Square Scared

컴퓨터 정보 공학과

12131579 이진아

12141579 윤찬미

게임 선택 과정

- 미로 찾기
- 지렁이 잡기 게임
- 땅굴 파기
- 떨어지는 장애물 피하기
- 스포츠 게임

위에서 언급한 여러 게임 종류 후보들이 나왔다. 수업시간에 배운 내용에 대한 활용도, 참신성, 난이도, 재미 4 가지 요소를 중점으로 게임을 선택하였다. 미로 찾기와 지렁이 잡기 게임은 이미 인터넷상에서 만든 예시가 많아 참신성 면에서 떨어진다고 생각하여 제외시켰다. 땅굴 파기와 스포츠 게임은 참신성은 높아 보였으나 수업시간에 배운 내용들로만 하기에는 어려움이 있어 난이도가 높아 보여 제외시켰다. 모든 측면에서 가장 알맞다는 회의 결과가 나와 '떨어지는 장애물 피하기'로 게임을 선택하였다. 단순히 위에서 떨어지는 장애물을 피하는 것만 하면 재미가 떨어질 것 같아 장애물과 다른 물체를 넣어 그 물체는 먹을 수 있도록 게임 요소를 추가하기로 결정하였다.

프로젝트 개요

- 타이머(CTC mode), 외부 인터럽트 (0 번, 1 번) , 조이스틱를 사용하여 위에서 떨어지는 장애물을 피하는 게임이다. 만약 캐릭터와 장애물이 부딪히면 생명력을 감소 시키고 하트를 먹으면 생명이 증가한다. 생명을 다 쓰게 되면 게임이 종료된다.

요소 설명

캐릭터

Data_ADC4 = Read_Adc_Data(4) / 8; // 아날로그 0 번 포트 읽기

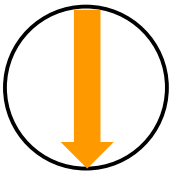
- GLCD_Rectangle(63,Data_ADC4,55,Data_ADC4+8);

8*8 정 사각형을 조이스틱의 값을 받아와서 움직이게 한다. ADC 4 를 이용하여 양 옆으로만 이동이 가능하다.

장애물, 하트 : x,y 값과 속도를 가진다.

```
fall[f_index].x = 15;
fall[f_index].y = (rand()%128);
fall[f_index].sp = (rand()%5)+2;
f_index++;
if(f_index >= 20)
f_index = 0;

if(sec % 3 == 0) {
        heart[h_index].x = 15;
        heart[h_index].y = (rand()%128);
        heart[h_index].sp = (rand()%5)+1;
        h_index++;
        if(h_index >= 10)
        h_index = 0;
}
```



반지름이 2 인 원과 하트모양의 픽셀을 각각 y 값과 속도를 랜덤으로 받아 바닥으로 떨어지게 한다. 하트는 원이 약 3 개가 떨어지면 1 개를 떨어 질 수 있게 한다.

게임 규칙

1. 게임 이름이 나온 후 게임이 시작된다.
2. 5 개의 생명력을 가지고 시작한다.
3. 생명력의 갯수를 LED 에 출력 한다.
4. 장애물과 하트가 위에서 부터 각자의 속도를 가지고 떨어진다.
5. 조이스틱을 이용하여 캐릭터를 움직여 장애물과 닿지 않도록 한다.
6. 장애물과 캐릭터가 닿으면 생명력이 1 감소한다.
7. 0 번 Pin 을 눌러 하트를 먹으면 생명력이 1 증가한다.
8. 생명은 최대 5 개 가능하다.

9. 하트가 존재하지 않는 곳에 pin 을 누르면 생명력이 1 감소한다.

10. 모든 생명력이 다 닳으면 게임이 종료된다.

11. 게임이 종료되면 스코어를 출력하고 재시작을 하기 위한 화면을 띄운다.

12. 1 번 Pin 을 누르면 게임을 다시 시작 한다.

작동 설명

INT0

- 0 번 인터럽트 PIN 이 눌러졌을때 인터럽트 서비스 루틴을 이용하여 캐릭터와 하트가 닿는 범위를 계산하여 그 범위 안에 들어왔으면 life 를 증가 시키고, 들어오지 않았을 경우 life 를 감소 시킨다.

만약 life 가 0 이 되면 Game 이 끝났다는 flag 를 true 로 만들고 게임 종료를 소리를 통해 알려준다.

범위 : if(life!=5&&((63 >= heart[i].x) &&(55 <= heart[i].x)&&(Data_ADC4+8 >= heart[i].y) &&(Data_ADC4-8 <= heart[i].y)))

게임 종료 : if(life == 0){ S_S6(); isGameOver = 1;}

INT1

- 1 번 인터럽트 PIN 이 눌러졌을때 인터럽트 서비스 루틴을 이용하여 게임을 재시작 한다.

