

C++ 프로그래밍 및 실습

# 팩맨 게임 개발

진척 보고서 #1

제출일자: 2024.11.17

제출자명: 이상현

제출자학번: 214784

## 1. 프로젝트 목표

### 1) 배경 및 필요성

최근 수업 시간에서 MUD GAME에 대해 접하다 보니, 이러한 게임 개발에 흥미를 가지게 되었음. 그래서 이번 프로젝트의 주제로 가장 널리 알려져 있고 죽기 전에 꼭 해야 할 비디오 게임 명단에도 존재하는 팩맨(Pac-Man)을 직접 구현해보고자 하였음.

### 2) 프로젝트 목표

주어진 미로 안에서 팩맨이 유령들을 피해다니며 미로에 놓인 코인들을 모두 줍게 되면 승리하도록 게임 구현하기.

### 3) 차별점

기존 팩맨 게임의 유령들의 AI는 각 색깔의 유령마다 고유의 AI가 존재했다. 빨간 유령(Blinky)는 현재 팩맨의 위치로 이동하고, 분홍 유령(Pinky)는 팩맨의 4칸 앞 지점으로 이동하고, 초록 유령(Inky)는 팩맨의 2칸 앞 지점을 중심으로 빨간 유령(Blinky)의 대칭된 좌표로 이동하고, 노란 유령(Clyde)는 팩맨으로부터 도망가도록 설계되어 있었다. 그러나 이번 프로젝트에서는 게임 난이도 설정을 도입하여 사용자가 정한 난이도에 따라 유령들의 이동 AI와 유령의 수를 변경하는걸 차별점으로 두었다.

## 2. 기능 계획

### 1) 메인 화면 구현

- 본격적인 팩맨 게임 시작 전 화면을 만들고 사용자의 입력에 따라 게임 시작, 점수판, 게임 종료를 다룰 메인 화면을 구현

세부기능

(1) 게임 시작을 뜻하는 입력을 받았다면 팩맨 게임을 실행하도록 구현

- 진행할 게임의 난이도를 입력으로 받아 그에 맞게끔 유령의 움직임과 수가 조절되도록 구현

- 팩맨 게임을 하다가 게임 오버가 된다면 다시 메인 화면으로 오도록 구현

(2) 게임이 끝날 때 마다 현재 점수를 자동으로 기록하고, 메인 화면에서 점수판을 뜻하는 입력을 받았다면 기록된 점수들을 볼 수 있는 점수판 구현.

(3) 게임 종료를 뜻하는 입력을 받았다면 프로그램을 종료하도록 구현

## 2) 팩맨 게임 맵 구현

- 팩맨과 유령이 움직이는 미로 형식의 맵 구현

세부기능

(1) 벽, 코인, 아이템을 2차원 배열로 구현(각각을 뜻하는 문자열을 설정하여 구분)

- 2차원 배열로 전체 맵을 구현하고, 팩맨과 유령이 지나갈 수 없는 벽을 지정함.

- 팩맨이 맵 통로에 있는 코인을 먹게되면 (지나가면) 점수가 증가하도록 함.

- 팩맨이 맵 통로에 있는 아이템을 먹게되면 (지나가면) PowerUp 모드가 되도록 함.

- 유령의 스폰 장소를 지정해두고, 일정 시간이 지날 때 마다 한 마리씩 추가로 움직이게 함.

## 3) 사용자의 입력에 따른 팩맨의 움직임 구현

- 사용자로부터 상, 하, 좌, 우를 입력 받아 팩맨의 이동 방향을 결정

세부기능

(1) 사용자의 input에 따라 방향을 전환하고, 설정한 프레임에 따라 움직이기

- input으로 상, 하, 좌, 우를 사용자에게 받아오고 그에 따라 앞으로 팩맨이 움직일 방향을 결정함.

- 프레임 속도를 설정하고, 그 시간 간격에 따라 콘솔창을 갱신하면서 팩맨과 유령이 움직이는 것을 구현함.

- 평상시에는 팩맨과 유령의 이동 속도는 동일하나, 팩맨이 PowerUp 모드가 된다면 유령보다 조금 더 빠르게 이동하도록 프레임 속도를 조절함.

