**소스 파일**

**Project 2\_Pacman Game 12184407 이상운**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <time.h>

#include <Windows.h>

#define KEY\_ESC 0x1B

#define KEY\_8 '8'

#define KEY\_2 '2'

#define KEY\_4 '4'

#define KEY\_6 '6'

#define KEY\_W 'w'

#define KEY\_X 'x'

#define KEY\_A 'a'

#define KEY\_D 'd'

#define UP 1

#define DOWN 2

#define LEFT 3

#define RIGHT 4

#define MAZE\_H 25

#define MAZE\_W 26

#define WALL 32

#define FOOD 1

#define PILL 2

#define NUM\_PILL 4

typedef struct \_POS {

int row;

int col;

} POS;

POS pacman;

int direction = RIGHT;

int speed = 100;

void init();

void turn(int key);

void move();

int check();

POS rand\_pos();

void draw\_maze();

void draw\_pacman();

void clear\_pacman();

void show\_instruction();

void goto\_pos(POS pos);

void gotoxy(int x, int y);

void showCursor(BOOL bVisible);

char\* tile[] = { " ", "┌", "┐", "└", "┘", "│", "─", "├", "┤", "┬", "┴", "┼" };

int maze[MAZE\_H][MAZE\_W] =

{

{ 33,38,38,38,38,38,38,41,38,38,38,38,38,38,38,38,38,38,41,38,38,38,38,38,38,34 },

{ 37, 1, 1, 1, 1, 1, 1,37, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,37, 1, 1, 1, 1, 1, 1,37 },

{ 37, 1,33,38,38,34, 1,37, 1,33,38,38,38,38,38,38,34, 1,37, 1,33,38,38,34, 1,37 },

{ 37, 1,35,38,38,36, 1,37, 1,35,38,38,38,38,38,38,36, 1,37, 1,35,38,38,36, 1,37 },

{ 37, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,37 },

{ 35,38,34, 1,33,34, 1,38,38,38,38, 1,33,34, 1,38,38,38,38, 1,33,34, 1,33,38,36 },

{ 33,38,36, 1,37,37, 1, 1, 1, 1, 1, 1,37,37, 1, 1, 1, 1, 1, 1,37,37, 1,35,38,34 },

{ 37, 1, 1, 1,37,35,38,34, 1,33,38,38,36,35,38,38,34, 1,33,38,36,37, 1, 1, 1,37 },

{ 37, 1, 1, 1,35,38,38,36, 1,35,38,38,38,38,38,38,36, 1,35,38,38,36, 1, 1, 1,37 },

{ 35,38,34, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,33,38,36 },

{ 32,32,37, 1,33,38,38,34, 1,33,38,38,38,38,38,38,34, 1,33,38,38,34, 1,37,32,32 },

{ 32,32,37, 1,37,33,38,36, 1,37, 7, 3,15, 3, 3, 3,37, 1,35,38,34,37, 1,37,32,32 },

{ 32,32,37, 1,37,37, 1, 1, 1,37, 3,11, 3,19, 3, 3,37, 1, 1, 1,37,37, 1,37,32,32 },

{ 33,38,36, 1,35,36, 1,37, 1,35,38,38,38,38,38,38,36, 1,37, 1,35,36, 1,35,38,34 },

{ 37, 1, 1, 1, 1, 1, 1,37, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,37, 1, 1, 1, 1, 1, 1,37 },

{ 35,38,34, 1,38,38,38,42,38,38,38, 1,33,34, 1,38,38,38,42,38,38,38, 1,33,38,36 },

{ 32,32,37, 1, 1, 1, 1, 1, 1, 1, 1, 1,37,37, 1, 1, 1, 1, 1, 1, 1, 1, 1,37,32,32 },

{ 32,32,37, 1,33,38,38,34, 1,33,38,38,36,35,38,38,34, 1,33,38,38,34, 1,37,32,32 },

{ 33,38,36, 1,35,38,38,36, 1,35,38,38,38,38,38,38,36, 1,35,38,38,36, 1,35,38,34 },

{ 37, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,37 },

{ 37, 1,33,38,34, 1,33,41,38,38,38, 1,33,34, 1,38,38,38,41,34, 1,33,38,34, 1,37 },

{ 37, 1,37,32,37, 1,37,37, 1, 1, 1, 1,37,37, 1, 1, 1, 1,37,37, 1,37,32,37, 1,37 },

{ 37, 1,35,38,36, 1,35,36, 1,38,38,38,42,42,38,38,38, 1,35,36, 1,35,38,36, 1,37 },

{ 37, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,37 },

{ 35,38,38,38,38,38,38,38,38,38,38,38,38,38,38,38,38,38,38,38,38,38,38,38,38,36 },

