

Training in the production of effective prototypes using Core Mechanic Diagram and Unreal Blueprint

Bu-ho Choi*

*Professor, School of Game, Chungkang College of Cultural Industries, ICheon, Korea

[Abstract]

In this paper, we propose a training method to efficiently create prototypes using 'core mechanical diagram' to make game design easier and 'BluePrint', a visual scripting tool of Unreal that enables relatively easy functionality without programming knowledge. With the development of game engines, game development is becoming faster and easier, and using this, non-programmers without knowledge of game development are clearly showing a tendency to participate in game development. However, existing game development methodologies and game implementation methods require complex design processes and specialized knowledge. In order to make it easier for non-programming experts to develop games, they will simplify the complexity of existing game development methodologies, and propose educational methods that can focus on and implement the essence of game design based on cases where non-programmers have implemented prototypes themselves.

▶ **Key words:** Game Design, Core Mechanic Diagram, Prototyping, Ureal, Blueprint, Education

[요 약]

이 논문에서는 게임의 진행방식을 보다 쉽게 설계할 수 있는 '코어 메카닉 다이어그램'과 프로그래밍 지식이 없어도 비교적 손쉽게 기능을 구현할 수 있는 언리얼(Unreal)의 비주얼 스크립팅 툴인 '블루프린트(BluePrint)'를 활용하여 효율적으로 프로토타입을 제작할 수 있는 교육방법을 제안한다. 게임엔진의 발전에 따라 게임 개발이 보다 빠르고 쉬워지고 있으며, 이를 이용하여 게임 개발 관련 지식이 없는 프로그래밍 비전문가들이 게임개발에 참여하는 경향이 뚜렷하게 나타나고 있다. 하지만 기존 게임 개발 방법론과 게임 구현 방법들은 복잡한 설계과정과 전문적인 지식을 요구한다. 프로그래밍 비전문가도 보다 쉽게 게임을 개발 할 수 있도록, 기존 게임 개발 방법론들의 복잡성을 단순화하고, 프로그래밍 비전문가들이 직접 프로토타입을 구현한 사례를 바탕으로 게임 디자인의 본질에 집중하고 구현할 수 있는 교육방법을 제안하려 한다.

▶ **주제어:** 게임기획, 코어 메카닉 다이어그램, 프로토타이핑, 언리얼, 블루프린트, 교육

• First Author: Bu-ho Choi, Corresponding Author: Bu-ho Choi
*Bu-ho Choi (sk8geenie@ck.ac.kr), School of Game, Chungkang College of Cultural Industries
• Received: 2020. 11. 10, Revised: 2020. 11. 25, Accepted: 2020. 11. 26.

I. Introduction

최근 게임엔진은 게임 분야뿐만 아니라 애니메이션, 건축, 설계 등 다양한 분야에서 활용되고 있으며, 비전문가도 게임엔진을 활용하여 콘텐츠를 제작할 수 있는 환경을 구축해 나가고 있다. 이런 변화된 게임 개발 환경에서는 게임 설계 방식 또한 보다 간편하고 손쉬운 설계 방법론으로 변화해야 한다. 기존의 게임 설계 방식 중 비교적 완성도가 높은 ‘4F Process’[1]의 경우, ‘린 캔버스(Lean Canvas)’와 유사한 절차를 따르지만, 린 캔버스에서 제시한 9단계의 절차를 4단계로 축소하였다. 이를 통해 보다 빠르고, 효율적으로 게임을 설계할 수 있는 방식을 제시하고 있다. 그러나 이 두 가지 방법론은 게임 플레이어의 요구 사항부터 게임 개발 과정과 게임 서비스 단계까지 게임 개발 이외의 분야까지 포함하는 복잡한 단계로 설계되어 있다.

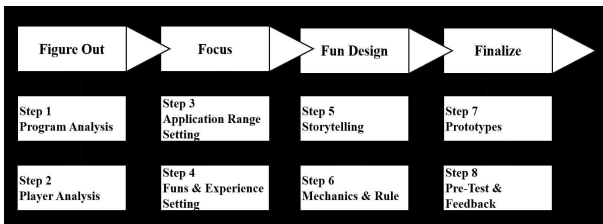


Fig. 1. 4F Process factors and process

‘MDA 프레임워크(MDA Framwork)’와 ‘게임 메카닉스 디자인(Game Mechaincs Desing)’은 앞의 두 방법론에 비해 게임 디자인 단계에 보다 집중한 방법론이다.

