

## Unity 3D를 활용한 1인 모바일 캐주얼 게임 제작

정서원\*, 김진모\*\*

### 요약

2007년 이후 도입된 스마트폰의 영향으로 모바일 콘텐츠 제작이 활발해지고 이러한 추세와 더불어 다양한 연령층에서 모바일 게임 제작에 대한 관심이 높아지고 있다. 과거의 게임 개발은 전문적 지식을 갖춘 개발자만의 영역이었다면 최근 들어 에디터 형태의 엔진들이 제공되어 게임 제작의 높은 진입장벽이 낮아지는 추세이다. 본 논문은 에디터 형태의 게임엔진 중 하나인 Unity 3D를 활용하여 일반 사용자의 관점에서 혼자 게임을 제작하기 위한 방법으로 이벤트기반 모듈 기획 방법을 제안하고자 한다. 이는 1인 게임 제작의 관점을 기획에 맞추어 게임의 전체적인 진행과정에서 동작하는 행동과 역할들을 모듈 단위로 기획하는 방법이다. 그리고 각 모듈은 게임성을 표현하는 다양한 이벤트들을 포함한다. 또한 Unity 3D에서 제공하는 스크립트 기능은 제안하는 모듈 구조를 구현하기에 적합하여 Unity 3D를 활용한다. 마지막으로 제안하는 방법을 통해 기획에서 개발까지 효과적인 게임 제작이 가능한지 여부를 확인하기 위해 간단한 3차원 모바일 캐주얼 게임을 제작하도록 한다.

키워드 : 캐주얼 게임, 이벤트기반 모듈 기획, 1인 게임 제작, 유니티 3D 엔진

## One-man Mobile Casual Game Production Using Unity 3D

Seo-Won Jung\*, Jin-Mo Kim\*\*

### Abstract

Due to smart phones introduced since 2007, mobile contents production become activated and with this trend, attention among different age groups to production of mobile games has been growing. Whereas in the past, development of games was the domain of developers equipped with technical knowledge, provision of editor-type engines in recent times is lowering the high barrier of entry to game production. This paper proposes an event-based module design method from the perspective of general users, aimed at producing games by oneself with a Unity 3D, one of editor-type game engines. This is to plan behaviors and roles in the unit of modules in the whole process of a game to be in line with the perspective of game production by one person. Each module includes diverse events that express game characteristics. In addition, the script function provided by the Unity 3D is appropriate to embody the proposed module structure and utilizes the Unity 3D. Lastly, this study produce a simple 3D mobile casual game in order to verify whether effective game production from the planning to the development is possible through the proposed method.

Keywords : Casual Game, Event-based Module Design, One-man Game Production, Unity 3D Engine

### 1. 서론

※ 교신저자(Corresponding Author): Jin-Mo Kim  
접수일: 2014년 07월 24일, 수정일: 2014년 08월 27일  
완료일: 2014년 08월 28일  
\* 동국대학교 멀티미디어학과  
Tel: +82-10-9191-1523  
email: jungseowon@dongguk.edu  
\*\* 부산가톨릭대학교 소프트웨어학과  
Tel: +82-51-510-0645, Fax: +82-51-510-0648  
email: jmkim11@cup.ac.kr

게임은 국내 뿐 아니라 해외에서도 단순한 오락의 범주가 아닌 산업으로 크게 발전하여 누구나 즐기는 문화의 하나로 자리매김하였다. 과거에는 콘솔이나 PC기반 게임이 중심으로 이루어던 게임 시장에서 차츰 모바일 및 디지털 다운로드 게임 시장의 성장을 토대로 성장세를 이어가고

있다. 최근 들어 소셜 네트워크 서비스 시장에 대한 관심이 증대하고 관련 콘텐츠들이 개발되고 있는 것과 더불어 이를 활용한 모바일 게임 시장이 빠르게 성장하고 있다. 또한 스마트폰과 태블릿 PC 보급 확대는 성장하고 있는 모바일 게임시장의 매출 규모를 향후 더욱 증가할 전망이다. 모바일 게임 시장의 확대에 인하여 게임 사용자층의 연령대가 점차 높아지고 다양해지고 있으며 여성 게임 사용자 비율도 상당히 증가한 것으로 나타나고 있을 만큼 다양한 장르와 플랫폼의 게임시장이 더욱 그 영역을 확장해 나갈 것이다[1].

이렇듯 콘텐츠 산업의 꾸준한 발전은 3차원 콘텐츠에 대한 수요를 꾸준히 증가시켜왔다. 이에 따라 콘텐츠의 제작 및 활용에 대한 관심이 증대되고 있으며 이는 전문 제작자 및 개발자뿐 아니라 전문성이 없는 일반 사용자들도 제작에 대한 관심이 확산되는 추세이다. 하지만 전문적 지식을 기반으로 응용 소프트웨어나 개발도구를 활용하여 콘텐츠를 개발하는 기존의 제작 과정은 개발 프로세스, 프로그래밍 기술 등과 같은 여러 전문적 지식을 학습한 숙련된 전문가 이외에 일반 사용자에게는 제작의 어려움이 따르며 접근성 또한 높지 않다[2].

게임 제작과 관련하여 개발에 필요한 기능들을 모아 효과적으로 사용할 수 있도록 제공함으로써 개발의 단가를 줄이고 복잡한 제작 과정을 단순화시켜주어 정해진 일정에 고품질의 게임을 상용화할 수 있도록 도와주는 다양한 게임 엔진이 개발되었다. 과거에는 프로그래머를 비롯한 개발자의 관점에서 저 수준 언어 기반의 게임 엔진들이 개발되어 제공되었다면 최근 들어서는 전문적인 지식이 부족해도 저작도구의 사용방법만 학습하면 간단한 기능의 게임 정도는 쉽고 빠르게 제작하면서 동시에 품질 또한 일정 수준 이상은 보장할 수 있는 통합 에디터 형 엔진이 개발되고 있다, 이러한 통합 에디터 형 엔진은 그래픽, 사운드, 물리, AI등 게임에 필요한 다양한 기능들을 쉽게 개발하도록 도와준다. 대표적 소프트웨어로 Unity 3D 엔진이 있다. Unity 3D는 3차원 비디오 게임이나 건축 시각화, 실시간 3차원 애니메이션과 같은 대화형 콘텐츠를 제작하기 위한 통합 미들웨어로 윈도우나 맥, 안드로이드, Wii 등 멀티 플랫폼을 지원하며 렌더링 엔

진이나 물리 엔진 등을 자체적으로 탑재하고 있다. 또한 에셋스토어 개념을 도입하여 에디터 내에 필요한 이미지, 사운드, 그래픽 데이터, 스크립트 파일과 같은 리소스들을 다운로드하여 사용할 수 있도록 함으로써 게임 제작 과정을 단순화시켜준다[3].

