CSED232 Object-Oriented Programming (Fall 2017)

Assignment #2

Inheritance

Due date: 11/12(Sun), 23:59

이번 과제에서는 "어뢰 발사 게임"을 구현한다. 이를 통해 상속에 대해 공부한다.

1. 게임 기본 설명

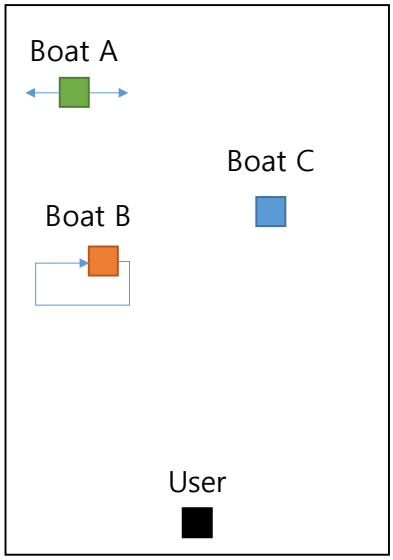


Figure 1 게임 컨셉 이미지

A. 게임의 기본 디자인은 figure 1 과 같다.

B. Screen 의 공간은 바다이며, 그 위에 배들이 떠 있다.

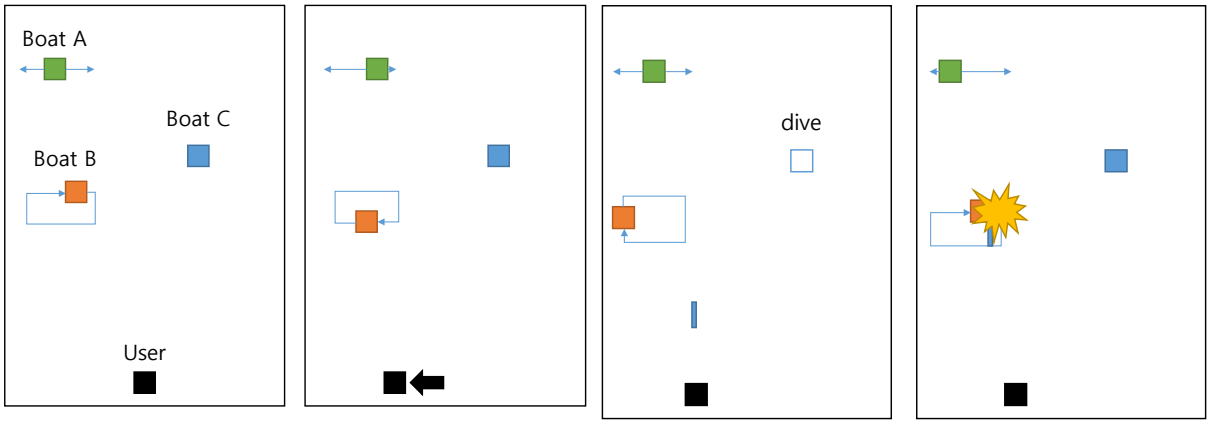
C. 세 가지 종류의 배가 존재하며, 각 배마다 특수한 움직임을 가진다.

D. User 는 좌/우 움직임, 어뢰 발사의 action 을 취할 수 있다.

E. User 가 어뢰를 발사하여 위의 배들을 격침시키는 게임이다.

F. Boat A, Boat B, Boat C 가 규칙적으로 움직이기 때문에 거리와 예상 위치를 잘 예측하여 격침시켜야 한다.

G. 아래 게임 flow 가 그림으로 설명되어 있다.



t = 0
게임 시작

t = 1
유저 이동

t = 2
어뢰발사

t =3
명중

2. 게임 상세 디자인

A. Basic

i. 유저가 입력을 줄 때마다 매 턴이 진행된다.

ii. 매 턴마다 각 배들은 한 턴에 해당하는 움직임을 가져간다 (후술).

B. Sea

i. 게임이 실행되는 바다는 x y 의 2차원 좌표이며, 각 배, 유저, 어뢰는 좌표 상에 존재한다. Figure 2 참고.

