12주차 예비보고서

전공: 수학과,컴퓨터공학과 학년: 3학년 학번: 20181256 이름: 김도현

1-1.

Pseudo code 를 작성하면서 옆에 같이 주석처리를 하여 코드 설명을 하였다.

해당 실험 내용은 11주차에서 만든 maze.maz 파일을 읽어들여 openframework 상에서 window창에 이를 다시 그리는 과정이다. 다음은 Pseudo code 이다.

bool readFile(const char\* filename) //readFile 함수에서 파일을 읽고 이를 자료구조화 하여 배열에 넣어주고자 한다.

ofFileDialogResult openFileResult=ofSystemLoadDialog(“select.maz file”);//openFileDialogResult 클래스에 openFileResult 를 설정하고 ofSystemLoadDialog에서 파일을 .maz 파일을 선택할 수 있도록 해당 창에 필요한 문구를 출력해주고 파일을 가져온다.

String filePath;//파일의 경로를 가지는 string 변수이다

Size\_t pos;// 확장자를 확인할 때 사용한다. 즉 여기서는 “.” 위치가 들어올 것이다.

if(openFileResult.bSuccess) //만약 파일을 잘 불러왔다면

ofLogVerbose(“User seleceted a file”);// 파일을 선택되었다는 로그를 남기는 클래스이다.

string filename=openFileResult.getName(); //파일 이름을 filename string 변수에 넣는다.

Cout<<”file name is “<<filename<<”\n”<<endl; //파일 이름을 출력해본다.

filePath=openFileREsult.getPath()//파일 경로를 filePath에 넣는다.

pos=filePath.find\_last\_of(“.”);// pos 에 위에서 말한 것 처럼 확장자를 구분하기 위해 “.” 의 위치를 입력한다.

if(pos!=0&& filePath.substr(pos+1)==”maz”) //pos가 0 이 아니고 pos 다음 문자열이” maz” 즉 .maz 확장자 파일이라면

ofFile file(filename) //ofFile을 통해 file을 불러온다

if(!file.exist) return false //file이 존재하지 않는다면 false를 return 한다.

else isOpen=1; //isopen을 1로 설정한다.

//여기서 template에서 주어지지 않은 내가 작성할 코드이다.

string num\_line; //getline을 통해 file을 가져올 때 사용할 string 변수이다.

If(drawing) =>freememory(); //미로가 이미 그려져 있다면 메모리를 해제시켜준다.

HEIGHT=0;WIDTH=0; //미로를 그릴 실제 가로와 세로 길이이다.

여기서 주의해야 할 점은 미로의 가로 세로 길이(방 기준 행,열)가 아니라 테두리 및 벽까지 모두 포함하는 길이이다.

while(getline(file,num\_line)) HEIGHT++;// getline을 통해 한 줄 씩 maze.maz 파일을 읽어오는데 끝까지 다 읽으면 이는 HEIGHT 길이와 같다.

input=new char\*[HEIGHT]; //input 메모리 동적할당 2차원 배열 만들기

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

input[i]=new char[num\_line.length()] //num\_line의 길이는 미로의 WIDTH와 같다.

//input은 .maz 파일의 모든 정보를 담는 글로벌 변수 2차원 배열로 다음과 같이 HEIGHT, WIDTH를 동적할당 해준다.

ofFile file2(fileName); //다시 파일을 읽어 이제 input 2차원 동적 배열에 그에 해당하는 값들을 넣어준다.

String word; // getline을 통해 읽어올때 사용하는 string 변수이다.

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

getline(file2,word); //한 줄 씩 .maz 파일의 내용을 getline 을 통해 word에 읽어들인다.

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

input[i][j]=word[j] //input 2차원 배열에 word[j] 즉 하나의 단어만큼 input에 넣는다.

else{ //확장자가 .maz 가 아니라면

return false; // false를 return해주어 파일을 못 읽음을 알린다.

}

1-2.

동적 할당받은 2차원 배열 메모리를 다시 해제 시켜주는 함수이다. 이 함수로 인해 미로가 이미 그려져 있어도 다시 미로를 윈도우 창에 그릴 수 있고 exit할 때 역시 동적 메모리를 해제 해주어 storage heap 영역에 memory garbage가 생기기 않도록 한다.

void freeMemory(){

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

delete[] input[i]; 우선 반복문을 돌면서 1차원 배열부터 메모리를 해제 시켜주고

delete[] input; 전체의 메모리를 해제시킨다.

