12주차 결과 보고서

전공: 아트앤테크놀로지학과 학년: 3학년 학번: 20201116 이름: 이수빈

* 화면에 미로를 그리기 위해서 텍스트 파일을 그대로 읽어 저장 할 maze라는 string형 2중 vector를 vector<vector<string>>으로 정의하였다. 본 실험에서는 메뉴를 구현하기 위해 ofxWinMenu라는 라이브러리를 사용하지만, 맥 os에서는 호출이 불가능하 전 프로젝트인 waterfall과 동일한 방식으로 버튼을 구현하였다.

우선 파일을 읽기 위해서 key 값인 ‘l’을 누른다. 그러면 waterfall과 동일하게 파일을 선택할 수 있고, readFile()을 호출하게 된다. readFile()에서는 미로를 그리지 못하기 때문에, 이를 옮겨 담아 draw()함수에서 그래픽으로 전환 해 주어야한다. 그러기 위해 위에 언급한 바와 같이 maze라는 이중 벡터를 정의하였고, 이에 저장하였다.

readFile()에서는 우선 파일을 열고 이에 성공하면 버퍼를 이용해 파일을 한 줄 씩 읽는 방식을 사용하였다. 한 줄에 담겨있는 문자의 갯수는 전체 maze의 넓이의 2배에 1을 더한 것과 같기 때문에, 문자열의 크기 -1 의 절반을 maze의 넓이로 지정해준다. 또한 문자열의 개수는 maze의 높이\*2+1과 같기 때문에 문자열의 개수-1의 절반을 maze의 높이로 정의한다. 그 후, 버퍼를 이용해 문자열 한 줄 씩 읽어온 후, 이 문자열 전체를 maze에 저장하면 된다.

maze에 저장한 것을 draw()함수에서 그래픽 전환해주면 된다. 가로 선과 새로 선이 있는 문자에 대해서만 이를 출력해주면 된다. 이때, 그래픽으로 나타내기 위해서 각 문자 (가로 문자와 새로 문자)를 두께를 갖는 선으로 표현해 준다. 또한, 이 과정을 거치며 나중에 DFS구현을 위한 그래프에도 저장을 해준다. DFS 구현은 다음 주차에서 자세히 다루도록 하겠다.

그래픽 전환 작업의 공간 복잡도는 maze의 크기와 같다. 즉, 공간 복잡도는 미로의 높이를 H, 넓이를 W라 하였을 때, O(2H\*2W)와 같다. 시간 복잡도는 미로를 읽을 때, maze 이중 배열의 크기만큼, draw에서도 이와 마찬가지로 maze의 이중 배열의 크기만큼 시간이 걸리기 때문에 시간 복잡도는 O(2H\*2W)와 같다.

* 본 실험을 통하여 텍스트 파일을 읽는 방법 그리고 그래픽으로의 전환 방법에 대해 배울 수 있었다. Buffer를 사용하며 ofSplitString, ofBufferFromFile 등의 Buffer를 이용시 함께 사용할 수 있는 함수들에 대해서도 공부하고 익힐 수 있었다. 그리고 text로 구현되어 있는 maze를 그래픽으로 전환하는 과정에서 어떠한 방식으로 이를 그림으로 나타낼 수 있을지, 그리고 어떤 함수들을 이용해서 그릴 수 있는 지를 고민하며 오픈 프레임웍스의 draw함수의 사용 방법을 배울 수 있었다.