More Shapes

- We’ve discussed rect, triangle, and ellipse already
- Processing has several other functions that can be used to draw various shapes
  - Will be presented throughout lecture
Lines

\[(x_1, y_1) \rightarrow (x_2, y_2)\]

\[\text{line}(x_1, y_1, x_2, y_2)\]
Example: House

- Let’s use these various functions to draw a picture of a house using processing
- Our goal should be to make it look like the picture to the left
Example: House

- Let’s use these various functions to draw a picture of a house using processing
- Our goal should be to make it look like the picture to the left
- **What shapes do we need?**
Example: House

size(400, 400);

rect(50, 200, 300, 150);  // house frame
rect(120, 280, 50, 70);   // door
rect(200, 280, 100, 50);  // window frame
line(200, 305, 300, 305); // window line
line(250, 280, 250, 330); // window line
triangle(30, 200, 200, 100, 370, 200);  // window
Example: House

Notice Each line ends with a ;

size(400, 400);
rect(50, 200, 300, 150); // house frame
rect(120, 280, 50, 70);   // door
rect(200, 280, 100, 50);  // window frame
line(200, 305, 300, 305); // window line
line(250, 280, 250, 330); // window line
triangle(30, 200, 200, 100, 370, 200);  // window
Color

- Now we have a house, great!
- But the colors are boring
- How can the color be changed of…
  - The inside of each shape?
  - The line/border of each shape?
  - The background?
Color

- A computer screen is composed of a grid (rows and columns) of pixels
  - … Like the canvas of a processing window
- Each pixel can display a unique color, which is controlled by a Red, Green, and Blue brightness value
Color

- Thus, when we specify to a computer program, we must do so using these three numbers
- The brightness values for each can be between 0 (the darkest) and 255 (the highest)
  - 0-255 inclusive
- Can use the `background()`, `fill()`, and `stroke()` functions to control the colors of things
Color

- **background(r, g, b)**
  - Control the color of the background of the canvas
- **fill(r, g, b)**
  - Control the color of all shapes drawn after
- **stroke(r, g, b)**
  - Control the color of all shape borders drawn after
Color Selector

- How does one determine what R, G, B values produce a color?
Color Selector

- How does one determine what R, G, B values produce a color?
- One way for beginners: use the color picker!
Color Selector

- How does one determine what R, G, B values produce a color?
- One way for beginners: use the color picker!
Use the color picker to determine

A. What color is (255, 0, 0) ?

B. What color is (100, 200, 255) ?

C. What color is (150, 200, 0) ?
Solutions

A. What color is (255, 0, 0) ?

A. What color is (100, 200, 255) ?

B. What color is (150, 200, 0) ?
Use the color picker to ...

A. Match this color:  

B. Match this color:  

C. Match this color:  
Solutions

A. RGB values for: 118, 253, 138

B. RGB values for: 255, 251, 55

C. RGB values for: 201, 253, 253
Example: House

- Let’s revisit the black/white house
Example: House

- Let’s revisit the black/white house
- Using the function we just learned about, how can we change this . . .
Example: House

- Let’s revisit the black/white house
- Using the function we just learned about, how can we change this . . .
- To this?
Add color to the house

• Add color to the house as pictured
• Use the functions we just discussed:
• `background(r, g, b)`
  ○ Control the color of the background of the canvas
• `fill(r, g, b)`
  ○ Control the color of all shapes drawn after
• `stroke(r, g, b)`
  ○ Control the color of all shape borders drawn after
background(100, 200, 255);
size(400, 400);

color(150, 0, 0);
rect(50, 200, 300, 150);  // house frame
color(0, 200, 0);
rect(120, 280, 50, 70);  // door
color(0, 0, 255);
rect(200, 280, 100, 50);  // window frame
line(200, 305, 300, 305);  // window line
line(250, 280, 250, 330);  // window line
color(255, 255, 0);
triangle(30, 200, 200, 100, 370, 200);  // roof
background(100, 200, 255);
size(400, 400);

fill(150, 0, 0);
rect(50, 200, 300, 150); // house frame
fill(0, 200, 0);
rect(120, 280, 50, 70); // door
fill(0, 0, 255);
rect(200, 280, 100, 50); // window frame
line(200, 305, 300, 305); // window line
line(250, 280, 250, 330); // window line
fill(255, 255, 0);
triangle(30, 200, 200, 100, 370, 200); // roof
background(100, 200, 255);
size(400, 400);

