

게임 엔진을 활용한 모바일 어플리케이션

서보경 · 황소영

부산가톨릭대학교

Mobile Application Using Game Engine

Bokyoung Seo · Soyoung Hwang

Catholic University of Pusan

E-mail : soyoung@cup.ac.kr

요 약

모바일 어플리케이션의 주요 분야로 게임을 들 수 있다. 본 논문에서는 게임 엔진을 활용하여 횡스크롤 방식의 모바일 게임 어플리케이션의 설계를 제안한다. 제안한 게임 모델을 유니티 엔진을 활용하여 구현하도록 하였다.

키워드

game engine, scroll game, unity, mobile application

1. 서 론

대부분의 사람들이 스마트폰을 사용하는 현대 사회에서 다양한 종류의 모바일 어플리케이션이 이용자들의 요구에 맞춰 제작되고 있다. 그 중 빼놓을 수 없는 분야가 오락 기능을 담당하는 모바일 게임이다.

게임개발사에서는 회사 자체적으로 개발한 엔진을 사용하기도 하지만 보편적으로 유니티 엔진을 많이 사용한다. 이용 가격이 저렴하고, PC, 모바일 기기, 콘솔 등 다양한 플랫폼으로 게임을 만들 수 있다는 장점이 있기 때문이다. 또한 셰이더, 물리엔진, 네트워크, 오디오, 애니메이션 등 게임을 만드는데 필요한 기능들을 지원해 게임 개발의 진입 장벽을 낮춰 초보 개발자도 쉽게 접근 할 수 있다.

본 논문에서는 멀티플랫폼으로 개발이 가능한 유니티 엔진에서 C#, Javascript 언어를 이용해 안드로이드 기반의 2D 횡스크롤 러닝 게임의 설계를 제안, 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 게임의 기획과 설계를 설명하고 3장에서는 게임의 제작에 대해 다룬다. 마지막 4장으로 논문의 결론을 맺는다.

II. 모바일 러닝 게임 기획 및 설계

(1) 게임 기획

본 논문에서 제안한 횡스크롤 러닝 게임은 '정글 토끼!' 라는 이름으로, 정글에 떨어진 토끼가 마주 오는 적들을 공격하고, 장애물을 피하

코인을 얻어 점수를 올리는 방식의 게임이다.

게임에 사용되는 2d 이미지들은 Adobe사의 포토샵cs6를 이용해 제작하고, 만들어진 스프라이트들을 유니티로 불러와 안드로이드폰에서 구동할 수 있도록 게임을 구현한다.

(2) 게임 설계

'정글토끼!'의 게임 진행 방식은 그림 1과 같다.

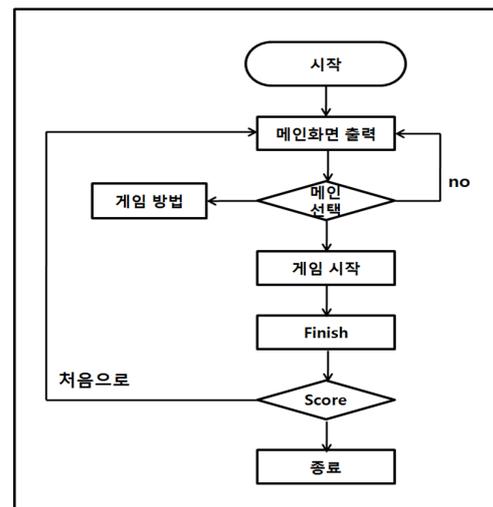


그림 1. 게임 진행 방식

게임이 시작되면 플레이어는 장애물들을 피하기 위해 점프버튼을 터치해 점프한다. 장애물이

아닌 적들은 공격버튼으로 처치할 수 있다. 흘러가는 시간과 플레이어가 얻은 코인을 더해 점수를 계산한다. 장애물이나 적에게 부딪히면 플레이어 캐릭터가 죽어 게임이 종료된다.

(3) 이미지 제작

본 논문의 ‘정글 토끼!’ 게임에서 사용된 2d 이미지들은 포토샵cs6으로 제작되었다. 제작된 이미지들은 그림 2와 같다.

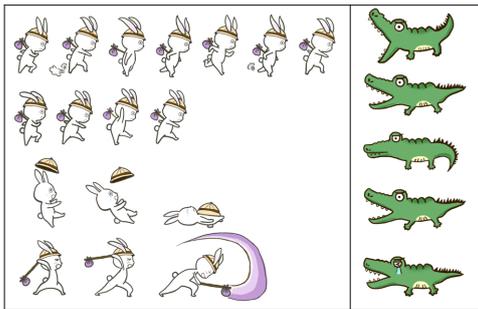


그림 2.1. 캐릭터 2d sprites

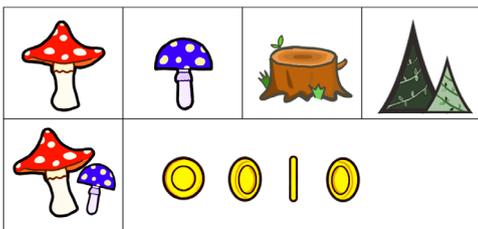


그림 2.2. 오브젝트 2d sprites

애니메이션이 필요한 캐릭터는 그 동작에 맞는 여러 개의 프레임으로 그리고 유니티에서 제공하는 애니메이터와 애니메이션 기능을 사용해 화면에 나타내고 Trigger, boolean값으로 동작을 전환한다.

