

Game Design of Augmented Reality RPG using Artificial Intelligence

Syoungog An*, Juengbean Bae*, Soo Kyung Kim*

Abstract

Currently the mobile augmented reality does not require supplemental hardware, such as markers like specific images or depth cameras, in order to acquire 3D environment information. This allows for increased accessibility and usability for the general user, and also broadened scope of development. The market research enterprise Counterpoint Research predicted that the growth of the augmented reality market will reach 70 trillion Korean Won by 2021. Application developments are also progressing. Wargaming, the developer of the global game World of Tanks, has launched an application that uses AR core at the GDC 2018. Thus, we must prepare for the impending augmented reality market and acquire top class augmented reality technology using 3D environment. This paper outlines the development of a game using AR CORE SDK to acquire 3D environment and Unity 3D game engine to combine camera images and 3D graphics, which all comes together to construct a natural augmented reality.

▶ Keyword: Camera Tracking, Artificial Intelligence, ARCore, Augmented Reality, Platform

I. Introduction

2017년 차세대 증강현실(Augmented Reality) 기술이라 불리는 애플의 AR 툴킷(Toolkit)[1]과 구글의 AR코어[2]는 앞 다투어 소프트웨어 발표하기 시작하였다. 이 기술들은 특정 지점을 인식하여 기기를 움직이기 전 지점의 위치와 기기를 움직인 후의 지점의 위치와의 차이를 계산하여 실제 지점의 위치와 각도 등의 지형 정보를 유추한다. 또한 부가 기기와 마커 없이 실제 3D 환경 정보를 습득할 수 있어서 사용자의 접근성을 크게 확장시킨 플랫폼이다. 두 플랫폼은 응용 소프트웨어 개발을 지원하기 위해 지형정보의 습득이나 습득된 정보에서 평면 정보를 도출하는 등 AR 프로그래밍을 지원하는 SDK를 제공하고 있다. SDK를 통해 간단하게 정보를 획득할 수 있기 때문에 AR개발자는 획득한 정보를 AR소프트웨어로 확장하는 일에 집중할 수 있다. 본 논문에서는 3D 환경을 유용하게 사용할 수 있는 형태의 AR 게임 앱을 개발한다. 개발 플랫폼은 13가지의 안드로이드 모바일을 지원하는 AR CORE SDK와 3D 환경의 조작성 간편한 유니티 3D 게임엔진[3-9]을 사용한다. 또한

카메라 트래킹과 실제세계의 지형정보와 대응되는 평면 찾기 및 환경 조명을 적용하고 이후 RPG(Role Playing Game)게임 개발 순으로 소개한다.

II. ARCore Design

본 절에서는 AR을 구현하는 ARCore 프로젝트 세팅, 카메라 트래킹, 평면 찾기, 환경조명에 대해 설명한 후, 게임을 개발하는데 필요한 플레이어 캐릭터 이동과 적 캐릭터 생성, 애니메이션, 줌인을 통한 캐릭터 포커스, 전투AI 대하여 설명한다.

1. ARCore Project

ARCore는 안드로이드 SDK를 ARCore라는 앱의 형태로 제

• First Author: Syoungog An, Corresponding Author: Soo Kyun Kim
*Syoungog An (sungohk@pcu.ac.kr), Dept. of Game Engineering, Pai Chai University
*Juengbean Bae (bjbin1234@naver.com), Dept. of Game Engineering, Pai Chai University
*Soo Kyun Kim (kimsk@korea.ac.kr), Dept. of Game Engineering, Pai Chai University
• Received: 2018. 08. 30, Revised: 2018. 09. 05, Accepted: 2018. 09. 17.
• This work was supported by the research grant of Pai Chai University in 2018.

공하고 있고 개발 프로그램에서는 정보만을 받아와 사용하는 형태로 제작한다. 따라서 ARCore를 사용하기 위해서는 사용자의 디바이스에 ARCore 앱을 설치해야 하며 프로그램에서는 통신을 구현해야 한다. ARCore 앱은 플레이스토어에서 다운로드 받을 수 있고, 여기서는 ARCore Unity SDK를 통해 Fig. 1과 같이 통신부분을 만들 수 있다.

