

A Design and Implementation Mobile Game Based on Kinect Sensor

Won Joo Lee*

Abstract

In this paper, we design and implement a mobile game based on Kinect sensor. This game is a motion recognition maze game based on Kinect sensor using XNA Game Studio. The game consists of three stages. Each maze has different size and clear time limit. A player can move to the next stage only if the player finds the exit within a limited time. However, if the exit is not found within the time limit, the game ends. In addition, two kinds of mini games are included in the game. The first game is a fruit catch game using motion recognition tracking of the Kinect sensor, and player have to pick up a certain number of randomly falling fruits. If a player acquire a certain number of fruits at this time, the movement speed of the player is increased. However, if a player takes a skeleton that appears randomly, the movement speed will decrease. The second game is a Quiz game using the speech recognition function of the Kinect sensor, and a question from random genres of common sense, nonsense, ancient creature, capital, constellation, etc. are issued. If a player correctly answers more than 7 of 10 questions, the player gets useful items to use in finding the maze. This item is a navigator fairy that helps the player to escape the forest.

▶Keywords: Kinect sensor, Color stream, Depth stream, Skeleton stream, XNA Framework,

I. Introduction

현재 IT, 의료, 교육, 예술 등 다양한 분야에서 널리 활용되고 있는 키넥트(Kinect)는 2009년 6월 E3에서 "프로젝트 나탈(Project Natal)"이란 이름으로 처음 발표되었으며, 2010년 공식 명칭인 키넥트로 발표되었다. 키넥트는 카메라 모듈이 장착되어 모션 캡처로 플레이어의 동작을 인식하며, 마이크 모듈로 음성을 인식한다. 2010년 11월 미국에서 17개의 게임을 키넥트와 동시에 출시하였다[1]. 키넥트는 Kinect for XBox 360와 Kinect for Windows로 분류한다. Kinect for XBox 360은 Microsoft XBox 360 게임기에 최적화 되어 있기 때문에 게임 개발 및 연구용으로 사용한다. Kinect for Windows는 Windows 운영체제에 최적화되어 있으며 Kinect for Windows Runtime을 지원하기 때문에 Windows 기반의 응용프로그램 개발에 널리 사용한다.

키넥트의 무한한 확장성에 개발자들이 주목하고 있으며 이

것을 Microsoft에서는 키넥트 효과라고 표현하고 있다. 키넥트의 모션 인식 기능을 활용하여 인간에게 도움을 줄 수 있는 용도로 활용되고 있다. 특히 인체의 관절을 20개로 나눠서 표현하는 스켈레톤(skeleton) 정보를 활용한다면 인간의 특정 부위에 대한 반복적인 자세를 인식하여 게임 캐릭터의 동작과 연동할 수 있는 응용프로그램을 개발할 수 있다. 또한, 재활 의학 분야에서는 키넥트의 관절 인식 기능을 이용하여 사용자의 특정 부위에 대한 반복적인 자세를 취하도록 하여 환자에게 신체적 및 정신적으로 긍정적인 영향을 제공하고 있다. 게임 개발 분야에서는 키넥트 센서로 인지한 인간의 스켈레톤 정보를 활용하여 모바일 게임 캐릭터의 동작과 연동시킴으로써 쉽고 편리한 게임 인터페이스를 제공한다.

이러한 키넥트의 기능을 활용하여 본 논문에서는 XNA Game Studio를 이용하여 키넥트 기반의 동작 인식 모바일 게

• First Author: Won Joo Lee, Corresponding Author: Won Joo Lee
*Won Joo Lee (wonjoo2@inhac.ac.kr), Dept. of Computer Science, Inha Technical College.
• Received: 2017. 08. 07, Revised: 2017. 08. 15, Accepted: 2017. 08. 30.
• This work was supported by INHA TECHNICAL COLLEGE Research Grant.