모든 장애물과 하트의 위치를 초기화 시키고, 점수, life 등을 초기 값으로 만든다. 게임 시작을

소리를 통해 알려준다.

void IsTouch(void)

- 장애물과 캐릭터가 닿았는지 판별하는 함수이다. 캐릭터와 장애물이 닿는 범위를 계산하여

그 범위 안에 들어오면 life 를 감소시킨다.

만약 life 가 0 이 되면 Game 이 끝났다는 flag 를 true 로 만들고 게임 종료를 소리를 통해

알려준다.

범위 :if((63 >= fall[i].x) &&(55 <= fall[i].x)&&(Data_ADC4+8 >= fall[i].y) &&(Data_ADC4-8 <=

fall[i].y))

게임 종료 :if(life == 0) { PORTB = 0xff; isGameOver = 1; S_S6(); }

void GLCD_Heart(int x1, int y1)

- 하트의 픽셀 값을 찍어주는 함수이다. 픽셀이 찍히는 위치를 GLCD_Dot(x,y)를 이용하여

찍는다.

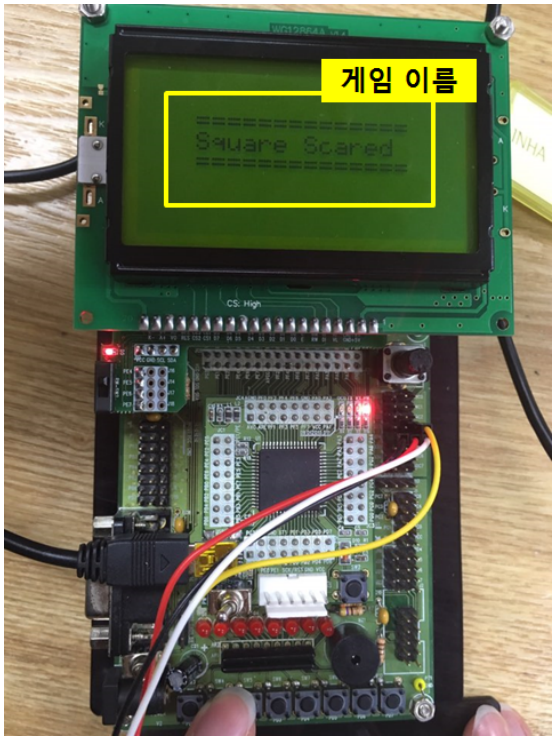
void Show_Screen(void)

- 게임에 필요한 모든 요소를 출력한다. 점수, 생명력, 캐릭터, 장애물, 하트 를 모두 LCD 에 출력하고, 남은 생명력의 갯 수만큼 LED 를 밝힌다.

void SquareScared(void)

- 게임 처음 시작 시 게임 이름을 출력한다.

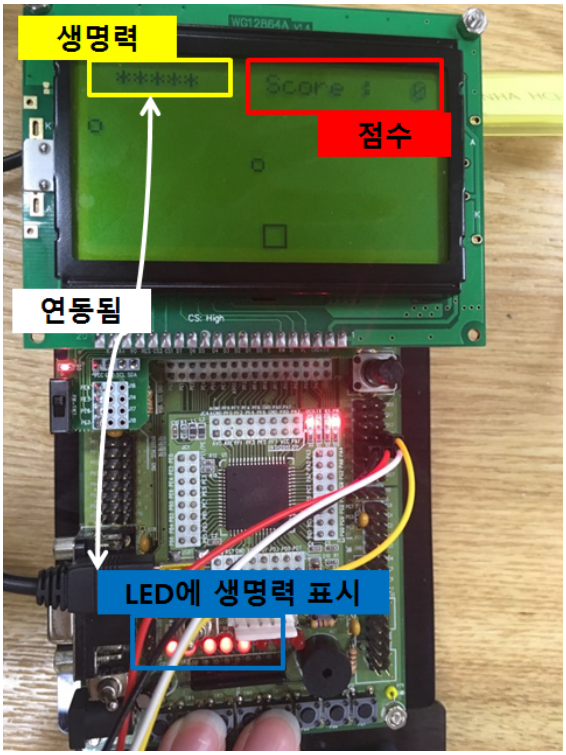
게임 화면



1)게임 시작 화면

게임이 시작되었음을 알리는 화면이다.

2) 게임화면



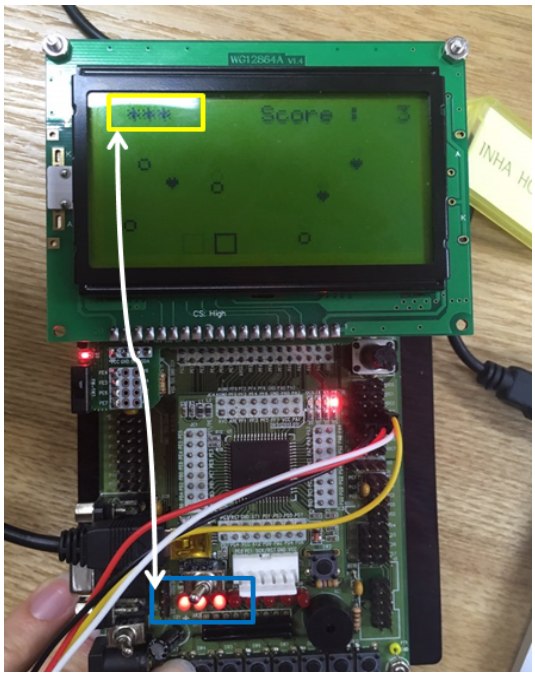
2-1. 생명력 5 개를 가지고 시작한다.

