

6주차 결과보고서

전공: 컴퓨터공학과

학년: 2학년

학번: 20231632

이름: Jumagul

Alua

main.cpp and ofApp.h functions

```
#include "ofMain.h"
#include "ofApp.h"

//=====
int main( ){

    //Use ofGLFWWindowSettings for more options like multi-monitor fullscreen
    ofGLWindowSettings settings;
    settings.setSize(1024, 768);
    settings.windowMode = OF_WINDOW; //can also be OF_FULLSCREEN

    auto window = ofCreateWindow(settings);

    ofRunApp(window, make_shared<ofApp>());
    ofRunMainLoop();
}
```

위에 있는 코드는 main 함수이다. 앞으로 설명할 예시들의 기반이며 openFrameworks 프로젝트의 실행 흐름을 정의한다. 먼저 ofGLWindowSettings settings에는 창의 크기와 윈도우 모드를 정의하는 인스턴스를 생성한다. 다음에 1024하고 768으로 창의 너비와 높이를 설정해준다. settings.windowMode = OF_WINDOW;을 통해 창을 윈도우 모드로 설정해주고 auto window = ofCreateWindow(settings)에는 앞에 사용한 설정을 가져서 openFrameworks 창을 생성한다. 다음 ofRunApp(window, make_shared<ofApp>()); 에서 생성한 창을 ofApp의 클래스와 연결하여 프로젝트를 실행한다. 마지막으로 ofRunMainLoop(); openFrameworks 루프는 상태를 살펴보고 그래픽 업데이트를 처리한다.

```
#pragma once

#include "ofMain.h"

class ofApp : public ofBaseApp{

public:
    void setup();
    void update();
    void draw();

    void keyPressed(int key);
    void keyReleased(int key);
    void mouseMoved(int x, int y );
    void mouseDragged(int x, int y, int button);
    void mousePressed(int x, int y, int button);
    void mouseReleased(int x, int y, int button);
    void mouseEntered(int x, int y);
    void mouseExited(int x, int y);
    void windowResized(int w, int h);
    void dragEvent(ofDragInfo dragInfo);
    void gotMessage(ofMessage msg);

};
```

OfApp.h 함수입니다. 각 줄의 설명:

setup(): 프로젝트를 초기화하고 프로그램이 시작될 때 한 번 호출된다.

update(): 프로젝트 업데이트하고 모든 프레임을 호출한다.

draw(): 프로젝트의 그래픽을 그리며 모든 프레임마다 호출한다.

keyPressed(int key): 키보드의 키를 누르면 호출된다.

keyRelease(int key): 키보드의 키를 놓으면 호출된다.

mouseMoved(int x, int y): 마우스를 이동할 때 호출된다.

mouseDragged(int x, int y, int button): 마우스 버튼을 누른 채로 움직일 때 호출된다.

mousePressed(int x, int y, int button): 마우스 버튼을 눌렀을 때 호출된다.

mouseReleased(int x, int y, int button): 마우스 버튼이 해제될 때 호출된다.

mouseEntered(int x, int y): 마우스가 프로그램 창에 들어갈 때 호출된다.

mouseExited(int x, int y): 마우스가 프로그램 창을 나갈 때 호출된다.

windowResized(int w, int h): 프로그램 창의 크기가 조정될 때 호출된다.

dragEvent(ofDragInfo dragInfo): 파일이 드래그 앤 드롭 이벤트가 발생할 때 호출된다.

gotMessage(ofMessage msg): 사용자 지정 메시지가 수신될 때 호출된다.

1. Basic shape(1)

```
//-----  
void ofApp::draw(){  
    ofBackground(0);  
    ofSetColor(255);  
    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);  
}
```

OfApp.cpp의 draw 함수에다가 코드를 추가해준다. 화면을 검은색으로 안을 그리는 색깔은 하얀색으로 설정해 주고 직사각형, 원, 타원, 삼각형, 선이 나타나기 위해 각자 위치와 너비, 높이를 설정해준다.

