



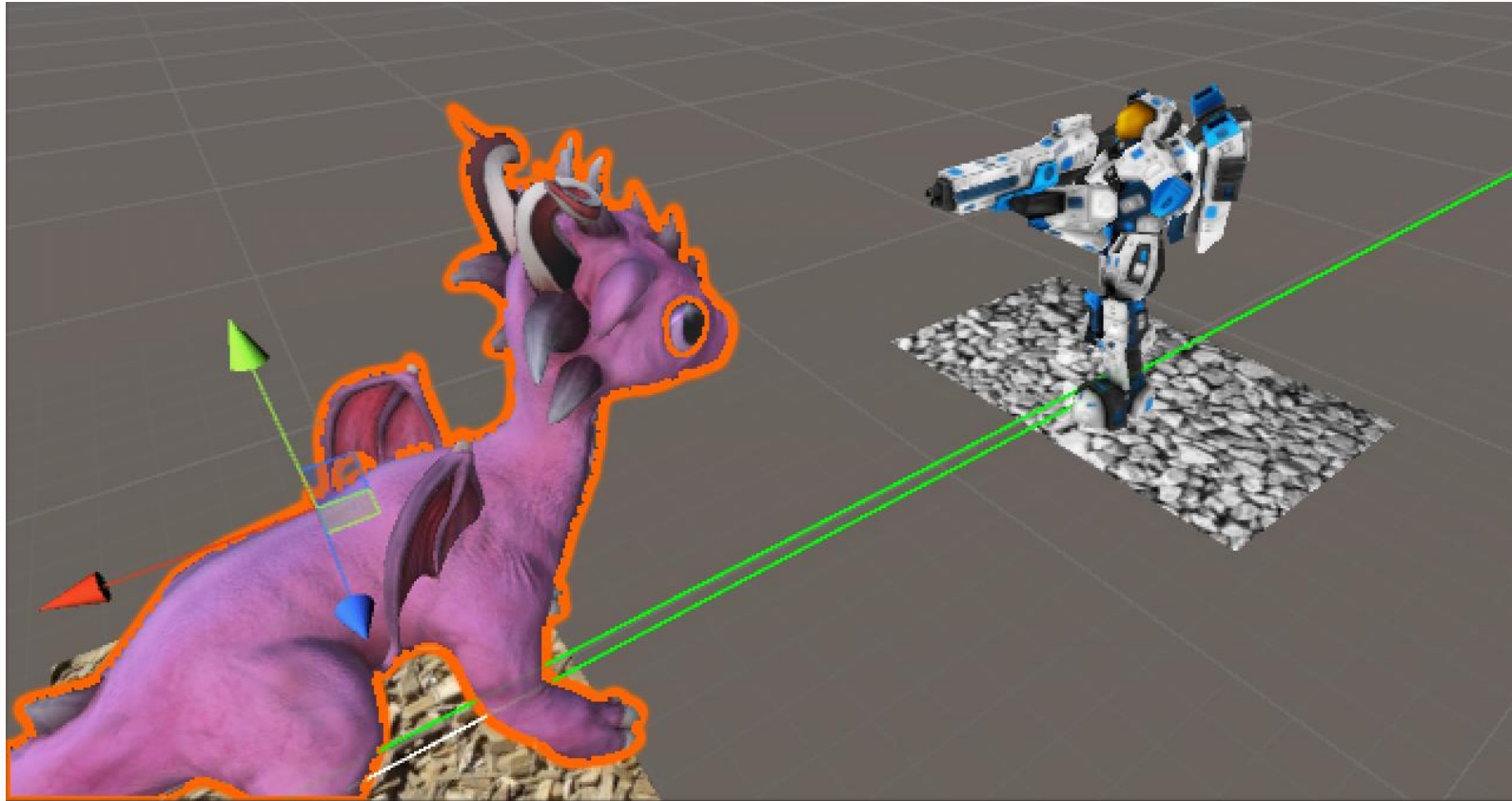


# AR 프로그래밍

이준

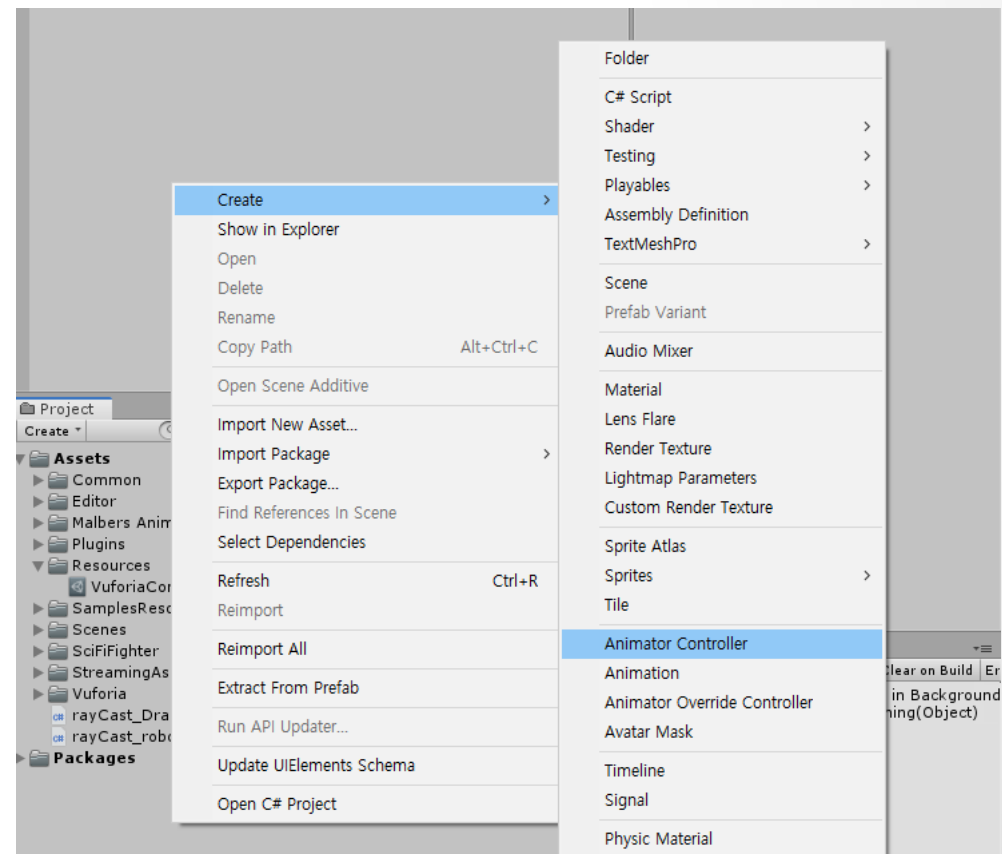
# 카드 배틀 게임 만들기

- 서로 마주 보면 Hit 메시지가 발생!



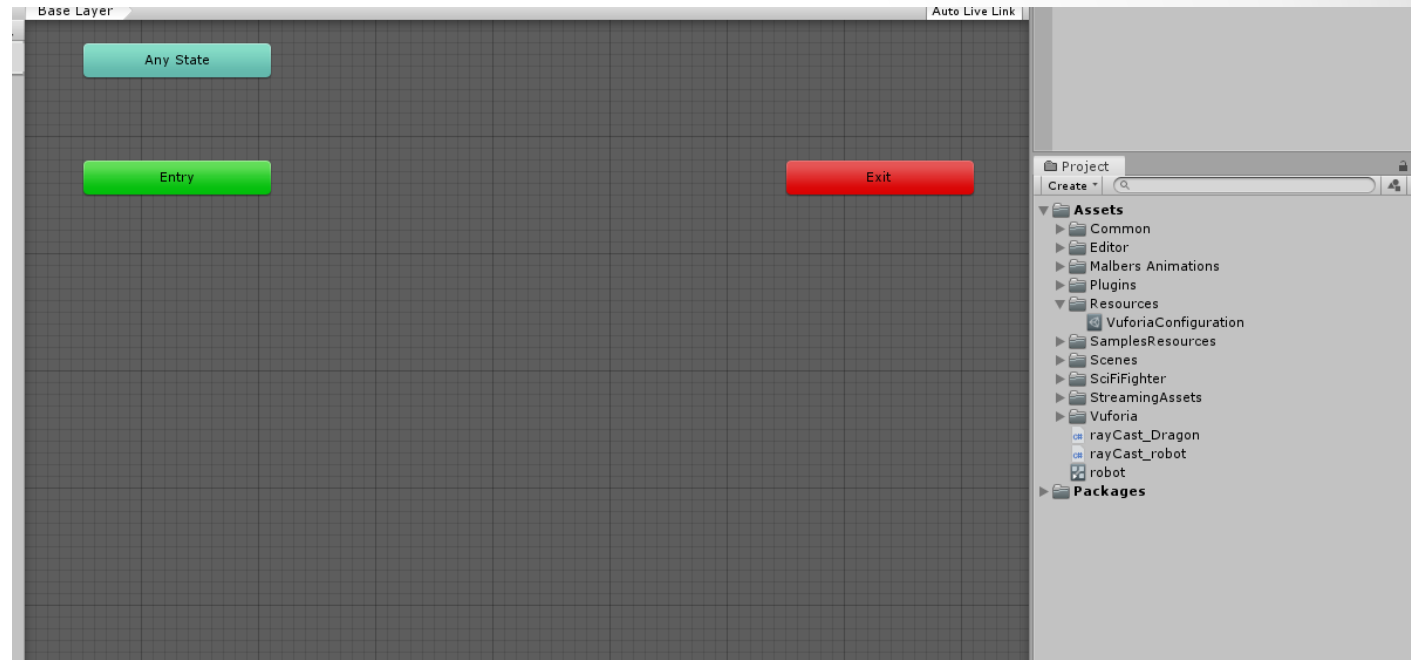
# 카드 배틀 게임 만들기

- 로봇 애니메이션 컨트롤러 넣기
  - 애니메이션 컨트롤러를 사용해야지만 각 상태에 따른 애니메이션 제어 가능!



