2020년도 2학기 컴퓨터공학설계및실험I

11주차 결과 보고서

20170175 김태안

1. 실습 목적

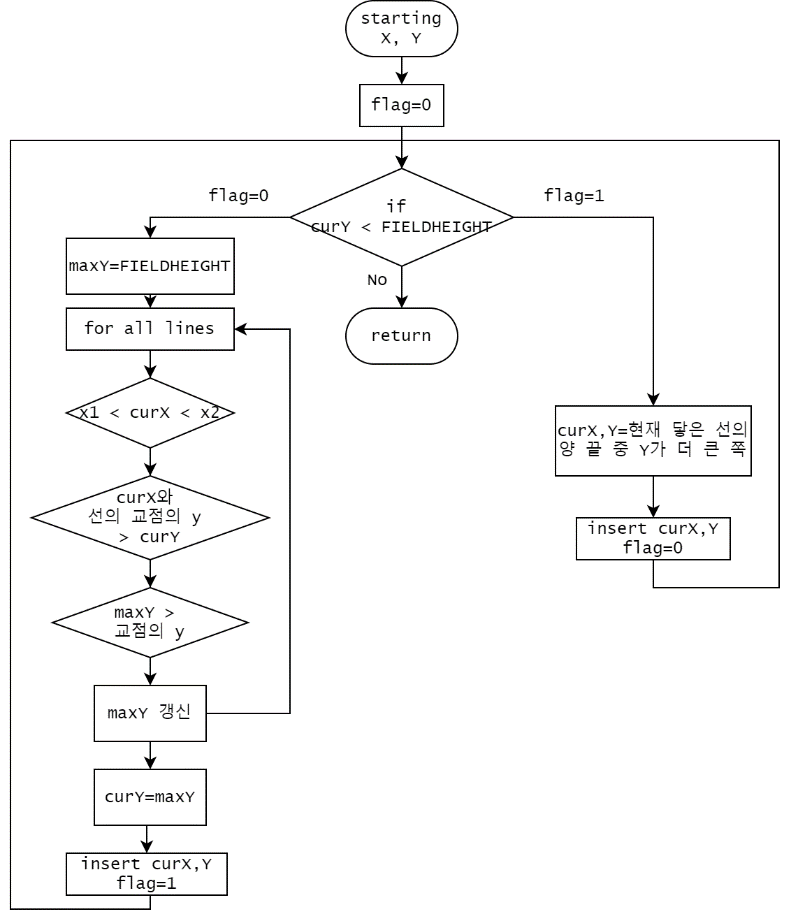
WaterFall 문제에서 물이 떨어져 흐르는 경로를 계산하여 화면에 나타내는 것을 openFrame Works로 구현한다.

1. 실습 구현 내용

물이 떨어져 흐르는 경로를 구하기 위해서는 물의 경로가 바뀌는 점의 좌표를 알아야한다. 물의 경로는 중간에 선을 만나거나, 선의 끝까지 흘렀을 때 바뀐다. 경로는 떨어짐과 흐름이 번갈아 이뤄지며, 두 선이 겹치는 경우가 없으므로 떨어짐 다음에는 반드시 흐름이, 흐름 뒤에는 항상 떨어짐이 진행된다. 따라서 물의 경로를 구하기 위해 물이 떨어지는 경우와 물이 흐르는 경우 두가지를 고려해야한다.

물이 떨어지는 경우는 x좌표는 그대로 인 체 y 좌표만 변화한다. 현재 점 아래의 선 중 가장 가까운 선과 만나는 점을 구한다. 그리고 현재 좌표를 교점으로 갱신한 뒤 저장하고, 현재 점과 닿은 선의 정보도 저장한다.

물이 흐르는 경우는 선분의 끝까지 흐른다. 따라서 현재 점과 닿은 선의 양 끝 점 중 y 좌표가 더 아래 있는 곳으로 흐른다. 따라서 현재 좌표를 한 쪽 끝으로 갱신한 뒤 저장한다.



이를 구현하기 위해, 연결 리스트 클래스를 만들어 사용했다. 사용한 연결 리스트의 구조체는 아래와 같다.

|  |  |
| --- | --- |
| class Node {  public:  float x;  float y;  Node\* link;  Node() {  x = NULL;  y = NULL;  link = NULL;  }  }; | class SLList {  private:  Node\* first;  Node\* end;  public:  SLList();  Node\* front();  void insert(float x, float y);  void freeSLL();  void printSLL();  }; |

연결 리스트는 다음의 멤버함수를 가진다.

SLList();

SLList의 생성자로 first와 end를 초기화한다.

