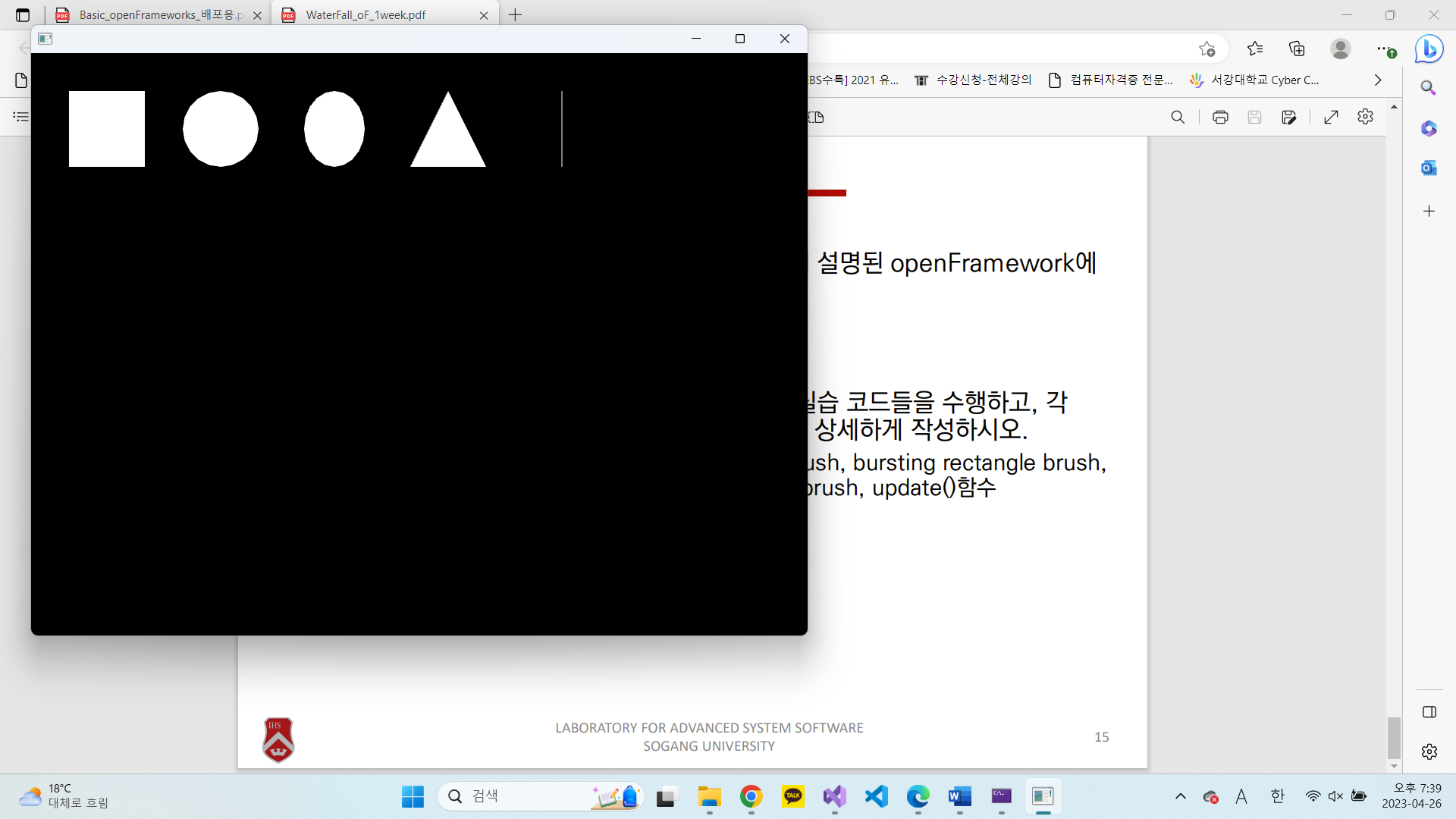
**6주차 결과보고서**

**전공: 경영학과 학년: 3학년 학번: 20190963 이름: 한다현**

**1.**

**Basic shape(1)**



ofBackground(0); //배경색을 설정하는 코드이다. 배경을 검정색으로 채운다.

