flip
Material Sprites-Default
Sorting Layer Default
Order in Layer 0

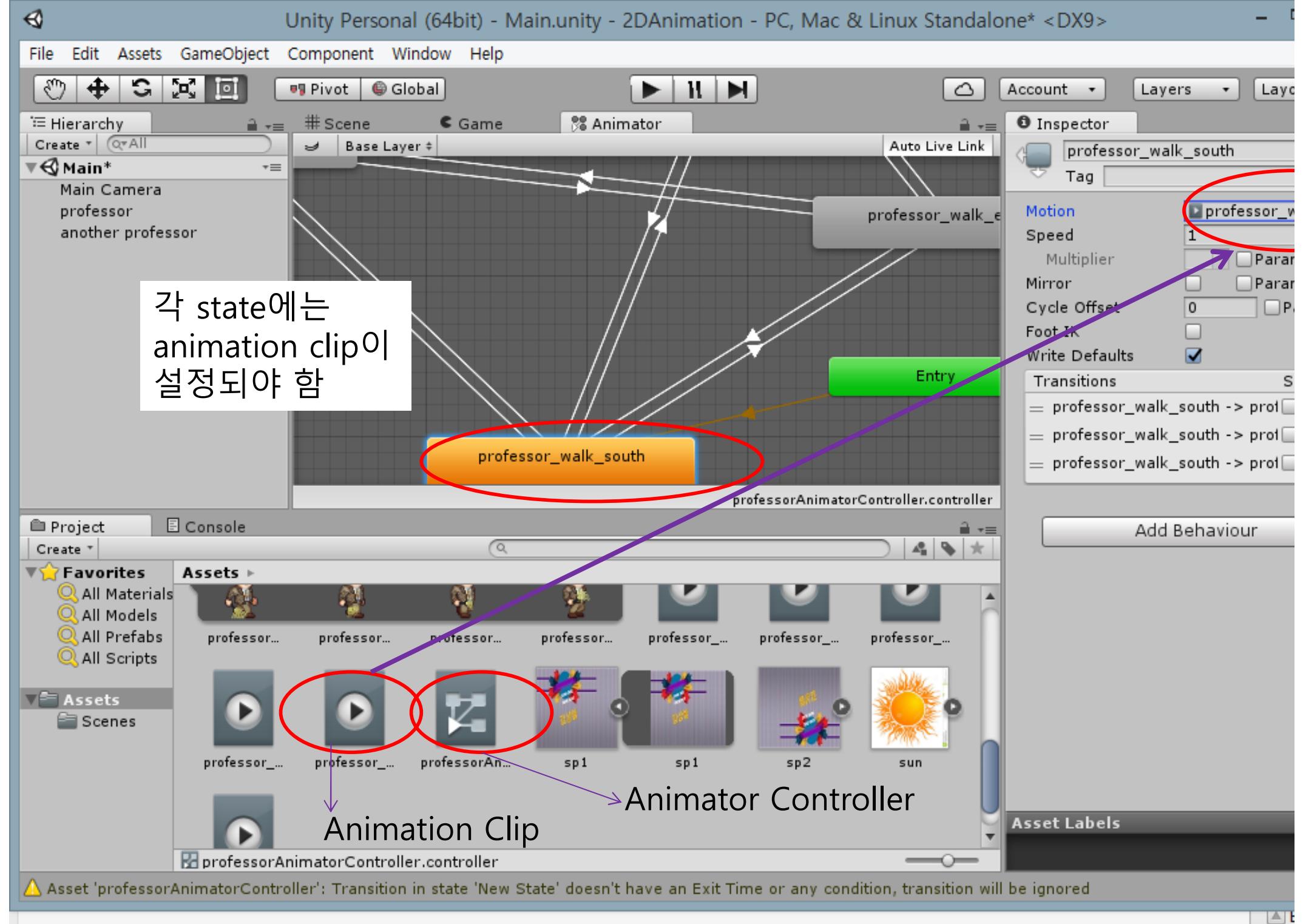
Animator

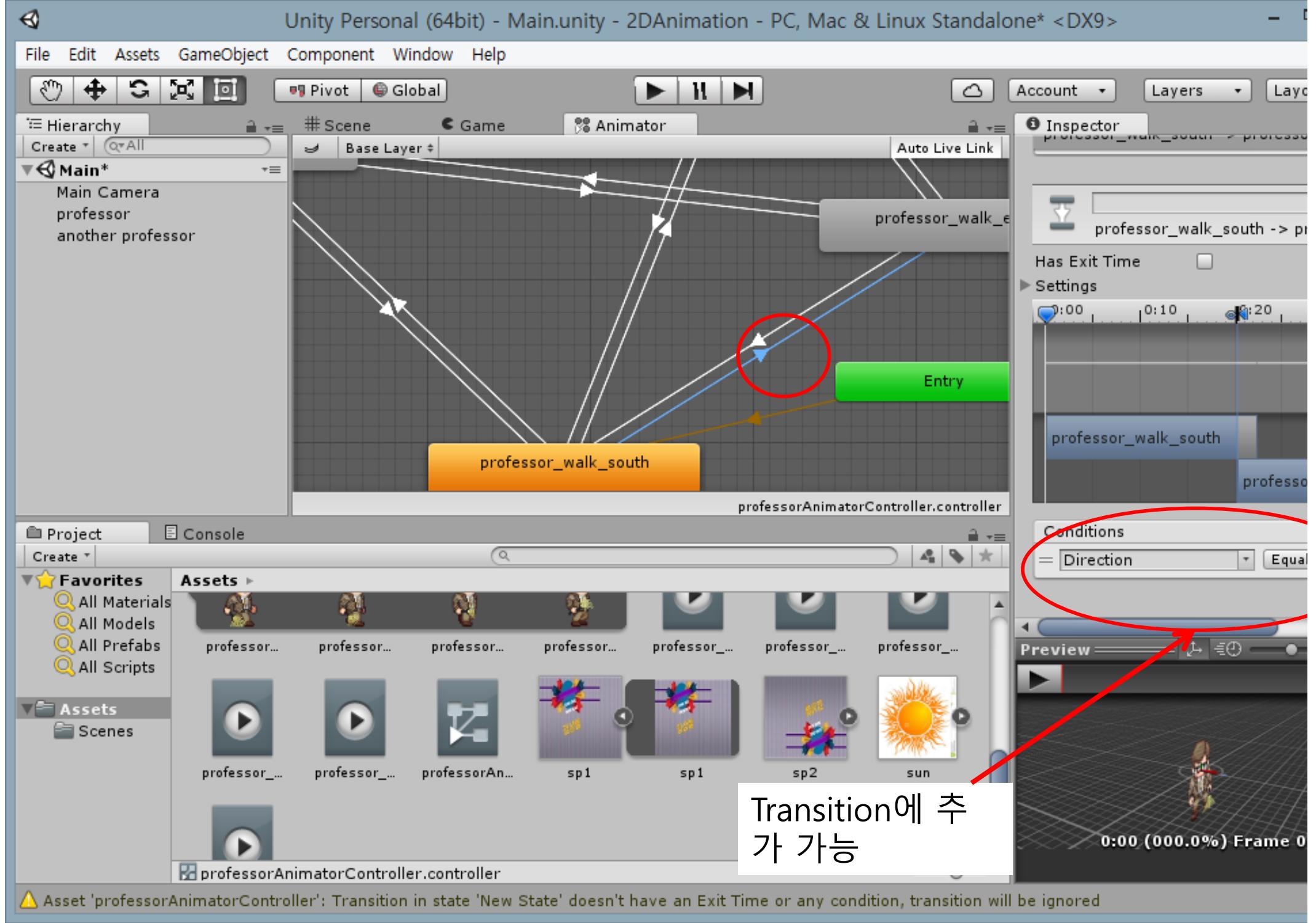
Controller professorAnimatorController
Avatar None (Avatar)
Apply Root Motion
Update Mode Normal
Culling Mode Always Animating

Clip Count: 4
Curves Pos: 0 Quat: 0 Euler: 0 S
Muscles: 0 Generic: 0 PPtr: 4
Curves Count: 4 Constant: 0 (0.0)
Dense: 0 (0.0%) Stream: 4 (100.0%)

Player Controller (Script)