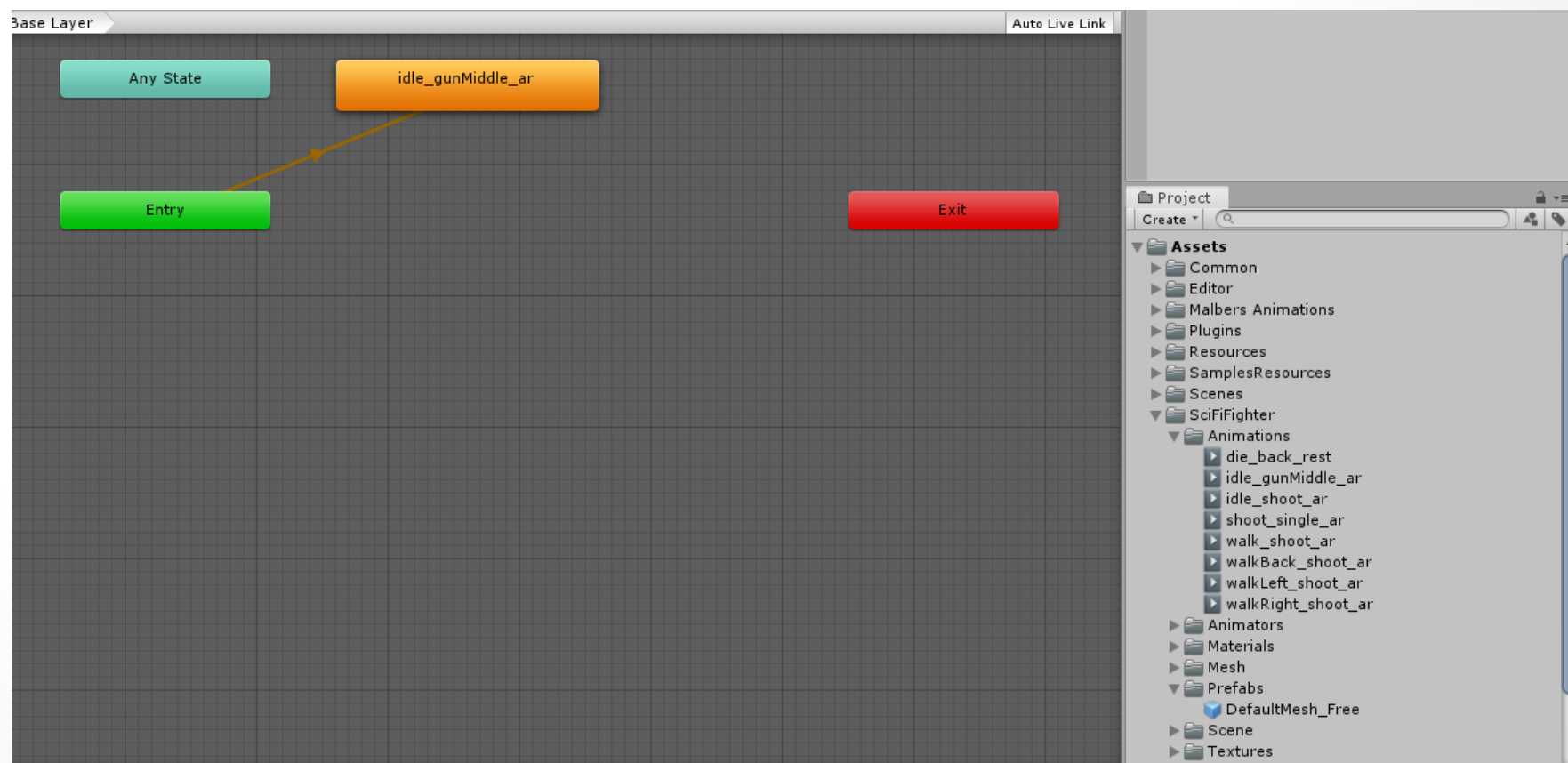
# 카드 배틀 게임 만들기

- robot 애니메이션 컨트롤러 더블 클릭
  - 다음의 그림과 같은 상태가 뜬
  - AnyState : 어떤 상태에서든지 이동
  - Entry : 시작할때의 상태
  - Exit : 애니메이션 종료 상태



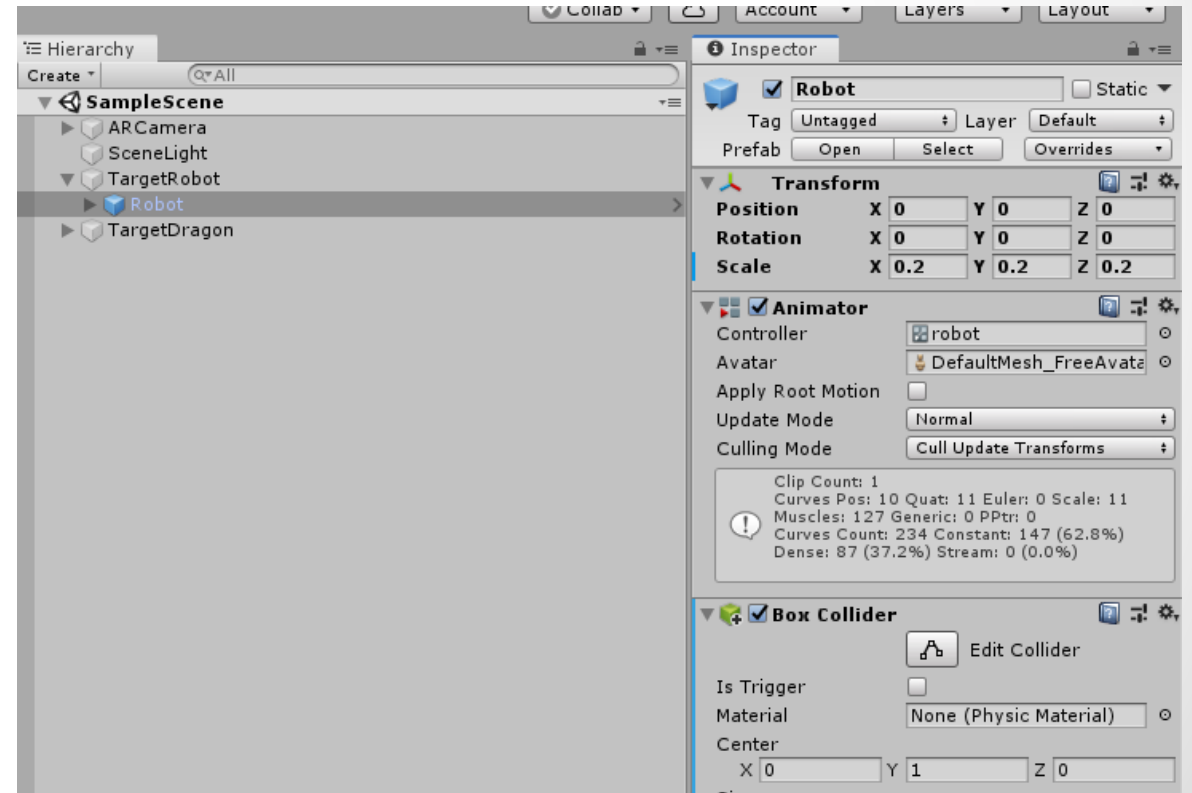
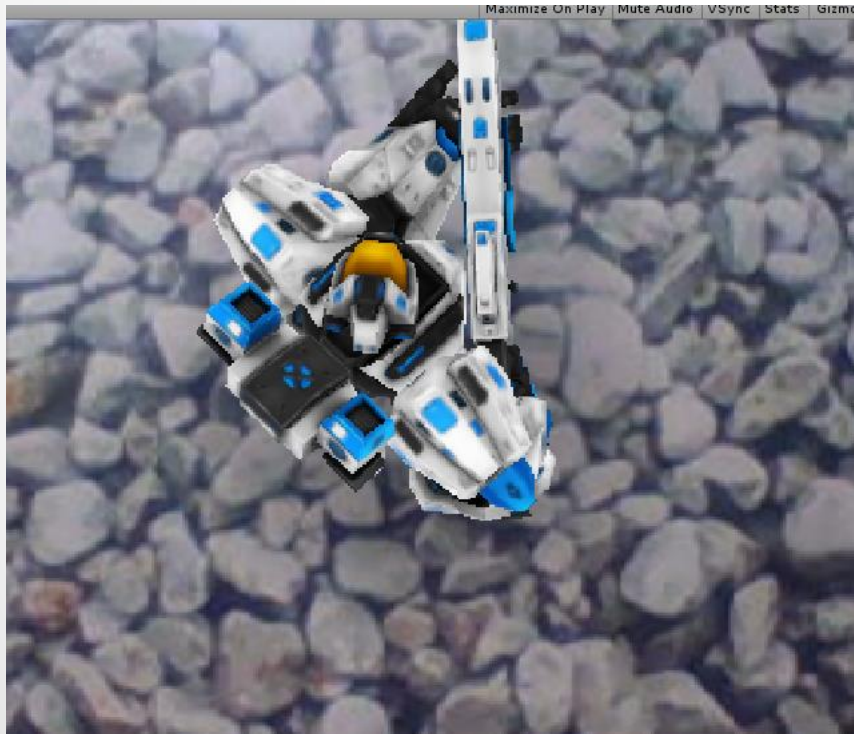
# 카드 배틀 게임 만들기