Node\* SLList::front();

연결 리스트의 맨 처음 값인 first를 반환한다.

void SLList::insert (float x. float y);

연결 리스트의 맨 뒤에 좌표 x, y를 삽입한다.

void SLList::freeSLL();

연결 리스트의 모든 노드를 해제하여 연결 리스트를 지운다.

void SLList::printSLL();

디버깅 용으로 사용한 출력함수로, 연결 리스트의 모든 값을 순서대로 출력한다.

연결 리스트를 사용하기 위해 입력한 점의 데이터를 담은 inputdot의 구조체도 수정하였다.

|  |
| --- |
| typedef struct inputdot {  int x1;  int y1;  SLList\* water;  } inputdot; |

물이 떨어져 흐르는 경로를 계산하기 위해 아래의 함수를 작성/수정했다.

void ofApp::initializeWaterLines()

물이 떨어지는 경로를 담은 연결 리스트를 생성하고, 값을 저장하는 함수이다. 위에서 설명한 물이 흐르는 방법을 계산하는 알고리즘에 따라 x, y 값을 구해 저장한 노드를 연결 리스트의 맨 뒤에 삽입한다. waterflag 변수를 사용해 물이 떨어지는 경우와 물이 흐르는 경우를 번갈아 처리한다. 현재 y 좌표인 curY가 화면 크기인 728을 넘어가면 함수가 종료된다.

void ofApp::keyPressed(int key), void ofApp::keyReleased(int key)

물은 D키를 누른 뒤, S키를 눌렀을 때만 그려져야 하며, 물이 그려진 이후에는 E를 입력 받기 전까지 Q와 E 이외의 어떤 입력도 무시해야 한다. 이를 위해 waterlineflag라는 변수를 추가해 입력 상황을 제어하였다. waterlineflag는 drawflag가 1이고, S를 입력 받았을 때만 1이다. E를

입력 받으면 다시 0으로 변한다. 방향키가 눌리면 flag에 상관없이 명령을 수행하던 keyReleased 함수도 waterlineflag가 0일 때만 명령을 수행하도록 수정하였다.

void ofApp::draw();

drawflag가 1이고 waterlineflag가 1인 경우, 물의 경로를 출력한다. 현재 선택된 점에 대한 연결 리스트를 모두 탐색하며 현재 노드의 점과 다음 노드의 점을 잇는 굵기 4의 파란 선을 그린다.