MDA 프레임워크는 게임 경험 구조의 이해에 특화된 분석 도구로써, 개발자와 플레이어가 게임을 어떻게 경험하는지를 입체적으로 분석할 수 있도록 고안된 게임 분석 기법이다.[2] 상용화에 성공한 게임을 MDA프레임워크를 이용하여 분석하고 게임의 장단점을 파악에 사용되기도 한다.[3] 그러나 개발 단계에서 게임 디자이너의 설계와 플레이어의 경험 순서 차이로 인해 게임의 메커니즘이 게임에 적합한지 검증하는 절차가 복잡하여 게임개발 입문자가 사용하기에는 적합하지 않다. ‘게임 메카닉스 디자인’[4] 역시 앞의 두 방법론에 비해 게임 디자인 과정에만 집중할 수 있게 한다는 긍정적인 측면이 있지만, 게임 메카닉스 디자인 핵심 설계 외의 4단계를 추가로 설계해야 한다. 결과적으로 비전문가가 앞서 제시된 방법론들을 게임 디자인 단계에 적용하기 쉽지 않다.

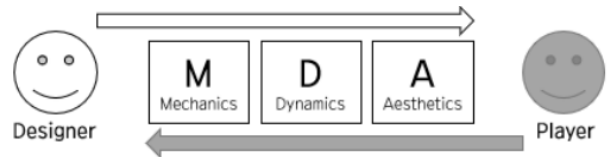


Fig. 2. MDA Framework

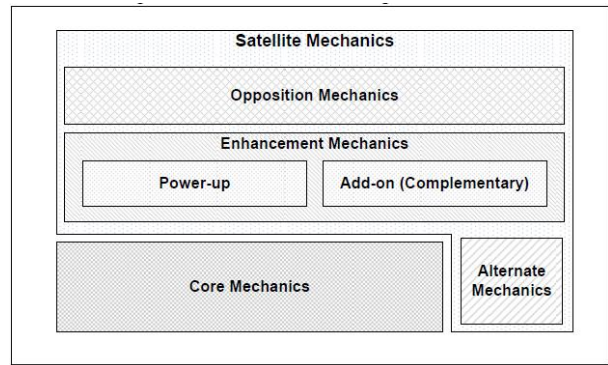


Fig. 3. An architectural model for game mechanics

본 논문에서 제시하는 ‘코어 메카닉 다이어그램(Core Mechanic Diagram)’은 앞에서 언급한 게임 개발 방법론들의 복잡성을 단순화하여 게임 디자인의 본질에 집중할 수 있도록 제시할 수 있는 방법론이다. ‘코어 메카닉 다이어그램’은 게임 개발자들뿐 아니라, 비전문가 역시 게임 디자인을 위한 설계에서 활용할 수 있는 방법론이다.

게임엔진의 발전에 따라 게임 개발이 보다 빠르고 쉬워지는 상황을 반영해 비전문가들을 대상으로 ‘코어 메카닉 다이어그램’과 프로그래밍 전공자가 아니더라도 어느 정도의 기반 지식이 있다면 게임 개발이 가능한[5] 언리얼(Unreal)의 ‘블루프린트(BluePrint)’를 활용해 게임 개발 교육을 진행하고자 한다. 본 논문에서 비전문가는 프로그래밍 지식이 전혀 없는 게임 학과 대학생들로 한정한다.

위의 두 가지를 활용해 직접 게임을 디자인하고 게임 프로토타입을 구현하고, 그 결과를 분석한 교육 사례를 바탕으로 비전문가도 효율적으로 게임엔진을 활용할 수 있는 교육 방법을 연구해 보고자 한다.

II. Preliminaries

1. Utilize ‘Core mechanic Diagram’

1.1 Introduction to ‘Core Mechanic Diagram’

게임 디자이너 Charmie Kim은 Gamasutra에서 게임 메커니즘을 게임 디자인에 연결하는 것은 쉽지만 이러한 메커니즘이 게임에 적합한지 항상 분명하지 않기 때문에

보다 간단한 방식을 이용하여 이를 검증하고자 '코어 메카닉 다이어그램'을 중심으로 게임을 설계하는 프레임워크를 공개하였다.[6]

'코어 메카닉 다이어그램'은 다음과 같은 4개의 다이어그램으로 구성되어 있으며, 각각의 역할은 아래와 같이 정의할 수 있다.

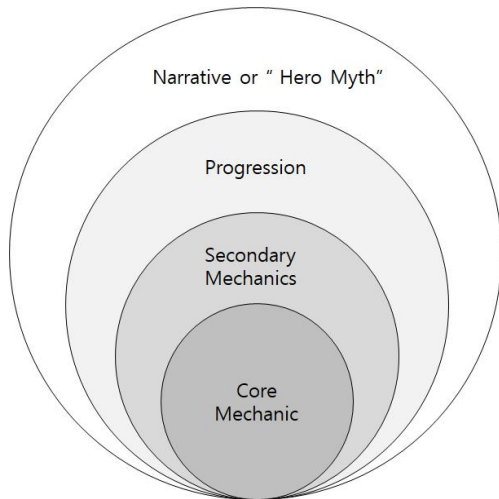


Fig. 4. Charmie Kim's 'Core Mechanic Diagram'

'코어 메카닉(Core Mechanic)'은 일반적으로 가장 자주 발생하는 의도적인 상호 작용 즉, 행동을 뜻한다. 예를 들어 플랫폼 게임에서 '코어 메카닉'은 일반적으로 점프이며, 슈팅게임에서는 사격이 가장 자주 발생하는 행동이다. 레이스 게임에서는 운전이 될 수 있다. '코어 메카닉'을 결정하는 또 다른 방법은 '그것이 없으면 게임을 전혀 할 수 없다'이다.