strokeWeight(5);
fill(150, 0, 0);
rect(50, 200, 300, 150);  // house frame
fill(0, 200, 0);
rect(120, 280, 50, 70);  // door
fill(0, 0, 255);
rect(200, 280, 100, 50);  // window frame
line(200, 305, 300, 305);  // window line
line(250, 280, 250, 330);  // window line
fill(255, 255, 0);
triangle(30, 200, 200, 100, 370, 200);  // roof
background(100, 200, 255);
size(400, 400);

strokeWeight(10);
fill(150, 0, 0);
rect(50, 200, 300, 150); // house frame
fill(0, 200, 0);
rect(120, 280, 50, 70);  // door
fill(0, 0, 255);
rect(200, 280, 100, 50); // window frame
line(200, 305, 300, 305); // window line
line(250, 280, 250, 330); // window line
fill(255, 255, 0);
triangle(30, 200, 200, 100, 370, 200); // roof
background(100, 200, 255);
size(400, 400);

strokeWeight(10);
stroke(0, 255, 100);
fill(150, 0, 0);
rect(50, 200, 300, 150);  // house frame
fill(0, 200, 0);
rect(120, 280, 50, 70);  // door
fill(0, 0, 255);
rect(200, 280, 100, 50);  // window frame
line(200, 305, 300, 305); // window line
line(250, 280, 250, 330); // window line
fill(255, 255, 0);
triangle(30, 200, 200, 100, 370, 200);  // roof
Alpha Values

- The `fill()` and `stroke()` functions can also take a fourth argument (value) when you call it.
- This is called the *alpha* value.
- Also a number between 0 and 255.
- Specifies the transparency level.
  - 0 is lightest (don’t display).
  - 255 is darkest.
Three Circles

- How can an image like this be created?
- What shapes and features of processing should be used?
Three Circles

background(204, 226, 225);
size(400, 150);
noStroke();
fill(255, 0, 0, 80);
ellipse(132, 82, 200, 200);
fill(0, 255, 0, 80);
ellipse(228, -16, 200, 200);
fill(0, 0, 255, 80);
ellipse(268, 118, 200, 200);
Custom Shapes

- You are not limited to the “basic” shapes that we’ve been using so far
- Processing supports the drawing of custom shapes using `beginShape()`, `vertex()`, and `endShape()`
  - Start by calling `beginShape()`, then `vertex(x,y)`. This specifies the beginning point of a drawing.
  - Call `vertex(x,y)` again to specify the next x and y to draw to
  - Repeat for the next vertex (the next x and y)
  - `endShape()` when done
Consider drawing a box:

```plaintext
size(120, 120);
background(100, 238, 255);
beginShape();
fill(255, 255, 100);
vertex(30, 20); // upper-left corner
vertex(85, 20); // upper-right corner
vertex(85, 75); // lower-right corner
vertex(30, 75); // lower-left corner
endShape();
```
Create a custom shape

- Create a custom shape that looks as follows
- Set size to: `size(500, 500);
- Recall:
  - `background(r, g, b);
  - `fill(r, g, b);
  - `beginShape();
    - `vertex(x, y);
    - ...
  - `endShape();
Create a custom shape

```javascript
size(500, 500);
background(100, 200, 255);
fill(255, 255, 0);
beginShape();
vertex(200, 100);
vertex(100, 400);
vertex(400, 400);
vertex(300, 100);
vertex(200, 100);
endShape();
```
More Shapes

quad(x1, y1, x2, y2, x3, y3, x4, y4)
More Shapes

arc(x, y, width, height, start, stop)
Exercise

- Create this arrow using a custom shape as described on the last slide
background(204, 226, 225);
size(480, 150);

strokeWeight(4);
fill(255, 0, 0, 80);
beginShape();
vertex(180, 82);
vertex(207, 36);
vertex(214, 63);
vertex(407, 11);
vertex(412, 30);
vertex(219, 82);
vertex(226, 109);
vertex(180, 82);
endShape();
Exercise

- How can we create this shape, using the functions we just learned about?
size(300, 300);
background(200, 100, 100);
strokeWeight(7);
stroke(10, 100, 150);
fill(150, 250, 250);
beginShape();
vertex(50, 50);
vertex(250, 50);
vertex(160, 150);
vertex(250, 250);
vertex(50, 250);
vertex(140, 150);
vertex(50, 50);
endShape();
The Shape of Edges

- You can change the style of the edges of a shape using the `strokeJoin()` function
  - `strokeJoin(ROUND);`
  - `strokeJoin/MITER);`
  - `strokeJoin(BEVEL);`
Processing - Materials

- Required Materials
  - GSWP - Chapter 3