

RogueLike

이과대학 지구시스템과학과

2013134017

최민희

게임설명

Scanner 로 사용자의 input : A,W,D,S 를 받아들여 이동하게 되며, . M F @ R W #은 각각 길, 몬스터, 음식, 플레이어, 무기, 벽 을 의미한다. 길을 따라 이동할 수 있고, 벽은 통과할 수 없다. @는 플레이어를 의미하는데 많은 @가 있으면 게임 도중 캐릭터를 바꿀 수 있다. 하지만 초기 설정은 동일하다. HP가 떨어졌는데 보충할 수 있는 더이상의 F(ood)가 없을 경우 유용하다. Default 플레이어는 지도에서 가장 아래에 위치한 플레이어다. W(eapon)는 power를 증가하게 하고, F는 HP를 증가시킨다. R(ock)은 지나칠 수 없지만 R의 파워보다 플레이어의 파워가 크다면 옮길 수 있다. M(onster)는 먼저 공격하지 않으면 공격하지 않으나 모든 몬스터를 처리해야 끝나는 게임이므로 공격해야 이길 수 있다. Monster의 기본 체력과 파워는 랜덤으로 결정된다. 플레이어의 체력이 0이 되면 게임은 자동으로 종료되며, 더이상의 M이 없을 때 Clear 되는 게임이다.

클래스설명

Encapsulation 을 활용하고, 클래스간 구분을 확실하게 하여 깔끔한 정리를 하기위해 상속 관계가 아닌 이상 모두 다른 패키지로 구분하였으며 이를 위해 class들은 전부 public으로 설정하여 패키지가 다르고, 상속하지 않아도 서로 다른 클래스에서 각각의 이름을 사용하여 변수를 선언할 수 있도록 하였다.

1. RogueLike class : main 함수가 실행됨. FileManager 타입의 변수로부터 loadMap, buildMap 함수를 불러들이고, WorldMap 타입의 변수로 부터 initialize, run함수를 불러들인다.

2. FileManager class : 텍스트파일을 하나 받아들여 문자 하나하나를 char배열에 저장한다.

- loadMap method : 텍스트파일을 한 줄씩 읽어 Vector element 하나로 취급하여 Vector 변수 vector에 추가. 패키지가 다른 RogueLike 클래스의 메인에서 쓰이기 때문에 public으로 설정했다.

- buildMap method: 벡터에 저장되었는 String을 문자 배열로 바꾸어 char 2차원 배열 array에 한 줄씩 행을 늘려가며 저장한다. 마찬가지로 패키지가 다른 RogueLike 클래스의 메인에서 쓰이기 때문에 public으로 설정했다.

FileManager 에서 다루는 텍스트 파일은 '. M F @ R W #'가 나열된 파일으로 각각 길, 몬스터, 음식, 플레이어, 무기, 벽 을 의미하게 된다.

3. Config class : 프로젝트에 자주 사용되는 명령어 상수들과 스캐너타입 scan변수를 포함한다. 불러들이는 지도 파일 경로 또한 여기에 저장한다.

4. WorldMap class : main에서 initialize메소드의 부름을 받고 2차원 문자배열을 Entity타입의 2차원 배열으로 전환시켜준다. 정해진 문자에 따라 각각 Monster, Wall, Player, Food, Rock, Weapon 클래스로 객체를 생성하고 자동으로 업캐스팅된다. 정해진 문자가 아닐경우 null로 처리된다. 다시 main에서 run 메소드의 부름을 받고, 게임이 시작된다.

run 메소드에서는 showMap(char[][] 배열을 출력), showStatus(지정된 Entity의 HP, maxHP, power 를 출력) 을 차례로 부르고 showMenu를 통해 명령을 입력받는다. 'Quit' 명령이 들어오지 않을때까지앞의 과정을 반복한다. 현재 플레이어가 가고자 하는 방향에 무엇이 있는지 알아보기 위해 getNext메소드가 사용된다. getNext는 입력받은 방향과 Entity타입을 하나 인자