입을 설계하고 구현한다. 이 게임은 XNA Framework를 기반으로 키넥트 라이브러리 함수와 연동하여 개발한다[2-4]. XNA Framework은 크로스 플랫폼과 멀티 플랫폼 환경을 지원하기 때문에 PC와 XBox 360, Windows Phone용 게임을 개발할 수 있다. 이를 위해 2장에서 키넥트 센서의 구조 및 기능에 대하여 알아본다. 3장에서는 키넥트 기반의 동작 인식 모바일 게임을 설계한다. 4장에서 키넥트 기반의 동작 인식 모바일 게임을 구현한다. 그리고 마지막 5장에서 결론을 맺는다.

II. Preliminaries

1. Kinect Sensor

키넥트 센서는 콘트롤러 없이 이용자의 신체를 이용하여 게임과 엔터테인먼트를 경험할 수 있는 주변기기이다.

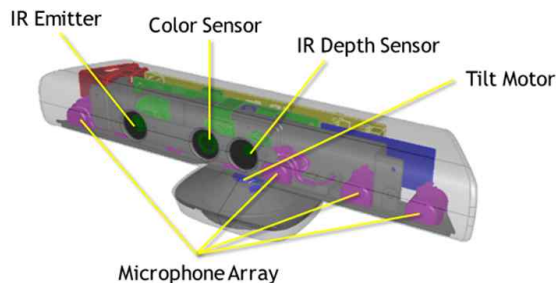


Fig. 1. Kinect Sensor

키넥트 센서는 그림 1과 같이 IR Emitter, Color Sensor, IR Depth Sensor, Tilt Motor와 4개의 Microphone array로 구성된다[5]. IR Emitter와 IR Depth Sensor는 키넥트 뷰 영역내에서 센서와 객체 사이의 거리에 대한 정보를 포함하는 깊이 데이터를 생성한다. Color Sensor는 컬러 스트림을 생성하는 기능을 제공한다. Tilt Motor는 키넥트 센서를 상하로 움직일 수 있는 기능을 제공한다. Microphone array는 음성 인식 기능을 제공한다. 키넥트 센서의 사양은 표 1과 같다.

Table 1. Specification of Kinect sensor

Item	Specification
Viewing angle	· Vertical 43° · Horizontal 57°
Vertical tilt range	· $\pm 27^\circ$
Frame rate (depth & color stream)	· 30 Frames Per Second
Default resolution, depth stream	· VGA (640 x 480)
Default resolution, color stream	· VGA (640 x 480)
Audio format	· 16 KHz, 16 bit Mono PCM
Audio Input characteristics	· 24 bit analog-to-digital converter(ADC)

표 1에서 키넥트 센서는 수직으로 43°, 수평으로 57°의 시야각을 제공하며, Tilt 각은 $\pm 27^\circ$ 이다. 컬러와 웹 스트림의 프레임 30 FPS로 기본 해상도는 VGA(640 x 480)으로 제공된다. 오디오는 16 KHz, 16 bit Mono PCM으로 제공되며, 입력은 24 bit ADC를 사용한다.

2. Kinect Sensor Function

2.1 Color Stream

키넥트는 일반 웹 카메라처럼 영상을 출력하는 기능을 제공한다. 키넥트는 일반 컬러 영상과 함께 각 포인트의 거리정보가 들어간 모노 영상을 통해 물체를 인식하고 동작을 인식 할 수 있다.

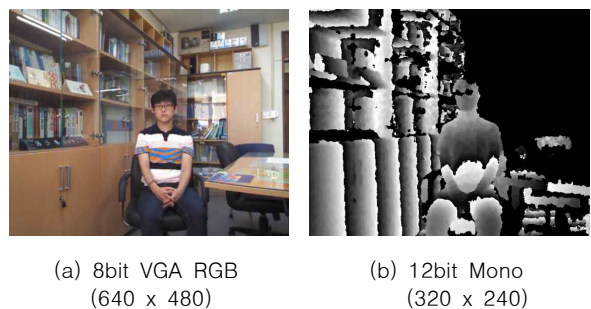


Fig. 2. Color Stream

그림 2의 (a) 8bit VGA RGB 영상은 해상도 640×480, 30FPS로 제공되는 컬러 영상이다. 컬러 영상을 출력하기 위해서는 ColorImageStream 클래스의 지정 필드인 ColorStream을 사용한다.

