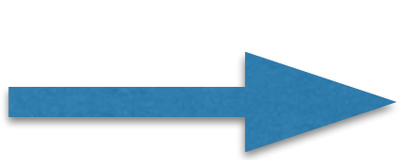
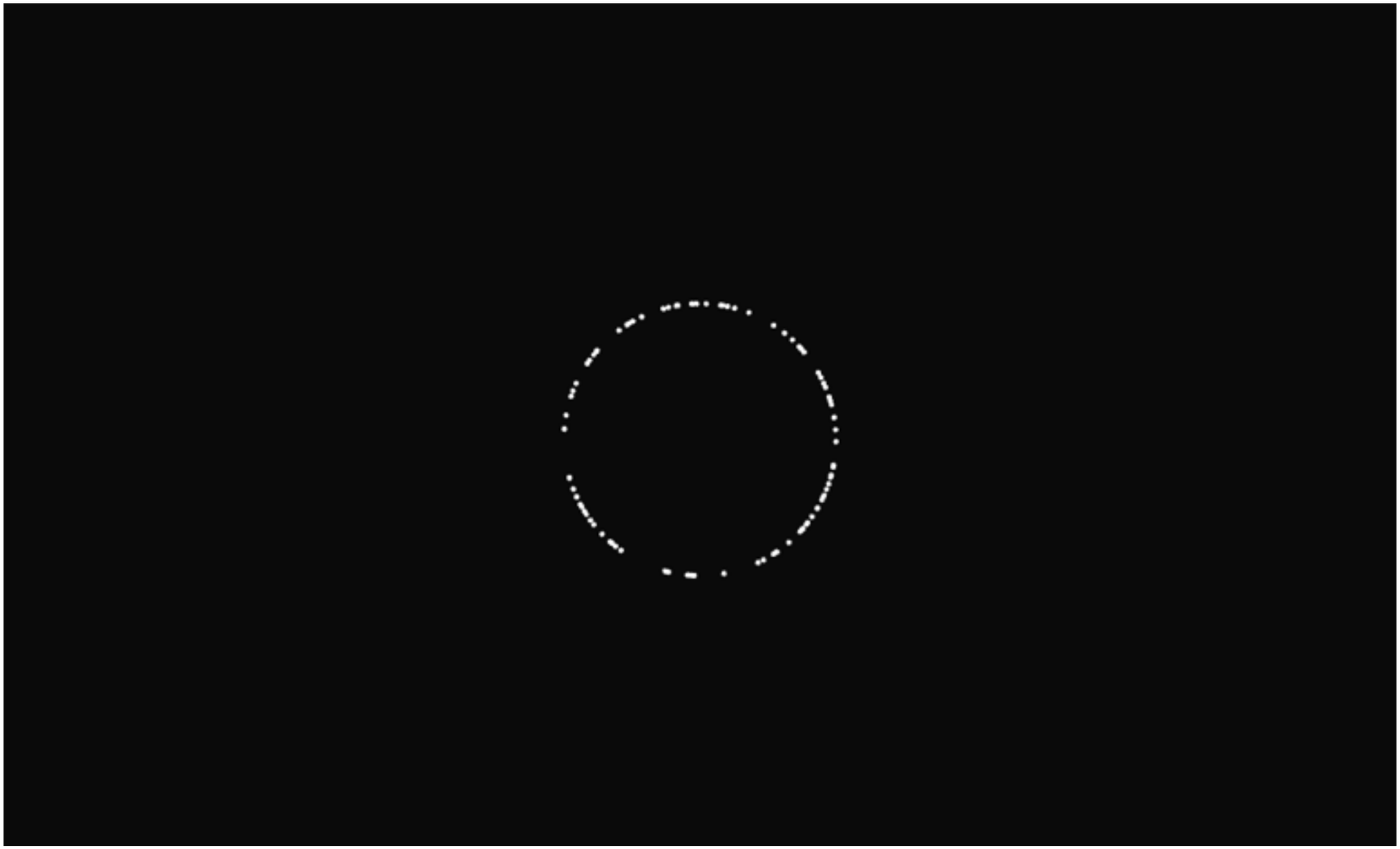
Kugel

