

Real Time Simulation과 횡스크롤을 이용한 전략 게임 개발

최준석*, 김석훈**, 안성욱*, 김수균⁰
배재대학교, 게임공학과⁰
배재대학교, 게임공학과*
배재대학교, 전자상거래학과**
e-mail: kimsk@pcu.ac.kr⁰

Development of Strategy Game using Real Time Simulation and Side Scrolling

JoonSeok Choi*, Seok-Hun Kim**, Syungog An*, Soo Kyun Kim⁰
Dept. of Game Engineering, Paichai University⁰
Dept. of Game Engineering, Paichai University*
Dept. of E-Commerce, Paichai University**

● 요약 ●

제안 방법은 유니티 5엔진을 이용하여 횡스크롤 실시간 전략(RTS, Real Time Simulation) 게임은 여러 공격로에서 상대방과 서로 여러 거점을 점령하여 실시간으로 전략을 거루는 게임의 개발을 목표로 한다. 제안 게임은 횡스크롤의 장점인 조작법의 단순함을 살리며 플레이어가 실시간으로 전략을 거룰 수 있도록 개발 한다. 게임을 처음 접하는 플레이어와 게임을 많이 플레이하여 숙련도가 높은 플레이어 양 쪽에게 모두 장점으로 작용 한다.

키워드: 전략 게임(Stratgy Game), 전쟁(War), 횡스크롤(Side scrolling), 유니티(Unity 3D), RTS(Real Time Simulation)

I. Introduction

횡스크롤 게임은 기본적으로 좌우로 움직이는 게임으로서 보통 실시간 전략을 거루는 RTS에 적합하지 않다. 제안 방법은 이런 횡스크롤 형식의 단점인 단순함을 덜고 플레이어의 조작법을 최소화하여 난이도가 낮다는 장점을 챙긴 게임을 개발 한다. 간단한 조작을 장점으로 삼는 게임임에도 불구하고 모바일 플랫폼이 아닌 PC 플랫폼에서 작동하도록 한다. 게임의 진행시간이 길고 집중력을 발휘해야하는 전략게임이기에 짧게 즐기는 모바일게임이 아닌 PC 플랫폼에 초점을 맞췄다.

미니 맵을 통해 현재 관리하는 공격로의 대략적인 상황을 볼 수 있도록 하였다[1][2].

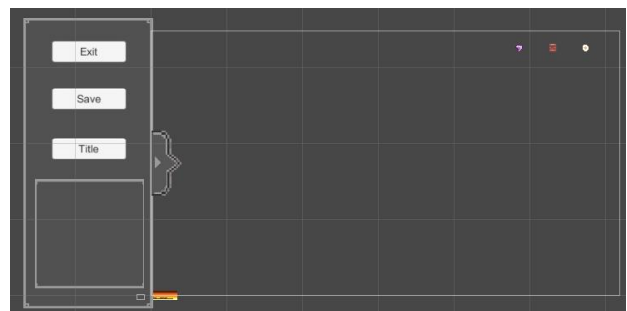


Fig. 1. Game UI

II. The Proposed Scheme

1. 화면 구성

게임화면(UI, User Interface)은 최대한 단순하고 신속하게 필요한 정보를 획득할 수 있도록 하였다. 그림1에서와 같이 현재 플레이어가 획득한 자원의 양을 표기하였으며, esc키를 누름으로서 좌측으로부터 새로운 UI가 나타나도록 표현했다. 게임을 혼자 할 경우 좌측으로부터 나오는 UI창이 나올 경우 게임을 일시정지 하도록 하였으며 UI창 내부의 상단에는 게임 종료나 저장과 같은 시스템버튼, 하단에는

2. 유닛 인공지능

유닛의 인공지능(AI, Artificial intelligence)은 그림 2에서 볼 수 있듯이 생성 시점부터 무작위 패턴을 일정 시간마다 진행하도록 설정하였다. 플레이어가 직접 유닛들과 상호작용을 할 때까지 진행되

도록 되어있으며, 상호작용을 멈출 경우 다시 원상태로 돌아가 무작위 패턴을 실행하도록 하였다. 전투유닛은 3가지, 생산유닛은 4가지의 무작위 패턴을 가지도록 설정하였다. 그림 3, 4에서와 같이 전투유닛은 인식 범위 내에 적 유닛이 발견되었을 경우 자동으로 그 유닛으로 달려가 적 유닛과 전투가 펼쳐지도록 하였으며, 반대로 생산유닛은 사령부까지 도망치도록 설정하였다[3].

```
IEnumerator RandomMove{
    randomPattern = random.Range(0, 3);
    if(randomPattern == 0) //0일 때 움직임
        animator.SetBool("Move", true);
    else //1, 2일 때 멈춤
        animator.SetBool("Move", false);
    yield return new WaitForSeconds(4f);
    StartCoroutine("RandomMove");
}
```

Fig. 2. Random pattern

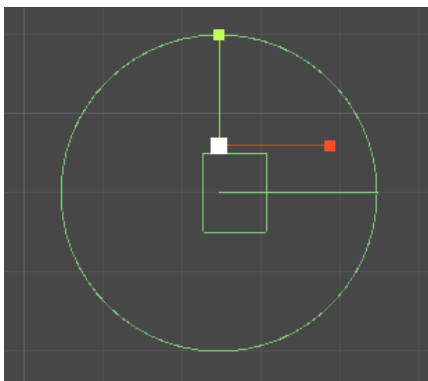


Fig. 3. Unit Collider

```
void OnTriggerStay2D(Collider2D other){
    if (other.gameObject.tag == "enemy"){
        if (일정 범위){
            animator.SetBool("attacking", true);
            movePower = 0f;
            animator.SetBool("isMoving", false);
            return; }
        animator.SetBool("attacking", false);
        animator.SetBool("isMoving", true); }
}
```

Fig. 4. Find Enemy

3. 전략성

게임을 진행하면 할수록 플레이어가 게임에 흥미를 느끼지 못하게 되고 지루함을 쉽게 느낀다. 횡스크롤 게임은 이의 정도가 심한 편이며, 단점을 극복하기 위해서 전략을 다양화 시킬 수 있도록 플레이어가 보지 못하는 여러 공격로를 형성하였다. 단면으로 보는 횡스크롤

게임이 입체적인 전략을 사용 하고, 플레이어에게 보다 전략적인 플레이를 요구하게 된다.

III. Conclusions

횡스크롤 형식의 가장 큰 단점인 단조로운 패턴으로 인한 지루함을 보완하기 위해 플레이어가 보지 못하는 복수의 공격로를 생성하는 방법을 사용하였고, 플레이어는 다양한 전략을 구사할 수 있게 하였다. 결과적으로 게임에 익숙하지 않은 신규 플레이어는 간단한 조작법으로 쉽게 게임을 접하게 되며, 게임의 속련도가 높은 플레이어 또한 매 판 다양한 과정과 결과로 인해 여러 번 플레이 하더라도 지루하지 않도록 설계하였다.

REFERENCES

[1] <https://docs.unity3d.com/ScriptReference/index.html>
 [2] Terry Norton, Learning C# by Developing Games with Unity 3D Beginner's Guide, Packt Publishing, September 25, (2013).
 [3] https://m.blog.naver.com/gold_metal/220888554348