- 로봇 idle 애니메이션 추가!
  - Idle\_gun\_Middle\_ar



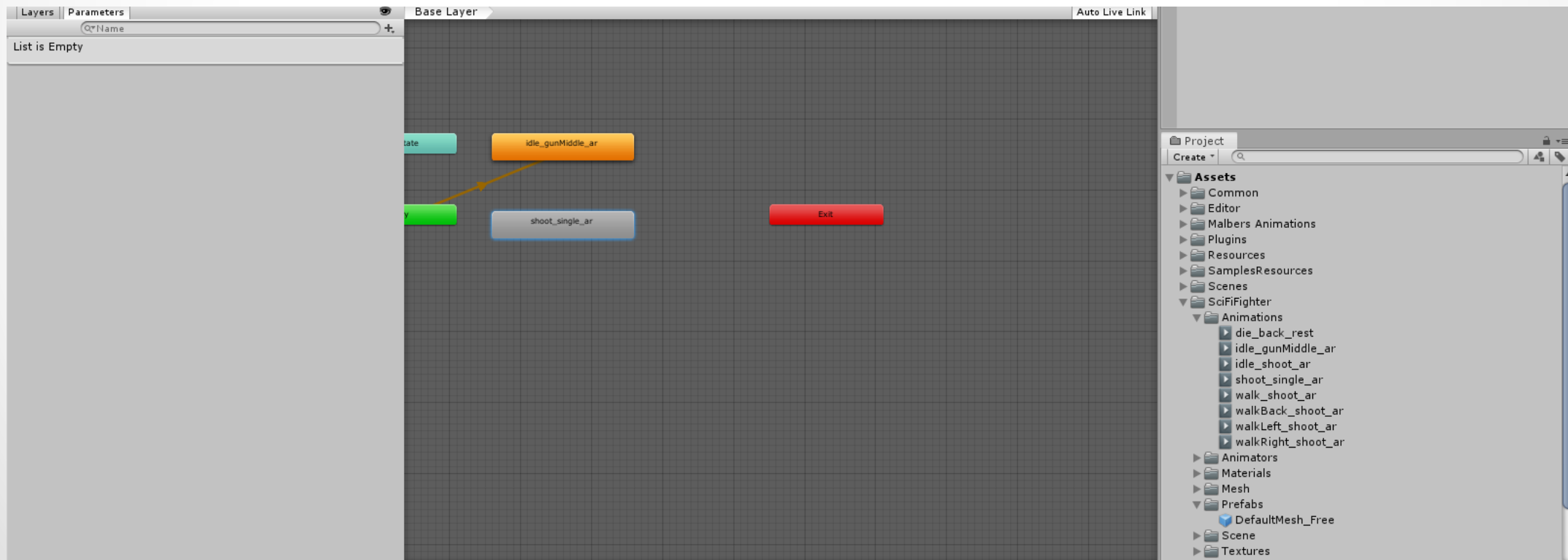
# 카드 배틀 게임 만들기

- Robot 객체의 Animator 컴포넌트의 Controller 부분을 robot 으로 교체후 테스트! – idle 애니메이션만 실행됨



# 카드 배틀 게임 만들기

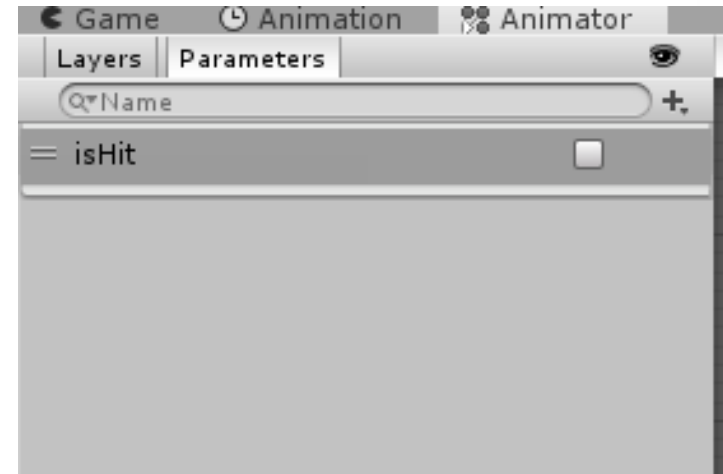
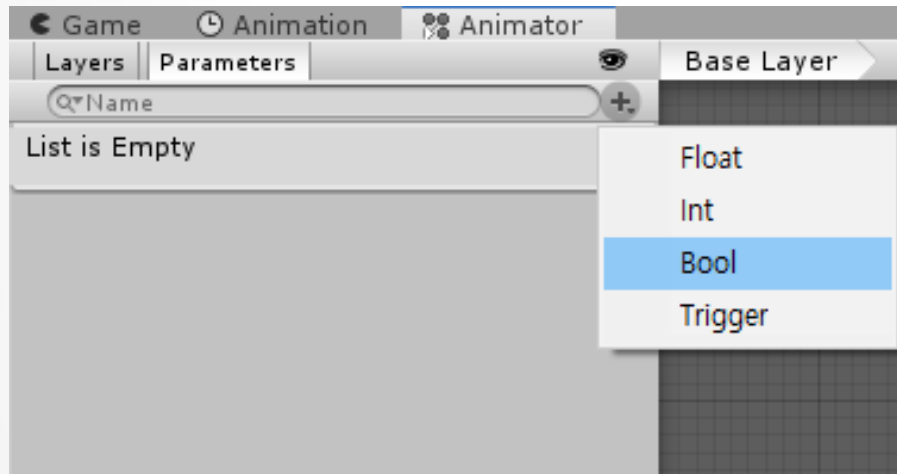
- 공격 애니메이션 추가
  - Soot\_single\_ar 애니메이션 드래그해서 추가!





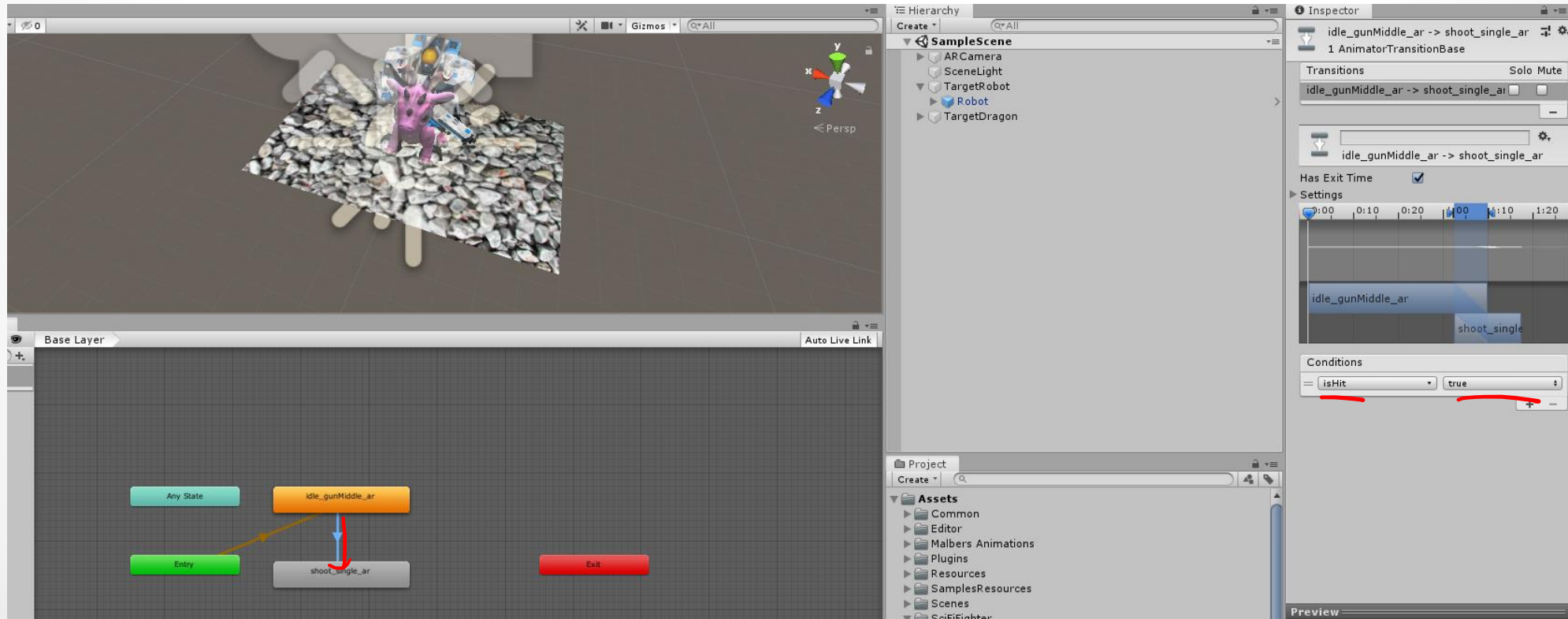
# 카드 배틀 게임 만들기

- 공격 애니메이션이 실행되기 위한 조건 실행
  - 파라미터 항목에서 Bool 타입으로 isHit 추가



# 카드 배틀 게임 만들기

- 상태 전이 그래프를 선택 (idle 에서 attack) 파란색 화살표
- Idle 에서 attac으로 변화되기 위한 조건에 isHit 가 true 를 명시



