Case study I.3–Computer graphics

First, you shall need ready-to-be-used in your programming language the following:

I) A function, for concreteness called `Visualise`, which receives an array of size $5 \times N$, the five entries of each column are (x, y, r, g, b), where (x, y) are 2D coordinates of a point and (r, g, b) is its color. The command plots these points on a square canvas of fixed size.

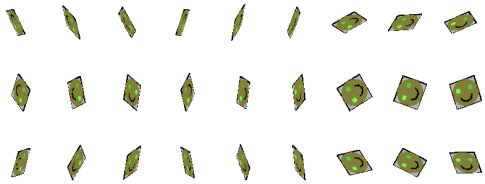
II) An array (with 2000 points at most) implementing a simple picture, to be fed to Visualise. The picture should be asymmetric (i.e. a circle is not ok) and it should occupy a square whose perimeter is twice smaller than the canvas' perimeter.

We prepared for you the function and an array example realised for Python and Mathematica, you can find them in studium. If you wish you can use them or create your own versions.

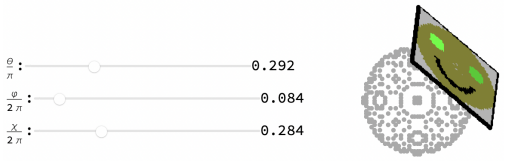
The images are to be produced using only `Visualise`. It is not allowed to use some other functions from graphical packages (except probably for creating function `Visualise` itself).

Q4.1. Rotate the picture by any angle φ in 2D (Note: it is convenient to represent $\varphi = 2\pi t$, $t \in [0, 1]$).

Q4.2. Rotate the picture around any axis φ in 3D, also do combination of rotations around different axes. The agreement is that the z axis is perpendicular to the screen pointing at you.



Q4.3. Let one has a two-dimensional sphere S^2 given by the equation $x^2 + y^2 + z^2 = r^2$, $r > 0$. Point on the sphere can be parameterised using spherical coordinates θ, φ. Draw the picture tangent to S^2 at given θ, φ, it should be possible as well to rotate this picture by an arbitrary angle, say ψ, in the tangent plane.



Q4.4. Assemble 6 pictures into a cube. The cube, as a whole, should be rotatable around the three axes as well.