};

int main(void)

{

int ch;

clock\_t tic;

init();

show\_instruction();

turn(\_getch());

tic = clock();

while (ch = check())

{

if (\_kbhit())

turn(\_getch());

clear\_pacman();

move();

draw\_pacman();

gotoxy(54, 8);

printf("%7.1f sec", (clock() - tic) / 1000.0);

Sleep(speed);

}

gotoxy(54, 10);

puts("Done!");

gotoxy(0, MAZE\_H);

return 0;

}

void init()

{

POS pos;

int i;

int temp = 0;

srand((int)time(NULL));

while (1)

{

pos.row = rand() % MAZE\_H; // 알약의 위치 랜덤으로 설정

pos.col = rand() % MAZE\_W;

if (maze[pos.row][pos.col] == FOOD) // 알약이 음식 위치에 있을 시, 조건문 실행

{

maze[pos.row][pos.col] = PILL; // FOOD를 PILL로 바꾼다.

temp++; // 알약의 개수를 저장하는 변수

}

if (temp == NUM\_PILL) // temp가 NUM\_PILL의 개수 까지 됬을 경우 반복문 탈출

break;

}

pos = rand\_pos(); // rand\_pos() => 팩맨의 위치 지정 함수

maze[pos.row][pos.col] = 0;

// 팩맨의 위치는 FOOD위치 이므로 FOOD를 먹은 위치(0)으로 설정

pacman = pos; // 전역변수 pacman에 지역변수 pos를 대입

system("cls");

showCursor(FALSE);

draw\_maze();

draw\_pacman();

}

void turn(int key)

{

switch (key)

{

case KEY\_4: // left

case KEY\_A:

direction = LEFT; // 사용자가 KEY\_4 or A를 입력 시 direction 변수에 LEFT대입

break;

case KEY\_6: // right

case KEY\_D:

direction = RIGHT; // 사용자가 KEY\_6 or D를 입력 시 direction 변수에 RIGHT대입

break;

case KEY\_2: // down

case KEY\_X:

direction = DOWN; // 사용자가 KEY\_2 or X를 입력 시 direction 변수에 DOWN대입

break;

case KEY\_8: // up

case KEY\_W:

direction = UP; // 사용자가 KEY\_8 or W를 입력 시 direction 변수에 UP대입

break;

case KEY\_ESC: // quit

gotoxy(54, 10);

puts("Exit!\n");

gotoxy(0, MAZE\_H);

exit(0);

break;

}

}

void move()

{

/\*

팩맨이 이동하려는 방향의 장소에 ". ", " ", "+ "의 문자열이 있어야 움직일 수 있다.

따라서 진행방향에 벽이 있을 시 팩맨은 움직일 수가 없다.

팩맨이 이동한 자리의 값은 0을 집어넣어 " "를 출력하게 한다.

팩맨이 이동 할 시, 팩맨의 배열의 위치 값 변경 및 FOOD제거.

\*/

if (direction == UP)

{

if (maze[pacman.row - 1][pacman.col] < 3)

{

maze[pacman.row - 1][pacman.col] = 0;

maze[pacman.row][pacman.col] = 0;

pacman.row--;

draw\_pacman();

clear\_pacman();

}

}

else if (direction == DOWN)

{

if (maze[pacman.row + 1][pacman.col] < 3)

{

maze[pacman.row + 1][pacman.col] = 0;

maze[pacman.row][pacman.col] = 0;

pacman.row++;

draw\_pacman();

clear\_pacman();

}

}

else if (direction == RIGHT)

{

if (maze[pacman.row][pacman.col + 1] < 3)

{

maze[pacman.row][pacman.col + 1] = 0;

maze[pacman.row][pacman.col] = 0;

pacman.col++;

draw\_pacman();

clear\_pacman();

}

}

else if (direction == LEFT)

{

if (maze[pacman.row][pacman.col - 1] < 3)

{

maze[pacman.row][pacman.col - 1] = 0;

maze[pacman.row][pacman.col] = 0;

pacman.col--;

draw\_pacman();

clear\_pacman();

}

}

}

int check()

{

int a, b;

int temp = 0;

for (a = 0; a < MAZE\_H; a++)

// 반복문을 통해 maze[0][0]부터 maze[MAZE\_H][MAZE\_W]까지 검사

{

for (b = 0; b < MAZE\_W; b++)

{

if (maze[a][b] < 3)

// ". ", " ", "+ " (1 , 0, 2)를 모두 temp에 더하면서 저장

temp += maze[a][b];

}

}

return temp;