'세컨더리 메카닉(Secondary Mechanic)'은 그다음으로 자주 일어나는 상호 작용을 뜻한다. 가장 자주 발생하는 행동에서 그다음으로 많이 발생하는 행동으로 계층화될 수도 있다.

'프로그레션(Progression)'은 게임의 규칙을 형성하며, 전체적인 수준에서 게임 시스템 내에서 게임의 진행 방식을 뜻한다.

'내러티브(Narrative)'는 '코어 메카닉 다이어그램'의 최고 외층을 뜻하며, 게임의 스토리 또는 세계관을 나타낼 수 있다.

1.2 Analysis of 'Core Mechanic Diagram' through game example

앞서 살펴본, '코어 메카닉 다이어그램'을 이용하여 Charmie Kim은 다양한 게임을 분석하였고, 다음과 같이 제시하고 있다.

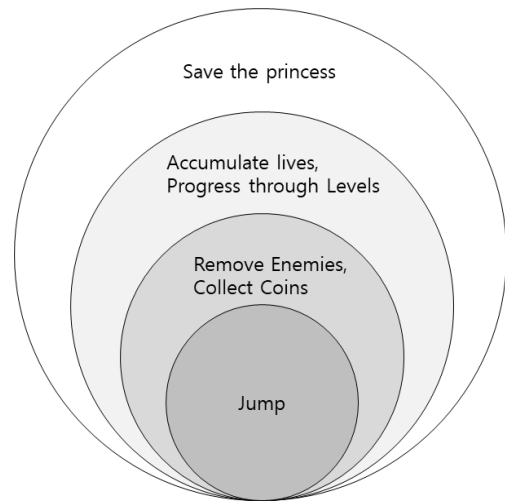


Fig. 5. Nintendo's <Super Mario> Core mechanic diagram.

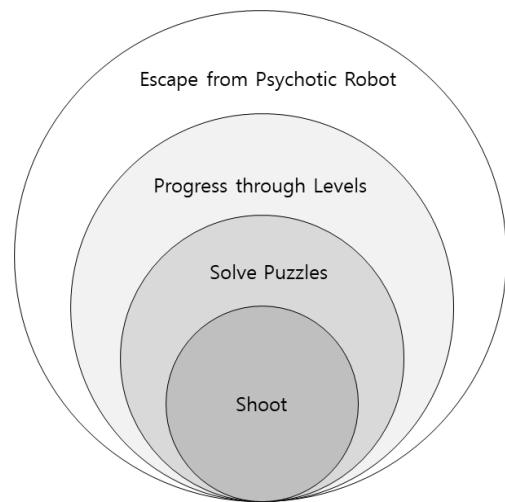


Fig. 6. Valve Corporation's <Portal> Core mechanic diagram.

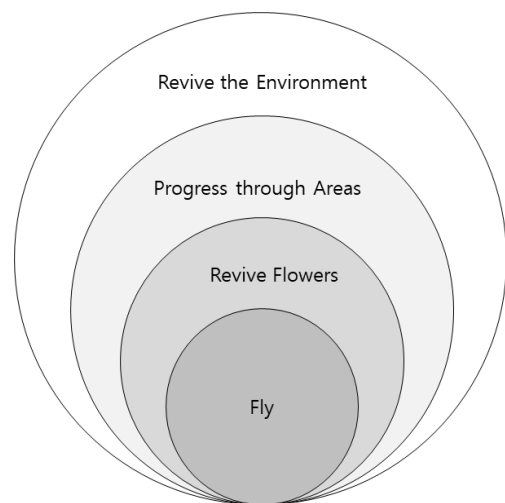


Fig. 7. Blueprint Games' <Flower> Core mechanic diagram.

위와 같이 '코어 메카닉 다이어그램'을 이용하여, 게임

에서의 주요 행동과 진행 방식, 스토리를 간단하게 나열하고 분석할 수 있다. 이런 분석을 통해 게임에서 제공하는 본질을 단순화할 수 있고, 게임에서 구현하고자 하는 핵심을 파악하기 쉽게 만드는 장점이 있다.

다음은 같은 분석방법을 이용하여, ‘코어 메카닉 다이어그램’ 교육에 참여한 학생이 게임을 분석해 보았다.

