

# 멀티미디어 고급 프로그래밍



## ASSIGNMENT 2

# 소개

- 배점
  - 30점 / 100점
- 기간
  - 2020년 6월 4일 00시 00분 까지 (6월 11일 00시 00분까지 연장 - 페널티 30%)
- 내용
  - 2-1 : Monster 구현 (VariousMonsters 포함) (10점)
  - 2-2 : Human 구현 (10점)
  - 2-3 : RankingBoard 구현 (10점)
- 제출 방식
  - 4개의 파일 (Monster.h, VariousMonsters.h, Human.h, RankingBoard.h) 압축
  - 압축파일의 형식은 **과제2\_학번.zip** 으로 한다 (형식 엄수!)

# 과제의 목표

1. 복잡한 코드를 분석 할 수 있다.
2. 필요한 내용을 검색하고 연구해서 해결할 수 있다.
3. 게임 화면 출력의 기본 원리를 이해하고 구현 할 수 있다.
4. 사용자의 입력을 받아서 처리 할 수 있다.
5. 프로그램의 실행 기록을 파일로 다룰 수 있다.

# 과제 설명

여러분은 아주 간단한 미니게임을 구현하게 됩니다. 미니게임 이름은 “몬스터 월드”라고 합니다. 정해진 크기의 화면이 있고, 화면의 각 위치에는 아이템이 존재합니다.

여러 종류의 몬스터들과 여러분은 아이템을 누가 더 많이 먹는지 경쟁을 하게 됩니다. 경쟁에 따라서 순위는 실시간으로 반영되어 화면에 표시 됩니다.

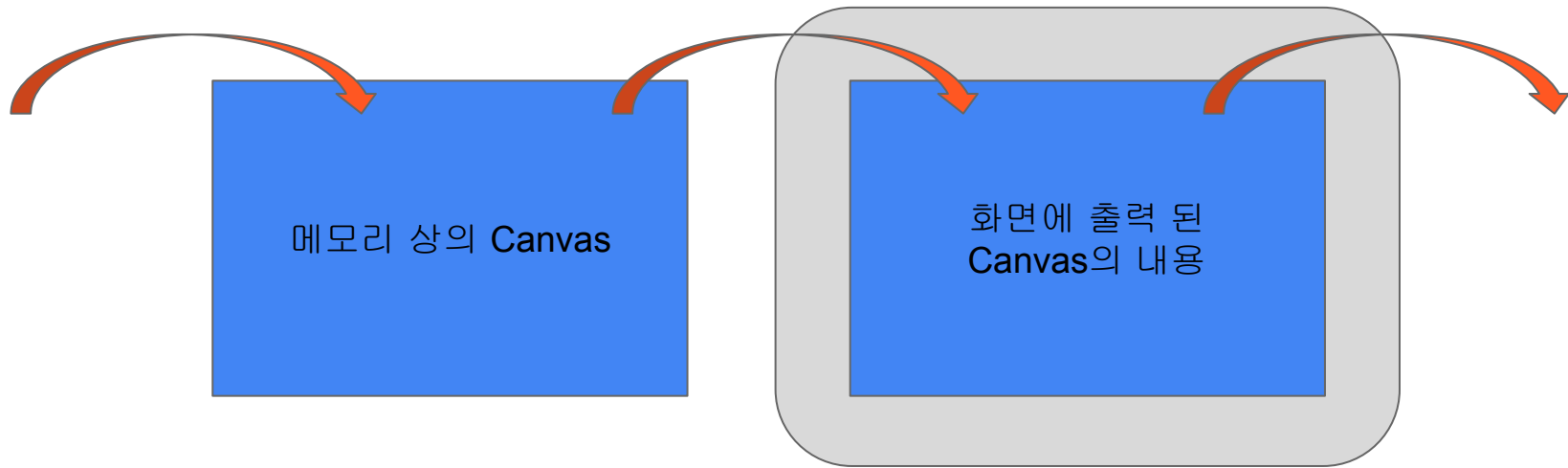
나의 기록은 매 플레이마다 판단하여 가장 좋은 5개의 기록을 저장 합니다. 기록의 저장은 파일시스템을 사용하여 이루어집니다.

# 화면을 그리는 원리

데이터 입력

화면을 통째로 갱신

이전 화면을 비움



# 화면의 흐름 (처음 시작)

Loading Ranking Board...