2.2 Depth Stream

웹스트림은 영상의 모양을 인식하고 크기와 거리정보를 포함하여 프로그램에 응용할 수 있는 정보를 제공한다. 웹스트림은 DepthImageStream 클래스로 지정된 DepthStream 필드를 사용한다[6]. 그림 2의 (b) 12bit Mono 영상은 해상도 320×240, 30FPS로 제공되는 모노 영상이다. 이 영상에서 거리가 멀어 키넥트가 감지하기 어려운 영역은 검은 색으로 표시된다. 또한, 키넥트에 너무 가까워도 검은색으로 표시가 된다. 키넥트가 객체를 인식하는 거리는 0.8~4m 범위로 가장 근접한 인식 거리는 0.8m이다. Kinect for Windows는 근접 모드를 지원하기 때문에 가장 근접한 인식 거리는 0.4m이다. 웹스트림은 픽셀마다 16비트 depth map 정보를 가진다. 16비트 depth map 정보에서 13비트(0~12)는 거리정보를 나타내며, 3비트(13~15)는 플레이어(Player) ID를 나타낸다.

2.3 Skeleton Stream

키넥트 센서는 일반 웹 카메라처럼 영상을 출력하는 기능을 제공한다. 인체의 뼈와 뼈를 연결하는 부분이 조인트(Joint)이다. 각각 관절의 위치가 조인트이고 각 관절의 골격 정보를 스켈레톤(Skeleton)이라 한다. 키넥트는 JointType를 통해 관절 정보를

알 수 있다. 인체에는 20개의 관절의 각 조인트가 존재 한다.

키넥트는 그림 3에서와 같이 기립자세에서는 20개의 조인트를 모두 인식하고, 앉은 자세에서는 10개의 조인트를 인식한다 [6, 7]. 키넥트는 6명의 플레이어를 동시에 인식할 수 있지만 스켈레톤 정보는 2명의 플레이어만 인식하여 제공된다.

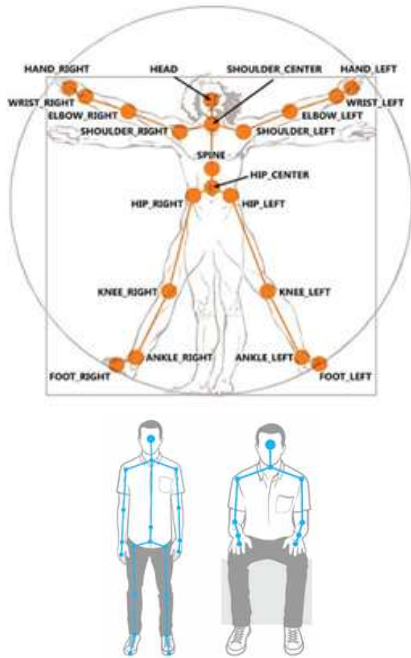


Fig. 3. Skeleton joint

플레이어 스켈레톤 위치는 X, Y, Z 좌표로 표현되며 X, Y, Z 축은 뎀스 센서의 신체 축이다.

키넥트 센서에서 인식한 플레이어의 컬러 영상과 스켈레톤 스트림은 그림 4와 같다.

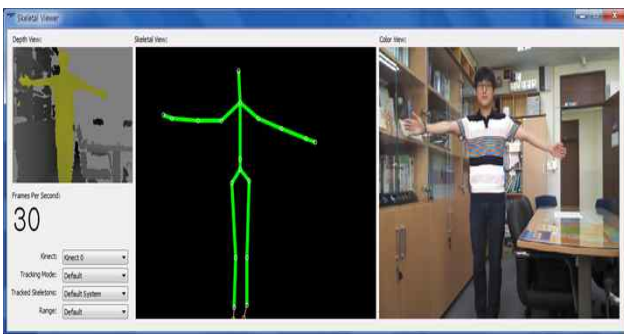


Fig. 4. Skeleton Stream

2.4 Audio

키넥트 센서는 4개의 마이크 어레이에서 좌우 ±50° 범위의 음성을 인식한다.