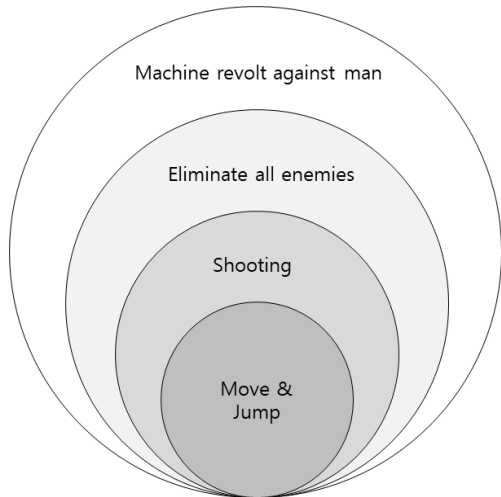


Fig. 8. <Rock Man>'s Using the Core Mechanic Diagram, Example of Student Analysis

1.3 'Core Mechanic Diagram' Practice

‘코어 메카닉 다이어그램’은 게임 분석에 활용할 수 있지만, 반대로 게임 설계에 활용하기에도 용이하다. 게임에서 구현하고자 하는 핵심요소를 나열함으로써, 제작하고자 하는 게임을 단순화하여 설계할 수 있다.

다음은 ‘코어 메카닉 다이어그램’ 교육 중 학생이 제작하고자 하는 프로토타입을 ‘코어 메카닉 다이어그램’으로 설계하였다.

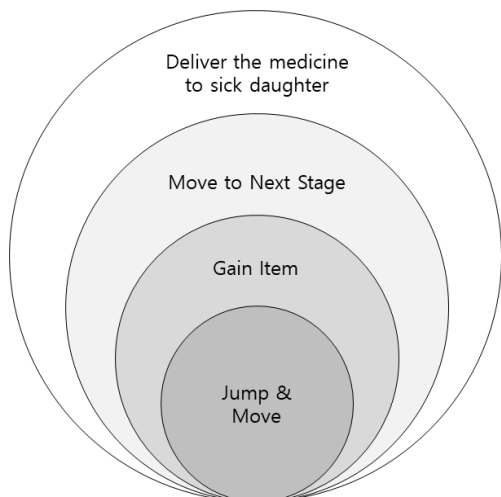


Fig. 9. Using the 'Core Mechanic Diagram', Example of Student Game Design 1

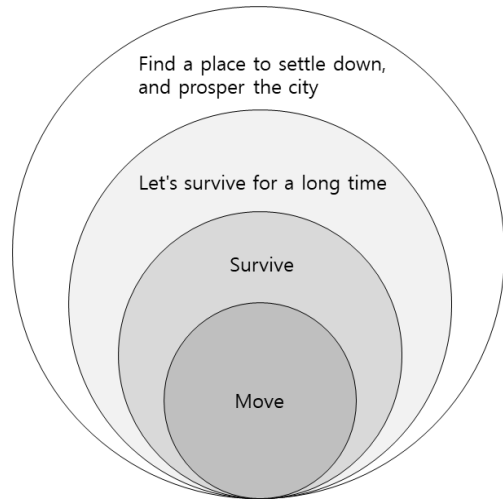


Fig. 10. Using the 'Core Mechanic Diagram', Example of Student Game Design 2

위와 같은 분석을 통해 코어 메카닉을 활용해 게임의 설계 과정을 단순화했을 때 비전공자도 손쉽게 게임을 설계할 수 있음을 확인할 수 있다. 또한 게임의 본질을 정의하는 시간을 단축할 수 있고, 구현 항목이 명확해지는 장점도 발견하였다.

2. Utilizing Visual Scripting

2.1 Introduction to Visual Scripting

비주얼 스크립팅의 등장은 기존의 텍스트 코딩과는 달리 시각적인 기능 구현을 지원해 프로그래밍 언어가 익숙하지 않는 아이들을 포함하여 모든 연령층이 보다 쉽게 프로그래밍 할 수 있는 환경을 제공한다. 최근 블록 코딩 방식으로 코딩 교육이 가능하도록 개발된 <스크래치>와 <엔트리>의 보급으로 비주얼 스크립팅이 대중화되었다. 스크래치를 활용해 복잡한 프로그래밍에 대한 지식이 없이도 쉽게 프로그램을 작성할 수 있고,[7] 이러한 변화는 비주얼 스크립팅을 게임엔진에 도입하여 보다 쉽고, 빠르게 게임을 개발하고자 하는 개념과 동일한 방향성을 제시하고 있다. 오픈소스 게임엔진인 <Godot엔진>의 공식 홈페이지에서는 다음과 같이 비주얼 스크립트의 발전과 자사 엔진에 비주얼 스크립트 도입의 이유를 설명하고 있다.



Fig. 11. Godot Game Engine

Visual Scripting은 프로그래밍 진입 장벽을 낮추기 위해 고안된 도구이다. 코드보다 더 시각적이어서 이해하는데 쉽다. 아티스트, 애니메이터, 게임 디자이너 및 다른 사람들도 논리의 흐름을 쉽고 빠르게 파악할 수 있으며 비주얼 스크립트가 필요한 가장 일반적인 사용 케이스는 다음과 같다.

