

## 게임 엔진에 기반한 3D 탈출게임 개발

조성민<sup>0</sup>, 안성욱\*, 김수권\*

<sup>0</sup>\*배재대학교 게임공학과

email: kimsk@pcu.ac.kr\*

## Development of 3D Escape Game Using Game Engine

Seong-Min Cho<sup>0</sup>, Syoungog An\*, Soo Kyun Kim\*

<sup>0</sup>\*Dept. of Game Engineering, Paichai University

### ● 요약 ●

국내외로 수많은 게임개발자들이 이용하는 Unity 3D 엔진을 이용한 게임들을 제작 하였다. Unity 엔진이 많이 쓰이는 이유 중 하나는 복잡한 알고리즘을 코딩해야만 구현 할 수 있는 기능을 컴포넌트 형식으로 지원해줌으로써 개발을 좀 더 수월하게 진행 할 수 있기 때문이다. 탈출 게임을 개발하려면 일단은 컨셉에 맞는 요소들을 만들어야 할 것이고 또한 탈출 게임의 기능들을 수행할 스크립트 구현을 하는 것 또한 매우 중요하다. 본 논문에서는 게임 엔진 내에 있는 기능들을 이용하여 탈출 개발 과정에 대해 소개한다.

키워드: Terrain, Animation, Game Engine

### I. 서론

Unity 엔진을 게임개발자들이 가장 많이 이용하는 이유 중 하나가 PC, Mobile Andriod OS, iOS등 다양한 플랫폼을 지원해주는 기능이 존재한다는 것이 가장 큰 이유라고 생각한다. 플랫폼에 맞는 게임을 개발하려면 플랫폼에 맞는 프로그래밍 언어로 개발을 해줘야하는데, 그렇게 게임 개발을 진행한다면, 출시할 플랫폼에 따라서 프로그래밍 언어를 달리 작성해야하는 단점이 생긴다. 하지만 멀티플랫폼을 지원 해줌에 따라서 1개의 프로그래밍 언어로도 여러 플랫폼의 게임을 개발할 수 있다는 것이다. 이러한 기능 외에도 편리한 기능들을 제공해 주는데, 본 논문에서는 그러한 기능들을 이용한 게임 개발 과정을 서술한다.



Fig. 1.Terrain

그림 1과 같이 게임 배경을 구현하였다. Terrain에 있는 기능들을 이용하여 지형에 텍스처를 입혀서 땅과 풀밭, 그리고 절벽을 구현하였고, Paint Tree라는 기능을 이용하여 지형 곳곳에 적절하게 나무, 수풀 오브젝트를 배치하였다.

### II. 시스템 개발

#### 1. 개발 배경 제작

게임 배경 제작은 주로 Terrain을 이용해 제작을 한다. Terrain을 이용하여 던전, 높은 산, 숲, 계곡, 협곡등등 다양한 지형을 구현할 수 있다.

#### 2. 캐릭터 체력 시스템

이무런 방해 요소 없이 게임 진행을 하다보면 게임 진행이 느슨해지는 경우가 생긴다. 게임 진행에 보다 긴장감을 부여하기 위해 제한 시간을 만들어둔다거나, 혹은 체력 시스템을 구현하는 등 여러 가지를 도입하여 플레이어로 하여금 긴장감을 유발시킨다. 본 절에서는 UGUI를 이용하여 캐릭터 체력 바를 구현하였다. 스크립트 작성으로 시간이 자남에 따라서 체력이 서서히 감소하도록 구현하였으며 또한

장애물에 피격당하거나 적 오브젝트에게 공격을 받으면 체력이 추가적으로 감소하도록 구현을 하였다.

### 3. 캐릭터 애니메이션

캐릭터 애니메이션(그림2)은 플레이어에게 시각적인 재미를 유발하는 요소 중 하나이다. 캐릭터 애니메이션은 캐릭터의 타입에 따라서 2가지의 타입을 적용 할 수 있는데, **Generic**, **Humanoid** 타입을 적용 할 수 있다. **Generic**타입은 주로 비인간형 캐릭터에게 적용시키고, **Humanoid**타입은 주로 인간형 캐릭터에게 적용한다[1].

