Shark Game

Team. 06

김경찬 | 이대건 | 안예주

Contents

- Shark Game?
- Structure
- Function
- Conclusion
- Q&A

Shark Game?

오마주
페르난도 보테로의 모나리자
레오나르도 다 빈치의 모나리자
7/9

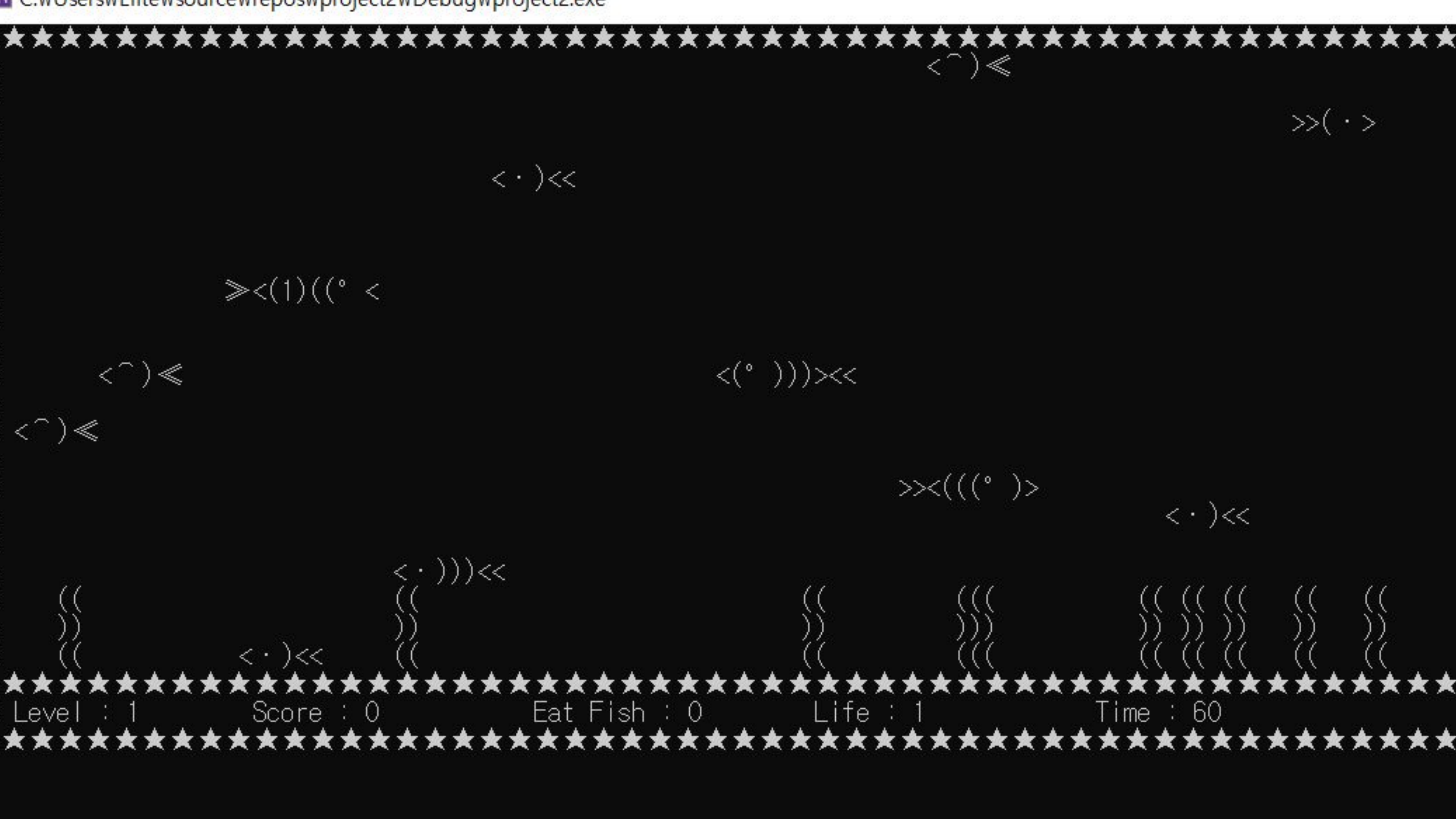
오마주(Hommage)란 '존경'을 의미하는 프랑스어로 주로 '선배 아티스트의 재능이나 공덕을 칭찬하며, 본떠 자신의 작품에 표현하는 행위'를 말합니다.

사진 | 페르난도 보테로, 레오나르도 다 빈치

플래시게임 + 오마주

Shark Game?

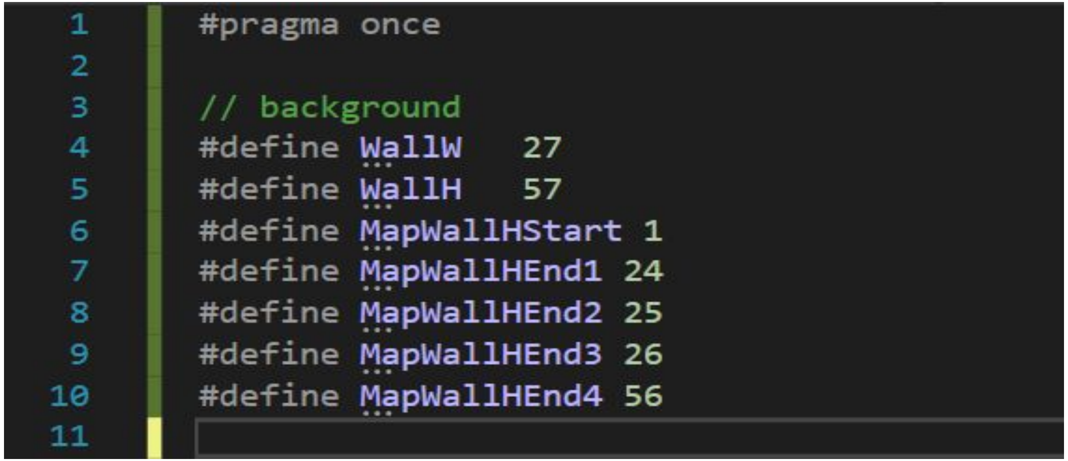
```
>°))(1)>≪ --- 물고기      (%d) LV UP!!+life

<·)<<   --->   <⌒)≪  --->   <·)))<<   --->

<(°)))><   --->   <°))))><<   --->   ≥(°°)≤

((        ))      -> 나는 미역
   ))    ((
 ((        ))
```



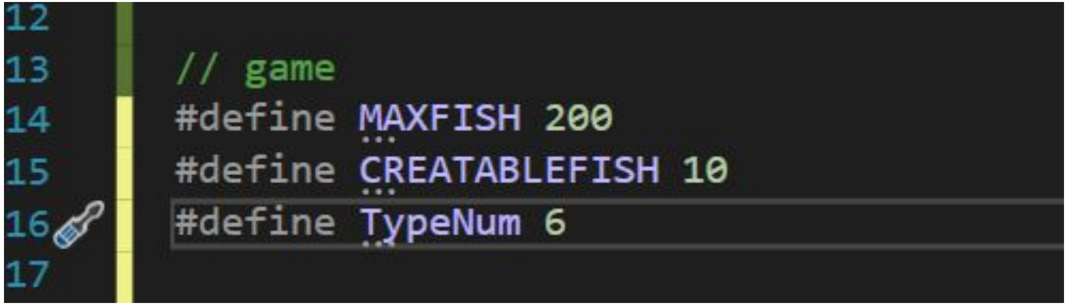
C:₩Users₩Eiffe₩source₩repos₩project2₩Debug₩project2.exe

```
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
                                                                 <^)≪

                                                                                           >>( · >

                                   < · )<<



                ≫<(1)((° <


       <^)≪                                    <(° )))><<

 <^)≪

                                                                >><(((° )>
                                                                                  < · )<<

                            < · )))<<
    ((                      ((                           ((         (((          (( (( ((   ((   ((
    ))                      ))                           ))         )))          )) )) ))   ))   ))
    ((           < · )<<    ((                           ((         (((          (( (( ((   ((   ((
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
 Level : 1        Score : 0           Eat Fish : 0        Life : 1            Time : 60
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
```

Define



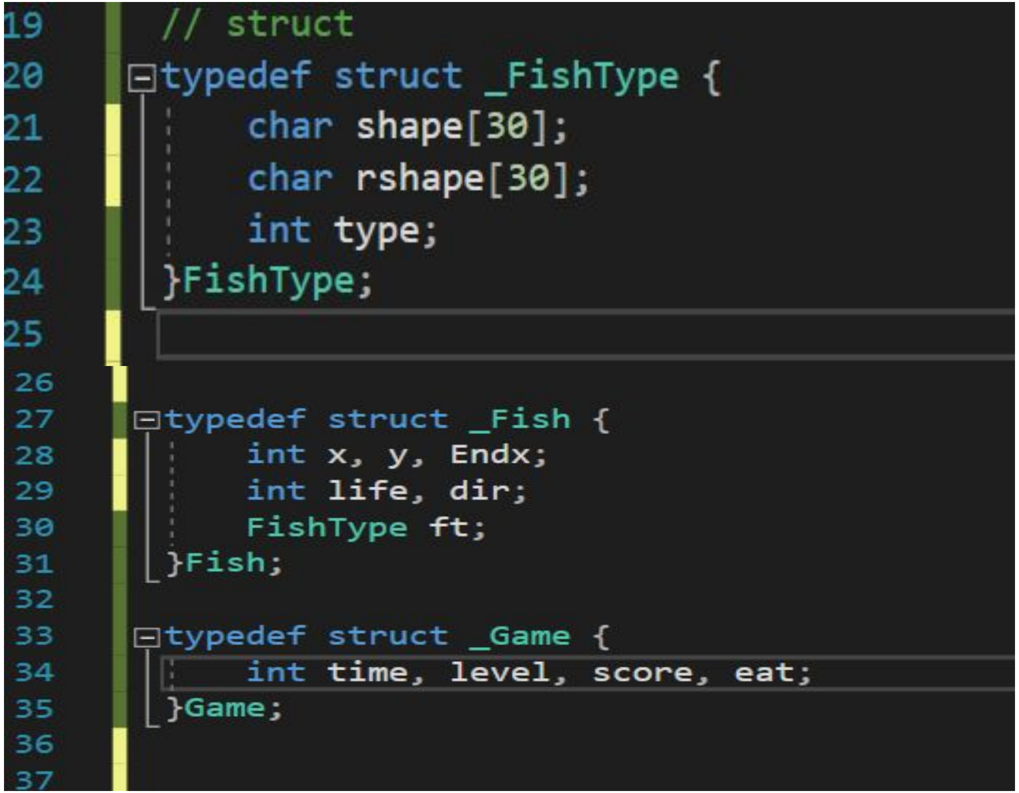
```
#pragma once

// background
#define WallW   27
#define WallH   57
#define MapWallHStart 1
#define MapWallHEnd1 24
#define MapWallHEnd2 25
#define MapWallHEnd3 26
#define MapWallHEnd4 56

```



```

// game
#define MAXFISH 200
#define CREATABLEFISH 10
#define TypeNum 6