- 엔진을 배우고 싶지만 아직 프로그래밍 경험이 전혀 없는 게임 개발 입문자.
- 프로그래밍에 대한 경험이 없고 간단한 프로토타입이나 게임을 만들고 싶어 하는 아티스트와 게임 디자이너.
- 팀으로 일하면서 아티스트나 게임 디자이너에게 게임 논리의 일부를 주어 그들의 작업량을 줄이기 원하는 프로그래머.[8]

<Godot 엔진>에서 밝힌 바와 같이 비주얼 스크립트가 필요한 경우는 게임 개발 환경에서 손쉽게 발견할 수 있다. 프로그래밍 능력이 없는 비전공자도 게임을 개발할 수 있는 가장 유용한 도구로써 비주얼 스크립팅을 기법으로 제시하고 있으며, 이를 활용하고 있다.

위와 같은 이유에서 가장 대중적인 게임엔진[9]인 <유니티>와 <언리얼> 또한 비주얼스크립트 도입과 지원에 적극적인 자세를 취하고 있다. 특히, 언리얼에서 제공하는 <블루프린트>는 텍스트 기반이 아니기에 문법에 대한 지식이 필요하지 않다. 그래서 프로그래머보다 디자이너나 기획자가 주로 사용한다.[10] 이러한 이유로 이 논문에서는 언리얼의 블루프린트를 활용하여 프로그래밍 비전문가들에게 프로토타입제작을 교육하고 그, 사례를 제시하려 한다.

2.2 Introduction to Unreal Blueprint

비주얼 스크립트를 지원하는 게임엔진 중 가장 대중적인 <언리얼>의 경우 공식 홈페이지를 통해 블루프린트를 다음과 같이 소개하고 있다.

언리얼 엔진의 비주얼 스크립팅 시스템인 *Blueprint* (블루프린트)는 언리얼 에디터 안에서 노드 기반 인터페이스를 사용하여 게임 플레이 요소를 만드는 개념을 토대로 한 비주얼 스크립팅 시스템입니다. 일반적인 스크립팅 언어와 마찬가지로, 엔진 내 객체 지향형(OO) 클래스 또는 오브젝트를 정의하는 데 사용됩니다. UE4 를 사용하다 보면, 블루프린트를 사용하여 정의된 오브젝트를 그냥 일상적으로 "블루프린트"라 하는 경우가 많습니다. 이 시스템은 매우 유연하고 강력해서, 보통은 프로그래머가 없으면 힘들었던 개념 구현이나 툴 작업 거의 전부를 디자이너 스

스로 해낼 수 있게 되었습니다. 게다가, 언리얼 엔진의 C++ 구현에서 가능한 블루프린트 전용 마크업을 통해 프로그래머가 어떤 기반 시스템을 제작하면, 디자이너가 그것을 확장하는 것도 가능합니다.[11]

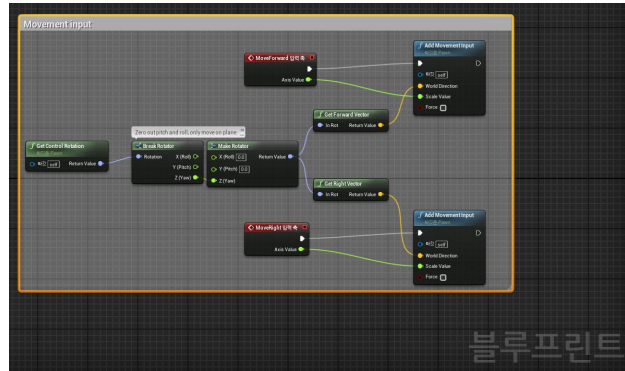


Fig. 12. unreal Blueprint

<블루프린트>의 작동 방식은 기본적으로 시각적인 스크립팅을 게임에 추가하는 것이다. 노드, 이벤트, 함수, 변수 등을 선으로 연결하여 복잡한 게임플레이 요소를 만드는 것이 가능하다. 또한, 오브젝트 생성, 개별 함수, 일반적인 게임 플레이 이벤트 등 각각의 블루프린트 인스턴스에 고유한 여러 가지 목적의 노드 그래프를 사용하는 식으로 작동한다. 다음으로는 언리얼 블루프린트를 활용한 프로토타입 제작 사례를 알아보려고 한다.