본 연구는 모바일 게임시장이 발전하고 게임 엔진이 진화하는 흐름과 더불어 전문적인 프로그래밍 개발 능력을 갖춘 개발자의 관점이 아닌 간단한 스크립트 편집 기능을 이해하고 있는 기획이나 그래픽 디자이너의 입장에서 효율적으로 게임을 제작하는데 필요한 기획 설계 방법을 제안한다. 이는 게임 제작에 필요한 화면전환, 움직임 등과 같은 동작과 행동을 모듈로 기획하고 Unity 3D 엔진을 활용하여 효과적으로 구현함으로써 1인이 보다 쉽고 빠르게 간단한 모바일 게임을 제작할 수 있는 방향을 제시하고자 한다. 그리고 제안하는 방법의 효율성을 실제 간단한 3차원 모바일 캐주얼 게임을 제작하여 이해를 돕고자 한다.

## 2. 관련연구

게임 엔진이란 용어의 사용은 1990년대가 처음이지만 게임 엔진 형태의 시스템은 1980년대에도 존재하였다. 시에라의 AGI와 인센티브 소프트웨어의 프리스케이프 엔진 등이 그 예이다. 하지만 1980년대 엔진들은 최근의 게임 엔진들과 달리 개발사외에 다른 회사의 게임에 사용된 적은 없었다. 반면에 게임 엔진이란 용어가 본격적으로 사용된 1990년 중반의 ID소프트웨어의 1인칭 슈팅 게임인 둠과 퀘이크가 주목을 받으면서 다른 개발사들이 관련 소프트웨어의 일부를 라이선스를 받아 그래픽, 사운드 등을 보완하여 게임을 만들기 시작하였다. 이후 퀘이크 3 아레나나 에픽 게임즈의 1인칭 슈팅 게임인 언리얼은 처음 개발부터 엔진과 콘텐츠를 따로 설계하여 개발되었다. 그리고 기술에 대한 이용 허가를 내주어 별도의 수익을 얻으면서 게임 엔진의 상용화를 시작하였다. 이러한 흐름들이 게임 엔진 발전에 중요한 역할을 하였고 상용 목적의 엔진뿐 아니라 재사용성을 염두에 두고 게임을 설계하였다. 게임 엔진들은 현재 가장 복잡한 응용

소프트웨어의 하나로 발전하였고 범용성을 갖추며 다양한 장르에 특화된 엔진들이 등장하기 시작하였다[4-7].

게임 엔진과 관련한 국내 연구의 경우 2000년대 초반에는 게임 엔진 개발과 관련한 핵심적 기술에 대한 연구들이 이루어졌다. 국외 기술에 비해 상대적으로 열세에 놓인 국내 게임 개발 업체에 게임 엔진기술을 제공하여 경쟁력을 확보함을 목적으로 관련 핵심기술들을 연구하였지만 꾸준한 기술 연구를 지원하는 환경이 이루어지지 않아 연구가 지속적으로 이루어지지 못하였다. 이후에는 공개 엔진이나 상용 엔진을 활용하여 실제로 게임이나 응용 소프트웨어를 개발하는 방향으로 연구가 진행되었다. 공개 렌더링 엔진인 Ogre3D를 활용하여 기능성 게임이나 RPG 게임의 프로토타입을 제작하거나 오픈소스 물리엔진을 이용하여 천장 크레인 시뮬레이터를 개발하는 등 공개엔진을 활용하는 방향으로 관심이 집중되기도 하였다. 최근에는 콘텐츠의 다양성이나 내용에 관심을 집중하여 음악 및 영어 회화 학습 교육 게임 제작과 같은 질을 높일 수 있는 분야로 연구들이 진행되고 있다[8-14]. 이 밖에도 Unity 3D 에디터를 활용하여 교육용 게임을 제작하는 2차적인 저작 도구를 개발하는 연구들이 진행되기도 하였다[15]. 저 수준 언어 기반의 공개 엔진들과 관련하여서는 개발자적 관점에서 일부 학문적 연구들이 진행되기도 하였지만 통합 에디터 형태의 게임 엔진들이 등장하고 나서부터는 매뉴얼에 대한 이해만 있어도 게임을 제작할 수 있는 환경이 제공되고 있다. 따라서 이와 관련한 학문적 연구보다는 실제 엔진을 활용한 제작사례들에 대한 연구들이 전부인 상황이다. 오히려 이러한 게임 엔진들이 갖는 특징을 기반으로 일반 사용자가 게임을 제작하기 위해서는 기획 단계에서의 이벤트, 동작 등에 대한 체계적인 정리가 중요하다고 할 수 있다.

본 연구는 게임 엔진이 가지는 이점을 최대한 활용하여 전문적인 지식을 갖고 있는 프로그램 개발자가 아닌 기획이나 그래픽 디자이너의 관점에서 보다 쉽고 빠르게 3차원 모바일 캐주얼 게임을 제작할 수 있는 개발 과정을 기획의 관점에서 정리함으로써 1인 개발 환경 아래서 품질 높은 게임을 효과적으로 제작하는데 도움을 주고자 한다.

