6주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20212022 이름: 이예준

**1.**

**1. Basic shape(1)**

- ofBackground(0);

🡪 전체 화면을 grayscale color의 0에 해당하는 색으로 지우는 기능을 하며,

이때의 0은 검은색을 뜻한다.

- ofSetColor(255);

🡪 그림을 그릴 때, 흰색으로 그리도록 설정한다.

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

🡪 (50,50) 위치에 사각형의 좌 상단 모서리가 위치해 있고,

너비는 100, 높이도 100인 사각형을 그린다.

- ofDrawCircle(250, 100, 50);

🡪 (250,100) 위치에 원의 중심이 있고, 반지름이 50인 원을 그린다.

- ofDrawEllipse(400, 100, 80, 100);

🡪 (400,100) 위치에 타원의 중심이 있고, 타원의 너비는 80이고, 높이는 100이다.

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

🡪 꼭지점의 위치가 각각 (500, 150), (550, 50), (600, 150)에 있는 삼각형을 그린다.

- ofDrawLine(700, 50, 700, 150);

🡪 (700, 50)부터 (700, 150)까지 line을 그린다.

\*코드를 실행하면 아래의 그림과 같이 사각형, 원, 타원, 삼각형, 선이 그려진 것을

알 수 있다.

블랙, 디자인이(가) 표시된 사진

자동 생성된 설명

**2. Basic shape(2) – (Basic shape(1)에서 추가되는 함수)**

- ofFill();

🡪 도형 내부를 채우는 모드를 활성화하는 함수로

도형을 그리면 도형을 채우는데 사용된다.

- ofNoFill();

🡪 채우기 모드를 비활성화하는 함수로 도형의 선만 그리는 모드로 전환된다.

\*코드를 실행하면 아래의 그림과 같이 사각형, 원, 타원, 삼각형, 선을

채워진 그림과 선만 그린 그림으로 나타낸다.

그래픽, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**3. Single Rectangle Brush: Using the Mouse(1)**

-if(ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)){};

🡪 ofGetMousePressed 함수는 마우스의 왼쪽 버튼이 눌러져 있으면 True, 아니면 False를

반환하여, 만약 마우스 왼쪽 버튼이 눌리면 if문을 수행한다.

- ofSetColor(255);

🡪 그림을 그릴 때, 흰색으로 그리도록 설정한다.

- ofSetRectMode(OF\_RECTMODE\_CENTER);

🡪 사각형의 중심점을 기준으로 사각형의 위치가 지정되는 그리기 모드를 설정한다.

예를 들어 ofDrawRectangle() 함수로 사각형을 그릴 때, 함수의 인자로 전달되는 좌표가

사각형의 중심점이 된다.

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

🡪 마우스의 중심에 너비 50,높이 50 크기의 사각형을 그린다.

\*코드를 실행하고 마우스의 왼쪽 버튼을 누르면 마우스 커서를 중심으로 사각형이

그려지는 것을 볼 수 있다.

소프트웨어, 컴퓨터, 멀티미디어 소프트웨어, 텍스트이(가) 표시된 사진

자동 생성된 설명

**4. Single Rectangle Brush: Using the Mouse(2) – (Using the Mouse (1)에서 추가되는 함수)**

- ofSetBackgroundAuto(false);

🡪 새로운 프레임을 그릴 때 이전 프레임을 자동으로 지우는 것을 비활성화한다.

- ofBackground(0);

🡪 전체 화면을 검은색으로 지운다.

\*코드를 실행하고 마우스의 왼쪽 버튼을 누르고 움직이면 그림과 같이 사각형들이 그려진다.

그래픽, 픽셀, 디자인이(가) 표시된 사진

자동 생성된 설명

**5. Single Rectangle Brush: Using the Mouse(3) – (Using the Mouse (2)에서 수정사항)**

- float randomColor = ofRandom(50, 255);

🡪 50부터 255까지 무작위의 값을 randomColor에 저장한다.

- ofSetColor(randomColor);

🡪 그림을 그릴 때, randomColor 값에 해당하는 grayscale color로 그리도록 설정한다.

\*코드를 실행하고 마우스의 왼쪽 버튼을 누르고 움직이면 그림과 같이 다양한 색의 사각형들이

그려진다.

스케치, 그래픽, 디자인이(가) 표시된 사진

자동 생성된 설명

**6. Bursting Rectangle Brush: Creating Randomized Bursts(1)**

- if(ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {};

🡪 마우스의 왼쪽 버튼이 눌리면 if문을 수행한다.

- ofSetRectMode(OF\_RECTMODE\_CENTER);

🡪 사각형의 중심점을 기준으로 사각형의 위치를 지정한다.

- int numRects = 10;

🡪 사각형의 개수를 10개로 지정한다.

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

🡪 10번 for문을 수행한다.

- ofSetColor(ofRandom(50, 255));

🡪 그림을 그릴 때, 50부터 255까지 무작위의 값에 해당하는 grayscale color로 그린다.