```

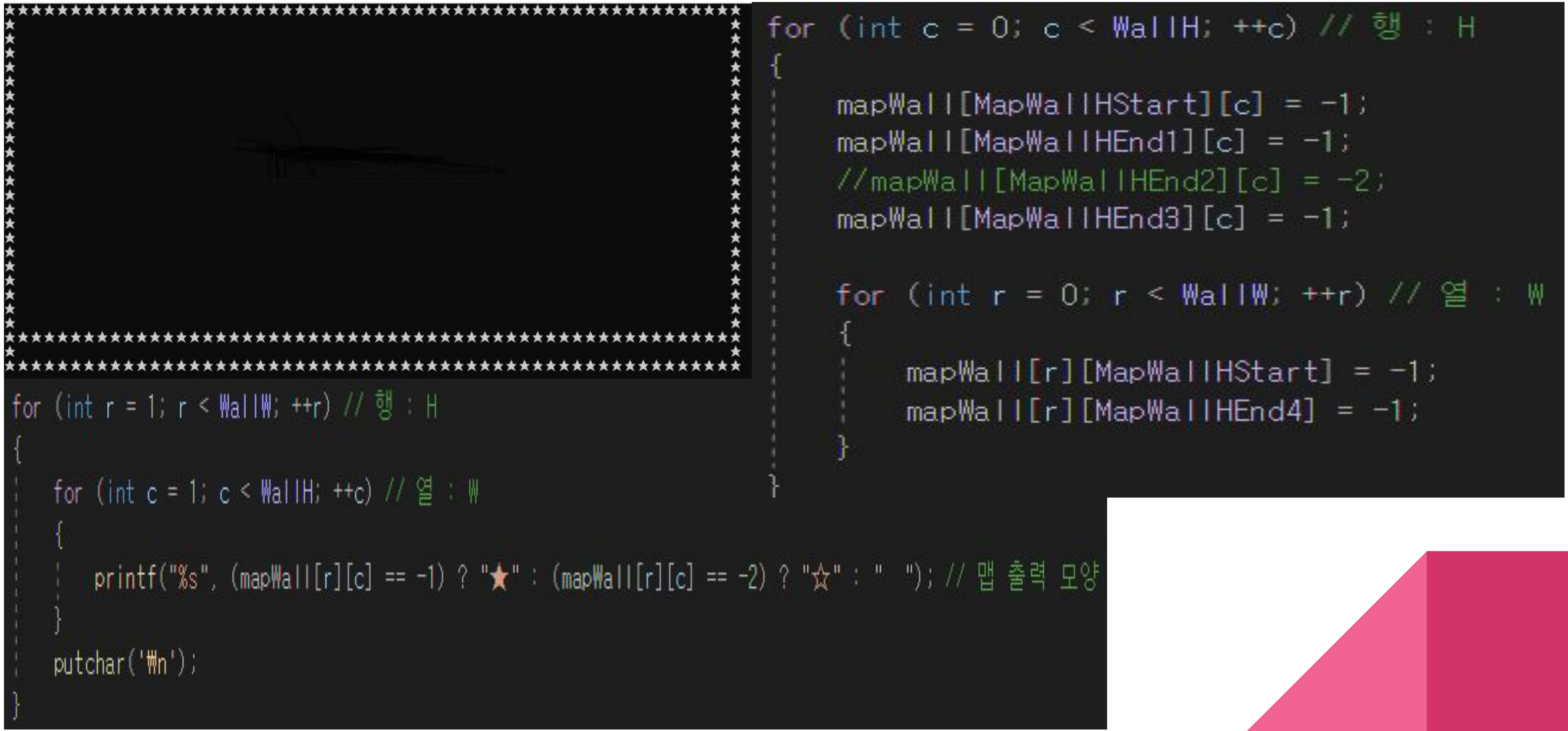
- #define 선언

Define



```
// struct
typedef struct _FishType {
    char shape[30];
    char rshape[30];
    int type;
}FishType;


typedef struct _Fish {
    int x, y, Endx;
    int life, dir;
    FishType ft;
}Fish;

typedef struct _Game {
    int time, level, score, eat;
}Game;
```

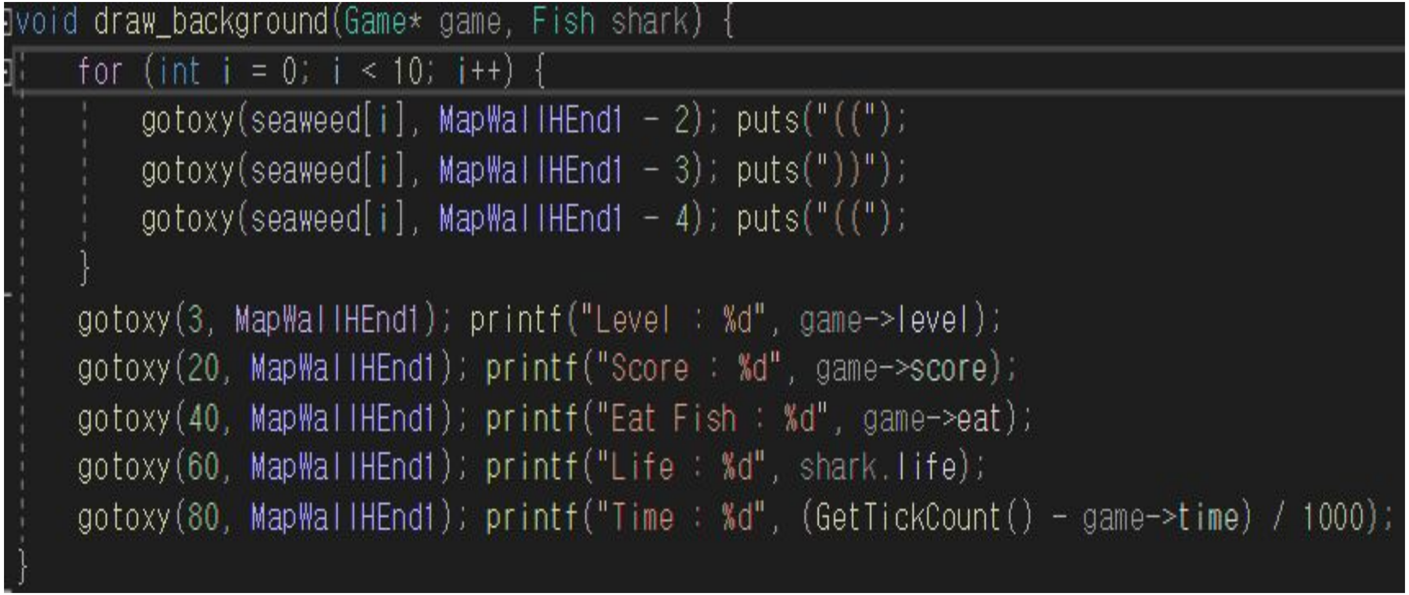
BackGround



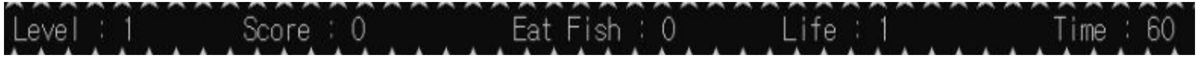
```
for (int c = 0; c < WallH; ++c) // 행 : H
{
    mapWall[MapWallHStart][c] = -1;
    mapWall[MapWallHEnd1][c] = -1;
    //mapWall[MapWallHEnd2][c] = -2;
    mapWall[MapWallHEnd3][c] = -1;

    for (int r = 0; r < WallW; ++r) // 열 : W
    {
        mapWall[r][MapWallHStart] = -1;
        mapWall[r][MapWallHEnd4] = -1;
    }
}
```

```
for (int r = 1; r < WallW; ++r) // 행 : H
{
    for (int c = 1; c < WallH; ++c) // 열 : W
    {
        printf("%s", (mapWall[r][c] == -1) ? "★" : (mapWall[r][c] == -2) ? "☆" : "  "); // 맵 출력 모양
    }
    putchar('₩n');
}
```

BackGround

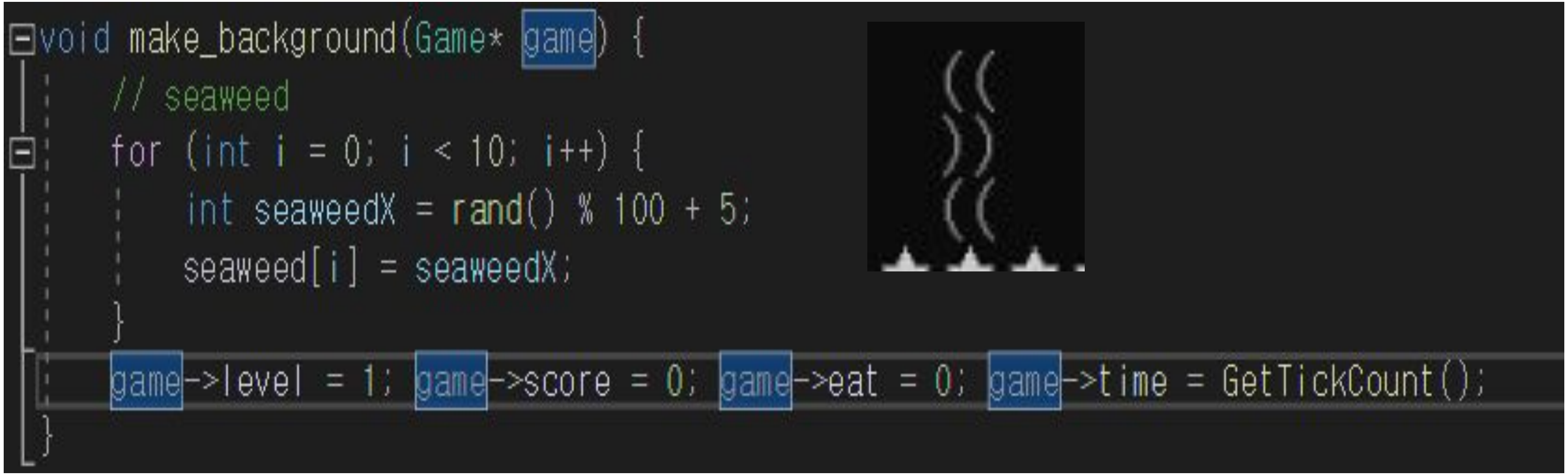


```
void draw_background(Game* game, Fish shark) {
    for (int i = 0; i < 10; i++) {
        gotoxy(seaweed[i], MapWallHEnd1 - 2); puts("((");
        gotoxy(seaweed[i], MapWallHEnd1 - 3); puts("))");
        gotoxy(seaweed[i], MapWallHEnd1 - 4); puts("((");
    }
    gotoxy(3, MapWallHEnd1); printf("Level : %d", game->level);
    gotoxy(20, MapWallHEnd1); printf("Score : %d", game->score);
    gotoxy(40, MapWallHEnd1); printf("Eat Fish : %d", game->eat);
    gotoxy(60, MapWallHEnd1); printf("Life : %d", shark.life);
    gotoxy(80, MapWallHEnd1); printf("Time : %d", (GetTickCount() - game->time) / 1000);
}
```



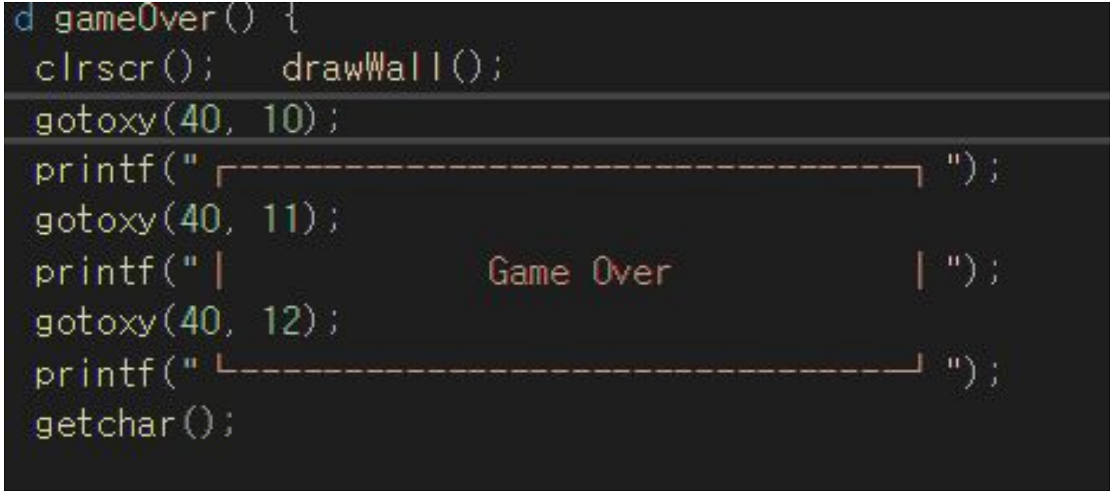
```
Level : 1          Score : 0            Eat Fish : 0          Life : 1            Time : 60
```

