6주차 결과보고서

전공: 기계공학과 학년: 3학년 학번: 20191820 이름: 김형준

**1. Basic Shapes (1)**

ofBackground(0); : 배경색의 색상을 검정색(0)으로 설정한다. (숫자 1개만 입력시 grayscale의 값으로 입력됨, n입력시 #nnnnnn의 색상으로 설정)

ofSetColor(255); : 그릴 도형의 색상을 흰색(255)으로 설정한다.

ofDrawRectangle(50, 50, 100, 100); : 왼쪽 위 꼭짓점의 좌표가 (50,50)이고, 가로와 세로 길이가 각각 100, 100인 사각형을 그린다.

ofDrawCircle(250, 100, 50); : 중심점이 (250,100)이고, 반지름의 길이가 50인 원을 그린다.

ofDrawEllipse(400, 100, 80, 100); : 중심점이 (400,100)이고, 가로 길이가 80, 세로 길이가 100인 타원을 그린다.

ofDrawTriangle(500, 150, 550, 50, 600, 150); : 세 꼭짓점의 좌표가 각각 (500,150), (550,50), (600,150) 인 삼각형을 그린다.

ofDrawLine(700, 50, 700, 150); : 두 점 (700,50), (700,150)을 잇는 직선을 그린다.

**2. Basic Shapes (2)**

ofBackground(0); : 배경색의 색상을 검정색(0)으로 설정한다.

ofSetColor(255); : 그릴 도형의 색상을 흰색(255)으로 설정한다.

ofFill(); : 도형을 그릴 때 도형 내부를 채우게 변경한다. (ofNoFill() 호출 전까지 유지됨)

ofDrawRectangle(50, 50, 100, 100); : 왼쪽 위 꼭짓점의 좌표가 (50,50)이고, 가로와 세로 길이가 각각 100, 100인 사각형을 그린다.

ofDrawCircle(250, 100, 50); : 중심점이 (250,100)이고, 반지름의 길이가 50인 원을 그린다.

ofDrawEllipse(400, 100, 80, 100); : 중심점이 (400,100)이고, 가로 길이가 80, 세로 길이가 100인 타원을 그린다.

ofDrawTriangle(500, 150, 550, 50, 600, 150); : 세 꼭짓점의 좌표가 각각 (500,150), (550,50), (600,150) 인 삼각형을 그린다.

ofDrawLine(700, 50, 700, 150); : 두 점 (700,50), (700,150)을 잇는 직선을 그린다.

ofNoFill(); : 도형을 그릴 때 도형의 테두리만 그리게 변경한다. (ofFill() 호출 전까지 유지됨)

ofDrawRectangle(50, 200, 100, 100); : 왼쪽 위 꼭짓점의 좌표가 (50,200)이고, 가로와 세로 길이가 각각 100, 100인 사각형을 그린다.

ofDrawCircle(250, 250, 50); : 중심점이 (250,250)이고, 반지름의 길이가 50인 원을 그린다.

ofDrawEllipse(400, 250, 80, 100); : 중심점이 (400,250)이고, 가로 길이가 80, 세로 길이가 100인 타원을 그린다.

ofDrawTriangle(500, 300, 550, 200, 600, 300); : 세 꼭짓점의 좌표가 각각 (500,300), (550,200), (600,300) 인 삼각형을 그린다.

ofDrawLine(700, 200, 700, 300); : 두 점 (700,200), (700,300)을 잇는 직선을 그린다.

**3. Single rectangle brush**

void ofApp::setup(){ // setup : 프로그램 시작시 한 번만 실행되는 함수

ofSetBackgroundAuto(false); // 배경을 설정한 배경색으로 자동 초기화 하는 것을 끔

ofBackground(0); // 배경색을 검정색(0)으로 설정함

}

void ofApp::draw(){ // draw : 프로그램 실행시 반복 호출되는 함수 (drawing 함수전용)