Mathematics
Formular

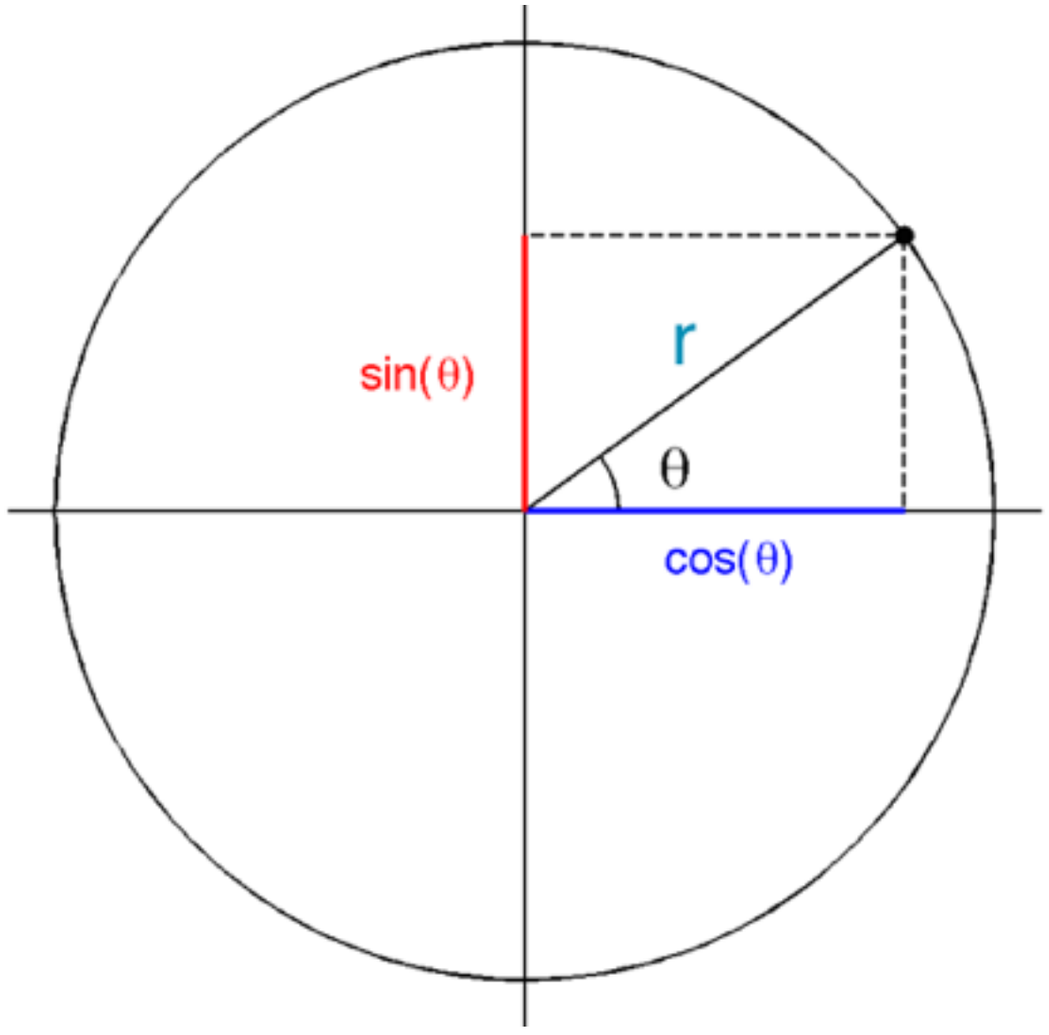


Code

Two Dimensional



Erste Stufe

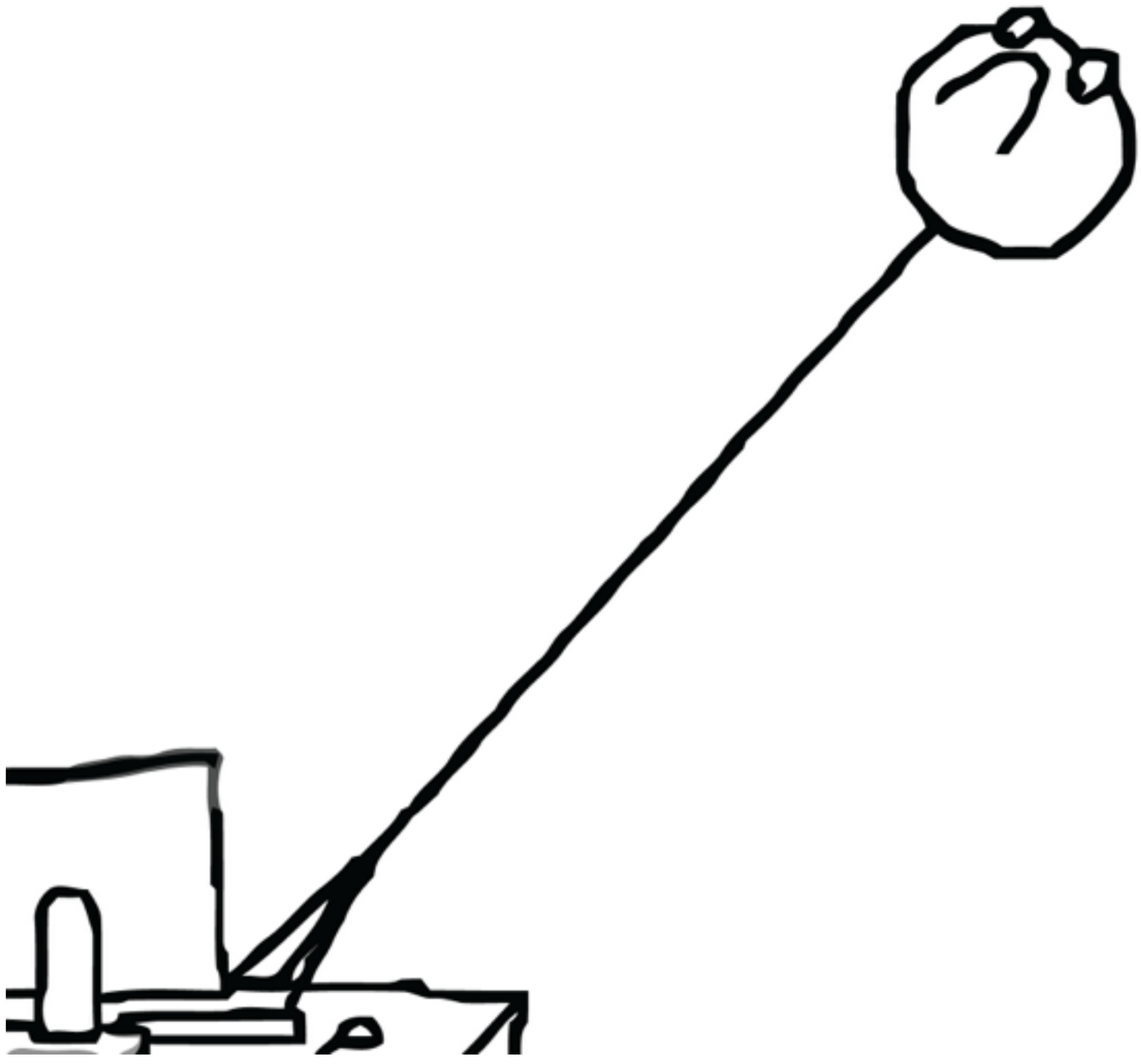


$$x = r * \cos(\theta)$$
$$y = r * \sin(\theta)$$

http://mathworld.wolfram.com/PolarCoordinates.html

Cartesian Coordinates

- $x = r * \cos \theta$
- $y = r * \sin \theta$
- θ : degree - ofRandom(θ)
- r : radius



