

Design of 3D Running Game using Gyro Sensor in Mobile Environment

Joo-Young Choi*, Seok-Hun Kim**, Ho-Young Kwak***, Soo Kyun Kim****

*Researcher, Tsis Co, Seoul, Korea

**Professor, Dept. of e-Business, Paichai University, Daejeon, Korea

***Professor, Dept. of Computer Engineering, Jeju National University, Jeju, Korea

****Professor, Dept. of Computer Engineering, Jeju National University, Jeju, Korea

[Abstract]

In our society, the current smart devices have been commercialized and more time than most people are in contact with smart devices than the time in contact with the PC, it shows the same trend in terms of the game industry. Thus in the study, it was to try to make a game by utilizing the characteristics of only the smart devices in mobile environments among them by using the gyro sensor implementing the movements of characters. In particular, undergraduate student made various attempts to implement a single game using the sensor used in smart devices. In this paper, we have planned the adventure 3D running game that allows users to easily fun to play. Our goal is to implement one stage. We have discussed that you have designed and implemented.

▶ **Key words:** Prefab, Animation, Gyro Sensor, 3D Running Game, Unity3D

[요 약]

현재 모바일 기기가 보편화된 사회에서 대부분의 사람이 PC를 접하는 시간보다 스마트 기기를 접하는 시간이 늘어가고 있으며 이러한 변화는 게임 산업 측면에서도 같은 변화의 추세를 보이고 있다. 이에 따라 본 연구에서는 모바일 환경에서 스마트 기기만의 특성을 살려 게임을 만들어보 고자 하였으며, 그중에서도 자이로 센서를 활용하여 캐릭터의 움직임을 구현하는 데 초점을 두었다. 특히 학부 학생이 스마트 폰에서 쓰이고 있는 센서를 이용하여 하나의 게임을 구현하기 위해 다양한 시도를 하였고, 제안한 게임에서는 사용자들이 재미있고 쉽게 플레이할 수 있는 어드벤처 3D 러닝 게임을 기획단계에서 부터 게임 설계까지 하였다. 이에 대한 검증으로 학부 학생들을 대상으로 본 게임에 대한 재미와 구현 난이도에 대해 평가하였다.

▶ **주제어:** 프리랩, 애니메이션, 자이로 센서, 3D 러닝게임, 유니티3D

-
- First Author: Joo-Young Choi, Corresponding Author: Soo Kyun Kim
 - *Joo-Young Choi (calewell@naver.com), Tsis Co
 - **Seok-Hun Kim (kimshn@pcu.ac.kr), Dept. of e-Business, Paichai University
 - ***Ho-Young Kwak (kwak@jejunu.ac.kr), Dept. of Computer Engineering, Jeju National University
 - ****Soo Kyun Kim (kimsk@jejunu.ac.kr), Dept. of Computer Engineering, Jeju National University
 - Received: 2021. 09. 02, Revised: 2021. 10. 14, Accepted: 2021. 10. 14.

I. Introduction

모바일 기기를 흔하게 볼 수 있는 현대 사회에서는 많은 사람이 모바일 기기를 이용하여 다양한 활동을 하고 있다. 이러한 현상은 게임 시장에서도 같은 양상을 보이며, 게임 시장 분석 기관인 뉴주(Newzoo)에서는 최근 보고서[1, 2]를 통해 2020년 세계 게임 시장의 규모가 약 1,593억 달러(한화 193조 원)로 전망되었으며, 특히 눈에여볼 대목으로는 모바일 게임과 온라인 게임 게임의 비중이 각각 27%로 '동률'이 될 것으로 보였다.

또한, 게임백서에 따르면 국내 게임 시장은 2019년을 기준으로 전체시장의 49.7%를 모바일 게임이 차지하고 있으며, 이는 시장의 절반을 모바일 게임이 선점하고 있는 것을 알 수 있다. 이를 통해 PC 게임의 하락세가 뚜렷한 것을 알 수 있다. 모든 게임 플랫폼의 매출이 전년 대비 증가하였지만, PC 게임은 감소하고 있다. 이는 2016년을 기점으로 온라인 게임과 모바일 게임의 게임 시장 규모의 격차가 점차 줄어들고 있다는 것을 알 수 있다. 이러한 추세를 통해서 본 연구에서는 모바일 기기에서 구동할 수 있는 게임에 역점을 두었다. 특히 제안 게임에서는 스마트폰 환경에서 손쉽게 사용하고 즐길 수 있는 어드벤처 게임 장르 중에서도 간단히 개발할 수 있는 러닝 게임을 디자인하고자 하며, 스마트 기기에 기본적으로 내장되어있는 자이로 센서를 사용하여 게임 내의 캐릭터 조작을 간편화하고, 사용자가 편하고 쉽게 게임 플레이를 할 수 있는 방법에 대해 소개한다.