// 마우스 왼쪽버튼을 눌렀을 때 색상이 (50~255)인 무채색 50x50 크기 사각형을 마우스 커서 중심위치에서 그림

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) { // 왼쪽 버튼을 클릭했을 때 True

float randomColor = ofRandom(50, 255); // grayscale 50~255 중 랜덤 색상값을 randomColor 변수에 저장함 (사각형을 칠할 색상)

ofSetColor(randomColor); // randomColor의 변수값으로 그릴 도형의 색상을 변경한다.

ofSetRectMode(OF\_RECTMODE\_CENTER); // 사각형을 그릴 때 입력 받은 좌표를 사각형의 중심으로 입력 받게 설정한다.

ofDrawRectangle(ofGetMouseX(),ofGetMouseY(), 50, 50); // 사각형의 중심점의 좌표가 각각 마우스 커서의 x, y좌표인 50 x 50 사각형을 그린다.

}}

**4. Bursting rectangle brush**

void ofApp::setup(){ // setup : 프로그램 시작시 한 번만 실행되는 함수

ofSetFrameRate(60); // 초당 60번 화면을 갱신하게 설정한다.

ofSetBackgroundAuto(false); // 배경을 설정한 배경색으로 자동 초기화 하는 것을 끔

ofBackground(0); // 배경색을 검정색(0)으로 설정함

}

void ofApp::draw(){ // draw : 프로그램 실행시 반복 호출되는 함수 (drawing 함수전용)

// 마우스 왼쪽버튼을 눌렀을 때 색상이 (50~255)인 무채색인 5~20x5~20 크기의 사각형을 마우스 커서 중심위치에서 (-40~40,-40~40)만큼 떨어진 위치에 그림 (클릭 1회당 10개씩 그림)

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) { // 왼쪽 버튼을 클릭했을 때 True

ofSetRectMode(OF\_RECTMODE\_CENTER); // 사각형을 그릴 때 입력 받은 좌표를 사각형의 중심으로 입력 받게 설정한다.

int numRects = 10; // numRects 변수에 10을 대입한다. (한번 클릭시 그릴 사각형의 개수)

for (int r = 0; r < numRects; r++) { // r = 0~numRects-1(9)까지 반복한다.

ofSetColor(ofRandom(50, 255)); // grayscale 50~255 중 랜덤 색상값을 randomColor 변수에 저장한다. (사각형을 칠할 색상)

float width = ofRandom(5, 20); // 5~20중 랜덤으로 뽑은 값을 변수 width에 저장한다. (사각형의 가로크기)

float height = ofRandom(5, 20); // 5~20중 랜덤으로 뽑은 값을 변수 height에 저장한다. (사각형의 세로크기)

float xOffset = ofRandom(-40, 40); // -40~40중 랜덤으로 뽑은 값을 변수 xOffset에 저장한다. (사각형을 그릴 위치의 x좌표 편차)

float yOffset = ofRandom(-40, 40); // -40~40중 랜덤으로 뽑은 값을 변수 yOffset에 저장한다. (사각형을 그릴 위치의 y좌표 편차)

ofDrawRectangle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset,width, height); // 중심좌표가 (마우스 커서 x좌표 + xOffset(-40~40), 마우스 커서 y좌표 + yOffset(-40~40))이고, 가로 세로 크기가 각각 width(5~20), height(5~20)인 사각형을 그린다.

}}}

**5. glowing circle brush**

void ofApp::setup(){ // setup : 프로그램 시작시 한 번만 실행되는 함수

ofSetFrameRate(60); // 초당 60번 화면을 갱신하게 설정한다.

ofSetBackgroundAuto(false); // 배경을 설정한 배경색으로 자동 초기화 하는 것을 끔

ofBackground(0); // 배경색을 검정색(0)으로 설정함

}

void ofApp::draw(){ // draw : 프로그램 실행시 반복 호출되는 함수 (drawing 함수전용)

