PennDraw

CIS 110

Harry Smith

Table of Contents

- Goals
- What is PennDraw?
- Working through the MyHouse.java example

Goals

- Understand the abstract model of how PennDraw translates code into shapes on a screen
- Practice reading code syntax and semantics
  - Syntax: “what do the individual letters and numbers cause the computer to do?”
  - Semantics: “what are the effects and meanings of the code that has been written?”

What is PennDraw?

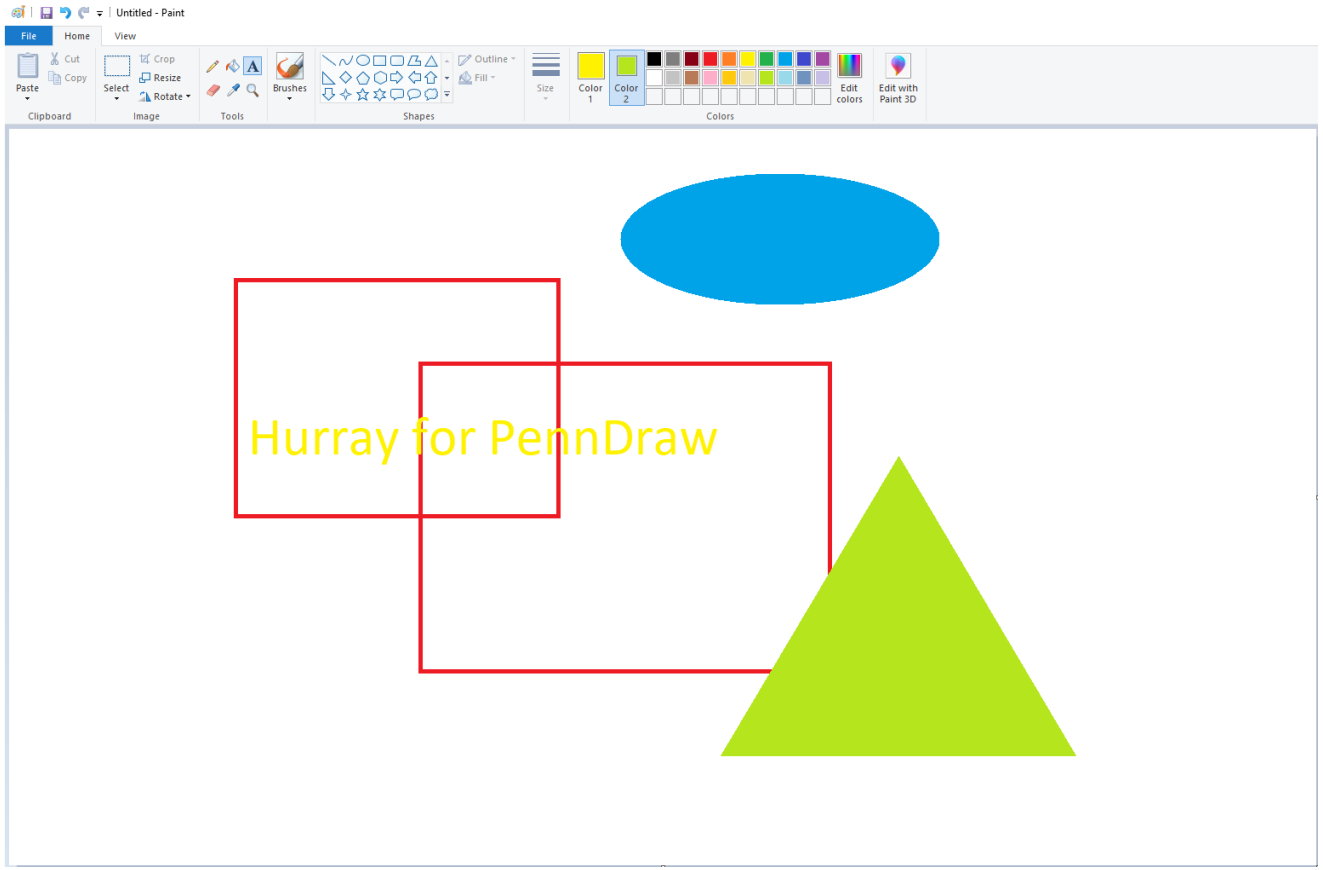
PennDraw

- The name of a group of related drawing tools available for you to use
  - Adapted from a library called “StdDraw” if you see that anywhere
- Any time we need to draw to the computer’s screen in CIS 110, we’ll use PennDraw.
- You can access a full listing of PennDraw’s features on the page for **PennDraw** on the course website