III. The Proposed Scheme

1. Direction of Prototyping Development Training Using Unreal Blueprint

게임 개발에 있어 쉽게 적용이 가능한 ‘코어 메카닉 다이어그램’과 프로그래밍 비전공자들이 손쉽게 개발할 수 있는 언리얼의 ‘블루프린트’는 보다 많은 사람들이 게임을 개발할 수 있는 환경을 제공한다는 점에서 지향점이 같다. 15주 동안 진행된 수업은 프로그래밍 비전공자 및 언리얼 엔진을 경험하지 못한 44명의 대학생을 대상으로 했다. 게임의 설계는 ‘코어 메카닉 다이어그램’을 활용해 이루어졌고, 게임의 구현은 ‘블루프린트’로 진행하였다. 특히, 언리얼은 FPS 게임에 특화된 엔진으로 볼 수 있으며[12], 엔진의 특성에 맞게 언리얼에서 제공하는 게임 템플릿 중 ‘일인칭(FPS)’ 템플릿을 이용하여 ‘블루프린트’의 기본 기능을 학습하고, 이를 확장하며 응용할 수 있도록 하였다.

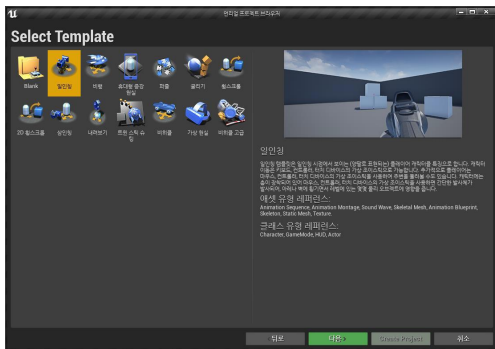


Fig. 13. Unreal Template

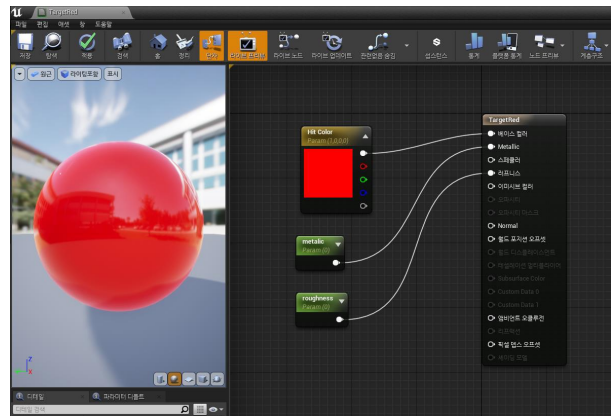


Fig. 15. Material Blueprint

2. A Case Study of Prototyping Development Training Using Unreal Blueprint

<언리얼> 엔진의 기본적인 툴 사용법은 가상의 공간을 구성하고, 간단한 오브젝트를 배치하는 과정을 통해 툴 사용법에 익숙해지도록 하였다. 또한 언리얼의 블루프린트에서 기본 이벤트 노드를 교육하고, 오브젝트에 기능을 구현하는 방식으로 비주얼 스크립팅을 이해하도록 교육을 진행하였다.

UMG UI Editor는 게임 내에서 사용되는 HUD, 메뉴, 그리고 사용자에게 보이고자 하는 기타 인터페이스 관련 그래픽 등의 UI요소를 만드는데 사용되는 툴로써, 이를 이용하여 UI를 구성하고 이벤트를 처리하는 방법을 가르치고, '위젯 블루프린트'를 이용하여 UI의 레이아웃 구성하는 방법에 대해 교육하였다.

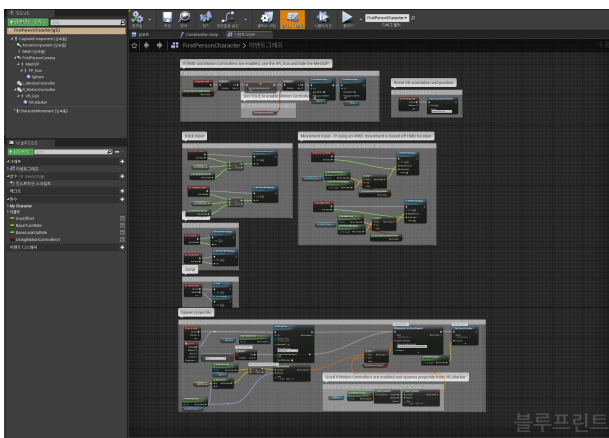


Fig. 14. unreal Blueprint

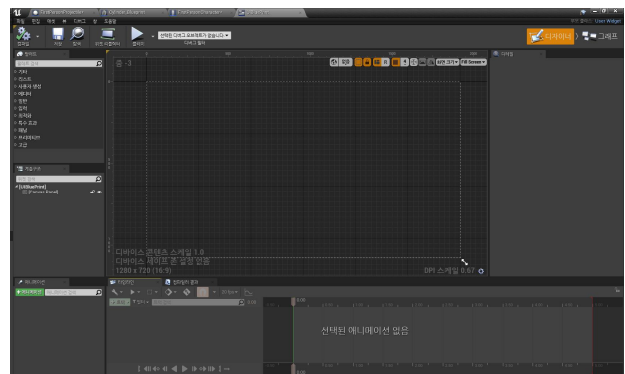


Fig. 16. Widget Blueprint

'매터리얼 블루프린트(Material Blueprint)'는 언리얼 엔진에서 제공하는 오브젝트의 시각적인 재질과 질감을 설정하거나 생성할 수 있는 툴로써 라이팅 효과와 텍스처의 다양한 노드를 조합하여 질감을 표현하며, 이를 이용하여 보다 현실적인 게임을 만들 수 있는 기법에 대하여 학습하였다.