Password:

게임을 처음 플레이 하는 경우 랭킹보드를 위한 파일이 없기 때문에 이를 생성하기 위한 암호를 물어본다.

틀리면 게임 종료

맞으면 게임 시작

Loading Ranking Board...

Password: jaskflasd

Incorrect password. Quit game.

[ Monster World ]

.....A.....  
.....  
.....  
.....V.  
.....  
.....H.....Z.....  
.....  
.....B.....  
.....  
.....

Total Move Count = 0

Rest Items Count = 320

Rank 1	Zombie Z	0	0
Rank 2	Vampire V	0	0
Rank 3	KGhost G	0	0
Rank 4	Jiangshi1 A	0	0
Rank 5	Jiangshi2 B	0	0
Rank 6	Player H	0	0

Press [ENTER] key...

# 화면의 흐름 (게임 진행)

[ Monster World ]

```
.....
.....
..... V .....
.....
.....
..... B.....
.....
..... H.....
..... Z .. . ..... A
..... G .....
.....
```

Total Move Count = 82  
Rest Items Count = 232

Rank 1	Vampire V	36	0
Rank 2	Zombie Z	24	10
Rank 3	KGhost G	16	6
Rank 4	Jiangshi2 B	6	0
Rank 5	Jiangshi1 A	5	0
Rank 6	Player H	1	0

ENTER key를 누르게 되면  
게임이 시작된다.

게임이 진행 될 수록  
실시간으로, 소요된 이동 턴 /  
남은 아이템 수 / 랭킹 및 기록이  
표시된다.

랭킹은, 랭킹 / 몬스터 이름 및  
아이콘 / 아이템 획득 수 / 남은  
휴식 턴 수 로 표시 된다.

게임이 종료 되고 나면 내  
기록과 비교해서 랭킹을 내고,  
5위 안에 드는 경우 이름을 입력  
받는다.

[ Monster World ]

```
.....
.....
..... B .....
.....
..... V.
..... H... .G..
..... A
..... Z . ...
```

Total Move Count = 500  
Rest Items Count = 128

Rank 1	Vampire V	78	0
Rank 2	KGhost G	49	8
Rank 3	Zombie Z	38	8
Rank 4	Jiangshi1 A	18	2
Rank 5	Jiangshi2 B	8	0
Rank 6	Player H	1	0

----- Game Finish -----

[Rank 1] Insert Name:

# 화면의 흐름 (게임 종료)

[Rank 1] Insert Name: test

[Game Ranking: FINISH]

[Rank 1]	test	1	1	
[Rank 2]	NewPlayer		0	0
[Rank 3]	NewPlayer		0	0
[Rank 4]	NewPlayer		0	0
[Rank 5]	NewPlayer		0	0

Press [ENTER] key...

Vampire V	Item:78	Distance:324
KGhost G	Item:49	Distance:1117.33
Zombie Z	Item:38	Distance:329.932
Jiangshi A	Item:18	Distance:428
Jiangshi B	Item:8	Distance:403
Player H	Item:1	Distance:1

플레이어의 기록중 가장 아이템 획득 수가 높은 5개만 저장된다. 처음에는 초기값으로 5개가 다 세팅이 되고, 게임기록이 추가 되면 갱신이 된다.

새 기록이 기존 기록과 같은 경우, 해당 기록보다 낮은 랭크로 기록 한다.

기록 되는 값은, 이름 아이템 수, 단위 거리당 아이템 수 이다.

이 부분은 구현이 필요 없다. 게임이 종료 될 때 몬스터 객체가 소멸자를 호출하고 그로 인해 자동으로 출력이 되는 부분이다.

몬스터 객체는 매 움직임마다 아이템 순으로 정렬이 되어있으므로 소멸자 호출 또한 1등부터 호출이 된다.



## 화면의 흐름 (처음 시작이 아닐 경우)

Loading Ranking Board...

[Game Ranking: START]

[Rank 1]	test	1	1	
[Rank 2]	NewPlayer		0	0
[Rank 3]	NewPlayer		0	0
[Rank 4]	NewPlayer		0	0
[Rank 5]	NewPlayer		0	0

Press [ENTER] key...