### 3. 모듈 기획을 통한 게임 제작

최근에는 쉽고, 편리하게 게임을 제작할 수 있는 다양한 개발 도구들이 제작되고 있다. 이러한 도구들은 점차 사용자에게 전문적인 지식을 요구하기 보다는 도구의 사용법만 친숙해지면 간편하게 원하는 게임을 제작할 수 있는 편의를 제공함을 목표로 하고 있다. 하지만 그렇다고 해서 누구나 쉽게 게임을 제작할 수 있는 것은 아니다. 게임 제작에 필요한 과정을 이해하고 내가 원하는 게임을 제작하기 위해서 요구되는 기본적인 요소들을 학습하며 이를 체계적으로 관리해야만 불필요한 과정 없이 효과적으로 원하는 게임을 제작할 수 있을 것이다.

본 연구는 여러 게임 엔진 가운데 Unity 3D 엔진이 제공하는 최적화 방식과 셰이더 등의 편리한 기능들을 이용하여 DirectX SDK와 같은 저수준 언어를 이용한 개발 방식이 아닌 고수준의 방식으로 게임을 개발하는 것을 토대로 혼자서 상대적으로 쉽고 효과적으로 게임을 제작할 수 있는 기획 방법을 정리하고자 한다[16].

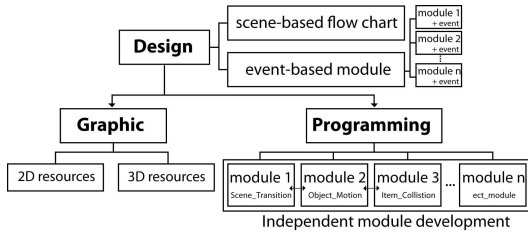
#### 3.1 게임 제작 개요

일반적인 게임 제작 과정은 크게 기획, 그래픽 디자인, 프로그래밍 개발의 3가지 단계로 구성된다. 1인 게임 제작 또한 이러한 구성을 기반으로 제작이 된다. 단, 1인 게임 제작은 각 단계별 역할과 행동들을 모두 전문적으로 파악하기 힘들다는 것을 전제로 이벤트를 포함한 모듈 단위의 구체적인 기획 단계를 중심으로 게임 제작 과정을 정리하는 것이 본 연구의 목표이다.

본 연구에서 제안하는 이벤트기반 모듈 기획은 게임의 전체적인 흐름을 나타내는 플로우 차트와 더불어 게임 과정에서 동작하는 행동과 역할을 모듈 단위로 정리하는 것으로 이 과정에서 발생하는 다양한 이벤트들을 포함한다. 그리고 이를 시각적으로 표현하여 정리하면 그래픽 디자인에서 필요한 2차원, 3차원 리소스들과 프로그래밍 개발에서 필요한 업무가 정리될 수 있다. (그림 1)은 제안하는 1인 게임 제작의 개발 과정의 개요를 정리한 것이다. Unity 3D 엔진의 스크립트 기능은 모듈 단위로 기획된 행동들을 구

현하기에 적합한 구조를 갖고 있어 1인 게임 제작에 이점을 갖는다.

(그림 1) 제안하는 개발 과정의 개요

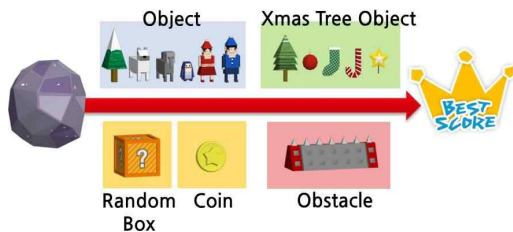


(Figure 1) Overview of proposed development process

### 3.2 이벤트기반 모듈 기획

기획의 첫 번째 단계는 게임의 목표 설정이다. 게임 진행에 구체적인 행동과 역할을 정의하기 위해서는 게임의 목표가 무엇인지 그리고 게임의 목표를 수행하기 위해 핵심적인 과정이 어떻게 되는지를 결정해야 한다. 본 논문에서 제작하는 캐주얼 게임은 “공을 굴려 장애물을 피하고 아이템들을 얻어 높은 점수를 얻는다” 이다. 그리고 직관적인 이해를 돕기 위해 추상적인 용어를 시각적으로 표현하여 나타낸다. (그림 2)는 게임의 목표를 그림으로 간단하게 표현하여 나타낸 것이다. 현재는 구현된 결과물의 이미지들이 존재하기 때문에 이를 활용하여 개요를 구성한 것이고, 제작 초기 단계에서는 좀 더 가벼운 스케치 이미지가 될 것이다.

(그림 2) 제안 게임의 개요 구성의 예

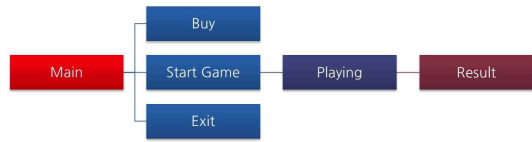


(Figure 2) An example of overview construction of proposed game

다음은 플로우 차트이다. 이는 게임의 각 장면을 어떻게 구성할 것인지를 결정하는 과정으로 게임의 전체적인 흐름을 알 수 있도록 간략히 정리한다. (그림 3)은 각 장면의 목적을 플로우

차트로 나타낸 것이다.

(그림 3) 플로우 차트의 예



(Figure 3) An example of flow chart

설계한 플로우 차트를 토대로 각각의 장면을 위의 목표와 같이 스케치로 작성하여 나타내면 보다 직관적인 이해를 도울 수 있을 것이다. (그림 4)는 이를 나타낸 것이다.

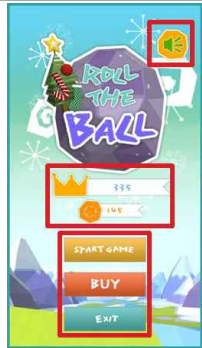


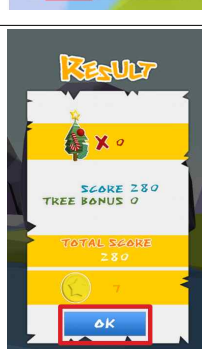
(그림 4)와 같이 결정된 장면의 흐름을 토대로 각 장면에서의 행동을 보다 구체적으로 정리하면 <표 1> 과 같이 나타낼 수 있다. 각 장면에 대한 세부적인 기획이 정리되면 2차원 그래픽 디자인 작업에서 필요한 리소스들도 자연스럽게 결정된다.