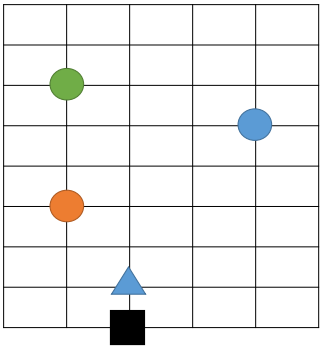


Figure 2

2차원 좌표의 게임 이미지

C. BoatA

i. BoatA 는 주기적으로 좌 우로 움직인다. Figure 3 참고

ii. 아래 움직임 예제 참고

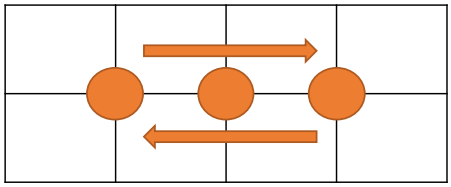
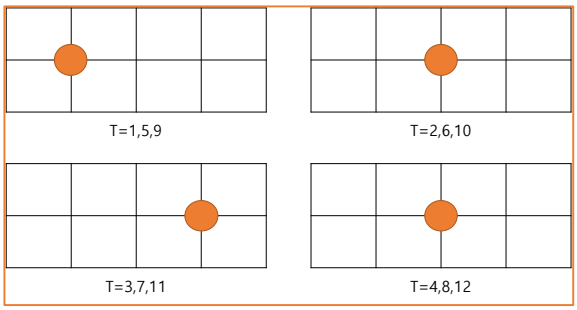
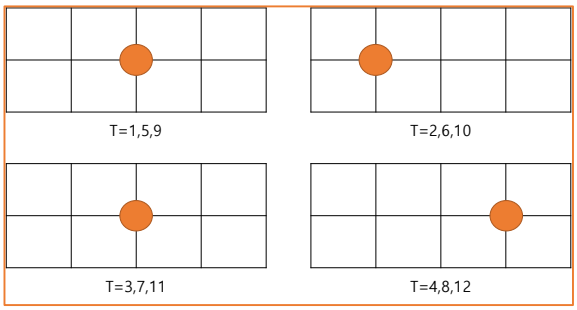


Figure 3 BoatA 의 움직임



Boat A 움직임 예제 1



Boat A 움직임 예제 2

D. BoatB

i. BoatB 는 시계 방향으로 움직인다. Figure 4 참고

ii. 아래 움직임 예제 참고

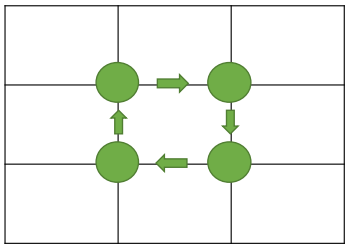
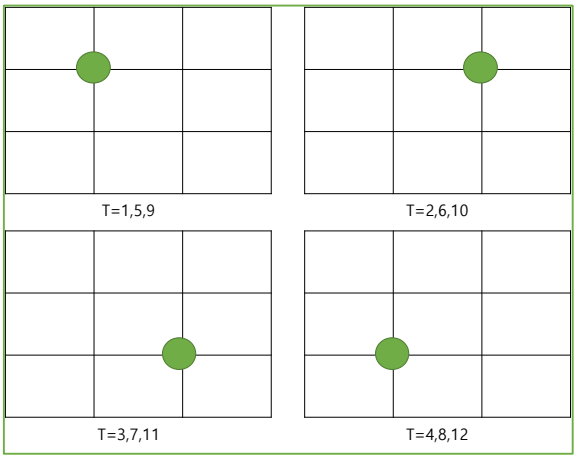
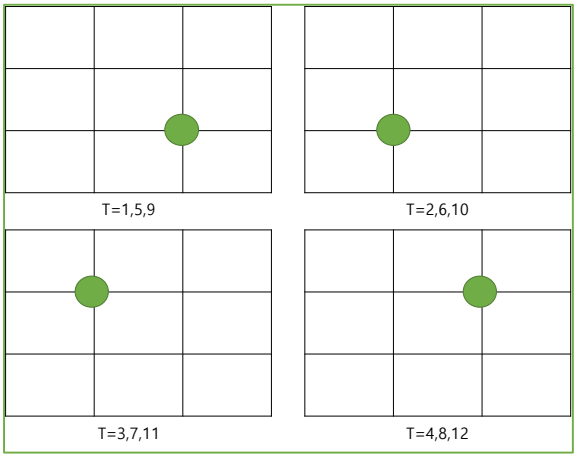


