

멀티 플랫폼에서의 2D 횡스크롤 캐주얼 게임 융합 설계

양동혁¹, 김수균^{1*}

¹배재대학교 게임공학과

The Design of 2D Side-Scrolling Casual Games in a Multi-platform Environment

Dong-Hyuk Yang¹, Soo-Kyun Kim^{1*}

¹Department of Game Engineering, Paichai University

요약 캐주얼 게임은 조작 난이도와 방식이 쉬워, 자투리 시간을 이용하여 누구나 즐길 수 있는 게임이다. 특히 캐릭터 게임 장르의 경우에는 귀엽고 다양한 캐릭터들이 나와 이목을 이끄는 특징을 가지고 있다. 이러한 캐릭터 게임의 경우에는 현재 PC용 게임 이외에도 많은 사용자를 가지고 있는 스마트폰으로도 즐길 수 있다.

본 논문에서는 유니티 3D 엔진을 이용하여 스마트 폰에서 조작할 수 있는 2D 횡스크롤 캐주얼 슈팅 게임을 설계하는 것을 목적으로 한다. 유니티 3D 엔진을 사용하는 이유는 적은 비용으로 멀티 플랫폼 지원을 해주기 때문이며, 특히 하나의 프로그램 소스코드를 이용하여 다양한 플랫폼에서 구동할 수 있게 해준다. 본 논문의 특징으로는 기존 슈팅게임의 딱딱함과는 달리 10~20대 전체를 대상으로 하여 귀엽고, 아기자기한 컨셉과 손쉬운 조작 방법을 도입하여 경쾌한 게임을 설계하는 것이다.

• **Key Words** : 스프라이트, 2D 횡스크롤 게임, 유니티 3D 엔진, 캐주얼 게임, 멀티 플랫폼

Abstract Casual games are easy in terms of control and mode, and thus everyone can enjoy them in their free time. Character games, in particular, feature a variety of cute and attractive characters. Such character games are now available for smartphones, which are used by a large number of users, as well as for PCs.

This paper is aimed at demonstrating the design of 2D side-scrolling casual shooting games utilizing Unity 3D Engine, which can be manipulated on smartphones. Unity 3D engine was used because it supports multi-platform with low costs, and it makes it possible for games to be operated on various platforms using a single program source code. This approach is unique because the design of the games uses cute characters, attractive concepts and easy control, targeting users in their 10s or 20s, unlike existing shooting games that are serious.

• **Key Words** : Sprite, 2D Side-Scrolling Game, Unity3D Engine, Casual Game, Multi-Platform

1. 서론

국내 게임 시장은 10조 7,183억 원의 시장규모를 가지며, 2014년과 2015년 전체 게임시장 역시 온라인 게임과

모바일 게임의 성장이 이루어 질 것으로 전망된다[1]. 이러한 모바일 시장의 성장을 통해 많은 모바일용 게임이 시장에 선보이고 있으며, 개발회사 또한 모바일 게임을

*교신저자 : 김수균(kimsk@pcu.ac.kr)

접수일 2015년 6월 12일

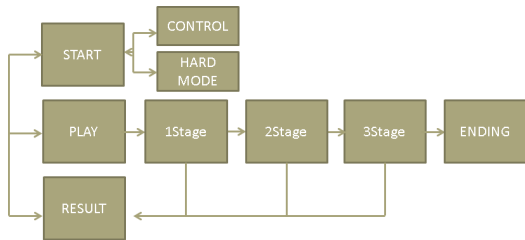
수정일 2015년 8월 13일

게재확정일 2015년 8월 20일

지속적으로 제작하고 있다. 특히 모바일 플랫폼의 장점 중 하나는 PC보다 적은 비용으로 게임을 제작 할 수 있고, 휴대성이 좋기 때문에 장소에 구애받지 않고 게임을 즐길 수 있다는 점이다.

본 논문은 모바일 플랫폼을 이용한 2D 횡스크롤 슈팅 캐주얼게임 제작을 목표로 하며, 유니티 3D 엔진[2,3,4,5]을 이용하여 손쉽게 게임을 설계하는 방법에 대해 보여준다. 본 논문에서는 캐릭터 디자인부터 게임 구현에 필요한 횡 스크롤 형식의 카메라워크 기술과 이미지의 생명력을 넣어주는 애니메이션 등에 대한 설계에 대해 구체적으로 설명한다.

