12주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20212022 이름: 이예준

**1.**

**< readFile >**

- 입력 읽기

버퍼에서 줄을 읽어 vector<string>인 lines에 저장하기 위해 for문을 이용한다.

버퍼를 순회하며 각 줄을 문자열로 변환하여 lines 벡터에 저장하고,

각 줄을 읽을 때마다 줄의 수(cnt)를 증가시킨다.

텍스트, 영수증, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 미로 크기 결정:

lines 크기와 cnt 변수 값을 이용하여 미로의 높이와 너비를 계산한다.

HEIGHT는 전체 줄 수의 절반, WIDTH는 첫 줄의 길이의 절반으로 설정된다.

텍스트, 폰트, 화이트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

- 입력 배열 할당:

미로의 구성을 저장하기 위해 2차원 문자 배열 input을 동적으로 할당하고,

lines의 각 줄을 input 배열에 복사한다.

텍스트, 폰트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명

- 미로 배열 할당:

각 위치에서 벽의 유무를 판단하기 위해 2차원 배열 Maze를 동적으로 할당한다.

Maze의 각 요소는 maze 구조체에 대한 pointer이다.

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

- 연결 설정:

이중 for문을 이용하여 셀 간의 연결을 설정한다. 공백, 즉 경로 역할을 하는 쉘의 입장에서

상하좌우에 대한 연결을 설정한다.

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

**< draw >**

- 사각형 모드 설정:

사각형을 그리기 위한 좌표를 계산하기 편하기 위해 ofsetRectMode(OF\_RECTMODE\_CENTER);

함수를 이용해서사각형의 중심을 기준으로 그리도록 설정한다.

- 미로 셀 순회:

이중 for문을 사용하여 미로 배열의 각 셀을 순회하면서 셀의 사각형과 선을 그리기 위한 좌표를

계산한다. x1과 y1은 셀의 왼쪽 위 모서리 좌표를, x2와 y2는 셀의 오른쪽 아래 모서리 좌표를

나타내고, 각 셀은 30픽셀 간격으로 배치되며, 5의 여백을 설정한다.

먼저 각 셀의 모서리에 회색의 사각형을 그린 다음 셀이 인접 셀과 연결되지 않은 경우,,

즉 벽이 존재하는 경우 해당 방향에 하얀색의 선을 그려서 벽을 제거한다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

**< freeMemory >**

마지막으로 input 배열과 Maze 배열에 대한 사용이 모두 종료되면 동적 할당된 메모리를 해제하기 위해 free() 함수를 이용한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

위에서 설명한 자료구조는 모두 HEIGHT와 WIDTH을 이용하여 이중 for문을 사용하고 있기

때문에 시간 복잡도는 O(n2) 이다.

draw와 free는 이미 할당된 메모리를 사용하거나 할당 해제를 하는 자료구조이므로

공간 복잡도가 O(1)이며, readFile은 이중 for문으로 HEIGHT\*WIDTH만큼 메모리를 할당하기 때문에

시간 복잡도가 O(n2) 이 된다.

**2.**

이번 실험을 통해 이전 실험에서 코드를 이용해서 만든 미로 파일을

Openframeworks에서 입력파일로 이용하고, 여러가지 함수를 이용해서 그림을 그려 나가면서

Openframeworks에 대한 이해도가 한층 더 늘어난 것 같고, 그림을 그리기 위한 좌표를 계산하는

과정에서 배열의 위치와 그림을 그리는 위치와의 상관관계에 대해 자세히 배울 수 있었다.