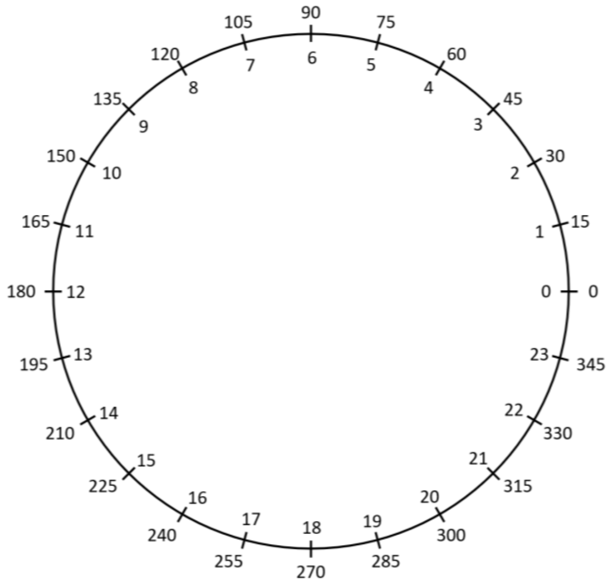
String Art on the Unit Circle

Student Activity

7 8 9 10 **11 12**

TI-Nspire™ Coding Student 50 min

Introduction



This activity is designed to build a series of beautiful mathematical visuals. The following mathematics is used to write the program:

- Modular Arithmetic
- Unit Circle (sine & cosine)
- Transformations

Modular Arithmetic: A simple way to think of modular arithmetic is 'remainder'. Try calculating mod(32,24) in the Calculator application.

In the circle (opposite) draw lines to connect the '2 times' table.

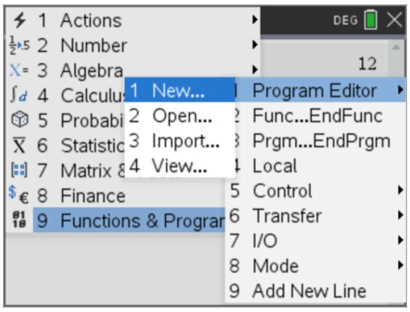
Example: 1 connects to 2, 2 connects to 4,
3 connects to 6 …

A problem arises for products involving numbers greater than or equal to 12. Modular arithmetic can be used to fix this problem: mod(#,24). Complete the mapping from 2×0 through to 2×23 to generate a 'low resolution' pattern. A program can be used to generate a much higher resolution of the pattern, and others in the family.

Building the Foundations

Start a new document and insert a basic program, call the program:

StringArt



If your program appears as a 'split page' press: **Ctrl** + **6** to ungroup the pages.

Author: P. Fox

The first step is to generate the 2 times table mapping. A set of numbers can be generated using a 'loop'. The start and end values are both known, so a "FOR" loop is appropriate.

Menu > **Control** > **For … EndFor**

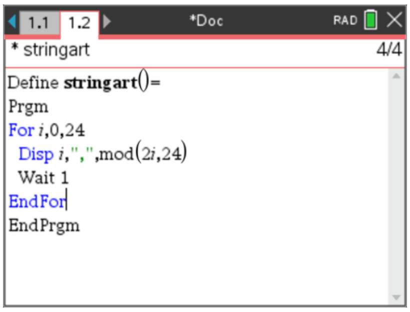
The "Disp" command can be used to display the numbers:

Menu > **I/O** > **Disp**

The 'wait' command can be added to slow the program down:

Menu > **Hub** > **Wait**

Press **Ctrl** + **R** to run the program.



The location of each point will be referenced using the unit circle.

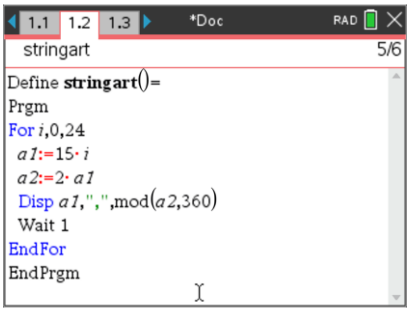
The 2 times tables values need to be converted to angles, the circle diagram includes these angles (degrees).

