6주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20221532 이름: 김민기

1. **Basic Shapes (1)**

ofBackground(0); //스크린 clear

ofSetColor(255); // draw() 색상 흰색으로 설정

ofDrawRectangle(50, 50, 100, 100); // 왼쪽 위의 좌표가 (50,50)이고 너비와 높이가 100인 직사각형 출력

ofDrawCircle(250, 100, 50); // 중심이 (250,100)이고 반지름이 50인 원 출력

ofDrawEllipse(400, 100, 80, 100); //중심이(400,100)이고 너비가 80, 높이가 100인 타원 출력

ofDrawTriangle(500, 150, 550, 50, 600, 150); //세 점이 (500, 150), (550, 50), (600, 150).인 삼각형 출력

ofDrawLine(700, 50, 700, 150); //양 끝이 (700, 50), (700, 150)인 선분 출력

1. **Basic Shapes (2)**

ofBackground(0);

ofSetColor(255);

ofFill(); //도형 내부 채움

ofDrawRectangle(50, 50, 100, 100);

ofDrawCircle(250, 100, 50);

ofDrawEllipse(400, 100, 80, 100);

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

ofDrawLine(700, 50, 700, 150);

ofNoFill(); //도형 내부 비움,

ofDrawRectangle(50, 200, 100, 100);

ofDrawCircle(250, 250, 50);

ofDrawEllipse(400, 250, 80, 100);

ofDrawTriangle(500, 300, 550, 200, 600, 300);

ofDrawLine(700, 200, 700, 300);

1. **Single Rectangle Brush**

#draw()

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) { //마우스 왼쪽클릭이 입력되면,

ofSetColor(255);

ofSetRectMode(OF\_RECTMODE\_CENTER); ofDrawRectangle(ofGetMouseX(),

ofGetMouseY(), 50, 50); //너비와 높이가 50, 50 인 정사각형을 마우스에 출력

}

#setup1

ofSetBackgroundAuto(false); // 배경이 지워지지 않고 유지

ofBackground(0);

#setup2

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) { // 마우스 왼쪽클릭이 입력되면,

float randomColor = ofRandom(50, 255);

ofSetColor(randomColor); //50에서 255사이에 랜덤한 색 지정.

ofSetRectMode(OF\_RECTMODE\_CENTER); ofDrawRectangle(ofGetMouseX(),

ofGetMouseY(), 50, 50);

}

1. **Bursting Rectangle Brush**

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

ofSetRectMode(OF\_RECTMODE\_CENTER); //사각형을 그릴때 사각형의 중심을 기준으로 그림

int numRects = 10; //폭발 반복할 수

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

ofSetColor(ofRandom(50, 255)); //색상 랜덤지정

float width = ofRandom(5, 20); //사각형 너비 5~20사이로 랜덤

float height = ofRandom(5, 20); //사각형 높이 5~20사이로 랜덤

float xOffset = ofRandom(-40, 40); //마우스로부터 떨어진 x좌표 길이

float yOffset = ofRandom(-40, 40); //마우스토부터 떨어진 y좌표 길이

ofDrawRectangle(ofGetMouseX()+xOffset, ofGetMouseY()+yOffset, width, height); //위에 설정한 값으로 사각형 출력

}

}

1. **Glowing Circle Brush**

ofColor myOrange(255, 132, 0); // 오렌지 색상 객체 생성

ofColor myBlue(0, 0, 255, 50); // 파란색 객체 생성

ofColor myGreen(0, 0, 255, 255); //초록색 객체 생성

cout << "Red channel:" << myGreen.r << endl;

cout << "Green channel:" << myGreen.g << endl;

cout << "Blue channel:" << myGreen.b << endl;

cout << "Alpha channel:" << myGreen.a << endl;

ofColor myYellow;

myYellow.r = 255; myYellow.b = 0; myYellow.g = 255; myYellow.a = 255;

//노란색 빈 객체 생성 후 RGB값 지정

ofColor myAqua = ofColor::aqua; //OpenFrameworks에서 제공하는 색으로 설정

ofColor myPurple = ofColor::plum;

1. **Fleeing Triangle Brush**

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

int numTriangles = 10; //생성될 삼각형 개수

int minOffset = 5; //마우스로부터 떨어질 최소 거리

int maxOffset = 70; //마우스로부터 떨어질 최대 거리

int alpha = 150;

for (int t=0; t<numTriangles; ++t) {

float offsetDistance = ofRandom(minOffset, maxOffset); //떨어질 거리 랜덤 생성

ofVec2f mousePos(ofGetMouseX(), ofGetMouseY()); //마우스 좌표 저장

ofVec2f p1(0, 6.25); //기본 삼각형 세점 설정

ofVec2f p2(25, 0);

ofVec2f p3(0, -6.25);

float rotation = ofRandom(360); //0부터 360까지 랜덤으로 회전각 설정

p1.rotate(rotation); //세점 모두 회전

p2.rotate(rotation);

p3.rotate(rotation);

ofVec2f triangleOffset(offsetDistance, 0.0); //마우스로부터 떨어진 거리 저장할 객체 생성

triangleOffset.rotate(rotation); //거리도 회전에 맞춤

p1 += mousePos + triangleOffset; //기본 삼각형을 마우스 위치와 마우스로부터 떨어질 위치로 옮김

p2 += mousePos + triangleOffset;

p3 += mousePos + triangleOffset;

ofColor aqua(0, 252, 255, alpha); //색상 설정

ofColor purple(198, 0, 205, alpha);

ofColor inbetween = aqua.getLerped(purple, ofRandom(1.0)); //aqua와 purple 사이에서 랜덤한 색 지정

ofSetColor(inbetween);

ofDrawTriangle(p1, p2, p3);

}

1. **update()**

void ofApp::setup(){

xPos = 5; // y축기준으로 시작할 좌표

ofBackground(ofColor::black); //black으로 배경 설정

}

void ofApp::update(){

xPos += 2; //x축으로 2칸 이동

if(ofGetWidth()<xPos){ //x좌표가 스크린 너비를 벗어나면

xPos = 5; // 초기화

}

}

void ofApp::draw(){

ofSetColor(ofColor::red); // 빨간색 지정

ofDrawCircle(xPos, 100, 10); // x로 xPos만큼, y로 100만큼 이동한 점에서 반지름 10인 원 출력

}