12주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20191619 이름: 이동석

**1. 실습 자료구조**

미로를 오픈프레임워크로 그리기 위해 사용한 자료구조는 2차원 포인터배열이며 변수명은 input이다. 이 포인터배열을 미로의 너비와 높이만큼(실제 미로의 높이와 너비와는 다르다. 텍스트파일로 할당 받을 경우 실제 미로의 너비와 높이보다 2배 +1 만큼이다. 하지만 편의상 미로의 너비와 높이라고 지칭한다.) 동적할당을 한 후, 각 줄을 읽고 값을 할당한다.

텍스트 파일에는 ‘+’ ‘-‘ ‘|’ ‘ ‘ 총 4가지 형태의 char형 데이터타입이 저장되어있다. 우선, 시작하기에 앞서 미로의 바깥 테두리를 그려준다.

ofDrawLine(0, 0, (WIDTH-1) \* 20, 0);

ofDrawLine(0, 0, 0, (HEIGHT-1) \* 20);

ofDrawLine(0, (HEIGHT-1) \* 20, (WIDTH-1) \* 20, (HEIGHT-1) \* 20);

ofDrawLine((WIDTH-1) \* 20, 0, (WIDTH-1) \* 20, (HEIGHT-1) \* 20);

// 20배를 해주었다.

input 2차원 배열에 저장되어 있는 값을 살펴보면, 첫 row 줄과 첫 column줄 그리고 마지막 row, 마지막 column 줄이 모두 미로의 테두리를 나타내고 있음을 알 수 있다. 따라서, 테두리를 그린 상태이므로 2중 포문은 i=1 to HEIGHT-1 , j=1 to WIDTH-1 로 해준다.

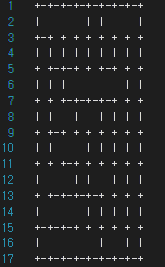
이후, input 배열을 row major order로 읽어준다. 앞서 말한 총 4가지 형태중에 ‘|’ 는 세로벽을 나타내고, ‘-‘는 가로벽을 나타낸다. 따라서 두가지 값에 대해 if 문을 통해 예외처리를 해준 후, ofDrawLine을 통해 선을 그려주면 된다.

if (input[i][j] == '-')

ofDrawLine((j-1) \* 20, i \* 20, (j+1) \* 20, i \* 20);

else if (input[i][j] == '|')

ofDrawLine(j \* 20, (i-1) \* 20, j \* 20, (i+1) \* 20);



8 \* 8 미로이지만, 텍스트파일로는 17 \* 17로 그려진다. 편의상 17 \* 17 을 너비와 높이로 부른다.

위 그림과 같은 파일을 읽고 미로를 그린다고 생각해보자. 2번째 줄을 읽으면서 ‘|’을 만나게 되면 세로벽을 그려주어야한다. 이때 세로벽의 경우에는 현재위치와 위 아래를 연결해주는 선을 그려주어야 한다. Row major order로 현재 반복문을 진행하고 있으므로, ‘|’의 좌표는 i가 y좌표 j가 x좌표로 생각된다. 따라서, 현재 텍스트상 ‘|’은 (j,i) 에 위치하며 그림은 (j-1,i) 와 (j+1,i)를 잇는 선분을 그려야한다.

같은 방법으로 ‘-‘역시 양 옆을 이어주어야 하므로, (j,i-1) 과 (j,i+1)을 이어주는 선분을 그려주어야 한다. 이때, 시간복잡도의 경우에는 2중포문을 사용했고 대략적으로 미로의 높이 \* 미로의 너비 이므로, O(HEIGHT \* WIDTH) 가 된다. 공간복잡도 역시 앞서 동적할당한 크기만큼인 O(HEIGHT \* WIDTH) 가 됨을 쉽게 알 수 있으며, 실험전에 생각했던 방법과 큰 차이는 없으며, 실수 했던 부분은 선분을 연결해 줄 때, + 부분도 고려해서 연결해 주어야 하는 점이었다.

**2. 습득한 내용**

windows.h 에 관한 내용을 조금 살펴보면서, 윈도우 api에 대해 대략적으로나마 이해할 수 있었다. 또한, 오픈프레임워크가 실행 될 때 함수 호출방식 같은 부분에 대해 이해하는데 있어서 도움이 많이 되었다. 혼자 힘으로 이 코드를 다 짤 수 있을까 라는 고민을 하며, 아직 부족한 부분이 많다는 점을 느꼈다.