# 카

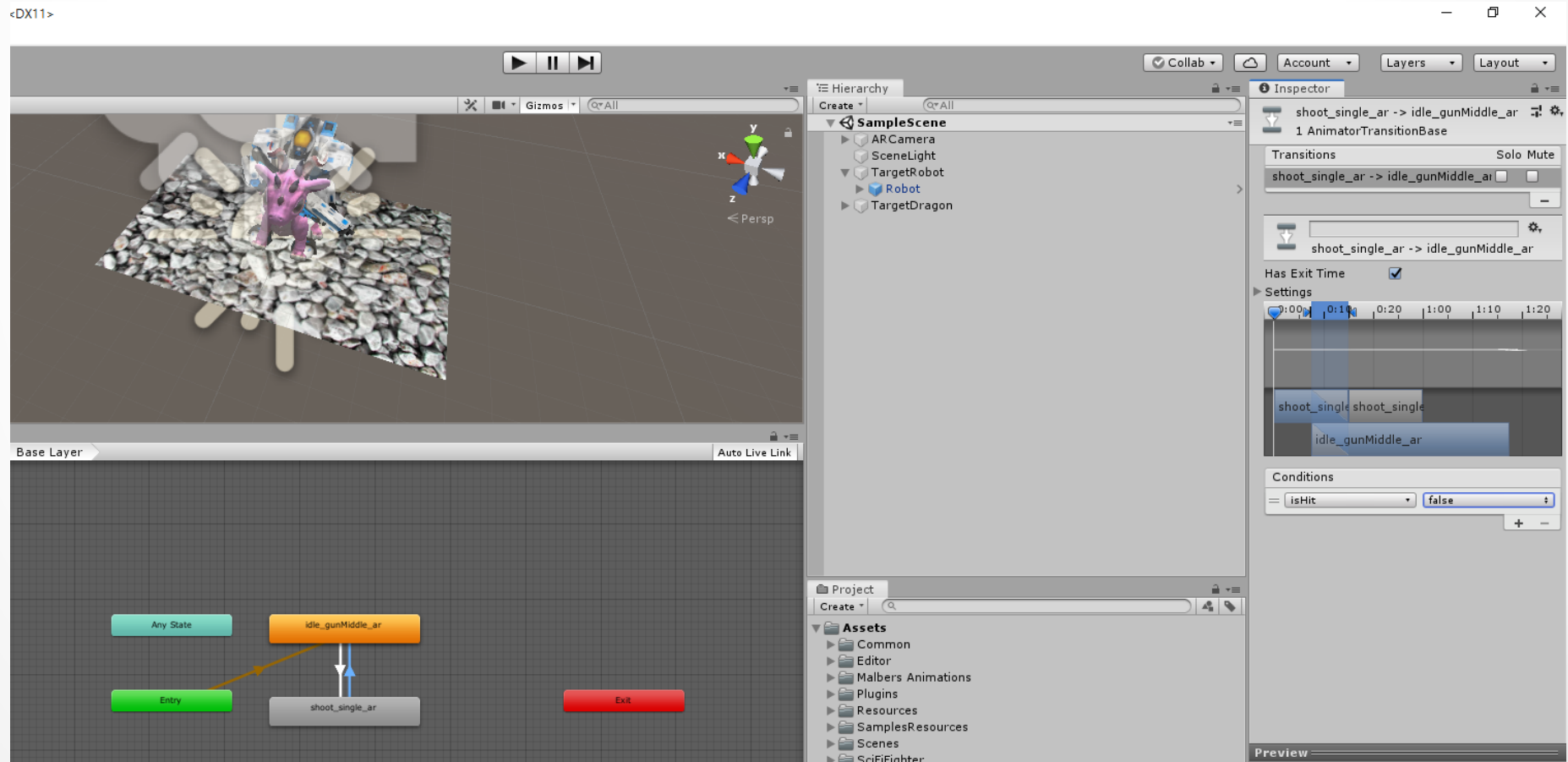
- rayCast\_robot.cs 파일 수정
- 애니메이터 부분만 추가됨

```
3 using UnityEngine;
4
5 public class rayCast_robot : MonoBehaviour
6 {
7     Animator anim; // 애니메이션을 실행시키기 위한 변수
8     void Start()
9     {
10         // 애니메이터 컴포넌트의 값을 얻어와서 저장
11         anim = transform.GetComponent<Animator>();
12     }
13     // Update is called once per frame
14     void Update()
15     {
16         //부딪히는 물체의 정보를 담는 변수 hit 선언
17         RaycastHit hit;
18         //Forward 라는 이름의 방향 변수 선언 및 대입
19         Vector3 forward = transform.TransformDirection(Vector3.forward) * 1000;
20         //가상의 레이인 레이캐스트를 테스트 중에는 보일수 있도록 함
21         Debug.DrawRay(transform.position, forward, Color.green);
22
23         if(Physics.Raycast(transform.position, forward, out hit))
24         {
25             // 레이 캐스트에 어떠한 물체가 맞는 다면 hit라는 문구를 넣음
26             Debug.Log("드래곤을 발견!");
27             // 애니메이션 실행하기
28             anim.SetBool("isHit", true);
29         }
30     }
31 }
32 }
```



# 카드 배틀 게임 만들기

- Attack 에서 다시 Idle로 돌아가는 조건도 설정
  - IsHit 가 false로 되어야 함!



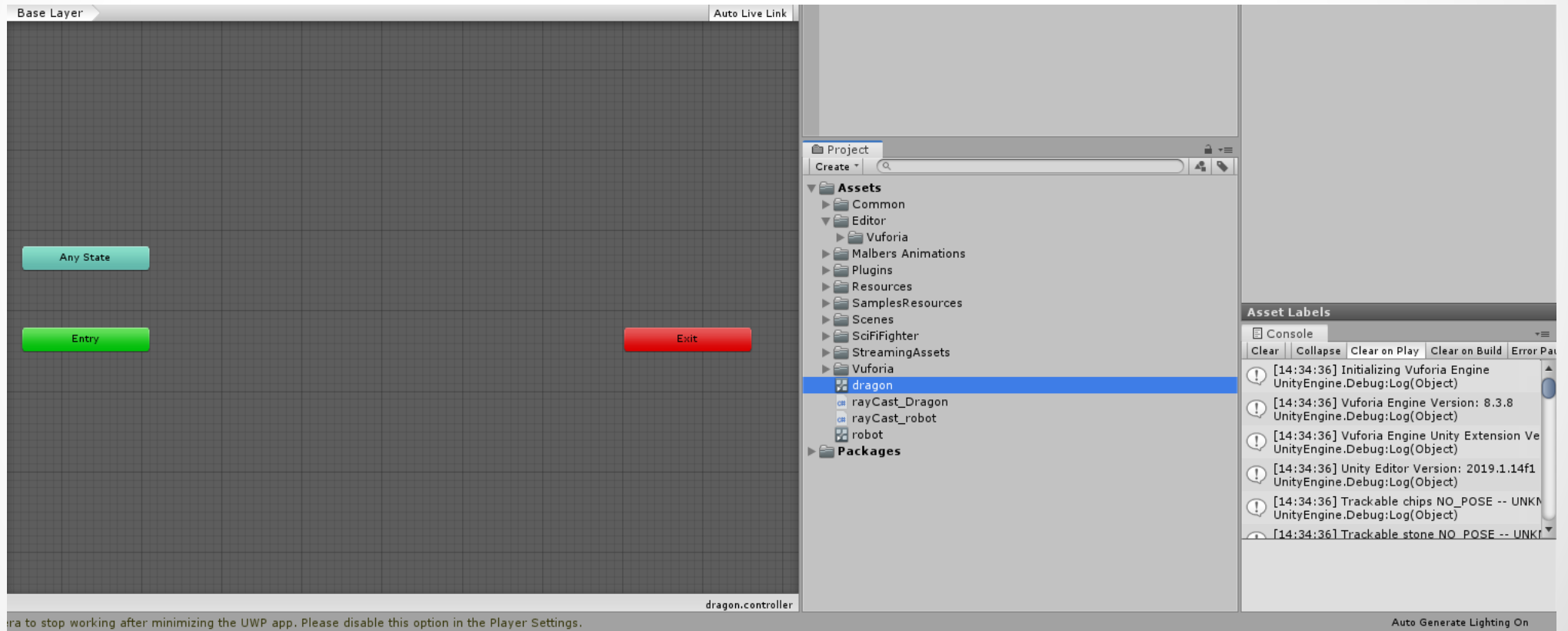
# 카드 배틀 게임 만들기

- Robot 소스 코드 수정
  - 로봇이 드래곤을 주시하는 경우만 공격 가능

```
// Update is called once per frame
void Update()
{
    //부딪히는 물체의 정보를 담는 변수 hit 선언
    RaycastHit hit;
    //Forward 라는 이름의 방향 변수 선언 및 대입
    Vector3 forward = transform.TransformDirection(Vector3.forward) * 1000;
    //가상의 레이인 레이캐스트를 테스트 중에는 보일수 있도록 함
    Debug.DrawRay(transform.position, forward, Color.green);
    if(Physics.Raycast(transform.position, forward, out hit))
    {
        // 레이 캐스트에 어떠한 물체가 맞는 다면 hit라는 문구를 넣음
        Debug.Log("드래곤을 발견!");
        // 애니메이션 실행하기
        anim.SetBool("isHit", true);
    }
    else
    {
        anim.SetBool("isHit", false);
    }
}
```

# 카드 배틀 게임 만들기

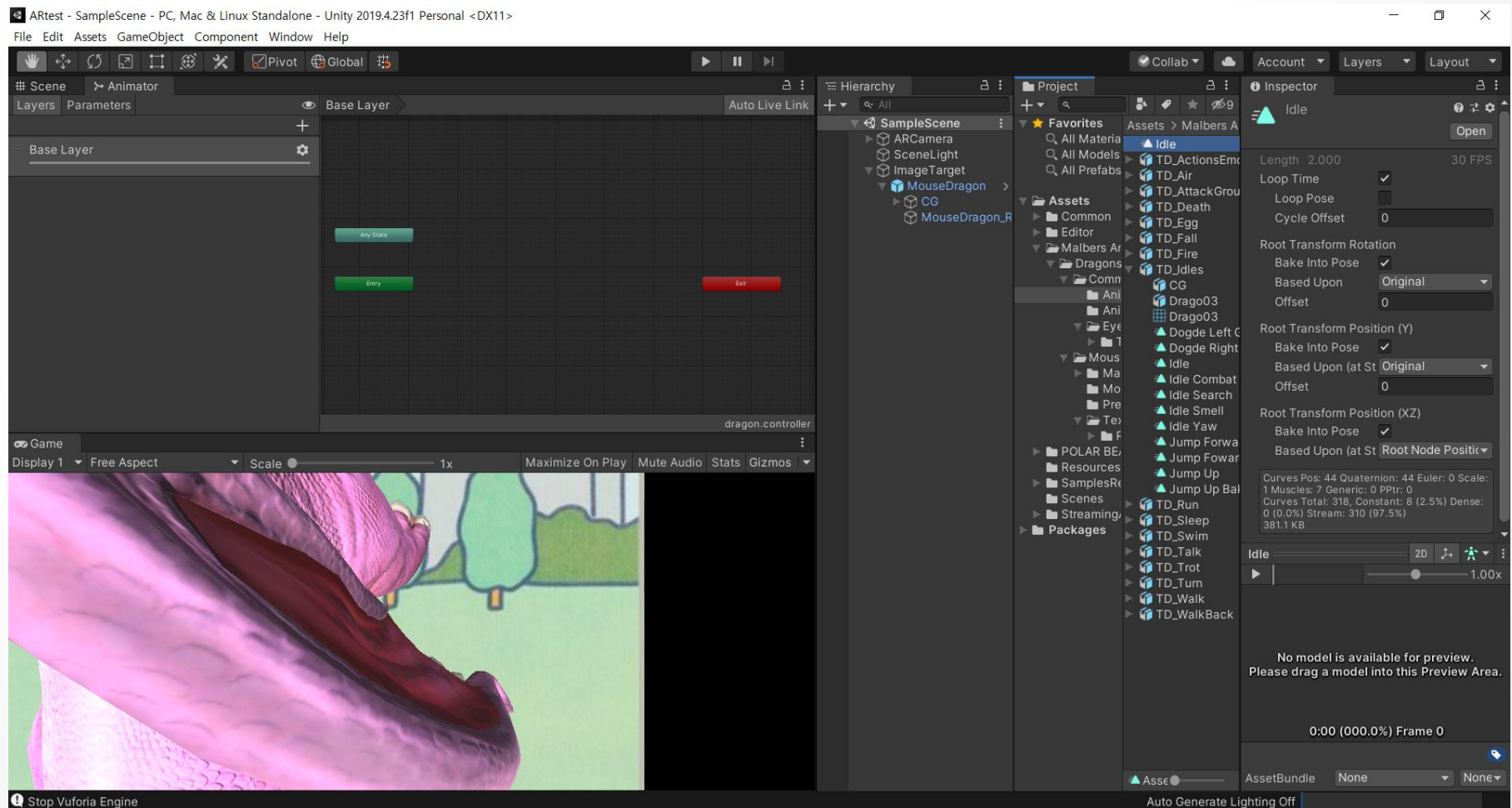
- 드래곤 애니메이션 제작 (로봇과 같음 – 애니메이션 이름만 참고)