LCD 왼쪽 상단 : 생명력 표시

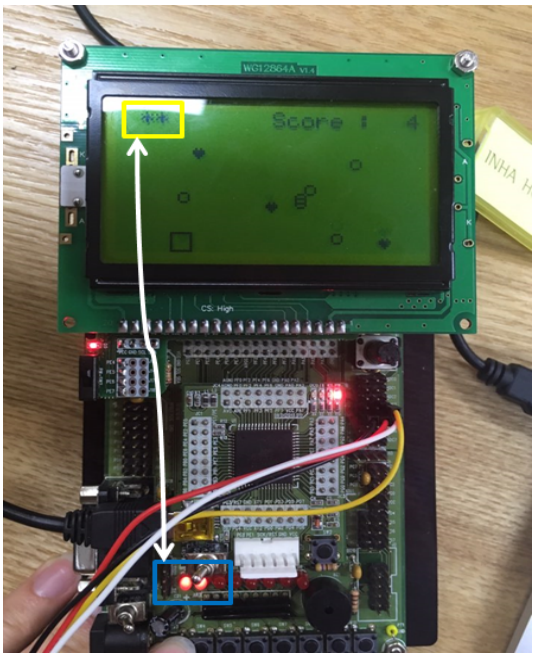
LCD 오른쪽 상단 : 점수 표시

LED : 생명력 만큼 LED ON

2-2) 생명력 감소 했을 시 LCD 와 LED

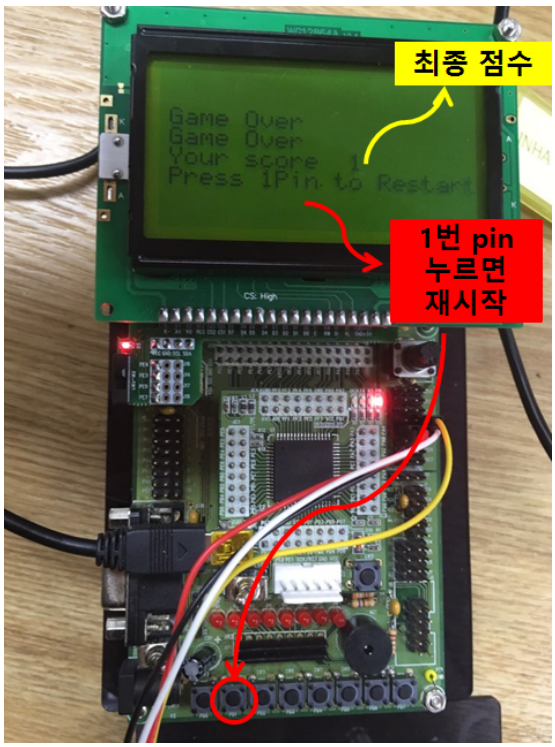


<생명력 3 인 경우>



<생명력 2 인 경우>

3)게임 종료



isGameOver 이 True 가 되었을 때, GameOver 이 되었다는 string 과 score 을 출력한 후, 재시작을 하고싶으면 1 번 Pin 을 눌러라고 알린다.

게임 실행 URL

https://youtu.be/9tPpInp4bJ4

역할 분배

- 소스코드는 장애물과 조이스틱 부분으로 나눠서 작성하였다. 서로 해결하지 못한 부분을 공유함으로서 오류를 고쳤으며, LCD 출력과 인터럽트 등은 수업에서 사용한 자료를 바탕으로

하여 소스를 함께 수정했다. 주석과 보고서는 Google Drive 를 이용하여 공유하여 수정 및 작성하였다.

프로젝트를 수행하며 배우고 느낀점

- 팀 프로젝트를 하는 과정에서 서로 의견이 맞지 않는 일이 생겼을 때, 맞춰가는 과정을 배웠으며, 각자 코딩을 하며 사용했던 변수명과 함수명이 달라 합치는데 어려움이 있었다. 이로 인해 GitHub 를 사용하여 소스 코드를 공유하며 관리하여 어려움을 해결하였다.
각자가 해결하지 못 했던 문제들에 대해 서로의 코드를 공유 함으로서 해결하지 못 했던 오류를 서로가 도우며 해결하였다.
- 수업시간에 배운 것을 활용하는 방법을 익히지 못했는데 프로젝트를 통해 활용함으로서 Atmega 에 대한 이해도를 높일 수 있었다.