Super Target

A user will be presented with a menu. They can choose to draw a target or run ‘Super Target’.

1. Super Target Mode: This mode asks the user for a number of targets to draw. It will then randomly draw that many targets in random places and of random sizes.
2. Single Target Mode: This mode asks the user for three items
   1. Location: x, y location of the center of the target
   2. Radius: this is the radius of the bullseye
   3. Rings: the number of rings the target has, including the bullseye (A target with 1 ring is simply a bullseye)

After drawing the user will be asked if they want to play again with the following options:

1. Play again and reset the drawing
2. Play again and keep the drawing
3. Quit

Code

1. Start the program in main.py
   1. This file should handle
      1. User input
      2. Calling module functions
   2. This file should not
      1. Perform arithmetic calculations
   3. Must contain the following code:

**if** radius == **"" and** rings == **""**:  
 target.drawTarget(centerX, centerY)  
**elif** rings == **""**:  
 target.drawTarget (centerX, centerY, eval(radius))  
**elif** radius == **""**:  
 target.drawTarget (centerX, centerY, rings=eval(rings))  
**else**:  
 target.drawTarget (centerX, centerY, eval(radius), eval(rings))

1. Write a module called ‘target’ that contains the following functions
   1. reset()
      1. Erase all of the targets and start over
   2. setup()
      1. Configure turtle to draw quickly
      2. Configure turtle to have a window of 1000 x 800
   3. drawSuperTarget()
      1. Use appropriate parameters
      2. Draw random size/location/ring targets
      3. It should call drawTarget()
   4. drawTarget()
      1. Use appropriate parameters
      2. Draw a single target
      3. Use default parameters if the user does not enter values
         1. Radius: 50
         2. Rings: 3
      4. It should call drawCircle()
   5. drawCircle()
      1. Use appropriate parameters
      2. Draw a single circle
   6. done()
      1. Called when the user quits
      2. Keeps the drawing on the screen