팀 프로젝트 보고서

17기 5조

조장: 노영진

조원: 오승원

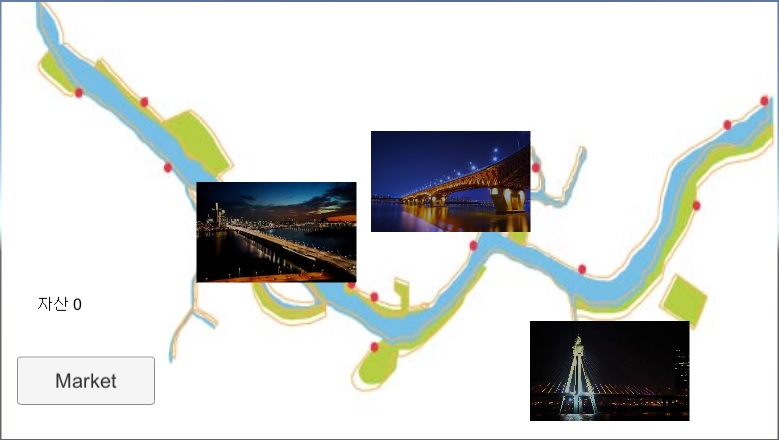
**게임명: GAZUAAAH!!!**

장르: 시세반영 턴제 코인배틀

**게임의 목적**

세간에서 화제인 코인을 소재로 하여, 소유한 코인을 일종의 덱으로 삼고, 코인의 시세를 전투력으로 삼아 전투를 진행하고, 거래소를 통해 코인을 사고 팔아 자신이 소유한 코인덱을 강화시켜 이기는 것이 게임의 목적이다.

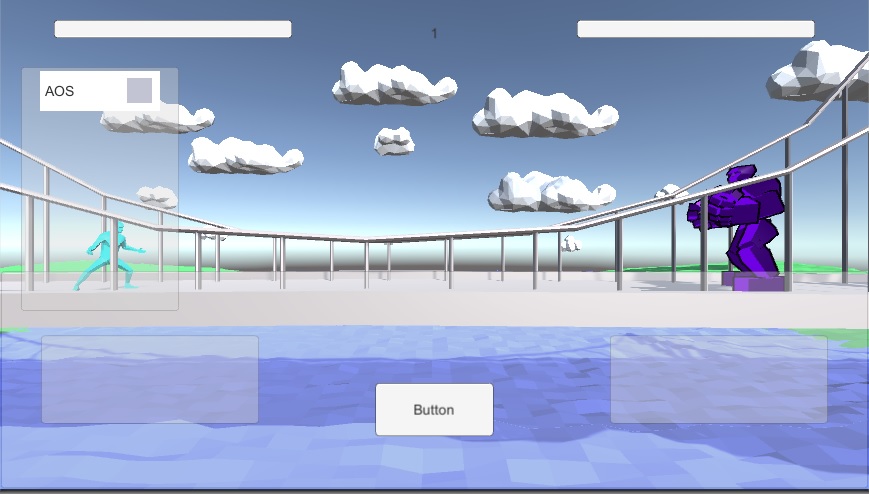
**게임의 흐름**

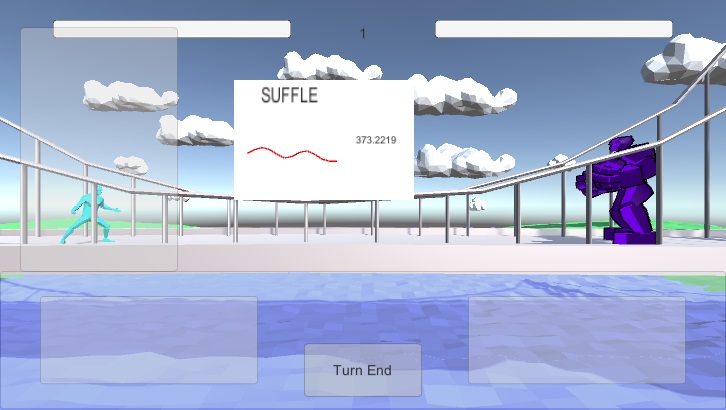
****

게임이 시작되면 대기화면에 들어가게 된다. 대기화면은 한강을 배경으로한 맵으로 되어있으며, 한강의 각 다리 위치에 스테이지가 준비되어있고, 대기화면 좌하단에는 코인을 거래할수 있는 거래소를 보여주는 거래소 버튼이 존재한다.



플레이어는 먼저 배틀에서 사용될 덱을 짜기 위해서 거래소에 들어가 현재 가지고 있는 자금으로 코인을 구입하게 된다. 코인은 시세가 분마다 변동하며, 한번 결정한 거래는 바꿀수 없게 되어있다.