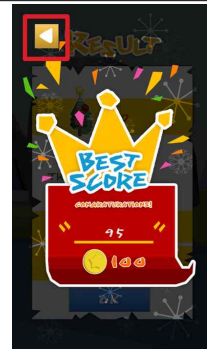
(그림 4) 장면 리스트



(Figure 4) Scene list

<표 1> 각 장면의 구성

Scene image	Contents
	Main scene 1. BGM Sound on/off 2. Best Score & Coin 3. Menu <ul style="list-style-type: none"> <li>Start game : move game scene</li> <li>Exit : move exit scene</li> </ul>
	Exit scene <ul style="list-style-type: none"> <li>OK : Finish</li> <li>NO : move main scene</li> </ul>
	Game scene 1. Main UI <ul style="list-style-type: none"> <li>S : current score</li> <li>C : current coin</li> <li>Xmas Tree Elements : the current number of elements for Xmas tree</li> <li>TimeBar(HP) : current HP</li> </ul> 2. BGM Sound on/off
	Result scene <ul style="list-style-type: none"> <li>The number of Xmas Tree</li> <li>Score : get objects</li> <li>Tree Bonus : The number of Xmas tree * 500</li> <li>Coin : get coins</li> <li>Ok : move</li> </ul>

	Best score scene <ul style="list-style-type: none"> <li>Back : move result scene</li> </ul>
--	---

<Table 1> Organization of each scene

게임의 목표와 플로우 차트는 기획 단계에서 일반적으로 진행되는 작업이다. 모듈 기획은 보다 뚜렷한 목표를 갖고 단계별로 주어진 미션을 수행하기 위해 필요한 구체적인 행동과 역할을 기획단계에서 결정하는 것이다. 그리고 모듈 단위로 행동하는 가운데 발생하는 게임의 다양한 이벤트들과 모듈이 실행하는데 필요한 기능이 존재한다면 이를 추가적으로 정리한다. 모듈 기획의 핵심은 실제 게임 제작의 구현 과정에서 보다 효율적으로 활용하기 위한 기획 단계로 단순히 플로우 차트 형식의 정리나 밸런스나 캐릭터 중심의 시스템 기획이 아닌 보다 구체적이며 직관적인 구조로 개발의 전체 과정을 이해할 수 있도록 정리하는 것이 중요하다. (그림 2)는 본 논문에서 제안하는 모바일 게임에 적용되는 이벤트기반 모듈을 정리한 예이다.

화면 전환 모듈의 세부적인 장면과 행동은 <표 1>에서 정의하였다. 물체 움직임과 아이템 충돌 모듈은 본 논문에서 제작하는 게임의 핵심으로 제작하고자 하는 게임이 갖고 있는 중요한 이벤트를 시각적으로 정리하여 표현할 수 있다면 이후 디자인 작업에서의 기획 설계는 물론 개발단계에서 필요한 업무를 제작 기획 측면에서 명확하게 정리할 수 있을 것이다. (그림 6)은 본 논문에서 제작하는 게임의 핵심 모듈인 아이템 충돌의 이벤트를 표현한 예이다. 이와 같이 정리가 되면 제작해야할 3차원 그래픽 리소스들도 자연스럽게 결정된다. 더욱 중요한 것은 모듈 단위의 요소들은 Unity 3D엔진에서 제공하는 스크립트를 사용하면 개별적인 구현이 가능하며 독립적으로 작동하기 때문에 모듈 각각을 관리하고 재사용할 수 있다는 장점이 있다는 것이다.

(그림 5) 이벤트기반 모듈의 예

<b>Module - Scene_Transition</b>	
<b>Interface</b>	<ul style="list-style-type: none"> <li>• Input key</li> <li>• Select image</li> </ul>
<b>Event</b>	<ul style="list-style-type: none"> <li>• Move scene</li> <li>• Active sub-menu</li> </ul>
<b>Module - Object_Motion</b>	
<b>Interface - none</b>	
<b>Event</b>	<ul style="list-style-type: none"> <li>• Acceleration - left/right</li> <li>• Add force(physics) - forward, left/right</li> </ul>
<b>Module - Item_Collision</b>	
<b>Interface</b>	<ul style="list-style-type: none"> <li>• Classify objects</li> <li>• Interchange information</li> </ul>
<b>Event</b>	<ul style="list-style-type: none"> <li>• Increase score or coin</li> <li>• Decrease HP</li> <li>• Get item or bonus</li> </ul>
<b>Module - Item_Object(Coin / Score)</b>	
<b>Interface - none</b>	
<b>Event</b>	<ul style="list-style-type: none"> <li>• Manage coin or score</li> </ul>

(Figure 5) Examples of event-based module

(그림 6) 게임 장면 안에서 다양한 이벤트의 예



(Figure 6) Examples of various events in game scene

### 3.3 그래픽 디자인

그래픽 디자인은 2차원 리소스와 3차원 리소스로 나누어 작업한다. 2차원 리소스는 배경과 버튼의 UI 이미지이며, 3차원 리소스는 게임 화면을 구성하는 모든 아이템들이 될 것이다. 이때 본 연구에서 중요한 것은 스스로 원하는 이미지

를 제작할 전문적 지식이 없어 작업을 의뢰할 경우에 필요한 리소스들을 기획 단계에서 체계적으로 정리하여 제공함으로써 의사소통을 원활하게 할 수 있게 함을 목적으로 하는 것이다.

#### 3.3.1 2차원 리소스

기획단계에서 각 화면에 대한 구성 요소를 동작별로 정리하였다면 2차원 리소스는 컨셉에 맞게 제작하기만 하면 된다. 주의해야 할 것은 이미지의 해상도를 일관성있게 유지해 주는 것이 중요하다. (그림 7)은 2차원 리소스들 가운데 배경 이미지를 제작한 결과이다.

