6주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20191612 이름: 윤기웅

**1.**

.................

Basic Shapes

1. **ofBackground(0);**
   * 화면의 배경색을 설정합니다. 여기서 0은 RGB 값에서 검은색을 의미합니다. 즉, 화면 전체를 검은색으로 채웁니다.
2. **ofSetColor(255);**
   * 도형의 색을 정하는 것이고  255는 RGB 값에서 순수한 흰색을 의미합니다. 이 설정은 이후에 그려지는 모든 도형에 적용됩니다. 즉, 모든 도형들은 흰색으로 그려집니다.
3. **ofDrawRectangle(50, 50, 100, 100);**
   * 좌표 (50, 50) 위치에 가로 100, 세로 100의 크기를 가진 사각형을 그립니다. 이 좌표는 화면의 왼쪽 상단 모서리에서 우측으로 50, 아래로 50 떨어진 위치입니다.
4. **ofDrawCircle(250, 100, 50);**
   * 중심이 (250, 100)에 위치하고 반지름이 50인 원을 그립니다. 이 중심 좌표는 화면의 왼쪽 상단 모서리에서 우측으로 250, 아래로 100 떨어진 위치입니다.
5. **ofDrawEllipse(400, 100, 80, 100);**
   * 중심이 (400, 100)에 위치하고 가로 80, 세로 100의 크기를 가진 타원을 그립니다. 이 중심 좌표는 화면의 왼쪽 상단 모서리에서 우측으로 400, 아래로 100 떨어진 위치입니다.
6. **ofDrawTriangle(500, 150, 550, 50, 600, 150);**
   * (500, 150), (550, 50), (600, 150) 세 점을 이어서 삼각형을 그립니다. 이 점들은 각각 화면의 왼쪽 상단 모서리에서 지정된 만큼 우측 및 아래로 떨어진 위치를 가리킵니다.
7. **ofDrawLine(700, 50, 700, 150);**
   * 시작점 (700, 50)에서 끝점 (700, 150)까지 선을 그립니다. 이 선은 화면의 왼쪽 상단 모서리에서 우측으로 700, 아래로 50 떨어진 위치에서 시작하여 같은 열에서 아래로 100만큼 이동하여 끝납니다.

Basic Shapes(2)

1. ofBackground(0);: 화면 배경을 검은색으로 설정합니다.
2. ofSetColor(255);: 이후에 그려지는 모든 도형의 색상을 흰색으로 설정합니다.
3. ofFill();: 이후에 그려지는 도형을 색으로 채웁니다.
4. ofDrawRectangle(50, 50, 100, 100);: 위치 (50, 50)에 100x100 크기의 사각형을 그립니다.
5. ofDrawCircle(250, 100, 50);: 위치 (250, 100)에 반지름이 50인 원을 그립니다.
6. ofDrawEllipse(400, 100, 80, 100);: 위치 (400, 100)에 80x100 크기의 타원을 그립니다.
7. ofDrawTriangle(500, 150, 550, 50, 600, 150);: (500, 150), (550, 50), (600, 150) 위치를 꼭짓점으로 하는 삼각형을 그립니다.
8. ofDrawLine(700, 50, 700, 150);: (700, 50)에서 (700, 150)까지 선을 그립니다.
9. ofNoFill();: 이후에 그려지는 도형을 색으로 채우지 않고 윤곽선만 그립니다.
10. ofDrawRectangle(50, 200, 100, 100);: 위치 (50, 200)에 100x100 크기의 사각형의 윤곽선을 그립니다.
11. ofDrawCircle(250, 250, 50);: 위치 (250, 250)에 반지름이 50인 원의 윤곽선을 그립니다.
12. ofDrawEllipse(400, 250, 80, 100);: 위치 (400, 250)에 80x100 크기의 타원의 윤곽선을 그립니다.
13. ofDrawTriangle(500, 300, 550, 200, 600, 300);: (500, 300), (550, 200), (600, 300) 위치를 꼭짓점으로 하는 삼각형의 윤곽선을 그립니다.
14. ofDrawLine(700, 200, 700, 300);: (700, 200)에서 (700, 300)까지 선을 그립니다.

