

PC 환경에서의 3인칭 액션게임 설계

안성옥*, 이희범*, 박동원*, 김수균*, 정진영**

Design of 3D Action Game for PC Environment

Sung-Ohk An*, HeeBum Lee*, Dong-Won Park*, SooKyun Kim*, Jinyoung Jung**

요약

3인칭 액션게임은 많은 마니아층의 사용자들에게 지속적인 관심을 받고 있는 장르이다. 이러한 3인칭 액션게임은 사용자가 캐릭터의 모습과 다양한 액션을 볼 수 있게 하여 게임에 대한 몰입도를 높일 수 있는 특징을 가지고 있다. 많은 장르의 게임들이 게임엔진을 사용하여 제작되고 있지만, 본 논문에서는 게임엔진의 특정화된 기술을 사용하지 않고, 단지 DirectX 라이브러리를 이용하여 3인칭 액션 게임을 설계하여 게임 개발에 들어가는 비용을 최소화 할 수 있다는 장점을 가지고 있다. 또한 본 논문에서는 여러 가지 기본 알고리즘을 이용하여 다양한 이벤트 처리와 애니메이션 효과를 좀 더 효율적인 방법으로 그래픽 디바이스에서 빠르게 처리하도록 한다. 성능에 대한 우수성은 게임의 실험 결과에서 잘 나타난다.

▶ Keywords : 스프라이트, 뼈대 애니메이션, 지형모델, 액션 게임

Abstract

Third Person Action Game is a genre receiving continuous interest from many game enthusiasts. The most distinctive feature about third person action games is that the user can actually see the character as well as the various actions, which in turn increases user engagement. Many games are developed using game engines. However, this study designs a third person action game using only DirectX library instead of the specialized techniques in game engines. By doing so, the game development costs will be minimalized. The study also uses several basic algorithms in order to process the various events and to make the animation effects more efficiently managed in the graphic device. The performance superiority is demonstrated by the results of the study.

▶ Keywords : Sprite, Bone animation, Terrain, Action game

•제1저자 : 안성옥 •교신저자 : 김수균

•투고일 : 2014. 2. 18, 심사일 : 2014. 3. 21, 게재확정일 : 2014. 4. 17.

* 배재대학교 게임공학부(Dept. of Game Engineering, Paichai University)

** 대전보건대학교 바이오정보과(Dept. of Bio-Information, Daejeon Health Science College)

※ 본 논문은 중소기업청에서 지원하는 2013년도 산학연협력 기술개발사업(No. C0145600)의 연구수행으로 인한 결과물임을 밝힙니다.

I. 서론

액션 게임(Action game)은 일정한 스토리 라인에 따라 실시간으로 캐릭터의 행동을 버튼 등을 통해 직접 조작하는 게임이다[1]. 기본적으로는 버튼 조작에 의해 캐릭터를 움직이는 게임 전반을 가리키므로 매우 폭넓은 장르를 포함한다. 특히 PC로 개발 되는 게임 장르 중 액션게임은 많은 사용자들의 관심을 받고 있는 분야이다. 이러한 관심은 타 장르와는 다르게 게임 내에서 표현하기 어려운 캐릭터의 현란한 움직임과 화려한 그래픽 효과를 보여주는 것을 특징으로 볼 수 있다. 본 논문에서 제안하는 액션게임에서는 사용자에게 지루하지 않은 볼거리를 제공하며, 빠르게 동작 할 수 있는 저수준의 DirectX를 사용하여 액션 게임의 특성을 살렸다[2]. 이러한 제작방식을 통하여 액션 게임을 적은 비용으로 제작할 수 있게 되었고, 또한 게임 내의 스프라이트를 개선하여 낮은 사양의 PC에서도 빠르게 동작할 수 있도록 설계한다.

본 절은 게임의 설계부분을 나타낸다. 게임 엔진을 사용하지 않고, 저수준의 라이브러리를 사용하여 게임을 개발하는 것은 쉽지 않다. 그러나 저수준의 게임 개발은 개발자에게 코드 개발 부분에 대한 자율성을 보장해 줄 수 있다.