(그림 7) 배경 영상의 결과

Background					
No.	1	2	3	4	5
Name	UI_Intro	UI_MainBG	UI_Loading	UI_ExitBG	UI_BuyBG
Image					
No.	6	7	8	9	
Name	UI_StageFrm	UI_ResultBG	UI_BestBG	BlackBG	
Image					

(Figure 7) Result of background images

그리고 버튼과 기능들을 표현하는 UI 영상들을 제작한다. 이 또한 기획 단계에서 계획대로 제작하면 된다. (그림 8)은 이를 나타낸 것이다.

(그림 8) UI 영상들의 결과

Button etc.							
No.	1	2	3	4	5	6	7
Name	UI_Start	UI_Buy	UI_OK	UI_No	UI_Exit	UI_Pause	UI_Back
Image							
No.	8	9	10	11	12	13	14
Name	UI_SoundOff	UI_SoundOn	UI_TimeStar	UI_Box1	UI_Box2	UI_ItemBG	UI_Item
Image							



















(Figure 8) Result of UI images

일반적으로 포토샵이나 일러스트레이터와 같은 그래픽 편집도구를 이용하여 2차원 리소스들을 제작하고 편집도구 사용 능력이 부족할 경우 제작된 리소스들을 원하는 연출에 맞게 웹에서 구하여 사용할 수도 있을 것이다.

#### 3.3.2 3차원 리소스

3차원 리소스는 게임화면을 구성하는 아이템들로 모듈 기획단계에서 (그림 6)처럼 필요한 리소스들이 명확하게 정리되면 바로 제작을 진행하면 된다. (그림 9)는 제작된 예제 게임의 3차원 리소스 결과로 그래픽 저작도구인 3ds Max를 이용하여 제작한 것들이다. 캐주얼 게임은 상대적으로 적은 폴리곤에 간단한 오브젝트가 사용되어 그래픽 작업이 수월할 수 있다.

(그림 9) 3차원 아이템 리소스들의 결과

3D Models						
No.	1	2	3	4	5	6
Name	Ball	Terrain	Girl	Boy	Girl@Walk	Boy@Walk
Image						
No.	7	8	9	10	11	12
Name	Tree	Penguin	Bear	Elephant	Obstacle	Coin
Image						
No.	13	14	15	16	17	18
Name	XMasTree	TreeBall	Socks	J	Star	ItemBox
Image						

(Figure 9) Result of 3D item resources

그래픽 디자이너의 관점에서라면 위의 과정을 직접 제작하여 사용할 수 있지만 기획자의 입장이라면 직접 제작은 불가능할 수 있다. 다만 그래픽 디자인 작업을 이해하고 필요한 그래픽 리소스들을 어떻게 정리하고 관리해야 작업 시간을 단축할 수 있으며 원하는 리소스들을 빠르게 구하여 효과적으로 게임을 제작할 수 있는지를 전달하기 위함이다. Unity 3D에서 제공하는 에셋스토어의 다양한 리소스들이 존재하기는 하지만 사용자가 원하는 컨셉의 이미지나 모델을 구하는 것이 쉽지 않기 때문에 그래픽 작업은 불가피하게 직접 제작해야 하거나 요청해야 되는 문제가 있음을 인지하고 있어야 한다.

### 3.4 프로그래밍 개발

Unity 3D는 자바스크립트와 C#스크립트를 기반으로 게임을 구성하는 객체들의 물리, 충돌 등의 기능들을 직관적인 구조로 제어할 수 있도록 하고 있다. 또한 여러 매뉴얼과 예제 스크립트 및 프로젝트들이 제공되어 일반 사용자가 전문적인 지식 없이도 쉽게 활용할 수 있다. 하지만

실제로 게임을 제작하고자 하면 규모가 작은 캐주얼 게임이라 할지라도 접근이 쉽지 않은 것이 현실이다. 본 논문은 이러한 접근성의 어려움을 해결하기 위한 방법으로 기획에서부터 개발까지의 일련의 과정을 이벤트기반 모듈 기획 방법이라는 제작을 고려한 기획의 전체적인 구조에서 개발 과정을 설계하고 구현함으로써 게임을 완성한다.

우선 첫 번째 화면 전환 모듈은 화면 이동과 서브 메뉴를 활성화하는 이벤트로 구성이 되어 있고 이때 필요한 기능은 키 입력과 이미지 선택이다. 키 입력과 같이 다른 모듈에서도 반복적으로 사용될 수 있는 기능은 한번만 정리하도록 한다. 모듈은 어떻게 정리하느냐에 따라 스크립트 단위로 분류되거나 더 작게는 함수 단위로 정의될 수 있다. 스크립트의 독립적인 특징과 재사용성을 고려하였을 때 모듈은 스크립트 단위로 정의하는 것이 효율적이다. 먼저 이벤트와 기능을 포함하는 모듈을 작성하고 이벤트와 기능 구현에 필요한 메소드들을 찾아 정리한다.

Unity 3D에서 제공하는 메소드와 필드들은 매뉴얼에 자세히 정리가 되어있기 때문에 이에 대한 구체적인 설명보다는 많은 양의 전문화된 자료에서 필요한 메소드들을 효율적으로 탐색하고 관리할 수 있는지에 초점을 맞추어 이벤트기반 모듈 단위의 기획을 설계하는 것이다.

<b>Module</b>
<b>Scene_Transition.js(or .cs)</b> function Update() { ② <b>Interface</b> <b>If</b> 마우스 입력 and GUITexture 선택 <b>Then</b> ① <b>Event</b> ( 다음 화면 이동 or 서브 메뉴 활성화 ) <b>End If</b> GUIText 출력 }
<b>Event</b>
<ul style="list-style-type: none"> <li>• Application.LoadLevel("scene_name") : 화면 이동</li> <li>• GameObject.SetActive(true) : 오브젝트(서브 메뉴 포함) 활성화</li> <li>• guiText.text = "문자열" + score : GUIText 출력</li> </ul>
<b>Interface</b>
<ul style="list-style-type: none"> <li>• Input.GetMouseButtonDown(index) : 마우스 입력</li> </ul>

- GUITexture.HitTest(new Vector3(Input.mousePosition.x, Input.mousePosition.y, 0)) : GUITexture 선택

두 번째는 물체 움직임 모듈로 이 역시 다음과 같이 정리할 수 있다.