마지막으로 '비헤이비어 트리(Behavior Tree)', '블랙보드(Black Board)', 'AI 컨트롤러(AI Controller)'는 비주얼 노드 기반 시스템을 통해 인공지능의 구현을 제공하는 툴이다[13]. 이를 이용하여 프로토타입에서 등장하는 적 캐릭터 인공지능을 설계하고, '내비게이션 메쉬(Navigation Mesh)'를 이용한 길 찾기 기능을 추가하였다. 수업을 통해 학생들이 기능을 확장하고, 액터(Actor)가 설정한 구간을 반복 이동/순찰하며, 플레이어 캐릭터 발견 시 추격하는 등 적 캐릭터가 좀 더 현실적인 행동을 할 수 있도록, 기능에 대하여 교육을 진행하였다.

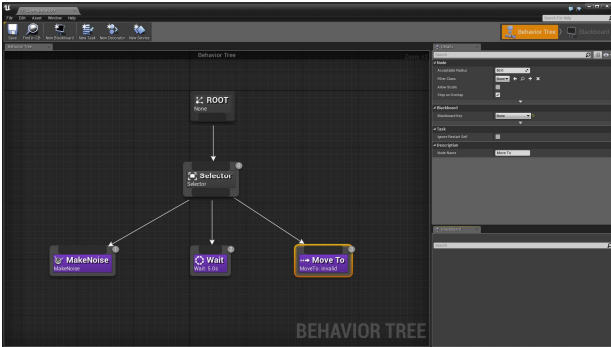


Fig. 17. Behavior Blueprint

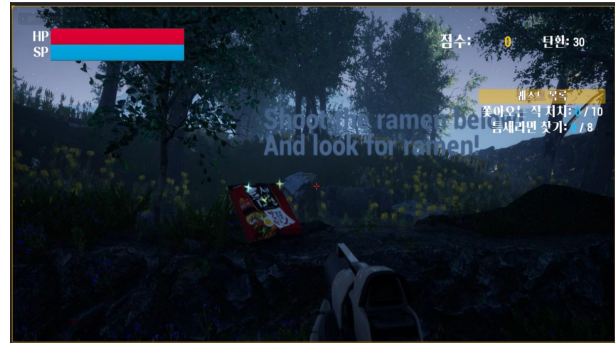


Fig. 18. the results of the students attending the class

이와 같이 언리얼에서 제공하는 다양한 블루프린트와 기능에 대해 교육을 진행하였고, 14주간의 ‘블루프린트’ 교육과 1주간의 ‘코어 메카닉 다이어그램’을 이용한 게임 설계를 진행하였다. 교육에 참여한 학생들은 원하는 프로토타이핑을 제작할 수 있도록 하였다.

이러한 수업을 통해 다음과 같이 교육에 참여한 학생들의 결과물을 확인할 수 있었다.

15주간 진행된 교육에 참여한 학생 44명 중 게임 프로토타입 개발에 성공한 학생은 다음 그래프와 같으며, 제작 성공 비율은 93%에 달했다. 이를 통해 15주간의 교육 기간 동안 비전공자들이 게임 설계와 구현에 있어 ‘코어 메카닉 다이어그램’과 언리얼 ‘블루프린트’가 유효한 방식을 입증할 수 있었다.

Table 1. Production Success Rate(Units:persons)

	Failure	Successfully	Total
Prototype Production	3	41	44

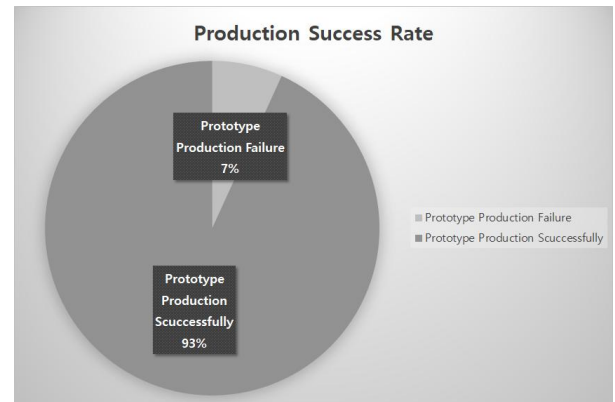


Fig. 19. Number of successful prototype students

IV. Conclusions

15주간 진행된 교육과정에서 ‘코어 메카닉 다이어그램’을 이용한 게임의 설계와 언리얼의 ‘블루프린트’를 이용한 프로토타이핑의 구현이 진행되었으며, 교육에 참여한 학생들의 결과물을 통해 기대 이상의 결과를 얻을 수 있었다. 직군별 학생들의 결과에서 차이점 또한 발견할 수 있었다. 게임 아트직군의 경우 프로토타이핑 구현 시 사용된 리소스 중 일부를 직접 제작하여 프로토타입에 반영하며 보

다 높은 완성도의 프로토타입을 개발하였으며, 게임 디자인 직군은 '코어 메카닉 다이어그램'을 통한 게임의 설계로 설계 시간을 단축하고, 자신이 설계한 게임의 본질을 명확하게 하는 측면에서 긍정적인 반응을 확인할 수 있었다.