게임에 전반적인 흐름은 그림1과 같이 메인 화면에서 시작하여 게임 진행부분으로 시작하게 된다. 게임은 NPC(Non Player Character)를 통하여 아바타에게 임무라는 게임에 필요한 시나리오를 부여 받게 되고, 사용자는 게임에서의 역할을 받게 된다.

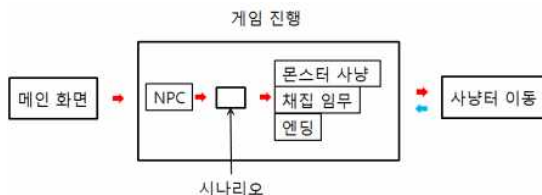


그림 1. 시스템 구조도
Fig. 1. System Overflow

제안 게임에서의 시나리오는 몬스터 사냥, 채집 임무, 최종 엔딩의 순으로 진행하게 된다. 각각의 시나리오는 각각의 임무라는 게임 조건을 통하여 사용자들에게 다른 방식의 진행

방식을 제공하게 되고 다른 방식의 게임 진행을 통하여 게임에 대한 흥미를 증가시키며 사용자가 게임을 즐기는데 있어 지루함을 줄일 수 있는 효과를 갖는다.

II. 3D 액션 게임의 설계

본 절은 DirectX 라이브러리를 이용한 3인칭 액션게임이 구현되는 부분에 대한 내용으로 게임에 사용된 개발기술의 구현방법에 대해 설명 할 것이다.

2.1 스프라이트를 이용한 몬스터 체력바 설계

사용자 인터페이스를 3차원 공간 안에 그려주기 위해 DirectX 라이브러리에서 2D 이미지를 간단히 그려줄 수 있도록 지원해주는 스프라이트 인터페이스를 이용한다. 스프라이트 인터페이스는 2D 이미지의 움직임이나 크기 등을 정하여 간편하게 렌더링 해주는 역할을 한다.

몬스터의 체력바는 각자의 크기와 체력량의 정보로 달라진다. 여기서는 몬스터 체력의 수치를 백분율을 통한 계산식을 사용하여 몬스터가 충격을 받았을 때, 최대 체력바의 길이를 몬스터의 잔여 체력량에 따라, 사용자들에게 보이는 크기를 다르게 만들어준다. 본 논문에서 제시하는 체력바의 크기를 계산하는 방식은 그림 2와 같다.

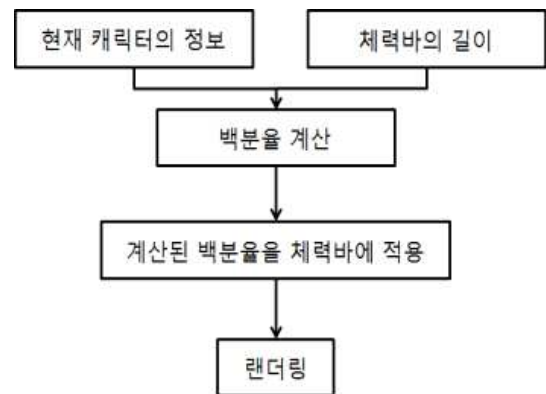


그림 2. 체력바 표시를 위한 계산식
Fig. 2. Calculation Formula for Health Bar

몬스터가 아바타에 의해 데미지를 받았을 때의 아바타가 현재 몬스터에게 준 데미지를 표시를 해주게 한다. 이는 사용자들이 좀 더 게임에 대한 타격감과 자신의 캐릭터에 대한 능력을 알려주는 중요한 역할을 하게 된다.



(a) 몬스터 데미지 (b) 아바타 데미지
 그림 3. 몬스터와 아바타의 데미지 표시
 Fig. 3. Damage of Monster and Avatar

그림 3(a)는 아바타가 몬스터에게 데미지를 주는 것이며, 그림 3(b)는 몬스터가 아바타에게 데미지를 주는 것이다. 몬스터에서의 체력 표시 방법으로는, 그림 2에서 사용된 계산식을 이용하며 몬스터의 머리위에 현재 몬스터의 이름과 체력 바를 띄워주게 되며 데미지를 아바타와 몬스터로 서로 다른 색채로 구분하여 사용자들에게 좀 더 게임에 대한 이해도를 높일 수 있는 효과를 줄 수 있다[3].