<b>Module</b> <b>Object_Motion.js(or .cs)</b> function FixedUpdate() { Vector3 dir Vector3 forward = Vector3(0,0,1) int speed ① <b>모바일 가속 센서(좌/우 움직임)</b> dir.x = Input.acceleration.x ② <b>시간에 따른 속도 증가</b> dir *= Time.deltaTime * speed ③ <b>전방, 좌/우 방향으로 힘 적용</b> ball.rigidbody.AddForce (transform.TransformDirection(forward * speed *(Time.timeScale))) ball.rigidbody.AddForce (transform.TransformDirection(dir)) }  <b>Event</b> <ul style="list-style-type: none"> <li>• Input.acceleration : 가속센서</li> <li>• transform.TransformDirection(direction) : 방향 결정</li> <li>• GameObject.rigidbody.AddForce() : 힘(물리)</li> </ul>
---

물체 움직임 모듈은 화면 전환과는 다르게 논리적인 구조로 접근하는 것이 중요하다. 하나의 메소드로 하나의 이벤트를 만드는 것이 아니라 이벤트들 사이의 인과관계가 존재하기 때문에 이를 논리적인 구조 안에서 단계별로 작성하는 것이 핵심이다. Unity 3D와 같은 게임 엔진을 사용하는 장점 중에 하나는 물체에 적용될 힘만 계산해주면 그 외에 움직임들은 자체적으로 자연스럽게 표현해 준다는 것이다. 이 밖에 부가적으로 물체의 움직임은 매 프레임 별로 호출되면서 일정한 시간 간격차를 두고 한 번씩 호출이 되므로 Unity 3D의 프레임워크 단계 중 FixedUpdate 함수에 구현을 해주면 일정한 움직임을 구현할 수 있다[16]. 또한 물체의 움직임을 따라가는 카메라의 이동은 Unity 3D에서 기본적으로 제공하는 카메라 움직임 스크립트(Smooth Follow.js)를 참고하여 이 역시 하나의 모듈로 활용하면 해결된다.

다음은 아이템과의 충돌 모듈로 이는 각각 독립적인 이벤트들이 하나의 모듈에서 처리되는 구조이며 이 과정에서 모듈 사이의 자원을 공유하는 문제를 효과적으로 처리해야 한다.

<b>Module</b> <b>Item_Collision.js(or .cs)</b> function OnCollisionEnter(Collision coll) { ② <b>Interface : 객체 분류</b> <b>If</b> 충돌 객체가 주인공 여부 <b>Then</b> ① <b>Event</b> ( 점수 증가 or 체력 감소 or 코인 증가 or 아이템 획득 or 보너스 획득 ) ⓐ 해당 모듈을 탐색 ⓑ 이벤트에 대응되는 함수 호출 or 이벤트에 대응되는 변수 수정 <b>End If</b> }  <b>Event</b> <ul style="list-style-type: none"> <li>• GameObject.Find("module_name") : module_name의 객체 탐색</li> <li>• SendMessage("function_name", value) : 탐색한 모듈 내 함수(function_name) 호출</li> <li>• PlayerPrefs.SetInt("key", value); : value 변수를 "key"로 지정</li> <li>• PlayerPrefs.Save() : "key"로 지정된 변수 저장</li> <li>• PlayerPrefs.GetInt("key") : "key"에 대응되는 변수 값 호출</li> </ul> <b>Interface</b> <ul style="list-style-type: none"> <li>• coll.gameObject.tag : tag를 이용한 객체 분류 ex) <b>If</b> coll.gameObject.tag == "Player" <b>Then</b></li> </ul>
---

마지막으로 객체 모듈은 기본적으로 동작하는 구조가 유사하다. 다음은 코인 관리 모듈로 코인을 획득할 때 호출되는 함수를 정의한다.

<b>Module</b> <b>Item_Coin.js(or .cs)</b> int coin function AddCoin(int value) { ① <b>Event</b> - value만큼 coin 증가 }  <b>Event</b> <ul style="list-style-type: none"> <li>• coin = coin + value</li> </ul>
---

아이템 충돌 모듈에서 코인 획득 이벤트를 예로 GameObject.Find("Item\_Coin").SendMessage("AddCoin", 1)과 같이 다른 모듈의 함수를 호출하거나 PlayerPrefs.SetInt("key\_coin", coin)과 같



이 다른 모듈의 변수를 제어할 수 있다.

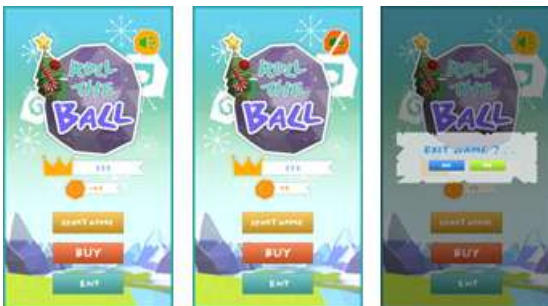
이벤트기반 모듈을 구현하는 과정은 개발자의 관점보다는 시스템을 게임의 핵심 이벤트에 초점을 맞추어 기획하는 것으로 모듈 개발을 풀어갈 수 있도록 하는 것이다. 이외에도 필요한 모듈들이 존재하지만 이 또한 같은 방법으로 해결할 수 있다. 본 연구에서 제안하는 방법은 기획에서부터 개발까지 게임 제작의 일련의 과정을 이벤트를 포함하는 모듈이라는 하나의 구조로 설계함으로써 혼자서 게임을 효과적으로 제작하고 관리할 수 있도록 하는 것이다.