PennDraw: a programmable Microsoft Paint

Features:

- Draw over a set canvas
- Has an imaginary “pen”
  - The pen has a color setting and a weight setting
- Draw shapes
  - Rectangles, ellipses, arbitrary polygons
- Draw text



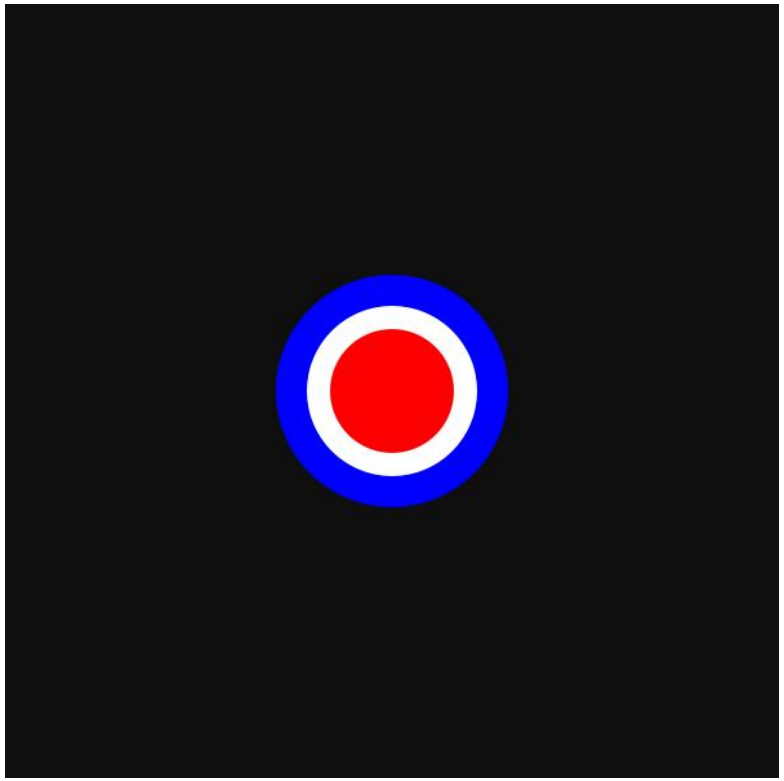
Microsoft Paint in 2021

A PennDraw Program and its Output

The Program

```
public class OrderDemo {
    public static void main(String[] args) {
        PennDraw.setCanvasSize(600, 600);

        PennDraw.clear(15, 15, 15);

        PennDraw.setPenColor(PennDraw.BLUE);
        PennDraw.filledCircle(0.5, 0.5, 0.15);

        PennDraw.setPenColor(PennDraw.WHITE);
        PennDraw.filledCircle(0.5, 0.5, 0.11);

        PennDraw.setPenColor(PennDraw.RED);
        PennDraw.filledCircle(0.5, 0.5, 0.08);
    }
}
```