2. 게임의 구성그림1은 전반적인 게임의 흐름에 대해 설명하고 있다. 게임은 시작 장면 후 플레이 화면에서 시작하고 각 장면이 끝나면 최종 장면으로 가게 되면 게임을 끝내게 된다. 각 장면마다 게임이 끝났을 때 다시 시작하거나 맨 처음 장면으로 되돌아가도록 구성한다.



[Fig. 1] System Overview

3. 게임 디자인

3.1 2D 스프라이트 제작

본 절은 2D 스프라이트[6,7] 제작에 대해 설명한다. 2D 게임을 제작하는데 가장 중요한 것은 각 캐릭터의 이미지를 만드는 것으로, 본 논문에서는 그림 2와 같이 직접 스케치하여 제작한다. 그림2는 적과 적 보스, 탄환에 대한 초기 스케치를 한 예를 보여준다.



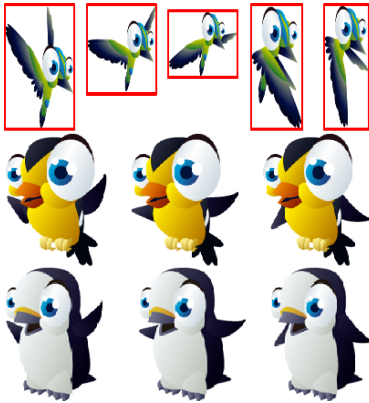
[Fig. 2] Character Sketch

그림3은 그림2의 스케치를 바탕으로 캐릭터, 보스, 배경에 넣을 다른 역할의 물체, 탄환 등을 게임에 도입할 이미지를 만들었으며, 캐릭터 이미지 제작은 어도비 포토샵 CS5, 어도비 일러스트레이터 CS5의 기능 중 Pen tool, Brush tool 등을 사용하였다.



[Fig. 3] 2D Sprite

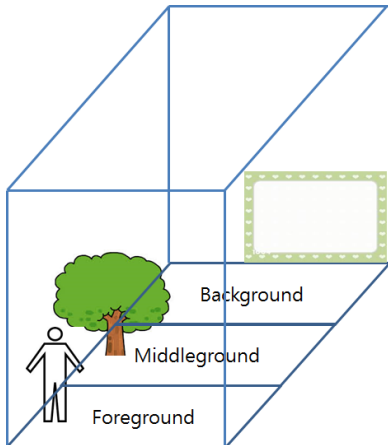
그림4는 게임에서 사용할 애니메이션을 위해 기존에 제작했던 이미지를 어도비 일러스트레이터 CS5로 분할 이미지를 제작한다. 어도비 일러스트레이터의 Rotate Tool과 Shear Tool을 사용하여 기존의 이미지에서 회전을 하거나 선택 부분을 기울여서 분할 이미지[5]를 제작한 장면을 보여준다.



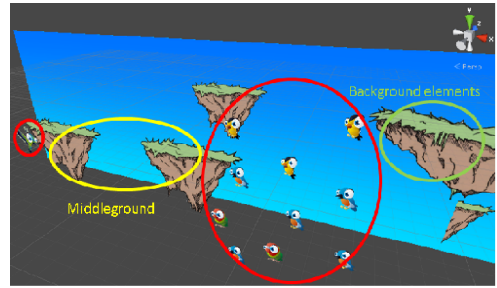
[Fig. 4] Division Images

3.2 배경 입체감

2D게임에서도 지형지물에 대한 원근감이 있어야 게임이 좀 더 자연스럽게 보이기 때문에 배경 입체감을 지정해야 한다. 그림 5에서처럼, 유니티 엔진 내의 지정된 분할된 레이어에 다른 좌표 값을 부여하고, 가장 가까운 전면(Foreground)에는 값을 주지 않고, 중간 장면(Midleground)에는 낮은 값을, 후면(Background)에는 가장 높은 값을 지정한다. 배경공간 안에서 앞으로 나오게 되면 크게 나타나며 멀어지게 되면 작게 보이게 된다. 그림 6은 최종적으로 3D 배경을 적용한 결과를 보여준다.