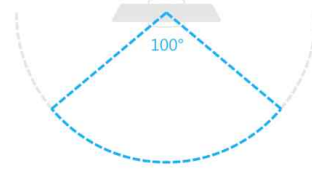


Fig. 5. Audio input range

키넥트 센서는 그림 5의 음성 입력 범위에서 10° 단위로 분할하여 음향의 위치를 추적한다. 키넥트 센서는 24 비트 ADC를 사용하고 음향 반향 취소와 소음 억제를 포함하여 현지 신호 처리를 제공한다.

3. NUI(Natural User Interface) Library

NUI Library는 그림 6과 같이 키넥트 센서와 어플리케이션 간의 인터페이스를 제공한다.

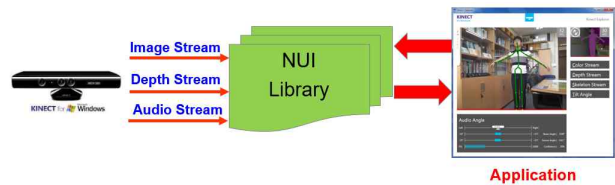


Fig. 6. Hardware and Software interface

그림 6에서 NUI Library는 키넥트 센서에서 전송하는 이미지와 Depth 스트림, 오디오 스트림을 어플리케이션에서 활용할 수 있는 정보로 전환하는 기능을 가지고 있는 핵심 요소이다. 좀 더 세부적으로 들어가면 그림 7과 같은 키넥트 SDK 아키텍처로 정리할 수 있다.

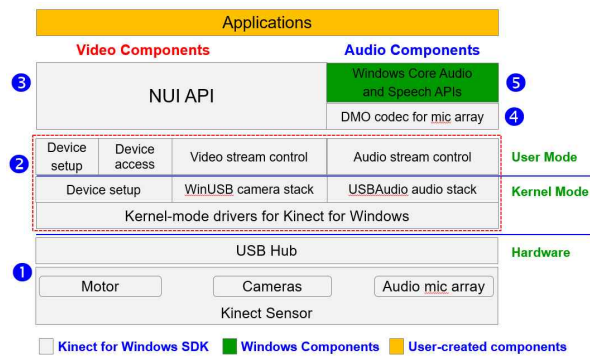


Fig. 7. Kinect SDK Architecture

그림 7에서 키넥트 하드웨어는 USB 기반의 기술로 PC와 연결하는 기능을 제공하는 키넥트 드라이버가 하단에 존재한다. User mode와 Kernel mode에서는 카메라와 오디오 스택이 존재하고, 그 위에 최상위 추상화 레벨에서 NUI API와 오디오/비디오 컴포넌트가 존재하고 있다.

III. Come Back Home Game Design

본 논문에서는 XNA Game Studio[8]를 이용한 키넥트 센서 기반의 동작 인식 미로 게임으로 Come Back Home을 설계한다[9]. 이 게임은 UI를 매우 단순하고 플레이 화면을 모션으로 처리하여 누구나 쉽게 게임을 조작할 수 있도록 설계한다. 각 스테이지 마다 맵의 크기와 제한시간을 다르게 두어 스테이지 마다 난이도를 다르게 설계하고, 그 안에 두 가지 미니게임 요소를 넣어 색다른 게임을 즐길 수 있도록 설계한다. 두 가지 미니게임은 과일잡기와 퀴즈로 플레이어에게 어드벤처지 혹은 페널티를 부여하며 이를 잘 이용하면 더욱 쉽게 클리어 할 수 있도록 네비게이터를 제공하도록 설계한다.

