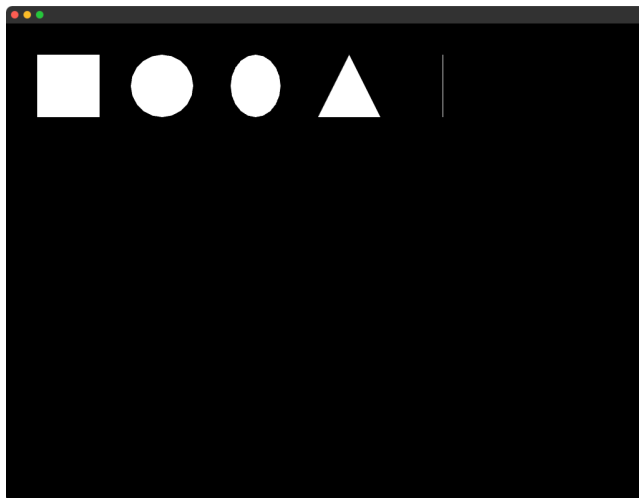


1. Basic_openFrameworks.pdf 강의자료 실습 코드 수행, 각 line의 기능

1) Basic shape (1)

```
void ofApp::draw(){
    ofBackground(0); //화면을 검정색(0)으로 채움
    ofSetColor(255); //그려질 도형의 색깔을 흰색(255)으로 설정함
    ofDrawRectangle(50, 50, 100, 100); //(50,50)에 왼쪽 위 꼭짓점을 두는 100*100의
직사각형을 그림
    ofDrawCircle(250, 100, 50); //(250, 100)에 중심을 두는 반지름이 50인 원을 그림
    ofDrawEllipse(400, 100, 80, 100); //(400, 100)에 중심을 두는 80*100의 타원을 그림
    ofDrawTriangle(500, 150, 550, 50, 600, 150); //(500, 150), (550, 50), (600, 150)에 꼭짓점을
두는 삼각형을 그림
    ofDrawLine(700, 50, 700, 150); //(700, 5)부터 (700, 150)까지 선을 그림
}
```



2) Basic Shape (2)

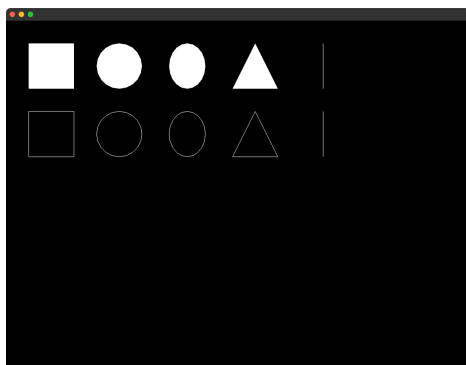
```

void ofApp::draw(){
    ofBackground(0); //화면을 검정색(0)으로 채움
    ofSetColor(255); //그려질 도형의 색깔을 흰색(255)으로 설정함

    ofFill(); //안을 채운 도형을 그리도록 설정함
    ofDrawRectangle(50, 50, 100, 100); //(50,50)에 왼쪽 위 꼭짓점을 두는 100*100의
직사각형을 그림
    ofDrawCircle(250, 100, 50); //(250,100)에 중심을 두는 반지름이 50인 원을 그림
    ofDrawEllipse(400, 100, 80, 100); //(400,100)에 중심을 두는 80*100의 타원을 그림
    ofDrawTriangle(500, 150, 550, 50, 600, 150); //(500,150), (550,50), (600,150)에 꼭짓점을 두는
삼각형을 그림
    ofDrawLine(700, 50, 700, 150); //(700,5)부터 (700,150)까지 선을 그림

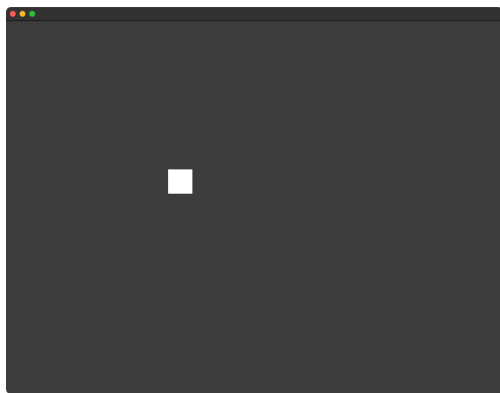
    ofNoFill(); //안을 채우지 않은 도형을 그리도록 설정함
    ofDrawRectangle(50, 200, 100, 100); //(50,200)에 왼쪽 위 꼭짓점을 두는 100*100의
직사각형을 그림
    ofDrawCircle(250, 250, 50); //(250,250)에 중심을 두는 반지름이 50인 원을 그림
    ofDrawEllipse(400, 250, 80, 100); //(400,250)에 중심을 두는 80*100의 타원을 그림
    ofDrawTriangle(500, 300, 550, 200, 600, 300); //(500,300), (550,200), (600,300)에 꼭짓점을
두는 삼각형을 그림
    ofDrawLine(700, 200, 700, 300); //(700,200)부터 (700,300)까지 선을 그림
}