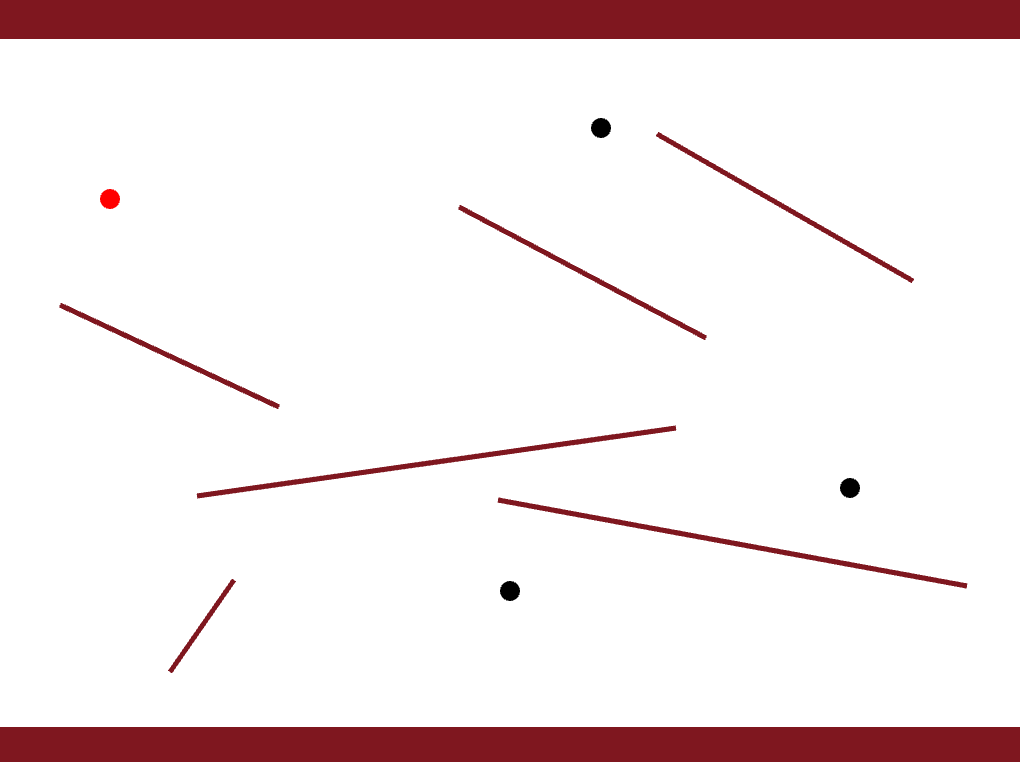
1. 실습 환경

Microsoft Visual Studio Community 2019 버전 16.7.7

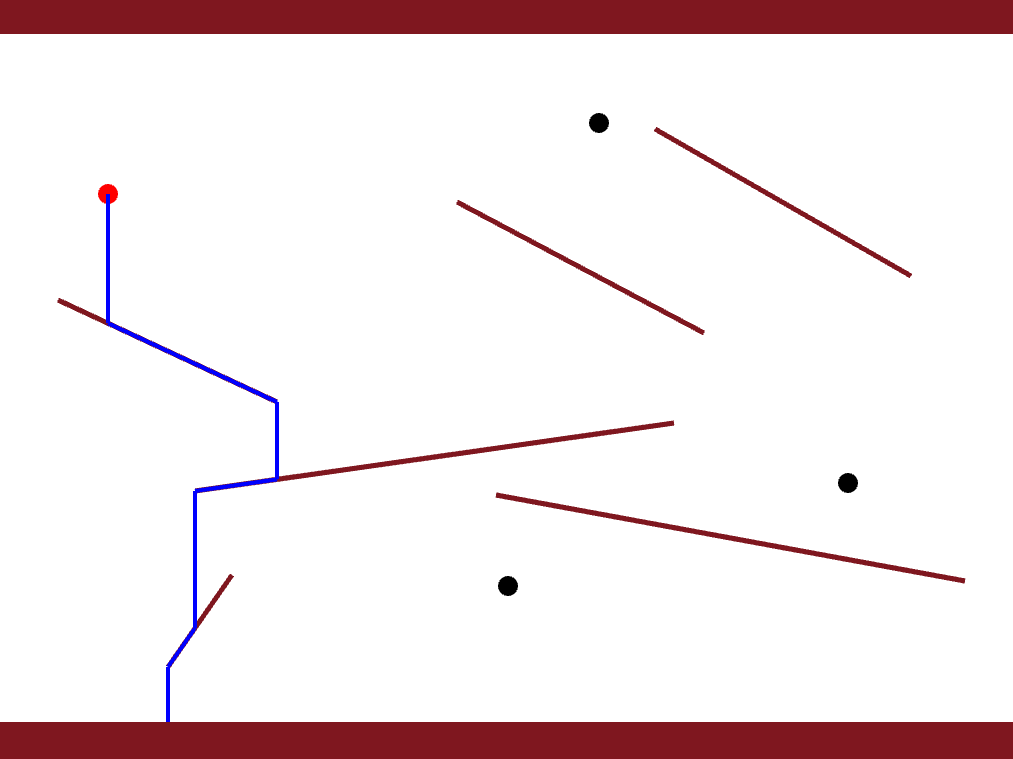
OpenFrameworks v0.11.0 vs2017 Release

1. 실습 결과 및 분석

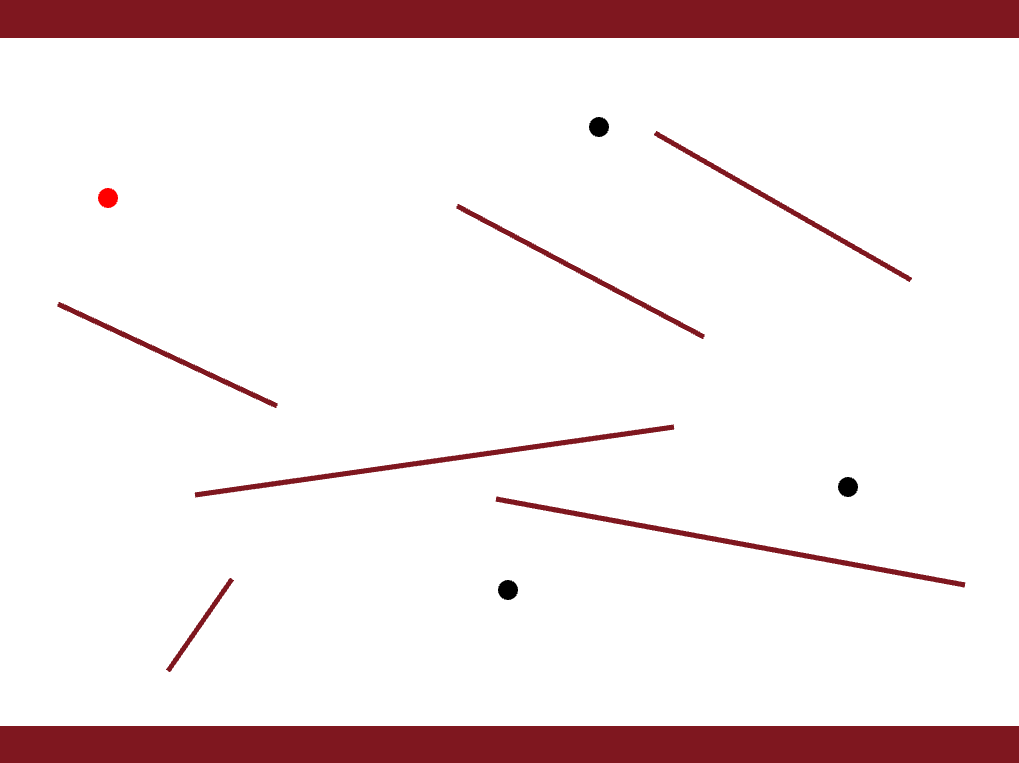
파일을 불러온 후, D를 누르면 아래의 상태이다.



