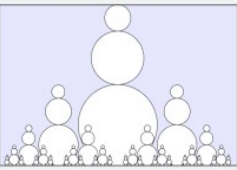


Recursive Drawing Functions

- additive pattern – each level adds more to the design
- replacement pattern – each level replaces the previous level

Implementing the Additive Pattern



what is the whole design? a family of snowmen
 what is one copy of the design? one snowman (at the top level there is one snowman)
 what differs from one copy of the design to the next? position and size

```

// draw a line of snowmen
// (x,y) is the center of the bottom circle
// w is the width of the snowman
void drawSnowFamily ( int x, int y, int w ) {
  if ( w < 10 ) { // "small enough" is less than 10 pixels
    return;
  } else {
    // draw one snowman
    ellipseMode(CENTER);
    fill(255);
    stroke(0);
    ellipse(x, y, w, w);
    ellipse(x, y-(w/2+w/3), 2*w/3, 2*w/3);
    ellipse(x, y-(w/2+2*w/3+w/6), w/3, w/3);
    // draw two smaller copies
    drawSnowFamily(x-(w/2+w/4), y+w/4, w/2);
    drawSnowFamily(x+(w/2+w/4), y+w/4, w/2);
  }
}
  
```

To draw the design, draw the shapes that make up one copy of the design, then draw the "smaller" copies.

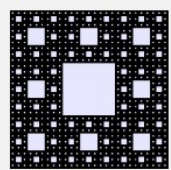
– "smaller" generally means a decrease in size, but could also be a countdown or some other thing

```


if ( the design is small enough ) {
  return; // we're done (draw nothing)
} else { // draw the design
  draw the shapes for one copy of the design
  draw the smaller copies of the design ← recursive calls
}
  
```

the design is small enough – based on the size of the snowman – w parameter
 draw the shapes for one copy of the design – one copy is one snowman, with the position and size indicated by the parameters
 draw the smaller copies – there are two smaller snowmen on either side of the one

Implementing the Replacement Pattern



what is the whole design? Sierpinski carpet
 what is the base shape? a black square
 what differs from one copy of the base shape to the next? position and size



```

// draw a sierpinski carpet
// (x,y) is the upper left corner of the carpet
// w is the size of the carpet
void drawCarpet ( float x, float y, float w ) {
  if ( w < 5 ) // "small enough" is less than 5 pixels
    // a simple carpet is just a black rectangle
    fill(0);
    rect(x, y, w, w);
  } else {
    // draw a carpet in each of the 8 subregions around the edge
    // (leave the middle region empty)
    drawCarpet(x, y, w/3);
    drawCarpet(x+w/3, y, w/3);
    drawCarpet(x+2*w/3, y, w/3);
    drawCarpet(x, y+w/3, w/3);
    drawCarpet(x, y+2*w/3, w/3);
    drawCarpet(x+w/3, y+2*w/3, w/3);
    drawCarpet(x+2*w/3, y+2*w/3, w/3);
  }
}
  
```

To draw the design, draw the "smaller" copies.

– "smaller" generally means a decrease in size, but could also be a countdown or some other thing

```

if ( the design is small enough ) {
  draw the base shape
} else {
  draw the smaller copies of the design ← recursive calls
}
  
```

the design is small enough – based on the size of the base shape – w parameter
 draw the base shape – one copy is one black square, with the position and size indicated by the parameters
 draw the smaller copies – there are smaller copies in each of the 8 subregions around the edge

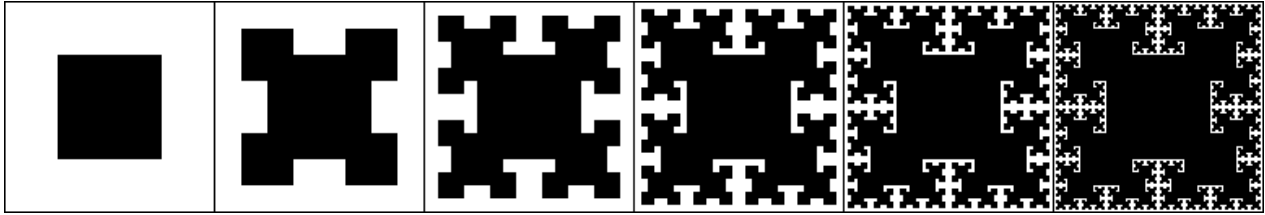
At the End of Class

Hand in whatever you have done during class, even if a sketch is incomplete.

- Make sure each sketch is named as directed and has a comment with the names of your group. Also be sure to save your sketches! (in Linux, this should be in your sketchbook `~/cs120/sketchbook`)
- Copy the entire directory for each sketch (not only the .pde file) into your handin directory (`/classes/cs120/handin/username`). You only need to hand in one copy for the group. (If you are running Processing on your computer instead of using the Linux virtual desktop, you will need to use FileZilla to copy the sketches.)

Exercises

1. Create a new sketch called **sketch_241011a** which draws a T-square fractal.



This follows the additive pattern.

- One copy of the design is a black square.
- There are four smaller copies of the pattern, each half the size and centered at the four corners of the square.

What differs from one copy of the design to the next? The position and size of the black square.

2. Create a new sketch called **sketch_241011b** which draws a Sierpinski gasket.



This follows the replacement pattern.

- The base shape is a black triangle.
- There are three smaller copies of the pattern, each half the size and arranged in the corners of the current region.

What differs from one copy of the base shape to the next? The position and size of the triangle. Hint: you could make the parameters to the drawing function be the three points of the triangle, but a simpler representation is to imagine the rectangle that just contains the triangle and to have its position (upper left corner) and size be the parameters.

3. Modify the sketch from #1 so that the squares get progressively lighter-colored with each level.

If you have time, create new sketches **sketch_241011c** and **sketch_241011d** which draw the designs shown. (The one on the left is known as the Koch snowflake, shown after three levels.) Hint: first identify the applicable pattern (additive or replacement).

