12주차 예비보고서

전공: 수학/컴퓨터공학 학년: 3학년 학번: 20181294 이름: 임승섭

**1.**

1 - 1. bool ofApp::readFile() {

ofFileDialogResult openFileResult = ofSystemLoadDialog(“Select .maz file”)

string filePath

size\_t pos

if (openFileResult.bSuccess) { // 파일이 잘 로드되었는지 확인한다.

ofLogVerbose(“User selected a file”) // 파일을 선택했다는 로그를 남긴다.

string fileName openFileResult.getName() // fileName에 파일 이름을 저장.

cout << “file name is” << fileName << endl // fileName 출력

filePath = openFileResult.getPath() // filePath에 파일 경로 저장

cout << “Open” << endl

pos = filePath.find\_last\_of(“.”) // 확장자 구분하기 위해 “.”의 위치 저장

if (pos != string::npos && pos != 0 && filePath.substr(pos+1) == “maz”) {

// 문자열이 존재하고 pos가 0이 아니고 pos 다음 문자열(확장자)가 maz인지 확인

ofFile file(fileName) // file 불러옴

if (!file.exist()) { // file이 존재하지 않을 때 에러 출력 후 false return

cout << “Target file does not exists.” << endl

return false

}

else { // file 존재하면 확인 메세지 출력 후 isOpen 1로 변경

cout << “We found the target file.” << endl

isOpen = 1

}

// 파일 정보를 읽어온다.

int HEIGHT = 0, WIDTH = 0; // 파일에 있는 미로의 가로 세로 길이를 저장할 변수. 초기값은 0이다.

ofBuffer buffer = ofBufferFromFile(file); // file을 로드

for (string line : buffer.getLines()) // getline을 반복해서 미로의 세로 길이를 얻어온다.

HEIGHT ++;

WIDTH = line.length(); // 미로의 가로 길이를 받아온다.

// 세로 길이와 가로 길이만큼 2차원 배열에 동적할당하여 미로의 정보를 저장할 배열 생성

maze\_array = new char\*[HEIGHT];

for ( int i = 0; i < HEIGHT; i++)

maze\_array[i] = new char[WIDTH];

string maze\_line // 한줄씩 저장할 예정. 한 줄을 저장할 문자열 변수

for (int i = 0; i < HEIGHT; i++) {

getline(file, maze\_line) // 파일에서 가로 읽어옴.

for (int j = 0; j < WIDTH; j++)

maze\_array[i][j] = maze\_line[j]; // maze\_array에 각 줄을 그대로 저장한다.

}

}

}

}

1 - 2. void ofApp::freeMemory() {

// 동적할당으로 주었던 메모리들을 해제한다. 2차원 배열인 maze\_array에 대해서 delete[]를 이용해 메모리 해제를 진행한다.

for (int i = 0; i < HEIGHT; i++)

delete[] maze\_array[i]

delete[] maze\_array

}

1 - 3. void draw() {

// 기본 세팅 및 변수 설정.

ofBackground(0, 0, 0, 0)

ofSetColor(100)

ofSetLineWidth(5)

int i, j

int curr\_x = 0, curr\_y = 0 // 현재 위치를 나타내는 좌표이다.

int length = 10 // 미로 방의 size를 나타낸다. 즉, 방 하나는 length x length의 size를 가진다.

for (i = 0; i < HEIGHT; i++) { // 2차원 배열 maze\_array의 모든 원소를 탐색한다.

for (j = 0; j < WIDTH; j++) {

if (i % 2 == 0) { // 수직 경로를 확인한다.

if (maze\_array[i][j] == ‘+’) // 방의 모서리를 나타내는 곳.

continue // 따로 처리할 일은 없다.

else if (maze\_array[i][j] == ‘-‘) // 수직 경로가 막혀있는 곳

// 막혀있다는 벽을 그려준다.

ofDrawLine(curr\_x, curr\_y, curr\_x + length, curr\_y)

// 다음 x좌표로 넘어간다.

curr\_x += length

else if (maze\_array[i][j] == ‘ ‘) // 수직 경로가 뚫려있는 곳

// 따로 그릴 것 없이 다음 x좌표로 넘어간다.

curr\_x += length

else if (i % 2 == 1) { // 수평 경로를 확인한다.

if (maze\_array[i][j] == ‘|’) // 수평 경로가 막혀있는 곳

// 막혀있다는 벽을 그려준다.

ofDrawLine(curr\_x, curr\_y, curr\_x, curr\_y + length)

// 다음 x좌표로 넘어간다.

curr\_x += length

else if (maze\_array[i][j] == ‘ ‘)

if (j % 2 == 0) // 수평 경로가 뚫려있는 곳

curr\_x += length // 다음 x좌표로 넘어간다.

else if (j % 2 == 1) // 미로의 방을 의미한다.

continue // 따로 처리할 일이 없다.

if ( j == WIDTH - 1)

curr\_y += length // 수평 경로까지 확인한 후에는 y좌표를 올려주어 다음 줄을 그릴 수 있게 한다.

**2.**

. 미로 프로그램 윈도우 창에서 메뉴와 버튼 추가 방법에 대해 조사하시오

ofxWinMenu 기능을 이용하면 윈도우 창에 미로 프로그램의 메뉴를 생성할 수 있다. 우선, “ofxWinMenu.h” 와 “ofxWinMenu.cpp’를 비쥬얼 스튜디오 프로젝트에 생성한다. 다음으로 “ofxWinMenu/src”를 추가적으로 “C/C++/General/Additional Include Directories”에 추가한다. ofApp의 헤더파일로 ofxWinMenu.h를 포함시킨다.

설정이 끝났다면 ofApp.h 파일에서 menu object를 선언한다. “ofxWinMenu \* menu”. 또한, ofApp 내에서 ofxWinMenu의 내용을 받아올 수 있는 함수를 선언한다. “void appMenuFunction(string title, bool bChecked)”. 이제 menu object를 생성한다. “menu = new ofxWinMenu(this, hWnd)” 응용 프로그램의 클래스 포인터와 창의 hWnd를 넣어 메뉴로 받는다. 이 메뉴들을 appMenuFunction에 넘겨주는 함수를 생성한다.. “void CreateMenuFunction(void(ofApp::\*function\*)(string title, bool bChecked));” “menu->CreateMenuFunction(&ofApp::appMenuFunction)

메뉴가 각각의 item으로 채워지고, 그 item들을 다룰 수 있도록 메뉴를 생성한다. “HMENU CreateWindowMenu()” 메뉴 bar에 나타날 수 있도록 popup menu를 추가한다. “HMENU AddPopupMenu(HMENU hMenu, string menuName)” popup menu에 item을 추가한다 “bool AddPopupItem(HMENU hSubMenu, string ItemName, bool bChecked)” bChecked를 이용하여 메뉴의 체크 여부를 판단할 수 있다. 만약 자동체크 여부를 판단하고 싶다면 뒤에 bool bAutoCheck 까지 추가해준다.

pop menu item들을 분리하는 seperator를 추가할 수도 있다. “bool AddPopupSeperator(HMENU hSubMenu)”.

마지막으로 “bool SetWindowMenu()”를 통해 윈도우 창을 위의 내용으로 설정한다.