전공 : 컴퓨터공학과 학년 : 4 학번 : 20212020 이름 : 박민준

1. 실험시간에 작성한 프로그램에서 자료구조와 구성한 자료구조를 화면에 그리는 방법들을 설명한다. 완성한 자료구조를 이용한 그래픽 전환 작업의 시간 및 공감 복잡도를 보이고 실험 전에 생각한 방법과 어떻게 다른 지 아울러 기술한다.

- 미로를 생성하고 그리기 위해 2차원 배열 2개를 사용하였다. 먼저 input 배열은 char\*\* input으로 동적 할당된 2차원 배열이다. “.maz” 파일에서 읽어 들인 미로의 원본 데이터를 임시로 저장한다. 해당 배열은 각 줄을 동적으로 할당하여 미로의 가로 벽('-'), 세로 벽('|'), 모서리('+'), 그리고 통로나 방(' ')을 저장한다. 이는 “.maz” 파일의 미로를 이미지로 변환하여 draw하는 데 사용된다.

- 또한 ofApp.h 파일에 maze라는 2차원 벡터(vector<vector<char>>)를 추가하여 미로를 표현하였다. maze 벡터는 input 배열에서 읽은 데이터를 벡터 형태로 저장한다. 이 벡터는 미로를 표현하며, DFS 및 BFS 알고리즘을 적용할 때 사용된다.

1) 아래는 readFile() 함수의 TO DO 코드를 구현한 결과이다.

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

- 파일 다이얼로그를 통해 사용자가 “.maz” 파일을 선택하면 for문을 통해 파일의 각 줄을 읽어 maze 벡터에 추가한다. 이때, 동적으로 메모리를 할당하여 각 줄의 데이터를 input 배열에 먼저 저장한 후, maze 벡터에 변환하여 저장하였다.

2) freeMemory() 함수

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 해당 함수는 동적으로 할당된 input 배열과 maze 벡터에 할당된 메모리를 해제하여 메모리 누수를 방지한다.

3) darw() 함수

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 마지막으로 draw 함수는 2중 for문을 통해 input 배열을 순회하며 화면에 미로를 그리는데, 벽('-', '|')에 해당하는 부분을 ofDrawLine을 사용해 그린다. 이 때, ofDrawLine 함수의 인자로 Line의 양 끝 점의 좌표가 주어진다.

- 미로를 그리기 위한 draw() 함수의 시간복잡도는 O(WIDTH \* HEIGHT)로, 미로의 모든 셀을 순회한다. input 배열과 maze 벡터는 미로의 크기에 따라 공간을 사용하므로 O(WIDTH \* HEIGHT)의 공간을 사용한다. 따라서 전체 프로그램의 시간복잡도와 공간복잡도는 모두 O(WIDTH \* HEIGHT)이다.

- 미로 데이터를 저장하는 방식에 대해 실험 전에는 그래프 탐색 알고리즘인 dfs를 위해 연결 리스트형태의 자료구조 또는 단순 배열을 사용하는 것이 간단하고 효율적이라고 생각하였다. 특히 정적 배열을 사용하면 인덱스로 접근하기 쉽고, 메모리 관리가 명확할 것이라고 예상하였다. 그러나 실험 과정에서 미로의 크기에 유연하게 대응하려면 동적 배열이 필요하다는 것을 깨닫게 되었다. 동적 할당을 통해 미로의 데이터가 늘어나거나 줄어드는 경우에도 유연하게 대응할 수 있었다. char\*\* input 배열을 사용하여 파일에서 읽어들인 원본 데이터를 저장하고, 이를 2차원 벡터 vector<vector<char>> maze로 변환하여 저장하는 방식이 더 적합하다는 것을 알게 되었다. 이를 통해 데이터를 처리하는 유연성과 관리의 편의성을 모두 확보할 수 있었다.

2. 본 실험을 통해 습득한 내용을 기술하시오.

- 본 실험을 통해 미로 데이터를 시각적으로 표현하는 방법을 습득하게 되었다. 특히 draw() 함수를 통해 미로의 데이터를 화면에 시각적으로 표현하면서, 데이터의 시각적 표현이 사용자의 이해를 돕는 중요한 요소임을 알게 되었다. ofDrawLine을 사용하여 미로의 벽과 통로를 그리며, 파일에서 읽어들인 미로의 데이터를 그래픽으로 변환하는 방법을 익힐 수 있었다.

- 또한, readFile() 함수를 통해 “.maz” 파일을 읽어들이는 과정에서 파일 다이얼로그를 사용하여 사용자와의 인터페이스를 구현하는 방법을 새롭게 알게 되었다. 이를 통해 파일 입출력의 기본적인 개념과 활용 방법을 익혔으며, 파일을 읽어들이고 데이터를 저장하는 과정을 자동화하는 방법을 배울 수 있었다.