**실험** 10주차(waterfall 1주차) 결과보고서

전공: 국제한국학 학년: 4 학번: 20181202 이름 : 김수미

## 1. OpenFramework 실습 코드들을 수행하고, 각 line이 어떤 기능을 수행하는지 작성한다.

```
4   void ofApp::setup(){
5       ofSetFrameRate(15); // Limit the speed of our program to 15 frames per second
6       // The background before we do anything with the brush
7       ofBackground(255,255,255); // draw white background
8       ofSetLineWidth(4);
9
10      draw_flag = 0;
11      load_flag = 0;
```

먼저 실습 구현에 필요한 도형들을 그리기 위한 배경을 setting 한다.

ofSetFramRate(15) : 프로그램의 프레임 속도를 제한하는 역할.

ofBackground(255,255,255) : 배경이 되는 화면의 색을 지정하는 부분. RGB로 색상을 지정해주면 되며 흰색(255,255,255)으로 지정해 주었다.

ofSetLineWidth(4) : 그려지는 선분의 굵기를 4로 지정해주는 역할.

draw_flag = 0; load_flag = 0; 프로그램 실행에 필요한 플래그들의 값을 초기화.

```
21  void ofApp::draw(){
22      ofSetColor(127,23,31);  // Set the drawing color to brown
23      // Draw shapes for ceiling and floor
24      ofDrawRectangle(0, 0, 1024, 40); // Top left corner at (50, 50), 100 wide x 100 high
25      ofDrawRectangle(0, 728, 1024, 40); // Top left bottom at (50, 50), 100 wide x 100 high
26
27
28      if( draw_flag ){
29          /* COMSIL1-TODO 3 : Draw the line segment and dot in which water starts to flow in the screen.
30          Note that after drawing line segment and dot, you have to make selected water start dot colored in red.*/
31
32          // Draw lines.
33          for (int i = 0; i < line_num; i++) {
34              ofSetColor(127, 23, 31);
35              ofSetLineWidth(5);
36              ofDrawLine(linearr[i].x1, linearr[i].y1, linearr[i].x2, linearr[i].y2);
37          }
38
39          // Draw dots(circles).
40          for (int i = 0; i < dot_num; i++) {
41              ofFill();
42              ofSetColor(0, 0, 0);
43              ofDrawCircle(dotarr[i].x, dotarr[i].y, 10);
44          }
45
46          // Draw red a dot(circle).
47          ofFill();
48          ofSetColor(255, 0, 0);
49          ofDrawCircle(dotarr[dot_index].x, dotarr[dot_index].y, 10);
```

draw( )는 화면의 상단과 하단의 장식, 물이 나오는 구멍, 선반을 그리는 함수이다.

먼저 화면 상단과 하단에 띠처럼 둘러져 있는 테두리 장식을 먼저 그린다.

ofSetColor(127,23,31) : 도형의 색을 갈색으로 지정.

ofDrawRectangle(0,0,1024,40) : 0,0 좌표 위치에 가로 1024, 세로 40 크기의 직사각형을 그린다.

ofDrawRectangle(0,728,1024,40) : 0,728 좌표 위치에 가로 1024, 세로 40 크기의 직사각형을 그린다.

전공: 국제한국학 학년: 4 학번: 20181202 이름 : 김수미

if(draw_flag) : 사용자가 d(D)키를 눌러서 draw_flag가 set된 경우에만 아래 코드를 수행한다.

for(int i=0; i < line_num; i++) : txt 파일에서 읽어 저장한 line_num 변수만큼 반복해서 선분을 그린다. 즉 총 line_num개의 선분을 그림.
ofSetColor(127, 23, 31) : 갈색으로 색깔 설정
ofSetLineWidth(5) : 선 굵기를 5로 설정
ofDrawLine ( linearr[i].x1, linearr[i].y1, linearr[i].x2, linearr[i].y2 ) : txt 파일에서 읽어 저장한 linearr배열의 값들을 이용. (x,y) 좌표의 점을 자례대로 입력하면 두 점을 잇는 선분을 그릴 수 있다.

for(int i=0; i<dot_num; i++) : txt 파일에서 읽어 저장한 dot_num 변수만큼 반복해서 점(물이 나오는 구멍)을 그린다. 즉 총 dot_num개의 점을 그림.
ofFill( ) : 내부가 색칠된 점(원)을 그릴 것이다.
ofSetColor(0,0,0) : 점의 색 = 검은색으로 설정.
ofDrawCircle(dotarr[i].x, dotarr[i].y, 10) : 원의 중심 좌표(x,y)와 원의 반지름을 입력하여 원을 그릴 수 있다. txt 파일에서 읽어 저장한 dotarr배열의 값들을 이용해 원의 위치를 지정하여 그린다.

