6주차 결과보고서

전공 : 컴퓨터공학과 학년 : 2학년 학번 : 20211558 이름 : 윤준서

**1. Basic shape - 1**

**ofBackground(0);** 스크린의 배경을 모두 검은색으로 채운다. (0 : Black)

**ofSetColor(255);** Draw 색을 하얀색으로 지정한다. (255 : White)

**ofDrawRectangle(50, 50, 100, 100);** 한 꼭짓점 위치가 (50, 50), 반대편 꼭짓점 위치가 (100, 100) 인 직사각형을 Draw.

**ofDrawCircle(250, 100, 50);** 중심 위치가 (250, 100) 이고 반지름의 길이가 50인 원을 Draw.

**ofDrawEllipse(400, 100, 80, 100);** 중심 위치가 (400, 100) 이고 폭의 길이가 80, 높이가 100인 타원을 Draw.

**ofDrawTriangle(500, 150, 550, 50, 600, 150);** 세 꼭짓점의 위치가 각각 (500, 150), (550, 50), (600, 150) 인 삼각형을 Draw.

**ofDrawLine(700, 50, 700, 150);** (700, 50) 부터 (700, 150) 까지 직선을 Draw.

**2. Basic shape - 2**

**ofFill();** draw 한 내부를 모두 같은 색으로 채운다.

**ofNoFill();** draw 한 테두리를 제외하곤 색을 채우지 않는다.

**ofDrawRectangle(50, 200, 100, 100); ---------------------------------------------------**

**ofDrawCircle(250, 250, 50);** ofNoFill() 다음의 Draw 함수들은 모두 그리는

**ofDrawEllipse(400, 250, 80, 100);** 방식은 Basic shape - 1 과 같지만 테두리를

**ofDrawTriangle(500, 300, 550, 200, 600, 300);** 제외한 안쪽 공간은 모두 색을 칠하지 않는다.

**ofDrawLine(700, 200, 700, 300); ---------------------------------------------------**

**3. Single rectangle brush**

**(1)**

**if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {** 마우스 왼쪽 버튼을 눌렀는지 판단하 는 함수다. 눌렀을 경우 진행한다.

**ofSetColor(255);** Draw 색을 하얀색으로 지정한다.

**ofSetRectMode(OF\_RECTMODE\_CENTER);**

사각형 Draw 모드를 설정하는 함수다. 이때 위치 좌표를 중심점으로 지정한다.

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

중심점을 마우스의 위치로 설정, 이를 기준으로 가로, 새로 길이가 50인 직사각형을 Draw.

**}**

**(2)**

**if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {** 마우스 왼쪽 버튼을 눌렀는지 판단하 는 함수다. 눌렀을 경우 진행한다.

**float randomColor = ofRandom(50, 255);** 색을 50(밝은 회색)부터 하얀색으로 무작위로 지정하는 변수.

**ofSetColor(randomColor);** Draw 색을 randomColor 변수로 지정한다.

**ofSetRectMode(OF\_RECTMODE\_CENTER);**

사각형 Draw 모드를 설정하는 함수다. 이때 위치 좌표를 중심점으로 지정한다.

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

중심점을 마우스의 위치로 설정, 이를 기준으로 가로, 새로 길이가 50인 직사각형을 Draw.

}

**ofSetBackgroundAuto(false);** setup 함수에서 작성한다. 마우스 클릭 후 모습을 유지시킨다.

**ofBackground(0);** 유지 외의 배경의 색을 모두 검은색으로 채운다.

**4. Bursting rectangle brush**

**if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {** 마우스 왼쪽 버튼을 눌렀는지 판단하 는 함수다. 눌렀을 경우 진행한다.

**ofSetRectMode(OF\_RECTMODE\_CENTER);**

사각형 Draw 모드를 설정하는 함수다. 이때 위치 좌표를 중심점으로 지정한다.

**int numRects = 10;**

**for (int r=0; r<numRects; r++) {** 마우스를 클릭 시 10개의 사각형을 Draw.

**ofSetColor(ofRandom(50, 255));** Draw 색을 밝은 회색부터 하얀색까지 무작위 로 지정한다.

**float width = ofRandom(5, 20);** 폭을 5에서부터 20까지 무작위로 지정한다.

**float height = ofRandom(5, 20);** 높이를 5에서부터 20까지 무작위로 지정한다.

**float xOffset = ofRandom(-40, 40);** 사각형의 가로 중심을 마우스에서 -40 ~ 40 만큼 떨어진 지점으로 무작위로 지정한다.

**float yOffset = ofRandom(-40, 40);** 사각형의 새로 중심을 마우스에서 -40 ~ 40 만큼 떨어진 지점으로 무작위로 지정한다.

**ofDrawRectangle(ofGetMouseX()+xOffset, ofGetMouseY()+yOffset, width, height);** 중심이 (xOffset, yOffset) 이고 폭과 높이가 width 와 height 인 직사각형을 Draw.

**}**

**}**

**ofSetFrameRate(60);** setup 함수에서 작성한다. 1초당 프레임의 개수를 설정한다. 크기가 클 수록 Draw 속도가 증가한다. 즉 마우스를 지속해서 누르고 있을 시 누 른 것으로 인식하는 수가 증가한다.

**5. Glowing circle brush**

**if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {** 마우스 왼쪽 버튼을 눌렀는지 판단하 는 함수다. 눌렀을 경우 진행한다.

**int maxRadius = 100;** Draw 시 반지름의 최대 반지름을 100으로 설정한다.

**int radiusStepSize = 5;** 반지름의 길이를 감소시킬 때 사용하는 변수.

**int alpha = 3;** Draw 의 투명도를 나타낸다. 클 수록 불투명하다.

**int maxOffsetDistance = 100;** Draw 하는 원들의 간격을 최대 100으로 설정한다.

**for (int radius=maxRadius; radius>0; radius-=radiusStepSize) {**

반지름의 길이가 100부터 시작해서 5씩 감소하는 원을 Draw.