로 받아서 입력받은 방향으로 한칸 떨어진 곳에 있는 map[][] Entity가 가리키는 객체를 next라는 Entity 타입 변수에 저장해준다. 이 next가 null이면 moveEntity 메소드를 이용해 현재 플레이어가 자리해 있는 곳의 map[][] 변수는 null을 가리키도록, next자리에 있는 map[][] 변수는 플레이어를 가리키도록 해 자리이동을 한다. next가 null이 아니었으면 next가 가리키는 Entity객체의 showMenu 메소드를 호출하여 옵션을 알아본 뒤 마찬가지로 그 객체의 inputCommand 메소드를 호출하여 명령을 받고 실행한다.

while문이 끝나면(Quit이 입력되었을 경우) 자동으로 M이 지도상에 남아있는지 검색하고 없으면 게임클리어를 출력하고 있으면 게임오버를 출력한다.

getNext, showMenu, moveEntity는 inputCommand로 불러진 Entity의 서브클래스 메소드에서 불러지게 되므로 public으로 설정하고, 그 외 함수는 WorldMap 클래스를 상속받는 함수도, 같은 패키지안에 있는 다른 클래스도 없으므로 private으로 설정하였다.

5. Entity class : 이 게임에서 사용되는 존재를 모아둔 것으로, Monster, Wall, Player, Food, Rock, Weapon 클래스의 슈퍼클래스이다. 추상메소드(showMenu, inputCommand)를 가지므로 추상클래스이다. showMenu 와 inputCommand 는 각각 subclass마다 출력해야하는 것과 수행해야하는 것이 다르므로 오버라이딩 해야하기 때문에 슈퍼클래스에서는 추상메소드로 정의하였다. 그리고 패키지가 다른 WorldMap에서 부르는 함수이므로 public으로 정의되었다.

- Player : Entity 클래스를 상속받아, 다른 @를 만났을 때 showMenu메소드가 호출되어 캐릭터를 바꿀 것이냐고 묻는다. 그리고 inputCommand 메소드가 호출되어 'Y'에 해당하는 명령을 받으면 현재 플레이어와 새로운 @를 바꾸고, 새로운 @가 플레이어가 된다. @를 가리키고 있던 것이 Entity 타입의 next였으므로 이것을 Player로 다운캐스팅 한 뒤에 player 변수가 next변수가 가리키는 객체를 가리키도록 하면 된다.

- Rock: 플레이어의 힘이 이 객체의 힘보다 크다면 옮길 수 있는데 Rock의 힘은 생성자가 호출될 때 random함수에 의해 정해진다. 이때 WorldMap class의 moveEntity 함수가 필요하다. 이 함수는 getNext, showMenu와 더불어 다른 subclass에서도 쓰이게 되므로 이를 위해, Entity class 에서는 WorldMap 타입 변수를 하나 가지고 있고, 이 변수를 현재 WorldMap을 가리키도록 하였다. run함수에서 this를 인자로 내려주며, inputCommand에서는 인자로 받은 WorldMap(this)과 Entity class에서 가지고 있는 WorldMap 변수 w 가 같은 객체를 가리키도록 했다. 이를 통해 getNext, showMenu, moveEntity는 전부 public으로 설정되었다. Rock은 그냥 moveEntity를 사용할경우 R이 사라지기 때문에 이를 보존하기 위해 getNext 함수를 한번 더 써서 현재 next의 next를 구하여 R과 플레이어 둘다 이동시켰다.

- Monster : 몬스터의 파워 및 체력은 생성자가 호출 될 때 random함수에 의해 정해 지도록 하였으며, attack 을 하면 체력이 다 떨어질 때까지 계속 inputCommand에서 while함수로 돌아가게 되며, flea를 하면 break된다.

- Weapon, Food: 이들 또한 생성자가 호출될 때 각각 자신의 파워 및 체력이 random함수에 의해 정해지게 되며, 플레이어가 이들을 'E(at)' 하게 되면 플레이어의 파워 및 체력이 올라가도록 설정 되었다. 이들은 'E(at)'후에 moveEntity로 인해 사라지게 된다.



:클래스