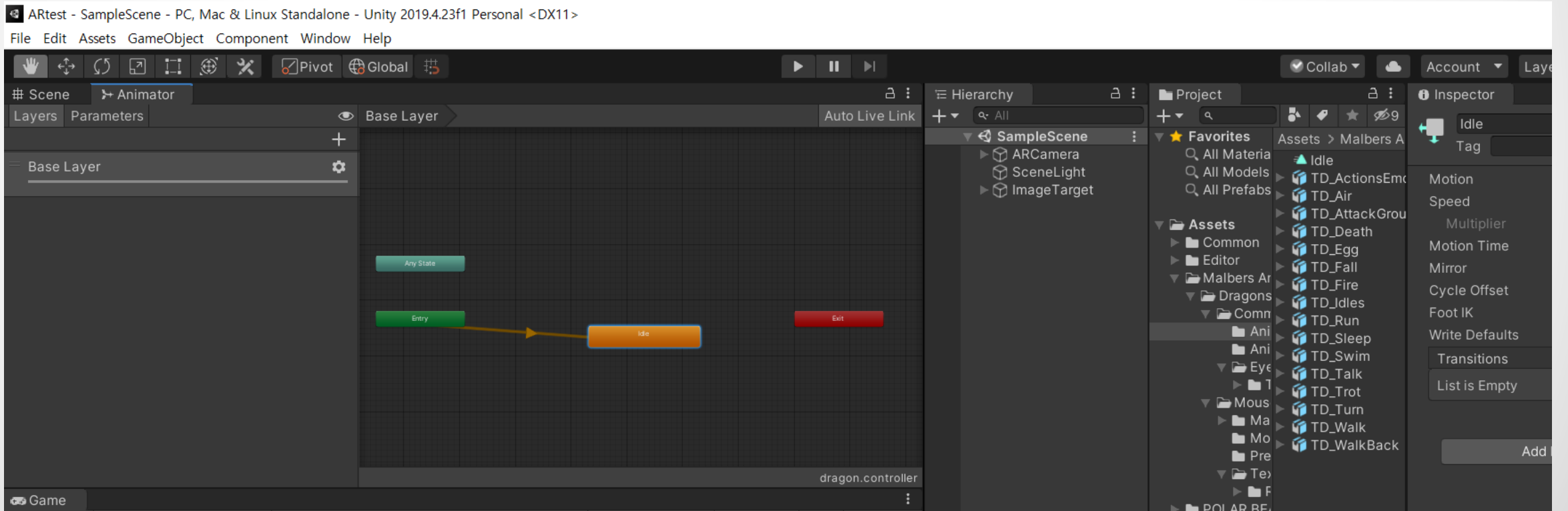
# 카드 배틀 게임 만들기

- 드래곤 애니메이션 제작 Idle 을선택한 다음 Ctrl + D 키를 누르기



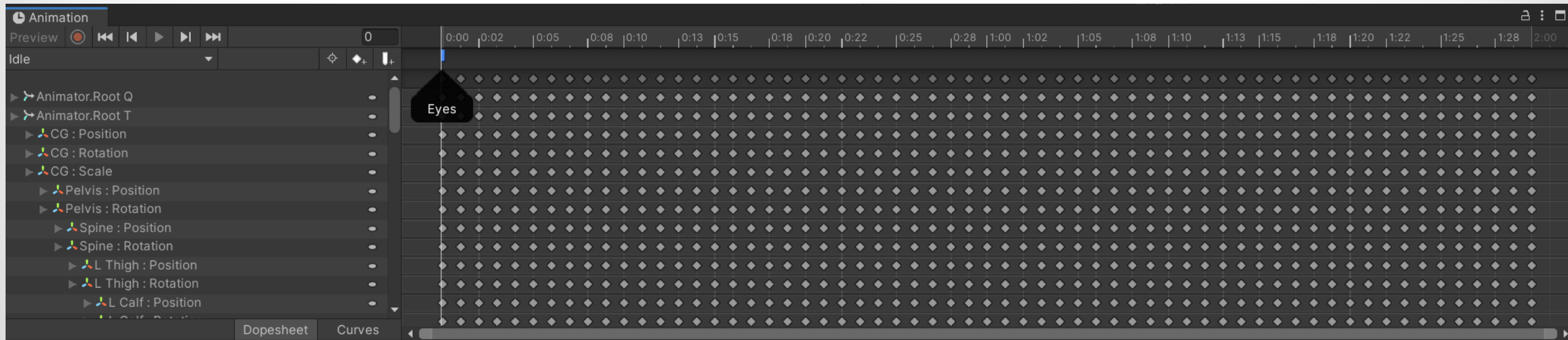
# 카드 배틀 게임 만들기

- 해당 idle 애니메이션을 dragon 애니메이터에 드래그&드랍



# 카드 배틀 게임 만들기

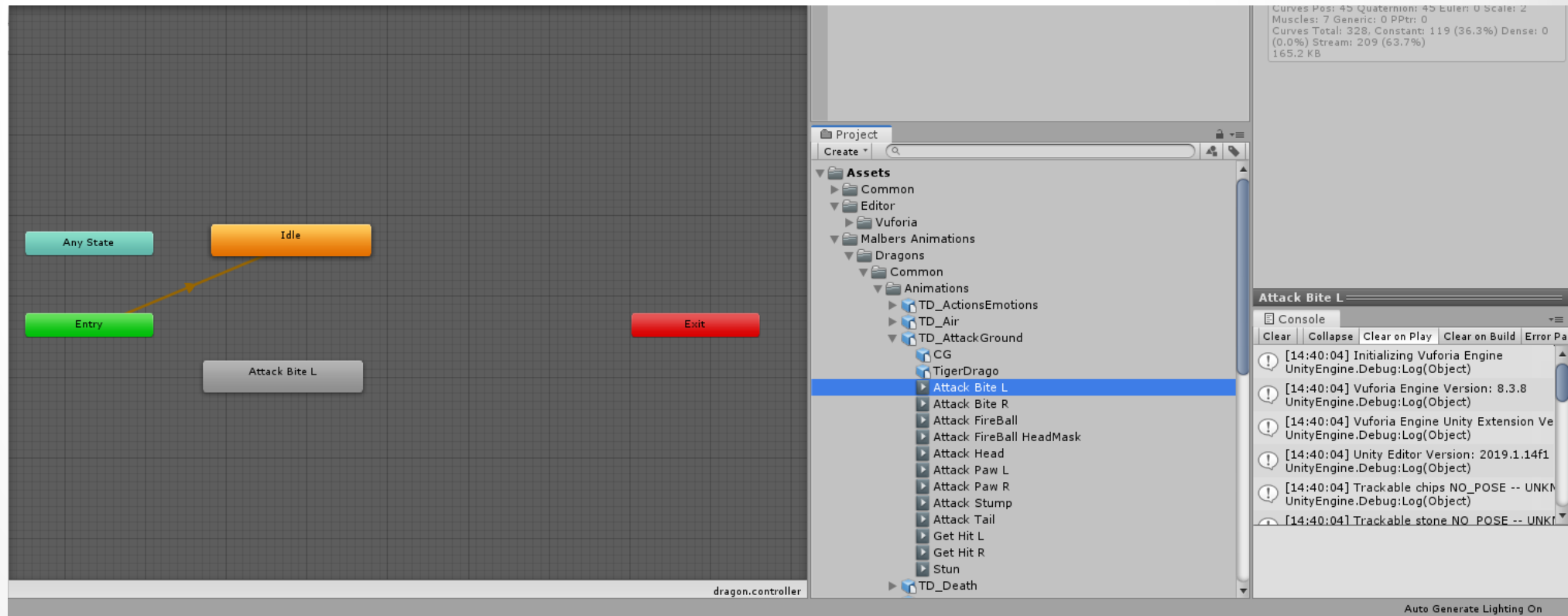
- Idle 애니메이션 더블클릭후..
- Eyes 부분 지우기





