스토리 형식의 RPG 게임 개발

윤대현*, 김수균O, 박동원*
O배재대학교 게임공학과,
*배재대학교 게임공학과
e-mail: kimsk@pcu.ac.krO

Development of a Story Based on RPG Game

Dae Hyun Yoon*, Soo Kyun KimO, Dong-Won Park*
ODept. of Game Engineering, Pai Chai University,
*Dept. of Game Engineering, Pai Chai University

● 요 약 ●

본 논문은 유니티 엔진을 이용하여 3인칭 육성 게임을 개발하는 것을 목표로 한다. 본 개발 게임에서는 인간족과 오크족으로 나누고, 인간족은 오크족에게 오크족은 인간족에게 대응하기 위해서 자신의 능력을 키우는 것을 기본 배경 시나리오로 계획하였다. 두 종족의 선택에 맞는 스토리 형식의 퀘스트(Quest)로 중립 몬스터를 격파하여, 이를 능력치로 키울 수 있게 하였다. 본 개발 논문에서는 스토리성 퀘스트를 기본으로 하고 있기 때문에 플레이어가 몰입 하기 쉬운 좋은 조건을 장점으로 하고 있다.

키워드: 전쟁(War), 유니티(Unity 3D), 육성(Promote), 3인칭(Third Person Shooter), 몬스터(Monster)

I. Introduction

본 개발 게임의 플랫폼은 PC에서 게임을 할 수 있도록 개발한다. 플레이어가 쉽게 플레이 할 수 있도록, 다른 RPG(Role Playing Game) 게임의 조작 방법과 비슷한 형식으로 설정하였으며, 플레이어가 최대한 빠르게 게임에 적응 할 수 있도록 개발한다. 각 종족마다 스토리 퀘스트가 다르고 스킬형식도 다르게 설정하였다. 이와같은 스토리 형식 퀘스트는 플레이어가 몰입을 할 수 있게 만드는 것이 장점이고, 퀘스트의 보상을 통해서 자신의 캐릭터가 점점 강해지고 있다는 성취감을 달성 할 수 있도록 해준다[1][2].

II. Preliminaries

1. User Interface

게임화면(UI, User Interface)은 Fig. 1과 같다. 본 개발 게임에서는 플레이어의 체력이 한 눈에 들어 올 수가 있게 배치한 것이 특징이다. 이는 플레이어가 퀘스트를 수락 할 수 있고, NPC를 쉽게 찾을 수 있게 NPC(Non-Player Character)의 상단에 느낌표를 주어 퀘스트를 있다는 것을 알려주게 된다. 그리고 하단 중앙에 보이는 슬롯은 스킬을 담아서 원하는 스킬을 자유자재로 등록을 하여 사용 하도록 하였다.

Fig. 1. User Interface

III. The Proposed Scheme

1. Monster Create&State

본 게임 개발에서는 몬스터를 생성하고, 몬스터가 인공지능을 갖고 움직일 수 있도록 한다. 몬스터의 생성은 오브젝트 풀링(Object Pooling)[3]을 이용하여 메모리의 낭비를 줄이기 위하여 몬스터가 죽으면 다시 그 몬스터를 재활용 하는 형식으로 개발하였다. 몬스터의 상태는 기본, 쫓기, 공격, 돌아가기, 죽었다, 죽는중, 죽음이 끝남과

같이 총 7가지의 상태를 두어 각각의 상태에 맞게 행동을 하도록 한다.

```
 else if (skeleton_PS == Skeleton.Chase)
        {
            slider.value = SkeletonHpState();
            Skeleton_OnDamage = false
            Skeleton_Attack_Distance();
            Skeleton_Distance();
           Skeleton_Anim.SetBool("SkeletonWalk",
true);
            Skeleton_Die();
        }
```

Fig. 2. Chase State Code

```
public enum Skeleton
{
    None,
    Idle,
    Chase,
    Attack,
    Return,
    Die,
    Dying,
    DieEnd
};
```

Fig. 3. Monster State Code

2. Omnibus

한 이야기의 구성으로 이루어지지 않고 옴니버스(omnibus)형식의 스토리는 플레이어에게 흥미와 재미를 충족 시켜주고 퀘스트를 수락할 때마다 매번 다른 몬스터의 사냥으로 반복적인 몬스터의 사냥을 줄여주고 플레이어의 지루함을 방지한다.

3. Experimetal Environment

본 제안 게임은 유니티 버전 2019.2.4.f 및 C#언어를 통해 구현하였고, 테스트 환경은 그래픽카드 (1060 6GB), 메모리 16GB, 라이젠 2600X에서 구현하였다.

IV. Conclusions

게임을 하는데 있어서 제일 중요한 부분은 플레이어가 조작법을 쉽게 익히고 게임의 플레이에 있어 쉽게 이해시키는 문제이다. 그 다음 문제는 흥미와 재미를 추구하는 것이다. 본 개발 게임은 사용자의 편의성을 보장시키며, 다양한 옴니버스(omnibus)형식의 퀘스트를 통해 플레이어를 지루하지 않게 해주었다. 따라서 플레이어가 이 게임의 조작법에 대하여 쉽게 받아드리고 흥미와 재미를 붙일 수 있게 제작한 것이 특징이다.

REFERENCES

[1] Singleton-pattern, http://lonpeach.com/2017/02/04/unity3d-singleton-pattern-example/

[2] Singleton-pattern, https://ko.wikipedia.org/wiki/%EC%8B%B1%EA%B8%80%ED%84%B4_%ED%8C%A8%ED%84%B4

[3] Object Pooling, https://learn.unity.com/tutorial/object-pooling