```



3) Single rectangle brush

```
void ofApp::draw(){
    if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) { //마우스 왼쪽 버튼이 눌리면
        ofSetColor(255); //색깔을 흰색(255)으로 설정
        ofSetRectMode(OF_RECTMODE_CENTER); //ofDrawRectangle로 그려주는 기준 좌표를
        중심으로 변경
        ofDrawRectangle(ofGetMouseX(), ofGetMouseY(), 50, 50); //마우스 위치를 중심으로
        50*50의 직사각형을 그림
    }
}
```



위 코드에 다음 코드를 추가해본다.

```
void ofApp::setup(){
    ofSetBackgroundAuto(false); //프레임이 갱신되는 것을 막음
    ofBackground(0); //화면을 검정색(0)으로 채움
}
```



draw() 함수에서 한 줄을 주석처리하고 코드를 추가했다.

```
void ofApp::draw(){
    if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) { //마우스 왼쪽 버튼이 눌리면
//        ofSetColor(255);
        float randomColor = ofRandom(50, 255); //색을 랜덤하게 선택(바탕색과 비슷하여
구분이 잘 가지 않는 0~50은 제외)
        ofSetColor(randomColor); //랜덤하게 선택된 색으로 색깔 설정
        ofSetRectMode(OF_RECTMODE_CENTER); //ofDrawRectangle로 그려주는 기준 좌표를
중심으로 변경
        ofDrawRectangle(ofGetMouseX(), ofGetMouseY(), 50, 50); //마우스 위치를 중심으로
50*50의 직사각형을 그림
    }
}
```



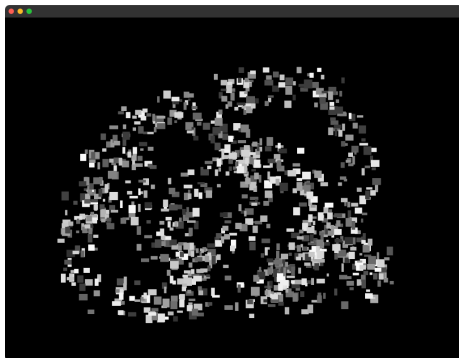
4) Bursting rectangle brush

```

void ofApp::setup(){
    ofSetBackgroundAuto(false); //프레임이 갱신되는 것을 막음
    ofBackground(0); //화면을 검정색으로 채움
    ofSetFrameRate(60); //frame rate을 60으로 설정
}

void ofApp::draw(){
    if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) { //마우스 왼쪽 버튼이 눌리면
        ofSetRectMode(OF_RECTMODE_CENTER); //ofDrawRectangle로 그려주는 기준 좌표를
        중심으로 변경
        int numRects = 10; //10개의 직사각형을 그리기 위해 설정
        for (int r = 0; r < numRects; r++) {
            ofSetColor(ofRandom(50, 255)); //랜덤하게 선택된 색으로 색깔 설정(바탕색과
            비슷하여 구분이 잘 가지 않는 0~50은 제외)
            float width = ofRandom(5, 20); //가로 길이를 5~20으로 랜덤하게 설정
            float height = ofRandom(5, 20); //세로 길이를 5~20으로 랜덤하게 설정
            float xOffset = ofRandom(-40, 40); //x 변화량을 -40~40으로 랜덤하게 설정
            float yOffset = ofRandom(-40, 40); //y 변화량을 -40~40으로 랜덤하게 설정
            ofDrawRectangle(ofGetMouseX()+xOffset, ofGetMouseY()+yOffset, width, height);
            //마우스 위치에서 x좌표는 xOffset, y좌표는 yOffset 떨어진 지점에 width*height의 직사각형을
            그림
        }
    }
}