1. Game Story Telling

Come Back Home 게임은 숲에서 엄마와 단 둘이 사는 주인공 루나(Runa)가 숲에 혼자 들어갔다가 길을 잃고 헤매던 중 네비게이터 기능을 하는 요정(Fairy)을 만나 집으로 돌아가는 과정을 각 단계별로 구현한다. 각 단계는 요정의 숲, 노을 진 숲, 잠자는 숲으로 각 숲의 난이도에 따라 다른 요정(Elf) 들의 방해로 이겨내고 집으로 돌아가도록 게임 스토리를 설계한다. 이 게임의 캐릭터는 그림 8과 같다.

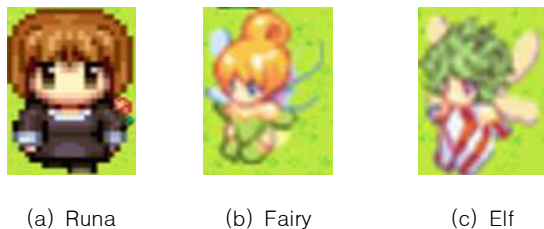


Fig. 8. Game character

그림 8에서 (a)Runa는 게임의 주인공이고, (b)Fairy는 주인공을 도와 숲속을 탈출하도록 도와주는 요정이다. 그리고 (c)Elf는 주인공이 숲속을 탈출하지 못하도록 방해하는 요정이다.



Fig. 9. Game stage

Come Back Home 게임의 스테이지는 그림 9와 같이 요정의 숲(Forest of fairy), 노을 진 숲(Forest of sunset), 잠자는 숲

(Forest of sleeping)으로 구성한다. 요정의 숲은 가장 낮은 수준의 난이도와 탈출 시간을 10분으로 설정한다. 노을 진 숲은 중간 수준의 난이도로 탈출 시간을 20분으로 설정한다. 잠자는 숲은 가장 높은 수준의 난이도로 탈출 시간을 40분으로 설정한다.

2. Mini game

Come Back Home 게임에 두 가지 미니게임인 과일잡기와 퀴즈를 넣어 색다른 게임을 즐길 수 있도록 설계한다. 플레이어가 숲속에서 요정을 만났을 때 요구하는 자세를 취하면 미니게임으로 입장하도록 설계한다.

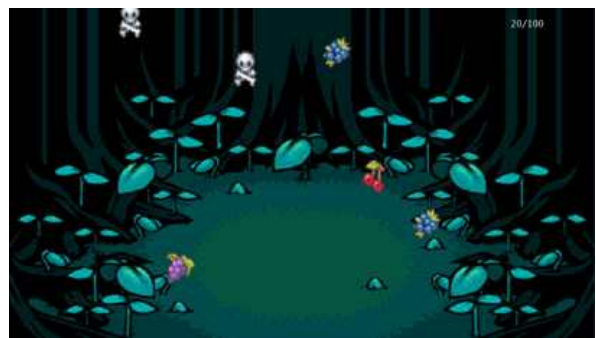


Fig. 10. Mini game of Fruit catch

그림 10의 미니게임 과일잡기는 플레이어의 양손 트랙킹을 이용하여 일정한 수의 과일을 잡으면 플레이어의 이동 속도를 빠르게 하도록 보상하는 기능을 제공한다. 하지만 랜덤하게 나타나는 해골을 잡게 되면 플레이어의 이동 속도를 느리게 하는 페널티를 받도록 설계한다.

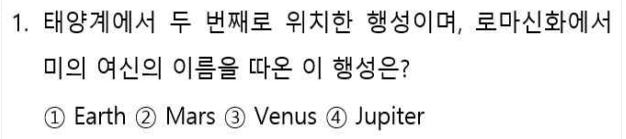


Fig. 11. Mini game of Quiz

미니게임 퀴즈에서는 그림 11과 같이 대화창에 나타난 퀴즈에 대한 답변을 음성으로 인식하도록 설계한다. 퀴즈에 대하여 7개 이상 정답을 맞추면 네비게이터 기능을 하는 요정(Fairy)이 숲속 탈출을 도와주는 보상을 제공한다.