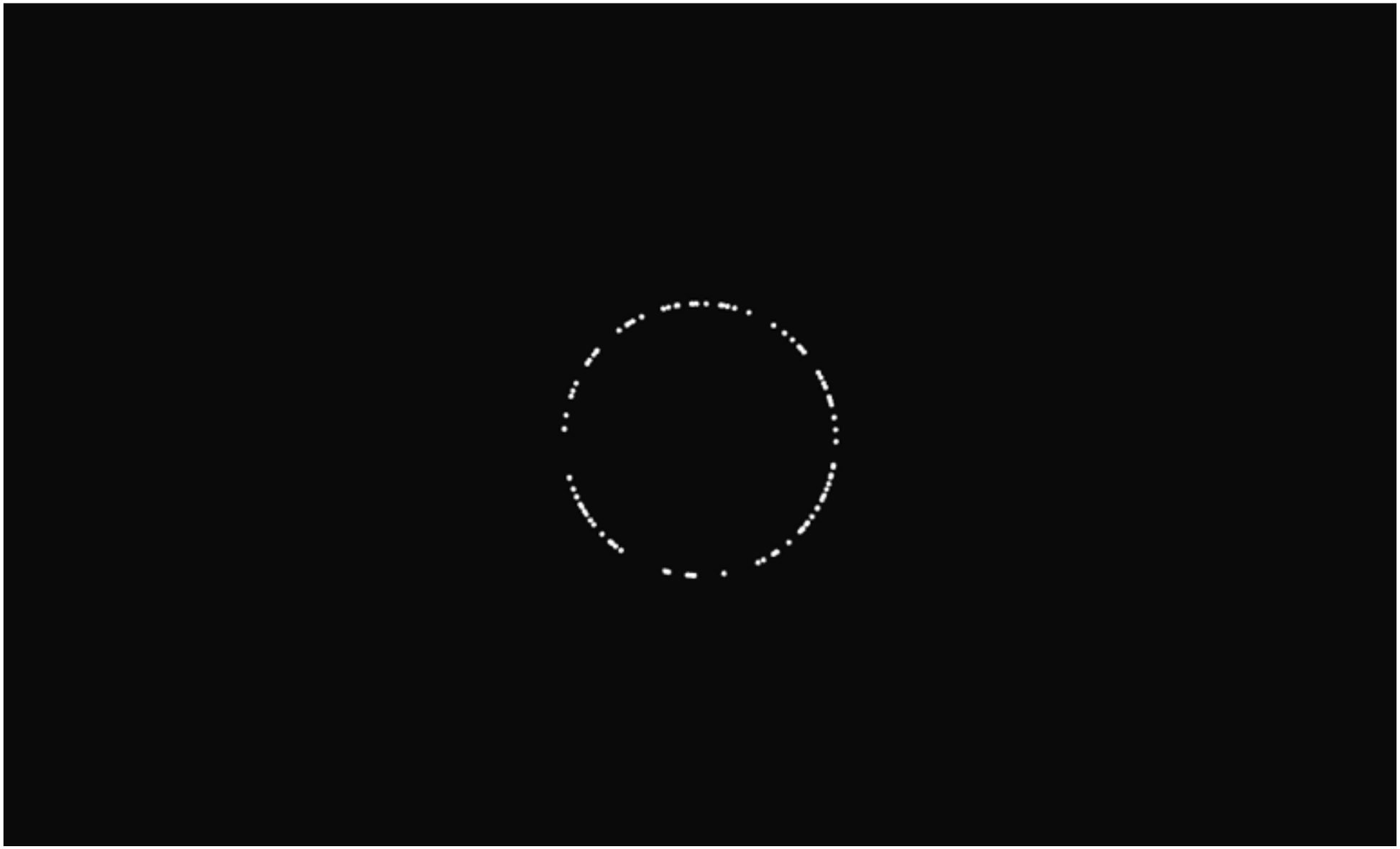
- vector<ofVec2f> **point2D**;
- point2dSetting();
  - float **_theta** = ofRandom(0, TWO_PI);
  - float **_radius** = 100;
  - float **_x** = cos(**_theta**) * **_radius**;
  - float **_y** = sin(**_theta**) * **_radius**;
  - ofVec2f **_point2D_temp** = ofVec2f(**_x**, **_y**);
  - **point2D**.push_back(**_point2D_temp**);

oder

- float _rad = ofRandom(0, TWO_PI);

