

## 해보고싶은 기술

AI

잘 만들어진 UI

메뉴/게임 등 전반적인 설계

EVENT PICK

프로그래밍 패턴

부쉬 구현

총알 구현

시야 구현

파괴되는 맵

템플릿을 사용한 범용 클래스

연산자 오버로딩

네트워크

# 브롤스타즈 2D

## 1. 게임 소개



- > 2:2 or 3:3 슈팅게임
- > 오프라인 – 플레이어를 제외한 캐릭터는 AI

## 2. 게임 요소

- > 캐릭터
  - > 캐릭터마다 탄환 / 필살기의 종류가 다름
  - > 탄환 제한 있음

- > 공격 / 피해 / 시간경과 에 따라 필살기 게이지 증가

- > 탄환 범위 표시

## > AI

- > 가장 가까운 상태편을 향해 이동 (A\*)

- > 공격범위에 들어오면 가장 강력한 공격 순으로 공격

- > 필살기 -> 일반탄환

- > 모든 적이 보이지 않을 경우는?

## > 지형지물

- > 맵 외부파일 파싱

- > 파괴가능 지형 / 파괴불가능 지형

- > 부쉬

- > 안에서 공격하지 않으면 비가시

- > 공격시 n초간 가시

## 3. UI

- > 브롤스타즈 메뉴와 완전히 같은 모양

- > 개별 클래스

- > 버튼과 기능 분리

#### 4. 게임 구조

#### 5. 추가 사항

> 네트워크 (서버 / 판정 / 네트워크 메뉴)

#### 6. 진행계획

> 전체 진행도

	8/19	8/20	8/21	8/22	8/23	8/25	8/26	8/27	8/28
설계									
구조									
UI									
게임									
디버그									

	8/29	8/30	9/1	9/2	9/3	9/4	9/5	9/6	9/8
구조									
UI									
게임									
디버그									
추가									

## > 세부일정

### > 구조

- > UI / 게임 분리

- > 네트워크로 소유 정보와 클라이언트 소유 정보 분리

### > 게임

#### > 맵

- > 타일형 맵 - 행렬과 X, Y축의 인덱스 실수 가능성 높음

- > 이펙트 - 피탄 / 발사 등

#### > 캐릭터

- > 이동 - 자유이동 / AI와 공유 가능한 컨트롤러

- > 이동 - 컨트롤러

- > 컨트롤러 - (유저 / AI)

- > 충돌 - 지형지물 / 상대 공격

- > 공격 - 직선 /

#### > 지형지물

- > 지형지물은 모두 사각형으로 판정함

#### > 충돌

- > bool - 충돌 가/불가

- > 가 - 벽 / 아이템 박스 / 장애물(박스, 나무벽 등)

- > 불가 - 부쉬

#### > 파괴

- > bool - 파괴 가/불가

- > 가 - 부쉬 / 장애물 / 아이템 박스

- > 불가 - 벽

- > 부쉬

- > 데이터 관리 (체력 / 레벨 등) - 추후 서버로 발전 가능성 염두

- > UI

- > 로비

- > 인게임

- > 종료

- > (온라인 로비)

## 프로그래밍 시 주의점

### 완성도

- > 완결성 - 메뉴부터 엔딩까지 완전하게 완성
- > UI 완성도
- > 그래픽적 부드러움 / 자연스러움

### 안정성

- > 예외처리 - 에러에 소지가 있는 코드는 전부 예외처리
- > 메모리 관리 - 모든 변수는 생성부터 소멸까지 파악

### 성능

- > 적절한 컨테이너 설정
- > 기능 추가 시 필요성을 고민

### 이식성

- > 다른 게임에서도 그대로 가져다 고쳐쓸 수 있는 UI
- > 역할에 충실한 클래스

### 가독성

> enum / define 사용

> 기능 별로 함수 분리