각각 다른 컬러의 이미지를 로드하여 구현한다.

Update()와 Draw() 메서드는 게임이 종료될 때까지 반복적으로 호출된다. Update() 메서드는 게임 로직에 관련된 기능을 구현한다. Update() 메서드에 구현한 사용자 정의 메서드와 그 기능은 표 3과 같다.

Table 3. User define method of Update()

Method	Function
OpeningUpdate()	update of opening stage
TitleUpdate()	update of Title stage
TodoUpdate()	update of control manual stage
PrologueUpdate()	update of Prologue stage
SelectUpdate()	update of Select stage
InitializeStage()	Initialize of each stage by common value
InitializeRestart()	Initialize of coordinate value at restart game
GameUpdate()	implementation of each stage
MiniUpdate()	implementation of mini game (fruit catch, quiz)
OverUpdate	implementation of game exit
EndingUpdate()	update of ending stage
PauseUpdate()	update of pause stage

Draw() 메서드는 게임 배경 및 관련 캐릭터를 화면에 그리는 기능을 지원한다. Draw() 메서드에 구현한 사용자 정의 메서드와 그 기능은 표 4와 같다.

Come Back Home 게임의 초기화면은 그림 15와 같다. 초기화면에서는 게임 시작인 Game start와 게임 방법을 설명하는 How to play, 게임 종료인 Exit을 선택한다.

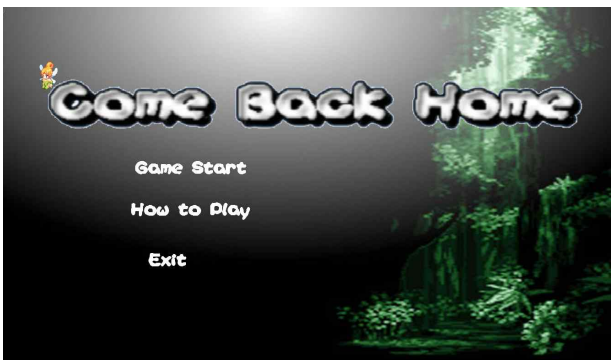


Fig. 15. Execution screen

Table 4. User define method of Draw()

Method	Function
DrawOpening()	draw of opening stage
DrawTitle()	draw of Title stage
DrawTodo()	draw of control manual stage
DrawPrologue()	draw of Prologue stage
DrawSelect()	draw of Select stage
DrawGame()	draw of game stage and time
DrawMini()	draw of mini game (fruit catch, quiz)
DrawOver()	draw of game exit
DrawEnding()	draw of ending stage
DrawPause()	draw of pause stage
nui_ColorFrameReady()	implementation of Color stream
nui_DepthFrameReady()	implementation of Depth stream
nui_AllFramesReady()	implementation of Skeleton stream
PlayerCommand()	classify motion using joint
Kinect_Click()	implementation of Click event
UpdateCellAnimation()	implementation of player image cell animation
UpdateCellpani()	implementation of Fairy image cell animation
sre_SpeechRecognized()	implementation of Recognized speech
UserFunc()	initialized audio stream

게임 방법을 설명하는 그림 16의 How to play screen에서는 키넥트 센서에서 인식하는 플레이어의 동작에 따라 게임 캐릭터의 이동 방향을 결정한다.

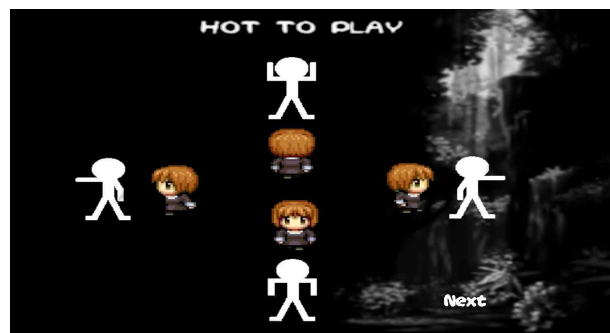


Fig. 16. How to play screen

그림 15에서 Game Start를 선택하면 3개 스테이지중에 첫

번째 스테이지인 그림 17 요정의 숲이 나타난다.