The Output

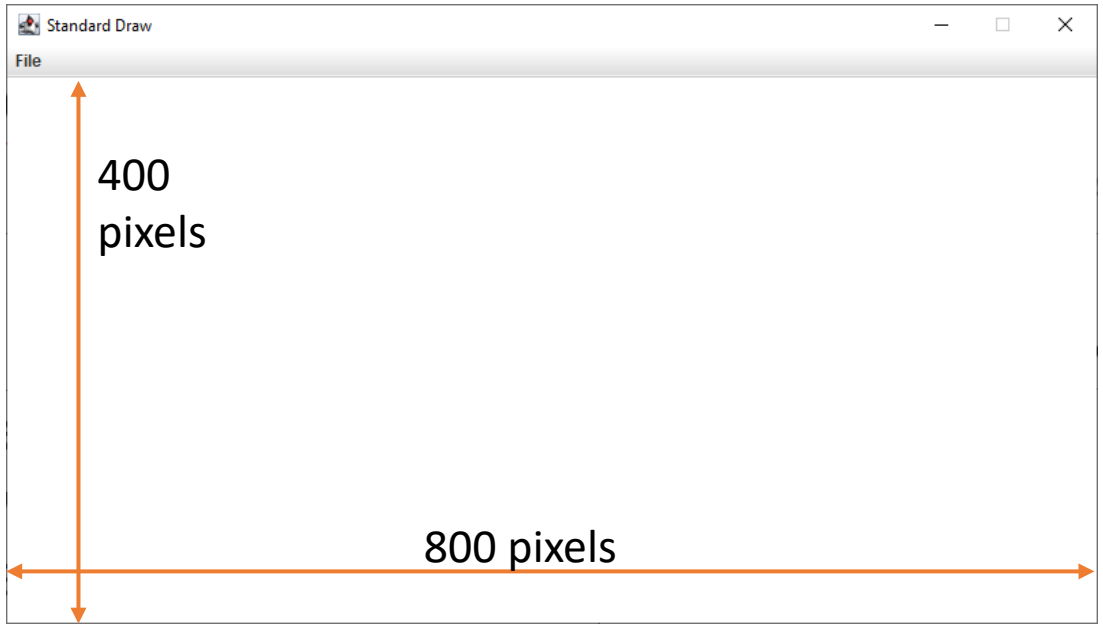


Penn Engineering
University of Pennsylvania

PennDraw: the Canvas

- The *canvas* refers to the window of space on which PennDraw can do its drawing
- It has a width and a height, both defined in pixels.
  - We usually express the size of the canvas like *width* x *height*.
  - Width is the “x dimension”
  - Height is the “y dimension”

A canvas of size 800x400



PennDraw: the Coordinate System

Canvas positions are accessed using coordinates

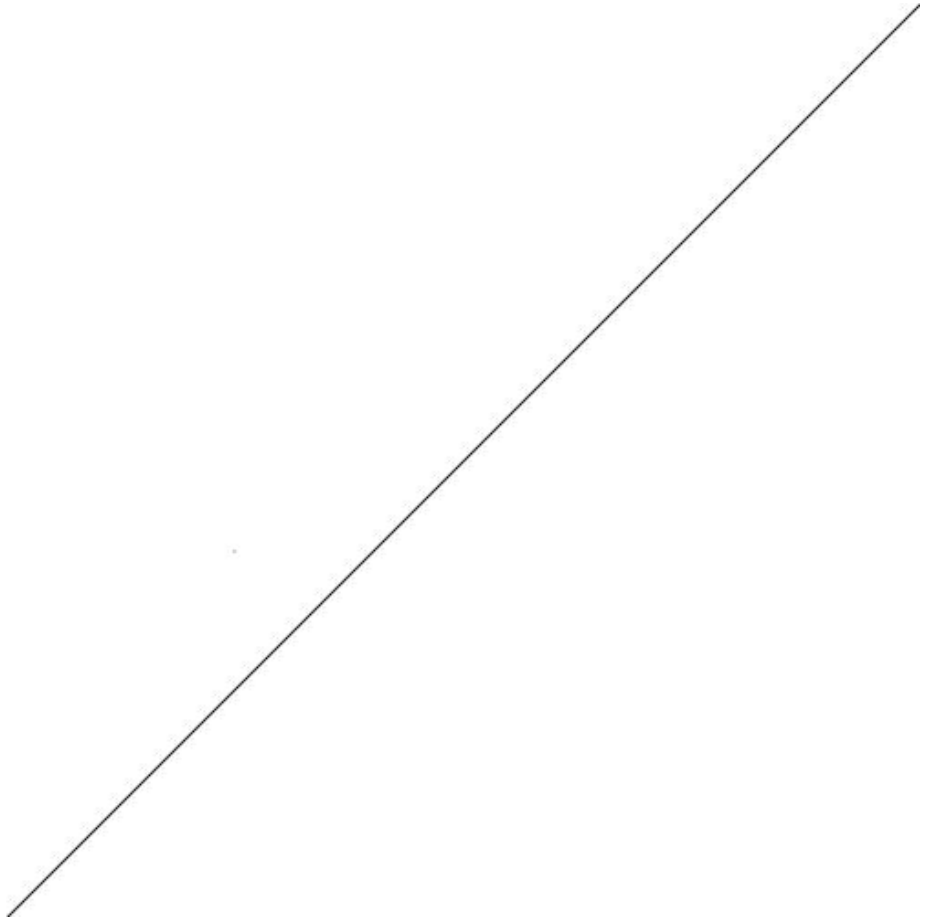
- **By default, the coordinates of a canvas range from 0 to 1 in both the x dimension and the y dimension.**
  - the coordinate (0, 0) refers to the bottom left position of the canvas.
  - Coordinate (1, 1) is found at the top right of the canvas.