}

이 함수는 exit 버튼을 누를 때나 readfile 함수에서 이미 미로가 그려져 있는 상태일 때 아마도 호출하여 input 배열을 지워야 할 것 같다.

1-3.

readFile 을 통해 input에 저장시킨 정보를 통해 다시 window 창에 미로를 그려주는 draw 함수이다.

void draw(){

float x=0; //미로를 그릴 때 어느정도 크기의 미로를 그릴지 모르지만 현재 선분을 그릴 위치의 x 좌표이다.

float y=0; //미로를 그릴 때 어느정도 크기의 미로를 그릴지 모르지만 현재 선분을 그릴 위치의 y 좌표이다.

//여기서는 임의로 선분의 길이를 30으로 지정하였다.

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

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

if(input[i][j]==’+’) {} //아무것도 그리지 않는다.

if(input[i][j]==’-‘) //만약 해당 input index에서 값이 ‘-‘ 이면 가로 선분을 그린다.

ofDrawLine(x,y,x+30,y); //Ofdrawline을 통해 (x,y)에서 (x+30,y)로 가는 가로 선분을 그린다.

x+=30; //이후 다시 x에 30을 더해주어 다음 가로 선분을 그릴 수 있도록 x좌표를 이동한다.

If(input[i][j]==’|’) //만약 세로벽을 그려야 한다면

ofDrawLine(x,y,x,y+30); //OfDrawLine을 통해 (x,y)에서 (x,y+30)으로 가는 세로 선분을 그린다. X좌표나 y 좌표는 이동시키지 않고 그대로 있는다.

If(input[i][j]==’ ‘)//만약 space 값이 들어있다면 이는 j가 홀수라면 x를 30만큼 이동시켜야 한다. (j가 홀수일 시 미로의 방이라는 의미이다.)

if(j%2==1) x+=30;

x=0// 다음 행을 그려야하기 때문에 다시 x좌표를 0으로 둔다.

If(i%2==1) //만약 i가 홀수라면 side 벽을 다 그렸기 때문에

y+=30; // y좌표 역시 아래로 30만큼 내려준다.

}

2.

깃허브에서 ofxWinMenu 에 대한 정보를 찾을 수 있었다.

윈도우 창을 만들 때는 ofXWinMenu.h 을 include 하여 윈도우 창을 만들고 여기에 메뉴 및 버튼 등을 추가할 수 있다. 우선 vs (비쥬얼스튜디오)의 경우 ofxWinMenu.h 를 ofApp 헤더 파일에 include 한다. ofApp.h에 ofxWinMenu \*menu; 즉 menu object를 선언한다. ofApp.cpp 의 appMenuFunction(string title, bool bChecked) 함수에서는 title 에 따라 bChecked 에 대한 내용이 return 된다.

이후 새로운 window 창에 새로운 menu object 를 만들기 위해서는 menu=new ofxWinMenu(this,hWnd) 로 응용프로그램의 클래스 포인터와 해당 창의 hWnd를 넣어 메뉴로 한다. menu->CreateMenuFuntion(&ofApp::appMenuFunction) 을 통해 해당 메뉴들을 위에서 설명한appMenuFunction에 return 한다. 즉 이 함수는 이미 선언 되어 있어야 한다. HMENU hMenu=menu->CreateWindowMenu() 를 통해 새로운 윈도우 메뉴를 만든다. (HMENU 는 windows.h 헤더 파일에 정의된 class이다.)

여기서 popup 메뉴를 만들 경우 HMENU hPopup=menu->AddPopupMenu(<HMENU> hMenu, “이름”) 을 하여 만든다. 여기 popup 메뉴에 다시 popup items(popup 메뉴의 하위 버튼이라고 생각하면 된다.) 를 만들기 위해서는 menu->AddPopupItem(<HMENU> hPopup, “이름”, bool bChecked, bool AutoCheck) 이다. 체크 유무를 판단하는 bChecked, 자동체크 유무를 확인하는 AutoCheck를 통해 자동검사를 활성, 비활성화 할 수 있다. 이는 전체 화면 혹은 box에 관련했을 때 필요할 수 있다.

이후 popup menu item 들을 분리시킬 수 도 있는데 이는 menu->AddPopupSeparator(<HMenu> hPopup) 으로 item들을 나눈다. 모든 popup menu 나 item 들을 추가 한 이후 이를 set 하기 위해 SetWindowMenu() 를 통해 윈도우 창을 위에서 만든 것과 같이 설정한다.