BackGround



```
void make_background(Game* game) {
    // seaweed
    for (int i = 0; i < 10; i++) {
        int seaweedX = rand() % 100 + 5;
        seaweed[i] = seaweedX;
    }
    game->level = 1; game->score = 0; game->eat = 0; game->time = GetTickCount();
}
```

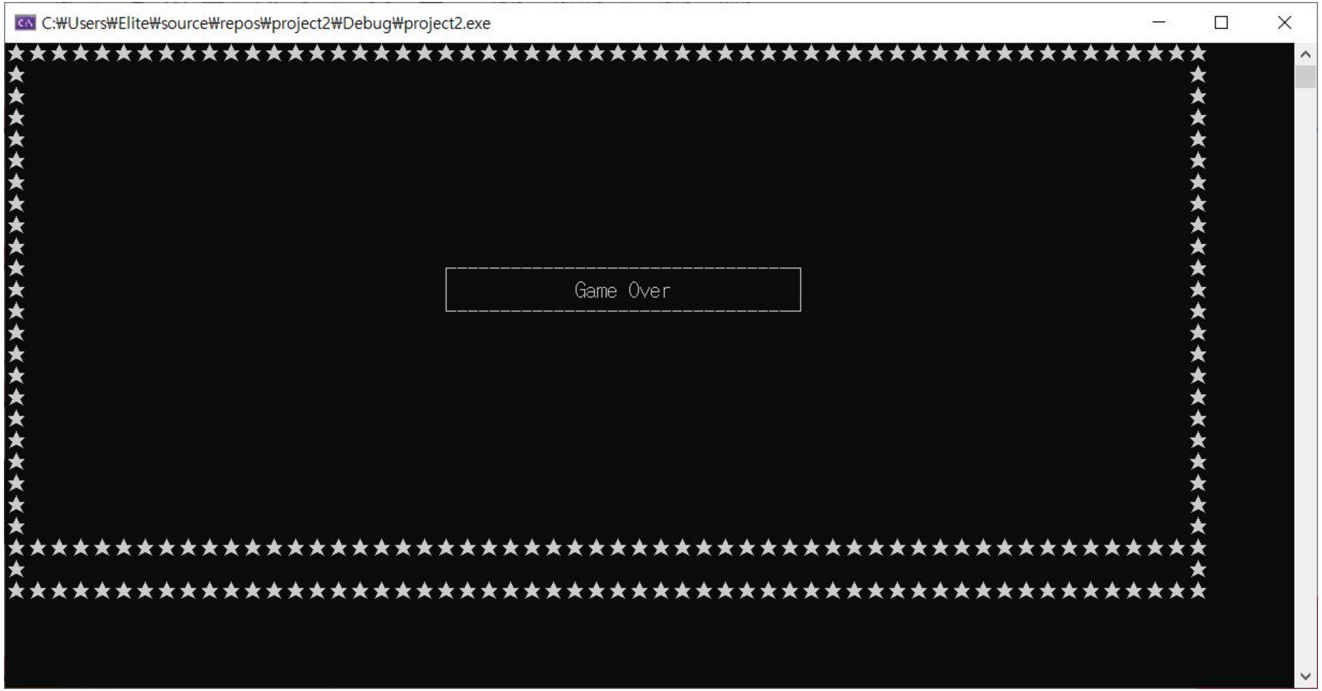
BackGround



```
d gameOver() {
  clrscr();   drawWall();
  gotoxy(40, 10);
  printf("┌──────────────────────────────────┐");
  gotoxy(40, 11);
  printf("│            Game Over             │");
  gotoxy(40, 12);
  printf("└──────────────────────────────────┘");
  getchar();
```

Shark Game - Function

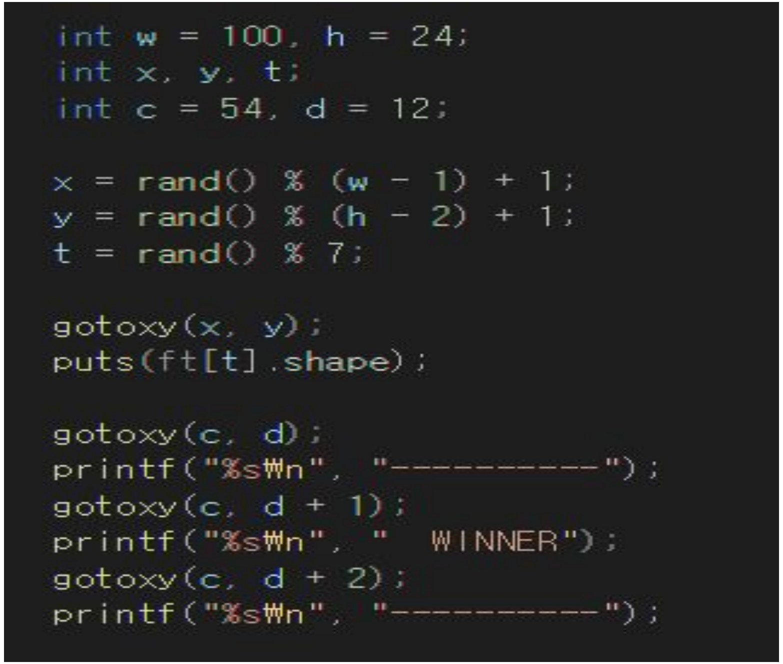
BackGround



C:₩Users₩Elite₩source₩repos₩project2₩Debug₩project2.exe

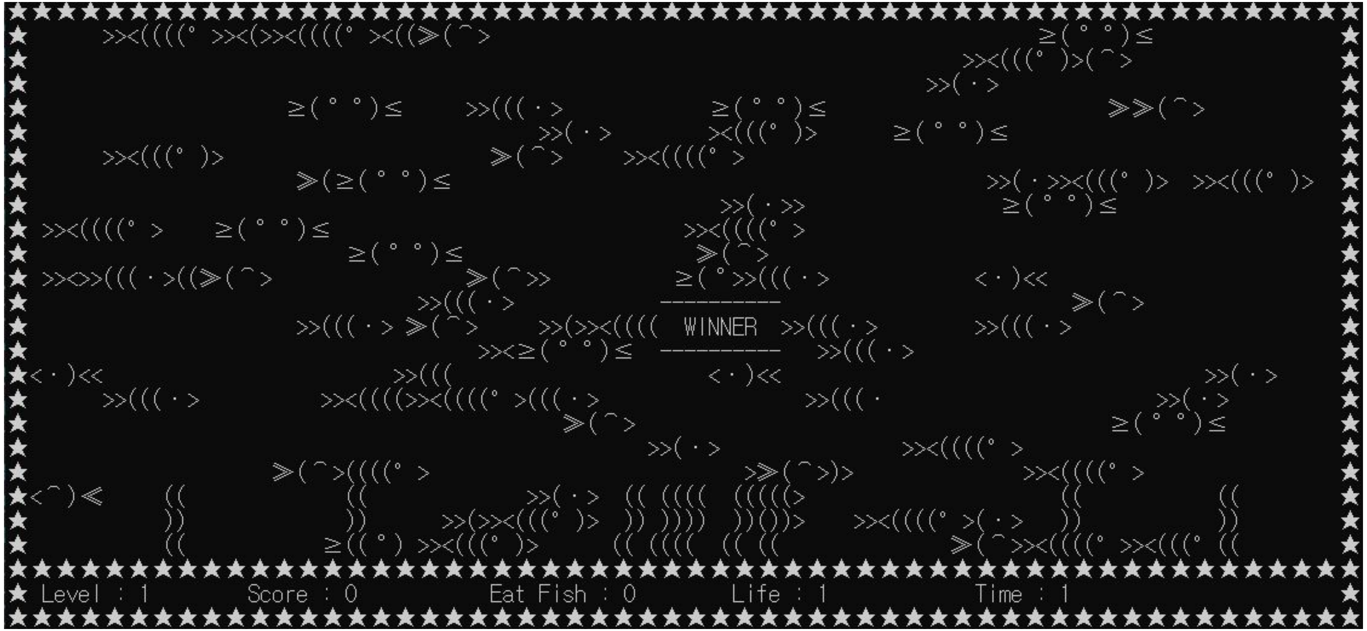
```
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                       ---------------------------------                                    ★
★                                       |           Game Over           |                                    ★
★                                       ---------------------------------                                    ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★                                                                                                            ★
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
```

BackGround



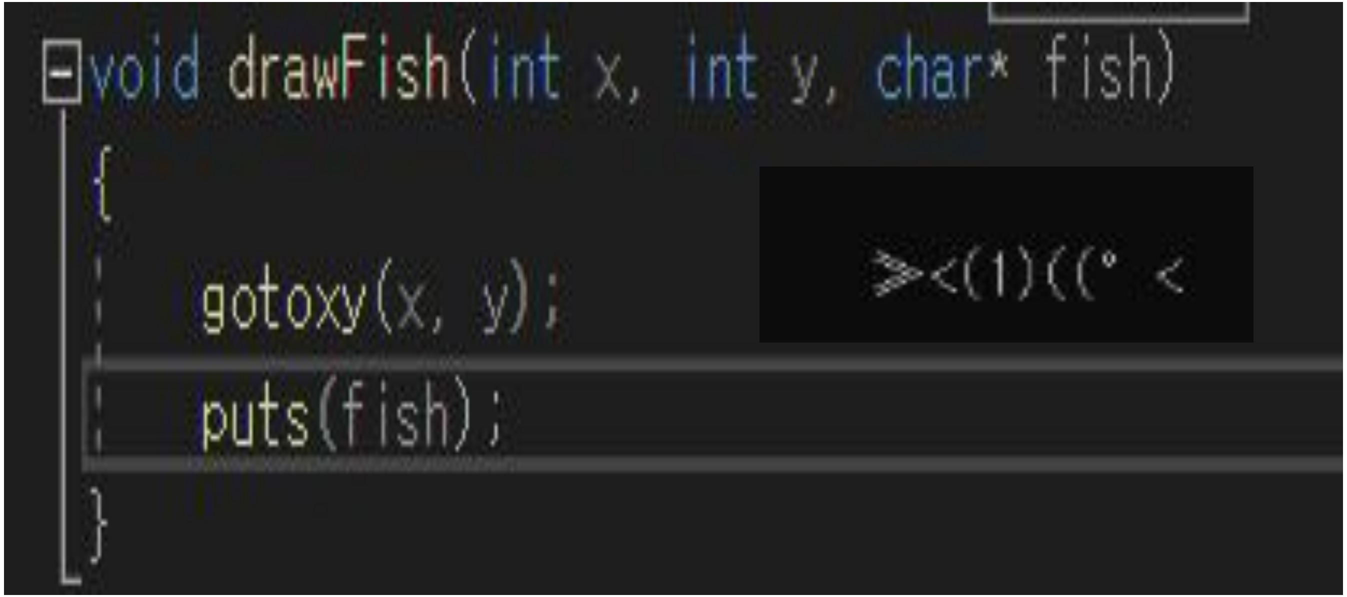
```
int w = 100, h = 24;
int x, y, t;
int c = 54, d = 12;

x = rand() % (w - 1) + 1;
y = rand() % (h - 2) + 1;
t = rand() % 7;

gotoxy(x, y);
puts(ft[t].shape);

gotoxy(c, d);
printf("%s₩n", "----------");
gotoxy(c, d + 1);
printf("%s₩n", "  WINNER");
gotoxy(c, d + 2);
printf("%s₩n", "----------");
```