#### 4) 유령들의 움직임 알고리즘 구현

- BFS를 활용하여 유령과 팩맨 사이의 최단 거리를 계산하여 이동하도록 구현

세부기능

##### (1) 유령들의 움직임 알고리즘 구분

- (A모드) BFS를 통해 현재 팩맨의 위치까지 최적의 경로를 찾아내서 이동하는 모드

- (B모드) 팩맨의 2칸 앞을 중심으로 하고, A모드 유령 위치와 대칭되는 좌표로 이동하는 모드

- (C모드) 일정 범위까지는 팩맨을 향해서 이동하다가, 팩맨과 너무 가까워지면 팩맨과 멀어지도록 이동하는 모드

- (D모드) 팩맨의 주변 4칸으로 이동하는 모드

- (E모드) 팩맨이 PowerUp 모드가 되었을 때, 유령이 팩맨으로부터 도망치는 모드

##### (2) 사용자가 설정한 난이도에 따라 유령들의 이동 알고리즘을 바꿔줄 예정

- 쉬움, 보통, 어려움 난이도로 구별함.

- 쉬움 난이도에서는 A, C모드를 사용하는 2마리의 유령을 소환

- 보통 난이도에서는 A, B, C, D모드를 각각 사용하는 4마리의 유령을 소환

- 어려움 난이도에서는 A모드 2마리, B모드 2마리, C모드 1마리, D모드 1마리 총 6마리의 유령을 소환

## 5) 팩맨이 유령을 잡아먹을 수 있는 PowerUp 모드 구현

- 팩맨이 맵에 놓여있는 아이템을 먹게 되면 일정 시간 동안 유령을 잡아먹을 수 있는 PowerUp 모드 구현

세부기능

- (1) 팩맨이 PowerUp 모드일 때 유령과 같은 좌표가 되면 유령을 없애고 초기 장소로 돌려보냄.
- (2) 팩맨이 PowerUp 모드가 된다면 유령들의 움직임 패턴을 E모드로 설정하여 팩맨으로부터 멀어지도록 구현
- (3) 팩맨이 PowerUp 모드일 때는 유령보다 이동 속도가 더 빨라지도록 구현

## 6) 잡아먹혔던 유령 부활 기능 구현

- PowerUp 모드로 인해 죽은 유령은 일정 시간 후 다시 부활하여 맵을 돌아다니도록 구현

세부기능

- (1) PowerUp 모드의 팩맨에 의해 죽게 된 유령들을 리스폰 장소에서 5초 후에 리스폰 하도록 구현

## 7) 게임 종료 후 점수 기록 기능 구현

세부기능

- (1) 게임이 종료되면 점수판에 점수를 자동으로 저장함.
  - 각 게임마다 팩맨(유저)에게 주어진 목숨은 한 게임.
  - 팩맨이 유령과 같은 좌표에 있게 되면 목숨이 닳아서 게임이 종료됨.
  - 게임이 종료되면 게임을 진행하면서 먹은 코인들로 인해 얻은 점수를 1차원 배열로 저장함.

- 이를 메인 화면의 점수판 항목에서 볼 수 있도록 구현.

### 3. 진척 사항

#### 1) 기능 구현

##### (1) 메인 화면 구현

- 본격적인 팩맨 게임 시작 전 화면을 만들고 사용자의 입력에 따라 게임 시작, 점수판, 게임 종료를 다룰 메인 화면을 구현

- 입력으로 1을 받을 시 게임 맵을 출력함
- 입력으로 2를 받을 시 점수판을 출력함
- 입력으로 3을 받을 시 게임을 종료함
- 1~3을 제외한 입력을 받을 시 재입력을 요청함.