ofFill( ); ofSetColor(255,0,0);
ofDrawCircle(dotarr[dot_index].x, dotarr[dot_index].y, 10);
= 현재 사용자에 의해 선택된 원의 색은 빨간색으로 바꿔준다. (dot_index의 Default 값 = 0)

```
57  void ofApp::keyPressed(int key){
58      if (key == 'v' || key == 'V') {
59          // HACK: only needed on windows, when using ofSetAutoBackground(false)
60          glReadBuffer(GL_FRONT);
61          ofSaveScreen("savedScreenshot_" + ofGetTimestampString() + ".png");
62      }
```

keyPressed(int key) : 사용자는 키보드를 눌러 명령을 내릴 수 있고, 해당 명령을 읽어 수행하는 역할을 하는 함수이다.
if(key == 'v' || key == 'V') : 사용자가 v(V)키를 누르는 경우 아래 코드가 수행된다.
glReadBuffer(GL_FRONT);
ofSaveScreen("savedScreenshot_" + ofGetTimestampString( ) + ".png");
= 현재 화면의 스크린샷을 생성한다.

전공: 국제한국학 학년: 4 학번: 20181202 이름 : 김수미

```
64      if (key == 'q' || key == 'Q'){
65          // Reset flags
66          draw_flag = 0;
67
68          // Free the dynamically allocated memory exits.
69          delete[] linearr;
70          cout << "Memory for line segment has been freed." << endl;
71          delete[] dotarr;
72          cout << "Memory for dot has been freed." << endl;
73          _Exit(0);
74      }
75
76      if (key == 'd' || key == 'D'){
77          if( !load_flag ) return;
78          /* COMSIL1-TODO 2: This is draw control part.
79          You should draw only after when the key 'd' has been pressed.
80          */
81          draw_flag = 1;
82          draw();
83      }
```

if(key == 'q' || key == 'Q') : 사용자가 q(Q)키를 누르는 경우 아래 코드가 수행된다.

사용자가 q 키를 누르면 프로그램을 종료한다. 이 때 동적으로 할당된 메모리를 모두 해제시킨다.

draw_flag = 0 : 플래그 값을 초기화시킨다.

delete[ ] linearr; delete[ ] dotarr;

cout << "Memory for line segment has been freed. " << endl;

cout << "Memory for dot has been freed. " << endl;

동적으로 할당된 linearr, dotarr 배열 메모리 해제 후 해제 안내 문구 출력.

_Exit(0) : 프로그램 종료.

if(key == 'd' || key == 'D') : 사용자가 d(D)키를 누르는 경우 아래 코드가 수행된다.

if(!load_flag) return : 사용자가 l(L)키를 눌러서 프로그램 수행에 필요한 txt 파일을 불러오지 않은 경우 d(D)키를 눌러 Draw명령을 내리는 것이 불가능하게 한다.

draw_flag = 1; draw( ) : draw_flag를 set 시키고 draw 함수를 수행시킨다.

전공: 국제한국학 학년: 4 학번: 20181202 이름 : 김수미

```
95  void ofApp::keyReleased(int key) {
96      if (key == 'l' || key == 'L') {
97          // Open the Open File Dialog
98          ofFileDialogResult openFileResult = ofSystemLoadDialog("Select a only txt for Waterfall");
99
100         // Check whether the user opened a file
101         if (openFileResult.bSuccess) {
102             ofLogVerbose("User selected a file");
103             cout << "We found the target file. " << endl;
104             // We have a file, so let's check it and process it
105             processOpenFileSelection(openFileResult);
106             load_flag = 1;
107         }
108     }
109
110     /* COMSIL1-TODO 4: This is selection dot control part.
111     You can select dot in which water starts to flow by left, right direction key (<- , ->).
112     */
113     if (key == OF_KEY_RIGHT) {
114         dot_index++;
115         if (dot_index > dot_num-1) dot_index = 0;
116         cout << "Selcted Dot Coordinate is (" << dotarr[dot_index].x << ", " << dotarr[dot_index].y  << ")" << endl;
117         draw(); // Change position of a red dot.
118     }
119     if (key == OF_KEY_LEFT) {
120         dot_index--;
121         if (dot_index < 0) dot_index = dot_num - 1;
122         cout << "Selcted Dot Coordinate is (" << dotarr[dot_index].x << ", " << dotarr[dot_index].y << ")" << endl;
123         draw(); // Change position of a red dot.
124
125     }
126 }
```

if(key == 'l' || key == 'L') : 사용자가 l(L)키를 누르는 경우 아래 코드가 수행된다.