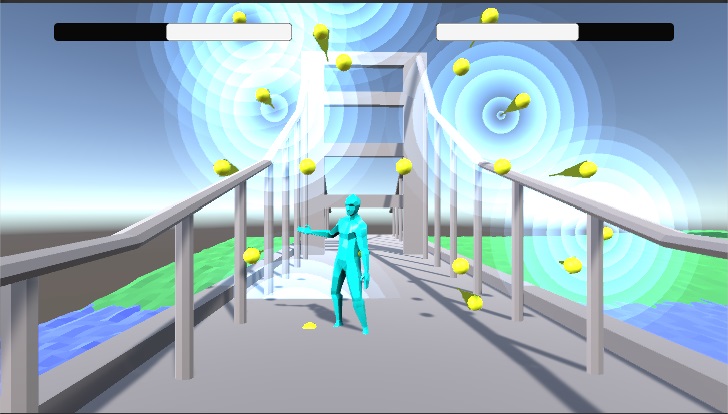


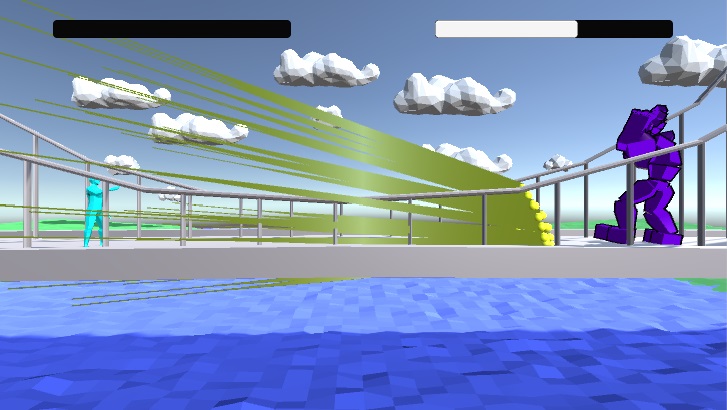


코인을 구입한 후에는 맵상에 구현된 각 스테이지버튼을 눌러 배틀에 돌입한다.

배틀은 자신이 가지고 있는 코인을 공격, 또는 방어 패널로 드래그하여 전투중 공격력과 방어력을 선택할수 있으며, 모든 선택이 끝나고 턴종료를 눌렀을시 각 코인들은 시세변동이 일어나고,

시세변동에 맟춰 전투력에 영향을 주어 전투가 진행된다.





배틀후에는 다시 거래소를 통해 자신의 덱을 강화할수 있다.

**게임조작**

모든 게임상의 조작은 터치 및 드래그로 이루어진다.

**게임 로직 구현**

대기,배틀화면 UI:

유니티 UGUI를 이용하여 기본 구성을 만들었으며, 만들어진 모든 UI들은 프리팹으로 저장, UIManager가 모든 프리팹을 보유하고 있다가, 필요한 시점에서 필요한 UI를 요청해주면 매니저가 알아서 초기화 및 세팅을 완료한다. 모든 UI의 요청은 enum으로 정의되어있으며, switch문을 통해서 각 요청에 맞는 작업을 수행하기 때문에 추가 및 삭제가 용이하도록 만들었다.

드래그&드롭:

공격 및 방어 패널에 코인을 드래그에서 선택하는 드래그 & 드롭은 유니티가 지원하는 idraghandler, idrophandler를 통해서 구현하였다. 특히 드래그를 가능하게 하는 기능은 draggable이라는 클래스로 구현하여, 어떤 UI오브젝트이던간에 draggable을 붙이기만 하며 드래그가 가능하게 구현하였다.

그래프:

유니티내에서 지원하는 linerenderer클래스는 선이 꺾일시 모양이 비틀어지는 경우가 있으므로 그래프의 선을 그리기에는 적합하지 않았다. 그래서 3d화면상에서 선을 그릴수 있는 여러방법중, 유니티가 지원하는 GL라이브러리를 통하여 직접 선을 그리는 코드를 구현하였다.

선을 그리는 기능을 Grapher라는 이름의 클래스로 구현하였고, 자신이 그릴 그래프의 높이값을 리스트로 보유하고있으며, 어떤 UI에서든간에 registgraph함수를 통해 등록하면 그 UI에 그래프가 그려지도록 구조를 만들었다.

코인의 시세변동:

코인의 시세를 단순히 랜덤값으로 주었을 경우, 값이 너무 들쭉날쭉하게 되기 때문에, 랜덤값을 보정해주는 perilnoise를 통해서 일관성을 가지는 랜덤시세를 구현하였다. Perilnoise가 반환해주는 값을 0에서 1사이이므로 실제 가격을 맞추기 위해선 재조정이 필요했으며, 이는 mathf의 inverseLerp를 통해 랜덤값의 퍼센테이지를 구하여, 다시 lerp로 실제가격에 맞게 조정하였다.

코인의 시세변동은 MarketManager가 담당하고 있으며, 언제 어디서나 ChangeMarketInfo함수를 통해서 시세변동을 실행할수 있다.

UI애니메이션:

UI애니메이션의 움직임은 unity가 제공하는 애니메이션 기능을 통해 구현하였다. 애니메이션 윈도우에서 지원하는 녹화기능과 커브를 통해 UI가 등장하는 애니메이션을 작성한뒤, 이를 레거시 애니메이션으로 변환, 코드상에서 UIManager가 UI를 세팅할 때 불러줄수 있도록 하였다.

UI애니메이션을 레거시애니메이션으로 만든 이유는, 레거시 애니메이션은 코드를 통해 역재생이 가능하기때문이다. 시작시간을 애니메이션의 끝으로, 속도를 음수를 주어서 등장애니메이션을 역재생하여 하나의 애니메이션으로 등장과 퇴장을 모두 쓸수있게 하였다.

전투애니메이션: