9주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20191619 이름: 이동석

**1. Basic\_openFramework실습**

**1-1. Basic shape(1)**

ofBackground(0);

프로그램이 실행 될 때, 배경 색을 지정한다. 0이기 때문에 검정색이다.

ofSetColor(255);

도형의 색을 지정한다. 이때 도형의 색은 흰색이다.

ofDrawRectangle(50, 50, 100, 100);

직사각형을 그린다. 위치 x,y 값을 받고, 높이와 너비 값을 받는다.

ofDrawCircle(250, 100, 50);

원을 그린다. 중심 위치와 반지름을 넘겨 받는다.

ofDrawEllipse(400, 100, 80, 100);

타원을 그린다. 중심 위치와 장축과 단축을 받는다.

ofDrawTriangle(500, 150, 550, 50, 600, 150);

삼각형을 그린다. 3꼭지점의 위치를 받는다.

ofDrawLine(700, 50, 700, 150);

두 점사이를 선을 긋는다.

**1-2. Basic shape(2)**

ofFill(); 도형을 그릴 때 현재 색으로 내부를 채운다.

ofNoFill(); 도형을 그릴 때 현재 색으로 둘레를 표현하고, 내부를 비운다.

**1-3. Single rectangle brush**

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {

마우스 왼쪽버튼이 눌리는 경우 참을 반환하며, bool 타입을 가진다.

ofSetColor(255);

ofSetRectMode(OF\_RECTMODE\_CENTER);

중심으로 직사각형 모드를 세트한다.

ofDrawRectangle(ofGetMouseX(), ofGetMouseY(), 50, 50);

마우스 위치를 중심으로 하는 높이 너비 50의 직사각형을 그린다.

}

ofSetBackgroundAuto(false);

bool 타입으로 자동으로 화면이 초기화 되는 것을 막는다.

float randomColor = ofRandom(50, 255);

그림을 그릴 때 마다 색을 랜덤으로 지정한다. 위코드는 50-255 사이의 색

**1-4. Bursting Rectangle brush**

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {

// If the left mouse button is pressed...

ofSetRectMode(OF\_RECTMODE\_CENTER);

int numRects = 10;

for (int r=0; r<numRects; r++) {

ofSetColor(ofRandom(50, 255));

float width = ofRandom(5, 20);

5-20 사이의 너비를 랜덤하게 할당 받는다.

float height = ofRandom(5, 20);

5-20 사이의 높이를 랜덤하게 할당받는다.

float xOffset = ofRandom(-40, 40);

x좌표 랜덤 지정

float yOffset = ofRandom(-40, 40);

y좌표 랜덤 지정

ofDrawRectangle(ofGetMouseX()+xOffset, ofGetMouseY()+yOffset, width, height);

현재 위치에서 지정한범위내의 랜덤한 위치에 랜덤한 사각형을 그린다.

}

**1-6. Glowing Cricle Brush**

ofColor 는 기본적으로 red, green, blue 로 이루어져있다. 이를 조합하여, 원하는 색을 지정할 수 있다.

ofSetFrameRate(60);

화면의 프레임을 초당 60까지로 늘린다.

ofDegToRad(360,0);

각도를 라디안 단위로 변환하여 값을 반환한다.

**1-7. Fleeing Triangle Brush**

ofSetWindowShape(1024,768);

높이와 너비만큼 화면 해상도를 설정한다.

ofVec2f mousePos(ofGetMouseX(), ofGetMouseY());

마우스 위치의 벡터정보를 저장한다.

ofVec2f p1(0,6,25);

ofVec2f p2(25,0);

ofVec2f p3(0,-6.25);

삼각형이 되도록 3개의 벡터를 만들고 저장한다.

p1.rotate(rotation);

벡터의 원점을 기준으로 각도만큼 회전시킨다.

p1 += mousePos; p2 += mousePos; p3 += mousePos;

마우스 위치만큼 벡터를 이동시킨다.

ofDrawTriangle(p1, p2, p3);

세 벡터를 이용해 삼각형을 그린다.

**1-8. update();**

설정된 프레임 마다 호출되어 함수의 코드를 수행한다.

xPos += 2;

xPos 값을 호출시마다 2씩 더해준다.

if(ofGetWidth()<xPos){

xPos = 5;

화면 밖으로 나갈 시 초기값으로 다시 지정한다.

}

**2. Waterfall**

파일을 읽고 줄의 좌표를 저장할 line\_segment 를 이중포인터로 선언하고, 점의 좌표를 저장할 dot을 이중포인터로 선언한다. 동적할당을 해주며 이후, 파일을 읽으면서 값을 할당한다. 점의 좌표들이 정렬되지 않았다면, 정렬해준다.

이후 drawflag를 입력된 키보드 값이 ‘d’라면 1로 설정해준다. ‘q’일 경우 동적으로 할당한 메모리를 모두 해제한 후 프로그램을 종료한다. 또, 키보드가 방향키로 들어올 경우, 점을 방향키에 따라 움직여 표시한다.

**3. 참고문헌**

ofTutorials - Introduction . (September 2013 - JeffCrouse). http://openframeworks.kr/tutorials/introduction/000\_introduction.html