II. Structure of Game Framework

1. Structure of Game Framework

현재 게임 엔진은 다양한 사용자에게 개방되어, 빠르고 쉽게 2차원 혹은 3차원 게임을 개발할 수 있는 도구로 자리 매김을 하였고, 여러 종류의 게임 엔진 중에서 제안 방법은 가장 많이 사용하는 게임 엔진인 Unity엔진[3,4,5]을 사용하였다. 제안 방법은 3D 환경을 게임 환경으로 구성하였고, 자이로 센서를 활용하여 캐릭터 조작을 구현하였고, 이를 위해 안드로이드 SDK[6]를 사용하여 안드로이드 플랫폼 환경에서 구동되도록 개발한다.

제안 방법에서 제시한 게임은 크게 4가지의 화면으로 나누어진다. 사용자가 게임을 시작하기 전 캐릭터의 현재 레벨을 알 수 있으며, 게임을 시작하기 위한 버튼이 있는 메인화면. 실질적으로 게임 진행을 담당하는 게임화면. 게

임화면에서 일시 정지 버튼을 눌렀을 때 다시 하기, 이어하기, 계속하기 등 상황에 따라 사용자가 게임을 어떻게 플레이할 것인지 정할 수 있는 일시 정지 화면. 캐릭터의 체력이 모두 소모되었을 경우 현재 캐릭터가 얻은 점수와 경험치를 보여주게 되는 점수 화면으로 나뉘며, 게임의 전체적인 흐름도는 (그림 1)과 같다.

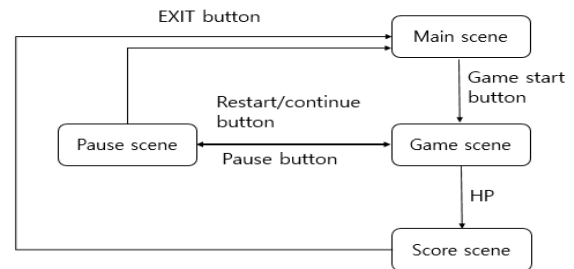


Fig. 1. Flowchart

2. Movement of Character using Gyroscope

현재 우리가 흔히 사용하는 모바일 장치에는 여러 가지 센서가 포함되어 있으며, 많은 모바일 게임은 보통 자이로 센서를 사용하여 캐릭터의 좌 또는 우 움직임을 구현한다.

자이로스코프(gyroscope) [7]는 방향을 결정하는 장치로써 보통은 지구의 중력을 사용한다. 자이로 센서는 각속도를 감지하는 장치이며, 시간 단위당 회전 각도에 대한 변화를 측정하며, 각속도를 감지하게 된다. 스마트폰에는 상하좌우 동작을 감지하는 자이로스코프 센서를 가지고 있기 때문에 다양한 동작을 인식하여 입체감과 균형감을 보다 정밀하게 감지할 수 있다는 장점을 가지고 있다.

자이로 센서는 스마트폰이 기울어짐에 따라 화면과 화면상에 위치되어 있는 물체들 사이의 각속도를 측정하여 움직이고자 하는 특정 물체의 움직임을 조정하는 방법으로, 사용자가 스마트 기기를 기울임으로써 물체가 동일하게 기울여지는 것을 활용하여 기본적인 장치 회전에 따른 화면 전환이나 다양한 게임에 접목 시킬 수 있는 센서이며, 현재 VR(Virtual Reality) 기술[8, 9]이 다양하게 접목되고 있는 게임 산업 시장에서 이제는 필수 불가결한 요소가 되었다고 보아도 무방하다.

Table 1. Pseudo code for movement

```

void Move(){
    float x = Input.acceleration.x;

    if(x < -0.2f && canLeft && isGround)
        dirX = -0.6f;
    if(x > 0.2f && canRight && isGround)
        dirX = 0.6f;
}
  
```

<표 1>과 같이 Input.acceleration을 사용하여 구현 가능하며, 캐릭터의 기본적인 동작들을 미리 스크립트 상에 설정해놓으면, 스마트 장치가 움직이는 각도의 범위를 추가로 설정하여 사용자가 스마트 기기를 얼마큼 기울이며, 어떤 쪽 방향으로 기울임이 있는지에 따라 화면상의 물체가 사용자가 원하는 방향으로 움직이도록 개발한다.