playerController
Sprites-Default
Shader Sprites/Default





Transition에 추가 가능



Pivot Global



Account

Layers

Layc

Hierarchy

Create All

Main*

Main Camera

professor

another professor

Scene

Game

Animator

Auto Live Link

Layers

Parameters

Name

+

-

Direction

0

Base Layer

List

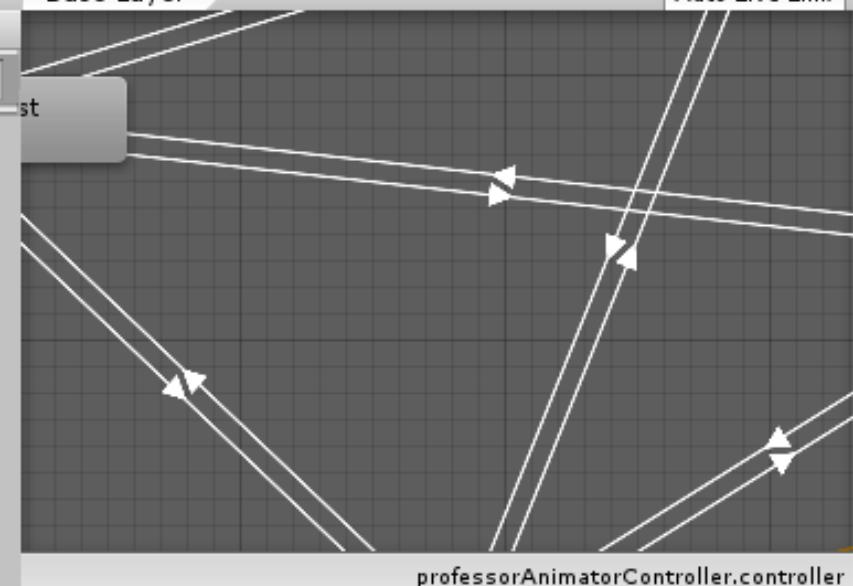
Inspector

Base Layer

-

Add Behaviour

Parameter를
추가 삭제 할 수
있음



Project

Console

Create

Favorites

- All Materials
- All Models
- All Prefabs
- All Scripts

Assets

Scenes

Assets



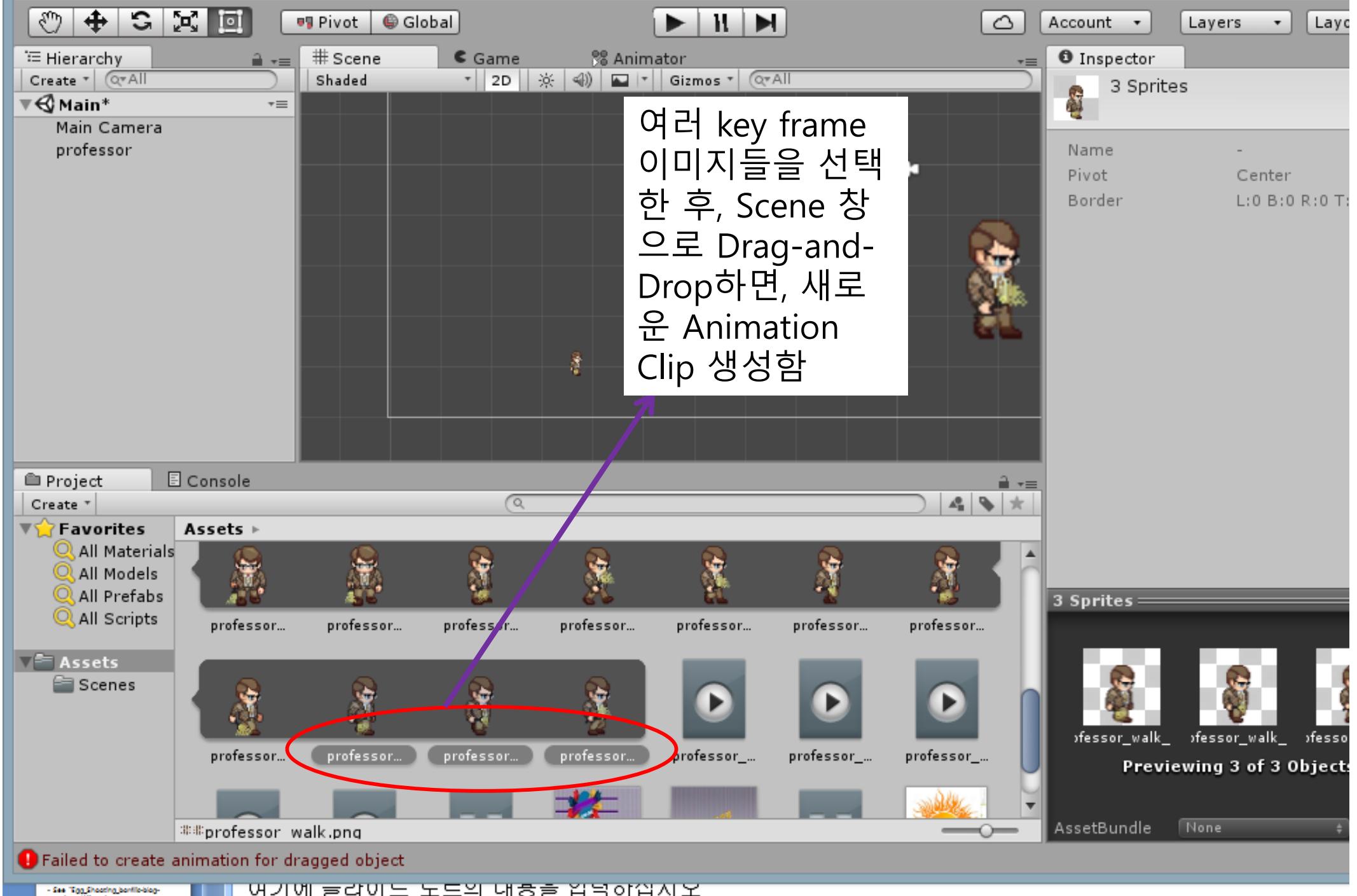
professorAnimatorController.controller

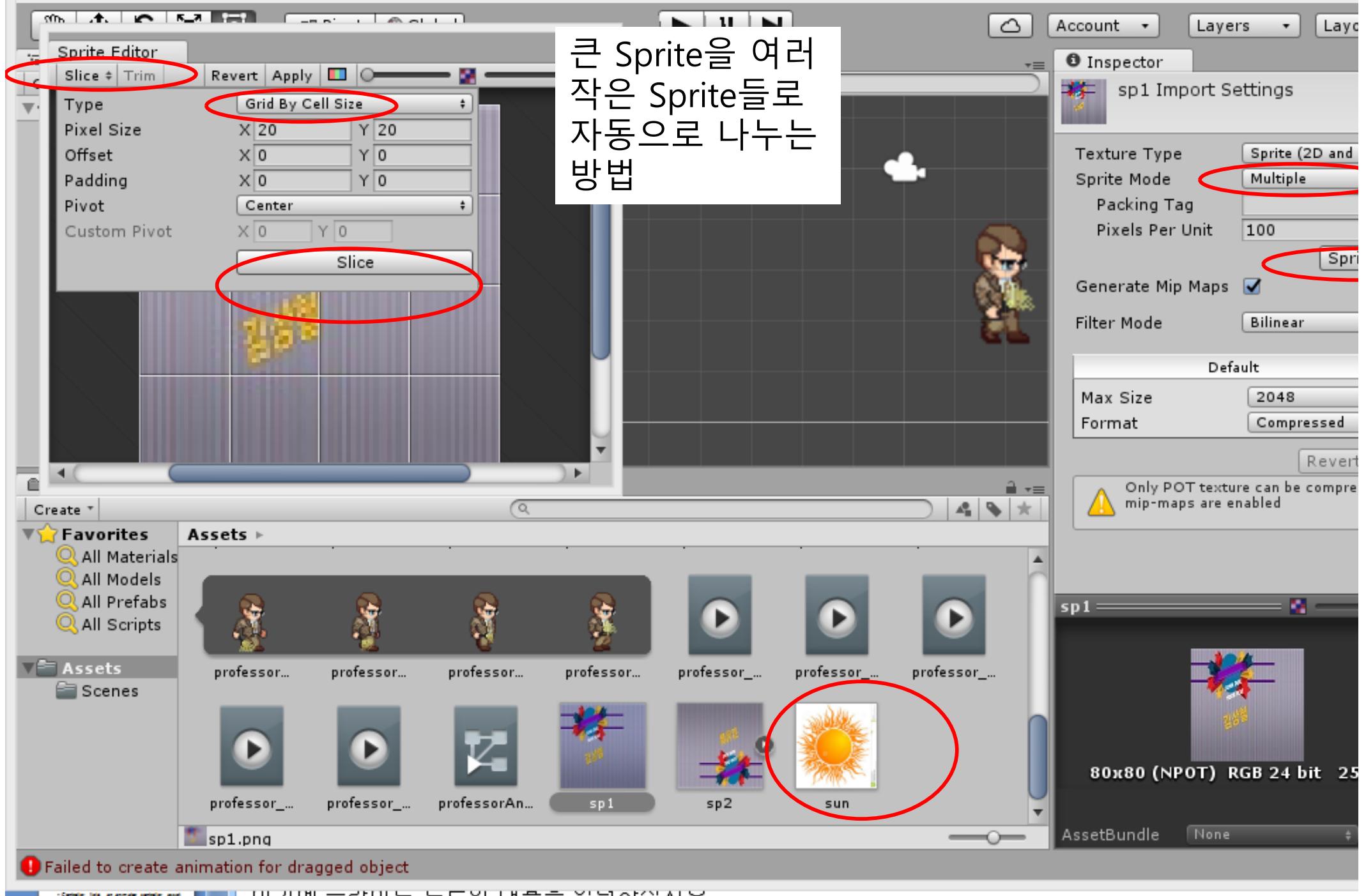
Asset Labels

Asset 'professorAnimatorController': Transition in state 'New State' doesn't have an Exit Time or any condition, transition will be ignored

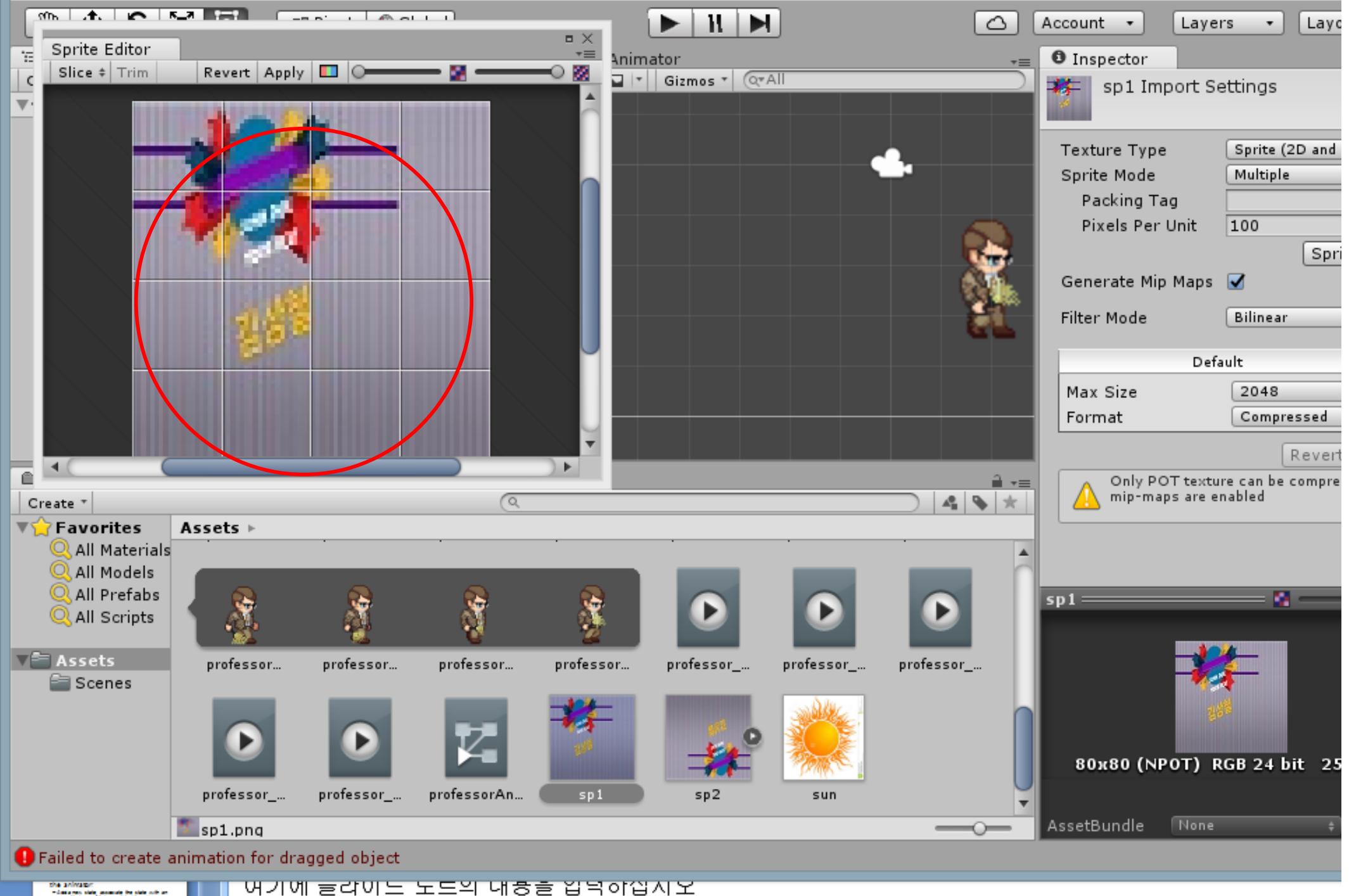
- Import Images
- The imported images will change their size.
- Drop more than one image into the same folder.

여기에서 틀라미는 노드의 내용을 입력해주세요





큰 Sprite을 여러
작은 Sprite들로
자동으로 나누는
방법



playerController.cs (professor 객체 에 부착된 script)

```
using UnityEngine;  
using System.Collections;  
  
public class playerController : MonoBehaviour  
{  
  
    private Animator animator;  
  
    // Use this for initialization  
    void Start()  
    {  
        animator = this.GetComponent<Animator>();  
    }  
}
```

```
// Update is called once per frame
void Update()
{
    var vertical = Input.GetAxis("Vertical");
    var horizontal = Input.GetAxis("Horizontal");

    if (vertical > 0)
    {
        animator.SetInteger("Direction", 2);
    }
    else if (vertical < 0)
    {
        animator.SetInteger("Direction", 0);
    }
    else if (horizontal > 0)
    {
        animator.SetInteger("Direction", 1);
    }
    else if (horizontal < 0)
    {
        animator.SetInteger("Direction", 3);
    }
}
```

Lab 10: 2D Animation Sprite Exercise

- Create a 2D project
- Import images
 - They will become sprites.
- Select more than one sprite and drag them into the scene view.
 - game object, animator controller, anim file will be created
- Create a Sprite game object
- Add an Animator component to the Sprite game object.

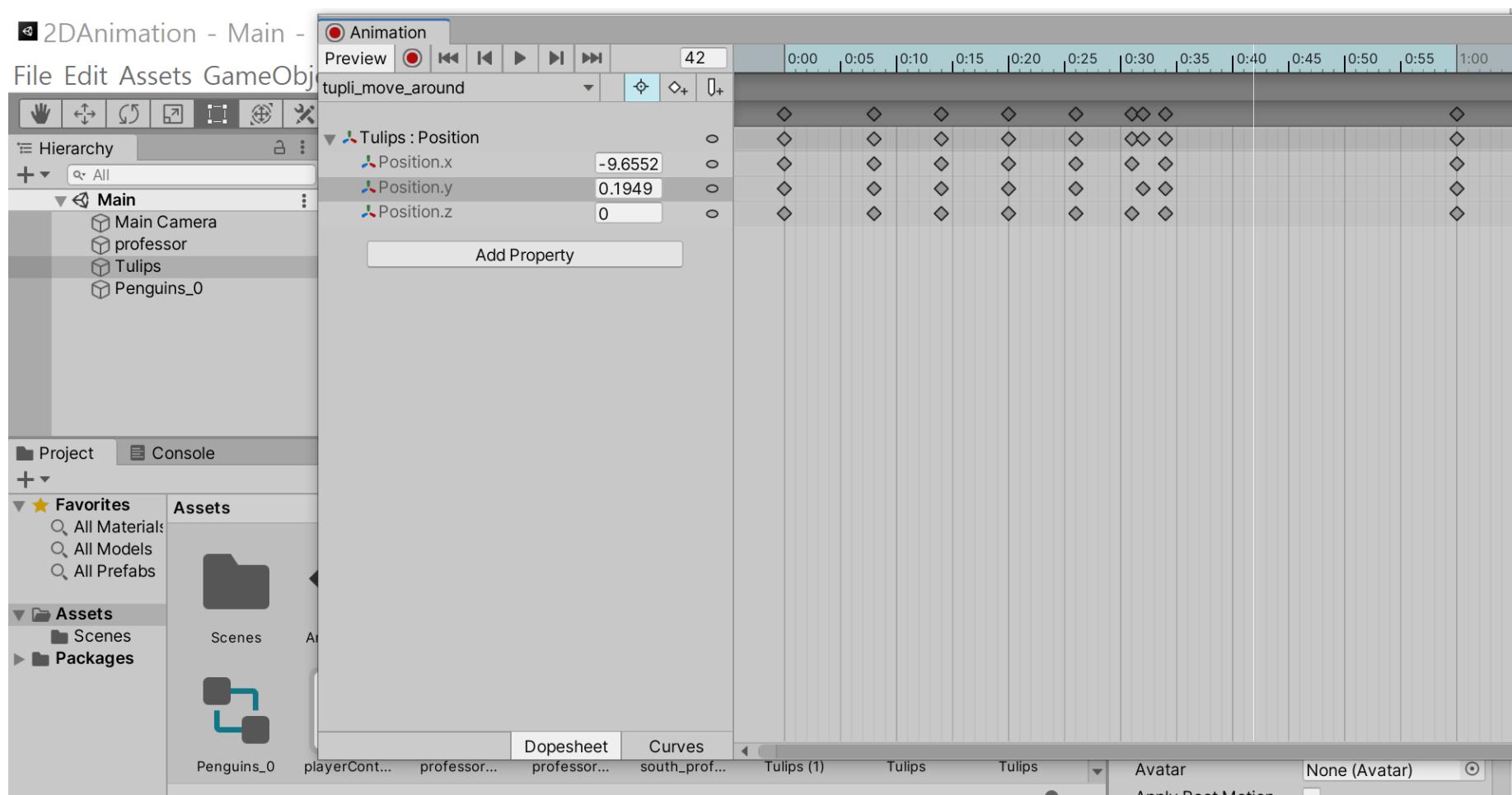
- Edit the animator controller and assign it to the animator:
 - Add a new state, associate the state with an anim file (in Motion attribute), add a transition between two states, create a proper parameter, add a condition (using the parameter) to the constraint, etc
- Add a C# script to the game object, where the parameter is set to some value on a certain condition.