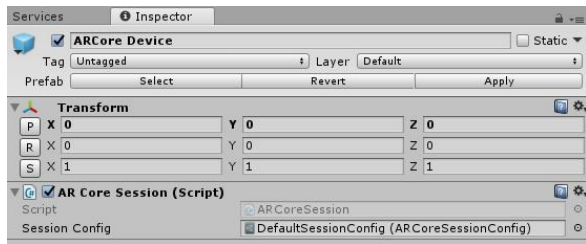


Fig. 1. ARCore Device Object

ARCore Unity SDK를 프로젝트에 임포트하면 프로젝트에 포함되는 ARCore 디바이스(Device) 프리팹을 유니티 씬에 생성한다. 해당 오브젝트의 ARCore 세션(Session) 컴포넌트가 앱과 연결되어 통신할 수 있는 가교 역할을 수행한다. 이후 유니티 내의 카메라가 가상의 오브젝트를 렌더링한 결과에 카메라에 찍힌 모습을 배경으로 합쳐주는 ARCoreBackgroundRenderer컴포넌트를 메인 카메라에 컴포넌트로 설정해준다.

마지막으로 플레이어 셋팅 메뉴에서 안드로이드 SDK 레벨을 7.0으로 맞춰주고 XR 세팅(Settings)메뉴에서 ARCore Supported을 선택해 주면 기본적인 유니티에서의 ARCore 프로젝트 세팅이 완료된다.

2. Camera Tracking

증강현실에서의 카메라 트래킹이란 사용자 디바이스의 각도나 위치변화를 측정하는 기술이다. 3D 좌표에 대한 명확한 기준이 없는 마커리스 기술에서는 카메라의 각도와 위치의 변화를 통해 프로그램 상의 렌더링 카메라를 변환하여 가상환경에서 해당 시야에 존재하는 그래픽을 화면에 그려 줄 수 있다. 때문에 카메라 트래킹은 가상의 그래픽이 카메라의 찍힌 이미지와 실제처럼 융합되는데 중요한 역할을 수행한다. 여기서는 유니티 엔진에서 XR플랫폼을 위해 제공하는 Tracked Pose Driver클래스를 통해 사용할 수 있다.

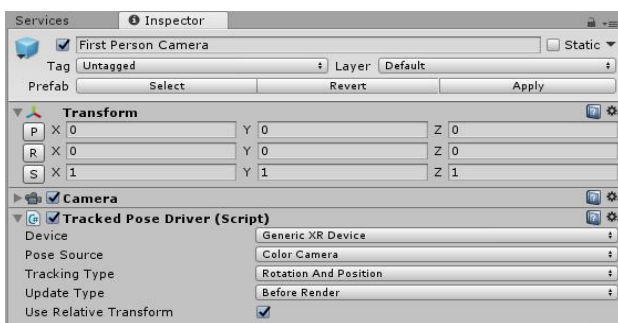


Fig. 2. Camera Object

장치의 변환을 게임 오브젝트의 변환에 적용시켜주는 해당 클래스를 메인 카메라의 컴포넌트로 설정하여 장치의 움직임을 변환 컴포넌트에 적용한다. 이후 설정 값을 Fig. 2처럼 기본 값으로 맞춰주면 디바이스의 위치이동과 각도에 맞추어 변환이 적용되는 카메라를 획득 할 수 있다.

3. Finding Plane

렌더링 카메라로 가상환경에서 측정된 지형에 대응되는 3D 캐릭터의 위치를 지정하기 위해서는, 지형정보를 사용하여 지형의 위치에 대응되는 가상의 지형을 생성하여야 한다. 앞서 3D 환경에 대한 정보는 ARCore 세션 컴포넌트를 통해 통신하는 것에 대해 설명하였다. Fig. 3과 같이 평면 찾기는 해당 컴포넌트의 함수를 통해 접근하여 평면 정보를 찾아올 수 있다. 세션 클래스에는 GetTrackables 함수가 존재하고 평면의 위치와 크기를 포함하는 클래스 인스턴트를 이 함수로 보내는 것으로 정보를 얻을 수 있다. 이후 평면오브젝트를 생성 한 후 리스트를 통해 저장하여 평면을 획득 할 수 있다.