```



5) Glowing circle brush

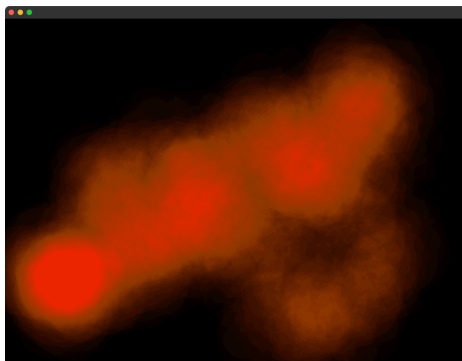
```

void ofApp::setup(){
    ofSetBackgroundAuto(false); //프레임이 갱신되는 것을 막음
    ofBackground(0); //화면을 검정색으로 채움
    ofSetFrameRate(60); //frame rate을 60으로 설정
}

void ofApp::draw(){
    if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) { //마우스 왼쪽 버튼이 눌리면
        int maxRadius = 100; //브러쉬 크기를 설정하는 변수
        int radiusStepSize = 5; //원의 크기 차이를 설정하는 변수
        int alpha = 3; //브러쉬 색의 투명도를 설정하는 변수
        int maxOffsetDistance = 100; //원의 퍼짐 정도를 설정하는 변수

        for (int radius=maxRadius; radius>0; radius-=radiusStepSize) { //가장 큰 원부터 원이
작아지면서 반복
            float angle = ofRandom(ofDegToRad(360.0)); //360도를 라디안으로 변환한 뒤
랜덤한 각도를 설정
            float distance = ofRandom(maxOffsetDistance); //원의 퍼짐 정도를 랜덤하게 설정
            float xOffset = cos(angle) * distance; //x 변화량 설정
            float yOffset = sin(angle) * distance; //y 변화량 설정
            ofSetColor(255, 103, 0, alpha); //색깔(R=255, G=103, B=0, 투명도=alpha) 설정
            ofDrawCircle(ofGetMouseX()+xOffset, ofGetMouseY()+yOffset, radius); //마우스
위치에서 x좌표는 xOffset, y좌표는 yOffset만큼 떨어진 위치에 원을 그림
        }
    }
}