3. Movement of Character using Animator

Unity에서 지원하는 Animation[10]이라는 기능을 통해 모델에 포함되어 있는 몇 가지의 모션을 연속적으로 연결하여, 스크립트로 별다른 조작을 하지 않더라도 캐릭터가 같은 동작을 반복하여 움직일 수 있는 기능을 구현하였다. 하나의 모션을 다른 모션에 연결해놓는 형식으로 (그림 2)와 같이 사용한다.

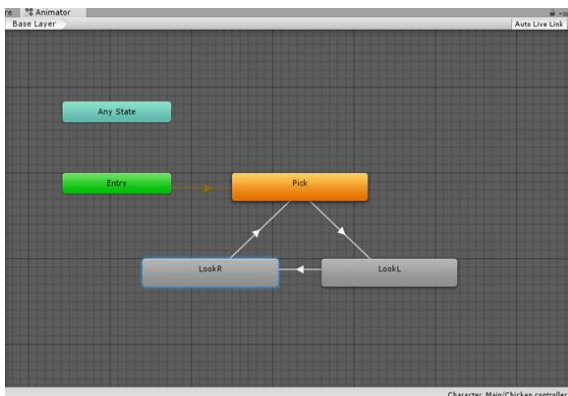


Fig. 2. character motion using the animator

4. Level Design

Unity 엔진에서 자체적으로 제공하는 PlayerPrefs[11]를 이용하여 구현하였다. PlayerPrefs는 데이터를 쉽고 간단하게 저장할 수 있게 해주는 클래스로 일종의 데이터베이스와 같은 기능을 한다고 보면 되는데, Key값과 데이터를 상황에 따라 저장하거나 불러올 수 있는 역할을 하며 서로 다른 스크립트에서 동일한 값을 처리 할 때 주로 사용한다. SetInt를 사용하여 Key값을 설정하고 해당 키에 넣을 데이터 값을 지정한 후 Save를 통해 값을 저장한다. 만약 저장한 값을 다른 스크립트 상에서 사용하고자 할 때는 GetInt를 사용하여 내가 불러오고자 하는 키(Key)값의 데이터를 불러와 사용하면 된다. 필요 때문에 삭제(Delete)를 사용하여 특정 데이터를 삭제하거나 저장되어 있는 모든 데이터를 제거할 수 있다.

본 논문에서 개발한 게임에서는 캐릭터의 레벨 값과 경험치 값을 PlayerPrefs를 통해 저장하였으며, 경험치나 레벨이 보여야 할 부분에서는 그 값이 보이도록 구현하였

다. 또한, 저장된 경험치의 수치에 따라 캐릭터의 레벨이 변하도록 설정하였고, 그 레벨을 저장하여 저장된 레벨의 수치에 따라 맵의 난이도 조절과 캐릭터 체력의 감소량을 조절하였다. 경험치와 레벨에 따른 맵의 난이도와 캐릭터 체력의 감소량 수치는 <Table 2>을 참고하면 된다.

Table 2. Experience value, level, degree of difficult of the map, exhaustion amount of correlation

Experience value	Level	Num. (0~9)	HP (Max 100)
0~1000	1	1, 3, 5, 7	-35(No3)
1000~5000	2	1, 2, 4, 5, 7	-20(No5)
5000~10000	3	1, 2, 3, 5, 6, 7	-15(No7)
10000~20000	4	1, 2, 3, 4, 5, 6, 7	-10(No10)

III. Implementation

1. Main screen

그림 3은 메인화면을 나타내고 있으며, 메인화면은 NGUI(Next-Gen UI) [12, 13, 14, 15]를 사용하여 사용자 인터페이스를 구현하였다. 이를 통해 현재 캐릭터의 레벨을 알도록 표시해 주었으며, 시작 버튼을 통해 게임이 실행되게 하였고, 이를 통해 게임화면으로 씬이 이동하도록 하였다. 가운데 위치하는 캐릭터가 수시로 움직이게 되는데 이 부분에서 Animator을 사용하여 캐릭터의 움직임을 구현하였다. Animator에 대한 캐릭터 움직임 구현은 본 논문의 2.2를 참고하면 된다.