2. Basic shape(2)

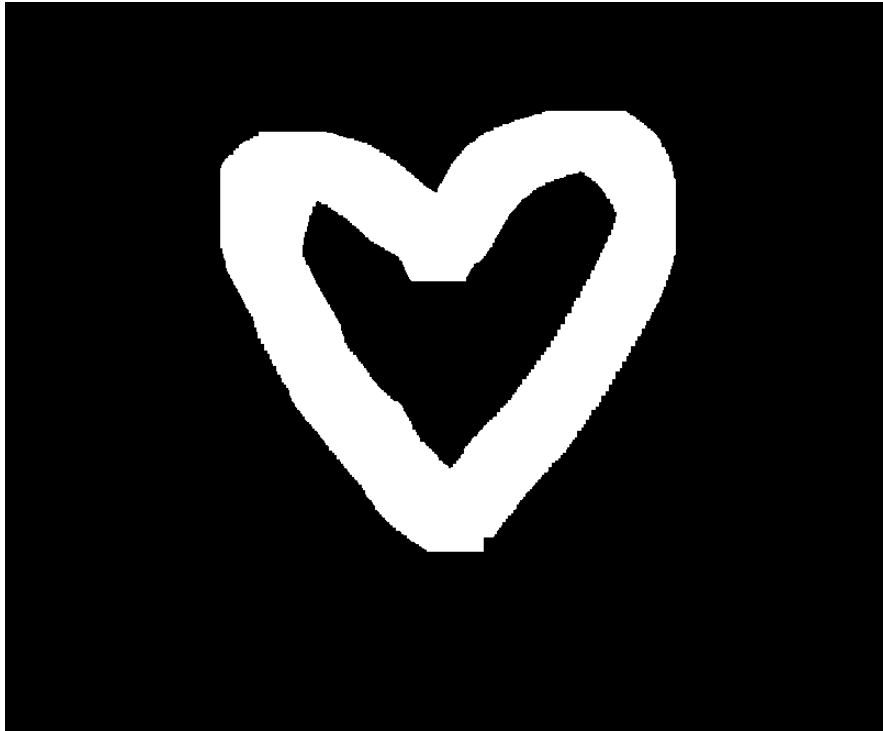
```
void ofApp::draw(){  
    ofBackground(0);  
    ofSetColor(255);  
    ofFill();  
    ofDrawRectangle(50, 50, 100, 100);  
    ofDrawCircle(250, 100, 50);  
    ofDrawEllipse(400, 100, 80, 100);  
    ofDrawTriangle(500, 150, 550, 50, 600, 150);  
    ofDrawLine(700, 50, 700, 150);  
    ofNoFill();  
    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);  
}
```

이 함수에는 `ofFill`과 `ofNoFill`으로 나뉘주며 도형 안을 채우고 안 채우는 두개의 설정이 돼 있다. `OfFill` 함수는 도형 안을 하얀 색으로 다 채울것이고 `ofFill` 함수를 포함 안 시키면 `ofNoFill` 함수만 설정되어 도양의 경계만 채울 것이다.

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

```
void ofApp::setup(){
    ...ofSetBackgroundAuto(false);
    > ofBackground(0);
}
//-----
void ofApp::update(){
}
//-----
void ofApp::draw(){
    > if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) {
    >     ofSetColor(255);
    >     ofSetRectMode(OF_RECTMODE_CENTER); ofDrawRectangle(ofGetMouseX(), ofGetMouseY(), 50, 50);
    > }
}
```

`of0App.cpp`에서 `setup`하고 `draw` 함수에 코드를 추가해준다. 여기서 `draw` 함수에는 만약에 마우스의 왼쪽 버튼을 눌렀으면 255으로 흰색을 정해주고 그리는 모양은 사각형으로 설정해준다. `Setup` 함수에는 그린 다음 자동 지우기 모드를 비활성화하면서 이와 같은 결과를 얻을 수 있게 된다.