#### 4. 구현 및 분석

본 논문에서 제작한 1인 모바일 캐주얼 게임은 Unity 3D 4.0엔진을 기반으로 제작되었고 그래픽 디자인 작업은 Adobe Photoshop/Illustrator CS5 를 사용하여 2차원 리소스를, Autodesk 3ds Max 2009를 사용하여 3차원 리소스를 제작하였다. 그리고 게임의 플랫폼은 안드로이드 스마트폰을 대상으로 하고 있어 안드로이드 4.1(최소 API Level - Android 4.1 'Jelly Bean'(API level 16))버전으로 빌드하여 실험하였다.

(그림 10)은 게임의 메인화면을 나타낸다. 화면 전환 모듈의 화면 구성을 통해 시작, 상점, 종료의 버튼과 점수, 코인 정보를 보여주고 있다. 현재는 상점 기능에 대한 세부적 사항을 제외하였다.

(그림 10) 우리 게임의 메인 화면



(Figure 10) Main scene of our game

(그림 11)은 물체 움직임과 이 과정에서 아이템과의 충돌 모듈을 실행한 결과 영상의 일부이

다. 게임 플레이는 안드로이드 4.1버전 이상부터 호환이 가능하며 720\*1280의 해상도를 가지고 있다. 본 논문에서 구현한 게임은 실험에 사용한 기기에 최적화되어 있지만 Unity 3D는 화면 비율에 맞게 인터페이스를 설정할 수 있어 사실상 거의 모든 안드로이드 기반 스마트폰에서 구동이 가능하다. 그리고 Unity 3D 엔진에서 제공하는 프리랩 기능을 활용하여 재사용되는 게임 오브젝트와 리소스들을 효율적으로 관리함으로써 PC뿐 아니라 모바일에서도 게임을 원활하게 진행할 수 있다.

(그림 11) 물체 움직임과 아이템 충돌 모듈을 포함한 게임 진행



(Figure 11) Game play scene with object motion and item collision module

게임 제작에 활용된 시간은 약 20일로 정도로 1인이 기획 이후에 그래픽 디자인과 프로그래밍 개발을 동시에 진행하면서 제작하였다. 그래픽 디자인 작업시간만 계산해보면 대략 10일 정도 소요되었고 이는 상대적으로 프로그래밍 개발에 소요된 시간이 게임 제작에 많은 비중을 차지하지 않음을 의미하기도 한다. 이러한 배경에는 Unity 3D에서 제공하는 다양한 기능이 불필요한 작업을 줄여주고 모듈 단위로 기획과 개발을 하나의 공정으로 처리하여 제작 시간을 단축시켜

주었기 때문이다.

Unity 3D를 활용한 모바일 아케이드 게임 제작과 관련된 기존 연구[16]와 비교하면 이는 개발자의 관점에서 Unity 3D 엔진의 기능에 초점이 맞추어져 있어 완성도에 비해 상대적으로 낮은 그래픽 질을 갖추고 있는 반면 본 연구의 모바일 캐주얼 게임은 그래픽 디자인 능력을 갖춘 기획자의 관점에서 1인이 유사한 수준의 기능을 갖춘 게임을 제작하였으며 오히려 그래픽 질이 높아 완성도 높은 게임이 제작되었음을 알 수 있다. 제작 시간은 정확한 비교는 어렵지만 혼자서 모든 작업이 20일 이내에 이루어진 점을 감안하면 제안한 방법과 같이 기획 단계가 1인 개발에 얼마만큼 중요한지 알 수 있다.

(그림 12) 기존 연구와의 결과 비교



existing game[16]

our game

(Figure 12) Comparison of result with existing study

만약에 그래픽에 능숙하지 않은 사용자의 경우 웹에 많이 올라오는 아이콘 이미지들을 활용하면 2차원 그래픽 작업에 유용하게 쓰일 수 있지만 포토샵 같은 그래픽 편집 도구의 기본적인 사용방법은 알고 있어야 색상이나 크기 정도의 수정은 가능할 것이다. 3차원 리소스들은 본인 게임에 최적화된 리소스를 찾기는 현실적으로 불가능하기 때문에 공동 작업이 불가피하다. 다만 게임의 컨셉이 간단한 캐주얼 게임이라면 Unity 3D에서 제공하는 구, 박스와 같은 게임 오브젝트에 적절한 텍스처들을 매핑하여 사용하

는 것도 하나의 방법이 될 것이다. 마지막으로 Unity 3D의 조명, 스카이박스, 안개, 파티클 등의 효과를 사용하면 게임의 질을 높이는 다른 방법이 될 것으로 생각한다.

본 논문은 혼자서 3차원 캐주얼 게임을 효과적으로 제작하기 위하여 이벤트기반 모듈 기획을 제안하여 제작을 고려한 기획과 이를 통해 구현하는 방법까지 일련의 과정을 다루고 있다. 또한 개발 과정에서 전문적 지식이 없는 일반 사용자도 보다 쉽게 접근하기 위하여 Unity 3D 엔진을 활용하였다. 이러한 관점에서 보면 비전문가가 드래그 앤 드롭으로 코딩 없이 게임을 쉽게 제작할 수 있으며 타 엔진들보다 제작난이도가 쉬운 장점을 갖는 게임 샐러드와 같은 제작 도구를 활용하는 것이 1인 게임 개발에 용이할 수 있다. 그러나 이는 복잡한 시스템이나 게임방식의 구현이 어렵기 때문에 게임의 장르가 정형화되어 있고 2차원 콘텐츠에 특화되어 있어 본 논문과 같이 3차원 게임을 제작하기에는 한계가 존재한다. 본 논문은 전문적인 지식이 다소 부족한 일반 사용자의 관점에서 혼자 게임을 효율적으로 제작하기 위한 방법을 제시하기 위한 이지만 향후 다양한 장르를 포함하여 발전시켜 나갈 수 있도록 하기 위하여 2차원과 3차원 콘텐츠를 모두 다룰 수 있으며 일반 사용자와 전문가를 모두 고려하는 것이 필요하다. 따라서 이러한 점을 모두 감안하여 Unity 3D를 이용한 게임 제작 방법을 설계하였다.

