게임프로젝트 발표

게임명: Makjang Froggggggy
교육과정: AI시스템반도체 설계
이름: 서윤철

목차

과제 개요

게임 설명

아들을 공격한 파리를 향한
엄마 개구리의 복수극

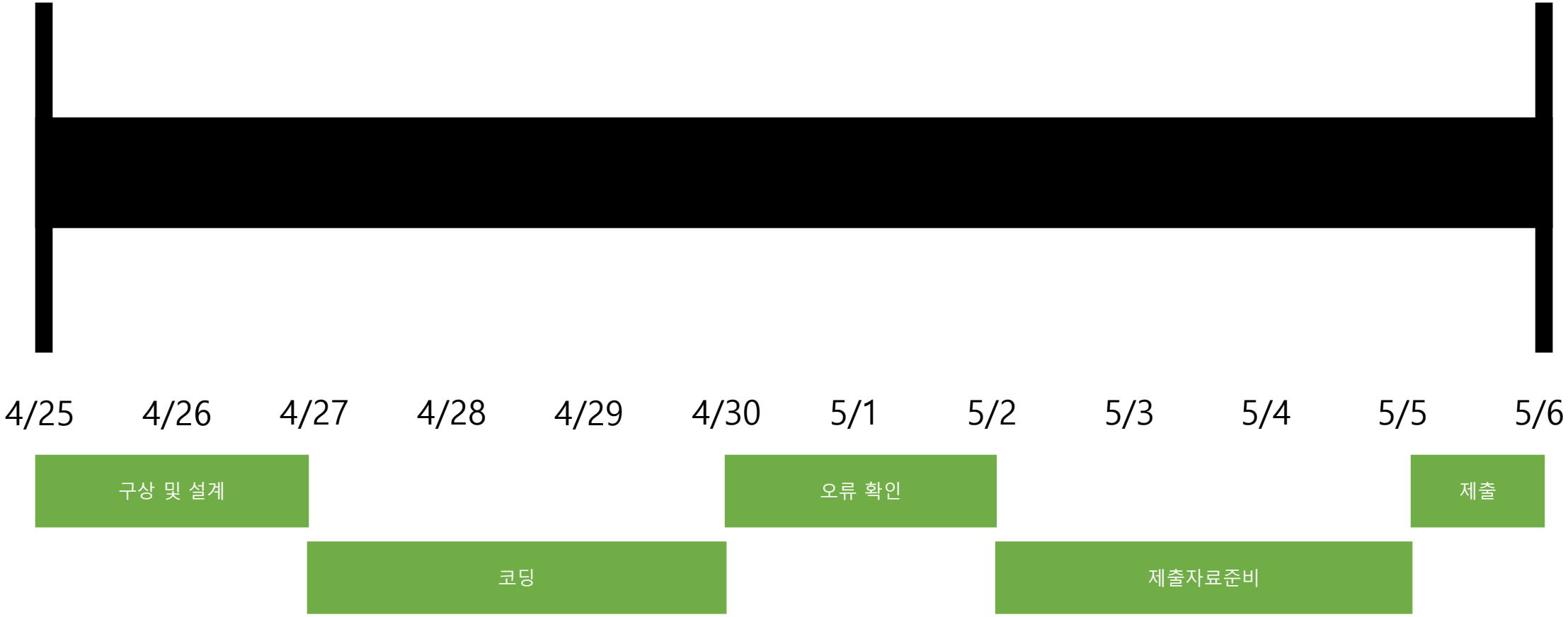
사용 실습 환경

Visual Studio Code

장비 소개

제품명: STM32F103

개발 일정



개발 결과

시연 영상

개발 결과

함수 목록

Main

for
- 초기 세팅
- for
  - if
    - Frog_Move
  - if
    - SysTick TIMEOUT
      - 충돌 관련 함수
      - 그물 관련 함수
      - Human_Move
      - Snake_Move
  - if
    - TIMER4 TIMEOUT
      - 배경음악 flag
  - if
    - 종료 관련 함수

타이머

Systick Timer: 100ms마다 Timeout. Main함수 내의 게임진행에 사용.
Timer3: 배경음(PWM) 생성에 사용.
Timer4: 10ms마다 timeout. 배경음 박자를 제어하는데 사용.