BackGround



```
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★      >><((((°  >><(>><((((°  ><((≫(^^>                                  ≥(° °)≤     ★
★                                                                   >><(((°  )>(^^>          ★
★                                                                 >>( ･>                      ★
★                      ≥(° °)≤     >>(((･>          ≥(° °)≤                    ≫≫(^^>        ★
★                                       >>(･>       ><(((°  )>     ≥(° °)≤                    ★
★      >><(((°  )>                     ≫(^^>    >><((((° >                                    ★
★                       ≫(≥(° °)≤                                     >>(･>><(((°  )>  >><(((°  )>  ★
★                                                   >>(･>>            ≥(° °)≤                 ★
★  >><((((° >     ≥(° °)≤                         >><((((° >                                  ★
★                           ≥(° °)≤                ≫(^^>                                      ★
★  >><>>(((･>((≫(^^>                  ≫(^^>>       ≥(°>>(((･>              <･)<<              ★
★                                   >>(((･>       -----------                    ≫(^^>         ★
★                       >>(((･> ≫(^^>     >>(>><((((   WINNER   >>(((･>      >>(((･>           ★
★                                    >><≥(° °)≤    -----------   >>(((･>                       ★
★<･)<<                          >>(((                 <･)<<                            >>(･>   ★
★      >>(((･>            >><((((>><((((°  >(((･>            >>(((･                  >>(･>    ★
★                                           ≫(^^>                               ≥(° °)≤        ★
★                                                >>(･>             >><((((° >                  ★
★                     ≫(^^>((((° >                      ≫≫(^^>)>          >><((((° >           ★
★<^^)≪       ((           ((                >>(･>  (( ((((  (((((>             ((          ((   ★
★            ))           ))           >>(>><(((°  )>  )) ))))  ))())>   >><((((° >(･>   ))          ))   ★
★            ((          ≥((°) >><(((°  )>         (( ((((  (( ((          ≫(^^>><((((° >><(((°  ((   ★
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★ Level : 1          Score : 0              Eat Fish : 0          Life : 1             Time : 1         ★
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
```

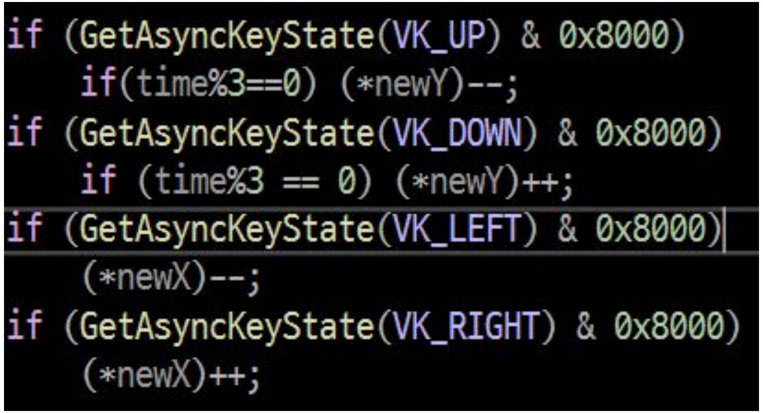
Fish - draw_Fish



```
void drawFish(int x, int y, char* fish)
{
    gotoxy(x, y);
    puts(fish);
}
```

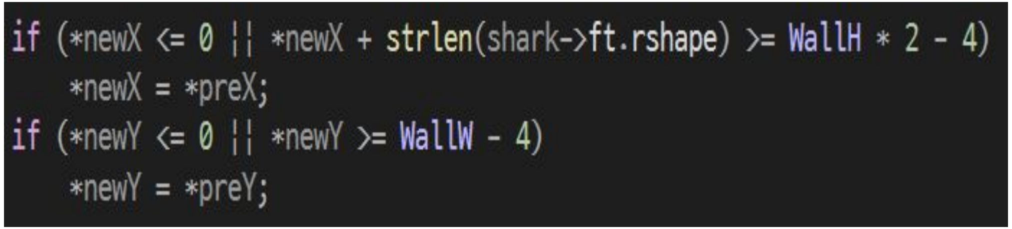
```
≫<(1)((° <
```

Fish - move_Shark()



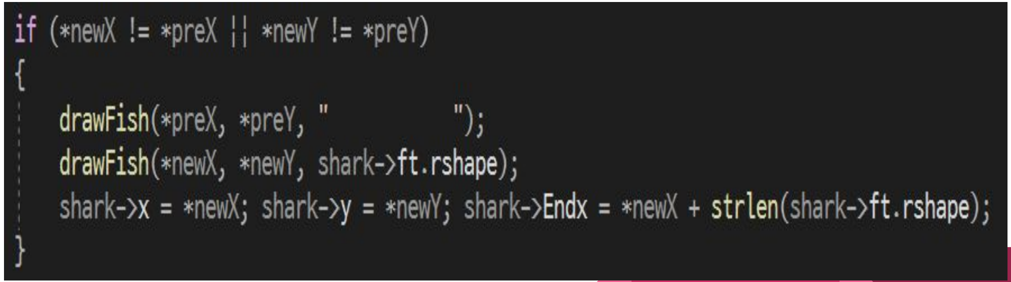
```
if (GetAsyncKeyState(VK_UP) & 0x8000)
    if(time%3==0) (*newY)--;
if (GetAsyncKeyState(VK_DOWN) & 0x8000)
    if (time%3 == 0) (*newY)++;
if (GetAsyncKeyState(VK_LEFT) & 0x8000)
    (*newX)--;
if (GetAsyncKeyState(VK_RIGHT) & 0x8000)
    (*newX)++;
```

키보드 입력 받아 움직임



```
if (*newX <= 0 || *newX + strlen(shark->ft.rshape) >= WallH * 2 - 4)
    *newX = *preX;
if (*newY <= 0 || *newY >= WallW - 4)
    *newY = *preY;
```

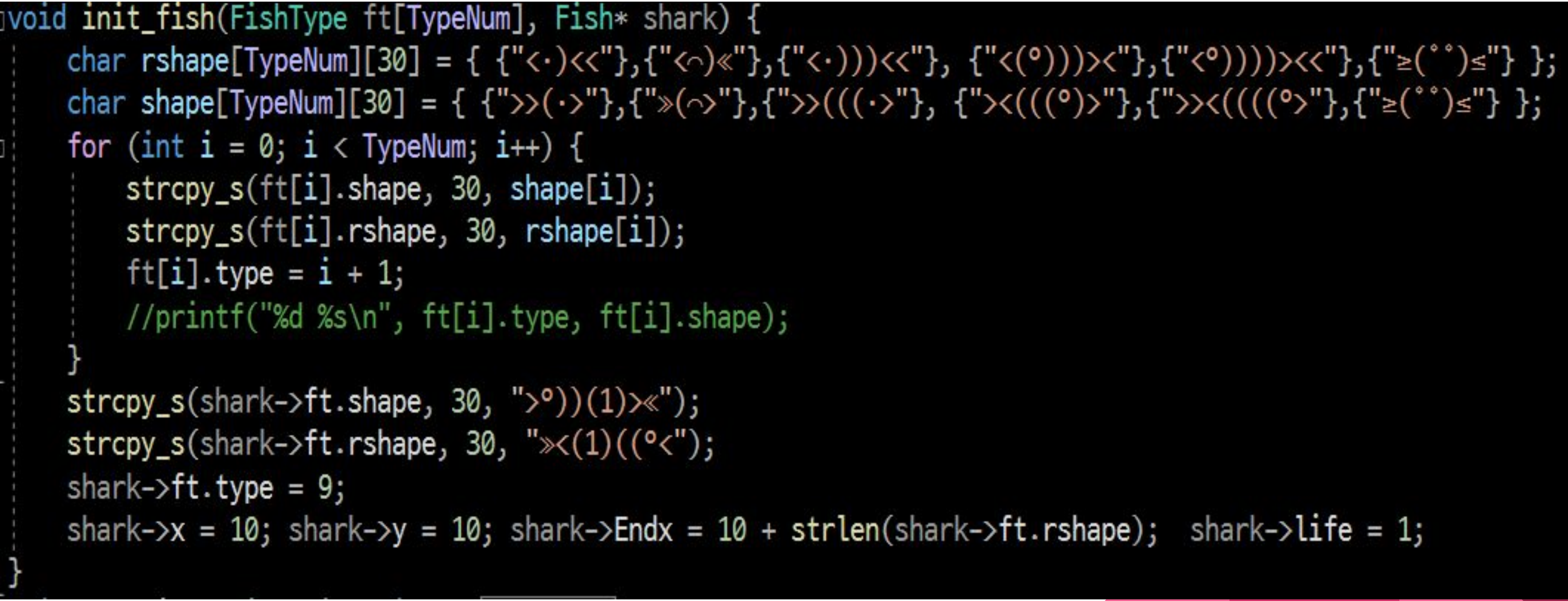
벽 체크



```
if (*newX != *preX || *newY != *preY)
{
    drawFish(*preX, *preY, "          ");
    drawFish(*newX, *newY, shark->ft.rshape);
    shark->x = *newX; shark->y = *newY; shark->Endx = *newX + strlen(shark->ft.rshape);
}
```

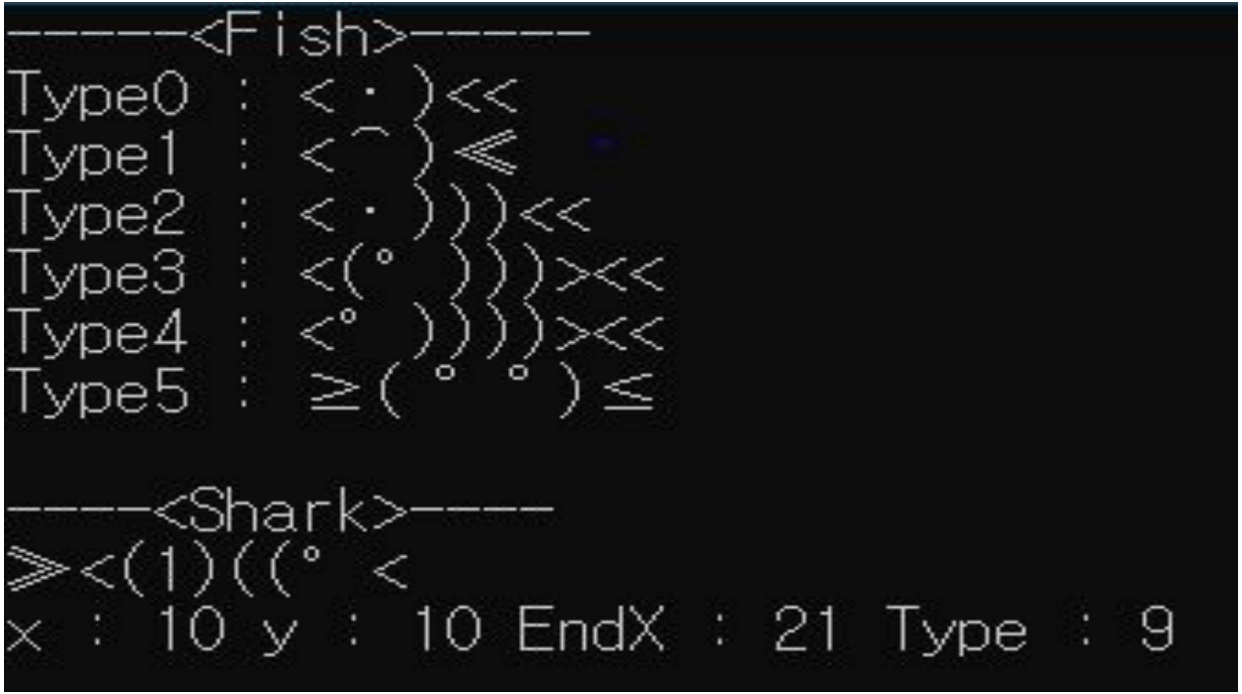
위치 변경 및 fish 다시 그리기

Fish - init_fish()