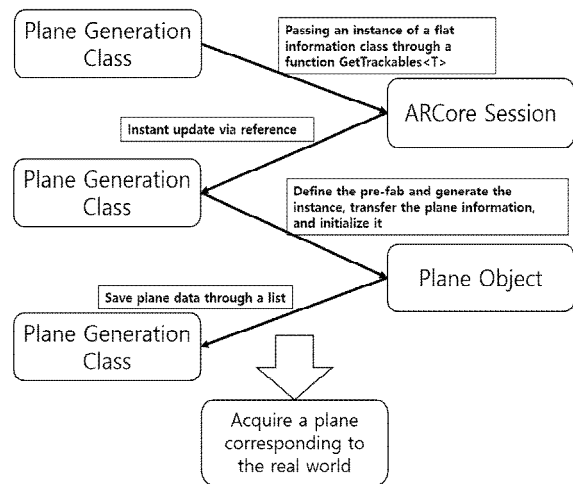


Fig. 3. Plane Generation

생성된 평면은 레이 캐스트를 활용하여 사용할 수 있으며 평면 오브젝트의 TrackedPlaneVisualizer 컴포넌트를 통해 평면을 꼭짓점 정보와 알파 값이 적용된 바닥 메쉬를 통해 Fig. 4와 같이 시각화하여 볼 수 있다.



Fig. 4. Visualized Plane Information

4. Lighting

조명정보는 입체감과 사실감을 표현하는 매우 중요한 요소이다. 명암이 표현되지 않은 그래픽은 카메라 트래킹이나 평면 찾기로 위치와 크기를 조정한다 하여도 입체적인 느낌을 받지 못하며, 특히 인간이 밝기 차이를 강하게 인식하기 때문에 현실의 카메라 이미지와 융합하지 못하게 된다. 증강현실에서는 현실 카메라 이미지에서 대략적인 조명의 정보를 획득하여 그래픽에 적용시켜야 한다. AR Core SDK에서는 이러한 내용을 환경 조명 (Environment Light) 객체와 AR Core 기본 셰이더를 통해 수행한다. 환경 조명 객체에서 빛의 강도 추정치를 획득한 후 셰이더에게 전달하며, 셰이더가 적용된 재질들이 픽셀 값을 수정하여 환경 조명을 구현한다.



Fig. 5. Difference of Character's Color

Fig. 5는 게임에서 사용하는 3D 모델에 환경 조명을 적용한 모습이다. 어두운 위치에 있는 왼쪽과 밝은 위치에 있는 오른쪽의 캐릭터 색상차이를 통해 환경조명이 적용되고 있는 모습을 확인할 수 있다.

III. AR Game Design

1. Movement of Player's Character

플레이어 캐릭터의 경우 사용자가 많이 조작하는 캐릭터 [10-13]이다. 따라서 플레이 화면에 가능한 표시되어야 조작하는 느낌을 받을 수 있으나 움직임 없이 화면상에 표시만 하게 될 경우 증강 현실이 아닌 2D 이미지 같은 느낌을 받게 된다. 따라서 자연스러운 증강 현실을 위해서는 화면 상의 일정 위치를 따라 다니는 캐릭터를 개발하여야 한다. 여기서는 카메라 트래킹을 이용하여 카메라의 변환을 적용 시킨다. Fig. 6에서와 같이, 카메라와 일정한 거리의 벡터를 목표지점으로 설정하면 화면의 일정한 부분을 목표지점으로 가질 수 있다.

벡터의 획득은 카메라의 자식 오브젝트로 타깃 포인트를 생성하여 카메라의 움직임이나 회전을 받아서 카메라와 타깃 포인트의 월드좌표를 통해 획득한다. 이후 캐릭터의 변환 컴포넌트의 좌표 값을 해당 카메라 위치에 해당 벡터 값을 더한 위치

로 일정한 속력으로 이동시킴으로 자연스러운 캐릭터의 움직임을 구현할 수 있다.

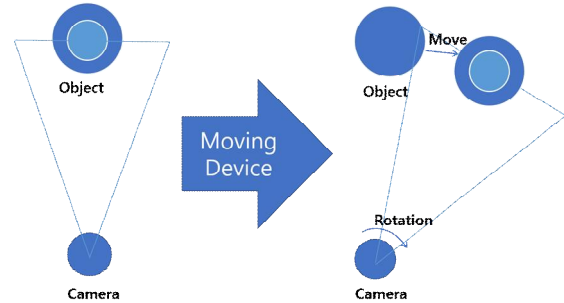


Fig. 6. Moving objects according to device motion

2. Generation of Enemy Character

적 캐릭터의 경우 벽이나 바닥에 걸치지 않고 바닥에 바로 위에서 있거나 아무런 장애물이 없는 공중에 떠 있어야 카메라 이미지와 자연스럽게 융합된 느낌을 받을 수 있다.