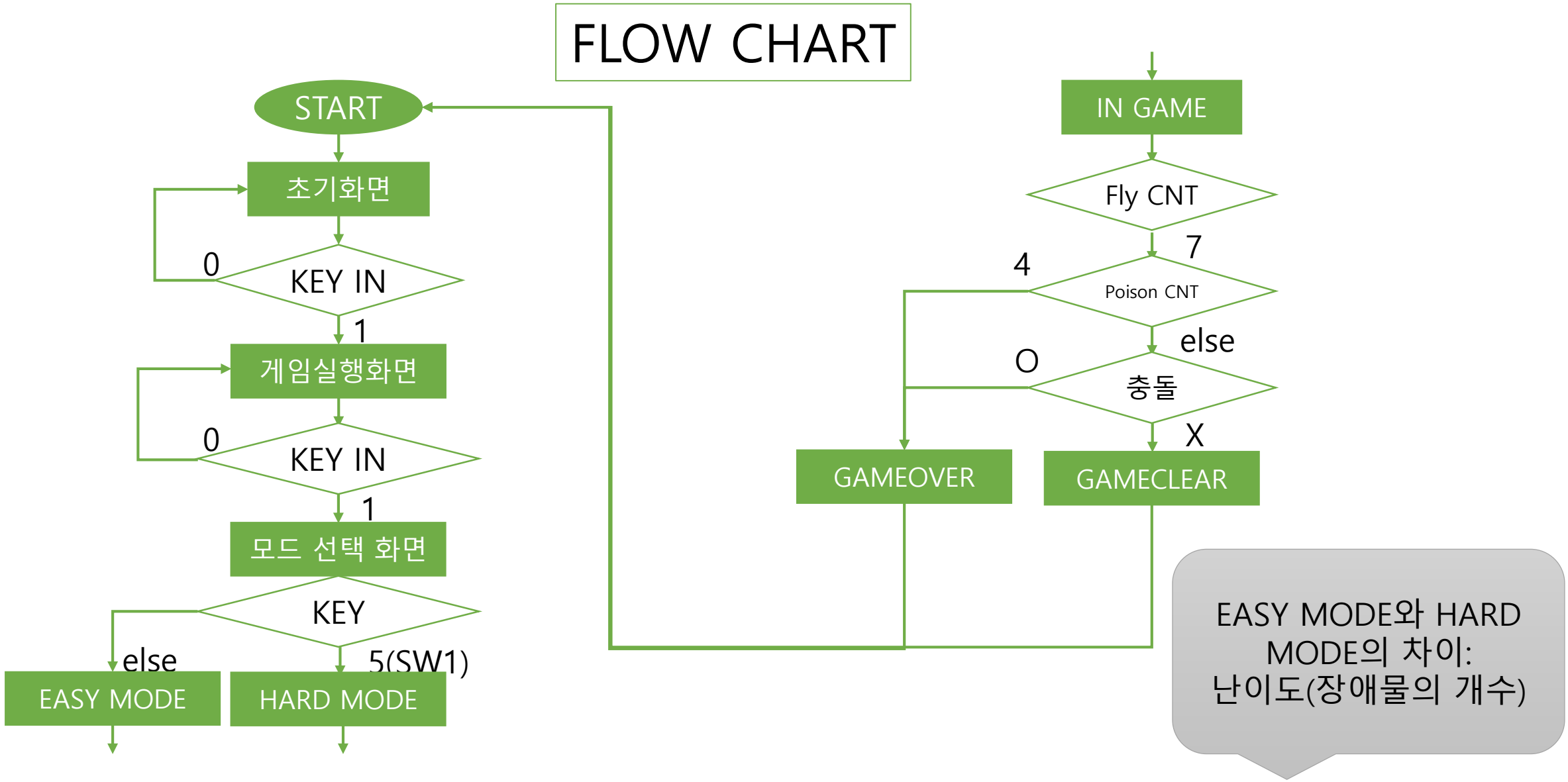
부함수

- Frog_Move
- 화면 출력
- Snake_Move
- Human_Move
- Fly_Move
- 충돌 관련 함수
- Place_Posion
- 초기 세팅
- 종료 관련 함수
- 그물 관련 함수

인터럽트

- EXTI9_5_IRQHandler
- TIM4_IRQHandler
- EXTI15_10_IRQHandler
- SysTick_Handler
- EXTI3_IRQHandler

개발 결과



EASY MODE와 HARD MODE의 차이:
난이도(장애물의 개수)

핵심 기술

CONCEPT

미로의 수호자: HUMAN

신성한 미로의 침입자를 처단한다.