Single Rectangle Brush:using the mouse

1. if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) {: 마우스의 왼쪽 버튼이 눌러져 있는지 확인합니다. 만약 눌러져 있다면, 중괄호 {} 안의 코드가 실행됩니다.
2. ofSetColor(255);: 그리는 사각형의 색을 흰색으로 설정합니다. 여기서 255는 RGB 컬러 모델에서 흰색을 의미합니다.
3. ofSetRectMode(OF\_RECTMODE\_CENTER);: 사각형을 그릴 때 사각형의 중심을 기준으로 그리도록 설정합니다. 오픈 프레임워크는 사각형의 왼쪽 위 모서리를 기점으로 사각형을 그립니다. OF\_RECTMODE\_CENTER 모드를 사용하면, 지정한 좌표가 사각형의 중심이 되어 사각형이 그려집니다.
4. ofDrawRectangle(ofGetMouseX(), ofGetMouseY(), 50, 50);: 현재 마우스의 위치를 기준으로 가로와 세로 길이가 각각 50인 사각형을 그립니다. ofGetMouseX()와 ofGetMouseY() 함수는 현재 마우스 커서의 x, y 좌표를 반환합니다. 따라서, 이 코드는 마우스에 의해 사각형을 그립니다.

Single Rectangle Brush: Using the Mouse

1. ofSetBackgroundAuto(false);: 프레임 간에 자동으로 배경을 지우지 않도록 설정합니다. 기본적으로 오픈프레임워크는 원래는 무엇인가 그리기 전에 배경을 지워서 그래픽이 잘 보이게 합니다.
2. ofBackground(0); 배경 색을 검정색으로 설정합니다. 여기서 0은 RGB 색상 코드에서 검정색을 의미합니다.

................

Single Rectangle Brush : Using the Mouse

1. if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT)) { ... }: 마우스 왼쪽 버튼이 눌려있는지를 검사합니다. 만약 눌려있다면, 중괄호 안의 코드가 실행됩니다.
2. float randomColor = ofRandom(50, 255);: ofRandom 함수를 사용하여 50과 255 사이의 무작위 값을 생성하여 randomColor 변수에 할당합니다. 이 값은 나중에 사각형의 색상을 결정하는 데 사용됩니다.
3. ofSetColor(randomColor);: ofSetColor 함수는 도형의 색상을 설정합니다. randomColor 변수의 값으로 색상을 정하고, 매번 클릭에서 다른 색상의 사각형이 그려지도록 합니다.
4. ofSetRectMode(OF\_RECTMODE\_CENTER);: 사각형을 그릴 때 기준점의 위치를 설정합니다. OF\_RECTMODE\_CENTER는 사각형의 중앙이 기준점이 되도록 설정합니다. 이는 사각형을 마우스 커서 위치에 중심을 맞추어 그리기 위함입니다.
5. ofDrawRectangle(ofGetMouseX(), ofGetMouseY(), 50, 50);: 마지막으로, ofDrawRectangle 함수는 화면에 사각형을 그립니다. ofGetMouseX()와 ofGetMouseY() 함수는 현재 마우스 커서의 x, y 좌표를 반환하며, 이는 사각형의 중심 위치로 사용됩니다. 사각형의 너비와 높이는 각각 50으로 설정됩니다.
6. ofSetFrameRate(60); 프레임 속도를 설정하는 함수입니다. 이 함수는 매개변수로 주어진 숫자값(이 경우 60)을 프로그램의 목표 프레임 속도로 설정합니다.
7. if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT): 마우스의 왼쪽 버튼이 눌러지면 아래의 항목들을 실행합니다.
8. ofSetRectMode(OF\_RECTMODE\_CENTER): 사각형의 중심을 기준으로 사각형을 그리도록 설정합니다.
9. for(int r = 0; r < numRects; r++): 조건에 맞는 for문을 수행합니다.
10. ofSetColor(ofRandom(50, 255)): 50~255 사이의 값을 가지는 무작위 색으로 도형을 만듭니다.
11. float width = ofRandom(5, 20)

float height = ofRandom(5, 20)

float xOffset = ofRandom(-40, 40)

float yOffset = ofRandom(-40, 40)