[Fig. 5] Schematic drawing of Picture space[8]



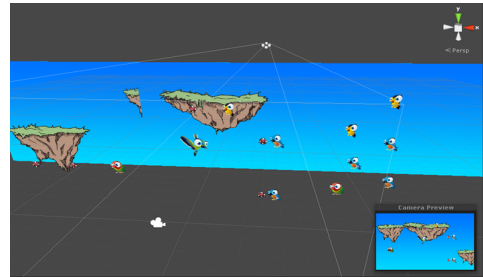
[Fig. 6] Setting for Three-dimensional Background

4. 게임의 설계

4.1 카메라 워크와 적군의 이동

횡 스크롤 방식의 게임은 주인공 주체가 직접 움직여서 카메라 워크가 일어나는 방식을 사용하거나 혹은 자동으로 한 시점을 기준으로 카메라가 주어진 이동 값으로 이동하는 방식 있다.

본 논문에서는 후자의 방식을 사용하여 카메라가 이동하도록 하고 있으며, 카메라 워크 범위에 적군이 감지되면 적군이 게임 화면에 나오는 방식을 사용하고 있다. 그림 7은 프로그램 소스코드에서 정해놓은 카메라가 링크된 카메라가 맞는다면, 시간에 따라 카메라가 정해놓은 이동 값을 받아 이동하게 된다.



```
void Update()
{
    Vector3 movement = new Vector3(
        speed.x * direction.x,
        speed.y * direction.y,
        0);

    이동 *= Time.deltaTime;
    transform.Translate(movement);

    if (링크된카메라)
    {
        Camera.main.transform.Translate(이동);
    }
}
```

[Fig. 7] Camera Work and Pseudocode

4.2 충돌체크와 개체 파괴

슈팅게임에서 적군에게 탄환을 명중시켜 격추 혹은 파괴하는 것은 매우 중요한 문제이다. 본 논문에서는 해당 개체에 대한 체력을 미리 지정하고 탄환에 명중 할 때 나 체력이 소모되어 0이 되었을 때 개체가 파괴 되도록 한다. 게임 물체와 충돌을 감지하는 기능인 콜라이더를 컴포넌트를 이미지에 넣어서 충돌을 확인하게 한다. 그림 8의 소스코드는 해당 게임 캐릭터의 체력이 0이 되거나 같아 졌을 때, 파괴하는 소스코드이다.



```
public int hp = 1; // 체력 포인트 설정

public void Damage(int damageCount)
{
    hp -= damageCount;

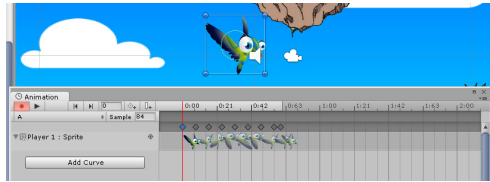
    if (hp <= 0)
    {
        Destroy(gameObject);
    } //체력이 0보다 작거나 같을경우 Object파괴
}
```

[Fig. 8] Pseudocode of the Object destruction and Collision detection

4.3 캐릭터 애니메이션

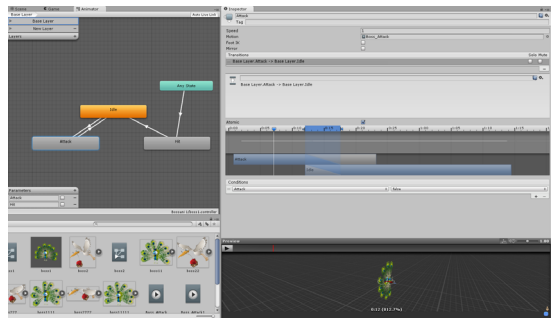
3D 게임에서는 캐릭터 애니메이션을 설계하기 위해 기본적으로 뼈대를 이용하여 애니메이션[9,10,11,12,13, 14,15]을 수행한다. 그러나 본 논문에서는 2D 게임을 설계하기 때문에 이러한 복잡한 설계를 필요로 하지 않는다. 2D 게임에서는 이미지가 살아있다는 생동감을 부여하기 위해서는 지속적인 움직임이 필요하다. 이 때 사용하는 기능이 애니메이션 기능인데, 전 절에서 기술했던 이미지 분할을 캐릭터 애니메이션에서 사용한다. 그림 9는 하나의 이미지에 애니메이션을 적용시켜서 여러

장의 이미지를 원하는 시간대에 적용시켜 캐릭터가 움직이는 생동감을 부여한다[16].



[Fig. 9] 2D Sprite Animation

그림 10은 애니메이션에 추가적인 부분인 애니메이터를 보여준다. 애니메이터는 특정 상황 때 정해진 애니메이션을 취하게끔 시간대를 조절하여 해당 시간대에 공격 애니메이션을 취하거나 다른 애니메이션을 적용 시킨다.



[Fig. 10] Animator

4.4 점수 추가

슈팅게임에서 생존하여 적을 격파한다면 이에 대한 평가를 정할 수 있는데 여기서는 점수를 매겨 게임의 지속성을 유지한다. 점수를 0점으로 시작하여, 지속적으로 점수가 갱신 되도록 한다. 적군을 파괴시켰을 때 내 점수에서 적군에 해당된 점수를 획득하게 되며, 이에 대한 슈도코드는 그림 11에서 보여준다.



```

시작 () {
    내점수 = 0;
}

void Update(){
    this.gameObject.guiText.text = "Score: "+ my_score.ToString();
}

void score_up( int score_obj)
{
    Debug.Log ("score");
    내점수 += 적군의점수;
}
}
    
```

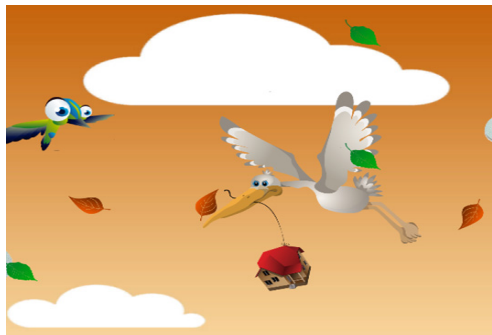
[Fig. 11] Score Pseudocode

5. 실행 결과

본 논문에서 제시한 게임은 윈도우7 64bit, 인텔 코어 i5-3570 3.40Ghz에서 제작하였다. 이미지 제작은 어도비 포토샵, 일러스트레이터를 사용하였고, 프로그램 소스코드는 C#을 이용하였으며, 유니티 엔진 4.31버전으로 제작했다.



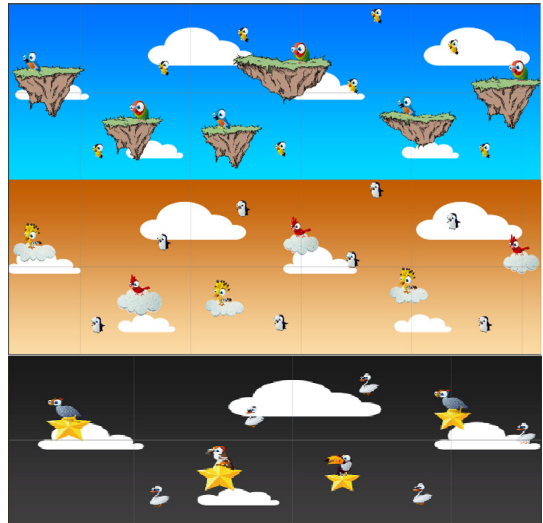
[Fig. 12] Main Screen



[Fig. 13] Game Play Screen

그림 12는 게임의 메인화면을 나타내며, 스토리를 보여주는 것으로 시작하여, 시작 버튼과 조작법, 하드모드 선택에 대한 장면으로 구성되어있다.

그림 13은 게임 플레이 화면 중 보스 전을 플레이하고 있는 그림이다. 인터페이스는 왼쪽 상단에 점수 창이 구성되어 있으며, 아군이나 적군이 격파되었을 때 파티클 시스템으로 만든 2D 이펙트를 보여주게 된다. 보스는 앞에 적군이 나오는 스크롤을 모두 지나치고 난 뒤에 나타나게 된다[17].



