Using Recursion in Graphics

- Recursion can be a powerful tool
- Closely related to *Fractals*
  - Self-similarity
  - Keep zooming in: still looks the same
- Can produce very interesting figures with very little input
- Serpinsky Gasket is just a lot of triangles
  - Define recursively
Serpinsky Gasket

- Start with triangle

- Then put (1/2 size) triangles within triangle
Serpinsky Gasket

- Continue process with $\frac{1}{4}$ sized triangles, etc

- *Insight:* use *Serpinsky Gaskets* instead of triangles
Rendering a Serpinsky Gasket

- Mathematically, Gasket is defined for infinitely small triangle.
  - Goes on forever
  - Zoom in as far as you like: always the same picture

- In *drawing* a Serpinsky Gasket what are the issues?
  - Time to draw
  - What can you see

- How do we handle this potentially infinite recursion?
  - What to use as the base case?
  - ???
Serpinsky Demo

- In code directory
  - Using Applet
  - Run Serpinsky.html
- Note feature to slow down drawing
  - Get better sense of how recursive calls work
  - Also see how incredibly fast computer is...
- Review recursive features
  - What is done in the base case?
- What would figure be like if we drew nothing except
  - In the base case?
Classwork/Lab

- Will be doing two different figures recursively
  - Target
  - Circle Art
- For each, will use 2 approaches

1. **One Object: Draw Recursively**
   - Our drawing technique will use recursion
2. **Object Creates Other Object Recursively**
   - Each object will create “clone” objects using `new`
   - Each of smaller size and in different positions
   - Will invoke the `paint` methods of these clones