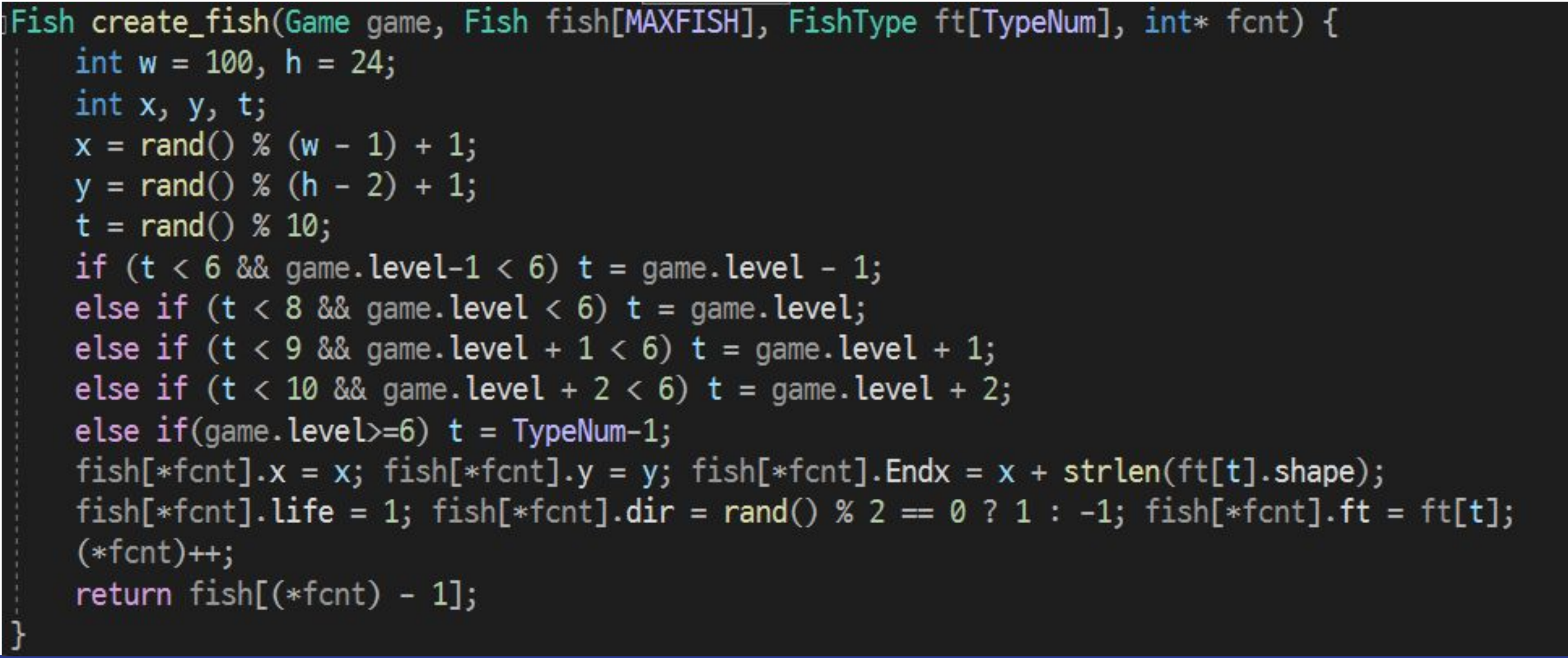
```
void init_fish(FishType ft[TypeNum], Fish* shark) {
    char rshape[TypeNum][30] = { {"<·)<<"},{"<⌒)«"},{"<·)))<<"}, {"<(°)))><"},{"<°))))><<"},{"≥(˚˚)≤"} };
    char shape[TypeNum][30] = { {">>(·>"},{"»(⌒>"},{">>(((·>"}, {"><(((°)>"},{">><((((°>"},{"≥(˚˚)≤"} };
    for (int i = 0; i < TypeNum; i++) {
        strcpy_s(ft[i].shape, 30, shape[i]);
        strcpy_s(ft[i].rshape, 30, rshape[i]);
        ft[i].type = i + 1;
        //printf("%d %s\n", ft[i].type, ft[i].shape);
    }
    strcpy_s(shark->ft.shape, 30, ">°))(1)><«");
    strcpy_s(shark->ft.rshape, 30, "»<(1)((°<");
    shark->ft.type = 9;
    shark->x = 10; shark->y = 10; shark->Endx = 10 + strlen(shark->ft.rshape);  shark->life = 1;
}
```

Fish - init_fish()



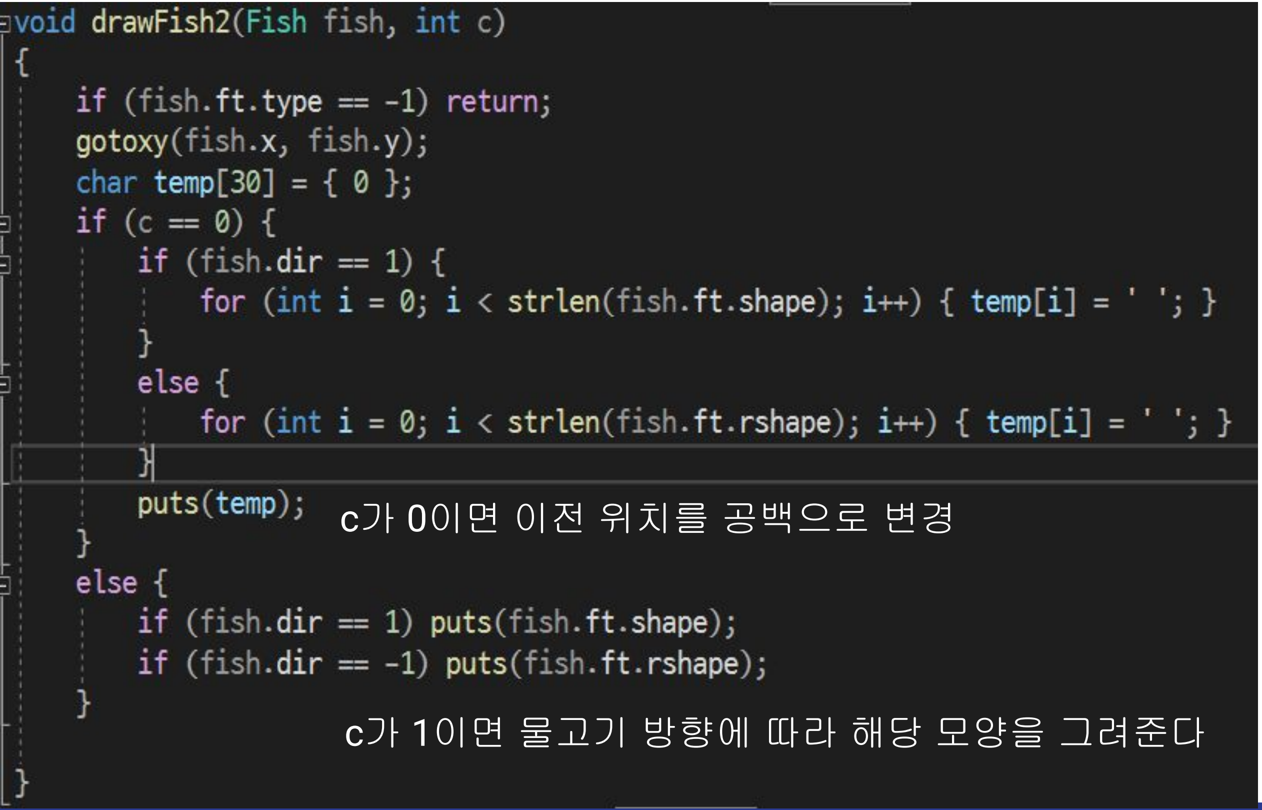
```
-----<Fish>-----
Type0 : < · )<<
Type1 : <⌒ )≪
Type2 : < · )))<<
Type3 : <(°  )))><<
Type4 : <°  ))))><<
Type5 : ≥( ° ° )≤

----<Shark>----
≫<(1)((° <
x : 10 y : 10 EndX : 21 Type : 9
```

Fish - create_fish()



```
Fish create_fish(Game game, Fish fish[MAXFISH], FishType ft[TypeNum], int* fcnt) {
    int w = 100, h = 24;
    int x, y, t;
    x = rand() % (w - 1) + 1;
    y = rand() % (h - 2) + 1;
    t = rand() % 10;
    if (t < 6 && game.level-1 < 6) t = game.level - 1;
    else if (t < 8 && game.level < 6) t = game.level;
    else if (t < 9 && game.level + 1 < 6) t = game.level + 1;
    else if (t < 10 && game.level + 2 < 6) t = game.level + 2;
    else if(game.level>=6) t = TypeNum-1;
    fish[*fcnt].x = x; fish[*fcnt].y = y; fish[*fcnt].Endx = x + strlen(ft[t].shape);
    fish[*fcnt].life = 1; fish[*fcnt].dir = rand() % 2 == 0 ? 1 : -1; fish[*fcnt].ft = ft[t];
    (*fcnt)++;
    return fish[(*fcnt) - 1];
}
```

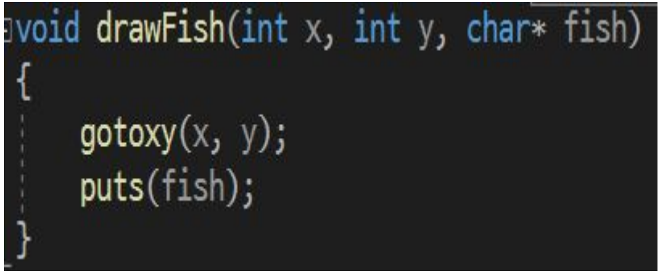
Fish - drawFish2()



```
void drawFish2(Fish fish, int c)
{
    if (fish.ft.type == -1) return;
    gotoxy(fish.x, fish.y);
    char temp[30] = { 0 };
    if (c == 0) {
        if (fish.dir == 1) {
            for (int i = 0; i < strlen(fish.ft.shape); i++) { temp[i] = ' '; }
        }
        else {
            for (int i = 0; i < strlen(fish.ft.rshape); i++) { temp[i] = ' '; }
        }
        puts(temp);
    }
    else {
        if (fish.dir == 1) puts(fish.ft.shape);
        if (fish.dir == -1) puts(fish.ft.rshape);
    }
}
```