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) { // 왼쪽 버튼을 클릭했을 때 True

int maxRadius = 100; // 변수 maxRadius에 100을 대입함 (그릴 원의 최대 반지름 길이)

int radiusStepSize = 5; // 변수 radiusStepSize에 5을 대입함 (반복문에서 한 Step당 줄일 길이)

int alpha = 3; 변수 alpha에 3을 대입함 (그릴 원의 투명도, 0일수록 투명한 값)

int maxOffsetDistance = 100; // 변수 maxOffsetDistance에 100을 대입함 (원이 마우스 커서로부터 떨어져서 그려질 거리의 최대값)

for (int radius = maxRadius; radius > 0; radius -= radiusStepSize) { // radius=maxRadius부터 0이하가 될때까지 radiusStepSize를 뺴면서 반복함 (100~5까지 총 20회 반복)

float angle = ofRandom(ofDegToRad(360.0)); // 0~2\*pi중 랜덤으로 뽑은 값을 변수 angle에 저장한다. (원이 마우스 커서로부터 떨어질 방향)

float distance = ofRandom(maxOffsetDistance); // 0~maxOffsetDistance(100)중 랜덤으로 뽑은 값을 변수 distance에 저장한다. (원이 마우스 커서로부터 떨어져서 그려질 거리)

float xOffset = cos(angle) \* distance; // xOffset에 (distance,angle)을 직교좌표계 x좌표로 변환한 값을 저장한다.

float yOffset = sin(angle) \* distance; // yOffset에 (distance,angle)을 직교좌표계 y좌표로 변환한 값을 저장한다.

ofColor myOrange(255, 132, 0, alpha); // ofColor 클래스타입 myOrange 변수를 Red = 255, Green = 132, Blue = 0, Alpha = alpha(3)으로 초기화 하면서 생성한다.

ofColor myRed(255, 6, 0, alpha); // ofColor 클래스타입 myRed 변수를 Red = 255, Green = 6, Blue = 0, Alpha = alpha(3)으로 초기화 하면서 생성한다.

ofColor inBetween = myOrange.getLerped(myRed, ofRandom(1.0)); // myOrange와 myRed 사이를 랜덤한(0~1.0) 비율로 내분한 지점의 색상값을 ofColor 클래스타입 변수 inBetween에 저장한다. (그려질 원의 색상값)

ofSetColor(inBetween); // inBetween에 저장된 색상값으로 그릴 도형의 색상을 설정한다.

ofDrawCircle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, radius); // 중심좌표가 (마우스 커서 x좌표 + xOffset, 마우스 커서 y좌표 + yOffset)이고, 반지름이 radius인 원을 그린다.

}}

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_RIGHT)) { // 오른쪽 버튼을 클릭했을 때 True

ofBackground(0); // 배경색을 검정색(0)으로 설정함 (화면을 검정색으로 모두 초기화함)

}}

**6. Fleeing triangle brush**

void ofApp::setup(){ // setup : 프로그램 시작시 한 번만 실행되는 함수

ofSetWindowShape(1024, 768); // 실행창의 크기를 1024x768로 설정한다.

ofSetFrameRate(60); // 초당 60번 화면을 갱신하게 설정한다.

ofSetBackgroundAuto(false); // 배경을 설정한 배경색으로 자동 초기화 하는 것을 끔

ofBackground(0); // 배경색을 검정색(0)으로 설정함

}

void ofApp::draw(){ // draw : 프로그램 실행시 반복 호출되는 함수 (drawing 함수전용)

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) { // 왼쪽 버튼을 클릭했을 때 True

int numTriangles = 10; // 변수 numTriangles에 10을 대입함 (그릴 삼각형의 개수)

int minOffset = 5; // 변수 minOffset에 5를 대입함 (마우스 중심으로부터 떨어질 최소 거리)

int maxOffset = 70; // 변수 maxOffset에 70을 대입함 (마우스 중심으로부터 떨어질 최대 거리)

int alpha = 150; // 변수 alpha에 150을 대입함 (그릴 삼각형의 투명도 값)

for (int t = 0; t < numTriangles; ++t) { // t=0~numTriangles-1(9)까지 반복함

float offsetDistance = ofRandom(minOffset, maxOffset); // 변수 offsetDistance에 minOffset~maxOffset (5~70) 중 랜덤으로 뽑은 값을 저장함 (마우스 중심으로부터 떨어질 거리)

ofVec2f mousePos(ofGetMouseX(), ofGetMouseY()); // ofVec2f 클래스 타입 변수 mousePos에 마우스 x좌표, 마우스 y좌표로 초기화 하면서 생성한다. (마우스 포인터 위치를 나타내는 점)

ofVec2f p1(0, 6.25); // ofVec2f 클래스 타입 변수 p1에 0, 6.25로 초기화 하면서 생성한다.

