Code (Tuesday Week 2)

Using the same `ShapeGraphics` library found in this week's exercise.

Ellipses Example

Haskell

```haskell
module Ellipses where

-- needed to display the picture in the playground
import Codec.Picture

-- our line graphics programming interface
import ShapeGraphics

simpleEllipsePic :: Float -> Picture
simpleEllipsePic n
      = map greenEllipse
             [0, pi/n .. (n-1)*pi/n]
  where
    centre :: Point
    centre = Point 400 400

    greenEllipse :: Float -> PictureObject
    greenEllipse angle
        = Ellipse centre 250 70 angle
                  (colourFor angle)
                  Solid SolidFill

    colourFor angle
        = let x = round (255 * angle / pi)
           in Colour (255 - x) 128 x 84



writeToFile pic
  = writePng "output.png"
         (drawPicture 3 pic)
```

Curves example

```haskell
module Curves where
```

```haskell
-- needed to display the picture in the playground
import Codec.Picture

-- our line graphics programming interface
import ShapeGraphics

curves :: Picture
curves = sinCurve ++ cosCurve
  where

    xvals = [0,10..780]
    start = Point 20 380

    moveBy func ampl x
     = movePoint
        start
        (Vector x
                (ampl*func((pi * x)/200)))

    makeShape func ampl x
       = Circle (moveBy func ampl x) 10
           red Solid SolidFill

    sinCurve = map (makeShape sin 99) xvals
    cosCurve = map (makeShape cos 50) xvals


movePoint :: Point -> Vector -> Point
movePoint (Point x y) (Vector dx dy)
    = Point (x + dx) (y + dy)
writeToFile pic
  = writePng "output.png"
       (drawPicture 3 pic)
```