These values will be stored in variables:

a_1 (Angle for angle 1) & a_2, (angle for angle 2)

Edit the program to include these new variables and update the "Disp" command to show the angles.

- The modular arithmetic is performed for 360 (degrees)
- The multiplier: 15 comes from 360° ÷ 24 = 15°.



Transformations & the Unit Circle

Unlike the cartesian plane, the drawing canvas on TI-nspire has the origin located in the top left corner. The canvas dimensions are 318 × 212 pixels. To place the 'string art' diagram in the centre of the screen, a simple set of transformations can be performed:

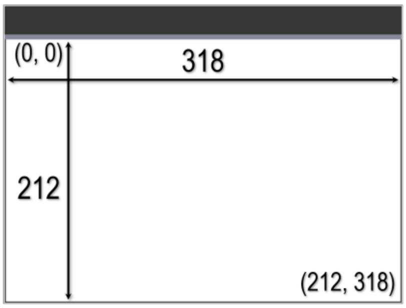
h := 159 (Horizontal Transformation)

v := 106 (Vertical Transformation)

A dilation can scale the 'unit' circle into something more appropriate:

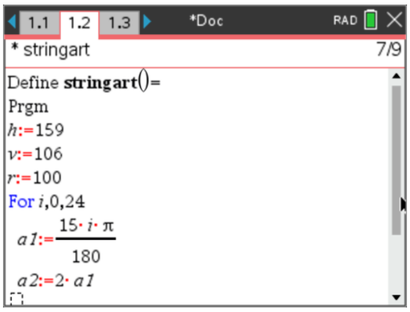
r := 100 (Dilation)

The transformations can be made at the start of the program. (See below right)



The angle can now be converted from degrees into radians.

a_1:=15 i π / 180 [Note that a_2 is simply double a_1]



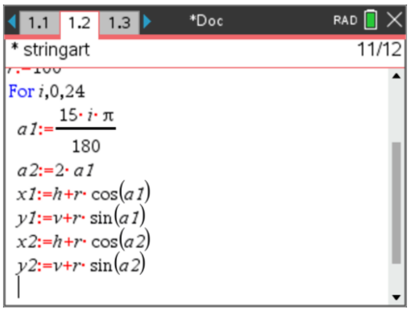
Each line consists of a starting point: (x_1, y_1) and end point: (x_2, y_2), for simplicity, these can be done individually:

$$x_1 \coloneqq h + r\cos(a_1)$$

$$y_1 \coloneqq v + r\sin(a_1)$$

$$x_2 \coloneqq h + r\cos(a_2)$$

$$y_2 \coloneqq v + r\sin(a_2)$$



Texas Instruments

Variables: x_1, y_1, x_2 and y_2 are used in the program. The same variables are aligned to 'parametric' equations in the Graphs application. These variables can be declared as "Local" at the start of the program, this means they only exist in the program. Alternatively, when graphing a parametric equation, start at: $x_3(t)$, $y_3(t)$

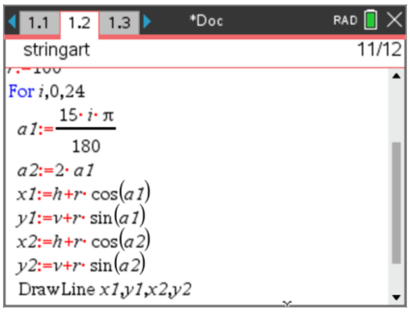
The last command is to draw the line on the canvas.

Menu > **Draw** > **Shapes** > **DrawLine**

The syntax for the DrawLine command is logical:

DrawLine x_1, y_1, x_2, y_2

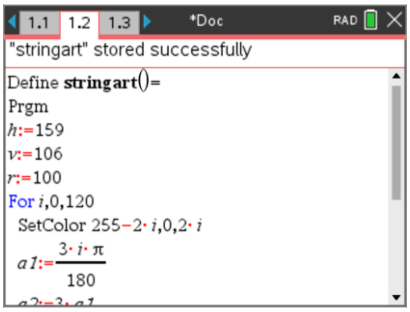
Press **Ctrl** + **R** to run the program.