// 최종적으로 PILL과 FOOD가 없을 시 " " = 0만 남으므로 0이 return 되어 main함수의 반복문 종

// 반면 PILL과 FOOD가 한 개 라도 있을 시, 0이 아닌 양수가 return 되어 main함수의 반복문 실행

}

POS rand\_pos()

{

POS pos;

int a, b;

while (1)

{

a = rand() % MAZE\_H; // 랜덤한 위치를 지정

b = rand() % MAZE\_W;

if (maze[a][b] == FOOD) // 랜덤으로 생성된 위치가 FOOD일 경우 실행

{

pos.row = a; // 해당 위치 pos에 저장

pos.col = b;

break; // 반복문 탈출

}

}

return pos; // 반복문에서 설정된 위치를 return

}

void clear\_pacman()

{

goto\_pos(pacman);

printf(" ");

}

void draw\_pacman()

{

goto\_pos(pacman);

if (direction == RIGHT)

// 전역변수 direction에 저장되있는 값이 RIGHT일 때 팩맨 위치에 "<"대입

{

maze[pacman.row][pacman.col] = "< ";

}

else if (direction == LEFT)

// 전역변수 direction에 저장되있는 값이 LEFT일 때 팩맨 위치에 ">"대입

{

maze[pacman.row][pacman.col] = "> ";

}

else if (direction == UP)

// 전역변수 direction에 저장되있는 값이 UP일 때 팩맨 위치에 "V"대입

{

maze[pacman.row][pacman.col] = "V ";

}

else if (direction == DOWN)

// 전역변수 direction에 저장되있는 값이 DOWN일 때 팩맨 위치에 "^"대입

{

maze[pacman.row][pacman.col] = "^ ";

}

printf("%s", maze[pacman.row][pacman.col]); // 팩맨의 모양 출력

}

void draw\_maze()

{

int i, j;

int MAZE[MAZE\_H][MAZE\_W];

for (i = 0; i < MAZE\_H; i++)

// maze의 숫자 값을 유지 시키기 위해 MAZE의 local변수 선언 후 복사

{

for (j = 0; j < MAZE\_W; j++)

{

MAZE[i][j] = maze[i][j];

}

}

for (i = 0; i < MAZE\_H; i++)

{

for (j = 0; j < MAZE\_W; j++)

{

if (MAZE[i][j] > WALL) // 33이상의 값이 벽이므로 조건을 건다.

{

if (MAZE[i][j] == 33)

// 각 숫자의 해당 벽 그림을 위치에 대입

MAZE[i][j] = tile[1];

else if (MAZE[i][j] == 34)

MAZE[i][j] = tile[2];

else if (MAZE[i][j] == 35)

MAZE[i][j] = tile[3];

else if (MAZE[i][j] == 36)

MAZE[i][j] = tile[4];

else if (MAZE[i][j] == 37)

MAZE[i][j] = tile[5];

else if (MAZE[i][j] == 38)

MAZE[i][j] = tile[6];

else if (MAZE[i][j] == 39)

MAZE[i][j] = tile[7];

else if (MAZE[i][j] == 40)

MAZE[i][j] = tile[8];

else if (MAZE[i][j] == 41)

MAZE[i][j] = tile[9];

else if (MAZE[i][j] == 42)

MAZE[i][j] = tile[10];

else if (MAZE[i][j] == 43)

MAZE[i][j] = tile[11];

}

else

{

if (MAZE[i][j] == FOOD) // FOOD(1)의 값의 위치에 ". "를 표시

MAZE[i][j] = ". ";

else if (MAZE[i][j] == PILL) // PILL(2)의 위치에 "+ "를 표시

MAZE[i][j] = "+ ";

else

MAZE[i][j] = tile[0];

// 상위의 값에 모두 해당 되지 않을 경우 " "를 표시

}

printf("%s", MAZE[i][j]); // 그림 출력

}

printf("\n");

}

}

void show\_instruction()

{

gotoxy(54, 1); puts("Left(4, A), Right(6, D)");

gotoxy(54, 2); puts("Down(2, X), Up(8, W)");

gotoxy(54, 3); puts("Quit(ESC)");

gotoxy(54, 5); puts("Press any key to start...");

}

void goto\_pos(POS pos)

{

gotoxy(pos.col \*2, pos.row);

// 먹이, 벽의 그림이 ". " 띄어쓰기가 포함되어 있으므로 \*2를 해준다.

}

void gotoxy(int x, int y)

{

COORD Pos = { x, y };

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), Pos);

}

void showCursor(BOOL bVisible)

{

CONSOLE\_CURSOR\_INFO CurInfo;

CurInfo.dwSize = 100;

CurInfo.bVisible = bVisible;

SetConsoleCursorInfo(GetStdHandle(STD\_OUTPUT\_HANDLE), &CurInfo);

}