Fig. 3. Main screen

2. Game screen

맵의 경우 사전에 만들어 둔 여러 개의 프리팹을 캐릭터의 레벨에 맞추어 랜덤으로 나오게끔 스크립트를 작성하였다. 다리는 0번부터 9번까지 총 10개의 다리가 나오도

록 구성하였으며, 9번째 다리에서는 교차로만 랜덤으로 나오게 하였다.

캐릭터의 움직임의 경우 PC 환경에서 플레이할 것과 모바일 플랫폼 환경에서 플레이할 것을 고려하여 키보드로 조작하는 방법과 자이로 센서로 조작하는 방법 두 가지를 사용하여 구현하였다. 캐릭터는 계속 앞으로 나아가도록 하였으며, PC 환경에서 플레이할 경우 특정 키를 누를 때 캐릭터가 좌 또는 우로 이동하며 점프를 할 수 있도록 하였고, 모바일 디바이스에서 게임을 플레이할 경우 자이로 센서를 활용하여 디바이스를 좌 또는 우로 기울였을 경우 캐릭터가 좌 또는 우로 이동하도록 하였고 화면을 아래에서 위로 스크롤 할 때 캐릭터가 점프하고 교차로가 있는 터닝 구간에서 왼쪽으로 오른쪽, 오른쪽에서 왼쪽으로 스크롤 하면 해당 방향으로 캐릭터가 회전하도록 구현하였다.

캐릭터가 움직이면서 동전을 먹으면 점수가 20씩 증가하게 되며, 캐릭터가 먹은 동전의 경우 사라지도록 하였다. 캐릭터가 얻은 점수는 상단 UI의 score 부분에 나타나게 되며 실시간으로 증가하기 때문에 현재 내가 얻은 점수를 한눈에 확인할 수 있다.

만약 게임을 실행하다가 장애물에 부딪혔을 경우 캐릭터는 뒤로 힘을 받아 밀려나게 되며, 체력 바가 일정 부분 깎이게 되는데, 체력 바의 경우 캐릭터의 레벨에 따라 줄어드는 양이 다르게 설정하였다. 만약 체력 바의 길이가 모두 줄어들게 된다면 목표가 충족되었다고 판단되어 게임이 종료되고 점수 화면으로 넘어가게 된다.



Fig. 4. Game screen

3. Pause scene

그림 4에서와같이, 게임화면에서 일시 정지 버튼을 누를 때 보여주는 화면이다. 왼쪽부터 다시 시작하기(Restart), 이어 하기(Continue), 끝내기(Exit) 총 세 가지의 메뉴로 UI를 구성하였으며 사용자가 어떤 메뉴를 클릭하느냐에 따라 게임을 어떻게 플레이할 것인지 정할 수 있게 된다.

일시 정제 화면(그림 5)이 나오면 Timescale을 0으로 하여 게임을 일시 정지시켰다. 만약 사용자가 다시 시작하기(Restart) 버튼을 선택한다면 게임화면의 첫 부분, 즉 게임화면으로 씬이 넘어가도록 하였으며 이때 Timescale을 1로 하여 정지하였던 게임을 시작하도록 하였다. 만약 사용자가 이어 하기(Continue) 버튼을 선택한다면 일시 정지 패널을 보이지 않도록 하고, Timescale을 1로 하여 일시 정지 버튼을 누르기 바로 직전부터 게임을 시작하도록 하였다. 만약 사용자가 끝내기(Exit) 버튼을 누른다면 메인 화면으로 씬이 넘어가도록 하였으며, 이때 Timesclae을 1로 하여 메인의 캐릭터가 움직일 수 있도록 설정하였다.

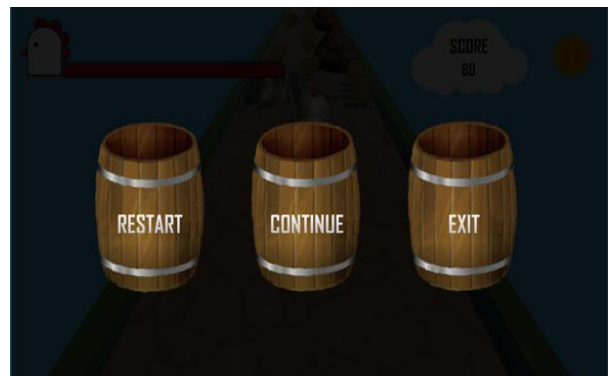


Fig. 5. Pause scene

4. Score scene

점수 화면은 캐릭터의 체력이 모두 소진되었을 때 보이는 화면이다. 맨 위에서부터 캐릭터가 현재 라운드에서 획득한 점수, 캐릭터가 지금까지 얻은 경험치의 양, 그리고 메인 화면으로 나가게 될 끝내기(Exit) 버튼으로 UI를 구성하였다.