Example C#

```
private Animator animator;  
void Start()  {  
    animator = this.GetComponent<Animator>();  
}  
void Update()    {  
    var vertical = Input.GetAxis("Vertical");  
    var horizontal = Input.GetAxis("Horizontal");  
    if (vertical > 0)  
    {  
        animator.SetInteger("Direction", 2); //Direction is  
a parameter  
        ...  
    }  
..
```

Animation Clip 제작

- <https://learn.unity.com/tutorial/working-with-animations-and-animation-curves#600c8a28edbc2a3270428113>



2DAnimation - Main -

File Edit Assets GameObject

Hierarchy

- + Main
 - Main Camera
 - professor
 - Tulips
 - Penguins_0

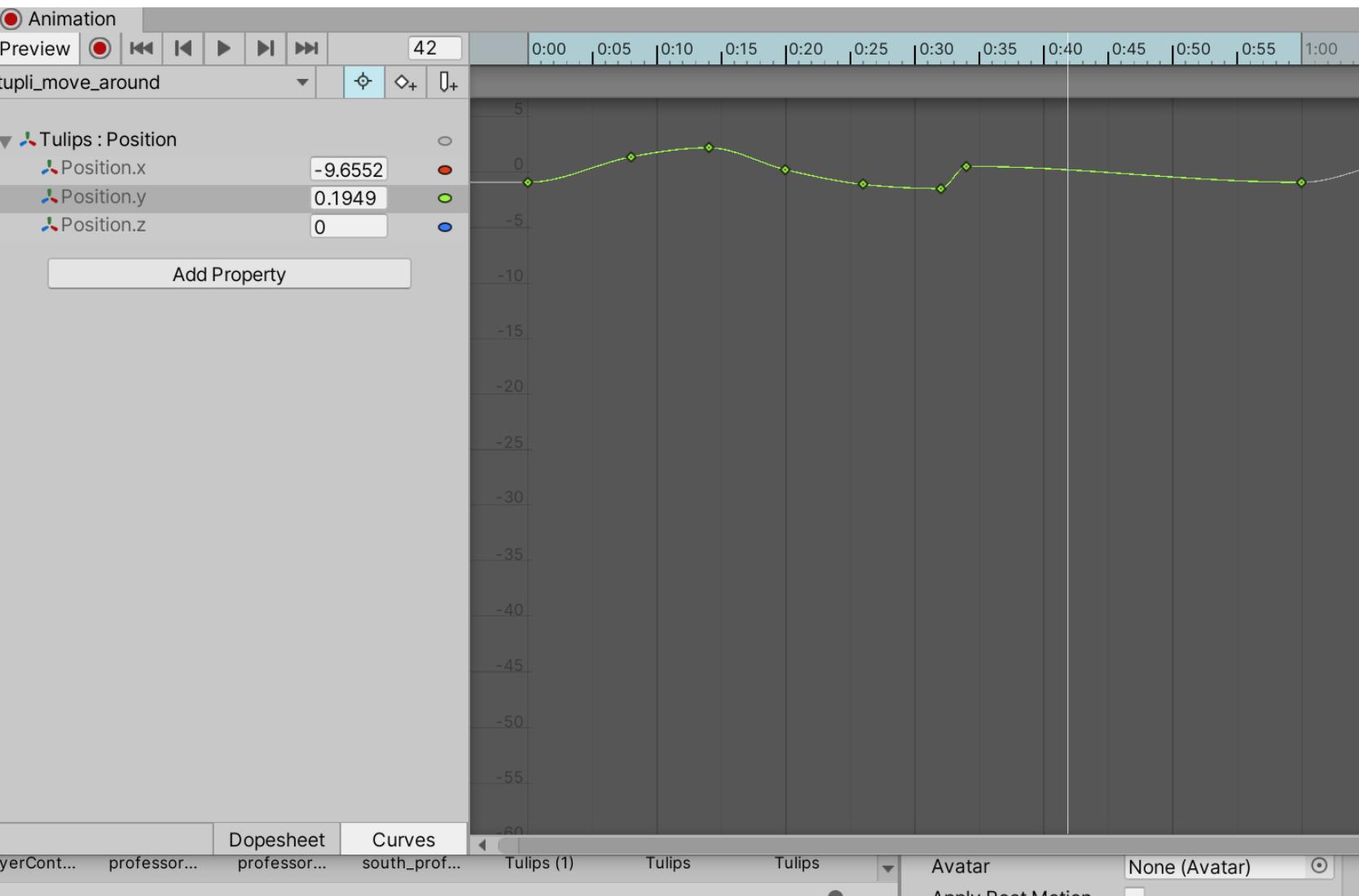
Project

Favorites

- All Materials
- All Models
- All Prefabs

Assets

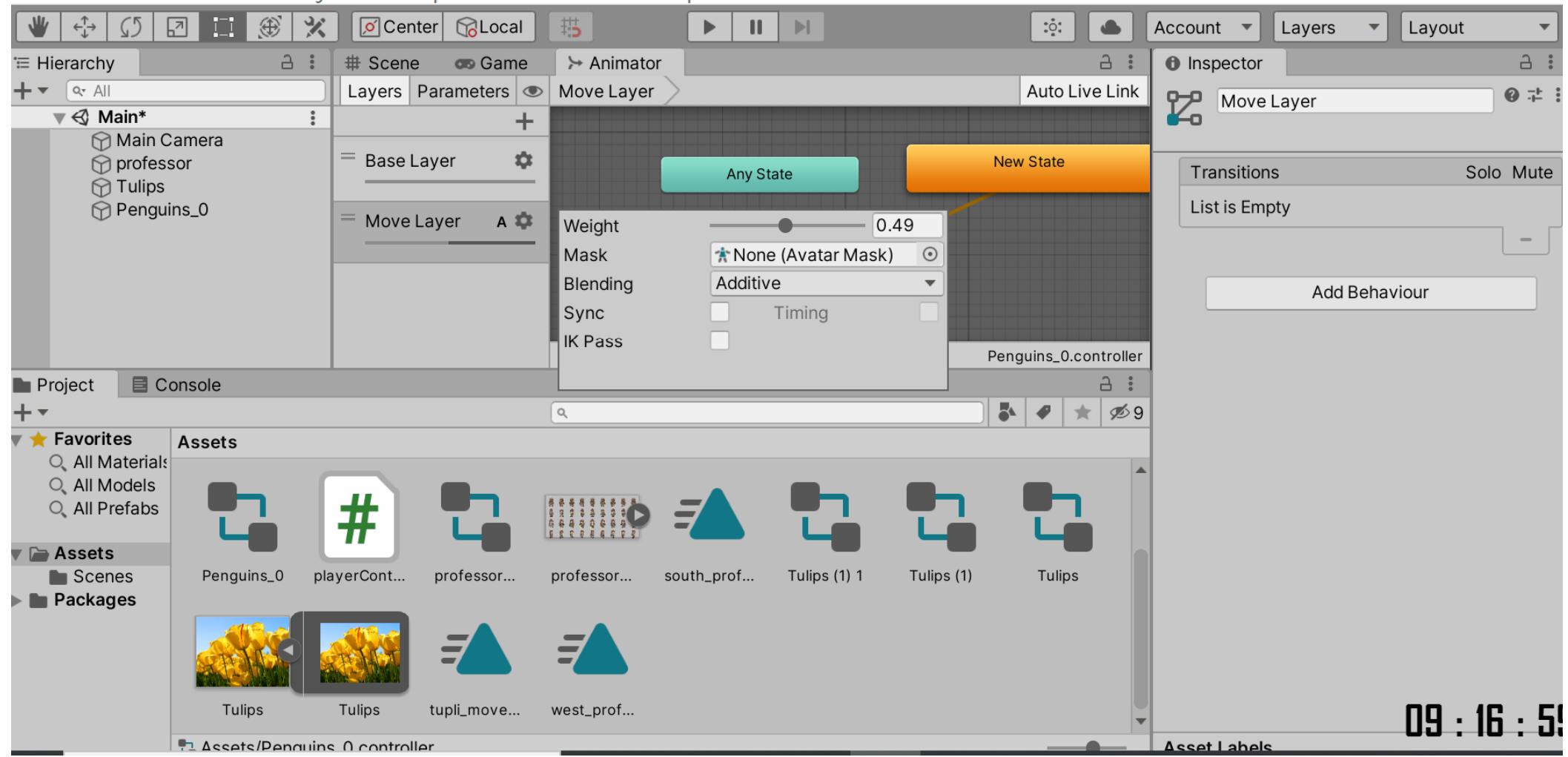
- Scenes
- Penguins_0



Animation clip blending

2DAnimation - Main - PC, Mac & Linux Standalone - Unity 2020.3.2f1 Personal* <DX11>

File Edit Assets GameObject Component Window Help



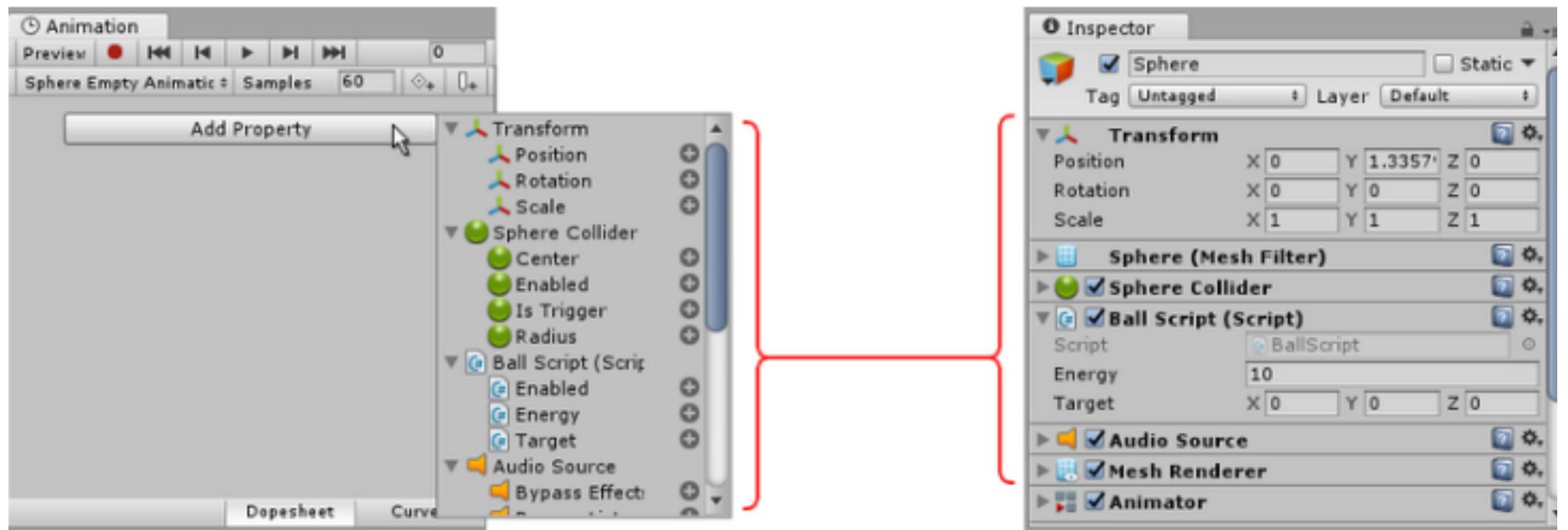
Unity3d Animation Concepts

Animation Types:

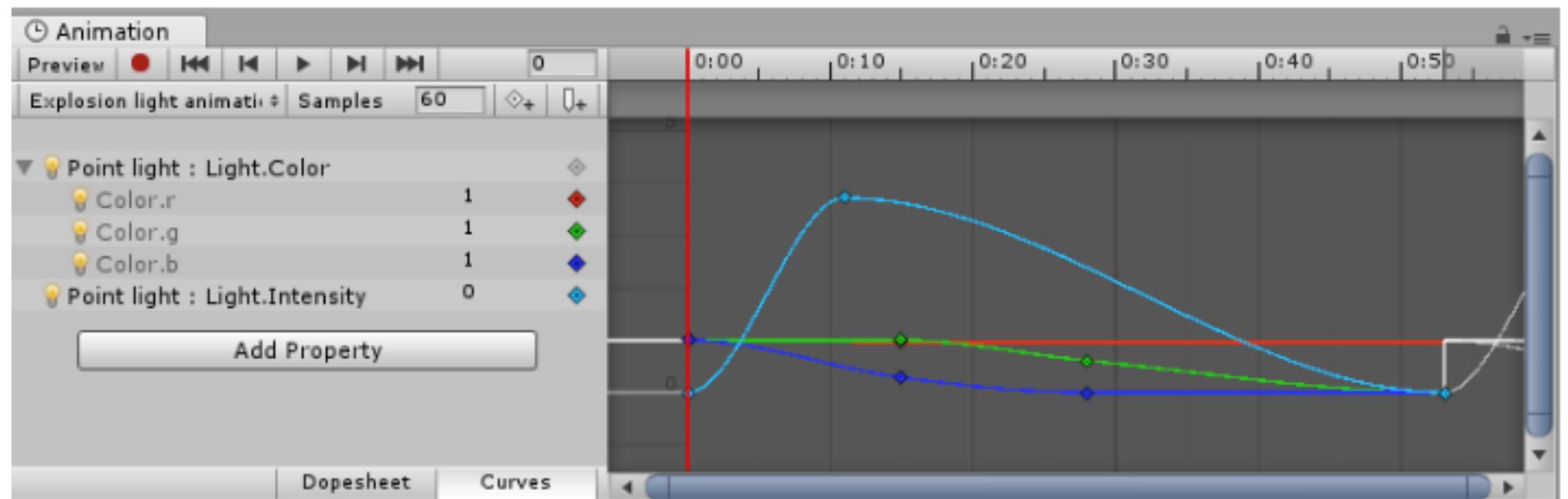
- 2D sprite-sequence animation
- 3D pose-sequence animation
- Animation Created and Edited Within

Unity

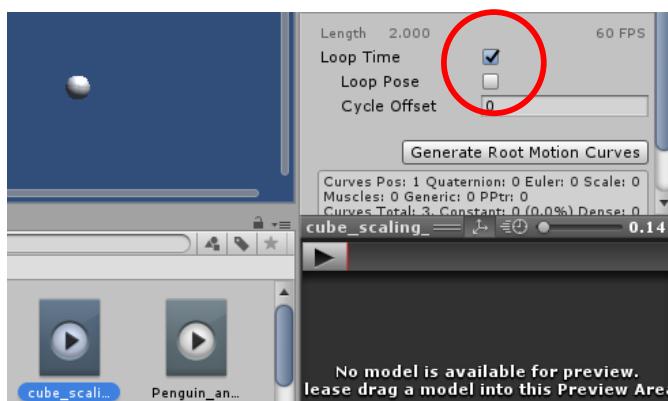
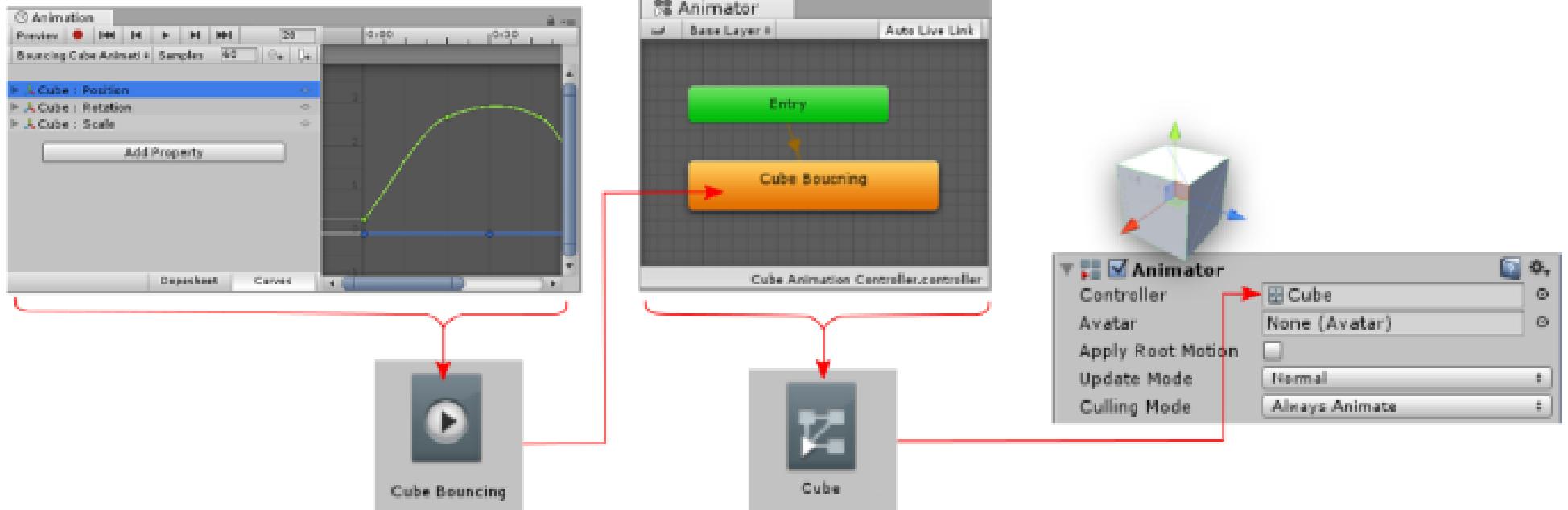
- The position, rotation and scale of GameObjects
- Component properties such as material colour, the intensity of a light, the volume of a sound
- Properties within your own **scripts** including float, integer, enum, vector and Boolean variables
- The timing of calling functions within your own scripts



Click a Game Object → Open Animation Window → Press “Create Button” for Creating Animator Comp and a Controller



Add a Property → Add Key (at a certain frame time) → Change Property values



Animatipn이 반복되려
면, Loop Time을 체크
필요

3D Game Object (with parts being hierarchically organized) in the scene window

~~Animation Component~~ (**old style – obsolete**)

Avatar

Animation Clips

Animator Controller

2D Game Object (called Sprite) in the scene window

~~Animation Component~~ (**old style – obsolete**)

Animation Clip

Animator Controller

1. Animator Component

- Controller field: an **animator controller** asset is assigned.
- The animation controller asset is a state diagram where each state is associated with an animation clip.

2. Animation Component (**old style – obsolete**)

- Animation field: an animation clip (more) is assigned.
- Role: In a script code, those animation clips can be called.

3. How to create animation clips ?

- 1) for 3d: 외부 3D character modeling 툴에서, 애니메이션 클립을 만들어 fbx 파일내 포함
- 2) for 3d: Use the animation window.
 - First click a game objet, and open an animation window.
 - Then create an animation clip using making key frames and setting values of properties.

- 3) for 2d: Use animation window.
- 4) for 2d, Sprite-Sequence Animation Clip:
Use an image sheet (an image file with a sequence of image blocks) and the "sprite editor".
 - First import an image sheet file into the project window → a sprite asset will be created.

- And change the sprite mode of the sprite to "multiple". And open the sprite editor and then slice the whole image into a sequence of image blocks. → the asset has children (image blocks)
 - Then select several of the image blocks and drag them into the scene window
 - ==> a sprite game object, its animator component, its animator controller(with one state), and an animation clip are created.

뼈대 (Bone)을 가진 3D Character Animation

3D Game Object (with parts being hierarchically organized) with

- ~~Animation Component~~
- Animator Controller
- Avatar (humanoid character 인 경우,
다관절-뼈대로 구성된 character)
- Animation Clips

FBX 파일 (animation 포함)을
import하면 자동 생성 됨

File Edit Assets GameObject Component Window Help



Pivot Global



Collab

Hierarchy

Create Q All

Avatar Configuration

Main Camera

Directional Light

ArmyPilot(Clone)

Mesh1

RootFrame

Pelvis

LLegUpper

LLegCalf

LLegAnkle

LLegToe1

Project

Console

Create



< Persp

Favorites

All Materials

All Models

All Prefabs

All Modified

All Conflicted

Assets



Materials



Anim Contr...



ArmyPilot



Controller



FPSMove



game



heart



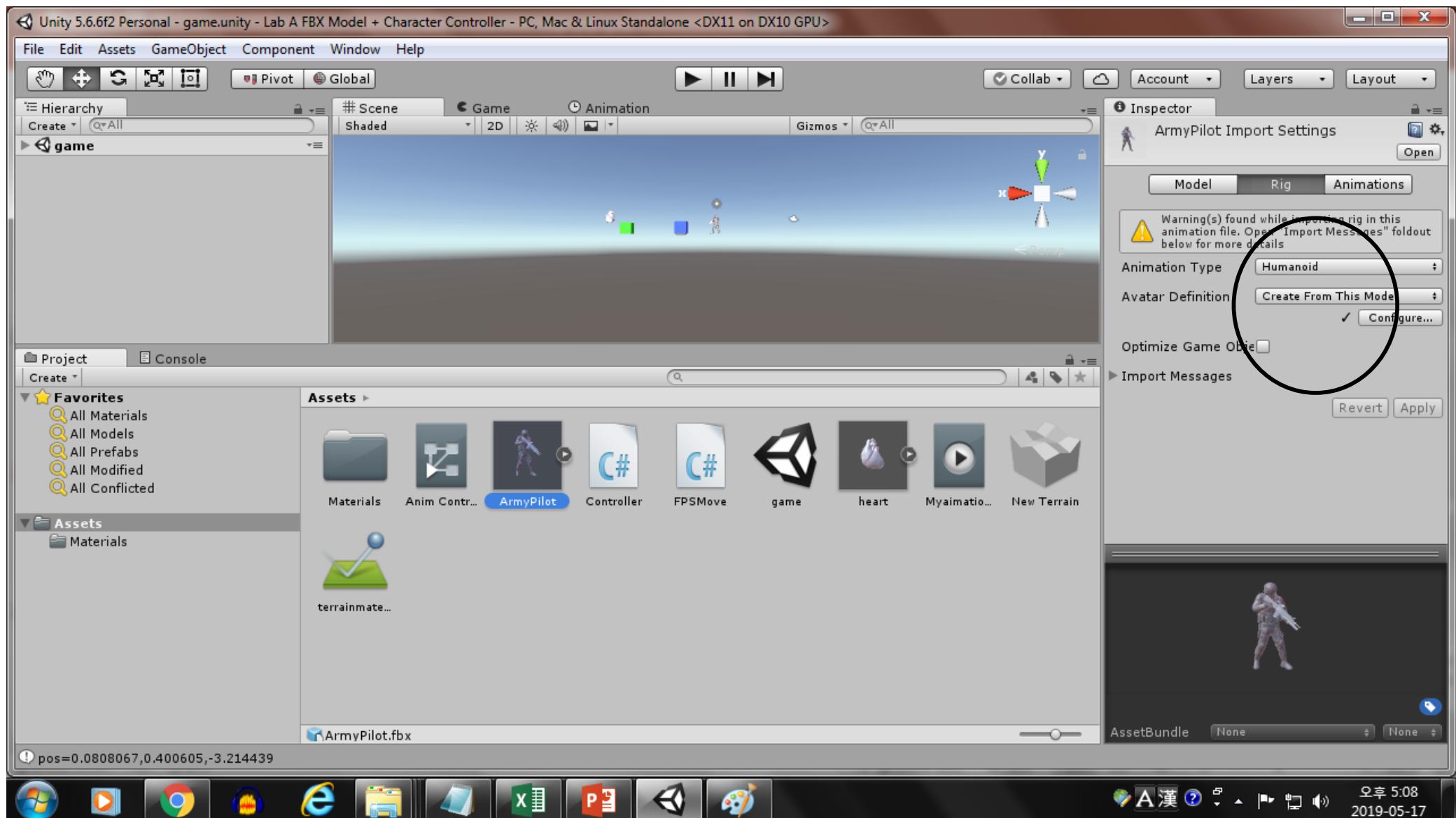
Myaimatio...