- float width = ofRandom(5, 20);

🡪 5부터 20까지 무작위의 값을 width 에 저장한다.

- float height = ofRandom(5, 20);

🡪 5부터 20까지 무작위의 값을 height 에 저장한다.

- float xOffset = ofRandom(-40, 40);

🡪 -40부터 40까지 무작위의 값을 xOffset 에 저장한다.

- float yOffset = ofRandom(-40, 40);

🡪 -40부터 40까지 무작위의 값을 yOffset 에 저장한다.

-ofDrawRectangle(ofGetMouseX()+xOffset, ofGetMouseY()+yOffset, width, height);

🡪 마우스 위치로부터 xOffset 과 yOffset 만큼 떨어진 위치에 높이가 height, 너비가 width 인

사각형을 그린다.

- ofSetFrameRate(60);

🡪 프레임 속도를 초당 60프레임으로 설정한다.

\*코드를 실행하고 마우스 왼쪽 버튼을 눌러 움직이면 그림과 같이 사각형들이 그려진다.

픽셀이(가) 표시된 사진

중간 신뢰도로 자동 생성된 설명

**7. Glowing Circle Brush: Using Color**

- ofColor myOrange(255, 132, 0);

🡪 각 인자는 순서대로 RGB에 따라 r, g, b를 뜻하고, (255, 132, 0)로 RGB값을 설정한다.

- ofColor myBlue(0, 0, 255, 50);

🡪 r=0, g=0, b=255 으로 myBlue 을 설정하고 Alpha 값 a=50 로 투명도를 설정한다.

- ofColor myGreen(0, 0, 255, 255);

🡪 r=0, g=0, b=255 으로 myGreen 을 설정하고 Alpha 값 a=255 로 투명도를 설정한다.

**\*** *a가 255이면 완전한 불투명함을 뜻한다. 반대로 a가 0이면 완전히 투명하다는 뜻이다.*

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

🡪 myGreen의 r값을 출력한다.

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

🡪 myGreen의 g값을 출력한다.

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

🡪 myGreen의 b값을 출력한다.

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

🡪 myGreen의 a값을 출력한다.

- ofColor myYellow;

🡪 myYellow를 설정한다.

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

🡪 myYellow의 RGBA값을 따로 직접 설정해준다.

- ofColor myAqua = ofColor::aqua;

🡪 openframeworks에서 미리 정의된 색상을 제공받아 사용한다.

- ofColor myPurple = ofColor::plum;

🡪 openframeworks에서 미리 정의된 색상을 제공받아 사용한다.

\*임의의 색 ofColor myColor(157, 61, 219, 80)으로 설정하고 코드를 실행한 뒤

마우스 왼쪽 버튼을 누르고 움직이면 그림과 같이 그려진다.

다채로움이(가) 표시된 사진

자동 생성된 설명

**8. Fleeing Triangle Brush: Vectors**

- ofVec2f mousePos(ofGetMouseX(), ofGetMouseY());

🡪 현재 마우스의 x, y좌표를 이용하여 ofVec2f(2차원) 형식의 벡터를 만든다.

- ofVec2f p1(0, 25.0); ofVec2f p2(100, 0); ofVec2f p3(0, -25.0);

🡪 삼각형의 각 꼭지점을 정의한다. 이 도형은 (0, 0)을 기준으로 오른쪽을 향하는 삼각형이다.

- p1 += mousePos; p2 += mousePos; p3 += mousePos;

🡪 각 꼭지점을 마우스 위치만큼 이동시켜 삼각형이 마우스 위치를 중심으로 이동한다.

\* 코드를 실행하고, 마우스 왼쪽 버튼을 누르면서 이동하면 그림과 같이 그려진다.

프랙탈 아트이(가) 표시된 사진

자동 생성된 설명

**9. update()**

- void ofApp::setup(){

xPos = 5;

ofBackground(ofColor::black);

}

🡪 원의 가로 위치 xPos값을 5로 초기화하고, 배경은 검은색으로 설정한다.

- void ofApp::update(){

xPos += 2;

if(ofGetWidth()<xPos){

xPos = 5;

}

🡪 매 프레임마다 xPos값은 2씩 증가하고, 만약 원이 화면 오른쪽 끝을 벗어나면

다시 xPos값을 5로 초기화한다.

- void ofApp::draw(){

ofSetColor(ofColor::red);

ofDrawCircle(xPos, 100, 10);

}

🡪 도형의 색은 빨간색으로 설정하고, 가로 위치는 xPos값, 세로 위치는 100에

반지름 10인 도형을 그린다.

\*코드를 실행하면 빨간색원이 화면 왼쪽 끝에서 오른쪽 끝까지 간 다음

다시 왼쪽 끝에서 이동하는 과정을 반복한다.

어둠, 천문학이(가) 표시된 사진

자동 생성된 설명어둠이(가) 표시된 사진

자동 생성된 설명어둠이(가) 표시된 사진

자동 생성된 설명어둠, 블랙, 천문학이(가) 표시된 사진

자동 생성된 설명