Creative coding

Programowanie Kreatywne

Piotr Makowski, III rok IS

Programowanie Kreatywne

- To pojęcie jest niezwykle obszerne, jednak wiąże się zazwyczaj z ciekawym użyciem grafiki, dźwięku itp. Dlatego niezwykle istotnym dla procesu takiego pisania jest skupienie się na tym co ma robić i jak ma wyglądać a nie na "ujarzmieniu" API.

Przykłady Frameworków

OpenFramework

- Język : C++

Cinder

- Język: C++

Processing

- Język : Java

openFramework

- Napisany w C++
- Wydajny, Szybki
- Świetnie działający EventListener
- Opakowane najbardziej uniwersane Biblioteki :

  OpenGL,GLUT, cairo, OpanAL etc.

openFramework

- Uniwersalność
- Notacja węgierska
- Wygoda i prostota

openFramework

OpenFramework daje mnóstwo możliwości zarówno programistom “zaawansowanym” jak i początkującym. Każdy projekt w openFramework jest podzielony na przynajmniej 3 pliki

openFramework

Plik main.cpp

- Plik ten składa się najczęściej z 3 linijek, w których następuje inicjalizacja okna za pomocą OpenGL, oraz wywołanie aplikacji.

openFramework

- Pozostałe 2 pliki to *.h oraz *.cpp, gdzie “dzieje się magia”. Tam opisane są wszystkie eventy, oraz cały proces przebiegu wykonania aplikacji.

openFramework

- Pozostałe 2 pliki to *.h oraz *.cpp, gdzie “dzieje się magia”. Tam opisane są wszystkie eventy, oraz cały proces przebiegu wykonania aplikacji.

Przykład 1

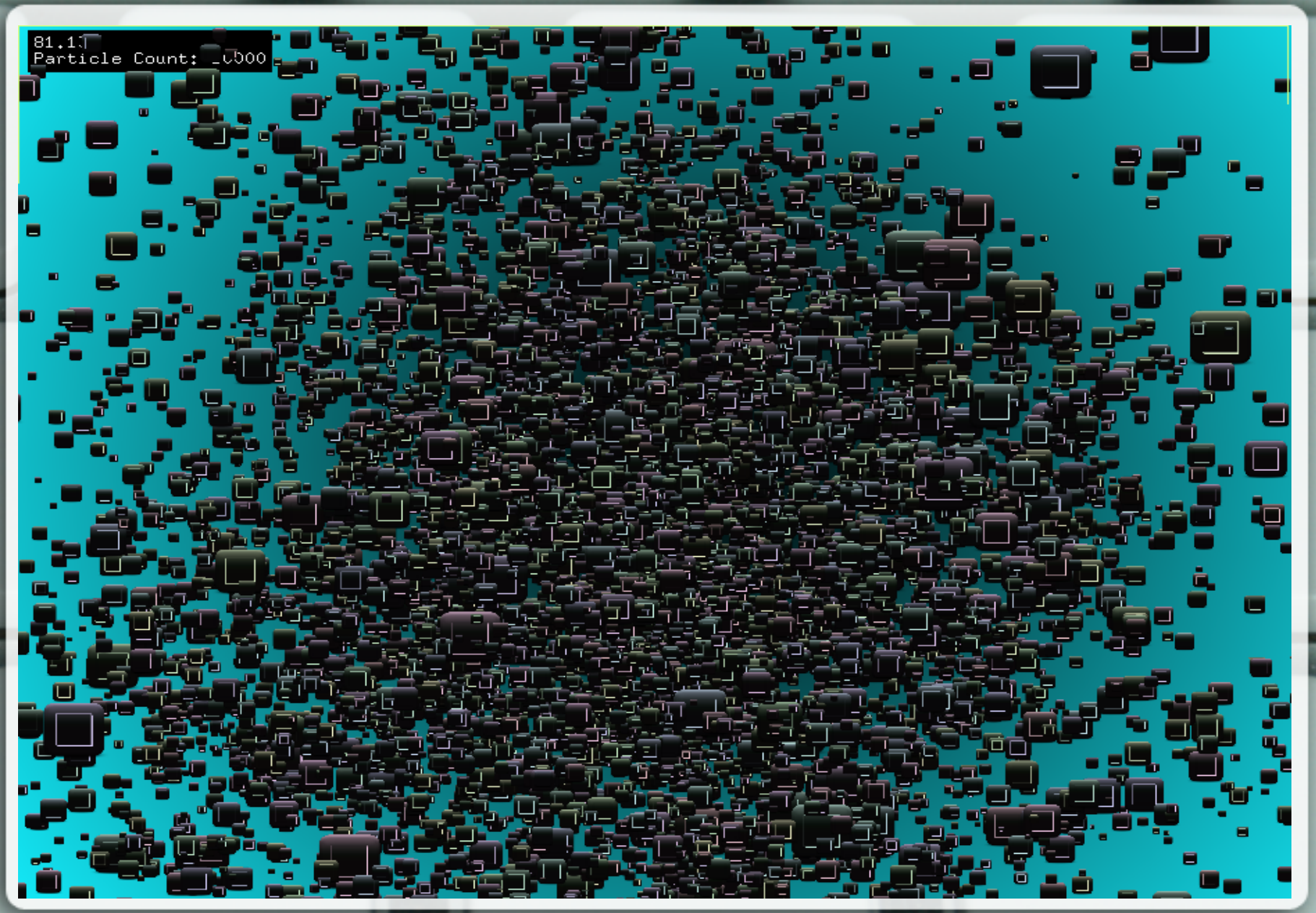
```
//--------------------------------------------------------------
void testApp::setup() {
    ofBackground(0, 0, 0);

    cameraRotation.set(0);
    zoom = -500;
    zoomTarget = 200;

    billboards.getVertices().resize(NUM_BILLBOARDS);
    billboards.getColors().resize(NUM_BILLBOARDS);
    billboards.getNormals().resize(NUM_BILLBOARDS,ofVec3f(0));

    // ------------------------- billboard particles
    for (int i=0; i<NUM_BILLBOARDS; i++) {

        billboardVels[i].set(ofRandomf(), -1.0, ofRandomf());
        billboards.getVertices()[i].set(ofRandom(-500, 500),
                                        ofRandom(-500, 500),
                                        ofRandom(-500, 500));

        billboards.getColors()[i].set(ofColor::fromHsb(ofRandom(50, 250), 50, 255));
        billboardSizeTarget[i] = ofRandom(4, 64);

    }


    billboards.setUsage( GL_DYNAMIC_DRAW );
    billboards.setMode(OF_PRIMITIVE_POINTS);
    //billboardVbo.setVertexData(billboardVerts, NUM_BILLBOARDS, GL_DYNAMIC_DRAW);
    //billboardVbo.setColorData(billboardColor, NUM_BILLBOARDS, GL_DYNAMIC_DRAW);

    // load the bilboard shader
    // this is used to change the
    // size of the particle
    billboardShader.load("Billboard");

    // we need to disable ARB textures in order to use normalized texcoords
    ofDisableArbTex();
    texture.loadImage("sqr.png");
    ofEnableAlphaBlending();
}
```

