

# STL 과제 2

## 보고서

[이름]: 원동욱

[학번]: 2014180026

[과목]: STL 월 23/화 56

## 목차

### 프로그램에 대한 설명

#

1. 메뉴
2. 게임
3. 리플레이

### 과제 해결 방법

#

1. 3D 과제로 제작
2. 스네이크로 변경

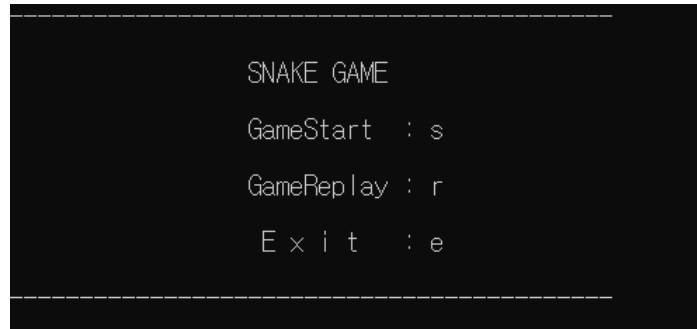
### 과제를 마치고 느낀점

#

1. 데이터의 정확한 제어에 대한 중요성
2. 파일을 입출력 할 때의 규칙에 대한 중요성
3. 벡터의 사기성

## 프로그램에 대한 설명

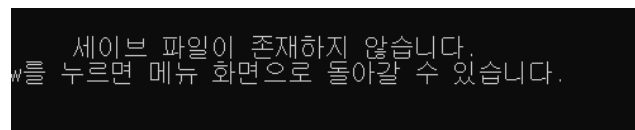
### 1. 메뉴



게임을 처음 실행 했을 시 나오는 화면이다. 다음과 같은 기능이 있다.

S 를 누를 시 게임을 시작한다..

R 을 누를 시 리플레이를 시작한다.



리플레이 파일이 없을 경우 파일 없다는 화면을 띄운다.

W 를 누를 시 메뉴 화면으로 돌아간다.

E 를 누르면 게임을 종료한다.

### 2. 게임



게임의 룰은 다음과 같다.

스네이크는 머리 부분이 벽이나 자신의 몸통과 부딪힐 경우 메뉴 화면으로 돌아간다.

별을 먹을 경우 자신의 몸통의 길이가 증가한다.

스네이크는 일정 시간이 지날 때마다 정해진 방향으로 움직인다.

플레이어는 wasd 키를 이용 해 스네이크의 방향을 상/좌/하/우로 설정 할 수 있다.

Q 를 누를 시 녹화를 시작하게 되며 e 를 누르거나 메뉴 화면으로 돌아가면 녹화는 종료 된다.

녹화는 '1 회'분만이 저장이 되며 새로 녹화 할 시 이전의 데이터가 사라지게 된

### 3. 리플레이



리플레이에서는 저장된 게임이 자동으로 진행되어진다.

W 를 누르면 메뉴 화면으로 돌아간다.

A 키를 누르면 재생 속도가 감소하고 d 를 누르면 재생 속도가 증가한다.

S 키를 누르면 영상을 정지하거나 다시 재생 할 수 있다.

재생이 끝나면 화면이 멈춘다.

## 과제 해결 방법

### 1. 3D 프로그램으로 제작

처음 과제를 만들려고 할 때 3d 게임 프로그램을 가져와 객체의 월드 행렬의 위치를 일정 시간마다 기록하여 다시 재생 하려고 했으나 파일을 입력하는 도중 월드 행렬의 위치 값이 정확히 기록이 되지 않았고 이를 해결할 방법이 떠오르지 않아 중도에 포기하게 되었다.

### 2. 스네이크로 변경

GitHub 에서 마땅한 게임이 없어 snake 게임으로 제작하기로 결정을 내렸다. 처음에는 시간마다 맵 을 나타내는 배열을 모두 저장하고 이를 불러들여 시간마다 인덱스 값이 증가해 해당 시간의 인덱스를 찾아가는 방식으로 녹화를 재생하는 하려 했으나 동적 배열에서는 자료형으로 과거의 정적 배열을 넣을 수 없는 걸 깨 달았다. 그래서 동적 배열로 녹화를 하는 정보를 스네이크의 위치와 별이 생성되는 위치로 바꾸어 제작하게 되었다. 이때 스네이크의 변경되는 몸통의 길이가 문제점이 되었는데 스네이크의 몸통들의 동적 배열로 저장하고 파일을 읽고 쓸 때 몸통의 길이를 적고 읽는 방식으로 이 문제를 해결했다.

## 과제를 마치고 느낌 점

### 1. 데이터의 정확한 제어에 대한 중요성

3d 게임에 대해서 원했던 기능이 구현 되어서 정보를 정확히 알고 쓰는 줄 알았는데 정보를 쓴 것을 열어보니 위치 값이 변하지 않아 고민을 하다가 결국 해결을 못하는 사태가 일어나고 말았다. 앞으로 프로그래밍을 할 때 내가 어떤 정보를 어떻게 바뀌는지 대하여 정확히 알고 있어야 함을 느끼게 되었다.

## 2. 파일 입출력을 할 때 규칙의 중요성

snake 에서 snake 를 저장 할 때 어떻게 저장 할 지에 대해서 고민을 많이 했었다. 그때 네트워크 기초 시간에 배운 프로토콜이란 개념이 생각났고 이를 내 프로그램에 대하여 사용해 보게 되었다. 하면서 느낀 것인데 단순한 정보 교환의 경우 딱히 규칙이 필요가 없어도 되지만 해석에 따라 다른 의미로 읽어지는 정보 같은 경우에는 규칙을 정해서 의미가 왜곡되지 않도록 만들어야 한다는 것을 느꼈다.

## 3. 벡터의 사기성

과거에 녹화 기능을 만들라고 했으면 정적 배열로 녹화시간에 한계가 있거나 프로그램을 만들 때 과도한 메모리를 낭비 했을 것 같은데 벡터를 쓰니 그런 고민을 안 해도 되어서 좋았다. 역시 95%의 해결법인 것 같다.