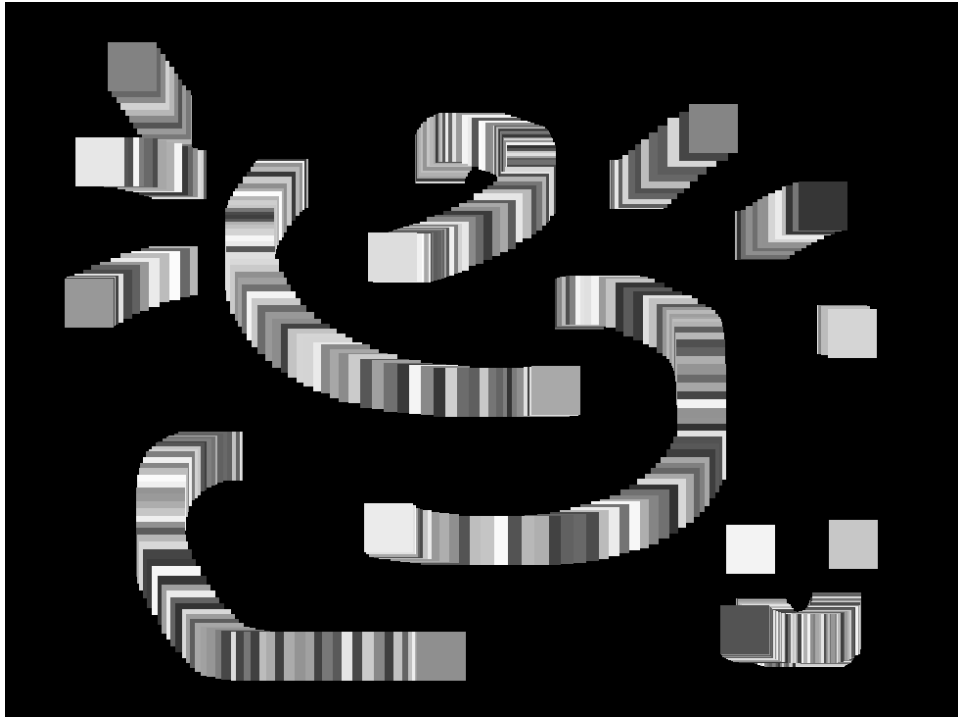


또한, draw 함수에는 이와 같이 정해준다면, : 50과 255 사이의 무작위 값을 생성하여 사각형으로 그리게 된다.

```
void ofApp::draw(){
    if (ofGetMousePressed(OFF_MOUSE_BUTTON_LEFT)) {
        float randomColor = ofRandom(50, 255);
        ofSetColor(randomColor);

        ofSetRectMode(OFF_RECTMODE_CENTER); ofDrawRectangle(ofGetMouseX(), ofGetMouseY(), 50, 50);
    }
}
```

아래와 같은 결과를 얻는다.



4. Single Rectangle Brush: Using the Mouse

```
void ofApp::setup(){
    ofSetBackgroundAuto(false);
    ofBackground(0);
    ofSetFrameRate(60);
}

//-----
void ofApp::update(){
}

//-----
void ofApp::draw(){
    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++) {
            ofSetColor(ofRandom(50, 255));
            float width = ofRandom(5, 20);
            float height = ofRandom(5, 20);
            float xOffset = ofRandom(-40, 40);
            float yOffset = ofRandom(-40, 40);
            ofDrawRectangle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, width, height);
        }
    }
}
```

여기서 ofSetFrameRate(60)라고 프레임 속도를 60으로 설정해준다. 그 다음 마우스 왼쪽 버튼을 눌렀을 때 그려질 사각형이 10개가 될 것이다. For loop에서는 색깔을 앞에 한 것처럼 50에서 255 중 무작위 그레이 색깔로 정해주고 사각형의 너비와 높이를 각각 5에서 20까지의 무작위 숫자를 가지도록 한다. 사각형의 위치도 x와 y의 대하여 -40에서 40까지의 무작위로 설정해주면 아래와

같은 결과가 나타난다.