Time to Explore

Now that your program has finished, try making a few simple changes:

- Change the "For" loop to count up to 120. Change the corresponding a_1 calculation then run the program. Explore different loop counts, with corresponding changes to the a_1 value.
- Explore what happens if the a_2 multiplier is changed from '2' to '3'.
- Just for fun: Change the line colour using the SetColor R, G, B command, (Red, Green, Blue). The patterns get even better if the colour depends on the loop count. The range of values for each colour can vary: [0, 255]



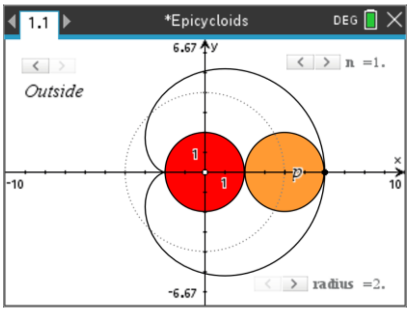
Investigation

The curves produced by these envelopes are called epicycloids:

- Cardioid [Greek: Kardia = Heart]
- Nephroid [Greek: Nefro = Kidney]
- Trefoiloid (and many more!) [Latin: Tri (3) folium (plant) Trefoil = 3 leaved plant]

Open the Epicycloid document, make sure the following settings have been selected: n = 1, radius = 2 and "outside" is toggled on.

- n = 1: Refers to the ratio between the radius of the two circles. Let the inner circle have radius R and the outer r.
- Radius = 2: Refers the radius of the static circle (centred at the origin)
- Outside/Inside: Refers to where the rolling circle is located.



For the following questions, assume that the two circles have the same radius (R = r)

Question: 1.

Point P is located at the centre of the outer circle. As the outer-circle traces around the inner circle, point P traces out a circle. Determine the parametric equations for this circle if P starts as shown above.

Question: 2.

As the outer-circle traces around the inner circle, how many times will the outer-circle rotate?

Question: 3.

Use your answers to the previous two questions to determine the parametric equations for the path of a point on the outer circle as the outer circle traces around the inner circle.

Question: 4.

Use calculus to help determine the exact values for the extreme points in both the x and y direction.

Question: 5.

Show that the cartesian equation is given by:

$$x^4 + 2x^2y^2 - 24x^2 - 64x + y^4 - 24y^2 - 48 = 0$$

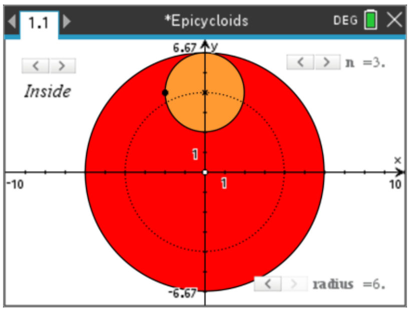
Question: 6.

Explore a calculus approach to determining the extreme points in both the x and y direction using the cartesian relation from Question 5. Compare the parametric and cartesian approaches.

Extension

The string art examples produce curves that can be modelled by a rolling one circle around the outside of another. Imagine however that the smaller circle rolls around the inside of the larger circle.

The epicycloids file allows you to explore this phenomenon. Adjust the toggle in the top left corner of the screen, then set n = 3 and radius = 6. The dotted line displays the path for the centre of the smaller circle as it moves around the inside of the larger circle. Watch closely for details such as the direction of rotation of the smaller circle.



Question: 7.

Determine the parametric equations for the path of a point on the smaller circle as it moves around the larger circle.

Question: 8.

Determine the exact values for the cusps on the curve.

Question: 9.

Explore the gradient of the function as it approaches each cusp.

Author: P. Fox