제안 방법에서는 바닥에 서 있는 캐릭터와 공중에 나는 두 가지 적 캐릭터를 사용한다. 일정한 위치에서 위에서 아래로 쏘는 레이 캐스트를 통해 '바닥 찾기'에 넘겨주어 적 캐릭터를 생성시킬 바닥을 찾는다. 이후 서 있을 적 캐릭터는 바닥의 좌표와 위쪽 방향 벡터에 맞추어 생성하여 플레이어를 보도록 Y축 회전을 시키고 공중에 있을 적 캐릭터는 바닥의 좌표에서 Y좌표를 증가시켜 생성하고 플레이어를 보도록 방향을 맞추어 줌으로써, 보다 자연스럽게 적 캐릭터를 생성할 수 있다. Fig. 7은 적 캐릭터를 생성한 결과를 보여준다.



Fig. 7. Enemy Character

3. Animation and Animator

게임 속 3D 캐릭터들의 자연스러운 움직임은 현실감을 크게 높여주며 반대로 자연스럽게 못한 움직임은 게임의 평가를 낮추는 요인이 되기도 한다. 하지만 개발자가 획득할 수 있는 애니메이션 자원은 한정되어 있기 때문에 한정된 자원을 최대한 활용하는 것이 중요하다. Fig. 8에서와 같이 유니티의 애니메이터 기능을 이용하여 FSM(유한 상태 기계) 모델로 시스템을 구현한다. 즉 주요 상태를 구분하고 캐릭터의 상황에 맞는 상태를 애니메이터에 적용하여 상황에 맞는 애니메이션을 실행 시킨다.

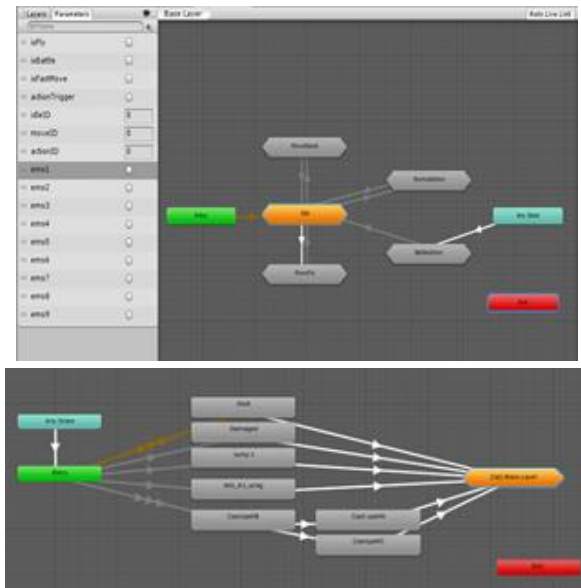


Fig. 8. Unity Animator

유니티 에디터에서 생성한 애니메이터를 실행시키면 애니메이터에 해당 애니메이터 탭이 실행시켜지며 코딩이 아닌 시각적인 설계가 가능하기 때문에 비교적 쉽게 애니메이션을 다룰 수 있다.

Fig. 8의 오른쪽 위처럼 기본적인 플레이어의 상태를 구분할 파라미터를 생성하며 사용하는 애니메이션이 많을 경우 좌측상단처럼 스테이트 머신을 통해 상태 집합을 구분하여 화살표로 표시되는 전이 조건을 정의해 주면 쉽게 애니메이션 그룹을 구분할 수 있다. 그리고 하단처럼 스테이트 머신 내부의 실제 적용되는 상태와 전이를 구현함으로써 애니메이터를 생성한다. 이렇게 제작된 애니메이터는 코딩에서 간단하게 파라미터만 수정해 주게 되면 상태에 맞는 자연스러운 애니메이션을 실행할 수 있다.

4. Combat Scene

ARCore기술은 아직 평면 만을 인식할 수 있다. 따라서 캐릭터들의 움직임을 자연스럽게 구사하기 어렵고 몰입감을 방해한다. 그렇게 때문에 제안 방법에서는 투사체를 통해 전투를 실시한다. 우측 하단의 타겟팅 버튼을 터치 시, 화면 중앙과 가장 가까운 적에게 타겟팅 표시를 해주며 내부적으로 해당 적의 정보를 플레이어에게 넘긴다. 그 후 스킬 버튼을 누르게 되면 스킬에 해당하는 투사체를 생성하며 투사체에게 타겟팅된 적의 정보를 넘긴다. 투사체는 설정된 속도로 이동하여 콜라이더(Collider)를 통해 충돌 체크를 하며 충돌 된 적의 캐릭터 클래스에 상속된 데미지 함수 등을 구현한 인터페이스로 접근하여 데미지를 부여한다.