Figure 4 BoatB 의 움직임



Boat B 움직임 예제 1



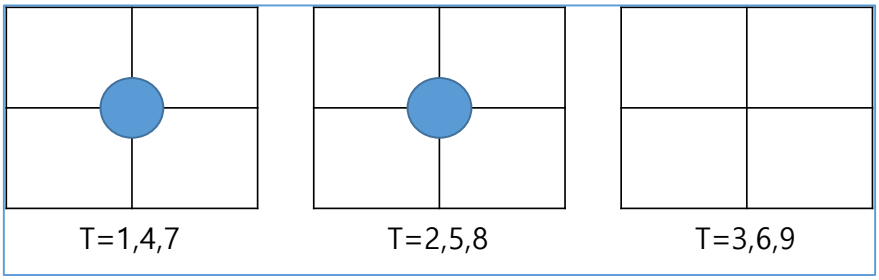
Boat B 움직임 예제 2

E. BoatC

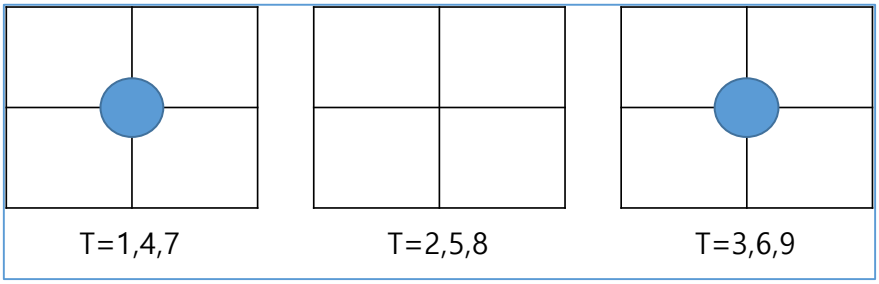
i. BoatC 는 움직이지 않고 주기적으로 잠수한다 (**잠수 시 출력 화면에 나타나지 않으며, 어뢰로 맞출 수 없다**).

ii. 세 번째 턴마다 잠수한다. (ex. 0,3,6 턴 혹은 1,4,7 턴 혹은 2,5,8턴)

iii. 아래 움직임 예시 참고



Boat C 움직임 예제 1



Boat C 움직임 예제 2

F. User

i. 유저의 초기 위치는 정 가운데(x가 짝수일 경우 한쪽으로 치우칠 수 있음), 맨 아래에 위치한다.

ii. 유저의 입력은 w,a,s,d 의 네 가지로 주어진다.

1. W : 어뢰 발사
2. A : 좌로 이동
3. S : 턴 넘기기
4. D : 우로 이동

G. Torpedo (어뢰)

i. 어뢰 발사 입력이 들어올 시 다음 턴에 유저 바로 위에 어뢰가 생성된다. Figure 5 참고

ii. 매 턴마다 어뢰는 위로 이동한다.

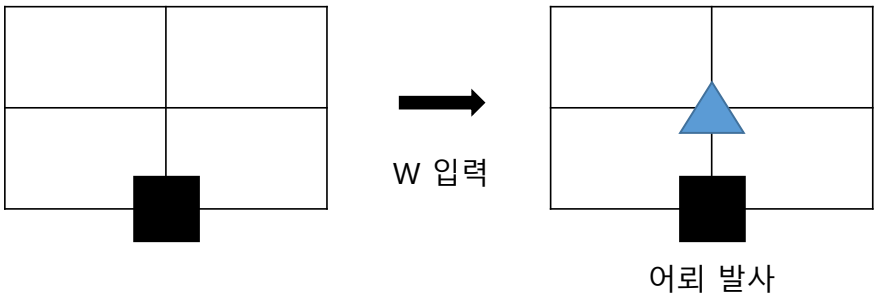


Figure 5 어뢰 발사 예제

H. Difficulty

i. 본 게임은 발사할 수 있는 어뢰의 수와 주어진 최대 턴 수를 기반으로 난이도 조절을 할 수 있다. 예를 들어 어뢰는 전체 배의 1.5배 개수만큼 줄 수 있고, 최대 턴 수는 (배의 수*바다의 너비+바다의 길이) 를 줄 수 있다.

ii. 난이도는 각자 원하는 대로 설정 가능하다.