암호 입력 과정 없이 읽어온 랭킹을 기반으로 나의 플레이 히스토리를 상위 5등까지 출력 해준다.

이 부분은 파일로 관리가 된다.

## 과제 2-1 (1)

게임 상에서 몬스터는 일정 시간 주기로 move 가 호출 된다. move가 호출 되면 몬스터는 이동을 하기도 하고 휴식을 하기도 한다. 휴식을 하는 경우는 휴식 카운트 (nSleep)을 1 감소 시키고 턴을 종료한다. 휴식이 아닌 경우다음의 과정을 공통으로 거친다.

1. 자신의 특성에 맞게 이동을 한다.
2. clip 함수를 통해 자신의 위치가 화면을 나가지 않게 조정한다.
3. eat 함수를 통해 자신의 위치에 아이템이 있다면 획득한다.
4. 총 누적 이동거리와 휴식 후 누적 이동거리를 갱신한다.
5. 휴식 후 누적 이동거리를 확인해서 20이 넘는 경우 10턴 휴식을 한다.

## 과제 2-1 (2)

**Monster**는 4가지 종류가 있다. **Zombie**, **Vampire**, **KGhost**, **Jiangshi** 이며 각각은 저마다 이동 특성이 있다.

**Zombie**: Monster의 기본 이동 특성을 그대로 사용한다. 제자리를 포함한 가로 세로 대각선 방향으로 1칸씩 총 9가지 이동 패턴이 있다.

**Vampire**: 가로 세로로만 1칸씩 이동한다. 제자리와 대각선 이동은 못한다.

**KGhost**: 화면 내 랜덤한 지역으로 순간이동을 한다. 제자리로도 갈 수 있다.

**Jiangshi**: 가로 또는 세로로만 이동 가능하다. 생성시에 방향을 위한 인자를 입력받는다. 이동은 1칸 또는 2칸을 점프 할 수 있다 (제자리 제외)

## 과제 2-1 (3)

**Monster.h, VariousMonsters.h** 를 완성하여 주어진 조건을 만족 하도록 구현 하시오.

- 각 클래스가 가지는 모든 멤버 변수는 전부 유의미한 값을 가지고 있어야 한다.
- 원하는 멤버함수 또는 전역 함수를 2개의 파일에 얼마든지 추가 할 수 있지만 비어있는 함수(Monster 의 clip, eat, move 와 Vampire, KGhost, Jiangshi 의 move)는 전부 유의미하게 작성이 되어야 한다.
- 이 과제를 위해 Monster.h, VariousMonsters.h 이외의 다른 파일에 내용을 추가 하거나 수정을 하면 안된다.

## 과제 2-2

인간(Human)은 Monster를 상속 받지만 몇가지 차이점이 있다.

1. 이동을 할 때에 사용자 입력을 받는다. 사용자 입력은 상/하/좌/우 이며, 각각 w/s/a/d 를 사용한다.
2. 누적 거리와 관계 없이 휴식을 하지 않는다.

Human.h 의 move를 조건에 맞도록 완성하시오. 이를 위해서 `_getch()`, `_kbhit()` 를 어떻게 사용하는지 조사 할 필요가 있을 것이다.

## 과제 2-3

**RankingBoard.h**의 함수를  
완성하여 플레이어의 랭킹을  
파일로 관리하는 프로그램을  
작성하시오.

해당 파일의 각 멤버 변수는  
유의미하게 사용 되어야 한다.

저장/로드 되는 값은, 사용자  
이름, 획득한 아이템 수, 이동  
거리당 아이템 수 이다.

```
// 주어진 파일의 데이터를 읽어온다. 파일이 없는 경우 예외 처리를 한다.
void load()
{
    ifstream is;

}

// 랭킹 리스트를 저장한다.
void store()
{
    ofstream os;

}

// 랭킹 리스트를 1위부터 순서대로 출력한다.
void print(string title = "Game Ranking")
{

}

// 게임이 끝난 후 플레이어의 기록을 입력한다.
// nItem : 아이템 획득 수, ipm : 이동거리당 아이템 획득 비율 (아이템 획득 수 / 총 이동 거리)
// 입력시 현재 기록과 비교하여 5위 이내인 경우 기록을 위한 이름을 받는다.
void add(int nItem, double ipm)
{

}
```