c가 0이면 이전 위치를 공백으로 변경

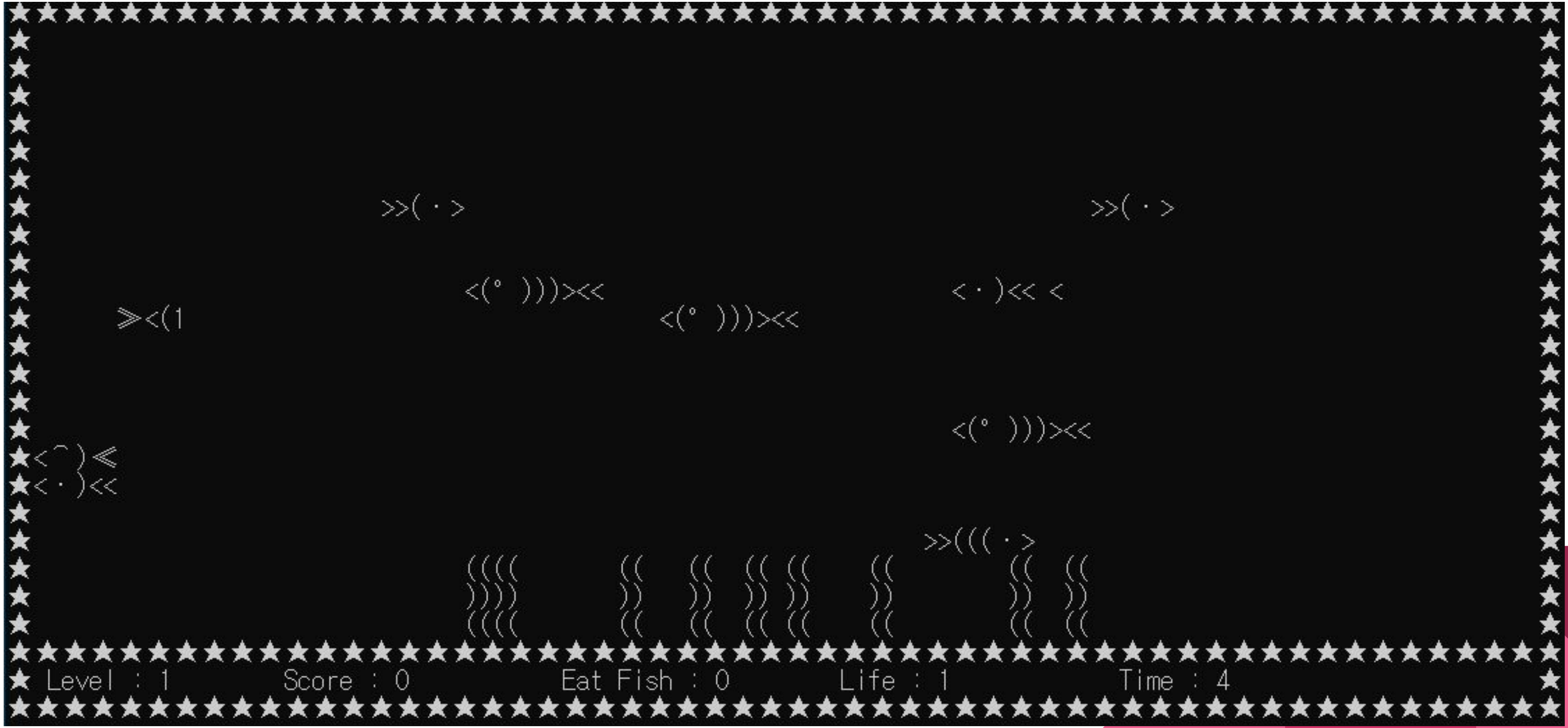
c가 1이면 물고기 방향에 따라 해당 모양을 그려준다



```
void drawFish(int x, int y, char* fish)
{
    gotoxy(x, y);
    puts(fish);
}
```

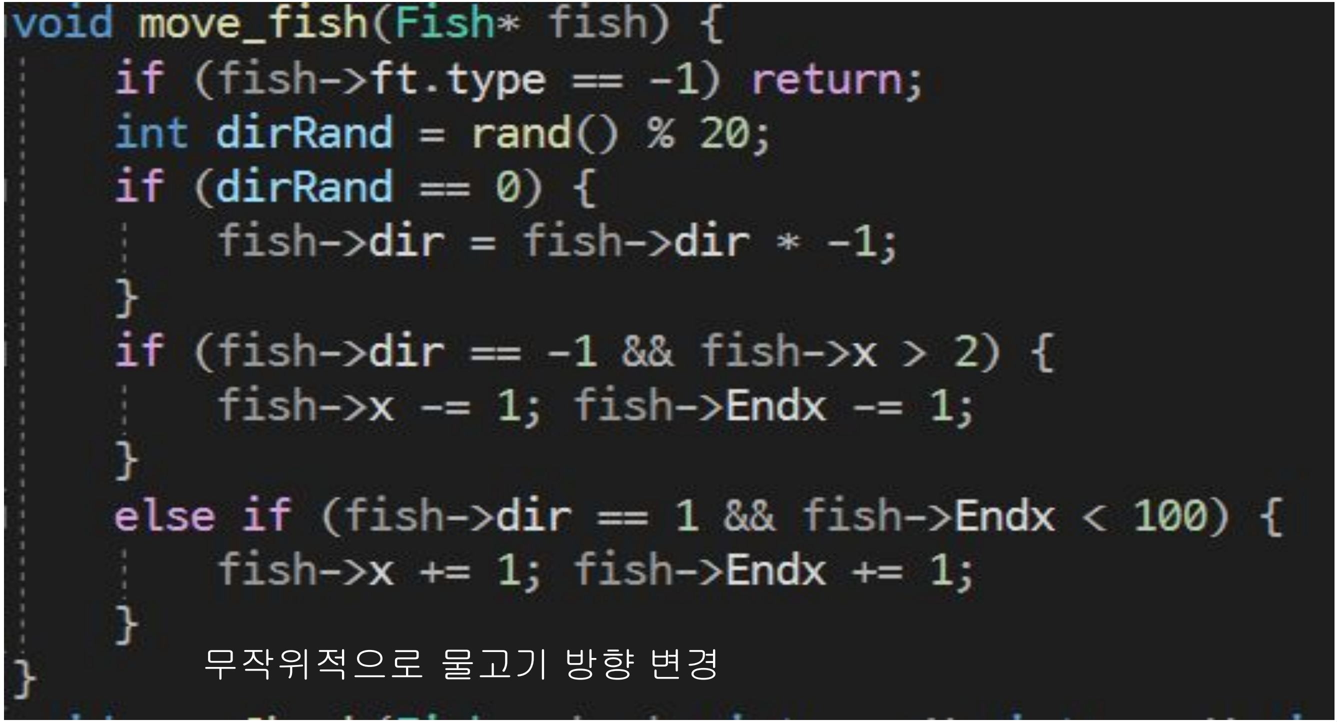
플레이어가 움직인 위치에 그려준다

Fish - drawFish2()



```
★ Level : 1        Score : 0        Eat Fish : 0        Life : 1        Time : 4        ★
```

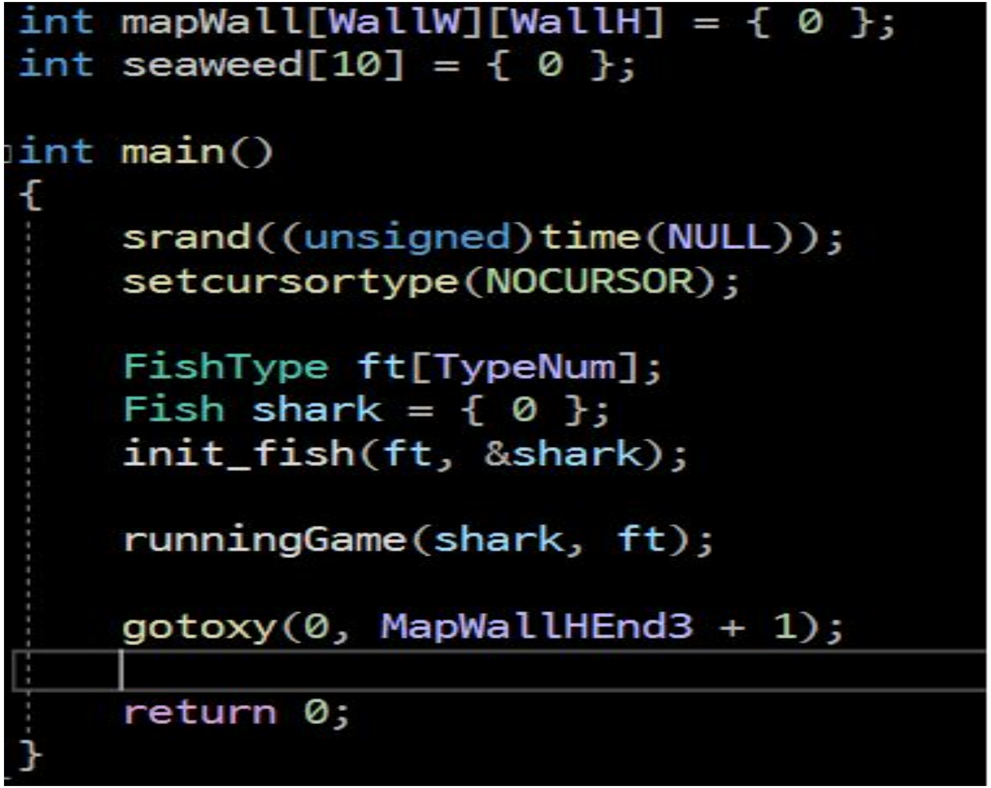
Fish - move_fish()



```c
void move_fish(Fish* fish) {
    if (fish->ft.type == -1) return;
    int dirRand = rand() % 20;
    if (dirRand == 0) {
        fish->dir = fish->dir * -1;
    }
    if (fish->dir == -1 && fish->x > 2) {
        fish->x -= 1; fish->Endx -= 1;
    }
    else if (fish->dir == 1 && fish->Endx < 100) {
        fish->x += 1; fish->Endx += 1;
    }
}
```

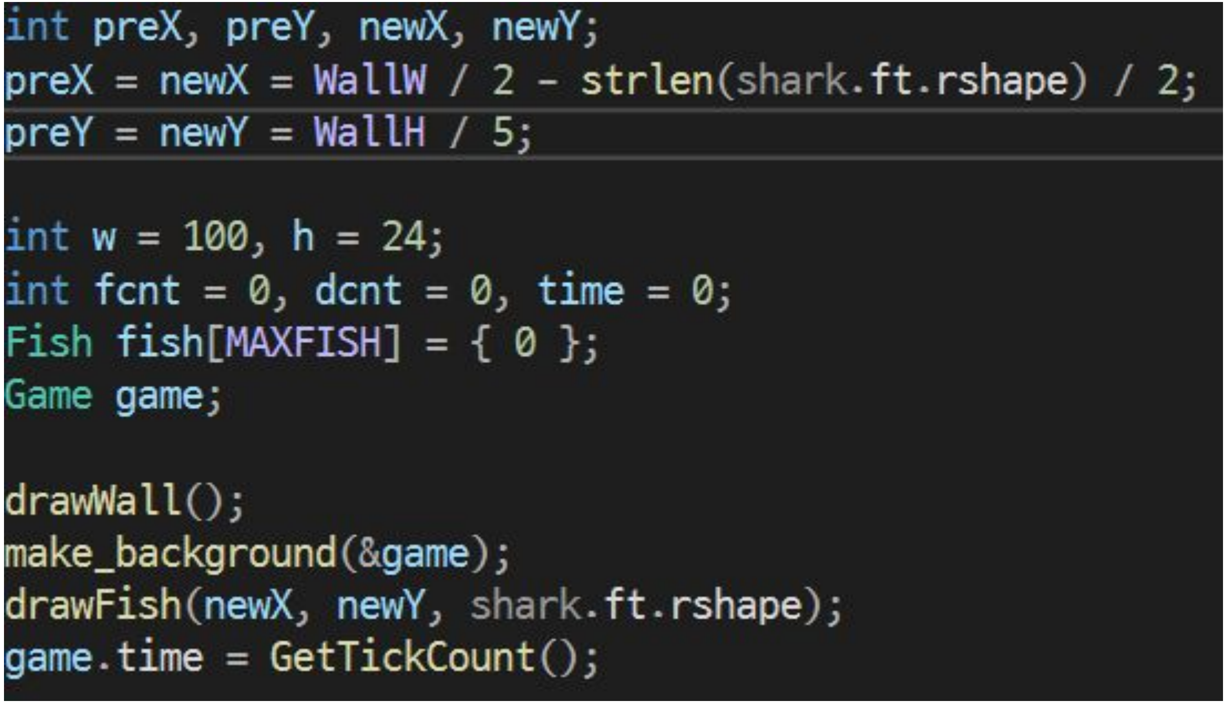
무작위적으로 물고기 방향 변경

main()



```
int mapWall[WallW][WallH] = { 0 };
int seaweed[10] = { 0 };

int main()
{
    srand((unsigned)time(NULL));
    setcursortype(NOCURSOR);

    FishType ft[TypeNum];
    Fish shark = { 0 };
    init_fish(ft, &shark);

    runningGame(shark, ft);

    gotoxy(0, MapWallHEnd3 + 1);

    return 0;
}
```

