

# PIPE GAME (파이프 게임)

2021 컴퓨터 공학 설계 및 실험 \_ 기말 프로젝트

학번 \_ 20181202 이름 \_ 김수미

# INDEX

## 1 \_ 프로젝트 개요

- (1) 프로젝트 목표
- (2) 프로젝트 동작 흐름
- (3) 창의적 구현

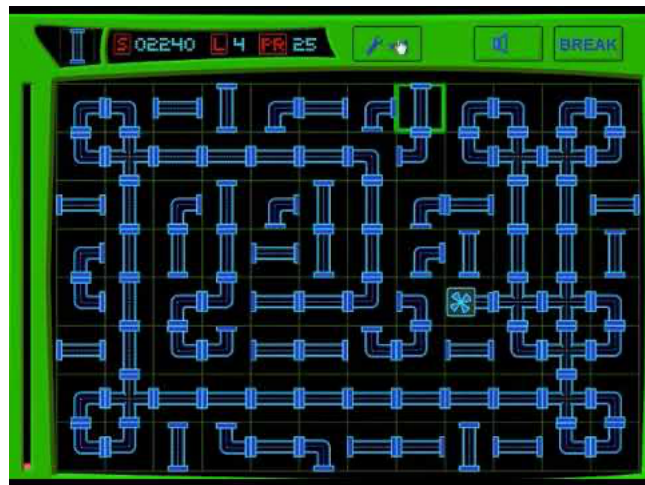
## 2 \_ 프로그램 내용

- (1) 자료구조 및 알고리즘
- (2) 핵심 변수
- (3) 핵심 함수

## 3 \_ 느낀 점 및 개선 사항

# 1. 프로젝트 개요 \_ 프로젝트 목표

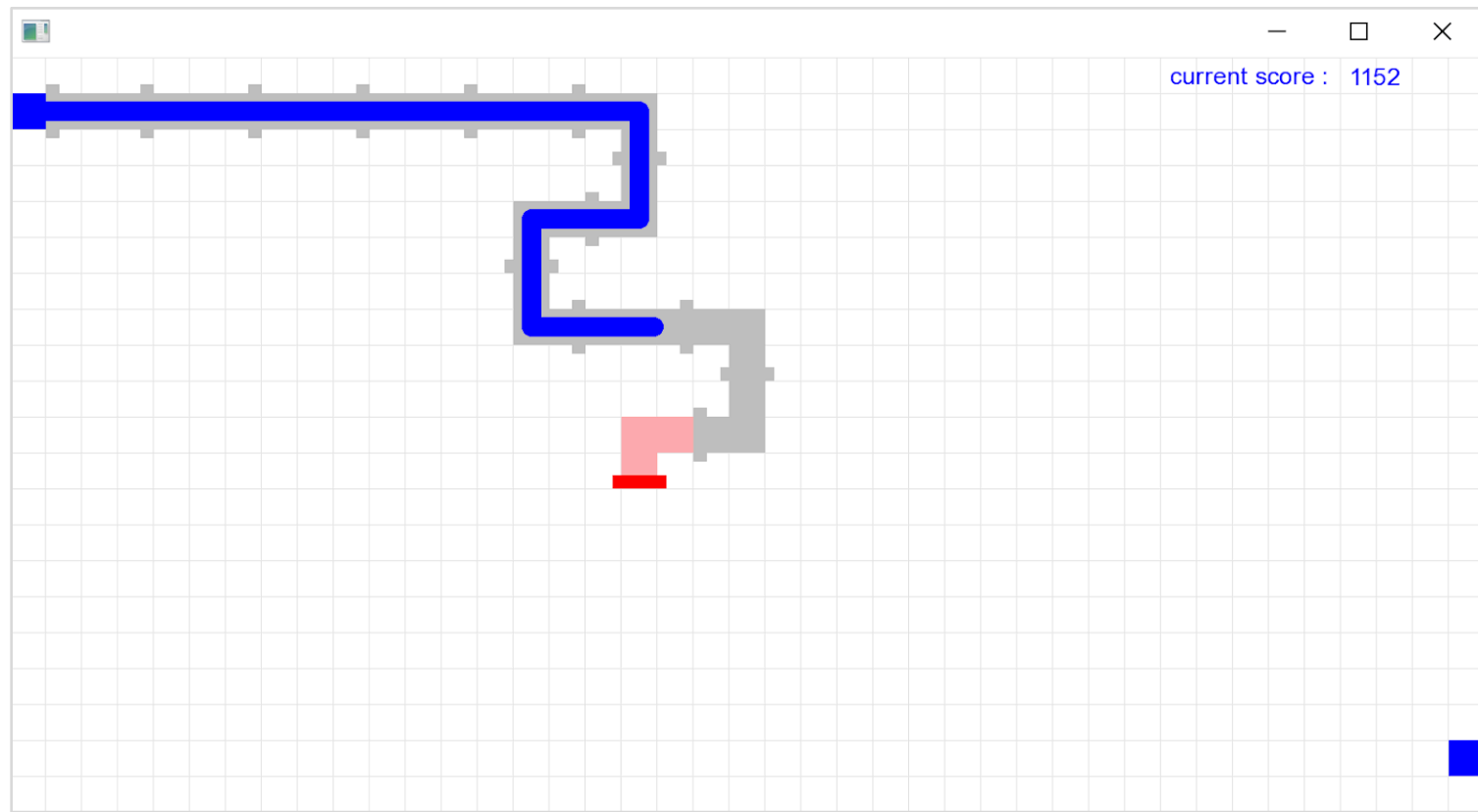
- 수업에서 배운 Openframework 사용 방법 및  
각종 자료구조, 알고리즘 구현
- 테트리스 + Waterfall 수업의 응용
- 파이프 게임 제작
- 일반적인 파이프 게임과는 조금 다르다
- 매 경로마다 새로운 파이프를 배치