점수(그림 6)의 경우 게임을 플레이했을 때 맨 마지막으로 저장되는 점수의 값 즉, 캐릭터가 죽기 바로 직전까지 캐릭터가 얻게 된 점수의 값을 불러와 보여주게 되며, 그 점수의 값을 계속해서 저장한 값을 경험치로 사용하였다.

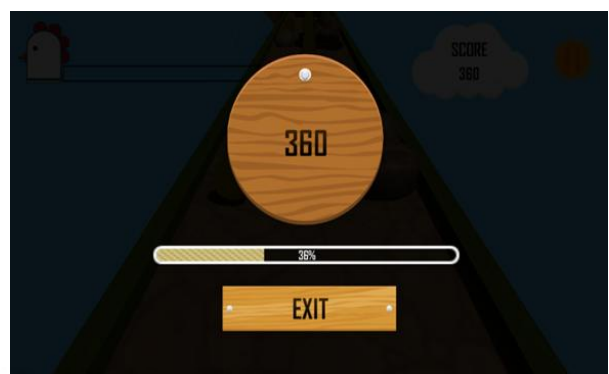


Fig. 6. Score scene

경험치의 경우 PlayerPrefs를 사용하여 사용자가 게임을 종료했다가 다시 실행시키거나, 게임을 여러 번 실행시켰을 때 데이터가 지워지지 않도록 저장하였으며, 레벨 별로 설정된 경험치의 값과 현재까지 모은 경험치의 값을 계산하여 경험치 바로 표현하도록 하였다. 레벨에 따른 경험치 양 계산 값은 <Table 3>와 같다.

Table 3. Calculation of experience point corresponding to the level

Level	Calculate the amount of experience points by level
1	experience point/ 10
2	(experience point - 1000) / 40
3	(experience point - 5000) / 50
4	(experience point - 10000) / 100

만약 사용자가 끝내기(Exit) 버튼을 누른다면 메인화면으로 썸이 넘어가도록 하였으며, 이때 Timescale을 1로 하여 메인의 캐릭터가 움직일 수 있도록 설정하였다.

제안 방법의 특징은 크기 2가지가 있으며 첫 번째는 스마트폰 기기에서 자이로 센서를 활용하여 캐릭터의 좌 또는 우 움직임을 표현하여 사용자들이 쉽게 게임을 플레이할 수 있도록 한 것이다. 두 번째로는 유니티에서 자체적으로 제공하는 Playerprefs라는 기능을 통해 데이터를 저장하고 불러오는 일종의 데이터베이스와 같은 기능을 만들었고, 이를 통해 캐릭터의 레벨과 경험치를 저장할 수 있게 되었다. 저장 후, 레벨에 따라 난이도를 다양화하고, 캐릭터 체력의 향상을 개발하였다. 세 번째, 레벨에 따라 달라지는 경험치의 총량을 계산하여 캐릭터가 얻은 점수를 경험치로 환산하여 캐릭터의 레벨과 현재 그 레벨의 경험치의 양을 표시하여 주었다. 네 번째, 게임을 플레이할 때 일어날 돌발 상황에 대해 사용자가 게임을 직접 일시 정지하도록 하였고, 일시 정지했을 때 다시 시작하기, 처음부터 시작하기, 끝내기 등 세 개의 메뉴를 사용하여 사용자가 일시 정지를 누른 뒤 게임을 어떻게 진행할 것인지를 정할 수 있도록 하였다. 다섯 번째, NGUI를 사용하여 메인화면과 게임화면의 UI를 제작하였고 마지막으로 메인화면의 경우 Animator라는 기능을 통해 따로 조정하지 않더라도 메인화면에 있는 캐릭터가 반복적으로 여러 동작을 마치 하나의 동작처럼 수행하도록 구현하였다.

IV. Conclusions

본 논문에서는 유니티 3D를 사용하여 3D 환경에서 모바일 디바이스의 자이로 센서를 활용하여 캐릭터를 조작하는 3D 러닝 게임을 개발하였다. 특히 학부생이 직접 설계부터 개발까지 하였고, 개발 과정에 대해 소개한 논문으로 기존의 개발된 연구 및 게임을 뛰어넘기에는 다소 부족할 수 있다.