I. Score

i. 배가 격침되면 점수가 올라간다. 각 배마다 다른 점수를 가진다. 예를 들면 BoatA 는 100점, BoatB 는 200점, BoatC 는 300점을 줄 수 있다.

ii. 남은 턴 수와 남은 어뢰의 수도 점수에 반영한다. 예를 들어 남은 턴 수가 3턴이면 3*500점, 어뢰가 2발 남은 경우 2*400점을 줄 수 있다.

iii. 단, 패배 시 남은 턴 수와 남은 어뢰 수는 점수에 가산될 수 없다.

iv. 점수 배점은 각자 원하는 대로 설정 가능하다.

J. Game Over

i. 게임은 승리와 패배로 종결될 수 있다.

1. 승리조건
    A. 모든 배가 격침될 경우
2. 패배조건
    A. 남은 턴 수가 모두 소진된 경우 혹은
    B. 발사할 수 있는 어뢰가 남아있지 않으며 바다 상에 이미 발사된 어뢰가 남아있지 않은 경우.

K. Etc

i. 새로운 게임을 할 때마다 배들은 랜덤하게 배치되어야 한다.

ii. 새로운 게임을 할 때마다 배의 종류는 랜덤하게 결정되어야 한다.

iii. 새로운 게임을 할 때마다 배의 움직임은 정해진 루틴 내에서 랜덤하게 결정되어야 한다. 예를 들어 x 좌표 4에 존재하는 BoatA 는 2,3,4 혹은 3,4,5 혹은 4,5,6 의 움직임을 가져갈 수 있다. (각 움직임 예제 그림 참고)

iv. 배들은 충돌할 수 없다. 다시 말해 배들은 같은 좌표에 존재할 수 없다. (단, c 가 잠수중이라면 다른 배가 위에 존재할 수 있다)

v. 배는 지도 밖으로 나갈 수 없다.

3. Class Design & Implementation (mandatory)

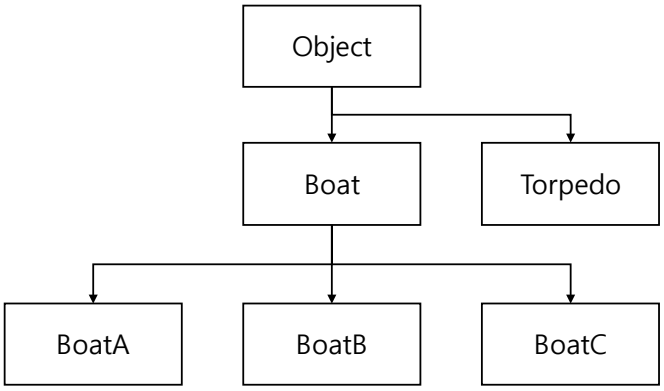
A. 기본 조건

i. 모든 데이터 멤버는 private 이어야 한다.

ii. Class 별로 cpp 파일과 h 파일을 작성한다.

iii. 프로그래밍 실습 서버에 구동이 되어야 함을 주의한다. (Makefile 작성 필수)

B. 기본 클래스 디자인은 아래와 같다.



C. 클래스의 **필수 구현 부분**은 아래와 같다 (데이터 멤버, 함수 추가 가능)

Object	
Private	Int axisX
	Int axisY
Protected	moveUp()
	moveDown()
Public	moveLeft()
	moveRight()
	Object()
	Object(int a, int b)

Torpedo	
Public	move()
	Torpedo()
	Torpedo(int a, int b)

Boat	
Private	Bool visibility
Public	Boat()
	Boat(int a, int b)