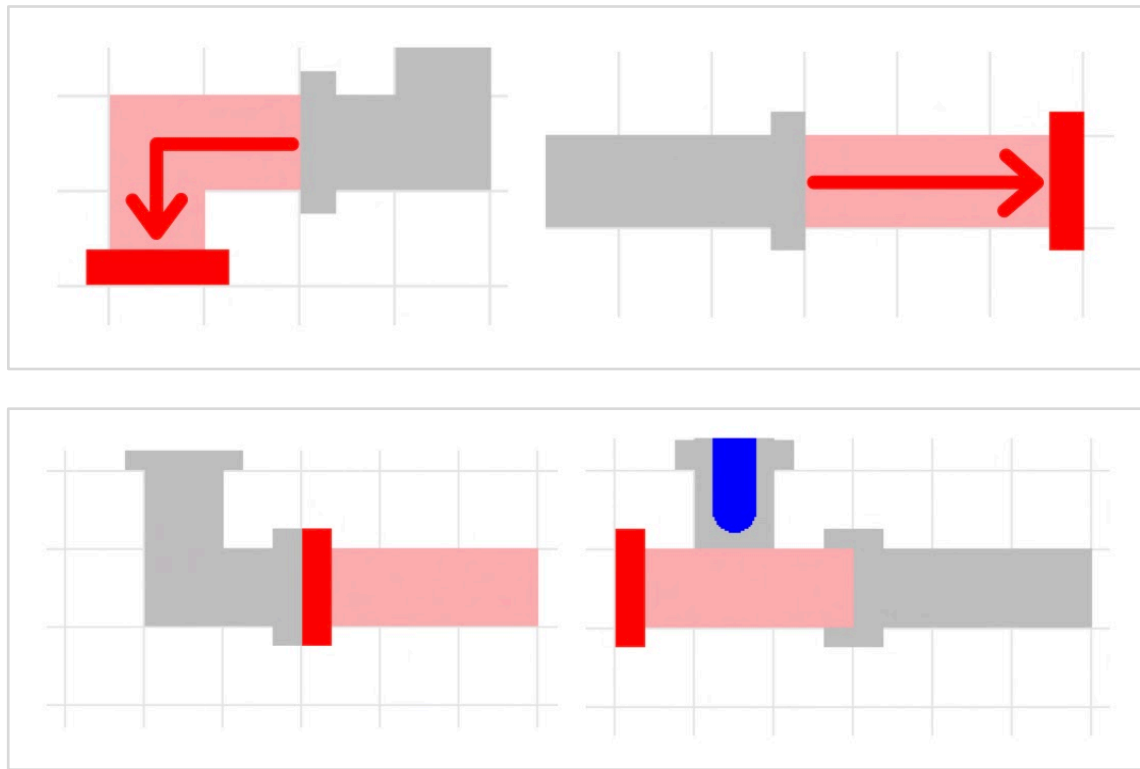
# 1. 프로젝트 개요 \_ 프로젝트 동작 흐름



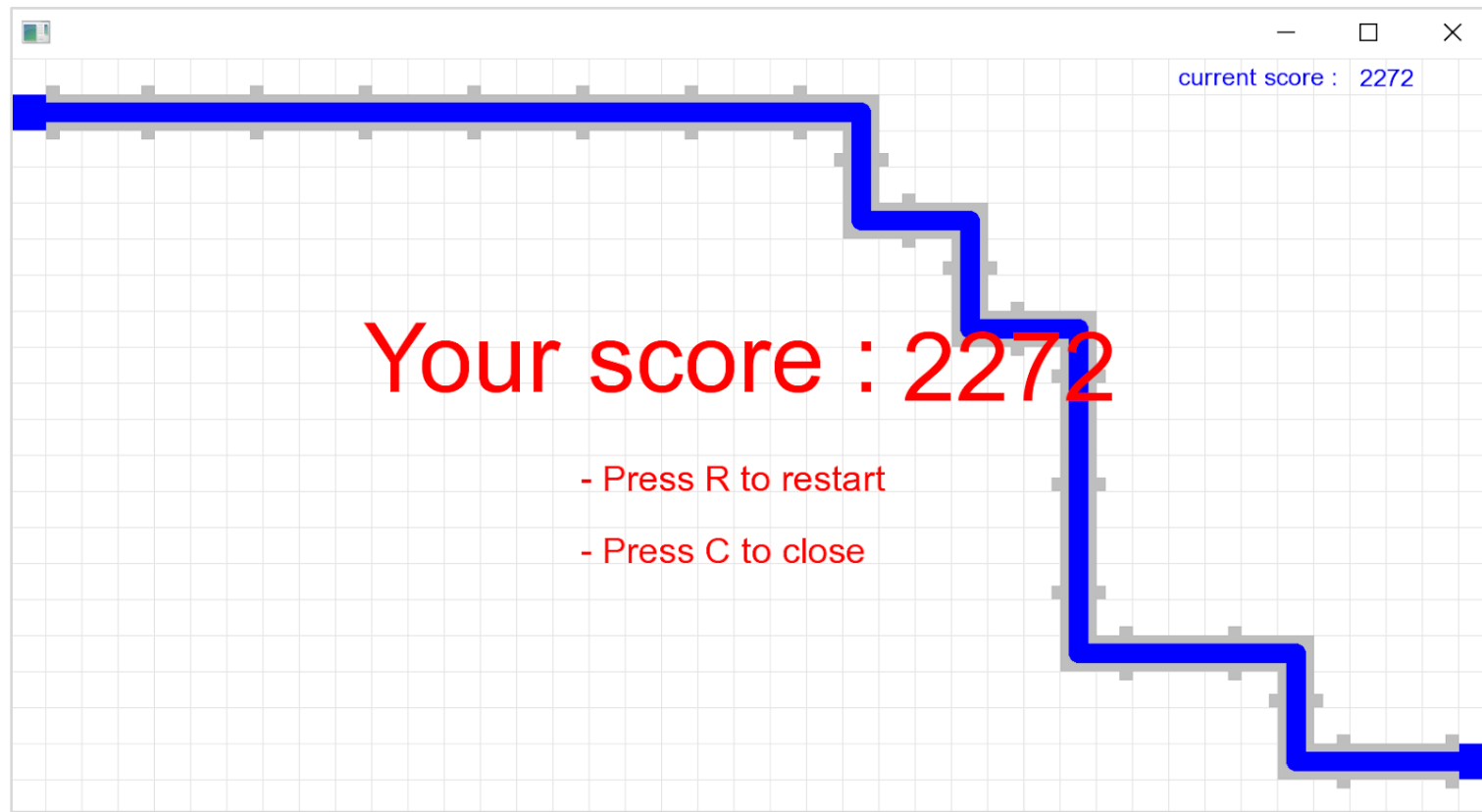
# 1. 프로젝트 개요 \_ 프로젝트 동작 흐름



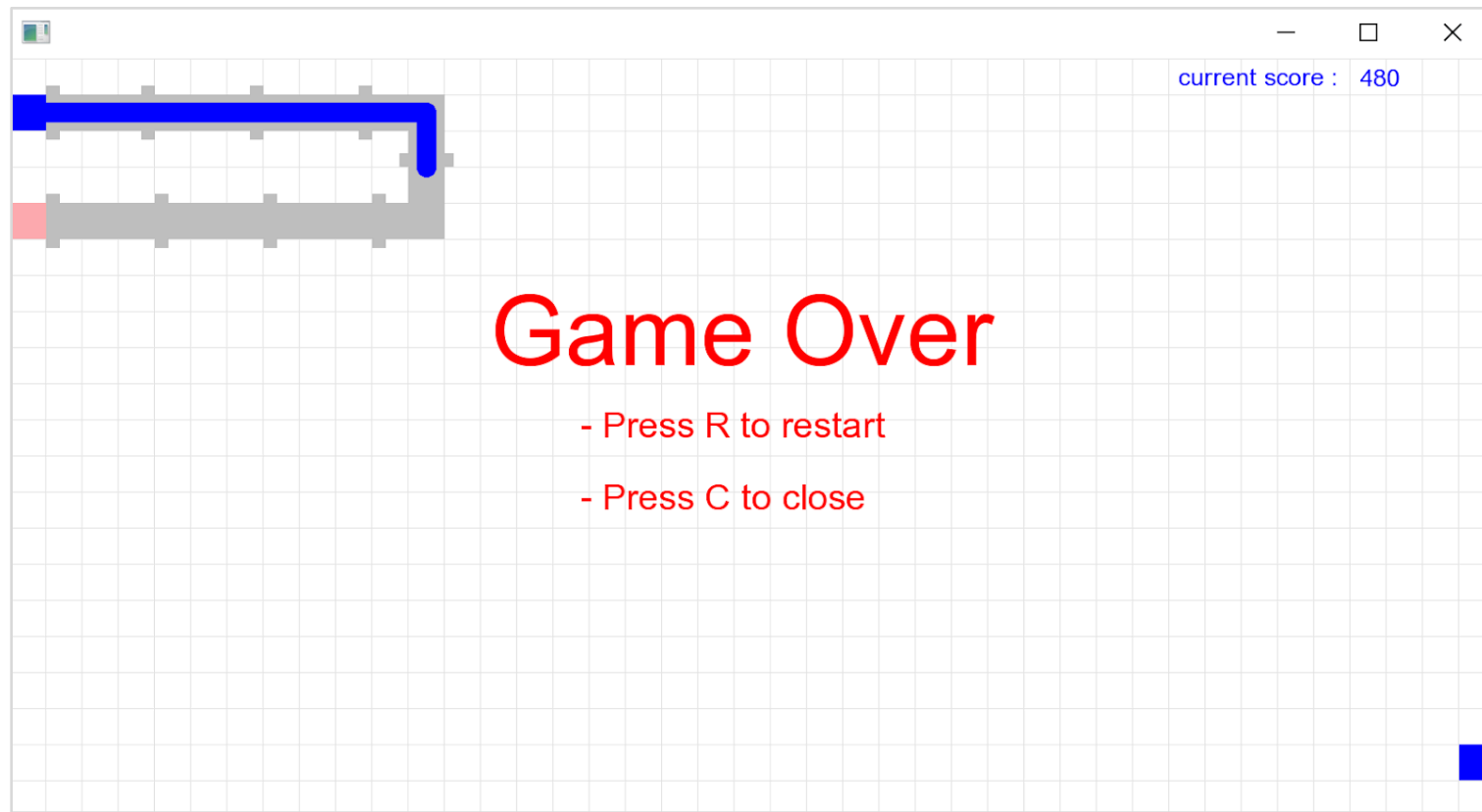
# 1. 프로젝트 개요 \_ 프로젝트 동작 흐름



# 1. 프로젝트 개요 \_ 프로젝트 동작 흐름

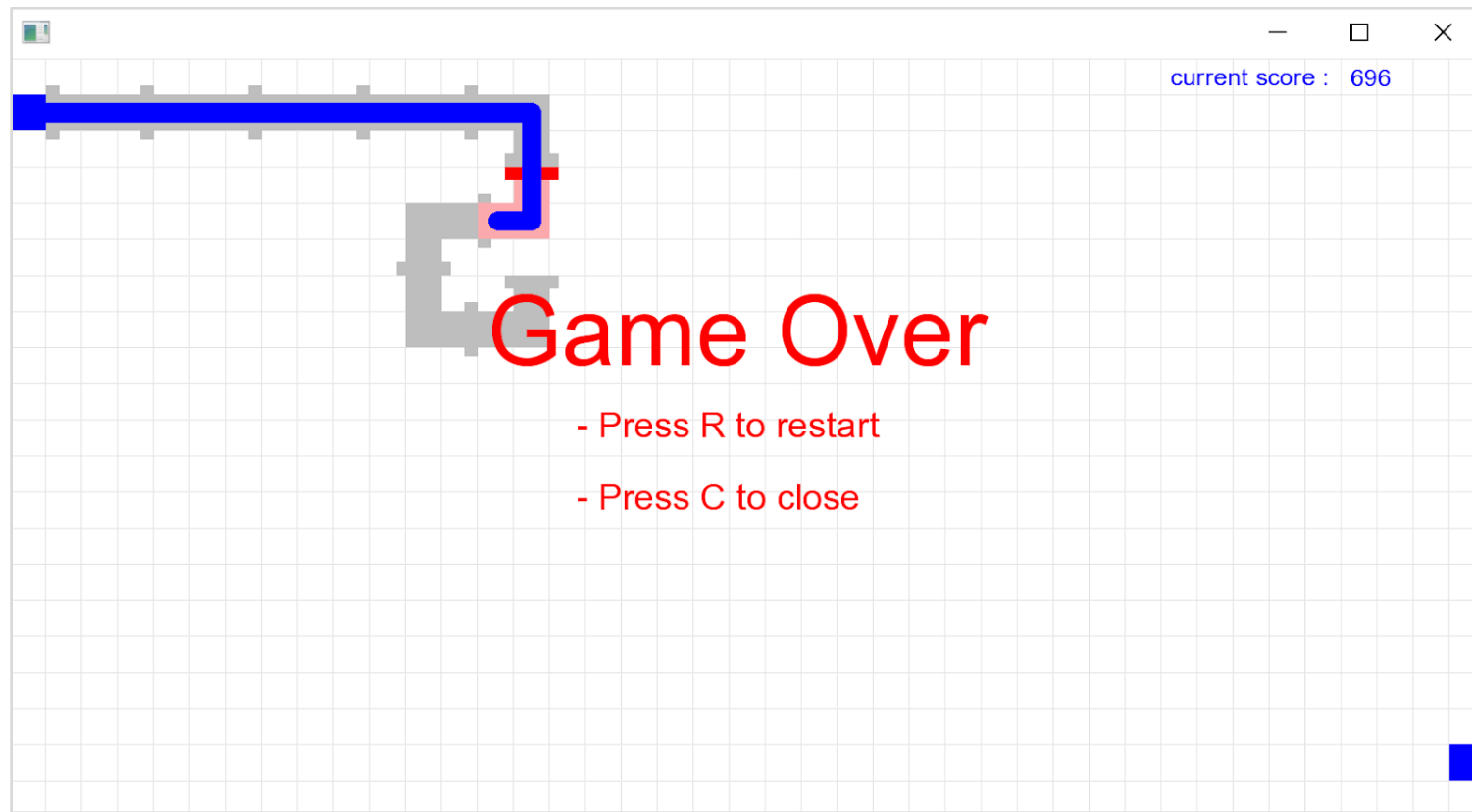