##### 1. 코드 블록 스크린샷

```
int main() {
    mainMenu();
    // score_board.push_back(score); 게임 실행 도중 종료되면 다음 코드를 사용하여 점수 저장
}
```

```
// 메인메뉴를 구현하는 함수
void mainMenu() {
    bool game_flag = true;
    while(game_flag){
        cout << "===== MAIN MENU =====" << endl;
        cout << "1. 게임 시작" << endl;
        cout << "2. 점수판 보기" << endl;
        cout << "3. 게임 종료" << endl;
        cout << "선택: ";
        int input;
        cin >> input;
        switch (input) {
            case 1:
                clearScreen();
                DisplayMap(map);
                break;
            case 2:
                showScoreboard();
                break;
            case 3:
                game_flag = false;
                break;
            default:
                cout << "잘못된 입력입니다. (1~3 입력)" << endl;
        }
    }
}
```

```
// 맵을 출력하는 함수
void DisplayMap(char map[][WIDTH+1]) {
    for (int i = 0; i < HEIGHT; i++) {
        for (int j = 0; j < WIDTH+1; j++) {
            if (i == pacY && j == pacX)
                cout << PACMAN;
            else if (i == ghostY && j == ghostX)
                cout << GHOST;
            else
                cout << map[i][j];
        }
        cout << endl;
    }
}
```

```
// 점수판 출력 함수 (MAIN 메뉴에서 2 입력 시 실행 가능, 지금까지 실행했던 게임의 점수 저장)
void showScoreboard() {
    cout << "===== 점수판 =====" << endl;
    if (score_board.size() <= 0) {
        cout << "No Game Data. Play Game Please" << endl;
    }
    else {
        for(int i = 0; i < score_board.size(); i++) {
            cout << i+1 << ". 점수: " << score_board[i] << endl;
        }
    }
}

// 맵을 출력하기 전에 콘솔창을 깔끔하게 하기 위한 함수
void clearScreen() {
    for (int i = 0; i < 10; ++i) {
        cout << endl;
    }
}
```

## 2. 입력

- input = 유저의 입력값 저장

## 3. 반환값

- 입력이 1이면 clearScreen(), DisplayMap() 함수 실행
- 입력이 2면 showScoreboard() 함수 실행
- 입력이 3이면 game\_flag를 false로 바꿈.
- 그 외의 값이라면 다시 입력 받음.

## 4. 결과

- 입력이 1일때 clearScreen() 함수로 콘솔창을 깨끗이 비우고,  
DisplayMap() 함수를 실행하여 현재의 게임 맵을 출력함

- 입력이 2일땐 showScoreboard() 함수를 실행하여 지금까지 저장된 게임 점수를 출력함
- 입력이 3일땐 game\_flag를 false로 바꿔 while문이 반복되지 않게 하여 프로그램을 종료함

## 5. 설명

- main() 함수에서 mainMenu() 함수를 실행함.
- mainMenu() 함수에서 사용자의 입력을 받아 1이면 현재의 게임 맵을 출력, 2이면 게임이 종료될 때 현재 점수를 저장해 둔 점수판을 출력, 3이면 프로그램을 종료, 그 외의 값이면 다시 입력 받도록 하였음.
- DisplayMap() 함수는 2중 for문을 통해 2차원 배열 map을 하나씩 순회하면서 해당 값과 팩맨의 좌표나 유령의 좌표가 동일하다면 팩맨과 유령에 해당하는 아이콘을 출력하고, 그렇지 않다면 map배열에 저장된 값을 출력함 (# = 벽, . = 코인(점수), \* = 아이템, P = 팩맨, G = 유령)
- showScoreboard() 함수는 게임이 종료될 때 마다 현재의 score의 값을 저장해 둔 score\_board 벡터의 값을 불러오며 몇 번째 실행에서 몇 점을 얻었는지 출력해줌. 만약 게임을 하지 않아서 점수가 없다면 게임을 먼저 수행하라는 안내 문구 출력.
- clearScreen() 함수는 게임 맵을 출력하기 전에 콘솔창을 깔끔하게 해주기 위한 역할로 cout << endl을 여러 번 반복함.

## (2) 팩맨 게임 맵 구현

- 팩맨과 유령이 움직이는 미로 형식의 맵 구현
- 벽, 코인, 아이템을 2차원 배열로 구현(각각을 뜻하는 문자열을 설정하여 구분)

### 1. 코드 블록 스크린샷



```

// 맵 구현 (# = 벽, . = 코인(점수), * = 아이템)
char map[HEIGHT][WIDTH + 1] = {
    "#####" ,
    "#.....#" ,
    "#.###.###.###.###" ,
    "#*###.###.###.###*" ,
    "#.###.###.###.###" ,
    "#.....#" ,
    "#.###.###.###.###" ,
    "#.###.###.###.###" ,
    "#.....#" ,
    "##### # #####" ,
    "  #.### # #####" ,
    "  #.# # #.#" ,
    "  #.# # #.#" ,
    "##### # # #####" ,
    "#      -      #" ,
    "##### # # # #####" ,
    "  #.# # #.#" ,
    "  #.# #.#" ,
    "  #.#.#####.#.#" ,
    "#####.#.#...#.#.#####" ,
    "#.....#.#.#####" ,
    "#.###.###.###.###" ,
    "#.###.###.###.###" ,
    "#*..#.#.....#.#.*" ,
    "#####.###.#####.#.#.###" ,
    "#####.#.#...#.#.###" ,
    "#.....#####.#.#####" ,
    "#.#####.#.#####.#" ,
    "#.....#" ,
    "#####.#.#####.#" ,
    "#.....#" ,
    "#####" ,
};