BoatA	
Public	move()
	BoatA()
	BoatA(int a, int b)

BoatB	
Public	move()
	BoatB()
	BoatB(int a, int b)

BoatC	
Public	move()
	BoatC()
	BoatC(int a, int b)

i. Object 이동 시 **주의할 점**

1. Object class 의 axisX, axisY 는 절대 좌표를 의미한다

2. BoatA, BoatB, BoatC, Torpedo 는 move() 를 call 할 때마다 루틴에 맞는 움직임을 가져가야 한다. 배가 움직일 때 move() 함수를 무조건 call 해야 한다.
3. move() 의 리턴형은 자유이나, 인수는 없다.
4. BoatA, BoatB, BoatC, Torpedo 의 move() 는 Object 의 moveUp, moveDown, moveLeft, moveRight 을 call 해야 한다.
5. Object 의 moveUp, moveDown 은 protected 로 선언된다.
6. User 는 Boat object 여야 한다.

D. 해당 클래스들은 바다 위의 object 들을 표현하기 위한 클래스이며, 전체적인 게임을 위해서는 추가적인 클래스 정의가 필요하다. (이는 학생들의 몫이다)

4. 프로그램 실행 흐름

A. 초기 메뉴를 띄운다

i. 1 입력 -> 기본 옵션으로 게임하기

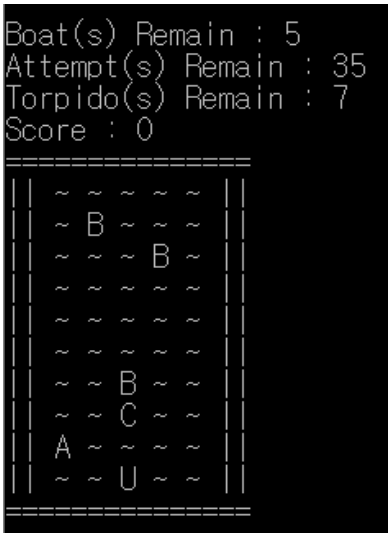
1. 기본옵션 : 바다 너비 5, 바다 길이 10, 배 5대

ii. 2 입력 -> 새로운 옵션을 유저가 정의하여 게임하기

1. 옵션입력 : 바다의 너비, 길이, 그리고 배의 개수

B. 게임 화면

i. 게임 화면은 아래와 같다

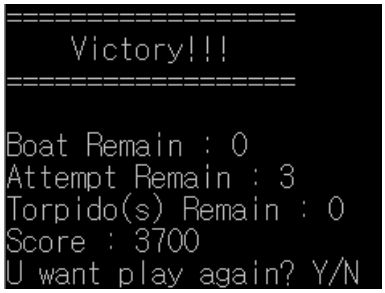


ii. 남은 배의 수, 남은 기회, 남은 어뢰 수, 그리고 점수를 출력한다.

iii. 기본적으로 알파벳 혹은 기호를 이용하여 배와 바다를 출력할 수 있으며, 이는

학생들이 자유롭게 표현하면 된다. (위의 경우 U : user, A : Boat A, C : Boat C 이다)

C. 게임 종료



i. 게임 종료 메시지를 출력하며 달성한 점수를 출력한다.

ii. 다시 플레이 할 지를 묻는 메시지가 뜨며, y 를 입력할 경우 초기 메뉴를 다시 띄워 새로 게임을 하며, n 을 입력할 경우 종료메시지와 함께 게임을 최종 종료한다.

D. 추가로 고려할 점

i. 정해진 명령어 이외의 입력이 들어올 경우 다시 입력받는다. (초기 메뉴 제외)

ii. 입력 문자는 대/소문자 구별을 하지 않는다.

5. 주의사항

A. STL 은 사용할 수 없다

B. 현재까지 배운 상속까지 사용한다. 다형성을 활용할 경우 0점 처리된다.

C. Makefile 을 작성하여야 하며, 리눅스 환경에서 make 를 입력 시 실행파일 'run' 이 생성되어야 하며, run 을 실행 시 작동하여야 한다.

D. 채점 기준은 AssnReadMe 를 참고한다.