2.2 스프라이트 설계

스프라이트 렌더링이란 항상 사용자가 보고 있는 카메라를 향하여 바라보게 하는 효과를 말하는 것이다[4,5]. 이러한 스프라이트 효과는 게임에서 많이 사용 되고 있으며, 사용자 인터페이스에 관한 2D 이미지를 또한 스프라이트 효과를 내기 위해 이용하게 된다.

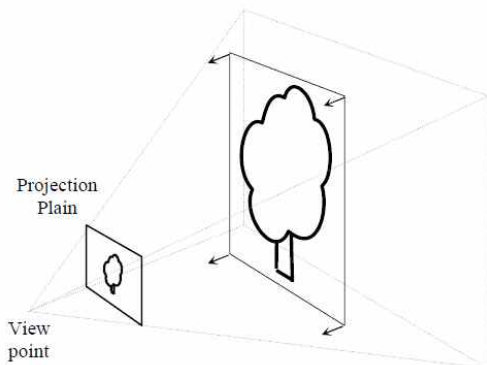


그림 4. 스프라이트 렌더링. 텍스처된 폴리곤은 카메라를 향하고 있음(5)

Fig. 4. Sprite Rendering. The textured polygons is always facing the camera(5)

그림 4에서 보여준 스프라이트 효과는 3차원 공간 안에서 2D 이미지의 회전을 통하여 보이며, 회전 값은 현재 사용자가 바라보고 있는 카메라의 회전각도 값을 불러와 계산된다. DirectX에서는 4x4행렬 좌표계를 통하여 이동, 크기, 회전을 지정해야 하며, 스프라이트 효과를 주기위한 회전은 카메라의 4x4행렬의 값을 가져와 2D이미지의 회전 행렬 값에 대입해준다. 두 개의 회전 행렬 값을 일치하게 만들면 항상 같은 곳으로 회전하기 때문에 2D이미지에 스프라이트 효과를 줄 수 있다.

카메라와 2D이미지의 회전행렬의 값이 같아지면 2D이미지가 항상 사용자 카메라를 향하여 보이는 스프라이트 효과가 있다는 걸 볼 수 있다.

2.3 지형충돌 설계

본 절에서는 사실적인 지형을 표현하기 위하여 3DMAX 툴을 이용하여 지형을 제작하여 3차원 공간으로 X파일을 불러와 지형을 표현한다. 지형모델은 일반적인 방법으로 지형의 틀에 맞춰 움직이거나, 막히는 것을 표현해 줄 수 없기 때문에 DirectX 라이브러리에서 지원하는 D3DXIntersect함수를 사용하여 충돌체크를 한다.

D3DXIntersect 함수는 한쪽 방향에 투명한 선을 쏘아 충돌여부를 판단하는 함수이다. 그림 5는 제안한 액션 게임에서의 지형 충돌 설계를 적용한 결과를 보여준다. 특히 그림 5의 화살표 방향과 같이 5가지의 방향을 향하여 선을 발사하게 된다. 이러한 선은 각각의 충돌 거리에 따라 현재 캐릭터와 지형 맵의 위치를 판별하게 되고 해당 거리에 따라 캐릭터의 위치가 밀리거나 당겨지는 방식으로 현재 지형에 맞춰 캐릭터의 높이와 앞에 있는 벽을 체크하게 된다[2].



그림 5. D3DXIntersect함수를 이용한 지형 충돌체크
 Fig. 5. Collision Detection using D3DXIntersect function

2.4 캐릭터 애니메이션 설계

액션 게임에서의 캐릭터 애니메이션은 게임의 가장 중요한 부분으로 작용하며 많은 애니메이션을 통하여 사용자에게 몰입도를 증가시키는 부분이기도 하다[2, 6]. 그림 6은 본 게임에서 적용하는 캐릭터 애니메이션에 대한 그림을 보여주며, 그림에서와 같이 캐릭터는 다양한 뼈대를 통하여 계층구조의 형태를 갖는다.

