12주차 결과보고서

전공: 아트엔테크놀로지 학년: 3학년 학번: 20191172 이름: 함승우

1. 프로그램에서 자료구조, 그 자료구조로 화면에 그리는 방법

vector<vector<char>> maze;

이 프로그램에서 readFile() 후, 읽은 후 자료구조에 저장하고, 화면에 그리기 전에 2차원 vector에 저장했다. 이 maze는 각각의 인덱스에 ‘+’, ‘-‘, ‘|’ 세 개가 들어있다. 그 후 struct를 요소로 가지는 vector를 구성했다.

struct mazeline {

float startX;

float startY;

float endX;

float endY;

};

vector<mazeline> mazel;

이 vector에는 화면에 그리기 위한, 시작점의 좌표와 끝나는 점의 좌표를 저장할 수 있다. maze (2차원 벡터)로 받은 결과를 통해서 mazel에 저장했다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

input된 index인 i와 j를 기준으로 해서 ofDrawLine을 하기위해 start point와 end point를 정했다.

이 작업의 공간 복잡도는 input file을 통해서 가로의 길이, 세로의 길이를 계산하고, 이를 각각 N과 M으로 한다면, 공간 복잡도는 O(N \* M)이며, maze를 하나씩 읽어가며 txt\_com(mazeline으로, 시작점과 끝점을 저장할 수 있는)에 저장을 하게 되므로 시간 복잡도 또한 O(N\*M)이다.

2.

실험 전에는 +, -, |를 입력받을 때 무리 없이 쉽게 할 수 있을 것 같았다. 하지만 이 실습의 스켈레톤 코드를 받은 뒤에 확인한 것은

ofFile file(fileName);

ofBuffer buffer = ofBufferFromFile(fileName);

로 알 수 있듯이, 기존의 쓰던 방식과 많이 달라서 당황하였다. 기존에, getline 함수를 사용하여 string을 읽어오곤 하거나, getline이 NULL인 것을 설정해서 read 동작을 끝내는 방식을 사용했지만, 여기선,

string first\_line = buffer.getNextLine();

!buffer.isLastLine())

buffer.rbegin();

를 사용하여 다양한 동작을 수행할 수 있었다.

또한, 실습자료에서 버튼 및 윈도우 창을 제시하였는데, 이를 예비보고서를 쓰며, 천천히 읽어본 결과 앞으로 openFrameWork상에서 구현 가능할 것 같았다.

https://openframeworks.cc/documentation/utils/ofFile/