5. Character's Focus

게임 속 장면이나 상호작용에 있어 해당 캐릭터에게 초점을 주는 것은 게임의 몰입도를 상승시키는 기법 중 하나다. 여기서

는 카메라의 줌인과 같은 효과를 주어 캐릭터에게 포커스를 줄 수 있도록 구현한다.

AR에서 줌인은 간단히 가상 캐릭터만을 확대시키는 것이 아니라 카메라에 찍힌 실제 배경 또한 같이 확대되어야 한다. 때문에 단순히 유니티 상의 카메라의 위치나 설정을 수정하는 것이 아니라 카메라와 함께 렌더링된 이미지를 확대시켜야 한다. ARCore는 백그라운드 렌더러 컴포넌트가 카메라의 렌더링 데이터에 배경이미지로 합성시켜 렌더링 한다. 그렇기 때문에 해당 카메라의 렌더링 데이터를 렌더링 텍스처로 가져와 원하는 위치를 기준으로 확대시키면 실제 카메라의 줌인과 비슷한 효과를 연출 할 수 있다. Fig. 9는 몬스터 캐릭터를 생성한 모습이고, Fig. 10은 줌인을 적용한 결과를 보여준다.

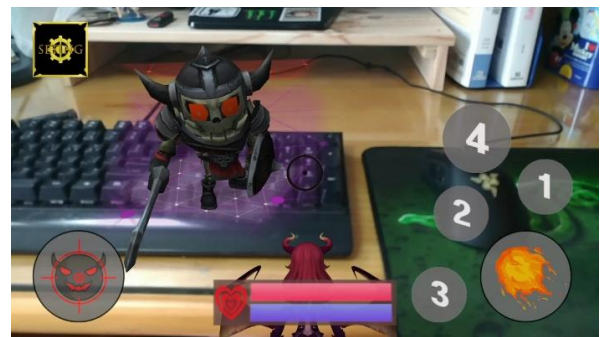


Fig. 9. Monster Character



Fig. 10. Zoom-In Scene

6. AI for Combat

게임에서의 AI[14,15]에 대한 정확한 정의는 있지 않지만 일반적인 AI와 마찬가지로 간단한 인간의 사고를 컴퓨터로 구현시킨 것을 말한다. 여기서 캐릭터 행동을 선택함에 있어 필요한 정보를 획득하고 미리 정해진 패턴으로 의사를 결정한 후 실행시키는 것으로 한다. 3.3의 애니메이터도 캐릭터의 움직임을 결정하는 AI라 할 수 있다. 애니메이터에도 사용되는 FSM(유한 상태 기계) 모델은 게임 AI의 초석이라 할 만큼 게임 AI와 잘 맞는 모델이라는 평가를 받고 있고 여기서도 전투 AI를 FSM 기반으로 한다.

특히 전투 AI는 전투 매니저 클래스에 의해 각각의 적 캐릭터에 붙여져 관리되며 남은 HP량에 따라 일정 시간마다 랜덤한 공격을 하는 것을 목표로 한다. 남은 HP에 따라 분노 상태

를 분류하고 공격의 쿨타임 상태를 체크하여 준다. 이후 쿨타임이 상태가 공격가능 상태 일 때 공격을 실행시킨 후 분노 상태에 따른 쿨타임을 적용 시켜주는 간단한 전투 AI를 구현한다.

IV. Experimental Results

본 절에서는 AR 게임 개발 환경에 대해 설명한다. 유니티 3D 2017과 ARCore 유니티 SDK 1.1.0을 이용하였고, 프로그래밍 언어는 C#을 사용하여 개발하였다. 특히 NGUI를 이용하여 메인 화면과 전투 UI를 구현하는 것으로 프로토타입을 개발하였다. Fig. 11은 제안한 게임에서의 메인 장면을 보여주고 있으며, Fig. 12는 제안한 AR 게임의 최종 결과를 보여주며, 각 캐릭터끼리의 전투 장면을 나타낸다.

