

Programmer Portfolio



김성혁

email :: rotten_bell_pepper@naver.com

URL :: https://github.com/pima86/TurnBase_TCG

A thick red vertical bar runs down the center of the slide. A white circle is positioned on this bar, containing the number 01.

01

프로젝트 소개

A thick red horizontal bar is located at the bottom center of the slide.

01 게임 장르



Risk of Lain 게임을 참고하여 기획하고 Unity 2D(URP)에서 C# 및 UGUI로 진행했습니다.

TurnBase_Rpg

2달동안의 1인 개발 프로젝트

유닛 간의 거리로 행동 가능 여부를 판단하는
엑스콤라이크 방식의 SRPG를 횡스크롤로
변형한 다음 TCG 방식과 병합했습니다.

진행 순서는 플레이어가 고유 덱을 생성하고
몬스터의 패턴을 파악하면서 처치 및 승리하는
게임입니다.



Story

스테이지 순으로 진행되며 순서대로 스토리 씬이 플레이됩니다. 대화하고 있는 상대와 장소가 나타나고 플레이어에게 1~3가지의 선택지를 제시해주며 진행합니다. 선택지를 결정하는데 타이머가 존재하며 답변이 없는 플레이어를 재촉하거나 침묵으로 인식하는 등의 상황을 연출했습니다.

Card

카드에는 필요한 마나 비용, 공격력, 수비력, 특수효과, 사거리, 공격방식을 기술했습니다. 특정 수치에 변경이 있으면 상승이면 초록색, 하락이면 빨간색으로 표현했습니다. 드래그를 통해 카드를 사용하며 그 과정에서 LineRenderer를 통해 시각적으로 보여지게 했습니다.

Equipment

몬스터를 처치하게 되면 해당 몬스터의 아이템 목록에서 확률적으로 아이템을 획득할 수 있습니다. 조합 및 장착을 통해 카드 또는 플레이어 캐릭터 강화가 가능합니다.



고민이 많았습니다. 기획한 프로젝트는 혼자 진행하기에 다소 큰 프로젝트였습니다만, 제게 졸업을 앞둔 마지막 프로젝트였기에 평소보다 더 개발에 욕심을 갖고 도전하기로 결심했습니다.

매월 초마다 한 달간의 계획을 캘린더를 활용하여 정리했습니다. 이를 통해 작업 사이의 간격을 줄이고 개발 과정에서의 망설임 없이 2달이라는 짧은 기간으로 완성해낼 수 있었습니다.

게임 내 데이터는 Script Object와 Json을 통해 Prefab 내의 값을 조정해주는 등으로 관리했습니다. 추후 게임 내 밸런스나 데이터 교체 등을 염두에 두어 이처럼 개발을 진행했습니다.

게임의 분위기를 살리기 위해서 Shader Graph와 TextMesh, URP 기능을 학습하고 이를 활용했습니다.



Shader Graph와 TextMesh
URP 활용



Script Object 및 Json
데이터 관리

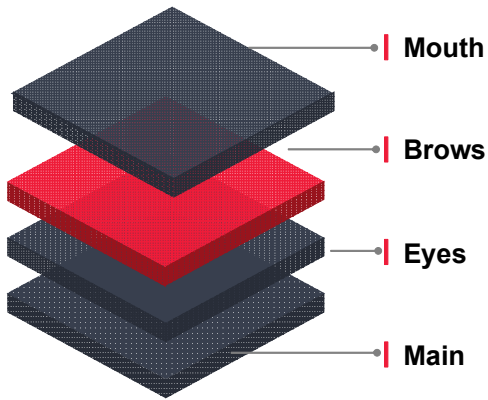


프로젝트 기획하기
계획적인 개발 진행



02

개발 결과



캐릭터의 SpriteRenderer를 부분별로 나눠 이를 쌓아 올리는 방식으로 생성하여 상황에 따른 표정 연출이 가능합니다.





TextMesh Pro의 ForceMeshUpdate를
통해 Text Mesh를 생성 및 활용했습니다.
각 글자마다의 4개 주소 값을 갖고 좌측과
같은 애니메이션 연출이 가능했습니다.

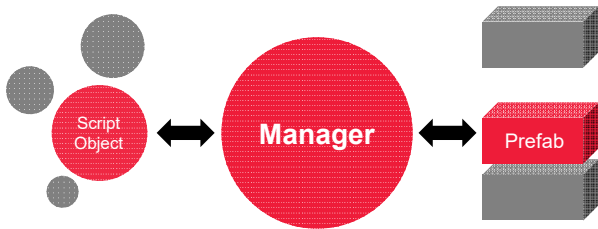




클릭을 통해 장착과 해제를 할 수 있고
상단 조합 테이블 UI를 통해 합성 또한 할 수
있습니다.