다만, 차후 연구에서는 기존 연구를 면밀히 분석하여 한계를 극복하고, 현재 회전에 대한 구현을 조금 더 부드럽게 구현할 예정이다. 특히 터닝구간에서 캐릭터가 갑자기 90도를 돌기 때문에 이러한 어색해 보일 수 있는 캐릭터의 움직임을 부드럽게 하여 캐릭터가 자연스럽게 회전하는 것처럼 보이도록 하고, 단순히 달리는 것이 아니라 부스터 모드나 아이템과 같은 다양한 콘텐츠를 추가하여 사용자가 게임을 플레이할 때 성취감과 만족감을 극대화할 수 있는 방향으로 개선할 예정이다. 그리고 한 캐릭터만으로 플레이하는 게임은 다소 지루해질 수 있으므로 레벨 조건을 일정 부분 충족하면 캐릭터의 외형이 변화하거나 다른 캐릭터가 나오도록 캐릭터의 다양성을 증가시키는 방법에 관한 연구가 필요하다.

ACKNOWLEDGEMENT

이 논문은 2021학년도 제주대학교 교육·연구 및 학생지도비 지원에 의하여 연구되었음.

REFERENCES

- [1] Yonhapnews: <http://www.yonhapnews.co.kr/bulletin/2016/04/25/0200000000AKR20160425165600017.HTML?input=1195m>
- [2] <http://www.ittoday.co.kr/news/articleView.html?idxno=69792>
- [3] Robert Wells, "Unity 2020 By Example: A project-based guide to building 2D, 3D, augmented reality, and virtual reality games from scratch, 3rd edition, Packt Publishing, September 30, 2020
- [4] Jared Halpern, "Developing 2D Games with Unity: Independent Game Programming with C#", Apress; 1st ed. edition, November 29, 2018
- [5] Unity3D: <https://unity3d.com/kr>
- [6] Android Studio: <https://developer.android.com/studio/index.html?hl=ko>
- [7] Gyroscope: <https://ko.wikipedia.org/wiki/%EC%9E%90%EC%9D%B4%EB%A1%9C%EC%8A%A4%EC%BD%94%ED%94%84>

- [8] Kisung Jeong, "A Study on Virtual Reality Techniques for Immersive Traditional Fairy Tale Contents Production", Korea Computer Graphics Society Vol. 22, No.3, P43-52, 2016.
- [9] Jason Jerald, "The VR Book: Human-Centered Design for Virtual Reality", Morgan & Claypool Publishers, October 16, 2015.
- [10] Animation: <https://docs.unity3d.com/kr/current/Manual/class-Animation.html>
- [11] Prefabs: <https://docs.unity3d.com/kr/current/Manual/InstantiatingPrefabs.html>
- [12] NGUI: http://www.tasharen.com/?page_id=140
- [13] Sung Soo Kim, "A development research for User Interface using Unity 3D in real life", Paichai University Master's Thesis
- [14] C. Pearson, Learning NGUI for Unity, Packt Publishing, December (2014).
- [15] Charles Bernardoff, NGUI for Unity, PACKT, 2014

Authors



Joo-Young Choi received the B.S. degrees in Game Engineering from Paichai University, Korea. She is currently a researcher in the Tsis Co.



Seok-Hun Kim received the M.S and Ph.D. degree in Computer Engineering from Hannam University in 2003 and 2006. He is an assistant professor Mobile Media at Suwon Women's University in from 2012 to 2017.

Dr. Kim is currently an assistant professor in the Electronic Commerce at Paichai University. His teaching and research specialties are in the fields Mobile computing, Web-App programming, E-commerce System.



Ho-Young Kwak received the B.S., M.S. and Ph.D. degrees in Computer Science from Hong-Ik University, Korea, in 1983, 1985 and 1990, respectively. Dr. Kwak joined the faculty of the Department of Computer

Engineering at Jeju National University, Jeju, Korea, in 1990. He is currently a Professor in the Department of Computer Engineering, Jeju National University. He is interested in IT-Medical convergence, Healthcare system, IoT and Software system.



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

and 2008. and He worked as a professor at the Department of Game Engineering at Paichai University until 2020. He is now a professor at the Department of Computer Engineering at Jeju National 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.