New Terrain

Assets

Materials



File Edit Assets GameObject Component Window Help



Pivot Global



Collab



Account

Layers

Layout

Hierarchy

Create Q All

Avatar Configuration

Main Camera
Directional Light
ArmyPilot(Clone)
Mesh1
RootFrame
Pelvis
LLegUpper
LLegCalf
LLegAnkle
LLegToe1

Scene

Game

Animation

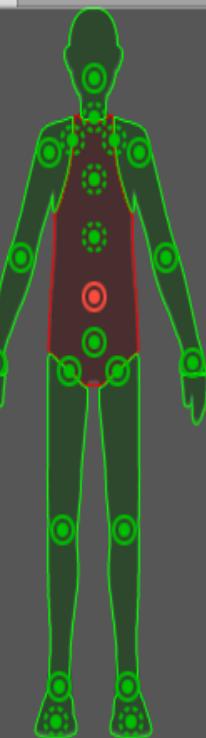
Gizmos



Q All



Inspector



Body

Head

Left Hand

Right Hand

Optional Bone

Body

Hips

Spine

Pelvis (Transf)

Spine1 (Transf)

Mapping Pose

Revert

Apply

Done

Assets

All Materials
All Models
All Prefabs
All Modified
All Conflicted

Assets

Materials

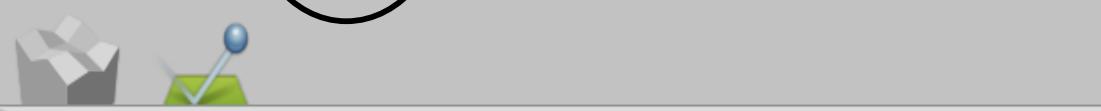
Idle Aim Idle Firing Idle Relo... Jump Left Fire Right FIre Run Firing Run_back...



Run_Forw... Stand Aim... Stand Aim... Standing 2 Strafe Left Strafe Rig... Walk Walk Aimi...



Walk Bac... Walk Firing ArmyPilot... Controller FPSMove game heart Myaimatio...



ArmyPilot.fbx

pos=0.0808067,0.400605,-3.214439



오후 5:07
2019-05-17

File Edit Assets GameObject Component Window Help



Pivot Global



Collab



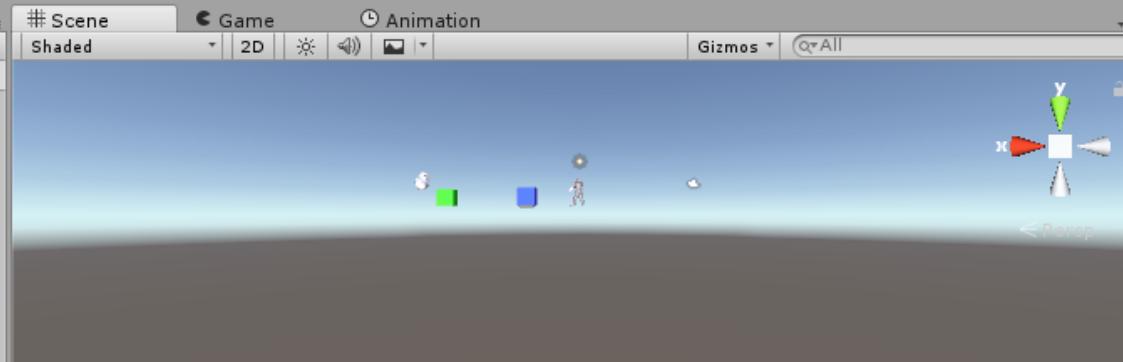
Account

Layers

Layout

Hierarchy

Create All game



Inspector

Aim Running

Length 1.733 30 FPS

Loop Time

Loop Pose

Cycle Offset 0

Root Transform Rotation

Bake Into Pose

Based Upon (at Sf) Body Orientation

Offset 0

Root Transform Position (Y)

Bake Into Pose

Based Upon Original

Offset 0

Root Transform Position (XZ)

Bake Into Pose

Based Upon Center of Mass

Mirror

Aim Running

IK

Frame 9 0:09 (018.2%)

Project

Create

Favorites

- All Materials
- All Models
- All Prefabs
- All Modified
- All Conflicted

Assets

Materials

Assets

Materials Anim Contr... ArmyPilot Mesh1 RootFrame Mesh1 Aim Run... Idle

Idle Aim Idle Firing Idle Relo... Jump Left Fire Right FIre Run Firing Run_back...

Run_Forw... Stand Aim... Stand Aim... Standing 2 Strafe Left Strafe Rig... Walk Walk Aimi...

ArmyPilot.fbx

pos=0.0808067,0.400605,-3.214439



오후 5:10
2019-05-17

Animator Component에서 Controller 속성
값 (Animator Controller)를 click

→ Animator Controller 를 편집하는
window 가 출현.

File Edit Assets GameObject Window Help

Pivot Global

Hierarchy Create (All) game Main Camera Directional Light heart ArmyPilot Mesh1 RootFrame Cube_blue Cube_white Cube_green field Terrain

Scene Game Animation Animator Shaded 2D Gizmos (All)

Collab Account Layers Layout

Inspector

ArmyPilot Static Tag Untagged Layer Default Model Select Revert Open

Transform Position X 0 Y 0.5 Z -3.23 Rotation X 0 Y 107.6° Z 0 Scale X 1 Y 1 Z 1

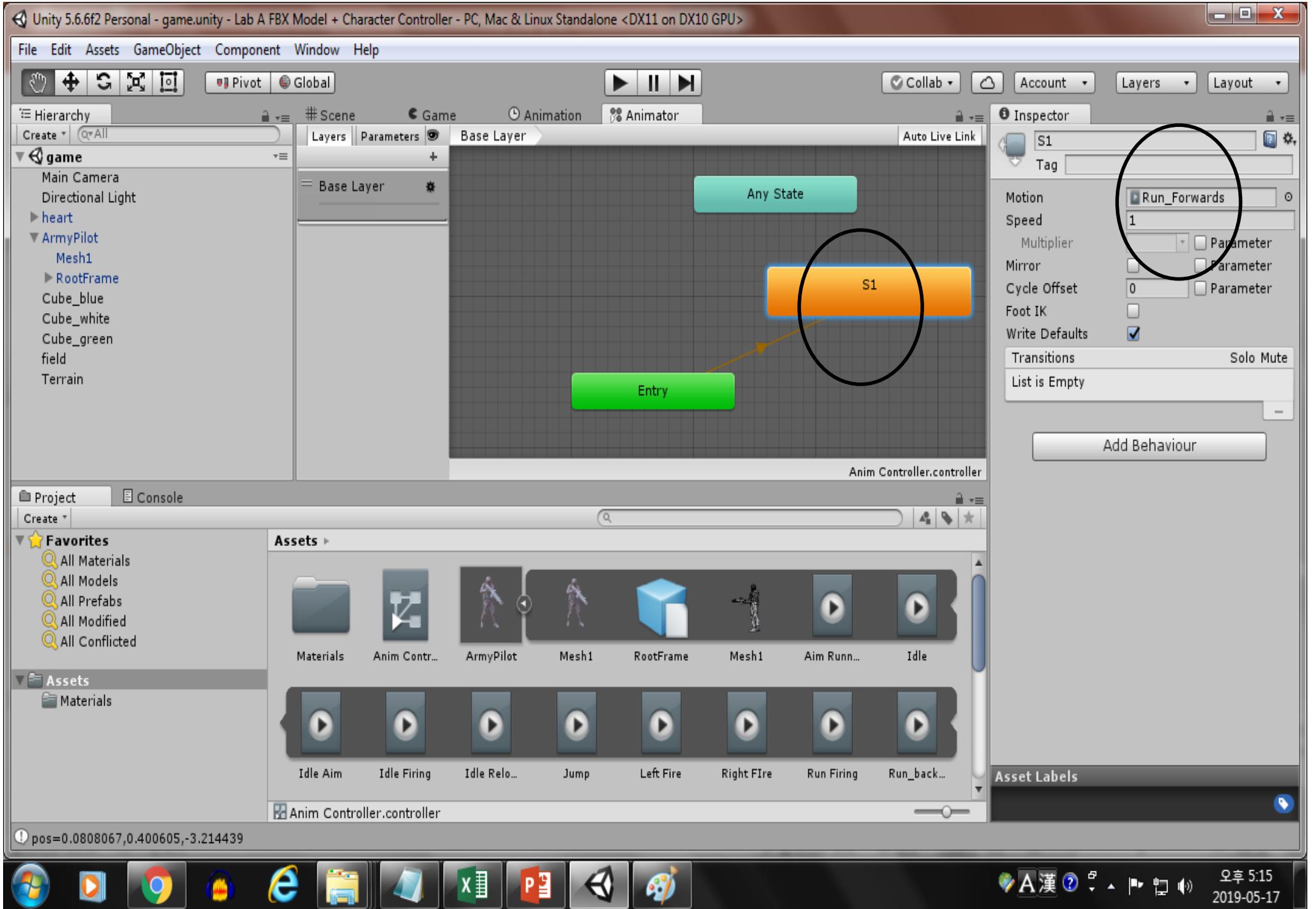
Animator Controller Anim Controller Avatar ArmyPilotAvatar Apply Root Motion Update Mode Normal Culling Mode Cull Update Transforms Clip Count: 0 Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 0 Generic: 0 PPtr: 0 Curves Count: 0 Constant: 0 (0.0%) Dense: 0 (0.0%) Stream: 0 (0.0%)

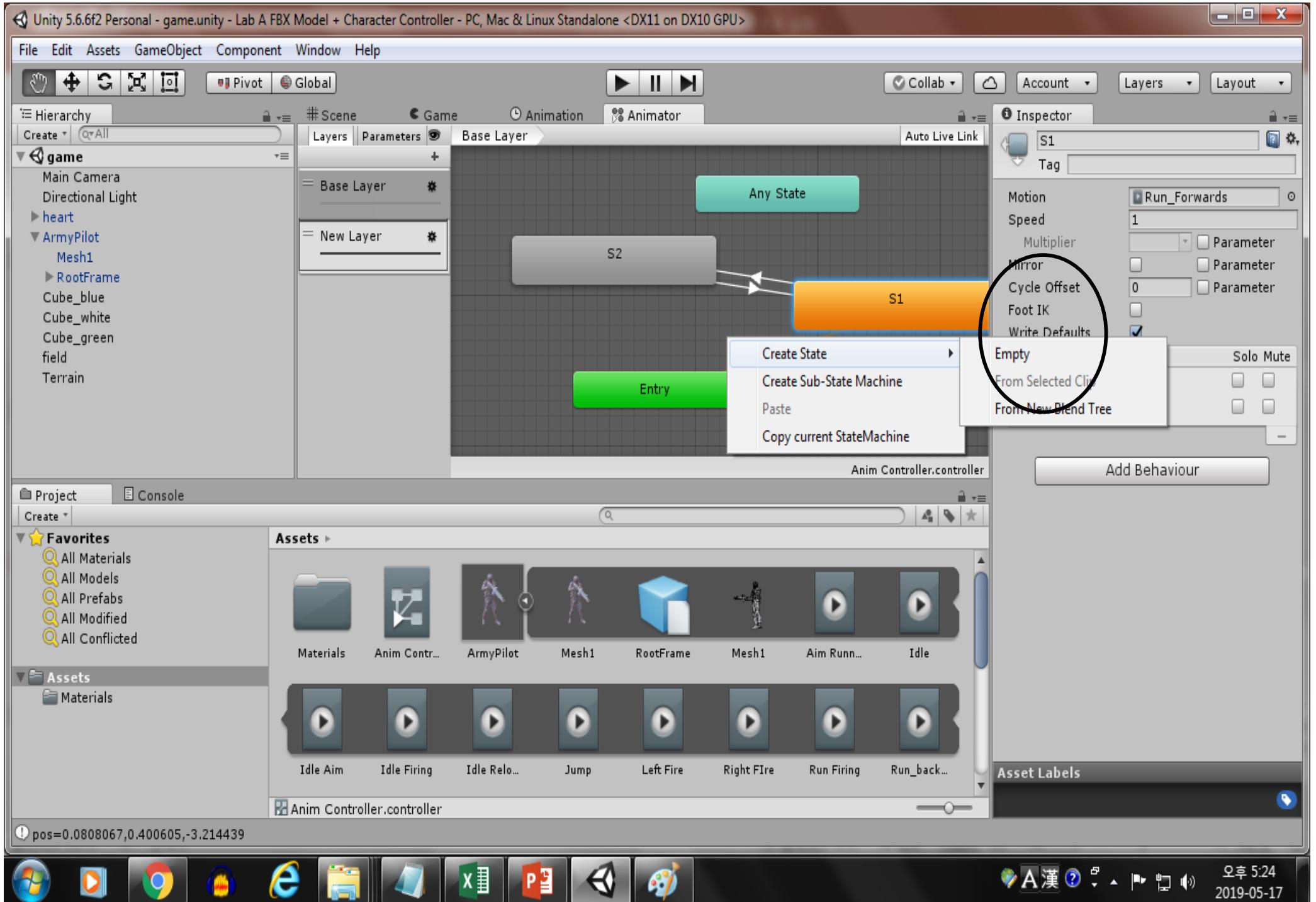
Character Controller Slope Limit 45 Step Offset 0.1 Skin Width 0.0006 Min Move Distance 0.001 Center X 0 Y 0 Z 0 Radius 0.15 Height 0