5. Glowing Circle Brush: Using Color

```
void ofApp::draw() {
    if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) {
        int maxRadius = 100;
        int radiusStepSize = 5;
        int alpha = 3;
        int maxOffsetDistance = 100;
        for (int radius = maxRadius; radius > 0; radius -= radiusStepSize) {
            float angle = ofRandom(ofDegToRad(360.0));
            float distance = ofRandom(maxOffsetDistance);
            float xOffset = cos(angle) * distance;
            float yOffset = sin(angle) * distance;

            ofColor myOrange(255, 132, 0, alpha);
            ofColor myRed(255, 6, 0, alpha);
            ofColor inBetween = myOrange.getLerped(myRed, ofRandom(1.0));
            ofSetColor(inBetween);

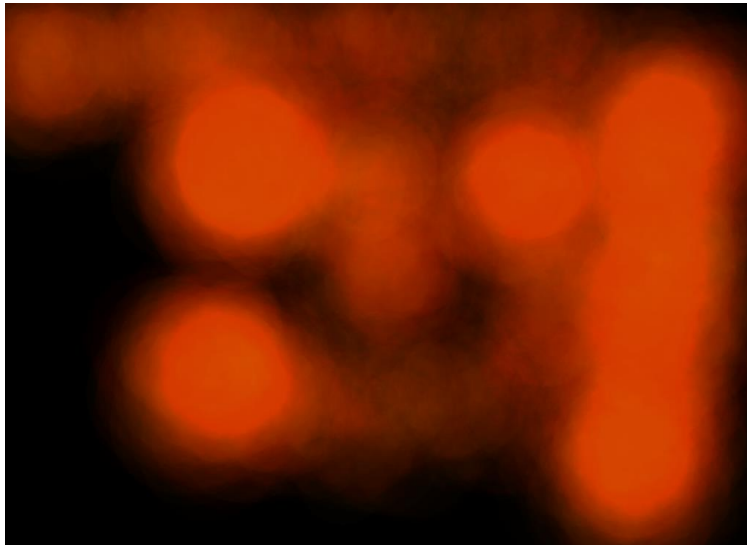
            ofDrawCircle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, radius);
        }
    }

    if (ofGetMousePressed(OF_MOUSE_BUTTON_RIGHT)) {
        ofBackground(0);
    }
}

//-----
void ofApp::keyPressed(int key) {
    if (key == 's') {
        glReadBuffer(GL_FRONT);
        ofSaveScreen("savedScreenshot_" + ofGetTimestampString() + ".png");
    }
}
```

이 예시에는 마우스 왼쪽 버튼이 눌릴 때 무작위 크기와 색상의 원을 마우스 위치 주변에 그리고, 마우스 오른쪽 버튼이 눌리면 화면의 그림을 지우는 프로그램을 구현한다. 또한, 키보드의 's' 키를 누르면 현재 화면을 스크린샷으로 저장한다. Alpha 투명도를 3으로 해주고, 여기서 for loop에는 각도를 0에서 360까지 무작위로 설정해주고 거리도 원의 중심에서 무작위로 설정한 다음 두

가지 생성한 색상 사이의 색깔로 원을 그려준다. 그리고 's' 키를 누를때의 화면 스크린샷을 하는 코드도 써준다. 아래에 나온 결과를 볼 수 있다.