: float 형태의 width, height, xOffset, yOffset 변수에 각각 범위에 맞는 임의의 값을 배정합니다.

1. ofDrawRectangle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, width, height): 현재 마우스 위치에서 xOffset, yOffset 만큼을 더해준 위치에서 가로가 width, 세로가 height의 크기를 갖는 직사각형을 그립니다.

Bursting Rectangle Brush : Creating Randomized Bursts

(1) ofSetFrameRate(60): 1초당 60개의 프레임을 설정합니다.

(2) ofSetBackgroundAuto(false): 자동적인 화면을 지우기 기능을 해제합니다.

(3) ofBackground(0): 바탕 화면의 색상을 검정색으로 설정합니다.

- draw()

(1) if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT): 마우스의 왼쪽 버튼이 눌러지면 아래의 항목들을 실행합니다.

(2) ofSetRectMode(OF\_RECTMODE\_CENTER): 사각형의 중심을 기준으로 사각형을 그리도록 설정합니다.

(3) for(int r = 0; r < numRects; r++): r이 0~numRects-1이 되는 동안 for문 안의 코드를 실행합니다.

(4) ofSetColor(ofRandom(50, 255)): 50~255 사이의 값을 가지는 랜덤 색상으로 도형을 그립니다.

(5) float width = ofRandom(5, 20)

float height = ofRandom(5, 20)

float xOffset = ofRandom(-40, 40)

float yOffset = ofRandom(-40, 40)

: float 타입을 가지는 width, height, xOffset, yOffset 변수에 5~20 또는 -40~40 사이의 임의 값을 저장합니다.

(6) ofDrawRectangle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, width, height): 현재 마우스 위치에서 xOffset, yOffset 만큼을 더해준 위치에서 가로가 width, 세로가 height의 크기를 갖는 직사각형을 그립니다.

Glowing Circle Brush: Using Color

(1) if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT): 마우스의 왼쪽 버튼이 눌러지면 아래의 항목들을 실행합니다.

(2) int maxRadius = 100: 브러쉬의 최대 두께입니다.

(3) int radiusStepSize = 5: 브러쉬의 원이 1단계에 변하는 크기입니다.

(4) int alpha = 3: 브러쉬의 불투명도 변화의 크기입니다.

(5) int maxOffsetDistance = 100: 원의 분포가 최대로 커지는 크기입니다.

(6) for (int radius = maxRadius; radius > 0; radius -= radiusStepSize): radius가 maxRadius에서 0까지 radiusStepSize만큼 작아지면서 for문 안의 코드를 실행합니다.

(7) float angle = ofRandom(ofDegToRad(360.0)): 코드는 0부터 360도까지의 랜덤한 각도 값을 라디안 값으로 변환하여 float 타입의 angle 변수에 저장합니다.

(8) float distance = ofRandom(maxOffsetDistance): 0~maxOffsetDistance 사이의 값을 랜덤으로 지정하여 float 타입의 distance 변수에 저장합니다.

(9) float xOffset = cos(angle)\*distance, float yOffset = cos(angle)\*distance

: 극좌표에서 데카르트 좌표로 변환합니다.

(10) ofColor myOrange(255, 132, 0, alpha): rgb를 (255, 132, 0)으로 갖는 주황색의 투명도를 alpha로 해서 myOrange 변수로 지정합니다.