벽 충돌 확인
파리와 개구리 사이의 거리 확인
다음 이동 방향 확인

핵심 기술

Human 변수

```
typedef struct{
    int x,y;//위치
    int w,h;//너비, 높이
    int ci;//색
    int dir;//방향 or 배치상태
}QUERY_DRAW;

static QUERY_DRAW snake[2], fly, poison[4], human;
```

핵심 기술

벽 충돌 확인

오브젝트의 위치 및 크기를 입력 받고,
해당 오브젝트의 범위가 벽에 해당하는 좌표와 겹치는 경우
=> 이전 위치로 복귀하도록 설계하여,
벽 범위로 침범하는 것을 방지

```
static int Check_Wall_Collision(int new_x, int new_y, int w, int h)
{
    int col_start = new_x / MAZE_CELL_SIZE;
    int row_start = new_y / MAZE_CELL_SIZE;
    int col_end = (new_x + w) / MAZE_CELL_SIZE;
    int row_end = (new_y + h) / MAZE_CELL_SIZE;
    int i, j;

    for(i=row_start; i<=row_end; i++)
    {
        for(j=col_start; j<=col_end; j++)
        {
            if(maze[i][j] == 1) return 1; // 충돌
        }
    }
    return 0; // 충돌 없음
}
```

핵심 기술

벽 충돌 확인

오브젝트의 위치 및 크기를 입력 받고, 해당 오브젝트의 범위가 벽에 해당하는 좌표와 겹치는 경우
=> 이전 위치로 복귀하도록 설계하여, 벽 범위로 침범하는 것을 방지

```
int prev_x = human.x;
int prev_y = human.y;
int step = 1;
```

```
if(Check_Wall_Collision(human.x, human.y, human.w, human.h))
{
    human.x = prev_x; // 충돌 시 X축 복원
}
```

핵심 기술

파리와 개구리 사이의 거리 확인

개구리가 파리를 해치우기 위해 근처로 다가가는 경우, 인간을 이를 막고자 개구리 방향으로 이동한다.

Manhattan Distance
$\|p1-q1\|+\|p2-q2\|$

```
int dx_fly = (fly.x > frog.x) ? (fly.x - frog.x) : (frog.x - fly.x);
int dy_fly = (fly.y > frog.y) ? (fly.y - frog.y) : (frog.y - fly.y);

// fly 주변 100px 이내에 frog가 있으면 추적 시작
if(dx_fly < 100 && dy_fly < 100)
```

핵심 기술

다음 이동 방향 확인

개구리가 파리를 해치우기 위해 근처로 다가가는 경우, 인간을 이를 막고자 개구리 방향으로 이동한다.

```
if(dx_fly < 100 && dy_fly < 100)
// X축 이동
if(human.x < frog.x) { human.x += step; }
else if(human.x > frog.x) { human.x -= step; }
// Y축 이동
if(human.y < frog.y) { human.y += step; }
else if(human.y > frog.y) { human.y -= step; }
```

핵심 기술

다음 이동 방향 확인

개구리가 파리를 해치우기 위해 근처로 다가가는 경우, 인간을 이를 막고자 개구리 방향으로 이동한다.

```
// X축 이동
if(human.x < frog.x) { human.x += step; }
else if(human.x > frog.x) { human.x -= step; }

// X축 충돌 검사
if(Check_Wall_Collision(human.x, human.y, human.w, human.h))
{
    human.x = prev_x; // 충돌 시 X축 복원
}

// Y축 이동
if(human.y < frog.y) { human.y += step; }
else if(human.y > frog.y) { human.y -= step; }

// Y축 충돌 검사
if(Check_Wall_Collision(human.x, human.y, human.w, human.h))
{
    human.y = prev_y; // 충돌 시 Y축 복원
}
```

