스네이크 게임 제작

202121161 나주엽

규칙

2

- 가능한 한 많은 과일을 먹어기
- 스네이크가 화면의 경계나 자신의 몸통에 부딪히면 게임이 끝남

+

- 진행할 수 록 벽이 생성됨

Main.c


```c
int main() {

    system("cls");
    Setup();
    spawnItem();
    spawnItem();
    spawnItem();
    display_update(1);


    while (gmaePlay) {
        if (dir != STOP) tick ++;

        Sleep(200);

        Input();
        if ((tick % 50) == 1) {
            objectSpawn();
        };
        objectBuffer();

        display_update(0);
        move();

    }
    printf("Game Over!\nPress Enter to exit...");
    getchar();
    return 0;
}
```

Display.h


```
typedef struct boxData {
    int x;
    int y;
    bool update;
    int object;

} boxData;

void boxUpdate(int x, int y, int type) {
    gameBox[y][x].x = x;
    gameBox[y][x].y = y;
    gameBox[y][x].object = type;
    gameBox[y][x].update = 1;

}

void display_update_border(int forceUpdate) {
    boxData *p = &gameBox[0][0];
    for (int i = 0; i < GAME_BORDER_HEIGHT * GAME_BORDER_WIDTH; i++) {

        if (forceUpdate || p->update) {
            gotoxy(p->x, p->y);
            printf("%s", getObject(p->object));
            if (p->update != NULL) {
                p->update = 0;
            }
        }
        p ++;

    }
}
```

Move.h


```
int move() {
    if (dir != STOP) {
        moving = dir;
        removeSnakeHead(snakeBody[snakeSize].x, snakeBody[snakeSize].y);

        for (int i = snakeSize; i > 0; i--) {
            snakeBody[i] = snakeBody[i - 1];
        }

        if (dir == UP) {
            setSnakeHead(snakeBody[1].x, snakeBody[1].y - 1, 4);
        } else if (dir == DOWN){
            setSnakeHead(snakeBody[1].x, snakeBody[1].y + 1, 5);
        } else if (dir == RIGHT){
            setSnakeHead(snakeBody[1].x + 1, snakeBody[1].y, 6);
        } else if (dir == LEFT){
            setSnakeHead(snakeBody[1].x - 1, snakeBody[1].y, 7);
        } else {
            setSnakeHead(snakeBody[1].x, snakeBody[1].y, 8);
        }

    }
    return 0;


}
```

```
void setSnakeHead(int x, int y, int face) {
    collisionCheck(x, y);
    // printf("%d", collisionCheck(x, y));
    snakeBody[0].x = x;
    snakeBody[0].y = y;
    boxUpdate(x, y, face);
}
```

Collision.h


```
int collisionCheck(int x, int y) {
    // printf("%d", gameBox[y][x].object);
    if (gameBox[y][x].object == NULL || gameBox[y][x].object == 0) {
        return 0;
    } else if (gameBox[y][x].object == 1) {
        collisionWalls();
        return 1;
    } else if (gameBox[y][x].object == 2) {
        collisionFood();
        return 0;
    } else if (gameBox[y][x].object == 20 || gameBox[y][x].object == 21) {
        return 0;
    } else {
        collisionWalls();
        return -1;
    }
    return 0;
}
```

Item.h


```
void spawnItem() {
    srand(time(NULL));
    int x, y;
    while(randomLoc(&x, &y));
    boxUpdate(x, y, 2);
}



bool randomLoc(int *xValue, int * yValue) {

    int x = (rand() % (GAME_BORDER_WIDTH - 1)) + 1;
    int y = (rand() % (GAME_BORDER_HEIGHT - 1)) + 1;
    if (gameBox[y][x].object == NULL || gameBox[y][x].object == 0) {
        *xValue = x;
        *yValue = y;
        return 0;
    } else {
        return 1;
    }

}
```

감사합니다