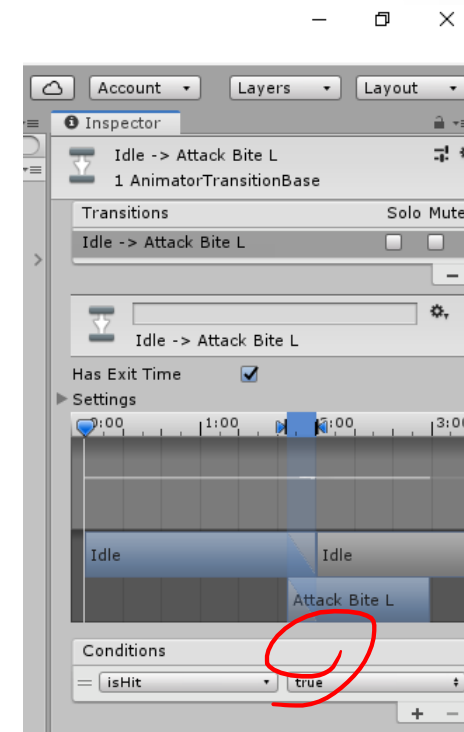
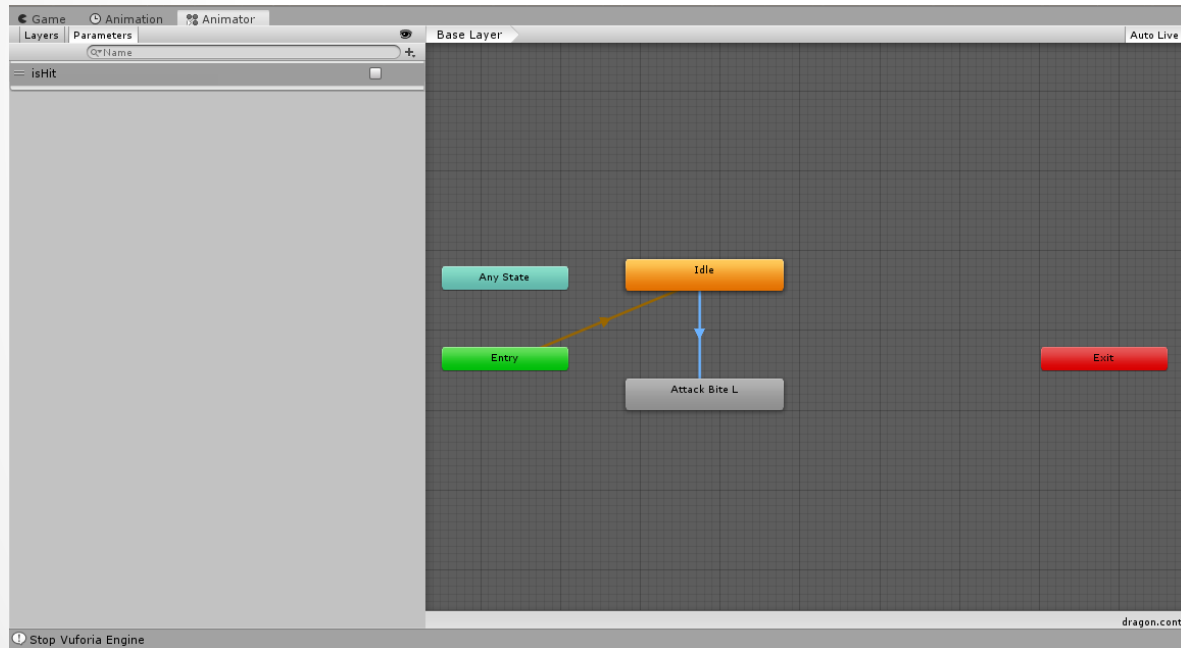
# 카드 배틀 게임 만들기

- 드래곤 애니메이션 제작 (로봇과 같음 – 애니메이션 이름만 참고)



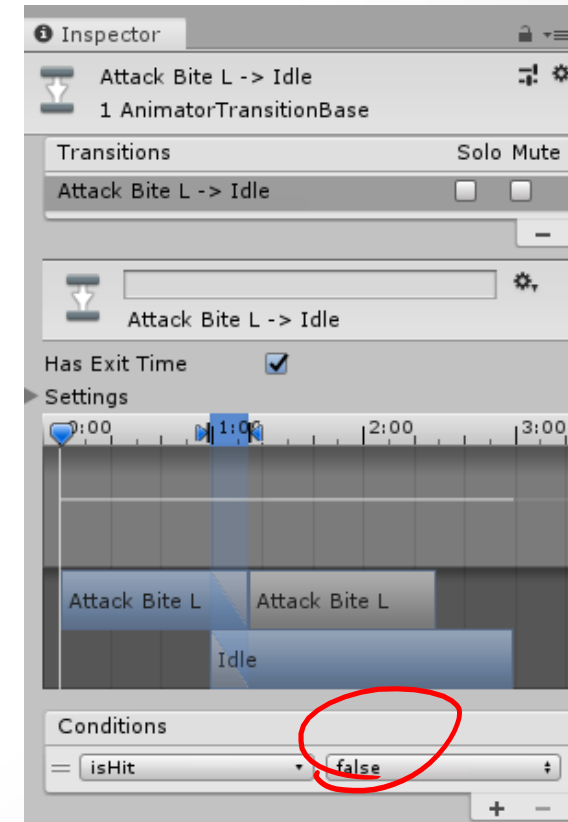
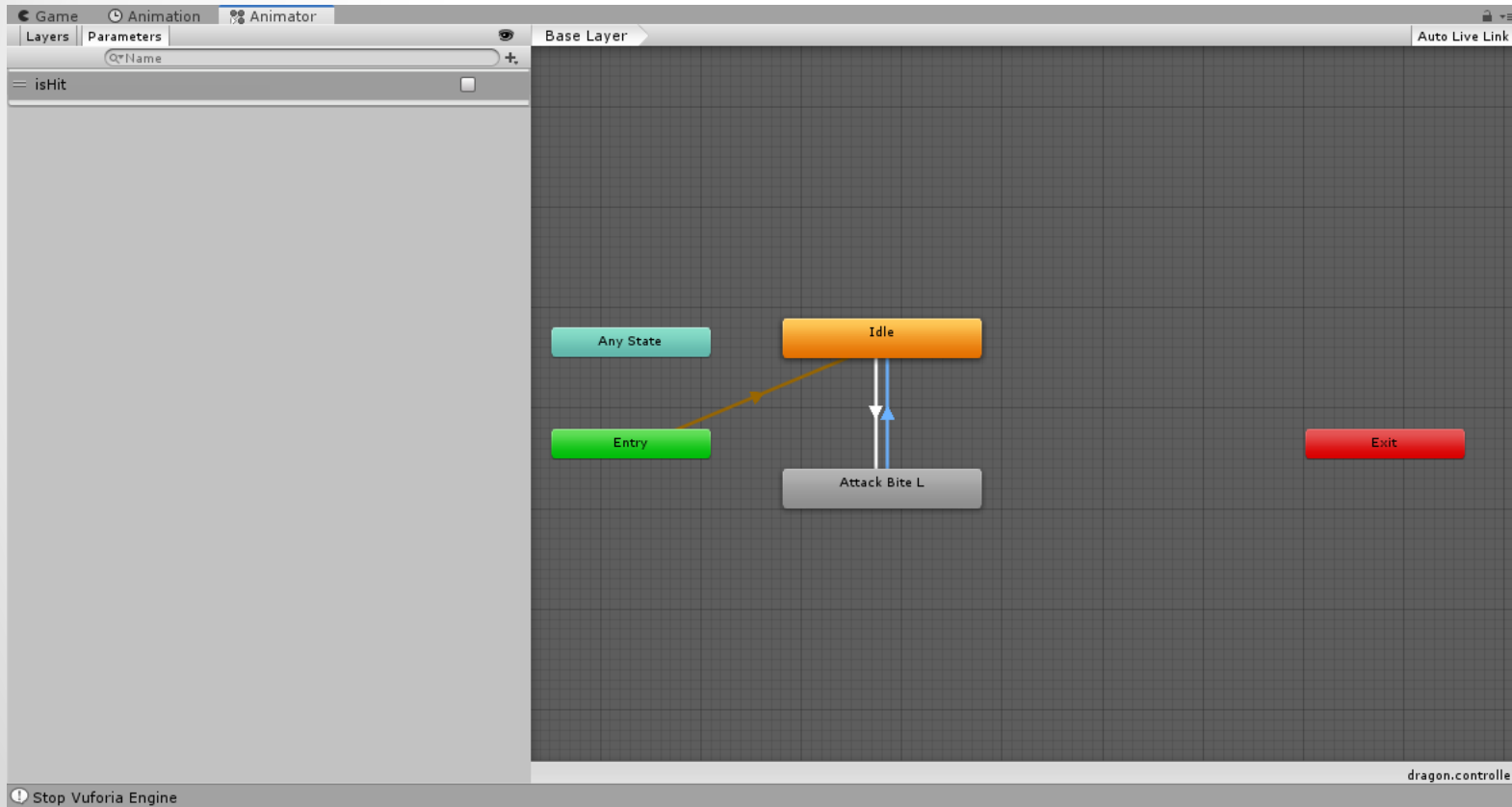
# 카드 배틀 게임 만들기

- 드래곤 애니메이션 제작 (로봇과 같음 – 애니메이션 이름만 참고)



# 카드 배틀 게임 만들기

- 드래곤 애니메이션 제작 (로봇과 같음 – 애니메이션 이름만 참고)