핵심 기술

전체 Human_Move 함수

```
static int Human_Move(void)
{
    if(human.dir != 0 || human_stopped) return 0;

    int dx_fly = (fly.x > frog.x) ? (fly.x - frog.x) : (frog.x - fly.x);
    int dy_fly = (fly.y > frog.y) ? (fly.y - frog.y) : (frog.y - fly.y);

    // fly 주변 100px 이내에 frog가 있으면 추적 시작
    if(dx_fly < 100 && dy_fly < 100)
    {
        int prev_x = human.x;
        int prev_y = human.y;
        int step = 1;

        // 이전 위치 지우기
        Lcd_Draw_Box(human.x, human.y, human.w, human.h, BACK_COLOR);

        // X축 이동
        if(human.x < frog.x) { human.x += step; }
        else if(human.x > frog.x) { human.x -= step; }

        // X축 충돌 검사
        if(Check_Wall_Collision(human.x, human.y, human.w, human.h))
        {
            human.x = prev_x; // 충돌 시 X축 복원
        }

        // Y축 이동
        if(human.y < frog.y) { human.y += step; }
        else if(human.y > frog.y) { human.y -= step; }

        // Y축 충돌 검사
        if(Check_Wall_Collision(human.x, human.y, human.w, human.h))
        {
            human.y = prev_y; // 충돌 시 Y축 복원
        }
        // 새 위치 그리기
        Lcd_Draw_Box(human.x, human.y, human.w, human.h, human.ci);
    }
    return 0;
}
```

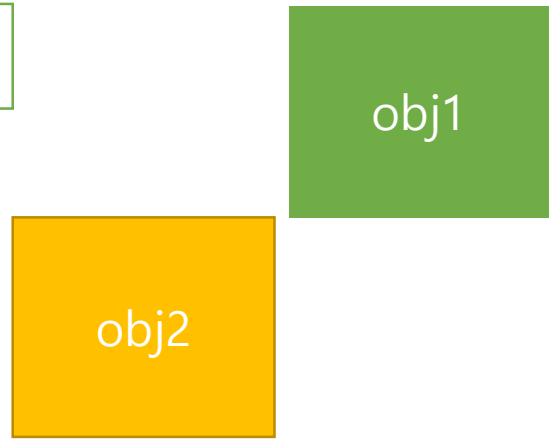
핵심 코드

선정이유:
거의 모든 순간에 사용된 함수이므로

```
Check_Collision(FROGs * obj1, QUERY_DRAW * obj2)
```