ofSetColor(255); //그림의 색깔을 설정하는 코드이다. 앞으로 그려질 도형을 하얀색으로 그린다.

ofDrawRectangle(50, 50, 100, 100); //직사격형을 그리는 코드이다. 왼쪽 위 모서리의 좌표가 (50,50)이며 가로, 세로의 길이가 100인 직사각형을 그린다.

ofDrawCircle(250, 100, 50); //원을 그리는 코드이다. 원의 중심의 좌표가 (250, 100)이고 반지름이 50인 원을 그린다.

ofDrawEllipse(400, 100, 80, 100); //타원을 그리는 코드이다. 타원의 중심의 좌표가 (400, 100)이고, 가로의 길이가 80, 세로의 길이가 100인 타원을 그린다.

ofDrawTriangle(500, 150, 550, 50, 600, 150); //삼각형을 그리는 코드이다. 삼각형의 세 꼭지점의 좌표가 각각 (500,150), (550,50), (600,150)인 삼각형을 그린다.

ofDrawLine(700, 50, 700, 150); //라인을 그리는 코드이다. (700,50)에서 (700,150)까지의 선을 그린다.

**Basic shape(2)**

텍스트, 모니터, 전자제품, 스크린샷이(가) 표시된 사진

자동 생성된 설명

ofBackground(0);

ofSetColor(255);

ofFill(); //현재 그림 색깔로 그림을 채우는 코드이다. 각각의 도형들이 하얀색으로 채워져 있는 것을 볼 수 있다.

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(); //현재 그림 색깔로 그림을 채우지 않는 코드이다. 각각의 도형들이 하얀색으로 채워져 있지 않는 것을 볼 수 있다.

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);

나머지 코드는 Basic shape(1)과 동일한 코드들이다.

**Single Rectangle Brush**

텍스트, 스크린샷, 전자제품, 디스플레이이(가) 표시된 사진

자동 생성된 설명

ofSetBackgroundAuto(false); //배경이 자동으로 갱신되는 것을 결정하는 코드이다. false는 배경이 자동으로 갱신되지 않도록 설정한다.

ofBackground(0); //배경색을 black으로

위 두 줄은 ofApp.cpp의 setup()함수에 입력한다.

다음은 draw()함수에 입력할 코드이다.

if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) { //마우스 왼쪽 클릭이 발생했을 때 다음의 코드를 실행한다.

float randomColor = ofRandom(50, 255); //randomColor라는 변수에 50~255 중 하나의 실수를 랜덤으로 저장한다.