```



6) Fleeing triangle brush

```
void ofApp::setup(){
    ofSetBackgroundAuto(false); //프레임이 갱신되는 것을 막음
    ofBackground(0); //화면을 검정색으로 채움
    ofSetFrameRate(60); //frame rate을 60으로 설정
}

void ofApp::draw(){
    if (ofGetMousePressed(OF_MOUSE_BUTTON_LEFT)) { //마우스 왼쪽 버튼이 눌리면
        int numTriangles = 10; //삼각형 수
        int minOffset = 5; //offset 최소 수치
        int maxOffset = 70; //offset 최대 수치
        int alpha = 150; //투명도

        for (int t=0; t<numTriangles; ++t) { //삼각형 수 만큼 반복
            //offset 거리를 랜덤하게 설정
            float offsetDistance = ofRandom(minOffset, maxOffset);

            //마우스 좌표를 벡터로 저장
            ofVec2f mousePos(ofGetMouseX(), ofGetMouseY());

            //(0,0)에서 오른쪽을 향하는 삼각형의 꼭짓점들을 설정
            ofVec2f p1(0, 6.25);
            ofVec2f p2(25, 0);
            ofVec2f p3(0, -6.25);

            //0~360도 중 랜덤으로 각도를 선택해서 삼각형을 회전 ((0,0)에서 회전)
            float rotation = ofRandom(360);
            p1.rotate(rotation);
            p2.rotate(rotation);
            p3.rotate(rotation);

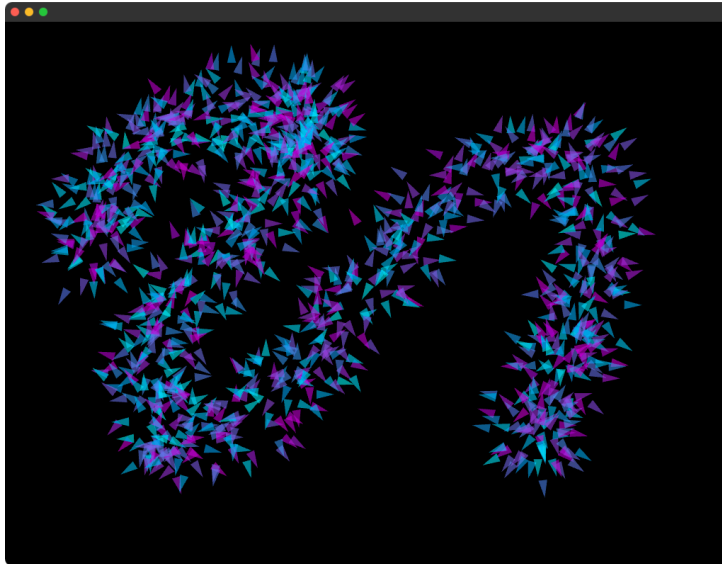
            //offset 설정
            ofVec2f triangleOffset(offsetDistance, 0.0);
            triangleOffset.rotate(rotation);

            //마우스 위치에서 offset만큼 더함
            p1 += mousePos + triangleOffset;
            p2 += mousePos + triangleOffset;
```

```
p3 += mousePos + triangleOffset;

//색깔을 파란색~보라색 계열로 랜덤하게 설정
ofColor aqua(0, 252, 255, alpha);
ofColor purple(198, 0, 205, alpha);
ofColor inbetween = aqua.getLerped(purple, ofRandom(1.0));
ofSetColor(inbetween);

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



7) update() 함수

ofApp.h

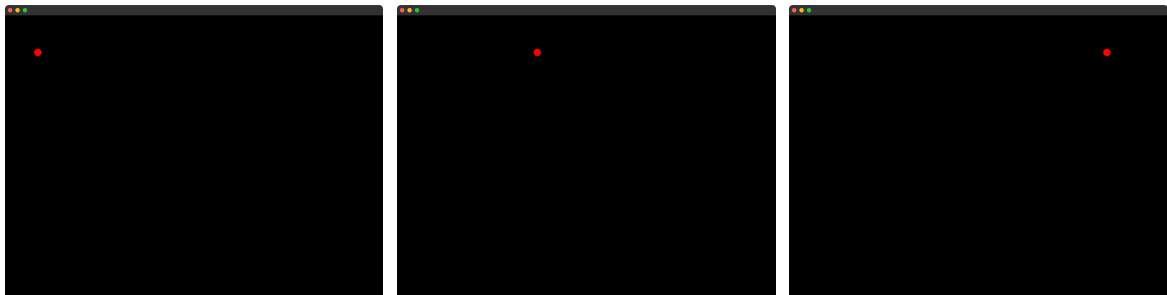
```
class ofApp : public ofAppBaseApp{
    ...
    int xPos;
    ...
};
```

ofApp.cpp

```
//-----
void ofApp::setup(){
    xPos = 5; //시작 x좌표를 설정
    ofBackground(ofColor::black); //화면을 검정색으로 채움
}

//-----
void ofApp::update(){
    xPos += 2; //x좌표를 2씩 더해줌
    if (ofGetWidth() < xPos) //x좌표가 화면 크기를 벗어난다면 x좌표를 5로 설정(리셋)
        xPos = 5;
}

//-----
void ofApp::draw(){
    ofSetColor(ofColor::red); //도형을 빨간색으로 그리도록 설정
    ofDrawCircle(xPos, 100, 10); //(xPos, 100)에 중심을 두는 반지름이 10인 원을 그림
}
```



빨간색 원이 왼쪽에서 오른쪽으로 이동한다.