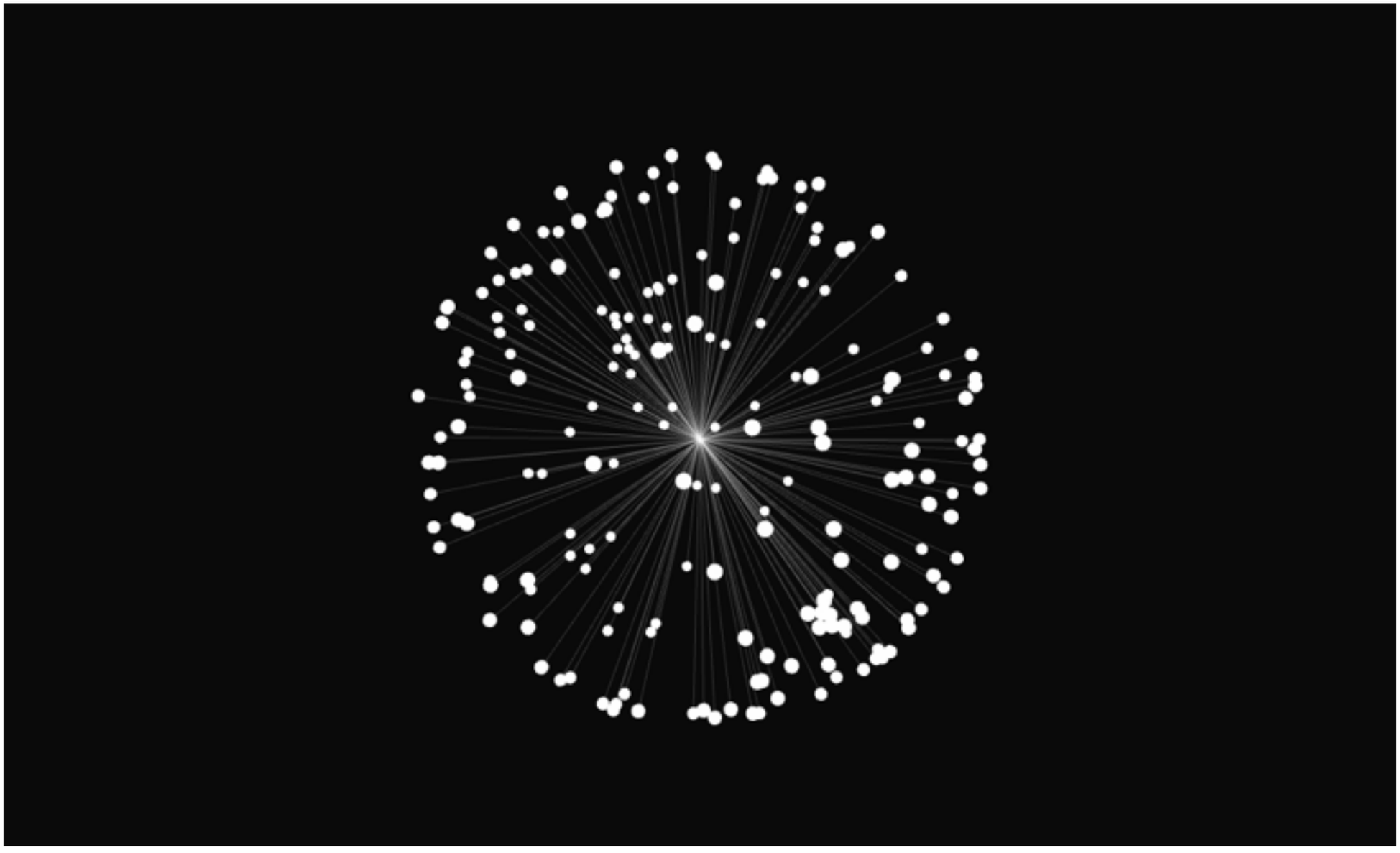
- float _radius = 100;

- point2D.push_back(ofVec2f(cos(_rad) * _radius, sin(_rad) * _radius)));