**float angle = ofRandom(ofDegToRad(360.0));** 각도를 라디안 단위로 한다. 0부터 최 대 360'까지로 설정한다.

**float distance = ofRandom(maxOffsetDistance);** Draw 하는 원들의 간격을 0부터 최 대 100까지로 설정한다.

**float xOffset = cos(angle) \* distance;**

**float yOffset = sin(angle) \* distance;** Draw 하는 원의 중심을 마우스를 중심으로 하고 반지름이 distance 인 원의 내부에서 무 작위로 설정한다.

**ofColor myOrange(255, 132, 0, alpha);** 색이 (255, 132, 0) 이고 투명도가 3인 색 변수.

**ofColor myRed(255, 6, 0, alpha);** 색이 (255, 6, 0) 이고 투명도가 3인 색 변수.

**ofColor inBetween = myOrange.getLerped(myRed, ofRandom(1.0));**

myOrange 색과 myRed 색의 중간값을 가지는 색 변수. 이때 ofRandom 으로 두 색 사 이의 보간값을 0부터 1까지 무작위로 지정한다.

**ofSetColor(inBetween);** Draw 색을 inBetween 변수로 지정한다.

**ofDrawCircle(ofGetMouseX()+xOffset, ofGetMouseY()+yOffset, radius);**

**}** 중심이 (xOffset, yOffset) 이고 반지름이 radius 인 원을 Draw.

**}**

**6. Fleeing triangle brush**

**if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {** 마우스의 왼쪽 버튼을 눌렀는지 판단 하는 함수다. 눌렀을 경우 진행한다.

**int numTriangles = 10;** Draw 하는 삼각형의 개수.

**int minOffset = 5;** offsetDistance 의 최솟값.

**int maxOffset = 70;** offsetDistance 의 최댓값.

**int alpha = 150;** Draw 의 투명도를 나타낸다. 클 수록 불투명하다.

**for (int t=0; t<numTriangles; ++t) {** 마우스를 클릭 시 10개의 삼각형을 Draw.

**float offsetDistance = ofRandom(minOffset, maxOffset);**

offsetDistance 의 크기를5부터 70까지 무작위로 지정한다.

**ofVec2f mousePos(ofGetMouseX(), ofGetMouseY());** 마우스의 위치를 나타내는 2D 벡터다.  **ofVec2f p1(0, 6.25);** 세 꼭짓점의 위치가 마우스의 위치를 (0, 0) 으로 기준을 잡은

**ofVec2f p2(25, 0);** 거리만큼 떨어져 있는 삼각형을 정의한다. 이 삼각형은 오른쪽

**ofVec2f p3(0, -6.25);** 을 향해 있다.

**float rotation = ofRandom(360);** 각도를 0' ~ 360' 로 무작위로 지정한다.

**p1.rotate(rotation);** 삼각형의 p1 꼭짓점 각도를 rotation 만큼 회전시킨다.

**p2.rotate(rotation);** 삼각형의 p2 꼭짓점 각도를 rotation 만큼 회전시킨다.

**p3.rotate(rotation);** 삼각형의 p3 꼭짓점 각도를 rotation 만큼 회전시킨다.

**ofVec2f triangleOffset(offsetDistance, 0.0);** offsetDistance 와 수평 방향을 가지 는 2D 벡터를 생성한다.

**triangleOffset.rotate(rotation);** 삼각형의 각 꼭짓점과 동일한 회전각도로 triangleOffset 을 회전한다.

**p1 += mousePos + triangleOffset;** 삼각형의 각 꼭짓점을

**p2 += mousePos + triangleOffset;** 마우스 위치 + triangleOffset 으로 이동시킨다.

**p3 += mousePos + triangleOffset;**

**ofColor aqua(0, 252, 255, alpha);** 색이 (255, 132, 0) 이고 투명도가 150인 색 변수.

**ofColor purple(198, 0, 205, alpha);** 색이 (255, 132, 0) 이고 투명도가 150인 색 변수.

**ofColor inbetween = aqua.getLerped(purple, ofRandom(1.0));**

aqua 색과 purple 색의 중간값을 가지는 색 변수. 이때 ofRandom 으로 두 색 사이의

보간값을 0부터 1까지 무작위로 지정한다.

**ofSetColor(inbetween);** Draw 색을 inbetween 변수로 지정한다.

**ofDrawTriangle(p1, p2, p3);** 세 꼭짓점의 위치가 각각 p1, p2, p3 인 삼각형을 Draw.

**}**

**}**

**7. Update function**

**int xPos;** ofApp 헤더 파일에 포함되는 클래스 변수다.

**void ofApp::setup(){** ofApp cpp 파일에 작성한다.

**xPos = 5;**  시작지점의 가로 좌표를 지정한다.

**ofBackground(ofColor::black);** 스크린의 배경을 모두 검은색으로 채운다.

**}**

**void ofApp::update(){**

**xPos += 2;** 재시작지점을 처음 좌표 +(2, 0) 으로 지정한다.

**if(ofGetWidth()<xPos){** 재시작지점이 스크린을 초과한 경우

**xPos = 5;** 시작지점을 처음 값이었던 5로 reset 한다.

**}**

**}**

**void ofApp::draw(){**

**ofSetColor(ofColor::red);** Draw 색을 빨간색으로 지정한다.

**ofDrawCircle(xPos, 100, 10);** 중심이 (xPos, 100) 이고 반지름이 10인 원을 Draw.