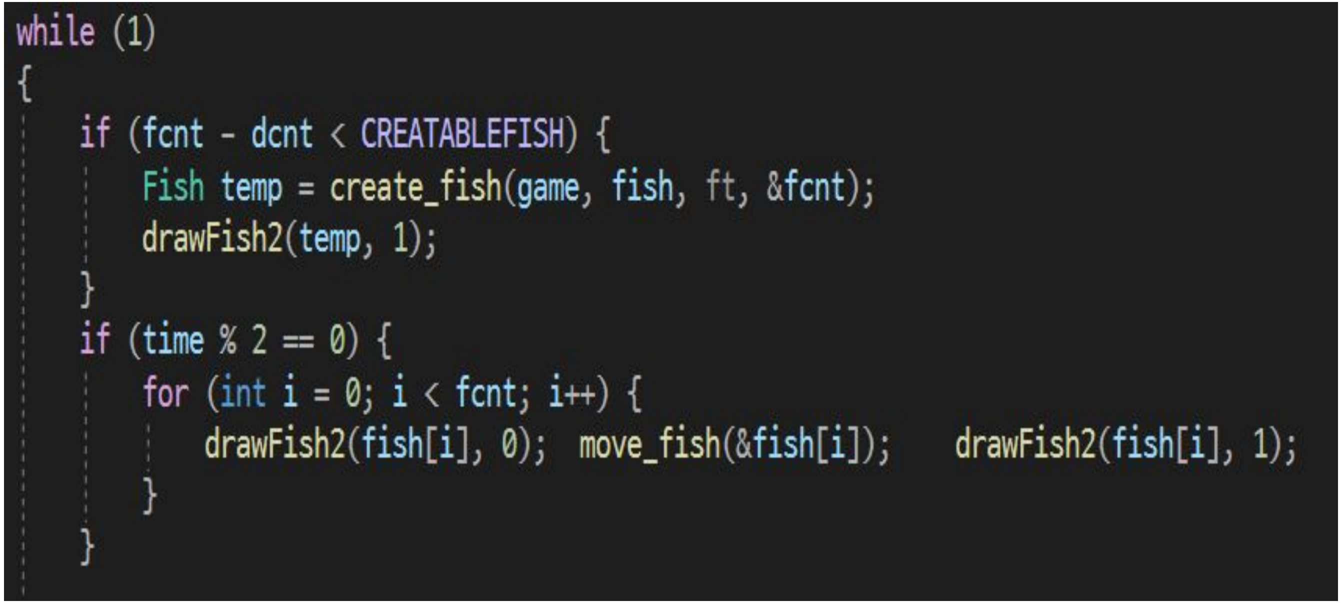
Game - runningGame()



```
int preX, preY, newX, newY;
preX = newX = WallW / 2 - strlen(shark.ft.rshape) / 2;
preY = newY = WallH / 5;

int w = 100, h = 24;
int fcnt = 0, dcnt = 0, time = 0;
Fish fish[MAXFISH] = { 0 };
Game game;

drawWall();
make_background(&game);
drawFish(newX, newY, shark.ft.rshape);
game.time = GetTickCount();
```

변수 선언 및 게임 요소 생성

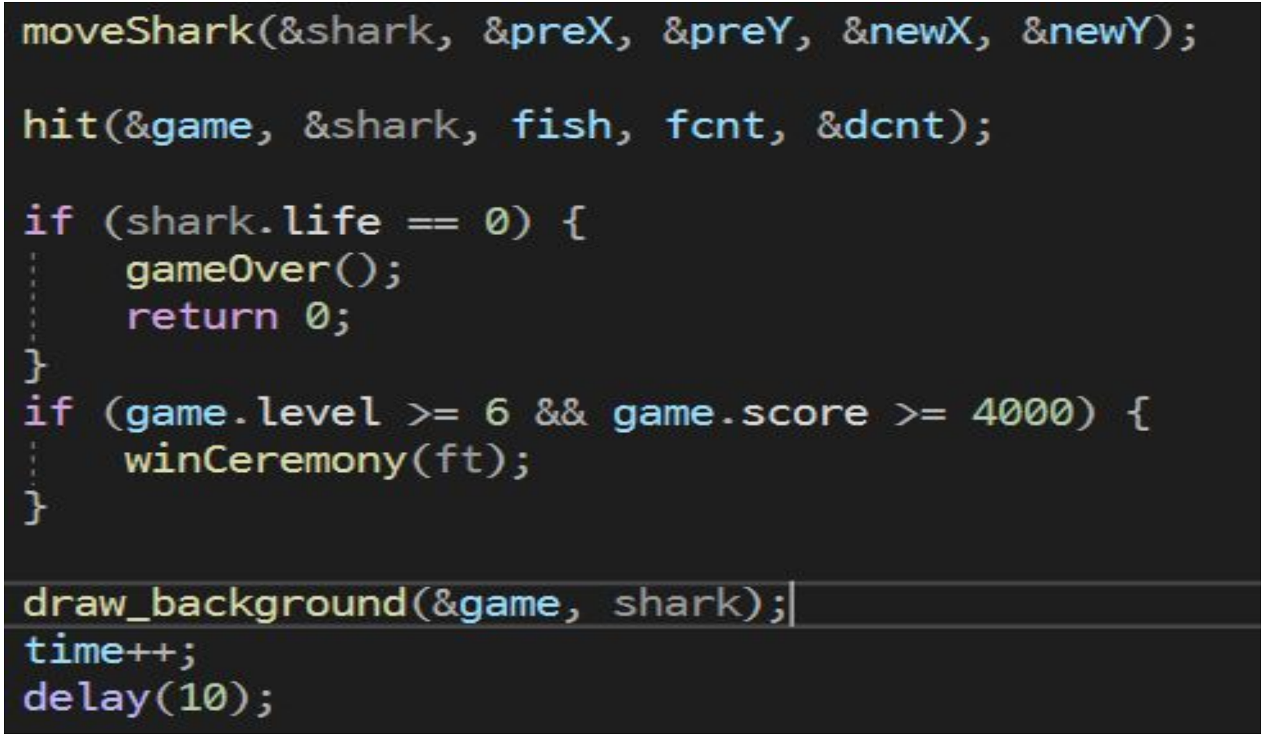
Game - runningGame()



```
while (1)
{
    if (fcnt - dcnt < CREATABLEFISH) {
        Fish temp = create_fish(game, fish, ft, &fcnt);
        drawFish2(temp, 1);
    }
    if (time % 2 == 0) {
        for (int i = 0; i < fcnt; i++) {
            drawFish2(fish[i], 0);  move_fish(&fish[i]);    drawFish2(fish[i], 1);
        }
    }
```

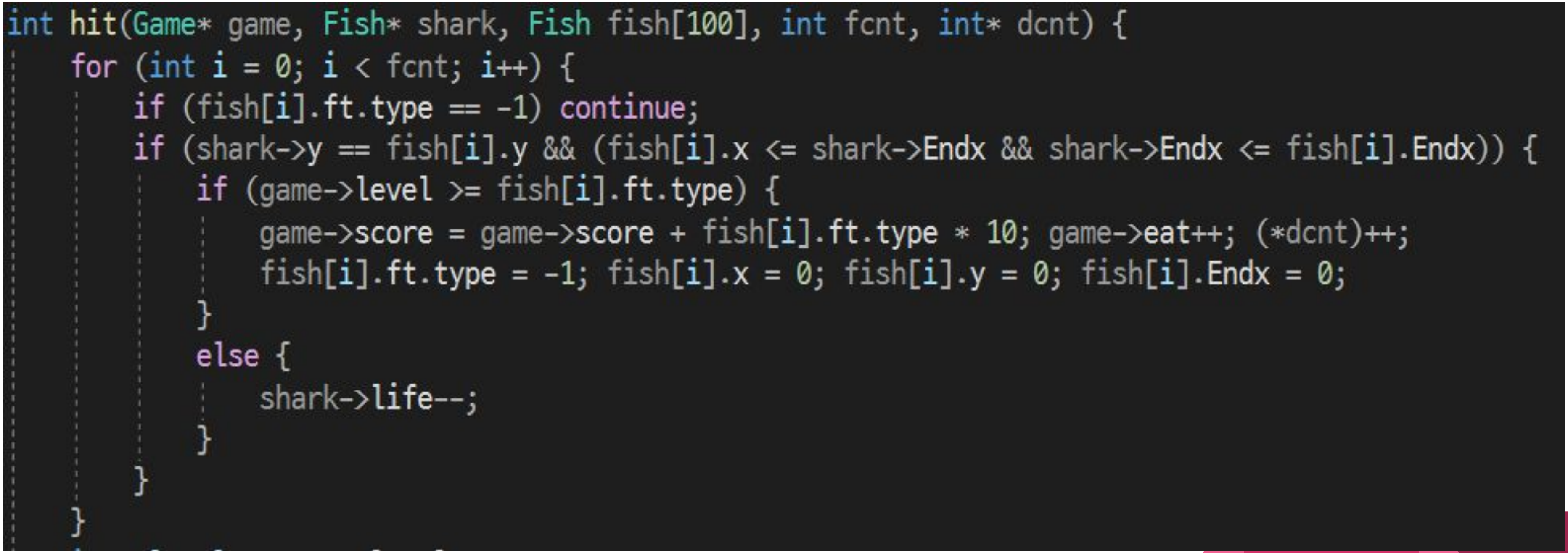
Fish 생성 및 이동

Game - runningGame()



```
moveShark(&shark, &preX, &preY, &newX, &newY);

hit(&game, &shark, fish, fcnt, &dcnt);

if (shark.life == 0) {
    gameOver();
    return 0;
}
if (game.level >= 6 && game.score >= 4000) {
    winCeremony(ft);
}

draw_background(&game, shark);
time++;
delay(10);
```

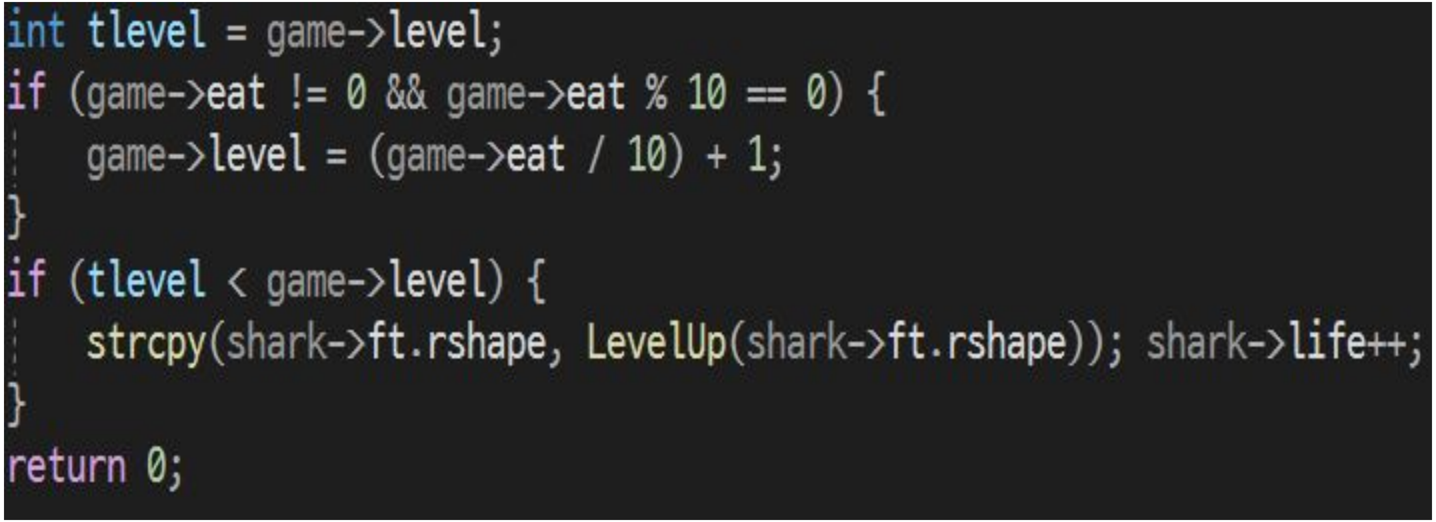
Game - hit()



