­­9주차 예비보고서

전공: 아트앤테크놀로지학과 학년: 3학년 학번: 20201116 이름: 이수빈

결과보고서

1. Basic Shape(1)

A picture containing text, wheel, clipart, light

Description automatically generated

ofBackground(0); //배경화면의 색을 0 (검정색)으로 설정

ofSetColor(255); //색을 255 (하얀색)으로 설정

ofDrawRectangle(50, 50, 100, 100); //중심이(50,50)이고 가로/길이가 각각 100인 직사각형을 그림

ofDrawCircle(250, 100, 50);//중심이 (250,100)이고 반지름이 100인 원을 그림

ofDrawEllipse(400, 100, 80, 100);//중심이 (400,100)이고 가로 반지름이 80, 세로 반지름이 100인 타원을 그림

ofDrawTriangle(500, 150, 550, 50, 600, 150);//세 꼭지점이 각각 (500,150), (550,50),(600,150)인 삼각형을 그림

ofDrawLine(700, 50, 700, 150); //점 (700,50)과 (700,150)을 잇는 선분 그림

1. Basic Shape(2)

A picture containing background pattern

Description automatically generated

ofBackground(0); //배경화면의 색을 0 (검정색)으로 설정

ofSetColor(255); //색을 255 (하얀색)으로 설정

ofFill();//도형을 채움

ofDrawRectangle(50, 50, 100, 100); //중심이(50,50)이고 가로/길이가 각각 100인 직사각형을 그림

ofDrawCircle(250, 100, 50);//중심이 (250,100)이고 반지름이 100인 원을 그림

ofDrawEllipse(400, 100, 80, 100);//중심이 (400,100)이고 가로 반지름이 80, 세로 반지름이 100인 타원을 그림

ofDrawTriangle(500, 150, 550, 50, 600, 150);//세 꼭지점이 각각 (500,150), (550,50),(600,150)인 삼각형을 그림

ofDrawLine(700, 50, 700, 150); //점 (700,50)과 (700,150)을 잇는 선분 그림

ofNoFill(); //도형을 채우지 않음

ofDrawRectangle(50, 200, 100, 100);

ofDrawCircle(250, 250, 50);// 중심이 (250,250)이고 반지름이 100인 원을 그림);

ofDrawEllipse(400, 250, 80, 100);// 중심이 (400,250)이고 가로 반지름이 80, 세로 반지름이 100인 타원을 그림

ofDrawTriangle(500, 300, 550, 200, 600, 300); 세 꼭지점이 각각 (500,300), (550, 200),(600,300)인 삼각형을 그림

ofDrawLine(700, 200, 700, 300); //점 (700, 200)과 (700, 300)을 잇는 선분 그림

1. Single rectangle brush

A picture containing text, clipart

Description automatically generated

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {//마우스 왼쪽 버튼을 누른다면

ofSetColor(255); } //색을 검정색으로 설정

float randomColor = ofRandom(50, 255); //50~255 사이의 랜덤 한 색을 randomColor라고 정의

ofSetColor(randomColor); //randomColor로 색 지정

ofSetRectMode(OF\_RECTMODE\_CENTER);

ofDrawRectangle(ofGetMouseX(), ofGetMouseY(), 50, 50); //마우스의 위치에 크기가 50\*50인 직사각형 그리기

1. Bursting rectangle brush

A black and white image of a person's face

Description automatically generated with low confidence

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {//마우스 왼쪽 버튼을 누른다면

ofSetRectMode(OF\_RECTMODE\_CENTER);

int numRects = 10; //직사각현의 개수 설정

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

ofSetColor(ofRandom(50, 255)); // 50~255 사이의 랜덤한 색으로 색 지정

float width = ofRandom(5, 20); //넓이: 5-20 사이의 랜덤 값으로 설정

float height = ofRandom(5, 20); //높이: 5-20 사이의 랜덤 값으로 설정

float xOffset = ofRandom(-40, 40); //위치의 x 값을 -40~40 사이의 랜덤 값으로 설정

float yOffset = ofRandom(-40, 40); //위치의 y 값을 -40~40사이의 랜덤 값으로 설정

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

}

}

1. glowing circle brush

A picture containing blur

Description automatically generated

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

int maxRadius = 100; //가장 큰 원의 반지름을 100으로 설정

int radiusStepSize = 5; //감소하면서 그려갈 원의 반지름의 감소하는 값 설정

int alpha = 3; //

int maxOffsetDistance = 100; //원이 퍼지게 하기 위해 설정

for (int radius=maxRadius; radius>0; radius-=radiusStepSize) { //원의 반지름을 maxRadius 인 것부터 radiusStepSize만큼 줄여가며 그리기