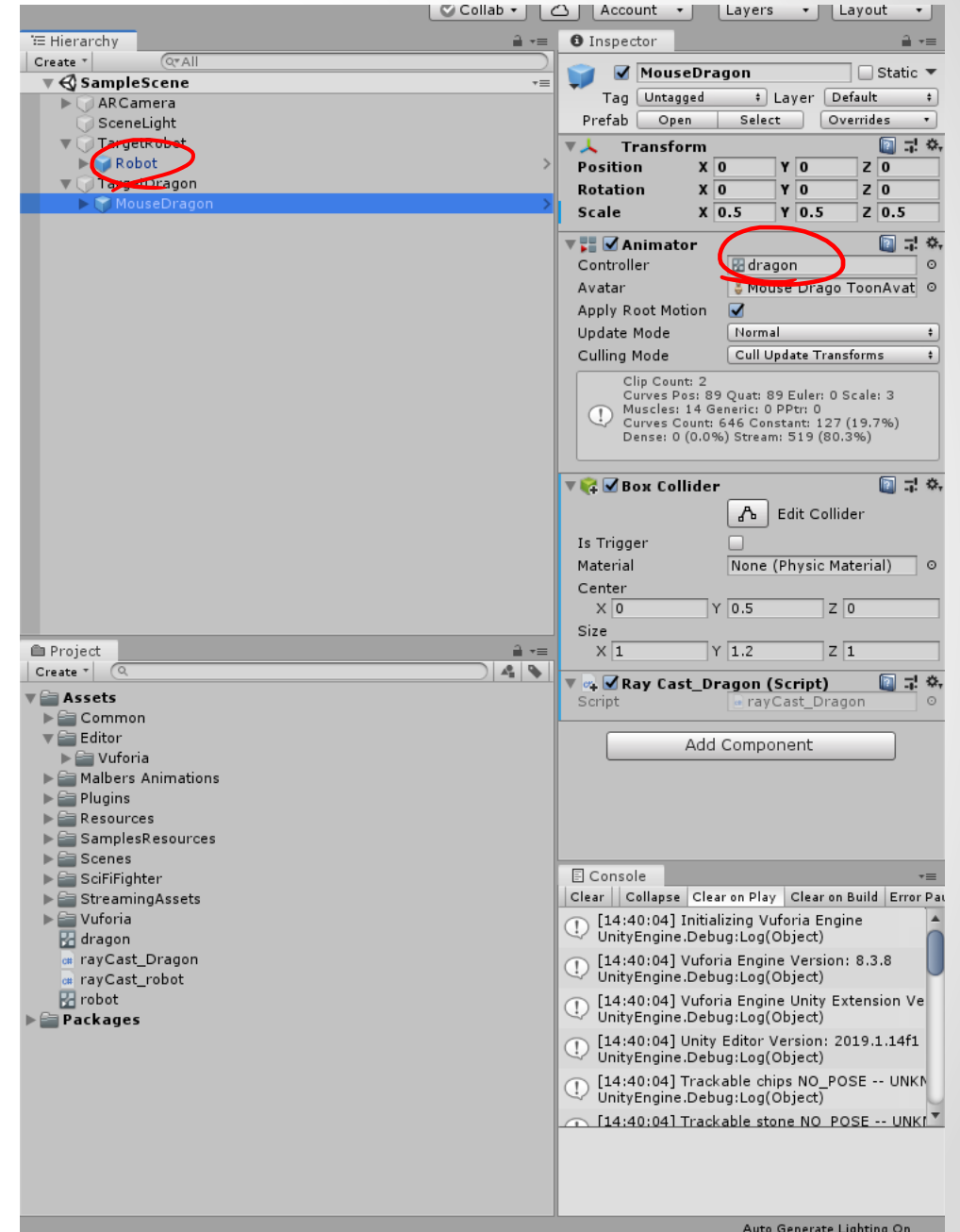
# 카드

- 코드도 동일

```
5 public class rayCast_Dragon : MonoBehaviour
6 {
7     Animator anim; // 애니메이션을 실행시키기 위한 변수
8     void Start()
9     {
10         // 애니메이터 컴포넌트의 값을 얻어와서 저장
11         anim = transform.GetComponent<Animator>();
12     }
13     // Update is called once per frame
14     void Update()
15     {
16         //부딪히는 물체의 정보를 담는 변수 hit 선언
17         RaycastHit hit;
18         //Forward 라는 이름의 방향 변수 선언 및 대입
19         Vector3 forward = transform.TransformDirection(Vector3.forward) * 1000;
20         //가상의 레이인 레이캐스트를 테스트 중에는 보일수 있도록 함
21         Debug.DrawRay(transform.position, forward, Color.green);
22         if (Physics.Raycast(transform.position, forward, out hit))
23         {
24             // 레이 캐스트에 어떠한 물체가 맞는 다면 hit라는 문구를 넣음
25             Debug.Log("로봇을발견!");
26             // 애니메이션 실행하기
27             anim.SetBool("isHit", true);
28         }
29         else
30         {
31             anim.SetBool("isHit", false);
32         }
33     }
34 }
```