Przykład 1

```
void testApp::draw() {
    ofBackgroundGradient(ofColor(0), ofColor(20, 240, 255));

    string info = ofToString(ofGetFrameRate(), 2)+"\n";
    info += "Particle Count: "+ofToString(NUM_BILLBOARDS);
    ofDrawBitmapStringHighlight(info, 30, 30);

    ofSetColor(255);

    ofPushMatrix();
    ofTranslate(ofGetWidth()/2, ofGetHeight()/2, zoom);
    ofRotate(cameraRotation.x, 1, 0, 0);
    ofRotate(cameraRotation.y, 0, 1, 0);
    ofRotate(cameraRotation.y, 0, 0, 1);

    // bind the shader so that wee can change the
    // size of the points via the vert shader
    billboardShader.begin();

    ofEnablePointSprites();
    texture.getTextureReference().bind();
    billboards.draw();
    texture.getTextureReference().unbind();
    ofDisablePointSprites();

    billboardShader.end();

    ofPopMatrix();
}
```

Zmiany

- Zmieniona ilość obiektów z 5000 na 10000
- Zmieniony wczytywany obraz
- Zmieniony zakres położenia
- Zmieniony zakres kolorystyczny I tło


81.1
Particle Count: 20000

```
int nPoints = 4096; // points to draw
float complexity = 3; // wind complexity
float pollenMass = .8; // pollen mass
float timeSpeed = .02; // wind variation
float phase = TWO_PI; // separate u-nois
float windSpeed = 40; // wind vector magn
int step = 10; // spatial sampling rate f
bool debugMode = false;
```

```
void testApp::setup() {
    ofSetVerticalSync(true); // don't go too fast
    ofEnableAlphaBlending();

    // randomly allocate the points across the screen
  points.resize(nPoints);
  for(int i = 0; i < nPoints; i++) {
    points[i] = ofVec2f(ofRandom(0, ofGetWidth()), ofRandom(0, ofGetHeight()));
  }

    // we'll be drawing the points into an ofMesh that is drawn as bunch of points
    cloud.clear();
    cloud.setMode(OF_PRIMITIVE_POINTS);
}

//--------------------------------------------------------------
void testApp::update() {
    width = ofGetWidth(), height = ofGetHeight();
  t = ofGetFrameNum() * timeSpeed;