ofFileDialogResult openFileResult = ofSystemLoadDialog("Select a only txt for Waterfall") : 파일을 오픈하는 프로그램을 실행시킨다. 안내문 "Select a only txt for Waterfall"을 파일을 오픈하는 창에 출력해 사용자가 확인할 수 있게 한다.

if(openFileResult.bSuccess) : 파일을 오픈하는 것에 성공한 경우.

ofLogVerbose("User selected a file");

cout << "We found the target file. " << endl : 파일 오픈에 성공했다는 안내 메시지 출력.

if(key == 'OF_KEY_RIGHT') : 사용자가 우측 방향키를 누르는 경우 아래 코드가 수행된다.

현재 점보다 한 칸 오른쪽에 있는 점을 물이 나오는 구멍으로 고르는 기능을 수행한다. 현재 선택된 점은 빨간색으로 표시된다. dotarr에 있는 점들의 정보를 이용하므로 인덱스를 이용해 배열에 접근한다.

dot_index++ : 인덱스 +1. 한 칸 오른쪽으로 이동.

if(dot_index > dot_num-1) dot_index = 0 : 만약 가장 오른쪽 점에서 오른쪽 방향키를 한번 더 누른 경우 가장 처음의 점, 즉 가장 왼쪽의 점으로 한바퀴 돌아가도록 한다.

cout << "Selected Dot Coordinate is (" << dotarr[dot_index].x << ", " << dotarr[dot_index].y << ")" << endl : 현재 선택된 점의 좌표 출력.

draw( ) : 현재 선택된 점을 빨간색으로 다시 그려주기 위해 draw 함수 호출.

**실험** 10주차(waterfall 1주차) 결과보고서

전공: 국제한국학 학년: 4 학번: 20181202 이름 : 김수미

if(key == 'OF_KEY_LEFT') : 사용자가 좌측 방향키를 누르는 경우 아래 코드가 수행된다.
현재 점보다 한 칸 왼쪽에 있는 점을 물이 나오는 구멍으로 고르는 기능을 수행한다. 현재 선택된 점은 빨간색으로 표시된다. dotarr에 있는 점들의 정보를 이용하므로 인덱스를 이용해 배열에 접근한다.

dot_index -- : 인덱스 -1. 한칸 왼쪽에 있는 점으로 이동.

if(dot_index < 0) dot_index = dot_num - 1 : 만약 가장 왼쪽 점에서 왼쪽 방향키를 한번 더 누른 경우 가장 마지막의 점, 즉 가장 오른쪽의 점으로 한바퀴 돌아가도록 한다.

cout << "Selected Dot Coordinate is (" << dotarr[dot_index].x << ", " << dotarr[dot_index].y << ")" << endl : 현재 선택된 점의 좌표 출력.

draw( ) : 현재 선택된 점을 빨간색으로 다시 그려주기 위해 draw 함수 호출.

```
173  void ofApp::processOpenFileSelection(ofFileDialogResult openFileResult) {
174      openFileResult.getPath();
175      ifstream input;
176      input.open(openFileResult.filePath);
177
178      // input >> :  read txt file line by line
179      // Save information of lines in linearr array.
180      input >> line_num;
181      cout << "The number of line is: " << line_num << endl;
182      linearr = new line[line_num];
183      for (int i = 0; i < line_num; i++) {
184          line newline;
185          input >> newline.x1;
186          input >> newline.y1;
187          input >> newline.x2;
188          input >> newline.y2;
189          // Check rather a line goes out from the screen or not.
190          if (newline.x1 <= ofGetWidth() || newline.x2 <= ofGetWidth() || newline.y1 <= ofGetHeight() || newline.y2 <= ofGetHeight())
191              this->linearr[i] = newline;
192      }
193
194      // Save information of dots in dotarr array.
195      input >> dot_num;
196      cout << "The number of dot is: " << dot_num << endl;
197      dotarr = new dot[dot_num];
198      for (int i = 0; i < dot_num; i++) {
199          dot newdot;
200          input >> newdot.x;
201          input >> newdot.y;
202          // Check rather a dot goes out from the screen or not.
203          if (newdot.x <= ofGetWidth() || newdot.y <= ofGetHeight())
204              this->dotarr[i] = newdot;
205      }
206      input.close();
207  }
```

processOpenFileSelection(ofFileDialogResult openFileResult) : txt 파일을 읽어 프로그램 실행을 위한 정보를 저장하는 역할을 하는 함수이다.

