

## 6주차 결과보고서

전공: 신문방송학과

학년: 3학년

학번: 20191150

이름: 전현길

1. Basic\_openFrameworks.pdf 강의자료 실습 코드들을 수행하고, 각 line 이 어떤 기능을 수행하는지 코드를 통해 상세하게 작성하시오.

(line-by-line으로 주석을 작성했습니다.)

### Basic shape(1)

```
void ofApp::setup() {
    ofSetFrameRate(60); // 프레임률을 60fps로 고정
    ofBackground(0);    // 배경을 명도 0(검정색)으로 고정
}

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

//-----
void ofApp::draw() {
    ofBackground(0); // 배경을 명도 0(검정색)으로 고정
    ofSetColor(255); // 흰색 설정

    ofDrawRectangle(50, 50, 100, 100); // (50, 50) 좌표 100*100 크기 사각형
    ofDrawCircle(250, 100, 50);        // (250, 100) 좌표 반지름 50 원
    ofDrawEllipse(400, 100, 80, 100);  // (400, 100) 좌표 80*100 크기 타원
    ofDrawTriangle(500, 150, 550, 50, 600, 150); // 순서대로 세 점의 x, y좌표
    ofDrawLine(700, 50, 700, 150);    // 순서대로 두 점의 x, y좌표
}
```

## Basic shape(2)

```
void ofApp::setup() {
    ofSetFrameRate(60); // 프레임률을 60fps로 고정
    ofBackground(0);    // 배경을 명도 0(검정색)으로 고정
}

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

//-----
void ofApp::draw() {
    ofBackground(0); // 배경을 명도 0(검정색)으로 고정
    ofSetColor(255); // 흰색 설정

    ofFill(); // 도형 안쪽 채우기
    ofDrawRectangle(50, 50, 100, 100); // (50, 50) 좌표 100*100 크기 사각형
    ofDrawCircle(250, 100, 50); // (250, 100) 좌표 반지름 50 원
    ofDrawEllipse(400, 100, 80, 100); // (400, 100) 좌표 80*100 크기 타원
    ofDrawTriangle(500, 150, 550, 50, 600, 150); // 순서대로 세 점의 x, y좌표
    ofDrawLine(700, 50, 700, 150); // 순서대로 두 점의 x, y좌표

    ofNoFill(); // 도형 안쪽 비우기
    ofDrawRectangle(50, 200, 100, 100); // (50, 200) 좌표 100*100 크기 사각형
    ofDrawCircle(250, 250, 50); // (250, 250) 좌표 반지름 50 원
    ofDrawEllipse(400, 250, 80, 100); // (400, 250) 좌표 80*100 크기 타원
    ofDrawTriangle(500, 300, 550, 200, 600, 300); // 순서대로 세 점의 x, y좌표
    ofDrawLine(700, 200, 700, 300); // 순서대로 두 점의 x, y좌표
}
```

## Single rectangle brush

```
void ofApp::setup() {
    ofSetFrameRate(60); // 프레임률을 60fps로 고정
    ofBackground(0);    // 배경을 명도 0(검정색)으로 고정
}

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

//-----
void ofApp::draw() {
    // 마우스 왼쪽 클릭 시
    if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) {
        ofSetColor(255); // 흰색 설정
        ofSetRectMode(OF_RECTMODE_CENTER); // 도형이 왼쪽 위가 아닌 중앙에서 그려짐
        // 마우스 좌표에서 50 * 50 크기의 정사각형 생성
        ofDrawRectangle(ofGetMouseX(), ofGetMouseY(), 50, 50);
    }
}
```

## bursting rectangle brush

```
//-----  
void ofApp::setup() {  
    ofSetFrameRate(60);           // 프레임률을 60fps로 고정  
    ofSetBackgroundAuto(false);   // 배경의 자동 갱신을 방지  
    ofBackground(0);              // 배경을 명도 0(검정색)으로 고정  
}  
  
//-----  
void ofApp::update() {}  
  
//-----  
void ofApp::draw() {  
    // 마우스 왼쪽 클릭 시  
    if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) {  
        ofSetRectMode(OF_RECTMODE_CENTER); // 도형이 왼쪽 위가 아닌 중앙에서 그려짐  
        int numRects = 10;                 // 직사각형 10개 생성  
        for (int r = 0; r < numRects; r++) { // 개수만큼 생성  
            ofSetColor(ofRandom(50, 255)); // 명도 50~255 범위 무작위 색 설정  
            float width = ofRandom(5, 20); // 직사각형 넓이  
            float height = ofRandom(5, 20); // 직사각형 길이  
            float xOffset = ofRandom(-40, 40); // 생성 x좌표 오차 범위  
            float yOffset = ofRandom(-40, 40); // 생성 y좌표 오차 범위  
            // 직사각형 그리기  
            ofDrawRectangle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, width,  
                             height);  
        }  
    }  
    // 오른쪽 클릭할 경우 배경 초기화  
    if (ofGetMousePressed(OF_MOUSE_BUTTON_RIGHT)) ofBackground(0);  
}
```



