//캐릭터 부모 클래스. 모든 캐릭터클래스는 이 클래스를 상속받는다.

class Character

{

Public:

Character();

Bool getDamage();

Bool isAttack();

Private:

Int HealthPoint, ManaPoint, AttackPoint, DefensePoint, AttackRange, Stemina;

Bool AttackAvailable;

Point Location;

Enum Rotation;

Enum Class;

Bool setClass;

}

//플레이어 캐릭터 클래스. 플레이어 캐릭터는 자신의 클래스가 가진 스킬을 사용가능하다.

Class PlayerCharacter : Character

{

Public:

PlayerCharacter();

Bool usingSkill();

private:

Bool setSkill();

}

//몬스터 캐릭터 클래스. 일정거리 안에 적이 들어오면 공격한다.

Class MonsterCharacter : Character

{

Public:

MonsterCharacter();

//적 행동 결정

AI();

Private:

//공격 우선 순위 변수

Enum priorityEnemy;

//감지 범위 안에 적이 있는지 확인

checkRange();

//이동

isMove();

isAttack();

checkPriorityEnemy();

}

//맵 구조체. 맵이 가진 정보는 여기에 저장한다.

Typedef Struct map{

Point 좌표;

//층

Int Layer;

//이동 가능한 방향 or 접근해올 수 있는 방향

Enum availableMove;

//맵위에 뭐가 있는가… 변수명 맘에 안들어

Enum State;

}MAP

//게임 초기화

Bool init()

{

makeMap();

각 캐릭터 객체 생성

}

//맵 생성! 한다 하드코딩!

Bool makeMap()

{

setMapProperties();

}

//이놈을 쓰던가 아니면 크…. 뭐 그건 해보고 판단합시다.

Bool setMapProperties(\*MAP map, int Layer, enum AvailableMove, enum State)

{

}

//턴을 받는다

getTurn(enum Turn)

{

턴을 받으면 턴 받은 팀만 움직이는거여! 그런거여!

턴 받은 팀의 행동력 풀로 차고 공격도 가능하게 되는거여 그런거여!

}

turnover()

{

내 턴 끝나면 넘기는 거여!

그런거여!

}

//마우스 왼쪽클릭 이벤트 메소드

Void onMouseLeft()

{

getBoundingBox()를 사용하면 스프라이트 객체의 영역을 rect에 저장할 수 있다.

받아온 스프라이트의 위치를 이용하여 클릭한 위치에 무엇이 있는지 확인하고,

행동을 결정한다.

아군 캐릭터 클릭하면 해당 캐릭터 선택되서 컨트롤 가능하게 적용.

캐릭터 컨트롤 가능 상태에서 빈 타일 왼쪽 클릭시 이동. 공격범위 안의 적 클릭시 공격.

이동 불가타일 클릭하면 아무 이벤트 없음.

적 클릭시 적캐릭터 상태 띄워줌.

}

//마우스 오른쪽클릭 이벤트 메소드

Void onMouseRight()

{

아무것도 선택하지 않은 상태로 돌린다.

}

//키보드 입력 이벤트

Void onKeyboardPressed()

{

esc버튼 클릭 이벤트.

게임 일시정지하고 계속하기, 다시시작, 게임종료 세 개의 버튼을 띄운다.

방향키 누르면 화면 이동.  
}

Class Menu : public layer

{

Public:

//마우스 왼쪽 클릭 이벤트 메뉴 버튼 눌렀는지 확인

onMouseLeft();

Private:

//게임 시작 버튼 누르면 실행, 인게임 화면으로 전환

startGame();

//게임 종료 버튼 누르면 실행, 종료하자

exitGame();

}