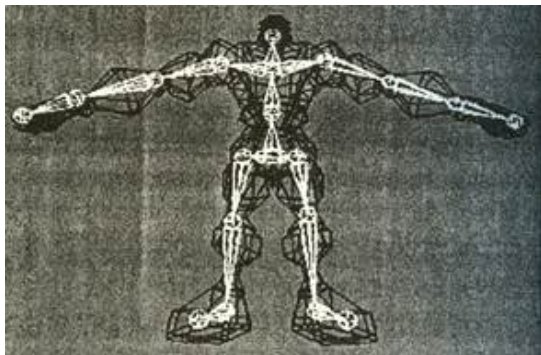


그림 6. 뼈대를 이용한 애니메이션(2)
Fig. 6. Character Animation using Bone(2)

이러한 뼈대 애니메이션은 뼈대를 이동시키거나 회전시키면, 메쉬의 표면도 이동되거나 회전된다[2,6,7,8,9,10,11]. 메쉬 표면은 사람의 피부처럼 보이며, 메쉬 표면의 각각의 정점들은 뼈대와 연결되어 연결된 뼈대에 의해서 정점의 위치가 결정된다. 이러한 뼈대의 움직임을 매 프레임에 저장하여 캐릭터의 동작을 만드는 것이 뼈대 애니메이션이다.

2.5 스키닝 설계

계층구조로 만들어진 뼈대에 메쉬 정보를 붙여 하나의 3차원 모델로 만든다. 만들어진 모델의 메쉬는 하나의 메쉬를 이

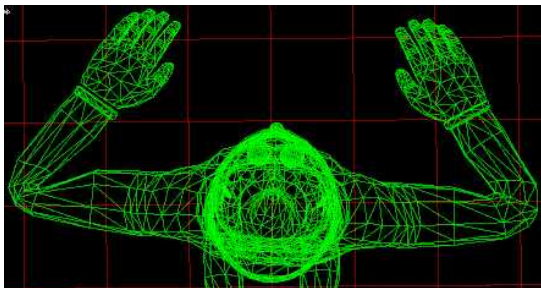


그림 7. 팔꿈치 구부러짐 [12]
Fig. 7. Bending Elbows [12]

루거나 분할 메쉬로 이루어지게 되며, 이러한 메쉬는 관절 비틀림 현상이 일어날 수 있게 된다. 그림 7은 관절 비틀림 현상의 예와 개선된 방법에 대해 보여준다. 왼쪽은 팔꿈치를 구부렸을 때 관절을 비튼 것처럼 보이고, 오른쪽은 이를 개선한 결과를 보여준다.

관절 비틀림 현상을 해결해주기 위해 스키닝 기법을 통하여 보간을 해야 한다. 스키닝 기법이란 서로 연결된 뼈대 정보의 해당하는 메쉬 정점에 다수의 뼈대 메쉬 가중치를 적용하여 자연스러운 애니메이션을 만들어주는 기술이다 [8,6,9,11,12].

$$V_{world} = V_{world}M_0W_0 + V_{world}M_1W_1 + \dots + V_{world}M_{n-1}W_{n-1}$$

스키닝 기법은 위에 표시한 수식과 같이 메쉬 행렬, 뼈대 월드 역행렬, 가중치를 곱하며 연결된 뼈대 정보만큼 추가적인 행렬 값을 더하여 서로 연결된 뼈대의 메쉬 월드 행렬에 대한 최적의 값을 구하여 각각의 메쉬 최종 월드 행렬에 값을 만들어 준다. V_{world} 는 최종 월드(world)에서의 스킨이며, V_{world} 를 추정하기 위해 V_{world} 에 M_0 인 월드 상에서 뼈가 움직인 만큼의 애니메이션이고, W_0 는 가중치를 말한다. 즉 로컬에서 변환된 뼈의 움직임을 추정하는 것이다.