# 1. 프로젝트 개요 \_ 프로젝트 동작 흐름

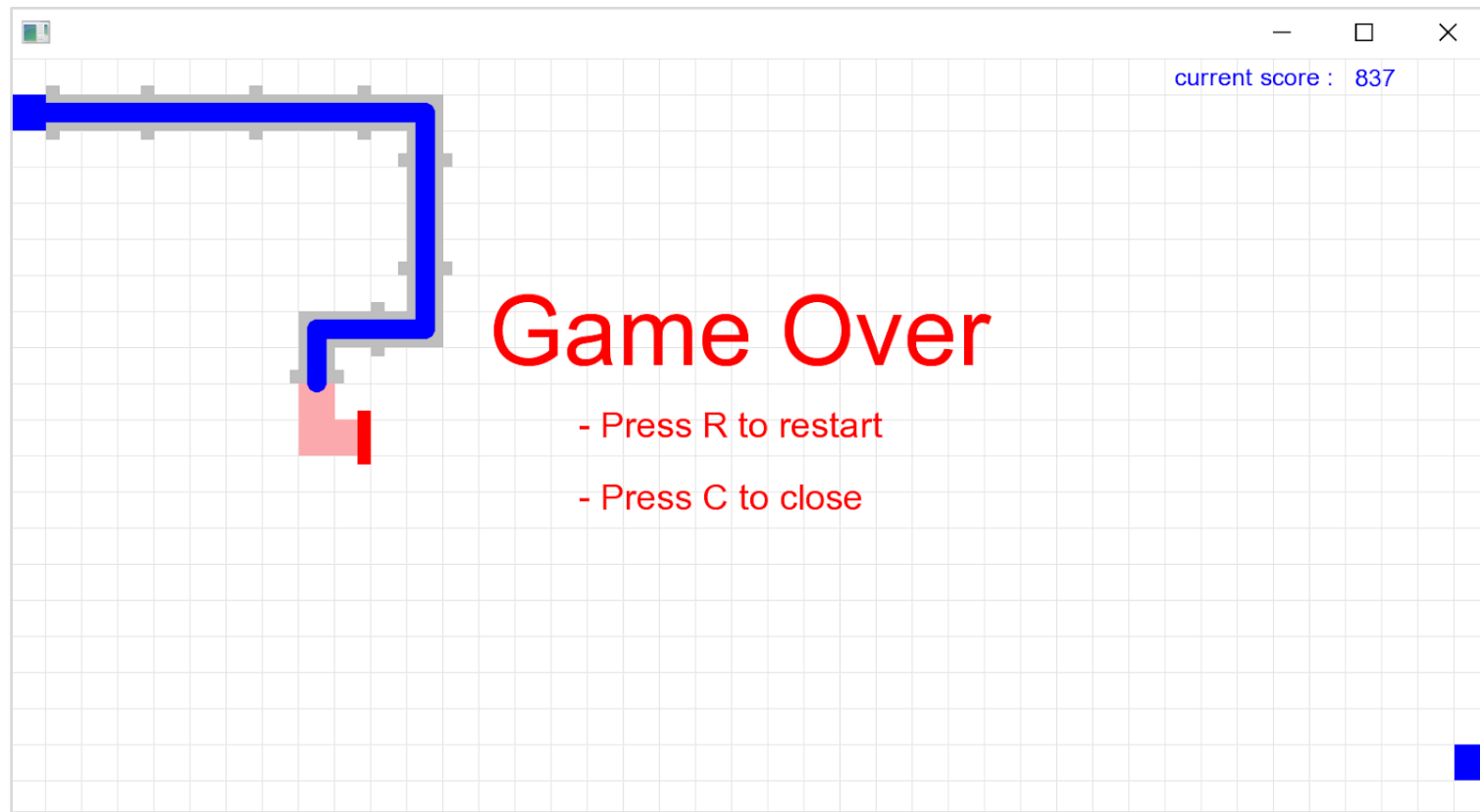




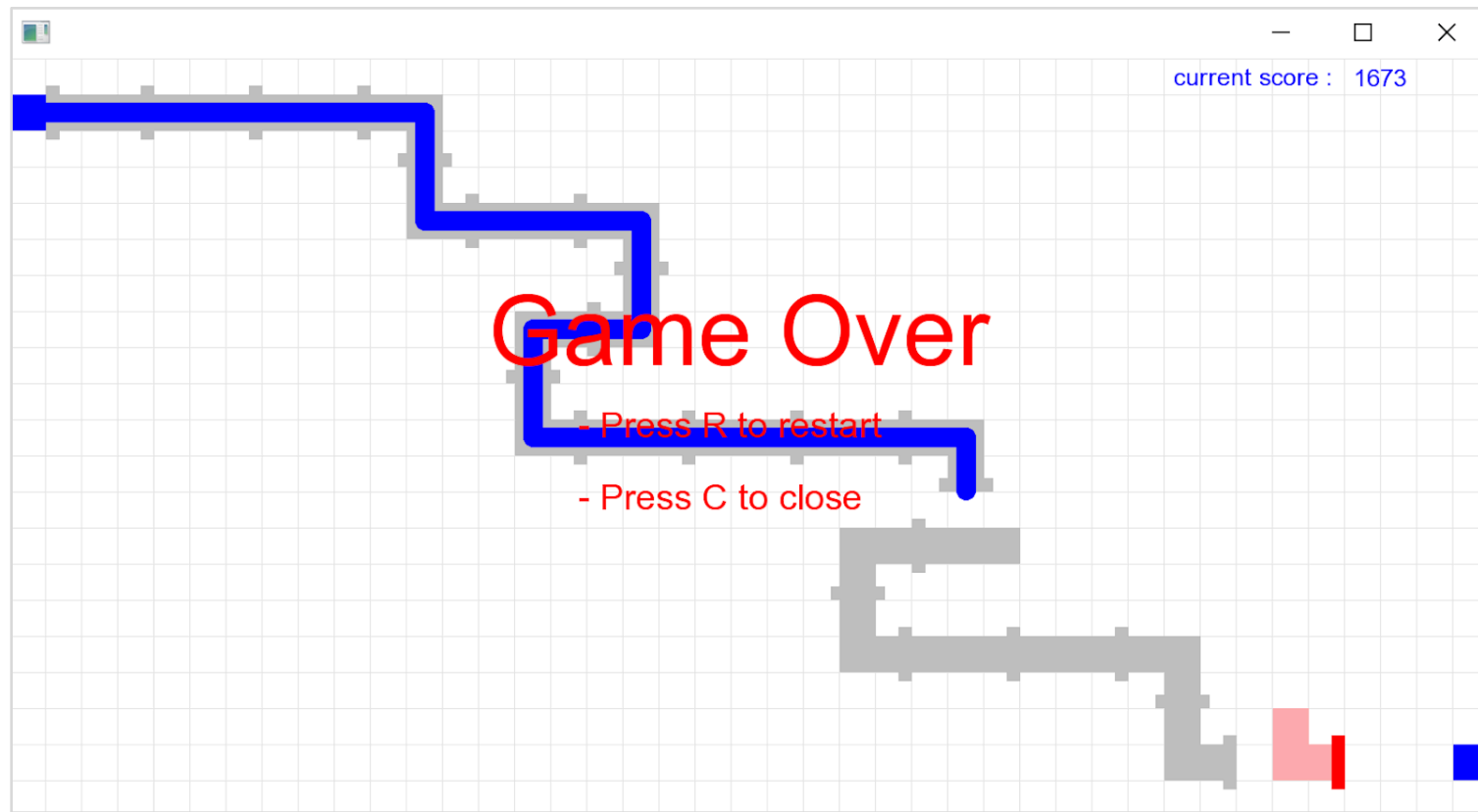
# 1. 프로젝트 개요 \_ 프로젝트 동작 흐름



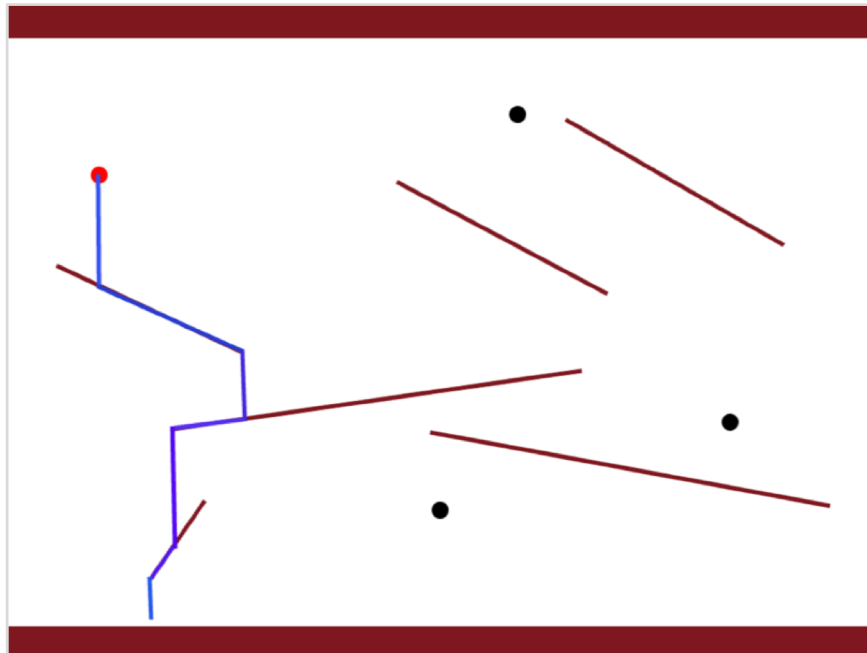
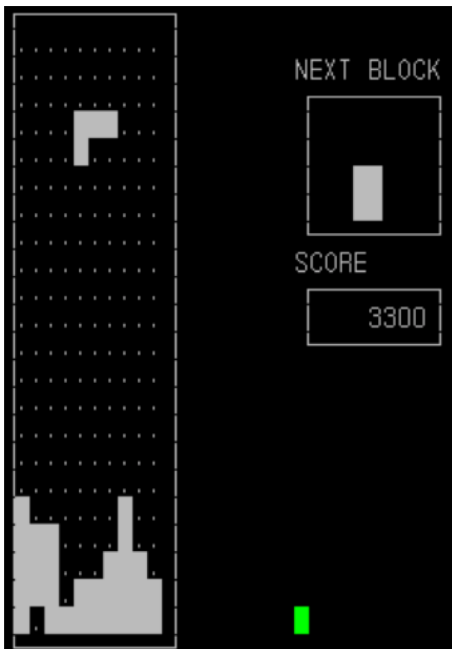