III. 모바일 러닝 게임 제작

유니티를 설치하면 기본적으로 MonoDevelop이 함께 설치되는데, 이는 유니티가 제공하는 통합 개발 환경(IDE)이다. IDE는 잘 알려진 텍스트 에디터의 기능, 디버깅 및 기타 프로젝트 관리 작업 등의 추가 요소를 조합하여 사용할 수 있다. 유니티에서 설정을 바꾸지 않는 한 코딩을 하려면 자동적으로 MonoDevelop으로 연결이 된다. 이 게임은 MonoDevelop환경에서 C# 언어로 코딩한다.

이 장에서는 본 논문에서 제안한 게임의 주요 기능인 캐릭터의 애니메이션과 움직임, 오브젝트의 충돌에 대해 다룬다.

(1) 캐릭터 애니메이션

유니티에서는 앞서 제작한 2d 스프라이트로 애니메이션을 만들 수 있다. 움직이는 과정을 하나

하나 그려서 만든 이미지 파일을 유니티로 불러와 sprite editor를 이용해 동작별로 하나씩 잘라주고, 잘려진 이미지들을 애니메이션 뷰에서 이어 붙여 애니메이션 파일을 만든다. 이렇게 만들어진 애니메이션들은 플레이어의 상태에 맞춰 전환된다. 먼저 어떤 상태와 애니메이션 파일을 연결한 다음 상태를 전환해서 재생할 애니메이션을 바꾸는 것이다. 이 설정은 Animator Controller로 관리한다.

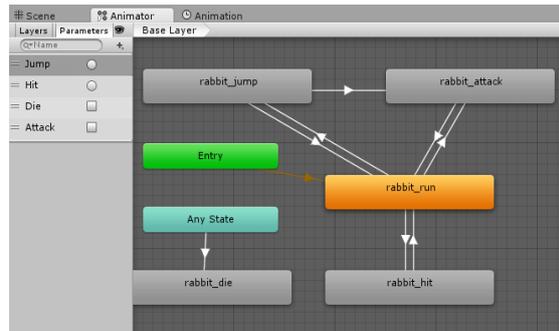


그림 3. Animator Controller

그림 3에서 보는 바와 같이 애니메이터가 설정이 되면 스크립트를 통해 플레이어의 상태를 전환한다. 오브젝트의 컴포넌트를 가져오는 기능을 하는 GetComponent<Type>(); 함수로 플레이어의 Animator를 받아온다. 특별한 이벤트가 발생해서 플레이어의 상태가 바뀌면 애니메이션을 전환한다. 그림 4와 같은 방식으로 캐릭터에 움직임, 애니메이션을 넣어준다.

```
private Rigidbody2D myRigidbody;
private Animator anim;

void Start(){
    myRigidbody = gameObject.GetComponent<Rigidbody2D> ();
    anim = gameObject.GetComponent<Animator> ();
}

void Update(){
    if (Input.GetButtonDown ("Jump")){
        myRigidbody.velocity = new Vector2 (myRigidbody.
            velocity.x, jumpPower);
        anim.SetTrigger ("Jump");
    }
}
```

그림 4. 애니메이션 코드

(2) 오브젝트 충돌

플레이어가 장애물을 피하지 못하면 죽고, 코인을 먹으면 점수를 올리기 위해서 플레이어와 기타 오브젝트들 간의 충돌 체크를 해 주어야 한다. 두 물체가 충돌하려면 기본적으로 양쪽 다 콜라이더(collider)가 필요하고 둘 중 하나는 리지드 바디가 있는 상태여야 한다. 이 조건이 만족한다면 충돌을 체크하는 함수로 OnTriggerEnter2D와 OnCollisionEnter2D를 사용한다. 전자의 경우 is Trigger 속성이 체크되어 있어 두 물체가 통과가

가능하고 그에 대한 충돌 처리를 하고, 후자의 경우 물체가 서로 통과하지 않고 튕겨나며 그에 대한 물리적 계산이 필요한 충돌을 판정한다.

```

void OnTriggerEnter2D(Collider2D coll) {
    if(coll.gameObject.tag == "Coin") {
        coinScore += 1;
        Destroy(coll.gameObject);
    }
}

void OnCollisionEnter2D(Collision2D collision) {
    if(collision.collider.gameObject.tag == "Obstacle") {
        anim.SetTrigger("Hit");
        Destroy(collision.gameObject);
        heart--;
    }
}
    
```

그림 5. 충돌 판정 코드

그림 5와 같이 오브젝트에 태그를 주어 플레이어와 장애물, 적 캐릭터의 충돌 시 플레이어의 애니메이션 상태를 hit로 바꿔 플레이어가 맞는 모션을 취하고, 하트를 깎이게 한다. 플레이어와 코인이 충돌하게 되면 점수를 1 올리고 코인은 사라지게 구현하였다.



그림 6. 게임 화면

이와 같은 방식으로 게임에 필요한 기능들을 구현하였고, 추가적으로 게임을 즐길 수 있는 요소들을 프로그래밍하였다. 실제 그림 6과 같이 게임이 진행된다.

IV. 결 론

본 논문에서는 안드로이드 기반의 모바일 게임인 ‘정글 토끼!’를 설계하고 유니티 엔진에서 C#언어를 이용해 구현하였다. 애니메이션, 물리 충돌 등 비교적 간단한 기능을 가지고 단순하지만 중독성 있는 게임을 만들어봄으로써 유니티 엔진을 사용하면 전문적인 지식을 가진 전공자가 아니더라도 만들 수 있는 모바일 게임을 제시하였다.

향후에는 본 논문에서 제작한 모바일 게임을 더 보완하고, 이 게임을 토대로 2d뿐만 아니라 3d, 더 나아가 VR게임까지 개발하고자 한다.

참고문헌

- [1] <http://unity3d.com/kr>
- [2] 마쓰다쓰 외 6인, ‘유니티5 3D 게임 제작 입문’, 길벗, p149-p156, 2015
- [3] <http://xischrist.blog.me/220139104915>