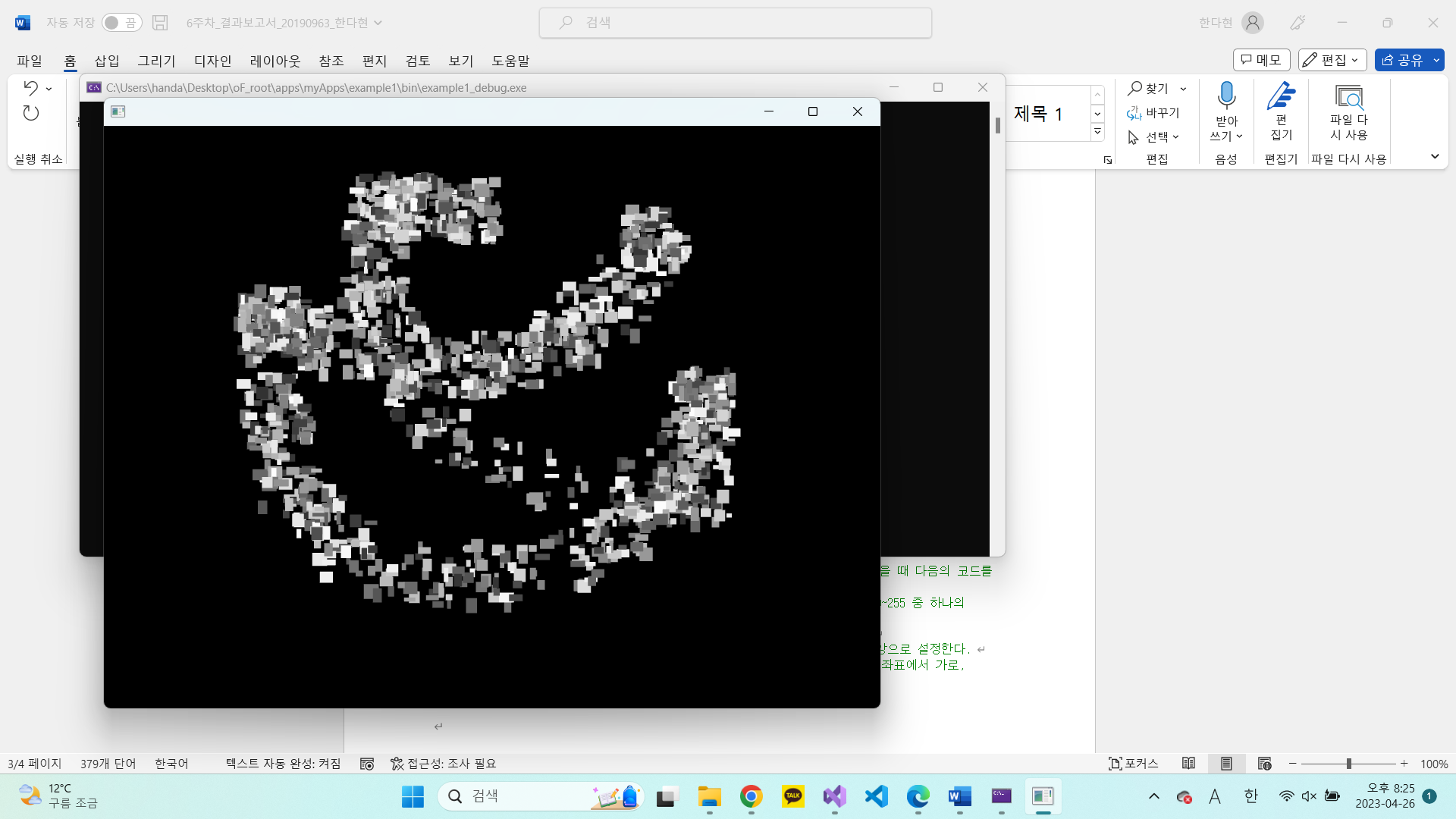
ofSetColor(randomColor); //randomcolor를 그림의 색깔로 설정한다.

ofSetRectMode(OF\_RECTMODE\_CENTER); //생성 위치를 왼쪽 상단에서 중앙으로 설정한다.

ofDrawRectangle(ofGetMouseX(), ofGetMouseY(), 50, 50); //마우스의 좌표에서 가로, 세로가 50인 직사각형을 그린다.

}

**Bursting Rectangle Brush**



setup() 함수에 다음의 코드를 추가한다. draw() 함수의 다음의 코드들을 작성한다.

ofSetFrameRate(60); //1초당 생성하는 프래임의 개수를 의미한다. 즉 1초에 60 프래임을 생성한다.

draw() 함수의 다음의 코드들을 작성한다.

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++) { //10번 반복

ofSetColor(ofRandom(50, 255));

float width = ofRandom(5, 20); //가로의 길이를 5와 20 사이의 랜덤의 실수로 설정

float height = ofRandom(5, 20); //세로의 길이를 5와 20 사이의 랜덤의 실수로 설정

float xOffset = ofRandom(-40, 40); //xOffset에 -40과 40 사이의 랜덤의 실수를 저장

float yOffset = ofRandom(-40, 40); //yOffset에 -40과 40 사이의 랜덤의 실수를 저장

ofDrawRectangle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, width, height); //마우스의 좌표에서 xOffset, yOffset만큼 떨어진 공간에 width, height 크기만큼의 직사각형을 그린다.