Fig. 17. Forest of fairy screen

그림 17에서 타이머 게이지는 Forest of fairy 스테이지 진입부터 현재까지의 진행 시간을 나타낸다. 플레이어(Runa)는 키넥트 센서에서 인식한 그림 18의 플레이어 동작에 따라 상/하/좌/우로 이동한다.



Fig. 18. Recognized player motion using Kinect sensor

그림 18와 같이 키넥트 센서는 플레이어의 동작을 인식한다. nui_DepthFrameReady() 메서드에서 영상의 거리정보를 추출하고, nui_AllFramesReady() 메서드를 이용하여 사용자와 관절 정보를 추출한다. Kinect_Click() 메서드로 캐릭터의 오른손과 왼손을 마우스처럼 사용할 수 있는 기능을 구현하고, 플레이어를 조정한다.

플레이어는 이동 중에 두 종류의 미니게임인 fruit catch와 quiz를 수행한다. 첫 번째 미니게임은 fruit catch 게임이다.

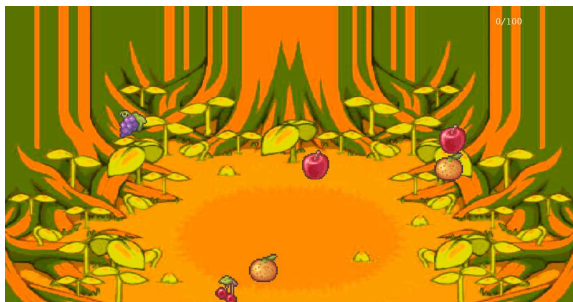


Fig. 19. Fruit catch mini game

그림 19 fruit catch에서는 키넥트 센서의 동작인식 트래킹 기능을 이용하여 떨어지는 과일을 정해진 숫자만큼 획득한다. 이때 일정한 수의 과일을 획득하면 플레이어의 이동 속도가 빨라진다. 하지만 랜덤하게 나타나는 해골을 잡게 되면 플레이어의 이동 속도를 느려진다.

두 번째 미니게임은 Quiz 게임이다. Quiz에서는 키넥트 센서의 스피치 음성 인식을 이용한 퀴즈 게임으로 상식, 년센스, 고대생물, 수도, 별자리 등의 다양한 장르의 문제를 무작위로 출제한다. 이때 sre_SpeechRecognized() 메서드에서는 키넥트 센서의 음성 인식 기능을 사용하여 플레이어의 음성을 인식하여 퀴즈 문제를 해결하는 기능을 구현한다.



Fig. 20. Quiz mini game

그림 20 Quiz에서는 플레이어가 총 10개의 문제 중 7개 이상 정답을 맞추면 미로 찾기에서 사용할 수 있는 유용한 아이템을 획득할 수 있다. 이 아이템은 네비게이터 기능을 하는 요정(Fairy)으로 그림 21와 같이 플레이어의 숲 속 탈출을 도와준다.



Fig. 21. Exit screen

플레이어는 제한된 시간 내에 출구를 찾으면 다음 스테이지인 노을 진 숲(Forest of sunset)과 잠자는 숲(Forest of sleeping)으로 순차적으로 진행한다. 하지만 제한시간 내에 출구를 찾지 못하면 게임이 종료된다.