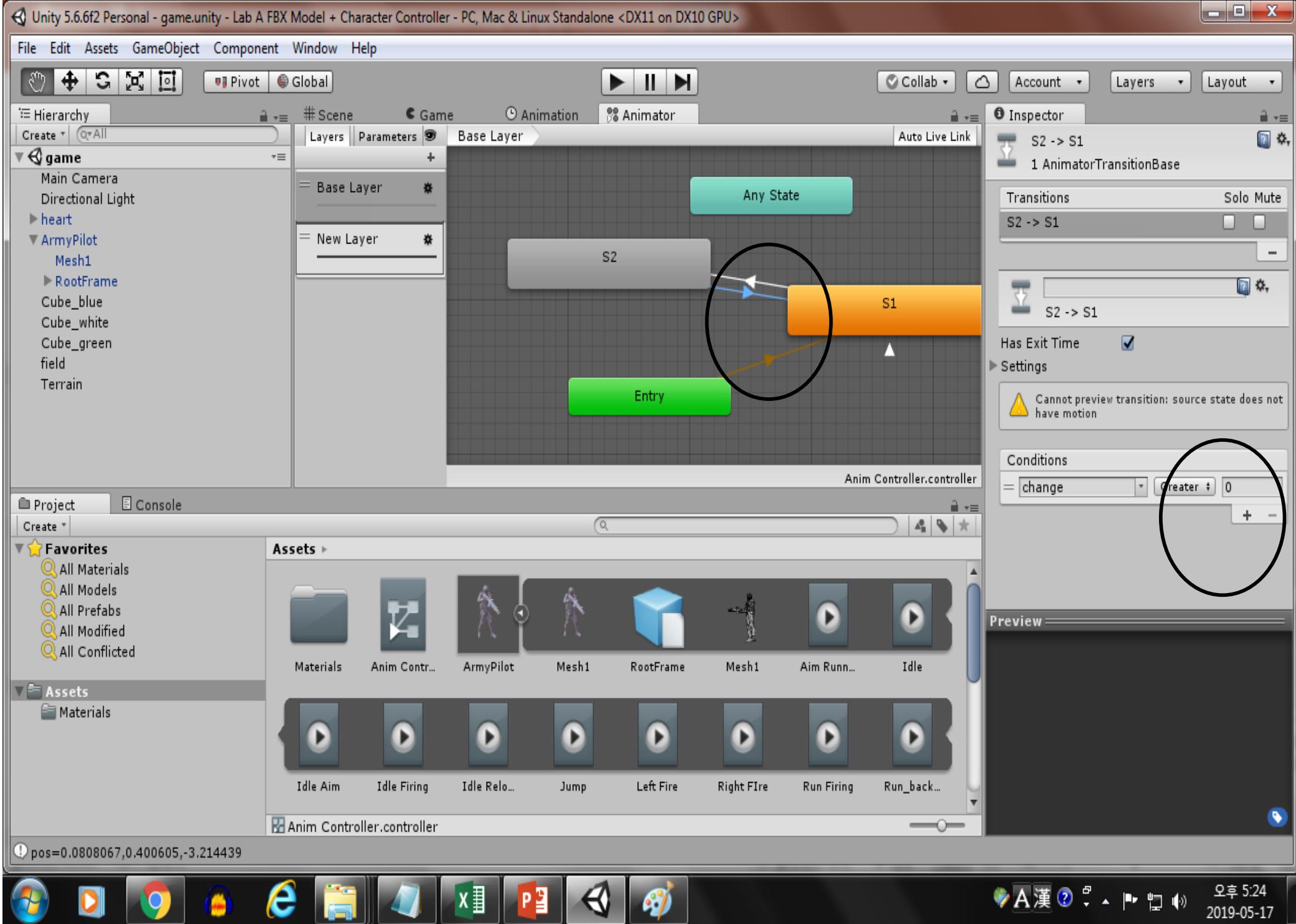
FPS Move (Script) Script FPSMove Speed 6

Assets Idle Aim Idle Firing Idle Relo... Jump Left Fire Right FIre Run Firing Run_back... Run_Forw... Stand Aim... Stand Aim... Standing 2 Strafe Left Strafe Rig... Walk Walk Aim... pos=0.0808067,0.400605,-3.214439

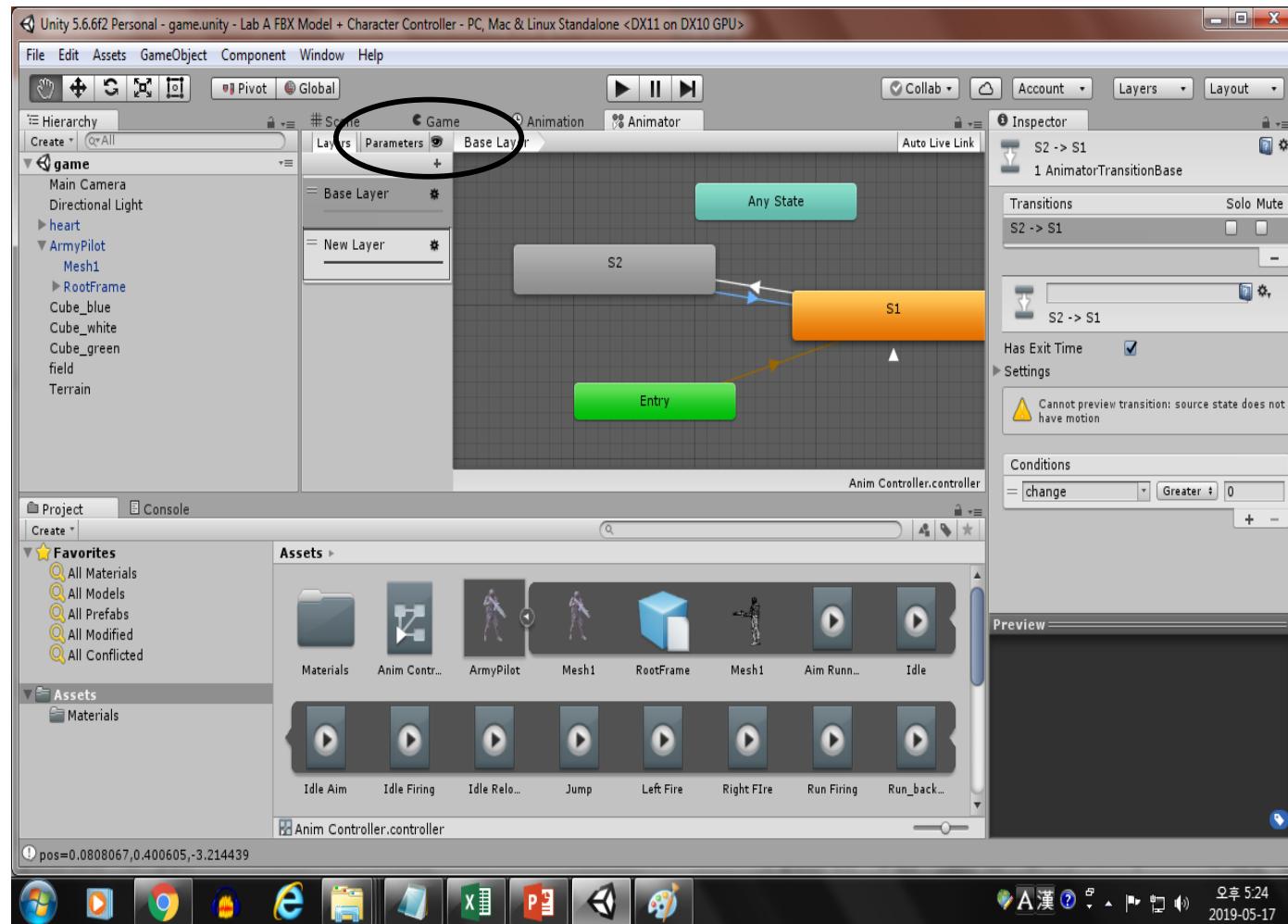








Parameters 템을 click해 → 새로운 parameter을 추가, 기존 것을 삭제



Animation Clip 천이 방법 1

```
...
private Animator animator;
animator = this.GetComponent<Animator>();

...
void Update()    {
    if(조건)
    {
        animator.SetInteger("change", 1);
        //change is a parameter
        ...
    }
    ...
}
}
```

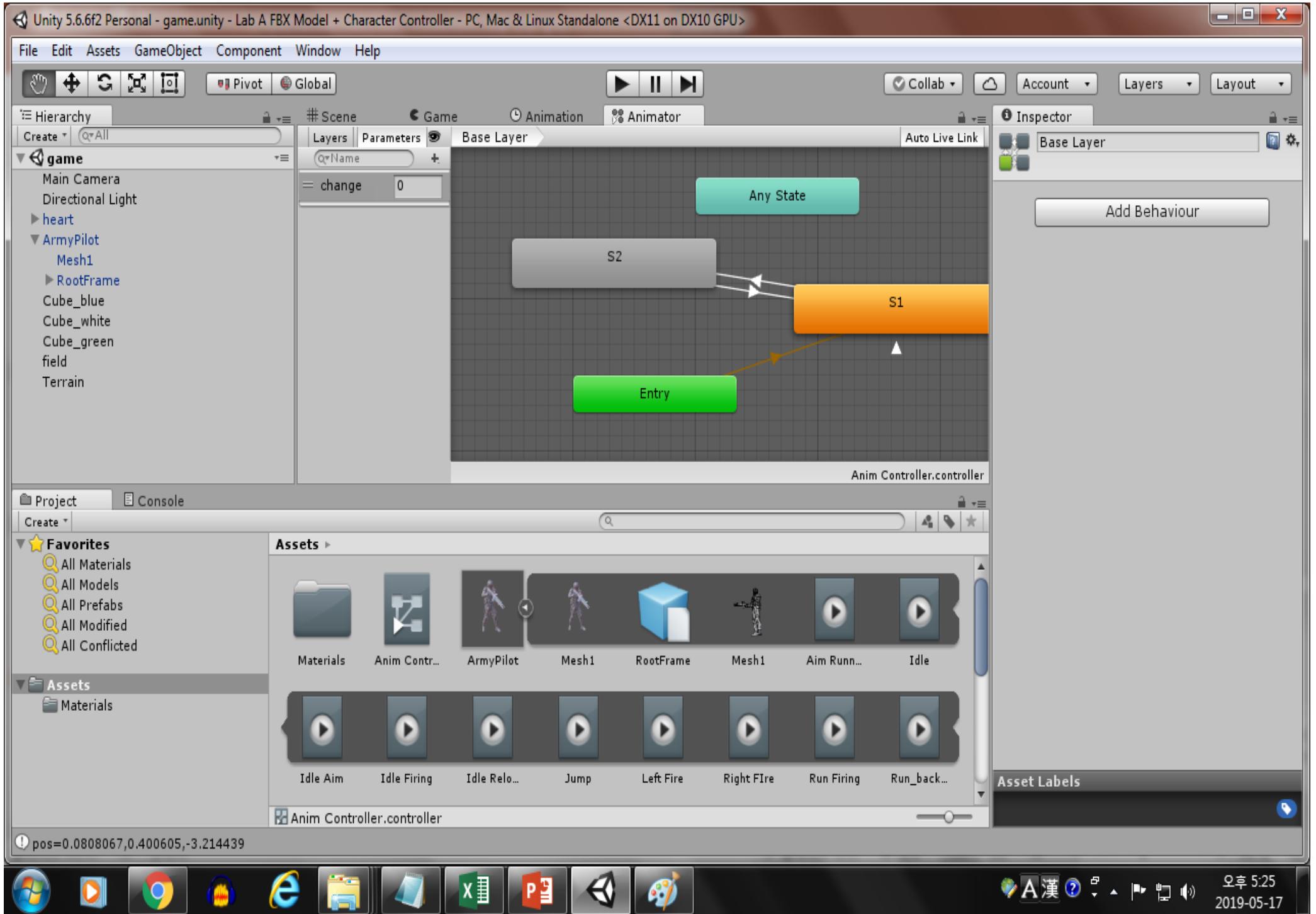
Animation Clip 천이 방법 2

...

```
private Animator animator;  
animator = this.GetComponent<Animator>();
```

...

```
void Update()    {  
    if(조건)  
    {  
        animator.SetTrigger("Jump");  
        //Jump is an animation clip  
  
        ...  
    }  
    ...  
}
```



Animation Loop 속성

Inspector 창에서 AnimationClip 선택 후,

- 'Loop Time'을 check하면 반복
- 'Loop Pose'를 check하면,
Enable to make the motion loop

seamlessly

(시작 frame과 last frame의 pose를 일치)

File Edit Assets GameObject Component Window Help

Scene View (Persp) showing a 3D model of a soldier named "ArmyPilot".