[Fig. 14] Background Changes

그림 14는 스테이지 진행에 따라 시간대가 변경되도록 변화를 준 장면이다. 첫 스테이지부터 마지막 스테이지까지의 진행 척도를 알 수 있다. 본 논문에서 디자인한 게임은 현재 PC에서 테스트 한 것으로 향후 모바일에서도 사용할 수 있도록 할 것이다.

5. 결론

본 논문은 남녀노소 모두가 즐길 수 있는 장르인 캐주얼 게임을 2차원 횡스크롤 형식에 맞춰 유니티 3D 엔진을 이용하여 멀티 플랫폼에서 즐길 수 있도록 제작한 것이 큰 특징이다. 특히 유니티 3D엔진은 비용 절감과 멀티 플랫폼 지원을 해주며, 특히 하나의 프로그램 소스코드를 이용하여 다양한 플랫폼에서 구동할 수 있게 해준다.

본 논문에서 제안한 게임의 주요 연령층인 10대에서

20대 남녀 전체를 대상으로 재미있는 시나리오와 간단한 조작을 통해 쉽게 게임에 적응하도록 하였다. 또한 드래곤 플레이트, 쿠키 런과 같이 스테이지로 한정된 게임이 아닌 무한 모드로 설정하는 방식으로 인터페이스 추가와 완성도를 높여서 더욱 완성도 있는 게임을 디자인 할 수 있었다.

REFERENCES

[1] Korea Creative Content Agency, "White Paper On Korean Games", 2013

[2] Creighton, "Unity 3D Game Development by Example Beginner's Guide", 2010

[3] Charles Bernardoff , NGUI for Unity, PACKT, 2014

[4] Will Goldstone, "Unity 3.x Game Development Essentials", Packt Publishing; 2 edition, 2011

[5] Sue Blackman, "Beginning 3D Game Development with Unity 4: All-in-one, multi-platform game development (Technology in Action)", Apress; 2 edition, 2013

[6] Kim Pallister, "Game Programming Gems 5", Cengage Learning, 2005.

[7] Szijarto, G. and K. Jozsef. High Resolution Foliage Rendering for Real-time Applications. Budmerice, Slovak Republic: SCCG, 2003.

[8] 2D Foundations, <http://2dfoundations.wikifoundry.com/page/Foreground,+Middleground,+and+Background>

[9] Lander, Jeff, "skin them bones game programming for the web generation", Game Developer Magazine pp. 11-16. 1998.

[10] Carl Granberg, "Character Animation With Direct3D," CharlesRiverMedia, pp. 47-49, 2009.

[11] Doug L, James and ChristopHer D, "Skinning mesh animations," ACM Transactions on Graphics (ACM SIGGRAPH 2005), pp. 399-407, 2005.

[12] Mark DeLoura, "Game Programming Gems", Charles River Media, 2000.

[13] Mark DeLoura, "Game Programming Gems 2", Charles River Media, 2001.

[14] Andrew Kirmse , "Game Programming Gems 4",

Cengage Learning, 2004.

[15] Weber, Jason, "Run-Time Skin Deformation", Game Developers Conference Proceedings (GDC 2000), pp. 703-721, 2000

[16] Unity DOCUMENTATION - Animation, <http://docs.unity3d.com/Manual/AnimationSection.html>

[17] Bahn Kyoungjin, Hyo Kim, Kyungwon Lee, Hyunhee Kim, "A Study on Effect on Flow of Customized User Interface in Game", Society of Design Convergence, pp. 1-12, 2007.

저자소개

양 동 혁(Dong-Hyuk Yang)

[정회원]



· 2015년 2월 : 배재대학교 게임공학
학과 (공학사)

<관심분야> : 게임 제작 기술

김 수 균(Soo-Kyun Kim)

[정회원]



· 2006년 2월 : 고려대학교 컴퓨터
학과 (이학박사)
· 2006년 3월 ~ 2008년 2월 : 삼성
전자 통신연구소 책임연구원
· 2008년 3월 ~ 현재 : 배재대학교
게임공학과 부교수

<관심분야> : 기하모델링, 게임그래픽, 실감미디어