PennDraw: the Pen

- PennDraw works in a model where the programmer (you!) gives a series of instructions, one by one, to a computer
- Some instructions are responsible for changing how shapes will be drawn
  - “changing the settings of the pen”
- Settings include radius and color
- The instructions change the pen settings until the next time the settings are explicitly modified

PennDraw: Pen Radius

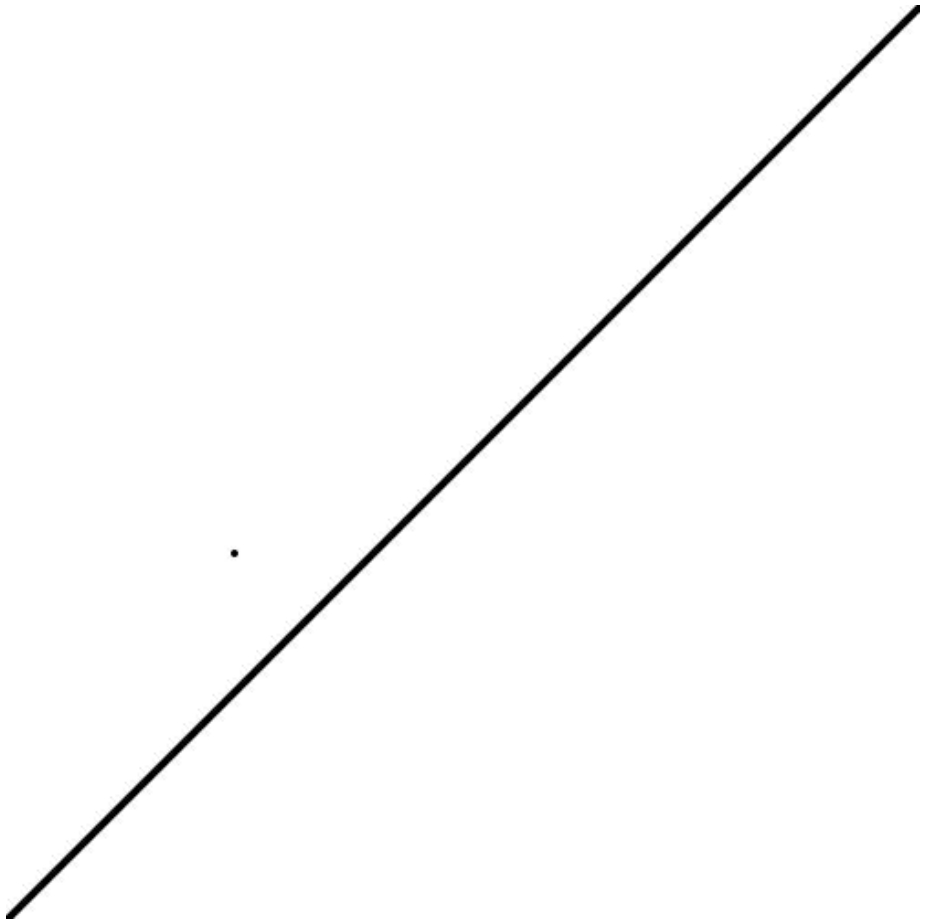
- Whenever we ask PennDraw to draw e.g. a point or line on the screen, these marks will appear with a certain thickness determined by the current setting for the radius of the pen.
- Pictured: a point and a line drawn with default radius setting of 0.002



PennDraw: Pen Radius

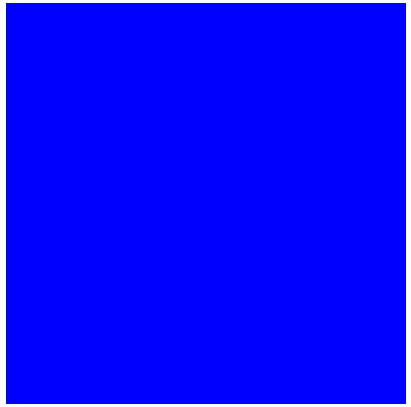
- On right is the same drawing with the pen radius set to 0.008, four times the default setting
  - Now the point is visible
- To change the pen radius:

```
PennDraw.setPenRadius(0.008);
```