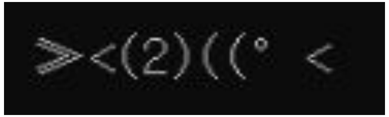
```
int hit(Game* game, Fish* shark, Fish fish[100], int fcnt, int* dcnt) {
    for (int i = 0; i < fcnt; i++) {
        if (fish[i].ft.type == -1) continue;
        if (shark->y == fish[i].y && (fish[i].x <= shark->Endx && shark->Endx <= fish[i].Endx)) {
            if (game->level >= fish[i].ft.type) {
                game->score = game->score + fish[i].ft.type * 10; game->eat++; (*dcnt)++;
                fish[i].ft.type = -1; fish[i].x = 0; fish[i].y = 0; fish[i].Endx = 0;
            }
            else {
                shark->life--;
            }
        }
    }
```

hit 판정

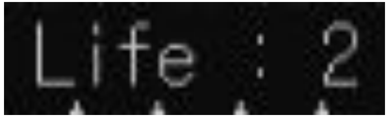
Game - hit()



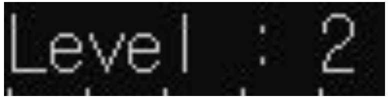
```
int tlevel = game->level;
if (game->eat != 0 && game->eat % 10 == 0) {
    game->level = (game->eat / 10) + 1;
}
if (tlevel < game->level) {
    strcpy(shark->ft.rshape, LevelUp(shark->ft.rshape)); shark->life++;
}
return 0;
```



```
≫<(2)((°  <
```



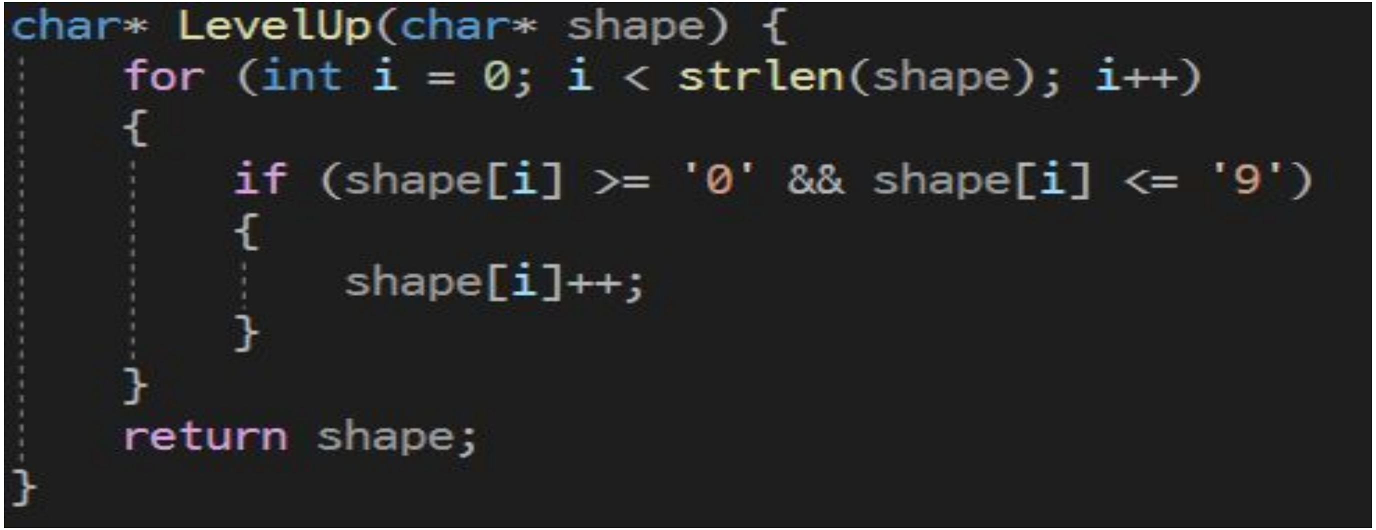
```
Life : 2
```



```
Level : 2
```

hit 후처리

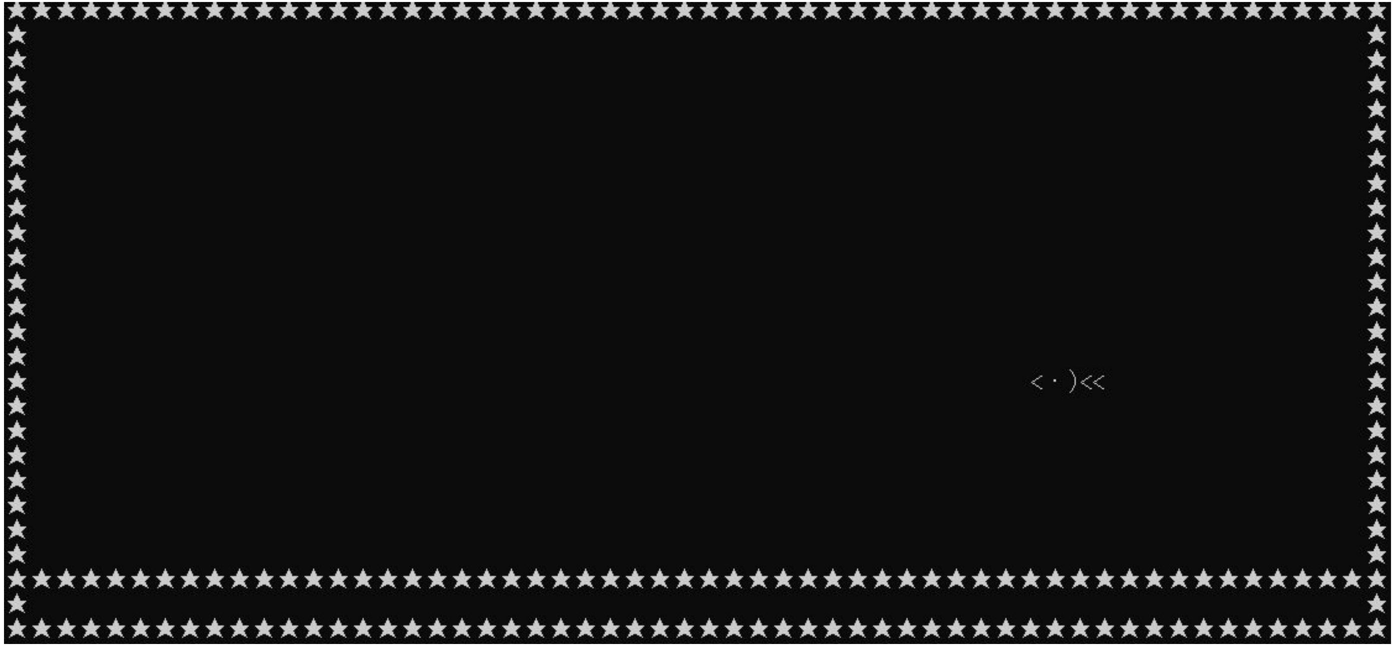
Game - LevelUp()



```
char* LevelUp(char* shape) {
    for (int i = 0; i < strlen(shape); i++)
    {
        if (shape[i] >= '0' && shape[i] <= '9')
        {
            shape[i]++;
        }
    }
    return shape;
}
```

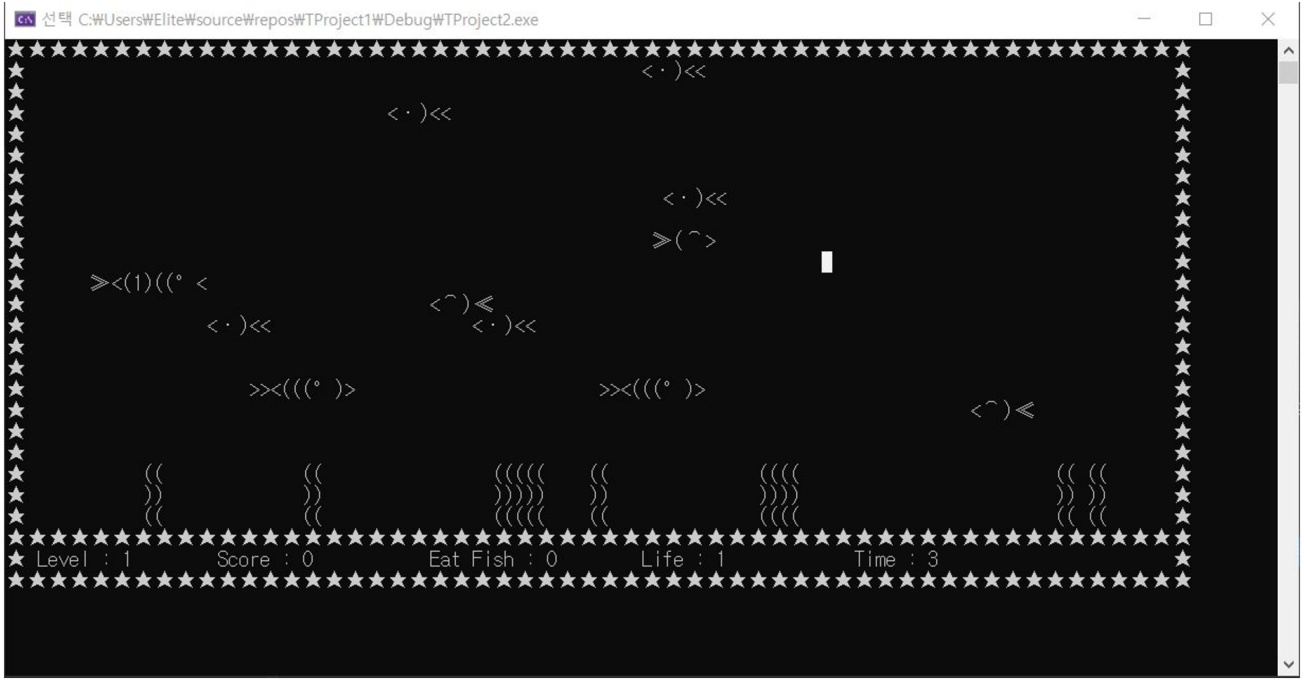
레벨업 후 Shark 모양 변경

Conclusion



```
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                <·)<<                       ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★                                                                                                            ★
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★                                                                                                            ★
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
```

Conclusion



선택 C:₩Users₩Elite₩source₩repos₩TProject1₩Debug₩TProject2.exe

```
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★                                                           <·)<<                                             ★
★                                                                                                             ★
★                                   <·)<<                                                                     ★
★                                                                                                             ★
★                                                                                                             ★
★                                                                                                             ★
★                                                             <·)<<                                           ★
★                                                                                                             ★
★                                                            ≫(^>                                             ★
★                                                                                                             ★
★       ≫<(1)((° <                                                                                            ★
★                                       <^)≪                                                                  ★
★                  <·)<<                    <·)<<                                                             ★
★                                                                                                             ★
★                                                                                                             ★
★                      >><(((° )>                       >><(((° )>                                            ★
★                                                                                          <^)≪               ★
★                                                                                                             ★
★                                                                                                             ★
★            ((             ((                (((((    ((              ((((                        (( ((      ★
★            ))             ))                )))))    ))              ))))                        )) ))      ★
★            ((             ((                (((((    ((              ((((                        (( ((      ★
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
★  Level : 1        Score : 0           Eat Fish : 0        Life : 1            Time : 3                      ★
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★★
```

Q n A

Team. 06

Thank you 4 Listening

Team. 06