|  |
| --- |
| **BrokeStory**  게임16기 3조 |

(한국정보기술연구원) 게임컨텐츠 팀 프로젝트 보고서

조장: 오주현

팀원: 조민 , 서호준

목차

[[1] 게임 소개 3](#_Toc487817959)

[[2] 게임의 목적 3](#_Toc487817960)

[[3] 게임의 목표 3](#_Toc487817961)

[[4] 개요 3](#_Toc487817962)

[[5] 핵심 플레이 싸이클 4](#_Toc487817963)

[[5-1] 게임 대전 구조 4](#_Toc487817964)

[5-2] 플레이어 스킬 사용 & 보스 패턴 4

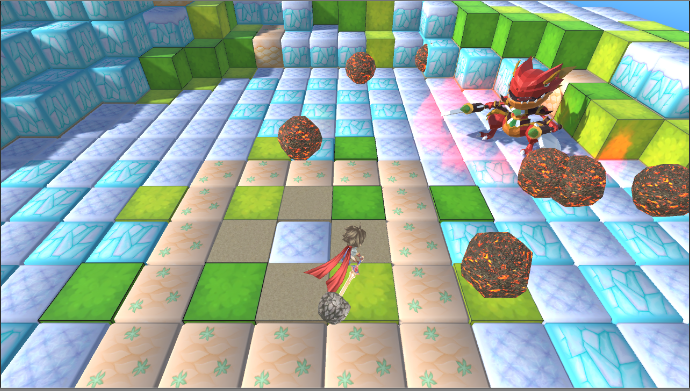
[[6] 게임 조작 방법 6](#_Toc487817967)

[[6-1] 게임 조작 설명 6](#_Toc487817969)

[[7] 게임 로직 구현 설명 7](#_Toc487817983)

# [1] 게임 소개

액션 RPG게임



# [2] 게임의 목적

지극히 평범한 RPG로써 플레이어가 보스와 대결을 펼치는 액션 RPG게임이다.

# [3] 게임의 목표

게임 시작시 플레이어와 보스몬스터가 생성되며 보스를 클리어 하면 게임이 종료된다.

# [4] 개요

|  |  |
| --- | --- |
| 항목 | 내용 |
| 제목 | BrokeStory |
| 장르 | RPG |
| 주요 타켓층 | 캐쥬얼한 느낌의 RPG를 좋아하는 유저. |
| 플랫폼 | PC |
| 이용등급 | ALL |

# 

# [5] 핵심 플레이 싸이클

핵심 플레이의 구조를 설명한다.

## [5-1] 게임 대전 구조

|  |  |
| --- | --- |
| 순서 | 설명 |
| 타이틀 | 전투화면으로 진입하기전 타이틀 화면 |
| 전투화면 | 타이틀화면에서 비동기화 로딩을 거쳐 전투화면 진입 |
| 스킬 사용 | 플레이어는 기본 조작외에 스킬을 사용할 수 있다 |

## 

## [5-2] 플레이어 스킬 사용 & 몬스터 패턴

몬스터 패턴

플레이어 스킬

|  |  |
| --- | --- |
| 순서 | 설명 |
| 플레이어 스킬 | 플레이어는 기본 조작 외에도 스킬을 시전할 수 있다.. |
| 몬스터 패턴 | 몬스터는 간단한 3가지의 패턴을 반복적으로 사용한다.. |





# [6] 게임 조작 방법

방향키 + WASD를 통해 조작 가능하며 스페이스바를 통한 점프와 Z키를 통한 일반 공격이 가능하다.