}

}

**glowing circle brush**

텍스트, 모니터, 실내, 전자제품이(가) 표시된 사진

자동 생성된 설명

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

int maxRadius = 100; //처음 radius 값

int radiusStepSize = 5; //radius가 감소하는 정도

int alpha = 3; // RGBA에서 불투명도를 나타내는 변수

int maxOffsetDistance = 100; // distance를 결정하는데 사용되는 변수

for (int radius = maxRadius; radius > 0; radius -= radiusStepSize) { //100에서 0보다 작아지기 전까지 5만큼 감소시키며 반복

float angle = ofRandom(ofDegToRad(360.0)); //360도를 라디안으로 변환한다

float distance = ofRandom(maxOffsetDistance); //0과 100 사이의 특정 숫자

float xOffset = cos(angle) \* distance; //xOffset의 값을 cos(angle) \* distance로 저장

float yOffset = sin(angle) \* distance; //yOffset의 값을 sin(angle) \* distance로 저장

ofColor myOrange(255, 132, 0, alpha); //myOrange의 값을 RGBA로 설정한다.

ofColor myRed(255, 6, 0, alpha); //myRed의 값을 RGBA로 설정한다.

ofColor inBetween = myOrange.getLerped(myRed, ofRandom(1.0)); //myOrange와 myRed를 혼합하여 새로운 색깔 inBetween을 설정한다.

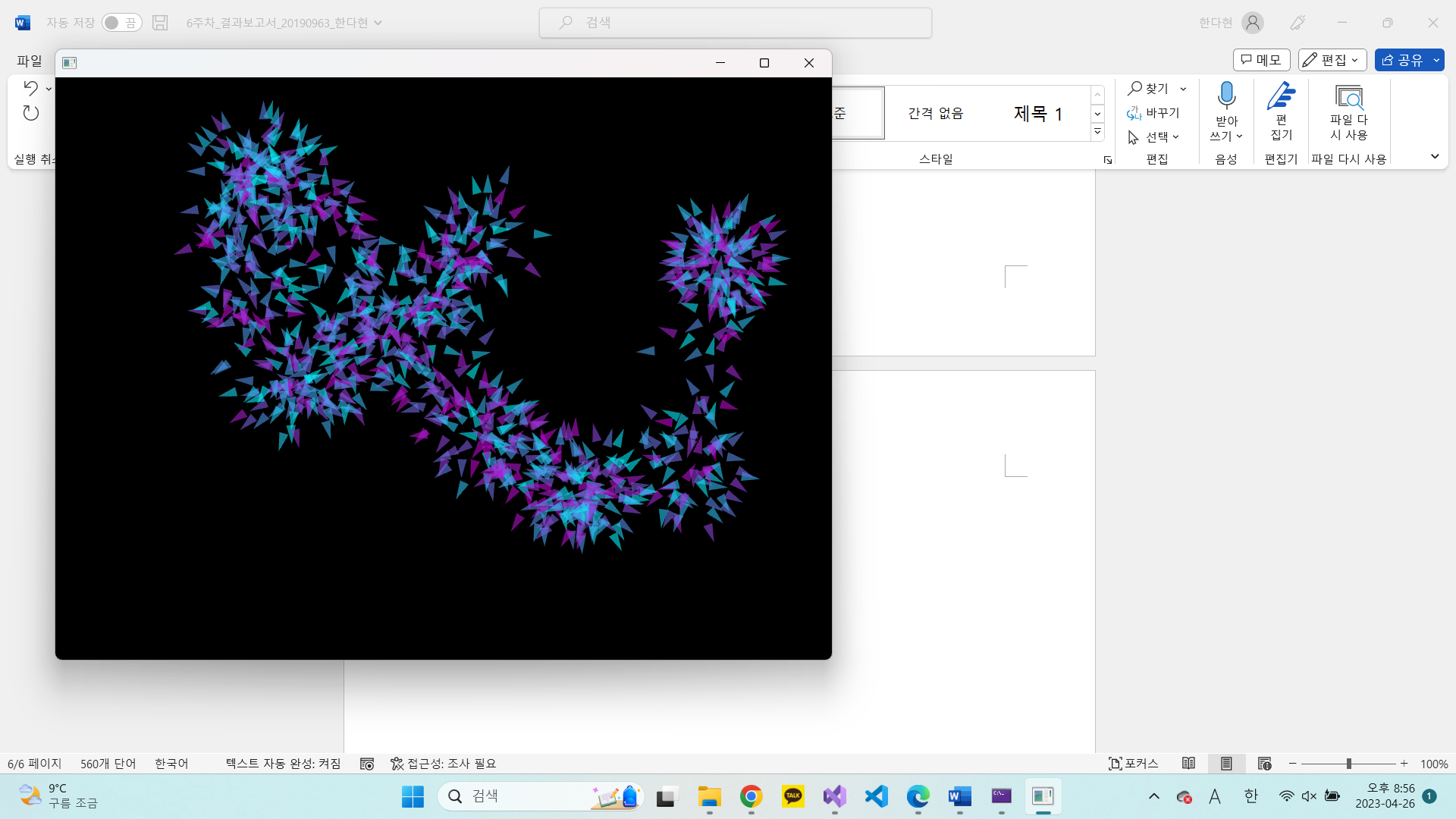
ofSetColor(inBetween); //inBetween으로 그림의 색깔을 정한다.