  for(int i = 0; i < nPoints; i++) {
        float x = points[i].x, y = points[i].y;
        ofVec2f field = getField(points[i]); // get the field at this position
        // use the strength of the field to determine a speed to move
        // the speed is changing over time and velocity-space as well
    float speed = (1 + ofNoise(t, field.x, field.y)) / pollenMass;
        // add the velocity of the particle to its position
    x += ofLerp(-speed, speed, field.x);
    y += ofLerp(-speed, speed, field.y);
        // if we've moved outside of the screen, reinitialize randomly
    if(x < 0 || x > width || y < 0 || y > height) {
      x = ofRandom(0, width);
      y = ofRandom(0, height);
    }
        // save the changes we made to the position
    points[i].x = x;
    points[i].y = y;
        // add the current point to our collection of drawn points
        cloud.addVertex(ofVec2f(x, y));
    }
}
```

Przykład 2

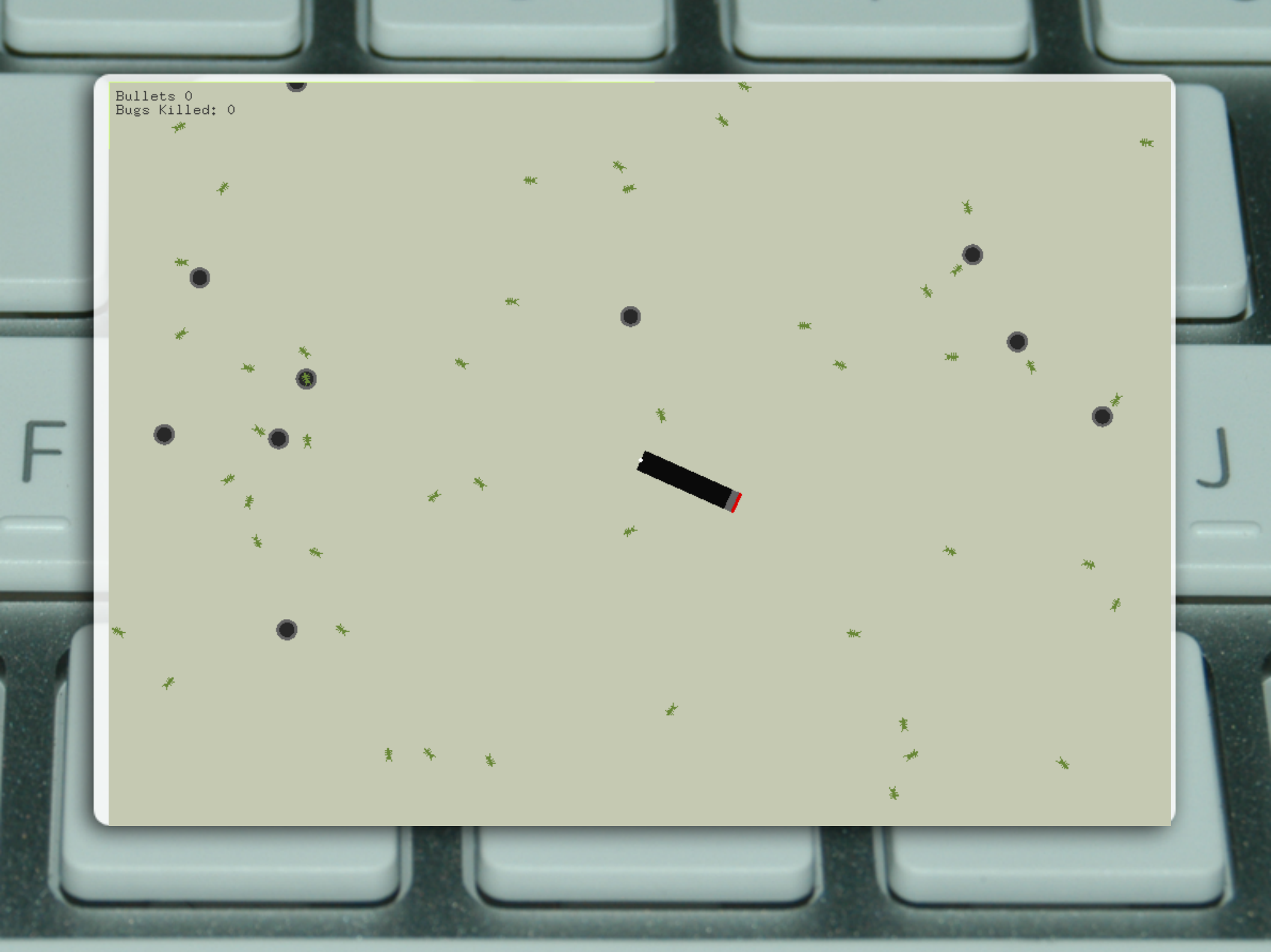
```
void testApp::draw() {
    ofBackground(255);
  if(debugMode) {
    ofSetColor(0);
    float s = windSpeed;
        // draw a vector field for the debug screen
    for(int i = 0; i < width; i += step) {
      for(int j = 0; j < height; j += step) {
                ofVec2f field = getField(ofVec2f(i, j));
        ofPushMatrix();
        ofTranslate(i, j);
                ofSetColor(0);
        ofLine(0, 0, ofLerp(-windSpeed, windSpeed, field.x), ofLerp(-windSpeed, windSpeed, field.y));
        ofPopMatrix();
      }
    }
        // draw the points as circles
        ofSetColor(ofColor::red);
        for(int i = 0; i < nPoints; i++) {
      ofCircle(points[i], 2);
        }
  } else {
        // when not in debug mode, draw all the points to the screen
    ofSetColor(30,10);
        cloud.draw();
    }

    ofDrawBitmapStringHighlight("click to reset\nhit any key for debug", 10, 10, ofColor::white, ofColor::black);
}

//--------------------------------------------------------------
```