openFileResult.getPath( ) : openFileResult 변수에 txt파일 이름이 저장되어 있다. 해당 변수 이름으로부터 파일 경로를 얻어 filePath에 저장한다.

input.open(openFileResult.filePath) : 입력 받은 제목의 파일을 오픈한다.

**실험** 10주차(waterfall 1주차) 결과보고서

전공: 국제한국학 학년: 4 학번: 20181202 이름 : 김수미

input >> line_num : 오픈한 txt 파일에서 첫번째 요소를 읽어와 line_num에 저장한다. txt 파일의 가장 첫번째 줄에는 선분의 개수가 주어지므로 변수 line_num에는 선분의 총 개수가 저장된다.

cout << "The number of line is: " << line_num << endl : 선분 개수 출력

linearr = new line[line_num] : 선분의 정보를 저장하기 위한 배열 동적 할당

for (int i=0; i<line_num; i++) : 선분 개수만큼 반복한다.

line newline : line 구조체 타입의 newline 변수를 선언한다.

input >> newline.x1~y2 : txt 파일에서 요소를 차례대로 읽어와 newline 변수의 x1, y1, x2, y2 필드에 저장한다.

if(newline.x1 <= ofGetWidth( ) || newline.x2 <= ofGetWidth() || newline.y1 <= ofGetHeight( ) || ofGetHeight( ) || newline.y2 <= ofGetHeight( )) this->linearr[i] = newline : 방금 newline 변수에 저장한 선분이 스크린 필드를 벗어나지 않는지 검사하여 벗어나지 않는다면 linearr 배열에 newline 정보를 저장한다.

input >> dot_num : txt 파일에서 요소 한개를 읽어와 dot_num에 저장한다. 선분 정보 다음에는 점의 개수와 점의 좌표가 주어지므로 변수 dot_num에는 점의 총 개수가 저장된다.

cout << "The number of dot is: " << line_num << endl : 점 개수 출력

dotarr = new dot[dot_num] : 선분의 정보를 저장하기 위한 배열 동적 할당

for (int i=0; i<dot_num; i++) : 선분 개수만큼 반복한다.

dot newdot : dot 구조체 타입의 newdot 변수를 선언한다.

input >> newline.x~y : txt 파일에서 요소를 차례대로 읽어와 newdot 변수의 x, y 필드에 저장한다.

if(newdot.x <= ofGetWidth( ) || newdot.y <= ofGetHeight( )) this->dotarr[i] = newdot : 방금 newdot 변수에 저장한 점이 스크린 필드를 벗어나지 않는지 검사하여 벗어나지 않는다면 dotarr 배열에 저 newdot 정보를 저장한다.

**실험** 10주차(waterfall 1주차) 결과보고서

전공: 국제한국학 학년: 4 학번: 20181202 이름 : 김수미

```
21      /* WaterFall-related member variables Regions */
22      // flag variables
23      int draw_flag;
24      int load_flag;
25
26      // Line segment and dot related variables
27      int line_num, dot_num;
28      float dot_diameter;
29
30      // index for dots
31      int dot_index = 0;
32
33      /* WaterFall-related member functions */
34      void processOpenFileSelection(ofFileDialogResult openFileResult);
35      // 2nd week portion.
36      void initializeWaterLines();
37
38      /* WaterFall-related member structure */
39      struct line {
40          int x1; int y1;
41          int x2; int y2;
42      }; line *linearr;
43
44      struct dot {
45          int x; int y;
46      }; dot *dotarr;
```

프로그램에 사용되는 전역변수와 구조체 변수를 선언한 것이다.

line 구조체는 시작 점의 좌표와 끝 점의 좌표를 저장하는 필드가 있고 linearr는 line 구조체 변수들로 이루어진 배열이다.

dot 구조체는 점 하나의 좌표를 저장하는 필드가 있고 dotarr는 dot 구조체 변수들로 이루어진 배열이다.