## 5. 결론

본 논문은 일반 사용자가 혼자서 원하는 게임을 제작하고자 하는 개발 환경의 변화에 맞춰 보다 효과적으로 게임 제작을 진행할 수 있도록 Unity 3D 게임 엔진을 이용한 이벤트기반 모듈 기획과 이를 통한 게임 제작 과정에 대하여 설명하였다. DirectX SDK와 같은 저수준의 게임 개발 방식이나 Ogre 3D 게임 엔진과 같이 전문적 지식을 갖춘 개발자에 적합한 방법들은 무엇보다 프로그래밍 언어에 대한 깊은 이해를 요구하기 때문에 많은 사전 준비를 필요로 한다. 하지만 Unity 3D와 같은 통합 에디터형 게임 엔진은 이러한 문제를 해결하여 상대적으로 쉽게 구

현할 수 있는 개발 환경을 제공하고 있어 본 연구는 이를 기반으로 혼자서 효율적으로 게임을 제작함에 있어 기획에 초점을 맞춘 연구와 전체적인 게임 제작 과정을 간단한 예시 게임을 제작하면서 설명하고자 하였다.

본 논문에서 구현한 게임은 비교적 간단한 구성의 모바일 캐주얼 게임만을 다루고 있어 사용된 기능들이 다소 부족할 수 있지만 향후 더욱 많은 기능들을 담고 있는 게임을 개발함으로써 다양한 방향을 제시하고자 한다. 그리고 제한한 이벤트기반 모듈 기획을 틀 형태의 편집 도구로 설계하여 보다 효율적인 제작이 가능하도록 연구할 계획이다. 이밖에 Unity 3D에서 제공하는 높은 품질의 파티클이나 애니메이션 시스템을 활용하여 게임의 완성도를 높임으로써 실제 상용화 단계까지 끌어들이므로써 단순히 취미로써 개발이 아닌 높은 수준의 게임 개발을 1인이 제작할 수 있는 방향을 연구할 계획이다.

## References

- [1] Myunghee Cha, "A Study for students through game education contents," Journal of Korean Society for Computer Game, Vol.24, No 4, pp.194-200, Dec. 2011.
- [2] Sumi Ji, Jeongjoong lee, Sunggook Kim, Kyeon-gdeok Woo and Sungwook Baik, "Development of intuitive author for 3D game/animation contents," Journal of Korea Multimedia Society, Vol.13, No.5, pp.780-791, May 2010.
- [3] Unity 3D Site : <http://korea.unity3d.com/>
- [4] Wikipedia-Game\_Engine : [http://en.wikipedia.org/wiki/Game\\_engine](http://en.wikipedia.org/wiki/Game_engine)
- [5] Sierra AGI Engine : <http://www.sierra.com>
- [6] Quake Engine : <http://www.quakelive.com>
- [7] Unreal Engine : <http://www.unrealengine.com>
- [8] Hunjoo Lee, Taejoon Park and Hyunbin Kim, "A study on development of an on-line 3D game engine," Journal of Korea Game Society, Vo-1.3, No.2, pp.42-55, Sep. 2003.
- [9] Myounjae Lee and Kyoungnam Kim, "A serious game development for historical education using OGRE engine," Journal of Digital Contents Society, Vol. 11, No.3, pp.385-392, Sep. 2010.
- [10] Hyunwoo Song and Seungtaek Ryoo, "The development of RPG prototype game using Ogre 3D engine," Journal of Korean Society for Computer Game, Vol.3, No.21, pp.167-174, June 2010.
- [11] Sooyol Ok and Sungkil Kim, "The development of overhead crane simulator using open source physics engine," Journal of Korea Game Society, Vol.9, No.5, pp.95-104, Oct. 2009.
- [12] Seungje Park and Taesuk Kihl, "Rhythm game design for effective music education," Journal of Korea Game Society, Vol.12, No.1, pp.33-42, Feb. 2012.
- [13] Jaeyoung Lee and Junsik Kwon, "Touch-based gaming system using augmented reality technology," Journal of Digital Contents Society, Vol.15, No.1, pp.69-76, Feb. 2014.
- [14] Sangmoon Jeon, Kyoungsu Kim and Gappyo-ung Lee, "Design and implementation of a smart-phone serious game for learning english conversation," Journal of Digital Contents Society, Vol.14, No.2, pp.235-245, Jun. 2013.
- [15] C. J. Lim, Wondae Han and Jeongyun Guen, "Educational game making-tool development using Unity3D engine: birth of game," Journal of Korea Game Society, Vol.14, No.1, pp.29-38, Feb. 2014.
- [16] Sookyun Kim, Jihoon Kang, Giseob Song, Hebum Lee and Syungog An, "Development of mobile arcade game using Unity 3D engine," Journal of Korea Knowledge Information Technology Society, Vol. 8, No.2, pp.9-16, April 2013.



### 정 서 원

2011년 : 동국대학교 게임멀티미디어  
공학과 (공학사)

2013년 : 동국대학교 영상대학원  
(디자인학석사)

2013년~현재 : 동국대학교 영상대학원 멀티미디어  
학과 박사과정

관심분야 : 3D애니메이션(3D Animation), 실시간렌  
더링(Real-time Rendering), 융합콘텐츠제작  
(Convergence Contents Production) 등

### 김 진 모



2006년 : 동국대학교 멀티미디어공학  
과 (공학사)

2008년 : 동국대학교 영상대학원  
멀티미디어학과 (예술공학  
석사)

2012년 : 동국대학교 영상대학원  
멀티미디어학과 (공학박사)

2012년~2014년: 동국대학교 영상문화콘텐츠연구원  
전임연구원

2014년~현재 : 부산가톨릭대학교 소프트웨어학과  
조교수

관심분야 : 게임프로그래밍(Game Programming), 컴  
퓨터그래픽스(Computer Graphics), 영상처리  
(Image Processing) 등