ofDrawCircle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, radius); //마우스의 좌표로부터 xOffset, yOffset만큼 떨어진 공간에 radius 크기의 원을 그린다.

}

}

**Fleeing Triangle Brush**



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

int numTriangles = 10;

int minOffset = 5;

int maxOffset = 70;

int alpha = 150;

for (int t = 0; t < numTriangles; ++t) { //10번 반복

float offsetDistance = ofRandom(minOffset, maxOffset);

ofVec2f mousePos(ofGetMouseX(), ofGetMouseY()); //mousePos의 좌표

// p1, p2, p3는 (0,0)이 기준인 삼각형의 세 꼭지점의 위치를 의미한다.

ofVec2f p1(0, 6.25);

ofVec2f p2(25, 0);

ofVec2f p3(0, -6.25);

float rotation = ofRandom(360); // The rotate function uses degrees!

p1.rotate(rotation); //p1을 rotation 각도만큼 회전

p2.rotate(rotation); //p2을 rotation 각도만큼 회전

p3.rotate(rotation); //p3을 rotation 각도만큼 회전

ofVec2f triangleOffset(offsetDistance, 0.0); //triagnleOffset vector의 값이 offsetDistance, 0.0

triangleOffset.rotate(rotation); //triangleOffset을 rotation 각도만큼 회전

p1 += mousePos + triangleOffset; //p1에 마우스 vector와 triangleOffset vector의 값을 더한다.

p2 += mousePos + triangleOffset; //p2에 마우스 vector와 triangleOffset vector의 값을 더한다.

p3 += mousePos + triangleOffset; //p3에 마우스 vector와 triangleOffset vector의 값을 더한다.

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

ofColor purple(198, 0, 205, alpha); //purple 색을 RGBA로 설정

ofColor inbetween = aqua.getLerped(purple, ofRandom(1.0)); //aqua와 purple을 혼합하여 inbetween이라는 색깔 정의

ofSetColor(inbetween); //그림의 색을 inbetween으로 설정

ofDrawTriangle(p1, p2, p3); //p1, p2, p3가 꼭지점인 삼각형을 그린다.

}

}

**update()**

int xPos; 코드를 ofApp.h 파일에 클래스의 멤버 변수로 추가한다.

텍스트, 모니터, 스크린샷, 전자제품이(가) 표시된 사진

자동 생성된 설명

void ofApp::setup(){

xPos = 5; // x 좌표를 5로 초기화

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

}

//--------------------------------------------------------------

void ofApp::update(){

xPos += 2; //xPos에 2를 더한다.

if (ofGetWidth() < xPos) { // xPos의 값이 창의 크기를 넘어서면

xPos = 5; // x좌표를 5로 초기화

}

}

//--------------------------------------------------------------

void ofApp::draw(){

ofSetColor(ofColor::red); // 그림의 색깔을 red로 설정

ofDrawCircle(xPos, 100, 10); // (xPos, 100) 좌표에서 반지름이 10인 원을 그린다.

}