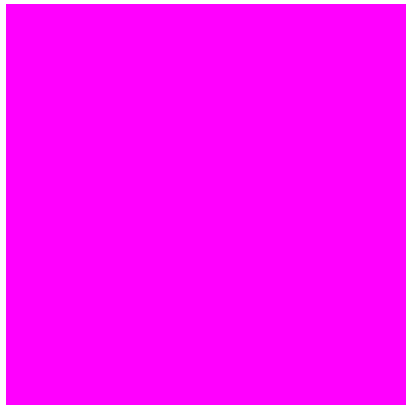


PennDraw: Pen Color

- Two ways to set the pen color:
  - **Referring to some of them by name**
  - Specifying the red, green, and blue values of the color from 0-255 each



PennDraw.setPenColor(PennDraw.BLUE);



PennDraw.setPenColor(PennDraw.MAGENTA);

PennDraw: Pen Color

- Two ways to set the pen color:
  - Referring to some of them by name
  - **Specifying the red, green, and blue values of the color using integers between 0-255**

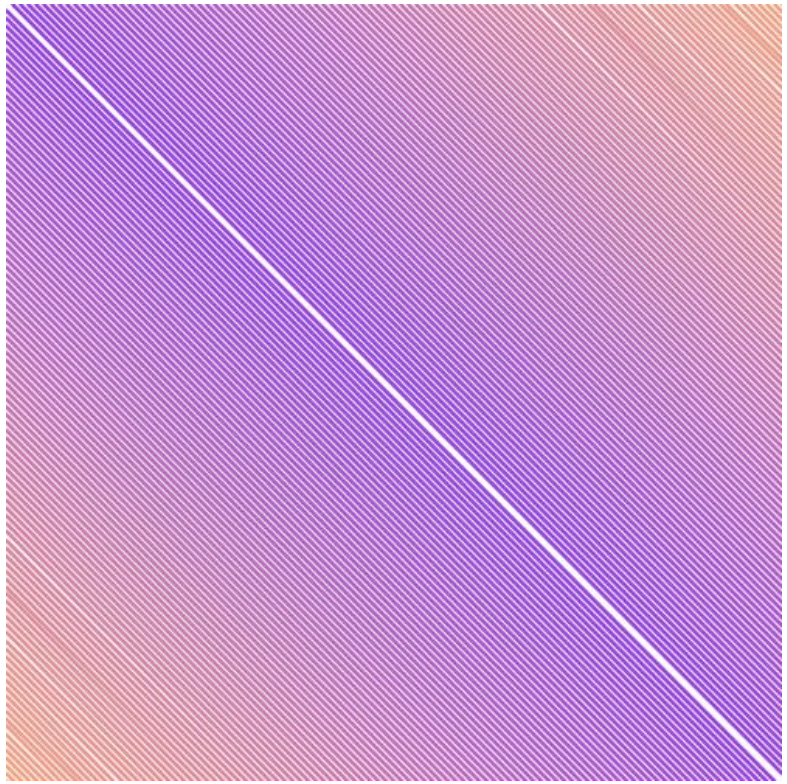
"pure red"

```
PennDraw.setPenColor(255, 0, 0);
```

