# Code journal: water droplet model

Advanced Computer Graphics 2020-1
*Mariana Ávalos Arce*

## 1 Introduction

The purpose of this document is to give an empirical approximation to a droplet's form for a water shader in Unity 3D. The equation is based on the idea of scaling a standard sphere radius so that the sphere looks like a droplet.

## 2 Droplet form

The idea was to start form a standard sphere with a constant radius, let's say 1. If the circunference radius is given, then we can also define a *top point*, that is (0, radius). For better visualization, I use Wolfram Mathematica 11.3 and a 2D approach:

```
circle = ParametricPlot[{Cos[t], Sin[t]}, {t, 0, 2 Pi}, PlotRange ->
{{-2, 2}, {-2, 2}}, AxesLabel -> {"x", "y"}]
top = Graphics[{Red, PointSize[0.03], Point[{0, 1}]}]
Show[circle, top]
```
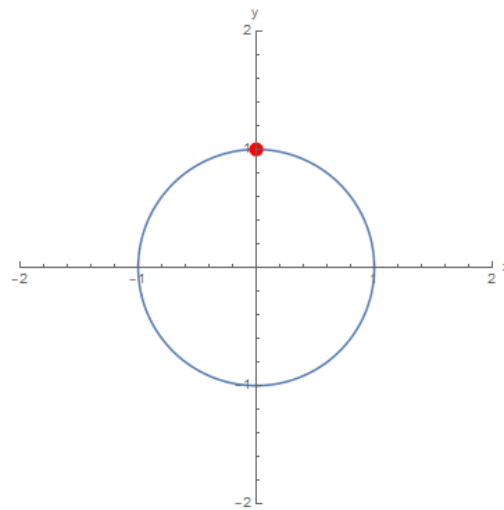


Figure 1: Circle with top point

I use polar coordinates because they make it easier to handle and see the fluctuations in a circle's radius. In order to create a droplet form out of a circle, I need to modify the *radius*. I use the top point as the basis for the droplet's radius: if I take each point of a well defined unitary circle and calculate the **distance** from **current point (cos(t), sin(t))** to **constant top point (0, radius)**, I will get a positive number. Then, that positive number

that represents distance is the **new radius** that will multiply `cos(t)`, assuming we want a vertical droplet, in order to scale the circle's width without moving its center.

```
fatness = 1.0
radius = 1
x0 = 0
y0 = radius
ParametricPlot[{(Sqrt[(x0 - radius * Cos[t])^2 + (y0 -
   radius * Sin[t])^2])* fatness* radius * Cos[t],
radius * Sin[t]}, {t, 0, 2 Pi}, PlotRange -> {{-2, 2}, {-2, 2}},
AxesLabel -> {"x", "y"}]
```
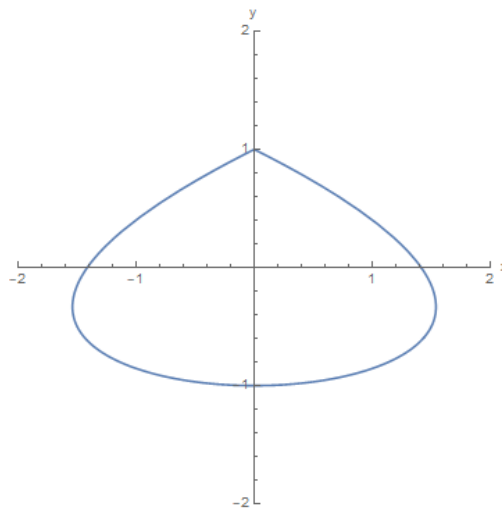


Figure 2: Droplet with distance as radius

Then, if we add a factor that determines how much strength will this new distance affect the original radius and this factor is 0.5 we get the desired approximation of a droplet.

```
fatness = 0.5
radius = 1
x0 = 0
y0 = radius
ParametricPlot[{(Sqrt[(x0 - radius * Cos[t])^2 + (y0 -
   radius * Sin[t])^2])* fatness* radius * Cos[t],
radius * Sin[t]}, {t, 0, 2 Pi}, PlotRange -> {{-2, 2}, {-2, 2}},
AxesLabel -> {"x", "y"}]
```
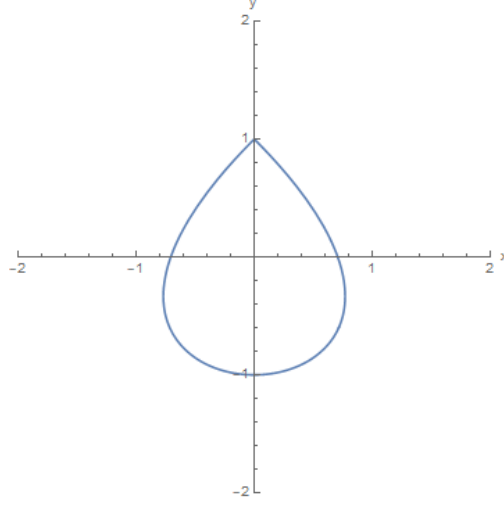
Figure 3: Droplet with distance as 0.5 influence on radius

I also noticed that the droplet form can be expressed in full polar form as:

```
circleRadius = 1
fatness = 1
r = fatness  * Sin[(circleRadius / 2) * t]
ParametricPlot[{ circleRadius * r * Sin[t],
circleRadius * Cos[t]}, {t, 0, 2 Pi},
PlotRange -> {{-2, 2}, {-2, 2}}, AxesLabel -> {"x", "y"}]
```
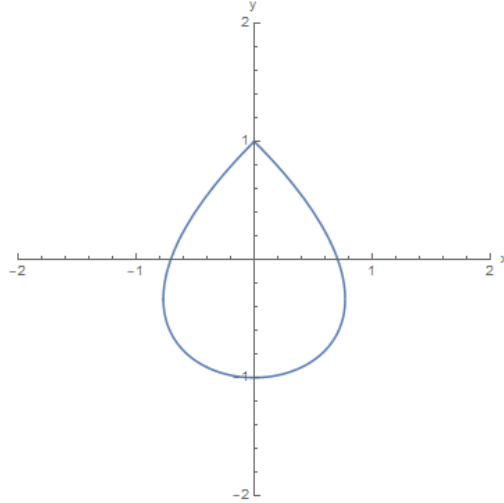


Figure 4: Droplet with x as sin(t/2)*sin(t)

In conclusion, the droplet form can be modeled by the following:

$$\overrightarrow{r}(t) = \left\langle \left( \sqrt{(x_0 - R cos(t))^2 + (y_0 - R sin(t))^2} \right) \times (\Phi R cos(t)), R sin(t) \right\rangle, 0 \leq t \leq 2\pi \quad (1)$$