Inspector Panel:

- Stand Aim Down
- Stand Aim Up
- Standing 2

Run_Forwards Animation Clip:

- Source Take: Run_Forwards
- Length: 2.300
- Timeline: 0:00, 0:10, 0:20, 1:00, 1:10
- Start: 0
- Loop Time:
- Loop Pose:
- Cycle Offset: 0

Bottom Status Bar:

- 스피커: 75% (0.0%) Frame 0
- 오전 11:12
2020-04-19

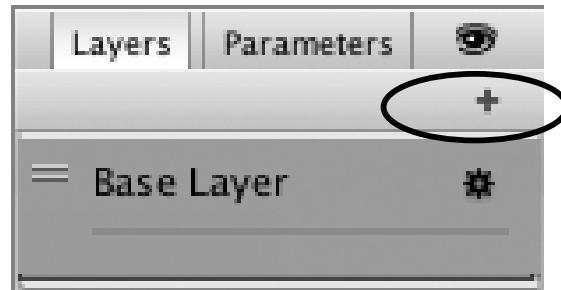
Toolbars and Buttons:

- Hand, Selection, Transform, Box Selection, Grid Selection, Pivot, Local
- Play, Stop, Next Frame, Collab, Account, Layers

Project and Assets Panels:

- Project: Favorites (All Materials, All Models, All Prefabs, All Modified, All Conflicts), Assets (Fonts, Materials, 2d_test_an..., Anim Contr..., anim_stop, anim_test, ArmyPilot, Mesh1)
- Assets: Fonts, Materials, 2d_test_an..., Anim Contr..., anim_stop, anim_test, ArmyPilot, Mesh1

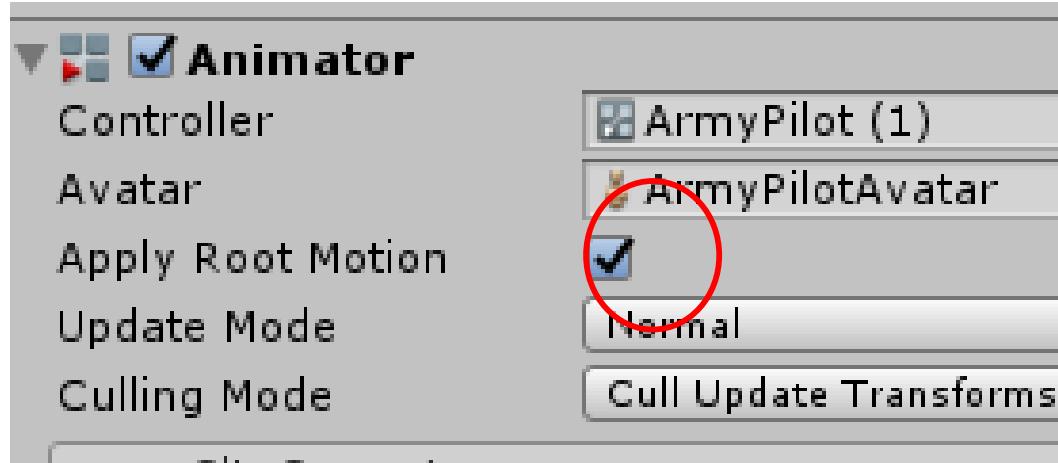
애니메이션 레이어



- 다른 신체 부분에 대하여 복잡한 스테이트 머신들을 관리 할 수 있음. 예를 들어, 하체 레이어에서 걷거나 점프하고, 상체 레이어에서 오브젝트를 던지거나 쏘는 것과 같은 경우가 있습니다.

Root Motion, Root Transform

- <https://m.blog.naver.com/PostView.nhn?blogId=dj3630&logNo=221469009943&proxyReferer=https%2F%2Fwww.google.com%2F>



<https://www.youtube.com/watch?v=Kn6jxLWA31M>



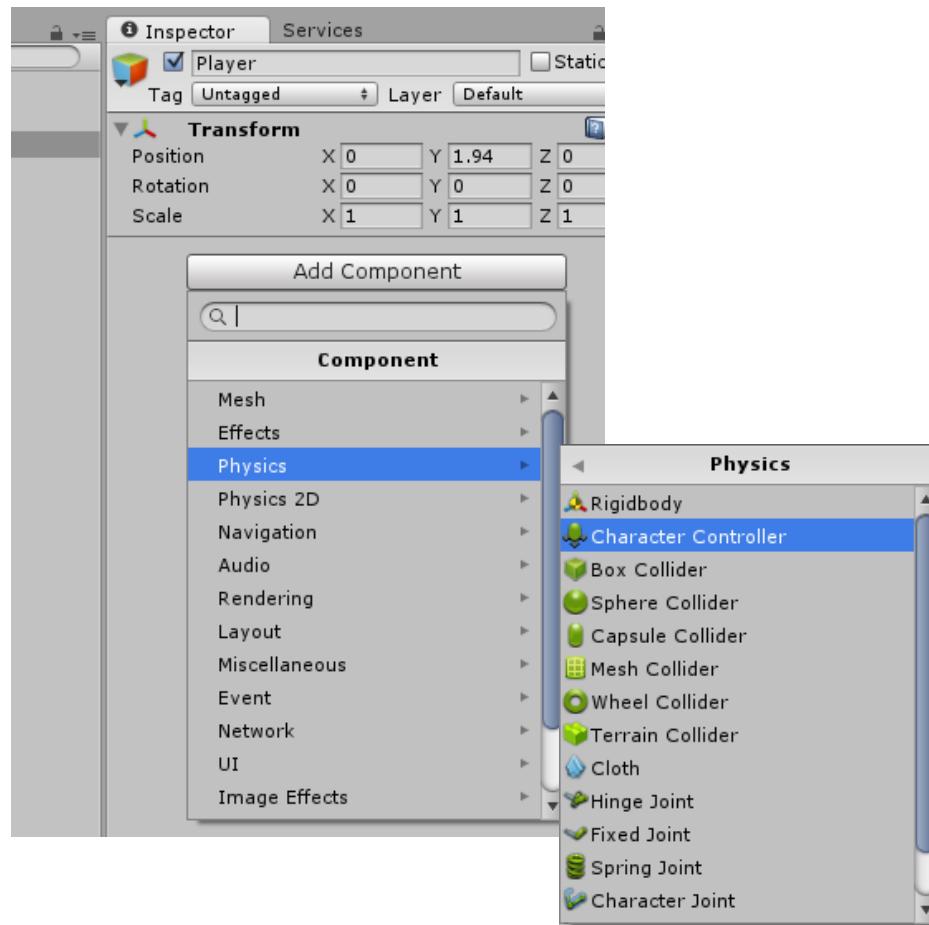
- 루트 모션은 모캡 애니메이션의 변위/회전 값을 유니티에 적용할 것인지 설정하는 옵션이다. 루트 모션을 적용하면 스크립트 없이 애니메이션이 갖고 있는 변위/회전 값만 이용해서 게임오브젝트를 움직일 수 있다. 루트 모션으로 움직이면 스크립트로 움직이는 물체보다 자연스러운 움직임을 표현할 수 있는 경우가 있다. 만약 모캡 애니메이션의 변위/회전 값을 적용하고 싶지 않다면 Animator 컴포넌트에서 Apply Root Motion 을 체크 해제한다.

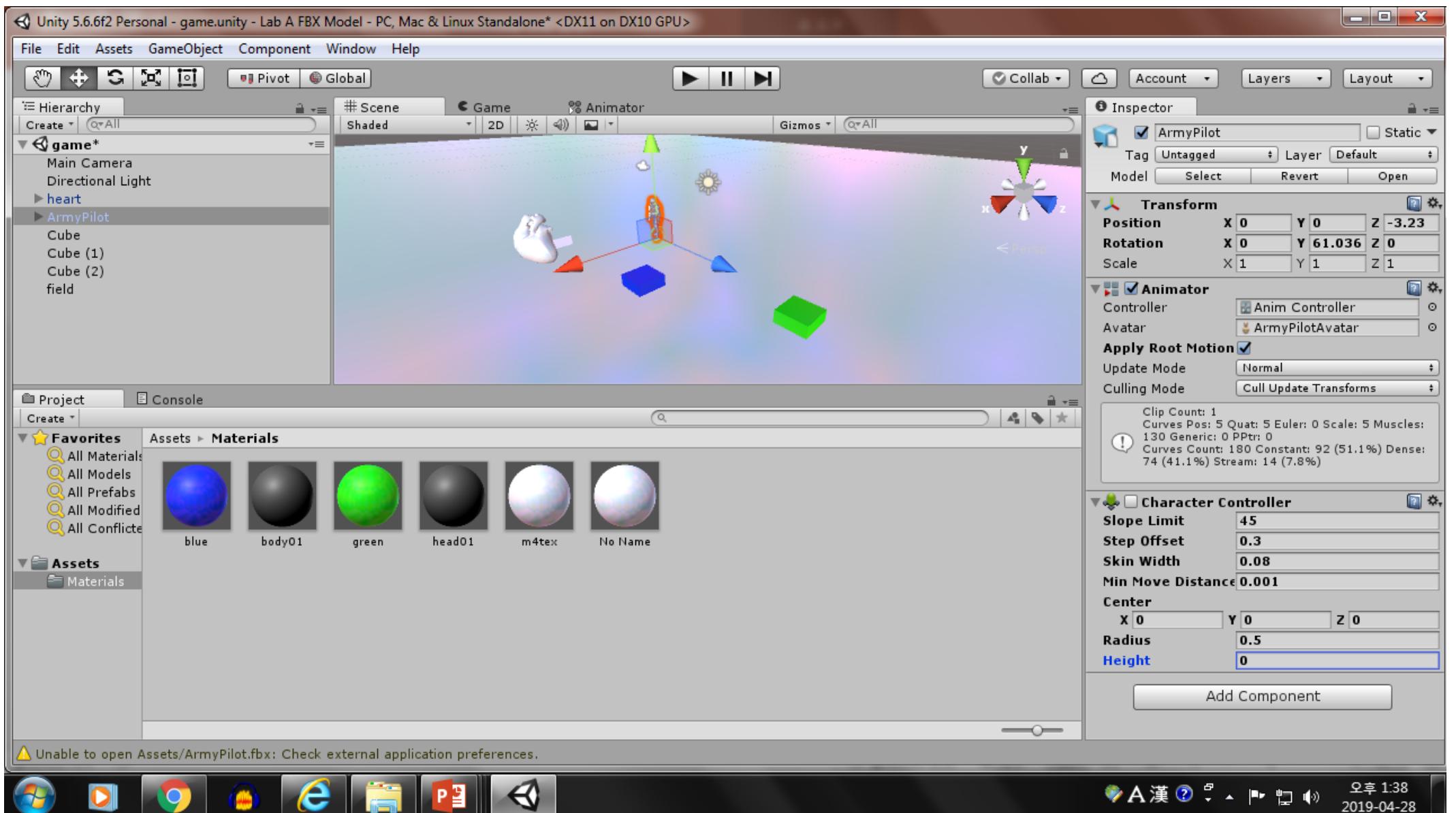
root motion을 사용하지 않으면, Script에서 방향바꿈(회전) 및 위치이동(변이)를 제어해야 함.

animation clip이 동작하는 character object
에 gravity를 적용하려면?

- Rigidbody & Collider Component가 있어야 함
- Animator Component의 “Apply Root Motion”을 unchecked 해야 함.

Character Controller Component





프로퍼티:	기능:
Slope Limit	콜라이더가 표시된 값 이하의 기울기만을 오르도록 제한합니다. (degree)
Step Offset	표시된 값보다 지면에 가까운 경우에만 캐릭터가 계단을 오릅니다. This should not be greater than the Character Controller's height or it will generate an error.
Skin width	두 콜라이더가 Skin Width와 동일한 깊이에서 서로 관통합니다. Skin Width가 커지면 지터가 줄어듭니다. Skin Width가 낮으면 캐릭터가 움직일 수 없게 되는 경우가 있습니다. Radius가 10%일 때 이 값을 설정하는 것이 좋습니다.
Min Move Distance	캐릭터가 표시된 값 미만으로 움직이려 해도 움직이지 않습니다. 지터(Jitter)를 줄이는 데 사용할 수 있습니다. 대부분의 경우 이 값은 0의 상태로 두십시오.

프로퍼티:

기능:

Center

월드 공간에서 캡슐 콜라이더가 상쇄되지만, 캐릭터 피벗이 어떻게 회전하는 지에는 영향을 주지 않습니다.

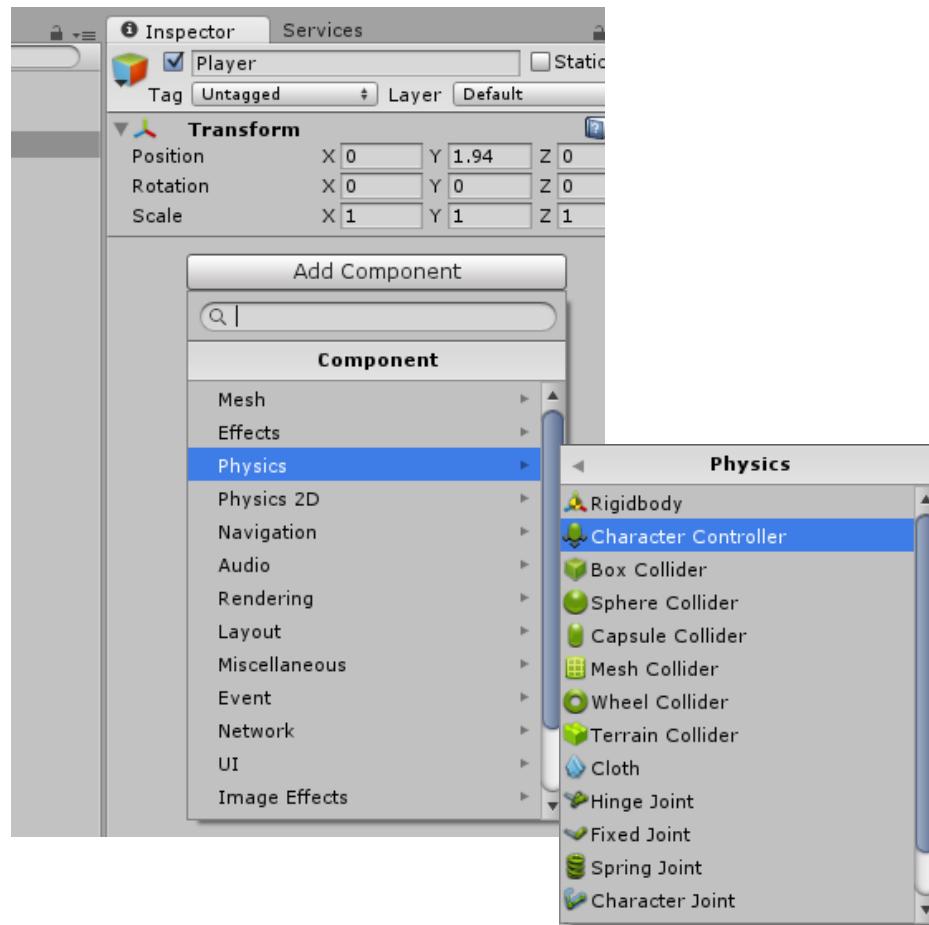
Radius

캡슐 콜라이더 로컬 반경. 이것은 기본적으로 콜라이더의 폭입니다.