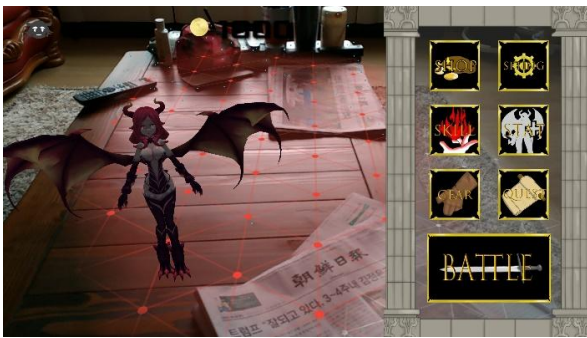


Fig. 11. Main Scene

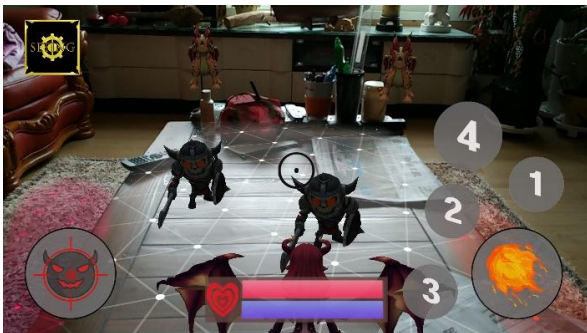


Fig. 12. Combat Scene

V. Conclusions

본 논문에서는 AI를 접목한 증강현실 게임을 제작하는 방법에 대해 소개하였다. ARCore의 사용 방법은 간단하여 쉽게 사용할 수 있으며 효과적으로 증강현실을 제작할 수 있기 때문에 증강현실 개발자들에게 유용한 개발도구가 될 것이다. 그렇지만 현재 ARCore는 처음 평면 인식에서 시간이 걸리고 연결된

지형이 아닌 단순한 평면만을 인식하는 등 완벽한 3D 환경을 인식할 수는 없는 단점을 가진다. 그렇게 때문에 실시간으로 진행되는 게임 개발은 어려울 수 있지만 이후 개발이 진행되어 ARCore의 기능이 개선된다면 게임으로서의 다양한 활용이 가능할 것이다.

REFERENCES

- [1] Apple AR Toolkit <https://developer.apple.com/arkit/>
- [2] Google AR Core SDK <https://developers.google.com/ar/develop/downloads>
- [3] Creighton, "Unity 3D Game Development by Example Beginner's Guide", 2010
- [4] Will Goldstone, "Unity 3.x Game Development Essentials", Packt Publishing; 2 edition, December 20, 2011
- [5] Charles Bernardoff , NGUI for Unity, PACKT, 2014
- [6] Creighton, "Unity 3D Game Development by Example Beginner's Guide", 2010
- [7] Will Goldstone, "Unity 3.x Game Development Essentials", Packt Publishing; 2 edition, December 20, 2011
- [8] Sue Blackman, "Beginning 3D Game Development with Unity 4: All-in-one, multi-platform game development (Technology in Action)", Apress; 2 edition, August 27, 2013
- [9] Charles Bernardoff , NGUI for Unity, PACKT, 2014
- [10] Lander, Jeff, "skin them bones game programming for the web generation:", Game Developer Magazine pp. 11-16. May 1998.
- [11] Carl Granberg, "Character Animation With Direct3D," CharlesRiverMedia, pp. 47-49, April 2009.
- [12] Doug L, James and ChristophHer D, "Skinning mesh animations," ACM Transactions on Graphics (ACM SIGGRAPH 2005), pp. 399-407, August 2005.
- [13] Mark DeLoura, "Game Programming Gems", Charles River Media, August 2000.
- [14] Ian Millington and John Funge, Artificial Intelligence for Games,CRC Press; 2 edition (August 6, 2009)
- [15] Mat Buckland, Programming Game AI by Example, Jones & Bartlett Learning; 1 edition (October 22, 2010)

Authors



Syungog An received the Ph.D. degrees in Computer Science and Engineering from Korea University, Korea, in 1989. She is currently a Professor in the Department of Game Engineering, Paichai University.



Juengbean Bae received the B.S. degrees in Game Engineering from Paichai University, Korea.



Soo Kyun Kim received Ph.D. in Computer Science & Engineering Department of Korea University, Seoul, Korea, in 2006. He joined Telecommunication R&D center at Samsung Electronics Co., Ltd., from 2006 and 2008. He is now a professor at

Department of Game Engineering at Paichai University, Korea. His research interests include multimedia, pattern recognition, image processing, mobile graphics, geometric modeling, and interactive computer graphics. He is a member of ACM, IEEE, IEEE CS, KACE, KMMS, KKITS and KIIT.