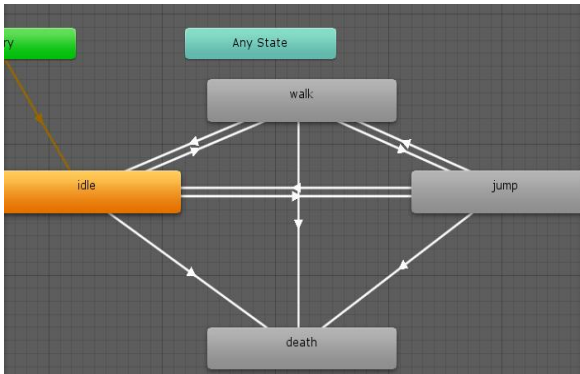


Fig. 2. Character Animator

**Humanoid** 타입을 적용하여 애니메이션을 적용하였다. 캐릭터 애니메이션 클립을 트랜지션으로 상태별로 애니메이션 재생이 전환되게끔 연결을 하였다. 또한 파라미터값을 추가하여 조작에 따른 애니메이션 전환을 할 수 있게 하였고 스크립트 작성을 통해 이를 동작시키도록 하였다[2].

### 4. 방해 요소



(a) (b)  
Fig. 3. Event Point

탈출 게임에 있어서 방해 요소의 존재는 필수인 부분이다. 주로 오브젝트 형으로 일정구간에 설치를 해서 게임 진행을 방해한다거나 혹은 적 캐릭터를 배치하여 주인공 캐릭터를 공격하는 식으로 구현을 한다.

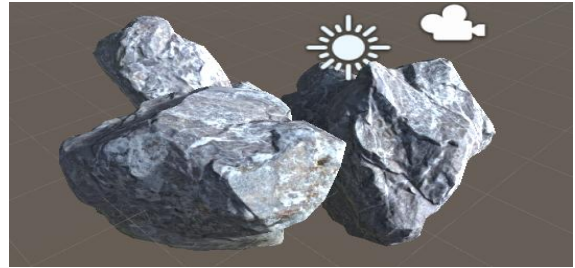


Fig. 4. 3D Object

위의 그림과 같이 방해 요소를 구현하였다. 낙석 이벤트를 구현하기 위해 빈 오브젝트를 생성하여 그림 3과 같이 낙석 이벤트가 발생하는 지점을 배치하였다. 또한 **Instantiate**함수를 이용하여 해당 지점에 위의 그림 4와 같은 오브젝트를 생성하였다.

## III. 개발 환경

본 논문에서의 게임 개발에 사용된 프로그램은 **Unity 5 2018.2.16.f**을 사용하여 개발하였으며, **C# 프로그래밍 언어**를 이용하여 스크립트를 작성하였다. 게임 개발을 진행하였던 하드웨어 운영체제는 **Windows10**이며, **AMD Ryzen 5 1500x CPU**를 장착하였고, **RAM 8GB**와 **GTX 1050**을 장착한 하드웨어에서 개발을 진행하였다.

## IV. 결론

본 논문에서는 3D 탈출게임을 어떠한 기능들을 이용하여 개발을 하였는지 설명하였다. 비록 복잡한 알고리즘은 사용하지 않았지만 최대한 효과적으로 유니티 엔진의 기능을 이용하여 게임을 제작하였고, 이러한 기능들을 익히고 개발을 진행한다면 초심자라하더라도 큰 어려움 없이 게임 제작을 할 수 있을 것이다.

본 논문에서는 무난히 활용할 수 있는 기술들을 활용하여 게임 개발을 진행 하였다. 향후 업데이트 방향으로는 일정 범위내로 들어오면 플레이어를 추적해오는 적 캐릭터 구현, 인터페이스 개선, 게임 배경 테마 추가, 스테이지 별 게임 배경 제작을 할 것이다.

## REFERENCES

[1] Joe Hocking, *Unity in Action: Multiplatform game development in C#*, Manning Publications; 2 edition (April 25, 2018)  
[2] <http://www.devkorea.co.kr/>