ofVec2f p2(25, 0); // ofVec2f 클래스 타입 변수 p2에 25, 0으로 초기화 하면서 생성한다.

ofVec2f p3(0, -6.25); // ofVec2f 클래스 타입 변수 p3에 0, -6.25로 초기화 하면서 생성한다.

float rotation = ofRandom(360); // 0~360중 랜덤값을 rotation 변수에 저장한다. (그릴 삼각형을 회전시킬 각도)

p1.rotate(rotation); // p1을 rotation도 만큼 회전시킨다.

p2.rotate(rotation); // p2을 rotation도 만큼 회전시킨다.

p3.rotate(rotation); // p3을 rotation도 만큼 회전시킨다.

ofVec2f triangleOffset(offsetDistance, 0.0); // ofVec2f 클래스 타입 변수 triangleOffset에 offsetDistance, 0.0으로 초기화 하면서 생성한다.

triangleOffset.rotate(rotation); // triangleOffset을 rotation도 만큼 회전시킨다.

p1 += mousePos + triangleOffset; // p1에 mousePos+triangleOffset을 더한다.

p2 += mousePos + triangleOffset; // p2에 mousePos+triangleOffset을 더한다.

p3 += mousePos + triangleOffset; // p3에 mousePos+triangleOffset을 더한다.

ofColor aqua(0, 252, 255, alpha); // Red=0, Green=252, Blue=255 Alpha=alpha(150)의 색상값을 ofColor 클래스타입 aqua변수에 저장함

ofColor purple(198, 0, 205, alpha); // Red=198, Green=0, Blue=205, Alpha=alpha(150)의 색상값을 ofColor 클래스타입 purple변수에 저장함

ofColor inbetween = aqua.getLerped(purple, ofRandom(1.0)); // aqua와 purple 사이를 랜덤한(0~1.0) 비율로 내분한 지점의 색상값을 ofColor 클래스타입 변수 inbetween에 저장한다. (그려질 삼각형의 색상값)

ofSetColor(inbetween); // inbetween에 저장된 색상값으로 그릴 도형의 색상을 설정한다.

ofDrawTriangle(p1, p2, p3); // 점 p1, p2, p3을 꼭짓점으로 갖는 삼각형을 그림

}}

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_RIGHT)) { // 오른쪽 버튼을 클릭했을 때 True

ofBackground(0); // 배경색을 검정색(0)으로 설정함 (화면을 검정색으로 모두 초기화함)

}}

**7. update()**

void ofApp::setup() { // setup : 프로그램 시작시 한 번만 실행되는 함수

xPos = 5; // 멤버변수 xPos의 값을 5로 초기화한다.

ofBackground(ofColor::black); // 배경색을 검정색(ofColor::black)으로 설정함

}

void ofApp::update() { // update : 프로그램 실행시 반복 호출되는 함수 (non-drawing 함수전용)

xPos += 2; // 매 프레임마다 멤버변수 xPos의 값을 2 증가시킨다. (원을 우측으로 2px 이동)

if (ofGetWidth() < xPos) { // xPos의 값이 화면 가로 크기보다 큰 경우 (원의 가로 좌표가 화면 오른쪽 바깥으로 넘어간 경우) True

xPos = 5; // 멤버변수 xPos의 값을 5로 초기화한다.

}}

void ofApp::draw() { // draw : 프로그램 실행시 반복 호출되는 함수 (drawing 함수전용)

ofSetColor(ofColor::red); // 그릴 도형의 색상을 빨강색(ofColor::red)으로 설정함

ofDrawCircle(xPos, 100, 10); // 중심점의 좌표가 (xPos,100)이고 반지름의 크기가 10인 원을 그린다.

}