III. 개발결과

본 논문은 다음과 같은 실험 환경에서 게임을 설계하여 실험하였다. 하드웨어는 VGA 디바이스 AMD radeon HD 6870 1GB램이며, 운영체제 Windows 7 환경에서 Microsoft Visual Studio 2008을 이용하여 C++, API, DirectX 라이브러리를 이용하여 하였다. 서 수행한 화면이다. 게임의 목표는 주인공이 NPC(Non Player Character)



그림 8. 게임 메인 화면
Fig. 8. Game Main Screen

에게 시나리오상의 임무를 받아 사냥터에서 해당하는 임무를 완수하여야 하는 것이고 3가지의 임무를 완수하면 엔딩을 볼 수 있다.

그림 8은 게임의 메인화면을 나타낸다. 게임의 메인화면은 게임의 제목과 함께 게임의 시작과 게임종료 버튼으로 구성되어 있으며 화려한 전투의 게임 배경으로 게임에 대한 이해를 도와주고 있다.



(a) 사냥터에서의 전투



(b) 마을



(c) 보스 몬스터의 등장

그림 9. 게임에서의 각각의 지형맵
Fig. 9. Various Terrain Map

게임은 그림 9와 같은 여러 개의 맵으로 나뉘어져 있고 (a)는 사냥터이고, (b)는 마을로 구분된다. 전반적인 게임 진행은 마을에서 임무를 받아 사냥터로 이동하여 적을 섬멸하고, 임무를 완수하는 방식이다. 각각의 사냥터에서는 몬스터

들이 아바타를 공격하게 되며, 본 제안 게임에서는 플레이어가 추가공격을 할 때만 그림 9(c)와 같이 보스가 피격 애니메이션을 취하도록 하였다.

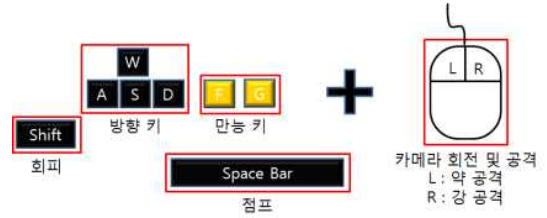


그림 10. 키 사용법
Fig. 10. The Usage of Key

그림 10에서 설명한 조작키를 통하여 게임에서의 여러 가지 액션을 취할 수 있다. 만능키를 통하여 임무의 수락과 완료 및 게임 내에서의 아이템을 옮길 수 있게 된다. 또한 마우스의 좌, 우 버튼을 통하여 게임 내에서의 공격의 방식이 바뀌게 되며 눌러진 순서에 따라 플레이어의 공격력에 차이가 있다.



그림 11. 임무 수락 창
Fig. 11. Window of the Responsibility

게임 내에서의 임무는 그림 11과 같이 NPC(Non Player Character)를 통하여 받을 수 있으며, 시나리오와 사냥터에서의 임무를 부여 받는다 또한 보상으로서는 새로운 무



그림 12. 연속기
Fig. 12. Combo Action

기를 받으며 무기를 받았을 때 주인공 캐릭터의 공격력이 증가한다.



그림 13. 점프 및 회피 연속기
Fig. 13. Jump and Evasion Action

연속기라는 액션게임에서의 재미요소는 다양한 애니메이션을 통하여 사용자에게 몰입감을 증가시키는 효과가 있다. 그림 12는 현재 게임에서의 캐릭터가 다양한 연속기를 통하여 다양한 액션을 선보이는 모습을 볼 수 있다.



그림 14. 다양한 몬스터
Fig. 14. Various Monster in Our Game

다양한 적 캐릭터를 통하여 매번 다른 시나리오를 진행하는 사용자에게 볼거리를 제공하며 서로 다른 적 캐릭터는 각자 다른 애니메이션과 공격방식을 취하게 된다. 이러한 요소는 사용자에게 게임에 대한 흥미와 시나리오에 맞는 몬스터를 통하여 이해감을 높일 수 있는 효과를 갖는다.