사용처: 장애물과의 충돌, 아이템 획득

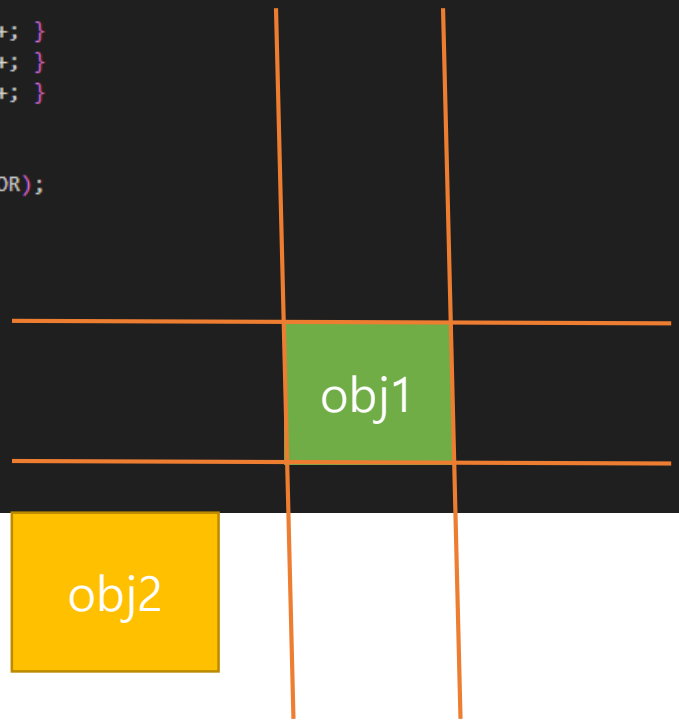
구현 방법



obj1의 범위가 obj2의 범위 밖에 있는 경우, 동작 변화x.
그 외의 경우, obj2의 종류에 따라 의도한 동작 실행

핵심 코드

```
static int Check_Collision(FROGs * obj1, QUERY_DRAW * obj2)
{// obj1:개구리, obj2:충돌대상
    if(obj2->dir != -1)
    {
        if((obj1->x>obj2->x+obj2->w)||(obj1->x+obj1->w<obj2->x)||(obj1->y+obj1->h<obj2->y)||(obj1->y>obj2->y+obj2->h)) { }// 충돌 x
        else // 충돌
        {
            if((obj2 == &snake[0]) || (obj2 == &snake[1]) || (obj2 == &human))
            {//GAMEOVER
                return 1;
            }
            else
            {
                if(obj2 == &poison[0]) { frog.stop = 30; eat_poison++; }
                else if(obj2 == &poison[1]) { frog.stop = 30; eat_poison++; }
                else if(obj2 == &poison[2]) { frog.stop = 30; eat_poison++; }
                else if(obj2 == &poison[3]) { frog.stop = 30; eat_poison++; }
                else { fly_CNT++; }//파리와 부딪히는 경우
                obj2->dir=-1;
                Lcd_Draw_Box(obj2->x, obj2->y, obj2->w, obj2->h, BACK_COLOR);
                Lcd_Draw_Box(frog.x, frog.y, frog.w, frog.h, frog.ci);
            }
        }
    }
    else//obj2가 이미 죽은 경우
    {
        return 0;
    }
    return 0;
}
```



결론

<게임 계획서(게임명: Makjang froggggggy)>

제출인: 서윤철

게임 컨셉:

게임은 파리 무리에게 된통 얻어맞고 온 아들 개구리를 위해 엄마 개구리가 복수하러 이동하던 중 들어간 미로 속에서 발생한 여정.

(예상)게임등장인물: 엄마 개구리, 뱀(천적), 인간(최종보스)

게임 규칙:

1. 엄마 개구리는 파리에게 복수하기 위해 최선을 다 해야한다.
2. 따라오는 뱀에게 잡히면 개구리에게 치명적이다.
3. 개구리에게 뱀보다 위협적인 존재는 인간이다.
4. 엄마 개구리가 모든 미션을 완수하면, 아들 개구리를 만나며 복수는 종료된다.

아이템

엄마 개구리의 여정 중, 다양한 아이템이 등장할 수 있다.
Ex) 파리: 총 6마리의 파리에게 복수를 가해야 한다.
Ex) 그물망: 뱀을 향해 맞추게 되면, 뱀은 2초 정지한다. 인간을 향해 맞추게 되면 1초 일시정지 시킬 수 있다.
Ex) 개구리밥: 엄마 개구리가 최종적으로 획득해야 복수를 완료할 수 있다.

장애물

엄마 개구리의 여정 중, 다양한 장애물이 등장할 수 있다.
뱀: 정해진 구역을 순찰하며 엄마 개구리의 움직임을 방해한다.
인간: 엄마 개구리의 최종보스로, 정해진 반경 내로 엄마개구리를 추적한다.

Ex) 아이템 중 복불복으로 쥐약이 등장한다.
⇨ 쥐약으로 위한 복통으로 엄마 개구리는 3초 동안 정지한다.

게임구현방식: STM32F103, C

조수: chat gpt, perplexity ai

계획한 게임 완성

결론

<게임 계획서(게임명: Makjang froggggggy)>

제출인: 서윤철

게임 컨셉:

게임은 파리 무리에게 된통 얻어맞고 온 아들 개구리를 위해 엄마 개구리가 복수하러 이동하던 중 들어간 미로 속에서 발생한 여정.

(예상)게임등장인물: 엄마 개구리, 뱀(천적), 인간(최종보스)

게임 규칙:

1. 엄마 개구리는 파리에게 복수하기 위해 최선을 다 해야한다.
2. 따라오는 뱀에게 잡히면 개구리에게 치명적이다.
3. 개구리에게 뱀보다 위협적인 존재는 인간이다.
4. 엄마 개구리가 모든 미션을 완수하면, 아들 개구리를 만나며 복수는 종료된다.