## glowing circle brush

```
void ofApp::setup() {
    ofSetFrameRate(60);          // 프레임률을 60fps로 고정
    ofSetBackgroundAuto(false);  // 배경의 자동 갱신을 방지
    ofBackground(0);             // 배경을 명도 0(검정색)으로 고정
}

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

//-----
void ofApp::draw() {
    // 마우스 왼쪽 클릭 시
    if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) {
        int maxRadius = 100;      // 브러쉬의 크기(=원의 크기)
        int radiusStepSize = 5;   // 원을 몇 번 겹칠지 결정
        // radius / radiusStepSize의 값만큼 원을 겹침
        int alpha = 3;            // 브러쉬의 불투명도, 높이면 진해짐
        int maxOffsetDistance = 100; // 원이 흩뿌려지는 정도
        for (int radius = maxRadius; radius > 0; radius -= radiusStepSize) {
            float angle = ofRandom(ofDegToRad(360.0));
            // 원이 어느 방향으로 이동할지에 대한 각도 결정
            // sin함수, cos함수에 angle 값을 넣어야 하므로 ofDegToRad() 함수 사용
            float distance = ofRandom(maxOffsetDistance); // 범위 난수 설정
            float xOffset = cos(angle) * distance;         // x좌표 이동
            float yOffset = sin(angle) * distance;         // y좌표 이동

            // ofColor class 생성자 → 객체명(R값, G값, B값, 투명도(alpha) 값)
            ofColor myOrange(255, 132, 0, alpha); // myOrange 객체 생성
            ofColor myRed(255, 6, 0, alpha);      // myRed 객체
            ofColor inBetween = myOrange.getLerped(myRed, ofRandom(1.0));
            // myOrange, myRed 객체의 색을 무작위 비율(0.0 ~ 1.0)로 섞음
            ofSetColor(inBetween); // 원의 색 초기화

            ofDrawCircle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, radius);
        }
    }
    // 오른쪽 클릭할 경우 배경 초기화
    if (ofGetMousePressed(OF_MOUSE_BUTTON_RIGHT)) ofBackground(0);
}
```

## Fleeing triangle brush

```
void ofApp::setup() {
    ofSetFrameRate(60); // 프레임률을 60fps로 고정
    ofSetBackgroundAuto(false); // 배경의 자동 갱신을 방지
    ofBackground(0); // 배경을 명도 0(검정색)으로 고정
}

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

//-----
void ofApp::draw() {
    // 마우스 왼쪽 클릭 시
    if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) {
        int numTriangles = 5; // 삼각형 5개
        int minOffset = 5; // 파티클 발생 최소 범위
        int maxOffset = 50; // 파티클 발생 최대 범위
        int alpha = 150; // 투명도(alpha)

        for (int t = 0; t < numTriangles; ++t) { // 삼각형 5개 생성
            float offsetDistance = ofRandom(minOffset, maxOffset); // 범위 난수 설정
            ofVec2f mousePos(ofGetMouseX(), ofGetMouseY()); // 마우스 x, y좌표 값

            // 2차원 벡터를 저장하는 클래스 ofVec2f
            ofVec2f p1(0, 6.25); // 꼭짓점 1
            ofVec2f p2(25.0, 0); // 꼭짓점 2
            ofVec2f p3(0, -6.25); // 꼭짓점 3

            float rotation = ofRandom(360); // ofRandom() 함수로 0 ~ 360도 난수 생성
            p1.rotate(rotation); // 세 꼭짓점을 동일하게 회전
            p2.rotate(rotation);
            p3.rotate(rotation);

            ofVec2f triangleOffset(offsetDistance, 0.0); // 생성 범위 벡터 변환
            triangleOffset.rotate(rotation); // 생성 지점 회전

            p1 += mousePos + triangleOffset; // 세 꼭짓점을 생성 지점으로 이동
            p2 += mousePos + triangleOffset;
            p3 += mousePos + triangleOffset;

            ofColor myAqua = ofColor::aqua; // 프리셋 ofColor 객체 aqua
            ofColor myPurple = ofColor::plum; // 프리셋 ofColor 객체 plum
            ofColor inbetween = myAqua.getLerped(myPurple, ofRandom(1.0));
            // myAqua와 myPurple 색을 무작위 범위(0.0 ~ 1.0)로 섞은 색상을 얻음
            ofSetColor(inbetween); // 섞은 색을 삼각형의 색으로 초기화

            ofDrawTriangle(p1, p2, p3); // 삼각형을 실제로 그림
        }
    }

    if (ofGetMousePressed(OF_MOUSE_BUTTON_RIGHT)) { // 오른쪽 클릭할 경우
        ofBackground(0); // 배경을 초기화
    }
}
```

update()

```
void ofApp::setup() {
    xPos = 5; // x좌표 시작점 셋
    ofBackground(ofColor::Black); // 배경 검정색 셋
}

//-----
void ofApp::update() {
    xPos += 2; // 1프레임당 좌표 2씩 이동
    if (ofGetWidth() < xPos) // 화면을 넘어갈 경우 시작점으로 복귀
        xPos = 5;
}

//-----
void ofApp::draw() {
    ofSetColor(ofColor::red); // 원 색 붉은색 프리셋 사용
    ofDrawCircle(xPos, 100, 10); // (xPos, 100) 좌표에 반지름 10 원 생성
}
```