Height

캐릭터의 Capsule Collider 높이. 이 값을 변경하면 양수와 음수의 방향으로 Y축을 따라 콜라이더가 확대 축소됩니다.

A player controlled object that has a Character Controller with a movement script that's pretty much based on the script.



Lab11_FBX Model + Character Controller

```
public class FPSMove : MonoBehaviour {  
  
    public float speed = 6;  
    public float jump = 8;  
    public float gravity = 20;  
    Vector3 move = Vector3.zero;  
  
    CharacterController con;  
  
    // Use this for initialization  
    void Start () {  
        con = GetComponent<CharacterController>();  
    }  
}
```

```
// Update is called once per frame
void Update () {
    if( con.isGrounded ) {
        move = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical"));
        move *= speed;
        if( Input.GetButton("Jump") ) {
            move.y = jump;
        }
    }
    else
        move.y = -gravity * Time.deltaTime;
}

Vector3 moveDirection = transform.TransformDirection(move);
con.Move(moveDirection); // new Vector3(0, 0, 0.01f));
}
```

CharacterController.Move (vector3 delta)

절대값의 이동의 델타를 가지고 좀더 복잡한 이동을 하는 함수.

모션은 충돌에 의해 제한됩니다. 콜라이더를 따라 슬라이드함.

함수는 중력에는 적용되지 않음.

C# Script 에서, 새로운 animation
clip (animation controller에서는
state)으로 변경

```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour {
    public float speed = 6.0F;
    public float jumpSpeed = 8.0F;
    public float gravity = 20.0F;
    private Vector3 moveDirection = Vector3.zero;
    void Update() {
        CharacterController controller =
GetComponent<CharacterController>();
        if (controller.isGrounded) {
            moveDirection = new Vector3(Input.GetAxis("Horizontal"), 0,
Input.GetAxis("Vertical"));
            moveDirection = transform.TransformDirection(moveDirection);
            moveDirection *= speed;
            if (Input.GetButton("Jump"))
                moveDirection.y = jumpSpeed;
        }
        moveDirection.y -= gravity * Time.deltaTime;
        controller.Move(moveDirection * Time.deltaTime);
    }
}
```

```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour {
    public float speed = 6.0F;
    public float jumpSpeed = 8.0F;
    public float gravity = 20.0F;
    private Vector3 moveDirection = Vector3.zero;
    void Update() {
        CharacterController controller =
GetComponent<CharacterController>();
        if (controller.isGrounded) {
            moveDirection = new Vector3(Input.GetAxis("Horizontal"), 0,
Input.GetAxis("Vertical"));
            moveDirection = transform.TransformDirection(moveDirection);
            moveDirection *= speed;
            if (Input.GetButton("Jump"))
                moveDirection.y = jumpSpeed;
        }
        moveDirection.y -= gravity * Time.deltaTime;
        controller.Move(moveDirection * Time.deltaTime);
    }
}
```

Unity 'humanoid 아바타' 애니메이션 동영상

<https://m.youtube.com/watch?v=wdOk5QXYC6Y>

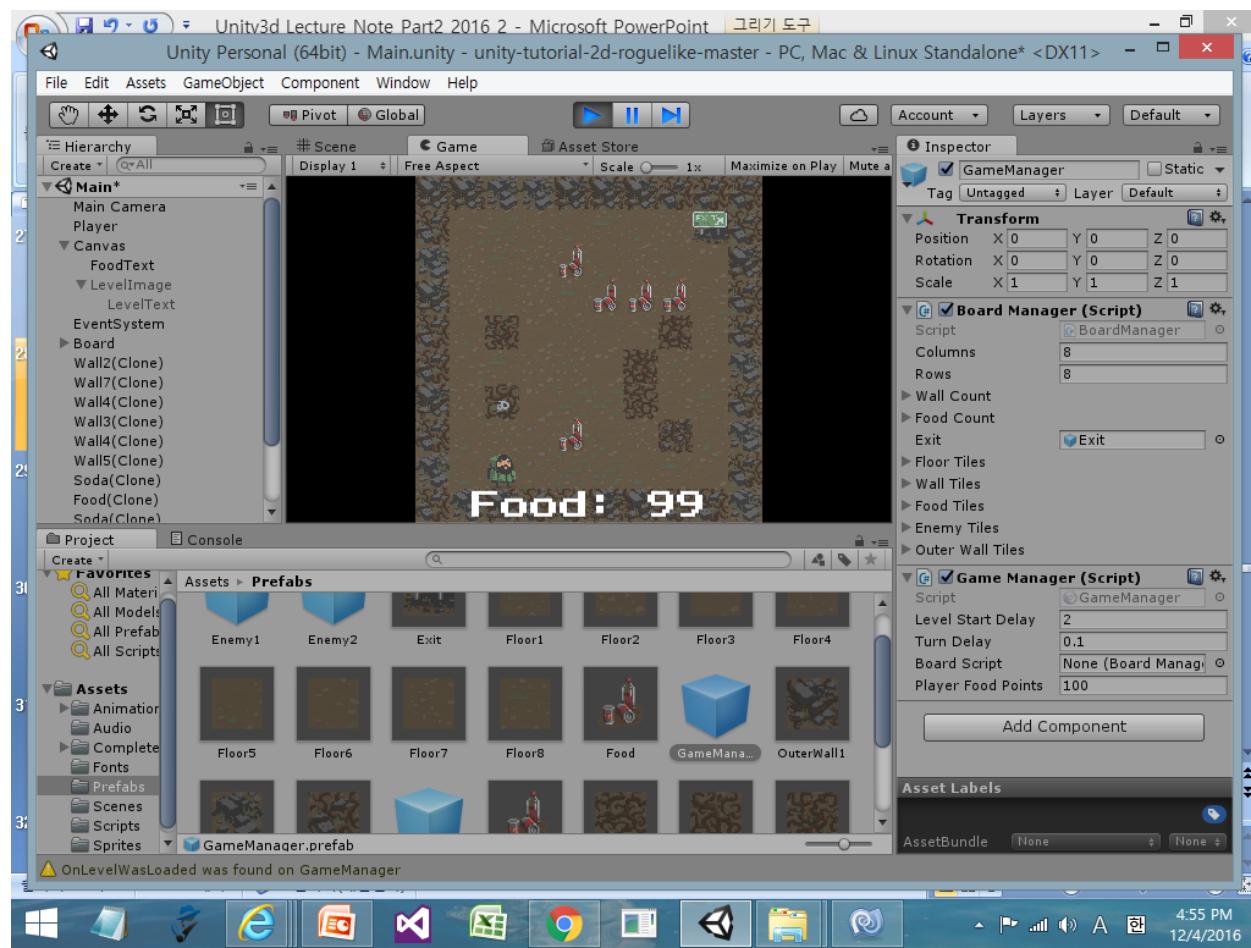
실전 Game Project-1

Understand “Lab100_2d_roguelike” Project

<https://unity3d.com/kr/learn/tutorials/s/2d-roguelike-tutorial>

<https://unity3d.com/kr/learn/tutorials/s/2d-roguelike-tutorial>

(소스코드 및 tutorial)



File Edit Assets GameObject Component Window Help



Pivot Global



Account

Layers

Default

Game

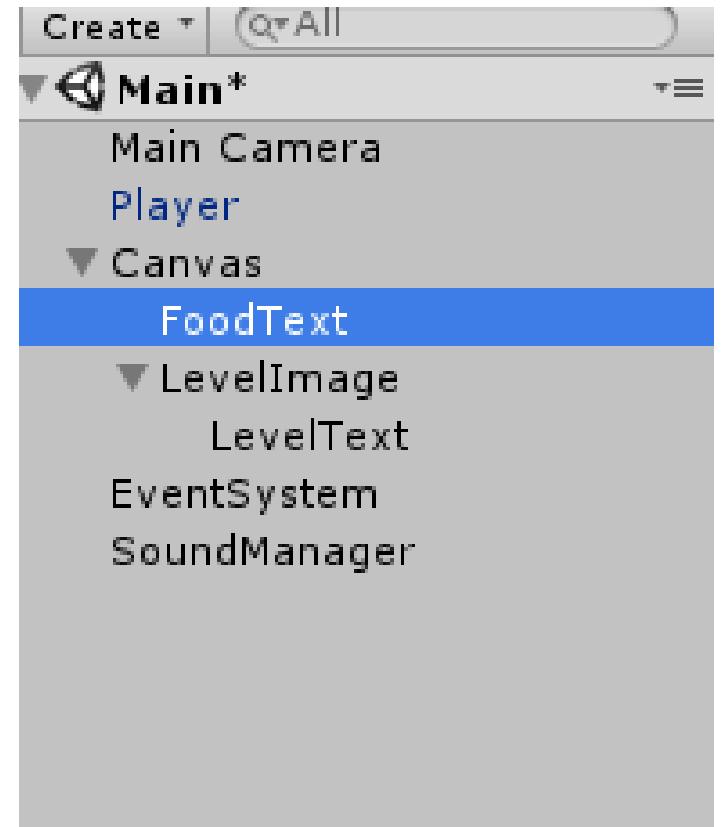
Display 1 Free Aspect

Scale 1x

Maximize on Play Mute audio Stats Giz



- Hierarchy Window
 - Player: 주인공 캐릭터
 - FoodText: food 잔량 표시
 - LevellImage: Level 표시
 - SoundManager: 사운드 처리용 script(SounderManager.cs)
부착



- Camera game object
 - Loader.cs 가 부착. GameManager prefab을 instantiation 시킴. 이 prefab은 BoardManager.cs, GameManager.cs로 구성
- Player game object
 - Player 라는 Animator Controller를 가짐
 - Box 2D Collider 컴포넌트를 가짐
 - Player.cs 가 부착. Player class는 MovingObject class의 subclass

- GameManager.cs
 - List<Enemy> enemies 변수를 Enemy 객체들로 채움
 - Update() 함수내에서, StartCoroutine(MoveEnemies) 호출
- BoardManager.cs
 - Board에 floor tile, wall tile, food tile, enemy tile, outerWall tile 들을 배치
- Player.cs
 - 키보드 또는 마우스 입력을 받아, player를 move 시킴
 - 만약 player가 food 와 만나면, food 점수 상승. 만약 wall가 만나면 food 점수 감소. 만약 food가 0되면, game over.
 - Player가 outerWall과 만나면 Application.Level (Application.loadedLevel) 호출 → scene을 구성하는 모든 game object들을 새로이 load하게 됨

참고: GameManger class내에 DontDestroyOnLoad (gameObject) 호출.

- **Animator** animator = GetComponent<Animator>();
animator.**SetTrigger** ("playerChop"); → "playerChop" state의
animation clip으로 전환
- **RaycastHit2D** hit
hit = **Physics2D.Linecast** (start, end, blockingLayer); → start 지점
에서 end 지점까지 ray를 발사해서 어떤 물체가 있는지를 검사. 만약
없다면 hit.transform은 null임. 이때 blockingLayer 변수의 값(즉
layer 이름)에 배치된 물체만 검사함.
****Physics2D.Raycast** can be used to get information from
objects you're casting the ray at, **Linecast** returns true if
there is a collider between the start and the endpoint of
the line.*
- **Invoke** ("Restart", 1); → "Restart" 함수를 1 sec후에 실행함

- In GameManager class

```
void Awake () {
    if (instance == null)
        instance = this;
    else if (instance != this)
        Destroy (gameObject);

    DontDestroyOnLoad (gameObject);
    enemies = new List<Enemy> ();
    boardScript = GetComponent<BoardManager> ();
    InitGame ();
}

//Obsolete function
void OnLevelWasLoaded (int index) {
    level++;
    InitGame ();
}
}
```