

# Eemil Tarnanen

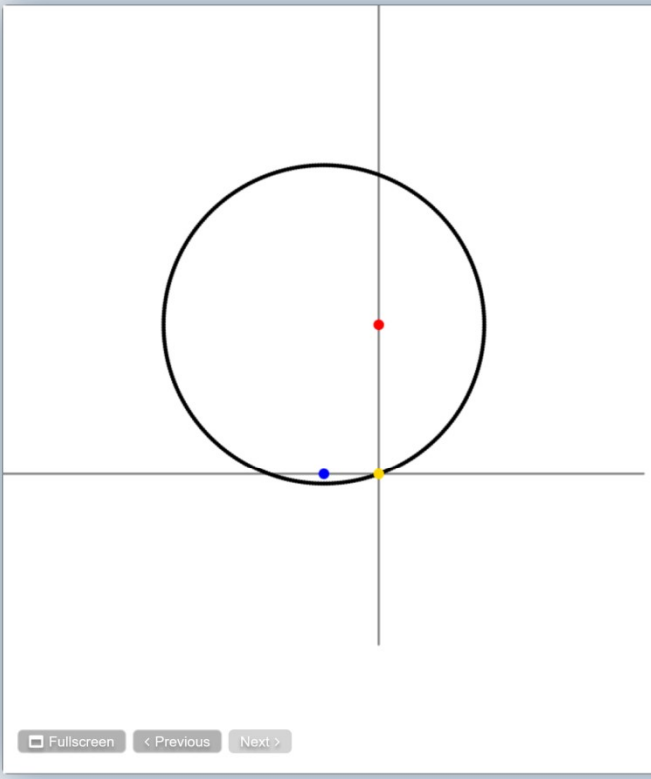
- High School Student from Finland

## **Modelling circle using sin and cos**

[jseden.dcs.warwick.ac.uk/construit/?load=182](https://jseden.dcs.warwick.ac.uk/construit/?load=182)

- A construal influenced by the Solar System construal

[jseden.dcs.warwick.ac.uk/construit/?load=39](https://jseden.dcs.warwick.ac.uk/construit/?load=39)



### Modelling ellipses

Circle isn't the only round thing this code can be used to create, as it can be used to create basically any kind of ellipse in any single spot!

That's why I want you to experiment with these variables down below and see how changing them changes the circle on the left. Have Fun!

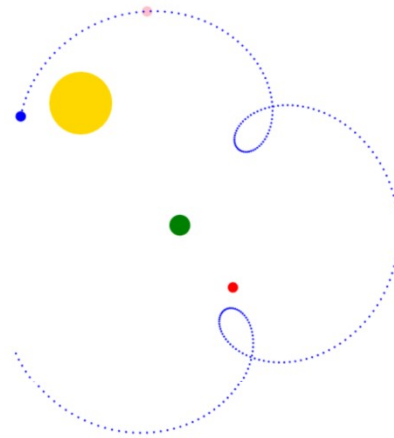
```
CenterX= 300;    X-coordinate of the circle
CenterY=300;    Y-coordinate of the circle
RadiusX=200;    X-Radius of the circle
RadiusY=150;    Y-Radius of the circle
```

Click the button here to reset the circle

```
CenterX= 300;    X-coordinate of the circle
CenterY=300;    X-coordinate of the circleY-coordinate of the circle
RadiusX=150;    X-coordinate of the circleX-Radius of the circle
RadiusY=150;    Y-Radius of the circle
```

And not only ellipses, this code is a great base on which to create all kinds of curves with just small changes to the X- and Y-value defining functions

```
XValue is CenterX + sin(5*tick/10) * RadiusX;
YValue is CenterY + cos(9*tick/10+216) * RadiusY;
```



### Show the Motion of Mercury

#### Mercury Trace

Display a memory of where Mercury has been as a trace of small circles

```
mercurysmall is Circle(mercuryX, mercuryY, 1, "blue");  
mercurytrace is mercurysmall with tick is (tick-memory)..tick;  
  
memory = 260;  
picture is [sun, earth, venus, mercury, mars]// mercurytrace;
```

#### Make Earth the Centre of the Solar System

Mercurys motion in the sky as observed from Earth is interesting, to see this in our Construal we need to make Earth the centre of the model and have the Sun rotate around us. This shows the mysterious motion of Mercury.

```
earthX is book_width/4;  
earthY is book_height/2;  
originX is earthX - sin(earthSpeed*tick) * earthDistance;  
originY is earthY - cos(earthSpeed*tick) * earthDistance;  
  
earthSpeed = 1;
```

# Features in common

- Trigonometric relationships
- Use of clocks
- Locus of moving object
- Use of presentation environment

Possible extensions to

modelling an oscilloscope, Lissajous figures

jseden.dcs.warwick.ac.uk/construit/?load=182&vid=3363

80% Search

Modelling circle using sin and cos

The start of visualization

In this page we visualize the circles made in the last page and show how the relation with circle occurs

To make any use of the circles we made last page, we need to make them visible using the following code

```
picture is [SinCircle, CosCircle];
```

Then, to make the circle happen we need to make the circles change location

```
do lib > clocks;
setedenclock(&tick, 30);
```

- tick is currently 35892
- XValue is currently 178.64745084375633
- YValue is currently 227.73694888474856

This movement still seems fairly random, so we shall create lines to represent the X of sin and Y of cosine, and show the movement of the intersection of those coordinates

```
SinLine is Line(XValue, 0, XValue, 600, "gray"); Line of Sine X-coordinate
CosLine is Line(0, YValue, 600, YValue, "gray"); Line of cosine Y-Coordinate
picture is [SinLine, CosLine, SinCircle, CosCircle];
```

Start to seem familiar, doesn't it. To further show that it actually is a circle, we shall trace the lines using this handy piece of code and add a circle to represent the cross section

```
Values is Text(SinX, 50, "SinX", "red");
CrossingMarker is Circle(XValue, YValue, 5, "gold");
CircleLine is Circle(XValue, YValue, 2, "black");
CircleFollow is CircleLine with tick is (tick-memory)..tick;
memory = 1627;
picture is [SinLine, CosLine, SinCircle, CosCircle, CircleFollow, C
```

And there it is. But circle isn't the only thing that can be modelled, so click "next" to see few more things

Fullscreen < Previous Next >

$XValue$  is  $CenterX + \sin(5 * tick / 10) * RadiusX$ ;  
 $YValue$  is  $CenterY + \cos(9 * tick / 10 + 216) * RadiusY$