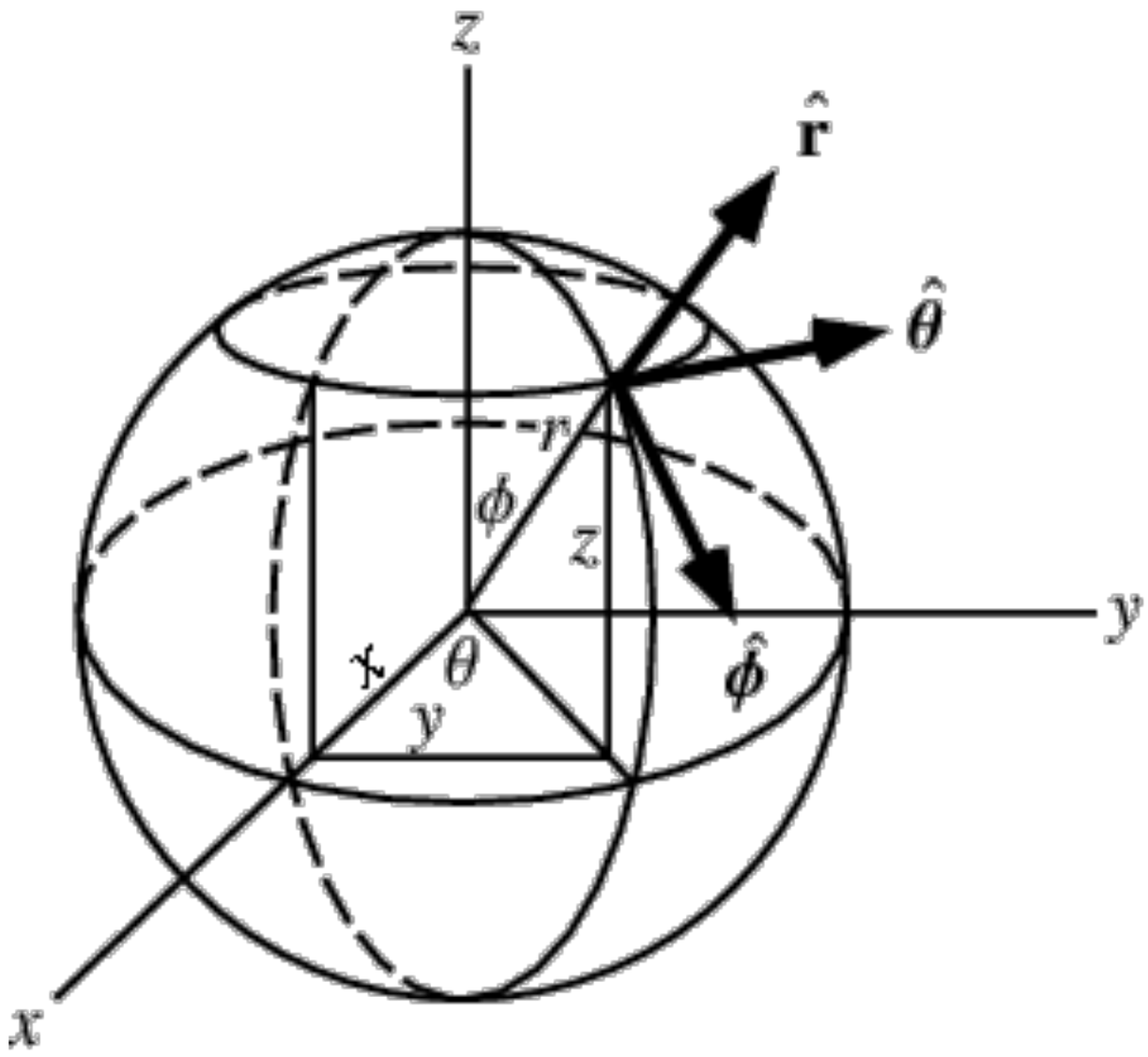


Erste Stufe

3 Dimensional



Zweite Stufe

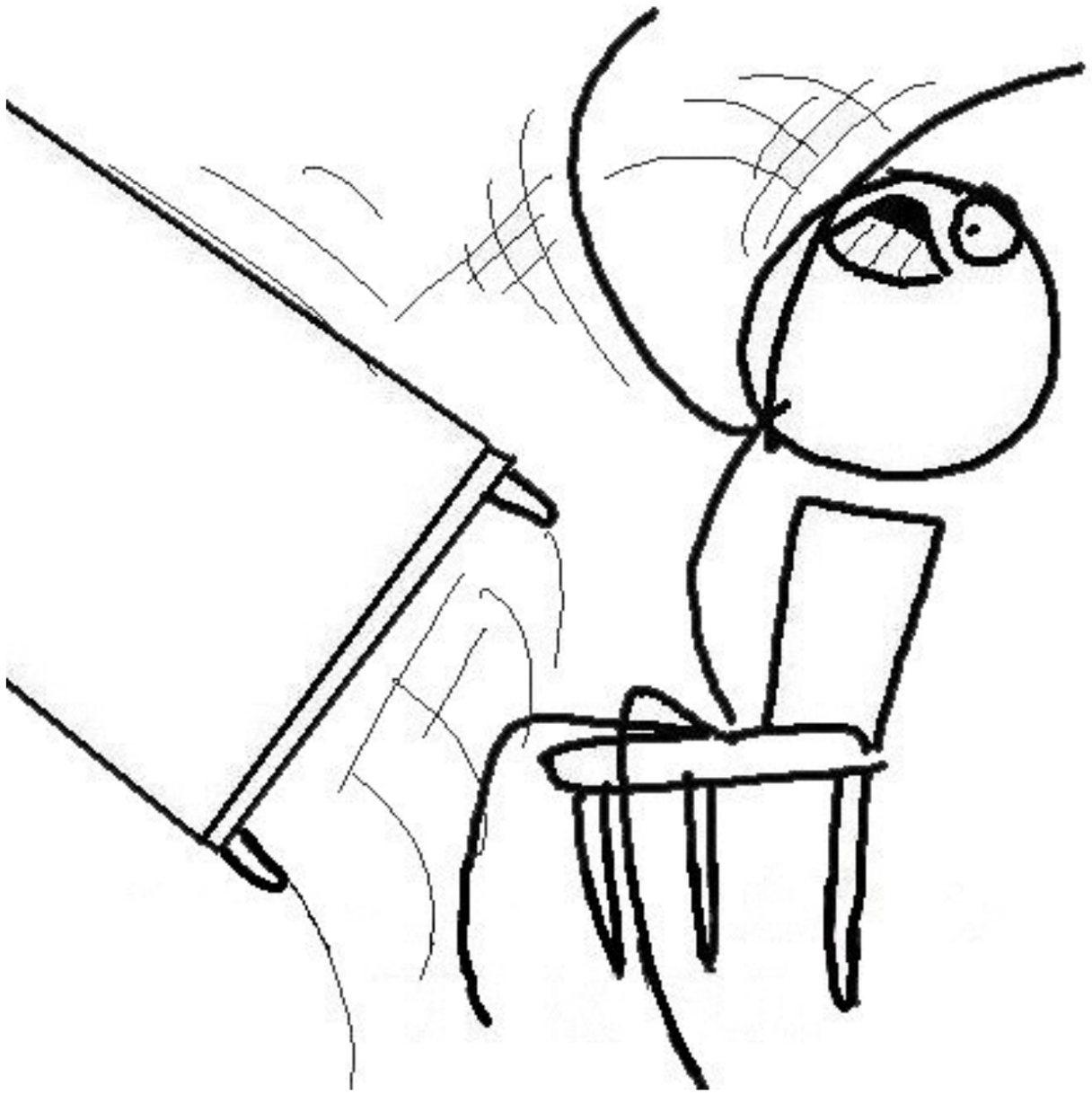


SphericalCoordinates

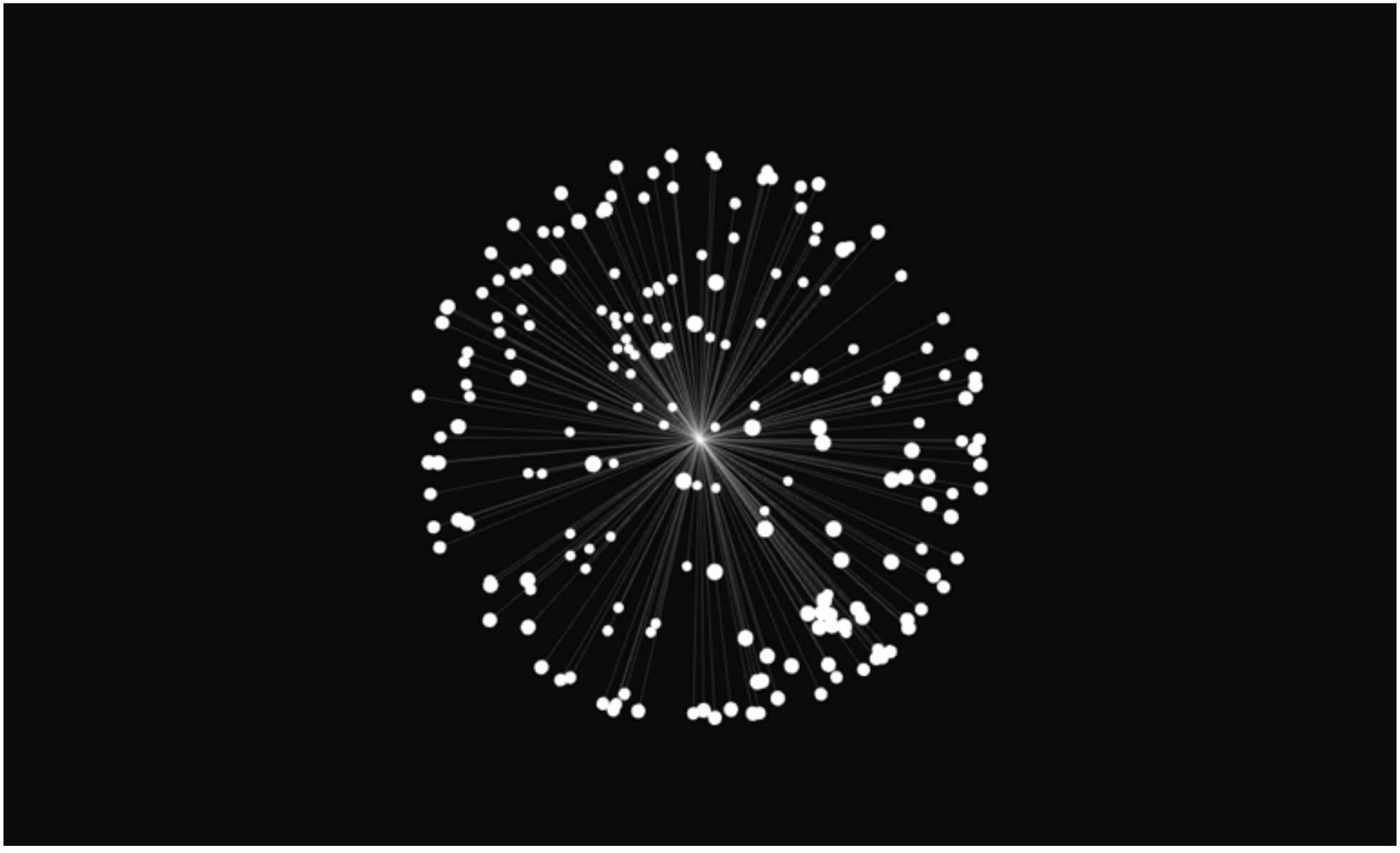
http://mathworld.wolfram.com/SphericalCoordinates.html

Cartesian Coordinates

- $x = r * \cos \theta * \sin \Phi$
- $y = r * \sin \theta * \sin \Phi$
- $z = r * \cos \Phi$



- vector<ofVec2f> **point3D**;
- point3DSetting();
    - float **_theta** = ofRandom(0, TWO_PI);
    - float **_pi** = ofRandom(0, PI);
    - float **_radius** = 200;
    - float **_x** = cos(**_theta**) * sin(**_pi**) * **_radius**;
    - float **_y** = sin(**_theta**) * sin(**_pi**) * **_radius**;
    - float **_z** = sin(**_pi**) * **_radius**;
    - ofVec3f **_point3D_temp** = ofVec3f(**_x**, **_y**, **_z**);
    - **point3D**.push_back(**_point3D_temp**);



Zweite Stufe