Menu
[openFrameworks - ...
src/testApp.cpp [noi...
Untitled 1 - LibreOffi...
noiseField2dExample
Untitled window
Pobranie zrzutu ekr...
1 °C sob 16 lut, 00:19

Przykład 3

```
void testApp::setup() {

    ofBackgroundHex(0xc5c9b2);
    ofSetFrameRate(60);

    bFire = true;
    bugsKilled = 0;
    maxBullets = 30;

    // add some random holes for the bugs to come out
    int nHoldes = 10;
    for(int i=0; i<nHoldes; i++) {
        ofVec2f p(ofRandomWidth(), ofRandomHeight());
        holes.push_back(p);
    }

    // listen to any of the events for the game
    ofAddListener(GameEvent::events, this, &testApp::gameEvent);

}

//--------------------------------------------------------------
```

```
void testApp::update() {

    if((int)ofRandom(0, 20)==10) {

        int randomHole = ofRandom(holes.size());

        Bug newBug;
        newBug.pos = holes[randomHole];
        newBug.vel.set(ofRandom(-1, 1), ofRandom(-1, 1));
        bugs.push_back(newBug);
    }

    for(int i=0; i<bugs.size(); i++) {

        bugs[i].update();

        // bug pos and size
        float   size = bugs[i].size;
        ofVec2f pos  = bugs[i].pos;

        // wrap the bugs around the screen
        if(pos.x > ofGetWidth()-size)  bugs[i].pos.x = -size;
        if(pos.x < -size)              bugs[i].pos.x = ofGetWidth()-size;
        if(pos.y > ofGetHeight()+size) bugs[i].pos.y = -size;
        if(pos.y < -size)              bugs[i].pos.y = ofGetHeight()-size;

    }

    // check if we should remove any bugs
    ofRemove(bugs, shouldRemoveBug);

    // update the bullets
    for(int i=0; i<bullets.size(); i++) {
        bullets[i].update();
    }

    // check if we want to remove the bullet
    ofRemove(bullets, shouldRemoveBullet);


    // did we hit a bug loop we are checking to see if a bullet
    // hits a bug. if so we are going to launch an event for the game
    for(int i=0; i<bullets.size(); i++) {
        for(int j=0; j<bugs.size(); j++) {

            ofVec2f a       = bullets[i].pos;
            ofVec2f b       = bugs[j].pos;
            float   minSize = bugs[j].size;

            if(a.distance(b) < minSize) {

                static GameEvent newEvent;
                newEvent.message = "BUG HIT";
                newEvent.bug     = &bugs[j];
                newEvent.bullet  = &bullets[i];

                ofNotifyEvent(GameEvent::events, newEvent);
            }

        }
```


Bullets 0
Bugs Killed: 0

Przejdzmy do prezentacji programów

- Wykorzystane materiały oraz przykłady pochodzą ze strony OpenFramework.cc
- Przy próbie zdefiniowania czym jest programowanie kreatywne pomógł serwis

  http://qwrp.blogspot.com

  http://qwrp.blogspot.com/2011/01/openframeworks-kreatywne-kodowanie.html