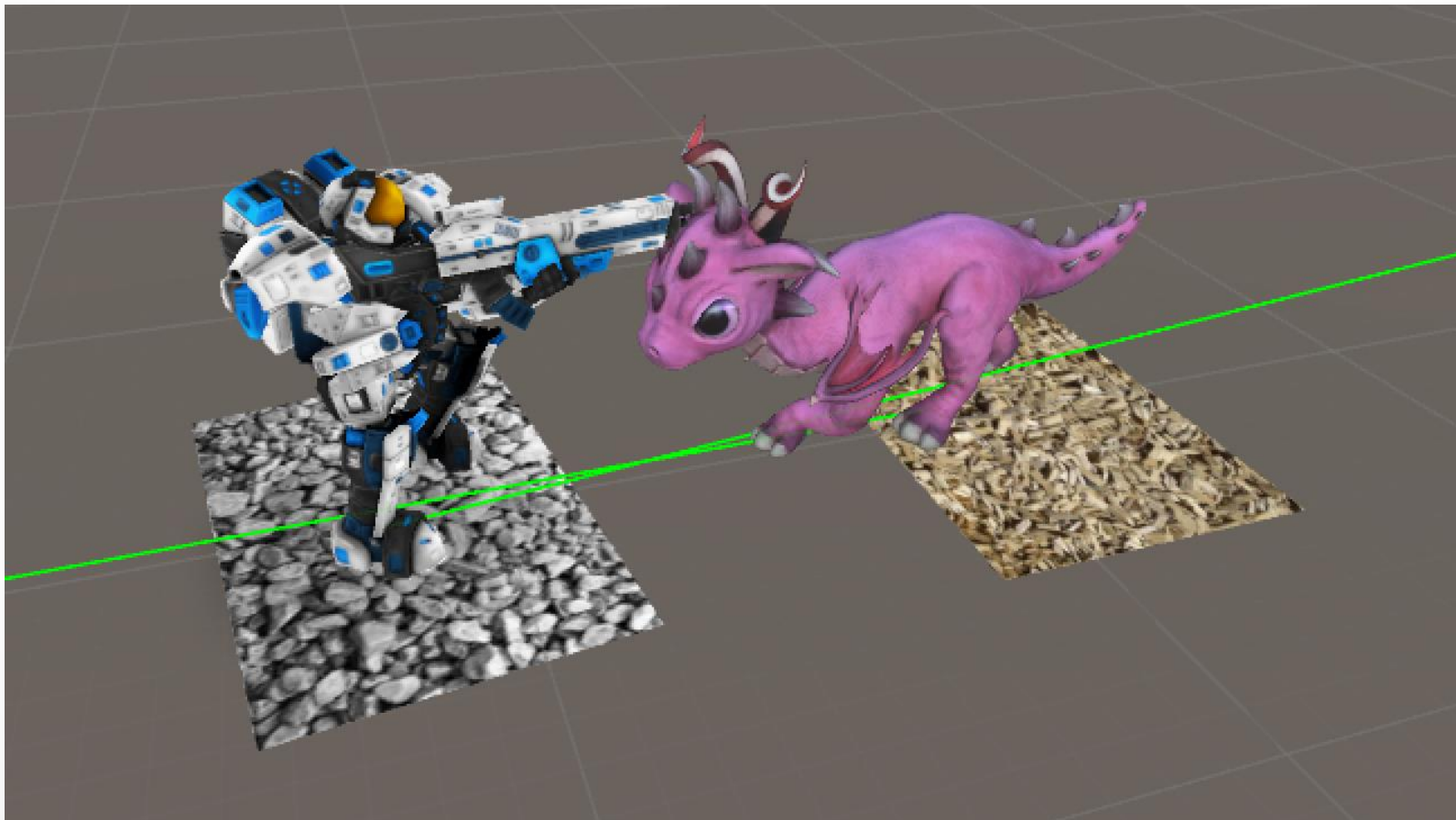
# 카드 배틀 게임

- 드래곤 애니메이션 컨트롤러 설정



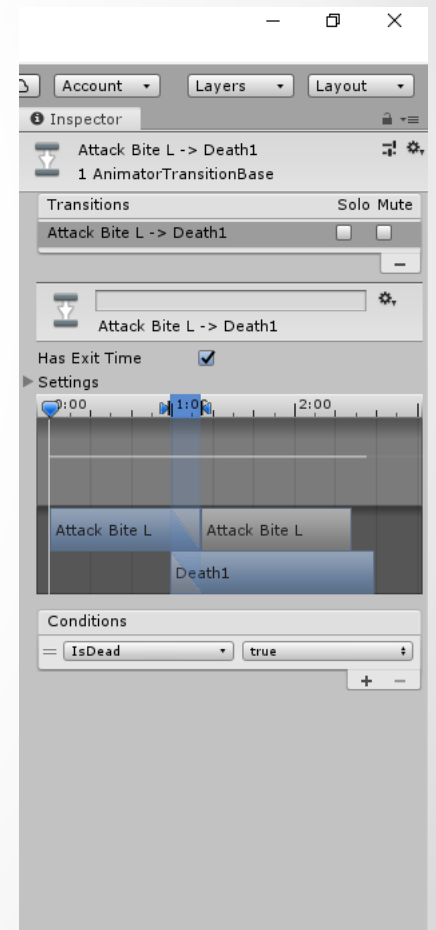
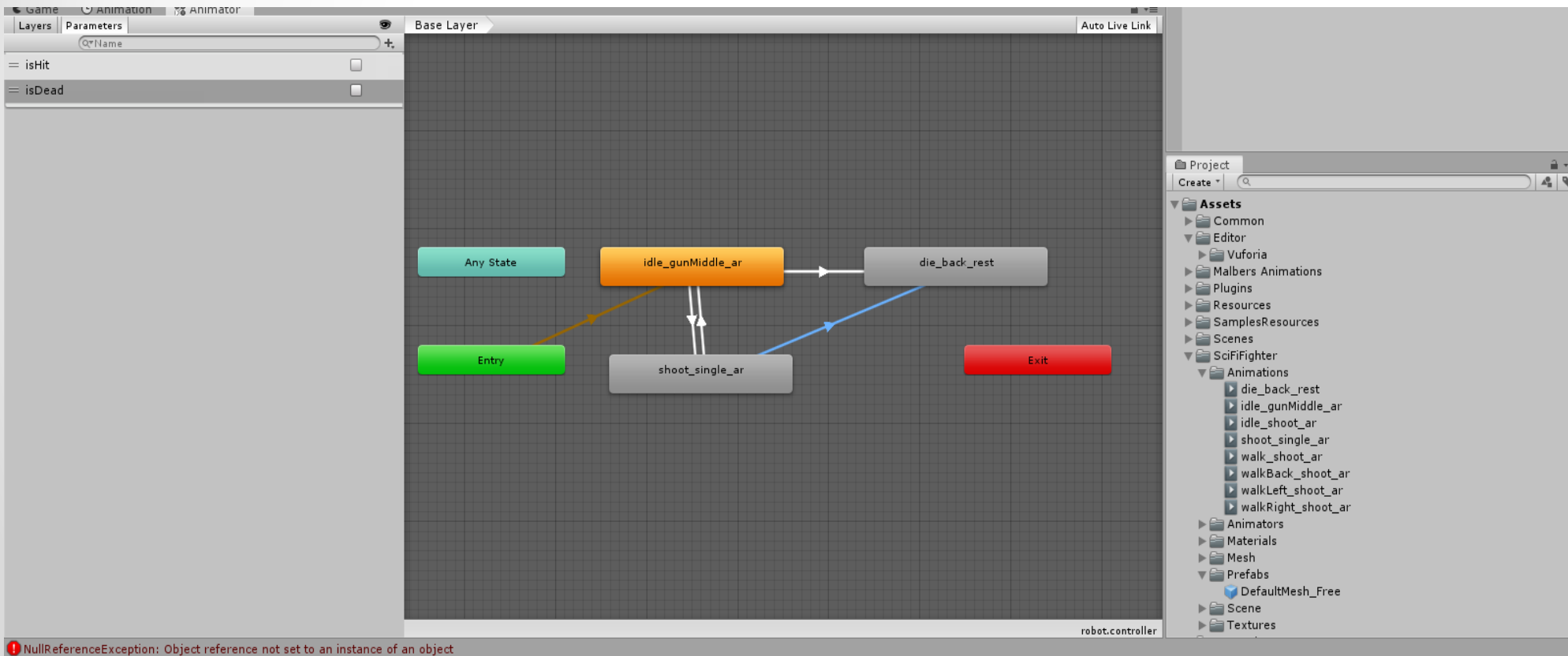
# 카드 배틀 게임 만들기

- 테스트



# 카드 배틀 게임 만들기

- 로봇의 Die 애니메이션 추가 (isDead 조건일때 발생 하기)
  - Die\_back\_rest 애니메이션 추가



NullReferenceException: Object reference not set to an instance of an object



# 카드 배틀 게임 만들기

- 기존 코드에서 다음을 추가
  - 생명력, 공격력 추가
  - 3초마다 공격하기
  - 생명력이 0이하면 죽는 애니메이션 실행

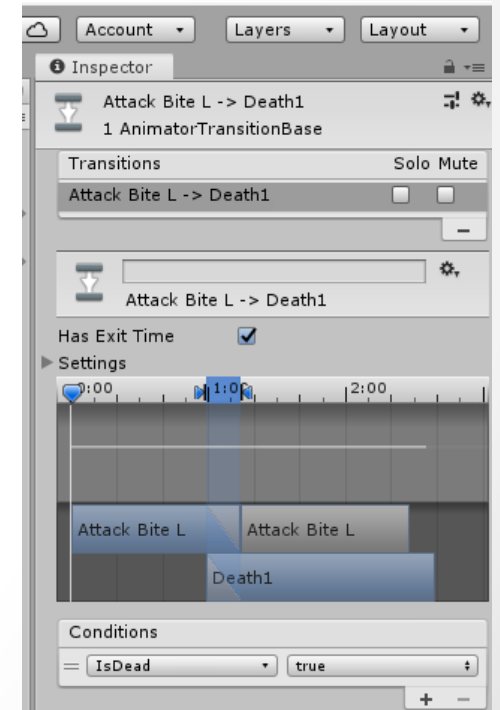
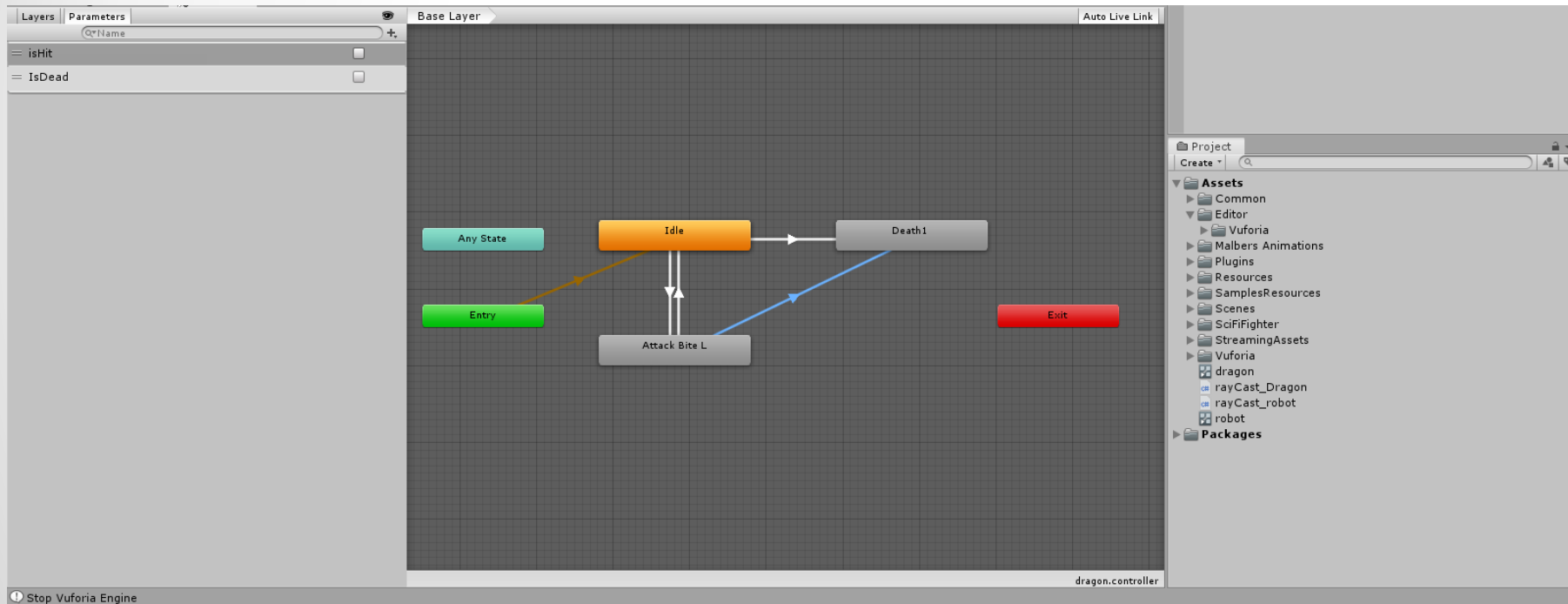
```
public class rayCast_robot : MonoBehaviour
{
    Animator anim; // 애니메이션을 실행시키기 위한 변수
    public int atkPnt = 200; // 공격력
    public int hltPnt = 5000; // 체력
    float timeElapsed; // 시간 계산
```

```
void Update()
{
    //부딪히는 물체의 정보를 담는 변수 hit 선언
    RaycastHit hit;
    //Forward 라는 이름의 방향 변수 선언 및 대입
    Vector3 forward = transform.TransformDirection(Vector3.forward) * 10;
    //가상의 레이인 레이캐스트를 테스트 중에는 보일수 있도록 함
    Debug.DrawRay(transform.position, forward, Color.green);
    if(Physics.Raycast(transform.position, forward, out hit))
    {
        timeElapsed += Time.deltaTime;
        Debug.Log("드래곤을 발견!");
        anim.SetBool("isHit", true);
        if(timeElapsed >= 3)
        {
            hit.transform.GetComponent<rayCast_Dragon>().hltPnt -= atkPnt;
            timeElapsed = 0f;
        }
    }
    else
    {
        anim.SetBool("isHit", false);
    }

    if(hltPnt <= 0)
    {
        anim.SetBool("isDead", true);
    }
}
```

# 카드 배틀 게임 만들기

- 드래곤의 Die 애니메이션 추가 (isDead 조건일때 발생 하기)
  - Death1 애니메이션 추가



# 카드 배틀 거

- 기존 코드에서 다음을 추가
  - 생명력, 공격력 추가
  - 3초마다 공격하기
  - 생명력이 0이하면 죽는 애니메이션 실행

```
public class rayCast_Dragon : MonoBehaviour
{
    Animator anim; // 애니메이션을 실행시키기 위한 변수
    public int atkPnt = 200; // 공격력
    public int hltPnt = 5000; // 체력
    float timeElapsed; // 시간 계산
```

```
void Update()
{
    //부딪히는 물체의 정보를 담는 변수 hit 선언
    RaycastHit hit;
    //Forward 라는 이름의 방향 변수 선언 및 대입
    Vector3 forward = transform.TransformDirection(Vector3.forward);
    //가상의 레이인 레이캐스트를 테스트 중에는 보일수 있도록 함
    Debug.DrawRay(transform.position, forward, Color.green);
    if (Physics.Raycast(transform.position, forward, out hit))
    {
        // 레이 캐스트에 어떠한 물체가 맞는 다면 hit라는 문구를
        Debug.Log("로봇을발견!");
        anim.SetBool("isHit", true);
        timeElapsed += Time.deltaTime;
        if (timeElapsed >= 3)
        {
            hit.transform.GetComponent<rayCast_robot>().hitPnt -= atkPnt;
            timeElapsed = 0f;
        }
    }
    else
    {
        anim.SetBool("isHit", false);
    }
    if (hltPnt <= 0)
    {
        anim.SetBool("isDead", true);
    }
}
```

# 카드 배틀 게임 만들기

- 테스트