(11) ofColor myRed(255, 6, 0, alpha): rgb를 (255, 6, 0)으로 갖는 빨강색의 투명도를 alpha로 설정하여 myRed라는 변수로 지정합니다.

(12) ofColor inBetween = myOrange.getLerped(myRed, ofRandom(1.0)): myOrange와 myRed 두 색상을 0부터 1.0까지의 임의의 값을 사용하여 선형 보간한 결과를 inBetween 변수에 저장하여 myOrange와 myRed 사이의 새로운 색상을 만듭니다.

(13) ofSetColor(inBetween): inBetween 해당 색을 이용해서 도형을 그립니다.

(14) ofCircle(ofGetMouseX() + xOffset, ofGetMouseY() + yOffset, radius): 현재 마우스 위치서 xOffset, yOffset 만큼 더한 곳에 radius의 반지름의 크기의 원을 그립니다.

Fleeing Triangle Brush:vectors

(1) if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_LEFT): 마우스의 왼쪽 버튼을 누르고 아래 코드를 실행합니다.

(2) for (int t = 0; t < numTriangles; ++t): t가 0~numTriangles(10) 사이의 값일 동안 증가시키면서 for문 안의 코드를 실행합니다.

(3) float offsetDistance = ofRandom(minOffset, maxOffset) : minOffset(5)에서 maxOffset (70) 사이의 값을 가지는 랜덤 값을 지정하여 float 타입의 offsetDistance 변수에 저장합니다.

(4) ofVec2f mousePos(ofGetMouseX(), ofGetMouseY()): 현재 마우스의 커서 위치를 2차원 벡터로 표현하여 mousePos 변수에 저장합니다.

(5) ofVec2f p1(0, 25.0): 벡터의 x좌표가 0, y좌표가 25.0으로 설정된 2차원 벡터를 p1에 저장합니다.

(6) ofVec2f p2(100, 0): 벡터의 x좌표가 100, y좌표가 0으로 설정된 2차원 벡터를 p2에 저장합니다.

(7) ofVec2f p3(0, -25.0): 벡터의 x좌표가 0, y좌표가 -25.0으로 설정된 2차원 벡터를 p3에 저장합니다.

(8) float rotation = ofRandom(360): 0~360 사이의 값을 무작위로 정해서 rotation에 저장합니다.

(9) p1.rotate(rotation) p2.rotate(rotation) p3.rotate(rotation)

: p1, p2, p3 벡터의 각각의 회전시킬 각도를 지정합니다.

(10) p1 += mousePos p2 += mousePos p3 += mousePos

: p1, p2, p3의 위치를 현재 마우스의 좌표로 옮깁니다.

(11) ofSetColor(255, 50): (255, 50) 색으로 해당 도형을 그립니다.

(12) ofTriangle(p1.x, p1.y, p2.x, p2.y, p3.x, p3.y): p1, p2, p3를 꼭짓점으로 갖는 삼각형을 그린다.

(13) if (ofGetMousePressed(OF\_MOUSE\_BUTTON\_RIGHT)): 마우스의 오른쪽 버튼이 누르고 다음의 코드를 실행합니다.

(14) ofBackground(0): 바탕 색을 검정으로 지정합니다.

Update

setup()

(1) xPos = 5: 수평 위치를 5로 설정합니다.

(2) ofBackground(ofColor::black): 바탕 색을 검정색으로 설정합니다.

-update()

(1) xPos += 2 : update 함수 실행 후 수평 위치를 2만큼 이동합니다.

(2) if (ofGetWidth() < xPos): 현재 열린 창의 폭이 xPos보다 작으면 다음 코드를 실행합니다.

(3) xPos = 5: x위치를 5로 지정합니다.

- draw()

(1) ofSetColor(ofColor::red): 색을 빨간색으로 하여 도형을 그립니다.

(2) ofDrawCircle(xPos, 100, 10): (xPos, 100)의 위치에서 반지름 10인 원을 그립니다.

.......................

.........................