조합의 경우 따로 레시피북을 만드는 대신
특정 아이템을 먼저 올려놓으면 조합할 수
있는 아이템들만 강조하는 연출로
표현했습니다.





Prefab을 생성하게 되면 해당 관련 Script Object에서 필요로 하는 값을 요청 및 전달하여 값을 초기화시켜 원하는 객체의 형태로 수정해서 사용합니다.

Script Object는 신규 Prefab의 초기화 외에도 장비 조합 레시피 등에도 활용되었습니다.





각 몬스터 객체들이 플레이어와의 거리를 획득하여 적절한 패턴을 미리 표시해줍니다. 이는 플레이어가 행동을 수행할 때마다 실행하며 최신화시켜줍니다.



03

개발 이후

1. 몬스터간의 행동예측

몬스터의 다음 행동이 다른 개체로 인해 앞마박히는 등의 이유로 방해되어 행동을 수행하는데 문제가 발생했습니다.

= 필드 칸만큼의 배열을 생성하여 몬스터가 예측하면 수행했을 때의 필드 상황을 기록하고 다음 개체에 전달하여 현재 필드 상황이 아닌 전달받은 필드 상황으로 다음 행동을 예측하도록 구현했습니다.

2. 조합법 최적화

배열로 이루어진 조합법에 Find로 특정 아이템을 포함하고 있는 배열을 찾아 처리하는 방식입니다. 해당 방식은 수직적으로 처리 속도가 느려졌고 중복 아이템의 경우에는 오류까지 발생했습니다.

= 검색 조건에 아이템의 유무와 개수를 제시하면서 충족하지 못하는 조합법은 미리 Remove()를 수행해 분량을 줄였습니다.



3. 국내 TMPPro Effect 자료 부족

TMPPro의 효과를 구현하는 과정에서 국내 자료가 부족해 개발에 지장이 있었습니다.

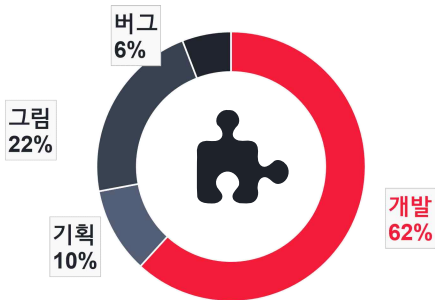
= 국외 자료를 이해하고 예시로 작성된 스크립트를 분석 및 가공하여 원하는 애니메이션을 연출했습니다.

4. 수행 순서

첫 플레이에서는 문제점이 느껴지지 않지만, 적응 후 빠른 속도로 진행하게 되면 카드 사용이 부드럽게 이어지지 않는 문제가 있었습니다.

= 먼저 카드를 사용할 수 있는 상황인지 판단하고 True라면 수행 중인 카드와 차례대로 수행하도록 설계했습니다. 또한 부드러운 사용과 빠른 속도감을 위해 카드 연출을 가능한 빠르게 진행하도록 조정했습니다.

- 일정 분할 비율



전체 일정 : 2달

오픈 소스를 활용하여 그림 작업 시간을 제외했더라면 더욱 퀄리티를 높일 수 있었을 것이라는 아쉬움도 있지만 그렇게 했기에 가장 만족스러운 프로젝트가 되어 힘들었을 과정을 즐기며 수행할 수 있었습니다.

비록 일정을 수행하는 과정에서 부족한 속면으로 고생했지만 일말의 후회 없이 모든 노력을 할 수 있었음에 뿌듯함을 느꼈습니다.



1 불필요한 디테일 욕심

디테일에 과하게 욕심을 내지 않았더라면 게임의 완성도를 올릴 수 있지 않았을까 싶습니다.

2 부족한 UI

게임 플레이 화면과 UI가 잘 섞이지 못하고 겹도는 느낌이 들었습니다. 보다 UI 구조에 관해 연구하고 관찰할 필요가 있다고 느꼈습니다.

3 Unity 활용 능력

Shader Graph과 TMP Pro Mesh와 같이 활용해보지 않은 기능들을 학습해 나아갈 것입니다.

4 튼튼한 설계

버그 수정에 생각 이상의 시간을 소비해야만 했습니다. 더욱 튼튼한 설계를 기획하도록 성장하여 작업 시간을 줄이도록 하겠습니다.





THANKS FOR WATCHING