### [6-1] 게임 조작 설명



|  |  |
| --- | --- |
| 구분 | 설명 |
| 기본 조작 | 기본 조작은 방향키 외에 WASD를 통해 가능하며 스페이스바를 누르면 점프를 할 수 있다. |
| 스킬 | 스킬은 1번을 누르면 작동한다. |
| 공격 | 공격은 Z 버튼 Z를 누르고 타이밍에 맞게 한번 더 누를 시 연타 공격이 가능하다. 점프는 스페이스바로 작동한다. |

# 

# [7] 게임 로직 구현 설명

|  |  |  |
| --- | --- | --- |
| 분류 | 기능 | 구현 로직 설명 |
| 맵tool | 맵 제작 | 맵은 타일맵 방식이고 유니티 에디터를 활용하여 인스펙터창에서 간단한 버튼클릭과 값입력으로 맵을 제작할수있는 맵툴을 만들었다.  타일들을 생성하는 MapGenerator 클래스와 각각의 타일들의 개별적인 추가 속성과 기능을 위한 Tile 클래스 2개로 나눠 만들었다.. |
| MapGenerator | 원하는 맵의 값 입력 | 맵의 사이즈는 맵의가로(x),세로(y) 값을 가지고있는 Coord라는 이름의 구조체를 만들어 입력받았고 타일사이즈,쌓는타일의수도 정할수있게하였다.  각각의 타일 종류들은 Enum형값으로 구분하여 사용하였다. |
| 타일생성 & 데이터 관리 | 모든 종류의 타일은 시작할때 전부 로드해와 각각의 타일종류 Enum형 값을 키값으로하여 DicTilePrefab이라는 Dictionary에 저장하여 원본타일프리팹을 관리하였다.  타일생성은 입력받은 맵의크기 Coord 구조체안의 x와 y의 크기만큼 이중 for문으로 돌며 생성을하고 맵의좌표를 실제 생성할 위치값으로 변환해주는 메서드를 만들어서 타일의 위치를 정해주었다.  현재 생성되있는 타일들을 확인하고 접근하기 용이하기 위해 생성된 타일들은 생성즉시 tileList라는 List안에 넣어서 관리하였다. |
| BossDragon | BOSS | 피격처리 부분은 코루틴으로 동작하고 피격후 1초동안은 또 다른 피격을 받을수없게 bool값과 시간값으로 새로운 처리를 딜레이 시켰고 피격시 애니메이션 변화와 AI동작의 일시적인 멈춤과 파티클 생성까지 피격 메서드에서 동작시켰다. |
| AI | 보스AI는 생성되면 플레이어를 타겟으로 지정하고 플레이어와의 거리 차이로 추격할지 공격할지를 판단하여 반복 계속해 실행한다.  보스AI는 시간을 계속 체크하여 지정된 시간마다 스킬을 발동하는데 스킬은 발동전 경고단계, 이펙트 처리로 시각적인 경고를 주며 3초동안 대기한 후 360도를 회전하면서 지정한 투사체의 개수만큼 일정한 간격으로 생성한다. |
| DragonSkill | 보스의 Skill로 생성된 투사체는 각각 개별적으로 시간을 체크하고 시간에 의해 바닥에서 날라갈 높이로의 상행, 단조로운 패턴을 깨기위한 투사체들의 흐트러짐 단계, 각각의 지정된 방향으로의 발사단계로 나뉘고 오브젝트가 충돌이 없으면 무한히 쌓이기 때문에 입력된  시간후에는 제거되게 하였다. 생성후 입력해둔 잠깐의 시간후부터는 플레이어와 충돌시 투사체는 플레이어에게 피격 신호를 보내고 이펙트를 생성하며 제거된다. |
| BossPig | AI | 기본적으로 NavMesh를 사용하며 타겟을 검사하여 일정 반경내에 플레이어가 감지 될 경우 기존 Idle 상태에서 Move상태로 상태변화를 일으키며 플레이어를 추적한다. |
| Parttern | 전처리과정으로 누적된 시간값을 계산하여 3가지의 스킬을 반복 처리 하며, 한번 시전한 스킬을 일정 시간안에 시전하지 못하도록 불값을 통해 진입하지 못하게 막았다. 그 후 일정 시간이 지나면 해당 불값들을 전부 true 상태로 바꿔주며 시간을 초기화 시켜 다시 처음 스킬부터 시전하도록 만들었다. |