6. Fleeing Triangle Brush: Vectors and Rotations

```
void ofApp::draw() {
    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) {
            float offsetDistance = ofRandom(minOffset, maxOffset);

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

            ofVec2f p1(0, 6.25);
            ofVec2f p2(25, 0);
            ofVec2f p3(0, -6.25);

            float rotation = ofRandom(360);
            p1.rotate(rotation);
            p2.rotate(rotation);
            p3.rotate(rotation);

            ofVec2f triangleOffset(offsetDistance, 0.0);
            triangleOffset.rotate(rotation);

            p1 += mousePos + triangleOffset;
            p2 += mousePos + triangleOffset;
            p3 += mousePos + triangleOffset;

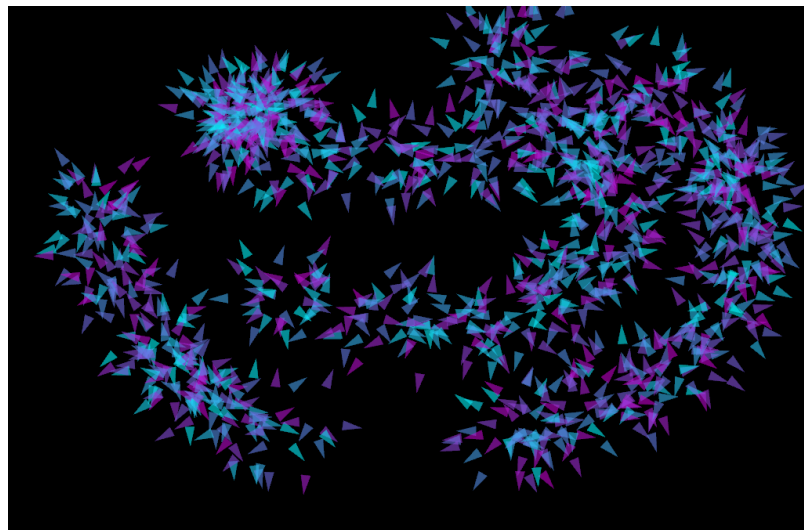
            ofColor aqua(0, 252, 255, alpha);
            ofColor purple(198, 0, 205, alpha);
            ofColor inbetween = aqua.getLerped(purple, ofRandom(1.0));
            ofSetColor(inbetween);

            ofDrawTriangle(p1, p2, p3);
        }

        if (ofGetMousePressed(OF_MOUSE_BUTTON_RIGHT)) {
            ofBackground(0);
        }
    }
}
```

마우스의 왼쪽 버튼을 누를때 화성화된다. 투명도를 150으로 해주며 for loop 시작한다. ofVec2f p1(0, 6.25);, ofVec2f p2(25, 0);, ofVec2f p3(0, -6.25)에는 삼각형의 각 꼭지점을 설정해주고 무작위로

삼각형을 회전시키면서 그려준다. p1, p2, p3: 마우스 위치와 오프셋을 더하여 삼각형을 이동시키는 역할을 한다. aqua하고 purple 두가지 색깔로 정의하면서 두 사이의 색상을 계산하면서 그려준다. 결과로는, 이러한 그림이 완성된다.



7. Update()

```
void ofApp::setup(){
    xPos = 5;
    ofBackground(ofColor::black);
}

//-----
void ofApp::update(){
    xPos += 2;
    if (ofGetWidth() < xPos){
        xPos = 5;
    }
}

//-----
void ofApp::draw(){
    ofSetColor(ofColor::red);
    ofDrawCircle(xPos, 100, 10);
}
```

이 예시에는 setup, update, draw 함수들 정의해주었다. Draw 함수에는 빨간색인 10 반지름하고 100 픽셀인 원을 생성하고, setup 함수에는 원의 가로 위치를 나타내는 5인 변수와 원을 오른쪽으로 이동시키는데 2씩 값을 정의해주는 update 함수가 있다. 또한 xPos 값이 너비를 초과하면 다시 원의 시작에 설정된다. 그러므로 결과는 빨간색 원이 왼쪽에서 오른쪽에 이동하는 프로그램이다. 아래와 같은 출력이 된다.