float angle = ofRandom(ofDegToRad(360.0));

float distance = ofRandom(maxOffsetDistance); //마우스로부떠 떨어질 거리의 정도를 랜덤 값으로 maxOffsetDistance보다 작게 설정

float xOffset = cos(angle) \* distance; //그려질 x좌표 위치

float yOffset = sin(angle) \* distance; //그려질 y좌표 위치

ofColor myOrange(255, 132, 0, alpha); //(255,132,0)에 해당하는 색을 myOrange로 정의한다. 투명도= alpha

ofColor myRed(255, 6, 0, alpha); //(255, 6, 0)에 해당하는 색을 myRed로 정의한다. 투명도= alpha

ofColor inBetween = myOrange.getLerped(myRed, ofRandom(1.0)); //inBetween이라는 색을 myOrange와 Red의 사이값 중 랜덤하게 설정

ofSetColor(inBetween); //inBetween에 정의된 색으로 지정

ofDrawCircle(ofGetMouseX()+xOffset, ofGetMouseY()+yOffset, radius);} //위에서 설정한 값들을 토대로 원 그리기

}

1. Fleeing triangle brush

A picture containing background pattern

Description automatically generated

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

int numTriangles = 10; //한번에 그릴 삼각형의 개수

int minOffset = 5; //마우스로부터 거리의 떨어질 최소 값

int maxOffset = 70; //마우스로부터 거리의 떨어질 최대 값

int alpha = 150;

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

float offsetDistance = ofRandom(minOffset, maxOffset); //마우스로부터 떨어진 거리를 minOffset와 maxOffset 사이의 랜덤 값으로 지정

ofVec2f mousePos(ofGetMouseX(), ofGetMouseY()); //마우스의 위치 저장

ofVec2f p1(0, 6.25); //삼각형의 한 꼭짓점 p1백터를 지정

ofVec2f p2(25, 0); //삼각형의 한 꼭짓점 p2백터를 지정

ofVec2f p3(0, -6.25); //삼각형의 한 꼭짓점 p3백터를 지정

float rotation = ofRandom(360); // 삼각형이 돌아간 각도를 랜덤하게 설정

p1.rotate(rotation); //꼭짓점 p1을 rotation만큼 돌리기

p2.rotate(rotation); //꼭짓점 p2을 rotation만큼 돌리기

p3.rotate(rotation); //꼭짓점 p3을 rotation만큼 돌리기

ofVec2f triangleOffset(offsetDistance, 0.0);

triangleOffset.rotate(rotation);

p1 += mousePos + triangleOffset; //p1의 위치 = 마우스의 위치 + offset

p2 += mousePos + triangleOffset; //p2의 위치 = 마우스의 위치 + offset

p3 += mousePos + triangleOffset; //p3의 위치 = 마우스의 위치 + offset

ofColor aqua(0, 252, 255, alpha); //aqua색 지정

ofColor purple(198, 0, 205, alpha); //purple 색 지정

ofColor inbetween = aqua.getLerped(purple, ofRandom(1.0)); //aqua와 purple 사이의 색으로 랜덤하게 지정

ofSetColor(inbetween); //위에서 정의한 inbetween이라는 색으로 삼각형 색 지정

ofDrawTriangle(p1, p2, p3);} //p1, p2, p3를 세 꼭짓점으로 갖는 삼각형 그리기

}

1. update()

update()함수는 매 frame 당 한번씩 불리는 함수이다. 이 함수는 종료되기 전까지 무한히 호출된다. 이 함수는 만약 변수의 지속적인 업데이트가 필요시 또는, 다른 개체의 지속적 업데이트 필요시 사용된다.

강의 자료에 나온 update() 함수 설명은 아래와 같다.

void ofApp::update(){

xPos += 2; //xPos를 매 프레임당 2만큼 증가시키낟.

if(ofGetWidth()<xPos){ // 만약 xPos가 화면의 가로보다 크게 되면

xPos = 5; //xPos를 5로 리셋한다. }

   }