**Requirements**

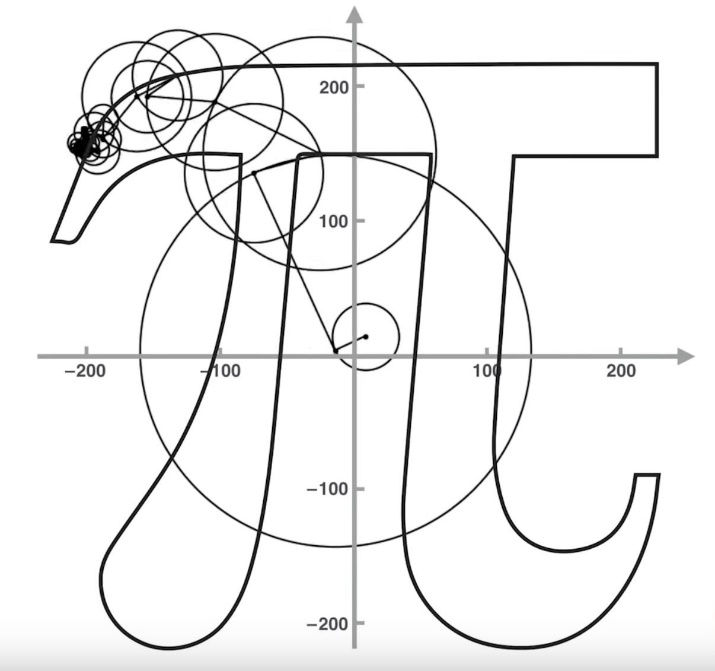
* Users can draw a path on a canvas. To complete the path, user should have two options:
  + Finish the loop by adding a transparent line in between starting point and last drawn point.
  + Finish the loop by adding a solid line in between starting point and last drawn point.
* Users can load a path from outside the program in a suitable format and it would show up on the canvas.
* Users can save the path on the canvas in a suitable format.
* Users can reset the path drawn on the canvas.
* Users can be done with the path and see the corresponding Fourier series representation of the path as circles.
* Obviously, a GUI should be used. Drawing window and result window should be separate.

**Constraints**

* There can only be one closed loop of path. So, when user is done with drawing the first path, it must reset before drawing again.
* The finishing line that is added after completing the path should be continuous and differentiable with both its neighboring lines.
* Once the drawing process is started, user cannot leave until the path is completed.
* Drawing window can be resizable with a minimum size in place but size of result window (which should be the same as canvas size in drawing window) cannot change after the user starts the animation.

**Intended Fourier Series Representation**

* Starting from a base point, computationally determined sized circles are placed on far point of each circles’ radii, which is spinning around said circle in with a determined period. Linear combination of each of these radius vectors over the entire length of the path should be equal / or close to the path itself. Like the photo [1] below:



* If there are transparent lines on the path, the Fourier representation should still follow that path, but it should not trace it on the resulting view.

**References**

[1] https://dsp.stackexchange.com/questions/59068/how-to-get-fourier-coefficients-to-draw-any-shape-using-dft