IV. 결 론

본 논문은 DirectX 라이브러리를 이용하여 3인칭 액션게임을 설계함으로써 기존의 게임엔진에서 제공하는 개발 기술을 사용하지 않고 게임을 제작하였다. 이러한 저 수준의 개발 방식은 프로그램의 하위 단위의 설정을 통하여 게임에 최적화된 효과를 줄 수 있으며, 개발자가 원하는 방식의 프로그래밍을 구현할 수 있다. 또한 최소의 비용 발생으로 인해 게임 개발에 있어 프로그래밍 이외의 기술에 더욱 투자를 할 수 있는 효과가 발생하게 된다.

본 게임 설계에서는 캐릭터 애니메이션, 스프라이트를 통한 인터페이스, D3DXIntersect 함수를 통한 지형충돌을 통하여 게임에 자연스러운 움직임, 사실적인 지형구현과 사용자에게 몰입감을 줄 수 있는 인터페이스 등을 직접 구현함으로써 게임 구현에 있어 완성도를 극대화 하고 그래픽을 최적화하여 사용자 플랫폼에 있어 최적화가 효율적인 장점을 가지고 있다.

참고문헌

- [1] WIKIPEDIA, "http://en.wikipedia.org/wiki/Action_game"
- [2] Frank D. Luna, "Introduction to 3D Game Programming with DirectX 9.0c a Shader approach," WORDWARE, pp. 523-530, June 2006.
- [3] 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, May 2007.
- [4] Kim Pallister, "Game Programming Gems 5", Cengage Learning, February 28, 2005.
- [5] Szijarto, G. and K. Jozsef. High Resolution Folaige Rendering for Real-time Applications. Budmerice, Slovak Republic: SCCG, 2003.
- [6] Lander, Jeff, "skin them bones game programming for the web generation:", Game Developer Magazine pp. 11-16. May 1998.
- [7] Carl Granberg, "Character Animation With Direct3D," CharlesRiverMedia, pp. 47-49, April 2009.
- [8] Doug L, James and ChristopHer D, "Skinning mesh animations," ACM Transactions on Graphics (ACM SIGGRAPH 2005), pp. 399-407, August 2005.
- [9] Mark DeLoura, "Game Programming Gems", Charles River Media, August 2000.
- [10] Mark DeLoura, "Game Programming Gems 2", Charles River Media, July 2001.
- [11] Andrew Kirmse, "Game Programming Gems 4", Cengage Learning, March 3, 2004.
- [12] Weber, Jason, "Run-Time Skin Deformation", Game Developers Conference Proceedings (GDC

2000), PP. 703-721, 2000

저 자 소 개



안 성 옥
 1989 : 고려대학교
 컴퓨터학과 이학박사
 현 재: 배재대학교 게임공학과 교수
 관심분야: 데이터베이스
 Email : sungohk@pcu.ac.kr



이 희 범
 현 재: 배재대학교 게임공학과 학생
 관심분야: 컴퓨터 그래픽스
 Email : charon112@naver.com



박 동 원
 1993: Texas A&M 대학교
 컴퓨터공학과 공학박사
 현 재: 배재대학교 임공학과 교수
 관심분야: 게임공학
 Email : dwpark@pcu.ac.kr



김 수 군
 2006: 고려대학교
 컴퓨터학과 이학박사
 2006~2008: 삼성전자 통신연구소
 책임연구원
 현 재: 배재대학교 게임공학과 조교수
 관심분야: 기하모델링, 게임그래픽,
 실감미디어
 Email : kimsk@pcu.ac.kr



정 진 영
 1994: 한남대학교
 컴퓨터공학과 공학석사
 2002: 한남대학교
 컴퓨터공학과 공학박사
 현 재: 대전보건대학교 바이오정보과
 교수
 관심분야: 바이오인포매틱스,
 웹기반 정보 시스템,
 모바일 컴퓨팅
 Email : jyjung@hit.ac.kr