

## 쿼터 뷰 형식의 FPS 생존 게임 개발

윤신영\*, 황어진\*, 김수권<sup>o</sup>, 안성욱\*

<sup>o</sup>배재대학교 게임공학과,

\*배재대학교 게임공학과

e-mail: kimsk@pcu.ac.kr<sup>o</sup>

### Design of FPS Survival Game Based on Quarterly View

Shin Young Yun\*, Eo Jin Hwang\*, Soo Kyun Kim<sup>o</sup>, Syungog An\*

<sup>o</sup>Dept. of Game Engineering, Paichai University,

\*Dept. of Game Engineering, Paichai University

#### ● 요약 ●

본 논문에서는 유니티 엔진을 이용하여 쿼터 뷰로 진행되는 FPS 생존 게임으로 수 많은 적들 속에서 생존하는 것을 게임 개발의 목표로 한다. 제안 게임은 플레이어가 총을 가지고 적을 자유롭게 제압할 수 있도록 개발한다. 또한 다양한 종류의 게임과 공략을 접해본 플레이어들에게는 큰 장점으로 작용 할 수 있다.

**키워드:** 생존 게임(Survival), 쿼터 뷰(Quarterly view), FPS(First-Person Shooter)

## I. Introduction

본 제안 게임은 쿼터 뷰(Quarterly view) FPS(First-Person Shooter) 게임으로 플레이어의 조작방식에 초점을 맞춘 모바일 기기에서 작동하도록 한다. 모바일 기기의 장접중 하나인 간단한 터치 조작법을 이용하여 사용자가 조작에 있어서 불편함을 느끼지 않도록 개발하며, 게임의 진행에 있어서 막힘이 없도록 개발한다. 특히 플레이어가 모바일 게임을 진행함에 있어 쉽게 지루해지는 것을 막기 위하여 쿼터 뷰(Quarterly view) 형식의 장르를 선택하였으며, 이는 사용자가 적으로부터 생존 하는 것에 있어서 큰 장점을 가지고 있다.

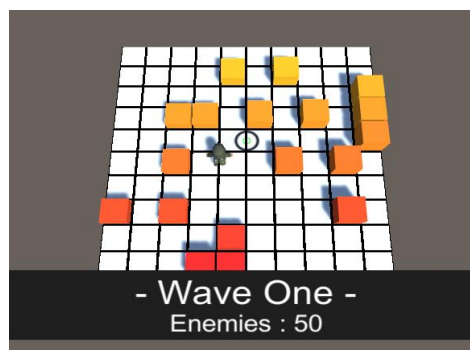


Fig. 1. Game UI

## II. Preliminaries

### 1. User Interface

게임화면(UI, User Interface)은 사용자가 플레이를 하기 쉽도록 하였다. 그림 1에서 현재 플레이어의 크로스헤어를 알 수 있으며, 하단에는 현재 게임의 웨이브와 적의 숫자를 알려주며, 적을 모두 소탕 시 플레이어가 다음 웨이브에서 진행 할 수 있도록 하였다. 그림 2와 같이 게임 내 타일에서 빨간불이 세 번 깜빡이게 하여 플레이어가 적이 어디서 리스폰이 되는지 쉽게 알 수 있도록 하였다.

[1][2]

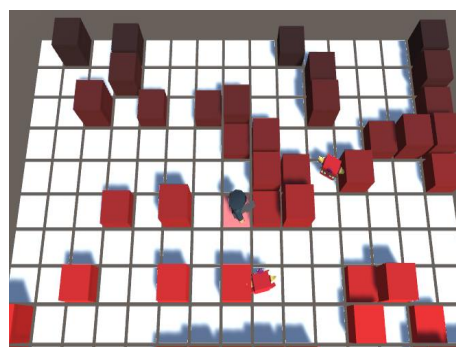


Fig. 2. Enemy Respawn UI

## 2. Quater View

모바일 게임은 화면이 작아 조작이 어려운 부분이 있다. 이런 단점으로 벗어나기 위한 방법으로 쿼터 뷰 방식을 선택하여 게임을 위에서 내려다보며 조작 할 수 있도록 하였다. 또한 플레이어가 게임에 재미를 붙이기 위하여 적으로부터 다양한 무기들을 드랍하여 난이도가 다 플레이어가 원하는 대로 무기들을 바꿔가며 플레이를 할 수 있도록 하였다.

## 3. Object Destroy

라운드 마다 생기는 장애물과 타일 오브젝트는 라운드 마다 파괴되고 생성되도록 하였다. 또한 충돌 발생할 때 생기는 총알과 탄피 오브젝트는 장애물이나 몬스터에게 닿았을 경우 나 닿지 않을 경우 몇 초 뒤에 파괴가 되도록 하였다. 적 몬스터 또한 체력이 방전되었을 경우 오브젝트가 파괴되도록 하였다. 이렇게 함으로써 게임 내의 용량이 커지는 것을 방지함으로써 게임을 원활하게 진행 할 수 있게 하였다.

```

while(percent <= 1)
{
    if(percent >= 0.5f && !hasApplieDamage)
    {
        hasApplieDamage = true;
        targetEntity.TakeDamage(damage);
    }
    percent += Time.smoothDelta
        * attackSpeed;
    float interpolation = (-Mathf.Pow(percent ,2)
        + percent) * 4;
    transform.position = Vector3.Lerp
        (originalPosition , attackPosition , interpolation);
}
    
```

Fig. 3. Enemy AI

## 4. Monster

적 유닛인 몬스터의 인공지능(AI, Artificial intelligence)은 처음 생성 되었을 경우, 유니티의 네비게이션(NavigationMash) 기능을 이용하여 지정된 맵 범위 내에서 무작위로 생성이 되며 생성 되었을 경우 플레이어를 향하여 오도록 설정하였으며, 또한 플레이어가 한 장소에서 가만히 있을 경우 게임의 난이도가 낮아짐을 막기 위하여 플레이어가 한 장소에 몇 초 이상 머무를 경우 그 장소에 몬스터가 리스폰 되도록 하였다. 몬스터가 플레이어의 일정 범위 안으로 들어왔을 경우 몬스터가 플레이어를 향하여 몸통박치기를 하여 공격하도록 하였다.

## III. Conclusions

쿼터 뷰 렌 생존 게임의 진행에 있어 플레이어가 지루하게 느낄 수 있는 반복되는 라운드 방식을 보완하기 위해 라운드 마다의 맵의

크기 와 장애물 오브젝트의 위치와 장애물 양에 변화를 주는 방법을 사용하였다. 또한 플레이어가 적을 처치하고 나오는 회복 탭과 다양한 무기 드랍을 통하여 플레이어가 생존하는데에 있어서 다양한 방법을 구사 할 수 있도록 하였다.결과적으로 플레이어는 반복되는 행동에도 매 라운드 마다의 변화를 통해 게임에 재미를 붙일 수 있도록 설계하였다.

## REFERENCES

- [1] <https://docs.unity3d.com/ScriptReference/index.html>
- [2] <https://assetstore.unity.com/account/assets>
- [3] <https://blogs.unity3d.com/kr/>