아이템

엄마 개구리의 여정 중, 다양한 아이템이 등장할 수 있다.
Ex) 파리: 총 6마리의 파리에게 복수를 가해야 한다.
Ex) 그물망: 뱀을 향해 맞추게 되면, 뱀은 2초 정지한다. 인간을 향해 맞추게 되면 1초 일시정
시킬 수 있다.
Ex) 개구리밥: 엄마 개구리가 최종적으로 획득해야 복수를 완료할 수 있다.

장애물

엄마 개구리의 여정 중, 다양한 장애물이 등장할 수 있다.
뱀: 정해진 구역을 순찰하며 엄마 개구리의 움직임을 방해한다.
인간: 엄마 개구리의 최종보스로, 정해진 반경 내로 엄마개구리를 추적한다.

Ex) 아이템 중 복불복으로 쥐약이 등장한다.
⇨ 쥐약으로 위한 복통으로 엄마 개구리는 3초 동안 정지한다.

게임구현방식: STM32F103, C

조수: chat gpt, perplexity ai

완벽한 추격 구현?

개선 방법

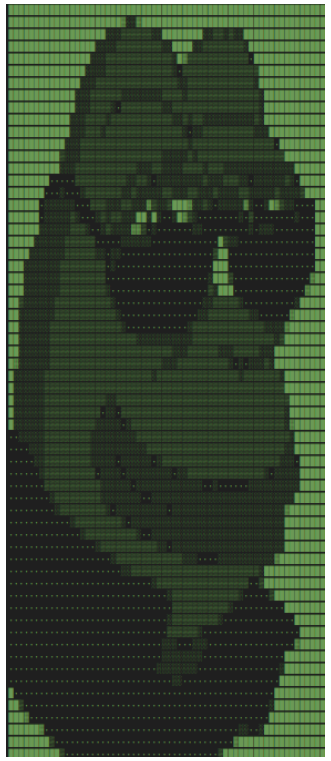
A* 알고리즘

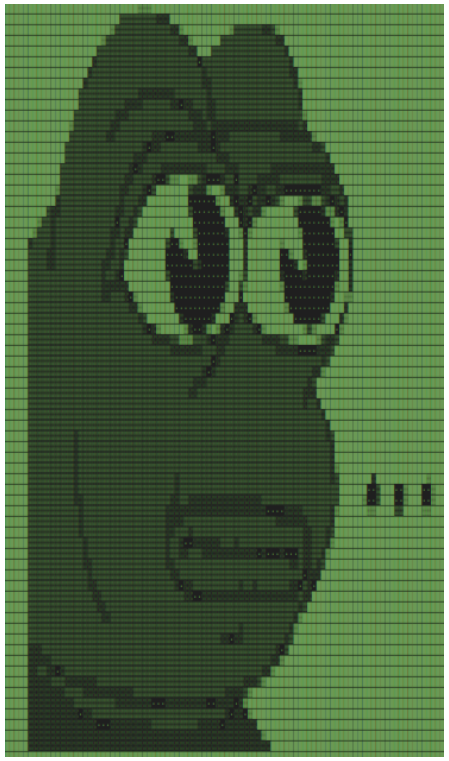
구현 실패..

개발 후기

게임은 그래픽이 중요하다.

개발 후기





```
lcd.h:29:3: error: conflicting types for 'tImage'
 } tImage;
   ^
In file included from device_driver.h:5:0,
                 from lcd.c:1:
lcd.h:29:3: note: previous declaration of 'tImage' was here
 } tImage;
   ^
In file included from lcd.c:2:0:
lcd.h:40:13: error: conflicting types for 'draw_image'
 extern void draw_image(int xs, int ys, tImage * d);
             ^
In file included from device_driver.h:5:0,
                 from lcd.c:1:
lcd.h:40:13: note: previous declaration of 'draw_image' was here
 extern void draw_image(int xs, int ys, tImage * d);
             ^
make: *** [clock.o] Error 1
```

게임이 클리어되거나 오버되었을 경우, 지정한 메시지를 출력하면서 각 경우에 맞게 두 그림을 출력하고자 했는데, 오류 발생...

똑같은 코드를 다른 파일에 복사해서 컴파일해보니 컴파일 성공..

감사합니다.