# 1. 프로젝트 개요 \_ 프로젝트 동작 흐름



# 1. 프로젝트 개요 \_ 프로젝트 동작 흐름



# 1. 프로젝트 개요 \_ 창의적 구현



## 2. 프로그램 내용 \_ 자료구조 및 알고리즘

### 1) 자료구조 : 연결 리스트

- 화면에 배치된 파이프 정보 저장
- 물줄기가 그려져야 하는 좌표 정보 저장
- 정적으로 선언된 배열보다 공간적으로 효율적

### 2) 알고리즘 : 파이프 위치 계산 방식

- 좌측 상단의 좌표
- 파이프가 그려지는 위치
- 물줄기가 지나가는 경로

## 2. 프로그램 내용 \_ 핵심 변수

```
typedef struct pipe {  
    int pipe_type;  
    int rotation;  
    float x; float y;  
    struct pipe *next;  
} pipe;  
struct pipe *currentP;  
struct pipe *startP;  
struct pipe *tempP1;  
struct pipe *tempP2;  
  
typedef struct water {  
    float x; float y;  
    struct water *next;  
} water;  
struct water *startW;  
struct water *currentW;  
struct water *tempW;  
struct water *endW;
```

## 2. 프로그램 내용 \_ 핵심 함수

- `void drawPipe[ ]` : 현재 배치하고자 하는 파이프를 현재 파이프 type 정보, rotation 정보, 좌측 상단 꼭짓점의 좌표 정보를 이용해 그린다.
- `void placePipe[ ]` : pipe 노드 연결리스트에 저장된 모든 파이프를 화면에 그린다.
- `void addPath[ ]` : 파이프를 새로 배치할 때 마다 배치된 파이프의 type 정보, rotation 정보, 좌측 상단 꼭짓점의 좌표 정보를 이용해 물줄기가 지나가야 하는 경로를 계산하고 이를 Water 노드 연결리스트에 저장한다.

### 3. 느낀 점 및 개선 사항

- 테트리스와 Waterfall 수업 내용에서 배운 것들을 활용해 재미있었다.
- 처음에는 과연 구현할 수 있을지 의문이었는데 성공해서 뿌듯하다.
- 파이프가 끊어졌거나 파이프를 미처 배치하지 못해  
게임오버가 되는 경우 : 그래픽을 추가하고 싶었는데 프로그램이 무거워 포기한  
것이 아쉽다.