V. Conclusions

본 논문에서는 XNA Game Studio를 이용한 Kinect 센서 기반의 동작 인식 미로 게임 Come Back Home을 설계하고 구현한다. 이 게임은 주인공 루나(Runa)가 숲에 혼자 들어갔다 길을 잃고 헤매던 중 네비게이터 기능을 하는 요정을 만나 집으로 돌아가는 과정을 각 스테이지 별로 구현한다. 각 스테이지는 요정의 숲, 노을 진 숲, 잠자는 숲으로 각 숲의 난이도에 따라 다른 요정(Elf) 들의 방해물을 이겨내고 집으로 돌아가도록 구현한다. 각 스테이지에서는 키넥트 센서에서 인식한 플레이어의 동작에 따라 상/하/좌/우로 이동한다. 키넥트 센서는 영상의 거리정보를 추출하고, 관절 정보를 이용하여 플레이어의 동작을 인식한다. 또한, 각 스테이지에서는 두 종류의 미니 게임을 구현한다. 첫 번째 미니게임은 동작인식 트래킹을 이용한 과일 줍기 게임으로 무작위로 떨어지는 과일을 정해진 수만큼 주워야 한다. 이때 일정한 수의 과일을 획득하면 플레이어의 이동 속도가 빨라진다. 하지만 랜덤하게 나타나는 해골을 잡게 되면 플레이어의 이동 속도를 느려진다. 두 번째 미니 게임은 스피치 음성 인식을 이용한 퀴즈 게임으로 무작위로 출제되는 다양한 장르의 문제의 정답을 찾는다. 이때 플레이어가 총 10개의 문제 중 7개 이상 정답을 맞추면 미로 찾기에서 사용할 수 있는 유용한 아이템을 획득한다. 이 아이템은 네비게이터 기능을 하는 요정(Fairy)으로 플레이어의 숲 속 탈출을 도와준다.

본 논문의 결과는 향후 키넥트를 이용한 모바일 게임 개발 분야에 기반 기술로 활용될 수 있다.

REFERENCES

- [1] <https://ko.wikipedia.org/wiki/%ED%82%A4%EB%84%A5%ED%8A%B8>
- [2] W. J. Lee, J. Y. Kang, M. H. Park, I. B. Ryu, Y. R. Yoo, M. R. Kim, "A Design and Implementation of Mobile Game Based on Windows Phone 7," In Proceeding of KSCI 2013 Winter Conference(ISSN 2005-1344), Vol. 21, No. 1, pp. 59-60, Jan. 2013.
- [3] <http://msdn.microsoft.com/en-us/centrum-xna.aspx>
- [4] <http://blogs.msdn.com/b/xna/>
- [5] <https://www.codeproject.com/Articles/528305/GetplustheplusRplusStreamplusandpluscontrolplusth>
- [6] J. K. Ko, "Kinect Programing," Korea Electronics Association , 2012.
- [7] <https://channel9.msdn.com/coding4fun/kinect/Tracking-skeleton-joins-code-sample>
- [8] S.H. Lee, D. H. Kim, E. D. Kim, J. C. Lim, B. K. Jung, "Windows Phone 7 Game Programming using XNA," BJPublic, 2011.
- [9] W. J. Lee, Y. H. Jeon, S. M. Han, B. S. Jung, E. J. Kim, K. T. Kim, "A Design and Implementation of Come Back Home Game Based on Kinect Sensor," In Proceeding of KSCI 2016 Winter Conference(ISSN 2005-1344), Vol. 24, No. 1, pp. 49-50, Jan. 2016.
- [10] J. Y. Park, "A Design and Implementation of Mobile Game Based on Windows Phone Platform," Journal of The Korea Society of Computer and Information, Vol. 19, No. 10, pp. 91-97, Oct. 2014.

Authors



Won Joo Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanyang University, Korea, in 1989, 1991 and 2004, respectively. Dr. Lee joined the faculty of the Department of Computer Science at Inha Technical College, Incheon, Korea, in 2008, where he has served as the Director of the Department of Computer Science. He is currently a Professor in the Department of Computer Science, Inha Technical College. He has also served as the Vice-president of The Korean Society of Computer Information and the Editor-in-Chief for the Journal of The Korean Society of Computer Information. He is interested in parallel computing, internet and mobile computing, and cloud computing.