```

2. 입력

3. 반환값

4. 결과

5. 설명

- map이라는 2차원 배열을 다음과 같이 선언함.

- # = 벽, . = 코인(점수), \* = 아이템, P = 팩맨, G = 유령을 뜻함.

## 2) 테스트 결과

### (1) 메인 화면 구현

```
===== MAIN MENU =====
1. 게임 시작
2. 점수판 보기
3. 게임 종료
선택: □
```

- 1 입력시 게임 맵 출력

```
#####
#P...G.....#.....#
#.#.#.#.#.#.#.#.#.#.#
#*#####.#.#.#.#*#
#.#.#.#.#.#.#.#.#.#.#
#.....#
#.#.#.#.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#.#.#.#
#.....#.....#.....#
#####.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#
#####.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#
#####.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#
#####.#.#.....#.....#
#.....#.#.#.#.#.#.#
#.#.#.#.#.#.#.#.#.#.#
#.#.#.#.#.#.#.#.#.#.#
#*..##.....##.*#
###.##.##.#####.##.##.##
###.##.##.....#.....#
#.....#####.#.#.#.#.....#
#.#.#.#.#.#.#.#.#.#.#.#
#.....#
```

- 2 입력시 점수판 출력

```
===== MAIN MENU =====
1. 게임 시작
2. 점수판 보기
3. 게임 종료
선택: 2
===== 점수판 =====
No Game Data. Play Game Please
```

### - 3 입력시 게임 종료

```
===== MAIN MENU =====
1. 게임 시작
2. 점수판 보기
3. 게임 종료
선택: 3
PS C:\CPP2409-P> █
```

### - 그 외의 값 입력 시 에러 문구 출력

```
===== MAIN MENU =====
1. 게임 시작
2. 점수판 보기
3. 게임 종료
선택: 6
잘못된 입력입니다. (1~3 입력)
===== MAIN MENU =====
1. 게임 시작
2. 점수판 보기
3. 게임 종료
선택: █
```

## 4. 계획 대비 변경 사항

### 1) 팩맨 게임 맵 구현

- 원래 계획상 11월 17일까지 팩맨 게임 맵 구현 및 팩맨의 위치에 따른 아이템과의 상호 작용 기능을 완성하려 하였음.

그러나 2차원 배열로 맵을 구현하여 벽, 코인, 아이템, 팩맨, 유령을 표기하는 것은 계획대로 되었으나 팩맨과 유령의 움직임 기능 구현을 다음 진척 보고서 발표인 12월 1일까지로 계획해서 이번 진척 보고서에서는 아이템과의 상호 작용 기능을 구현할 수 없었음.

그에 따라 12월 1일까지 제출인 진척 보고서에서 이번에 구현하지 못했던 기능2) 팩맨 게임 맵 구현의 세부 기능 구현을 마저 진행하려 함.

구현해야 할 세부 기능은 다음과 같음.

-> 팩맨의 유효한 움직임 판단, 팩맨이 코인을 먹을 시 점수 증가, 아이템을 먹을 시 PowerUp 모드 변신, 일정 시간마다 맵에 유령 추가하기, 팩맨과 유령이 동일 좌표에 있을 시 팩맨 사망 처리.

## 5. 프로젝트 일정

업무		11/3	11/17	12/1	12/15	12/22
제안서 작성		완료				
기능1	세부기능1		완료			
	세부기능2		완료			
	세부기능3		완료			
기능2	세부기능1		30%완료 (진행중)----->			
기능3	세부기능1			----->		
기능4	세부기능1			----->		
	세부기능2			----->		
기능5	세부기능1				----->	
	세부기능2				----->	
	세부기능3				----->	
기능6	세부기능1					----->
기능7	세부기능1					----->