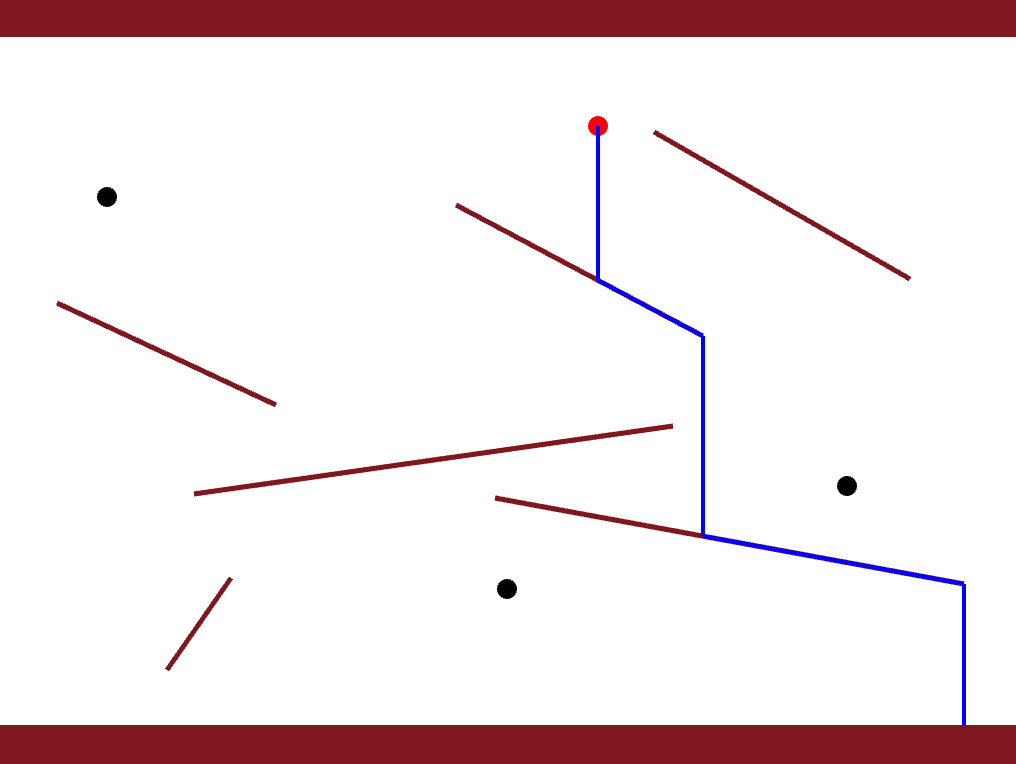
이때 S를 누르면 빨간 점에서 물이 흐른다.



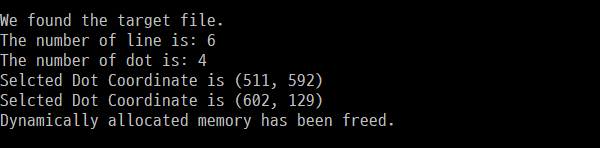
E를 누르면 물이 멈춘다. E를 누르기 전까지 E나 Q 이외의 다른 입력은 무시한다.



방향키를 사용해 다른 점을 선택하고, E를 눌러 물이 흐르게 할 수 있다.



Q를 누르면 동적 할당된 메모리를 모두 해제하고 프로그램이 종료된다.



1. 결론

이번 실습에서는 지난 실습에서 openFrameWorks를 통해 구현한 Waterfall의 기본 틀에 물이 흐르는 것을 표현하는 기능을 추가하였다. 물의 흐름은 물의 이동 경로의 좌표를 구해 해당 좌표를 잇는 선을 그리는 방식으로 표현하였다. 점은 두 직선의 방정식을 통해 교점을 구하였고, 선분의 직선의 방정식은 점의 양 끝점의 좌표를 통해 구하였다. 이렇게 구한 점의 좌표는 연결 리스트에 저장한 뒤, 흐르기 버튼을 눌렀을 때 연결 리스트를 처음부터 탐색하며 선분을 그려 나갔다.