전통적인 방식에서 게임의 설계와 설계의 재미 검증을 위한 프로토타입 제작 과정은 많은 인원과 긴 시간을 필요로 한다. 하지만, 여러 가지 개발도구의 발전으로 다양한 직군이 보다 쉽고 빠르게 자신이 원하는 게임을 직접 검증해 볼 수 있는 기회가 제공되고 있다. 본 논문에서 소개한 '코어메카닉 다이어그램'과 '블루프린트'를 활용하여 게임 개발자가 아닌 일반인 또는 게임개발에 입문하는 학생들에게 게임의 개발 과정과 게임의 메커니즘을 보다 손쉽게 이해시키는데 도움을 줄 수 있을 것이다. 이를 통해 보다 효과적인 게임개발 교육을 진행할 수 있을 것으로 기대된다.

REFERENCES

- [1] Sungjin Park, Sangkyun Kim, "Gamification Development Methodology - Design and Comparative Analysis of 4F Process", Journal of Digital Contents Society, Vol. 19, No. 6, pp.1131-1144, Jun. 2018.
- [2] Robin Hunicke, Marc LeBlanc, Robert Zubek, "MDA: A Formal Approach to Game Design and Game Research", the Game Developers Conference, San Jose, California, 2001-2004. DOI : <https://users.cs.northwestern.edu/~hunicke/MDA.pdf>
- [3] Jihye Kang, Ahyoung Jang, Inhee Song, "A Study on Game Mechanics and Dynamics of Survival Game Content", Journal of Korea Game Society, Vol.18, p.5-13, Aug, 2018.
- [4] Carlo Fabricatore, "GAMEPLAY AND GAME MECHANICS DESIGN: A KEY TO QUALITY IN VIDEOGAMES", ENLACES (MINEDUC Chile)-OECD Expert Meeting on Videogames and Education, p.14, Oct. 2007.
- [5] Kim Min-jae, Park Kyung-nam, "Development of Indie Games Using Unreal4", Korean Institute of Information Technology, Proceedings of KIIT Conference, Chosun University, Gwangju, pp.396-397, Aug. 2018.
- [6] Charmie Kim, "Designing around a core mechanic", Gamasutra, Jun. 2012, DOI : https://www.gamasutra.com/blogs/CharmieKim/20120612/172238/Designing_around_a_core_mechanic.php
- [7] Hong-Sub Lee, Hyung-Won Jeong, Young-Kyo Kim, "Design and Implementation of Game for Learning Game Production Principles: Centering on Scratch Language", Journal of Digital Convergence, Vol. 18, pp.404-405, May 2016.
- [8] Godot Engine Official Website, Godot Engine Docs. 2014, DOI : <https://godotengine.org/>
- [9] Myoun-Jae Lee, "A Study on Game Production Education through Recent Trend Analysis of 3D Game Engine", Journal of the Korea Convergence Society, Vol.4 No.1, pp. 15-20, Mar. 2013.
- [10] HanSeong, SeungTaek Ryoo, SangHyun Seo, "A Comparative Study on the Structure and Implementation of Unity and Unreal Engine 4", Korea Computer Graphics Society, Vol. 25, No. 4, p.19, Aug. 2019.
- [11] Unreal Engine Official Website, Unreal Engine Docs. DOI : <https://docs.unrealengine.com/ko/Engine/Blueprints/GettingStarted/index.html>,
- [12] Soo-Kyun Kim, Heau-Jo Kang, Kyung Sung, "First-Person Shooter Game Development using Unreal Engine", Journal of advanced navigation technology, Vol.14 No.5, pp.718-724, Oct. 2010
- [13] Sung-Jin Bae, Myung-Ju Kang, "Design And Development of Game AI Using Unreal Engine 4 Behavior Tree", The Korean Society Of Computer And Information, Vol. 24(1), p.267, Jan. 2016.

Authors



Bu-ho Choi received the B.S. degrees in software engineering from Seoul Digital University, in 2018, then during the master's course in Game Technology from Gachon University, Korea.

Choi has been working as a game developer since 2008. He is now an assistant professor of Game contents at ChungKang College of Cultural Industries. He is interested in game development, game artificial intelligence, and game design.