"twilight lavender"

```
PennDraw.setPenColor(138, 73, 107);
```

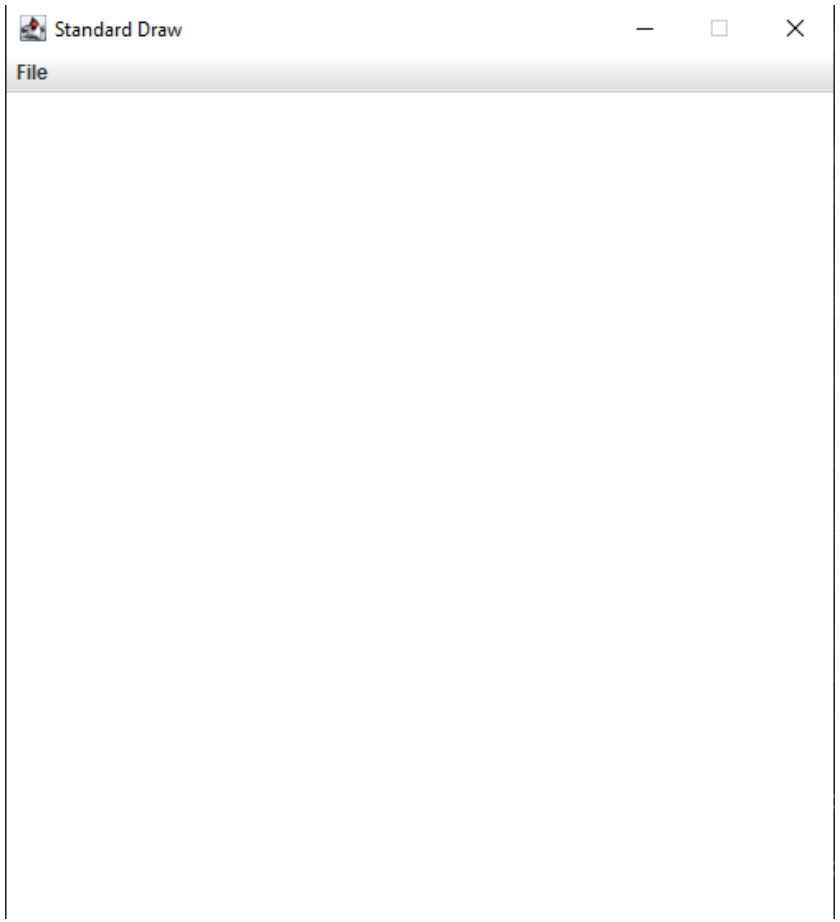
- Setting color by RGB allows for fine grained control in drawing:



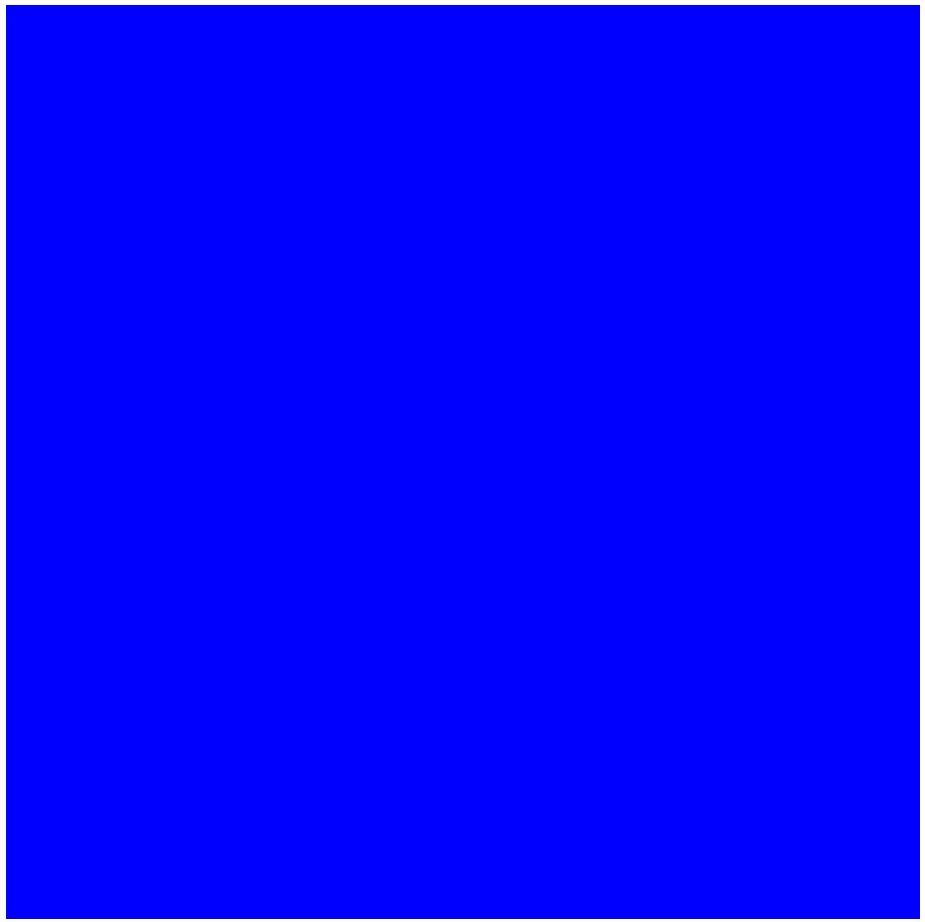
MyHouse.java

Done step by step

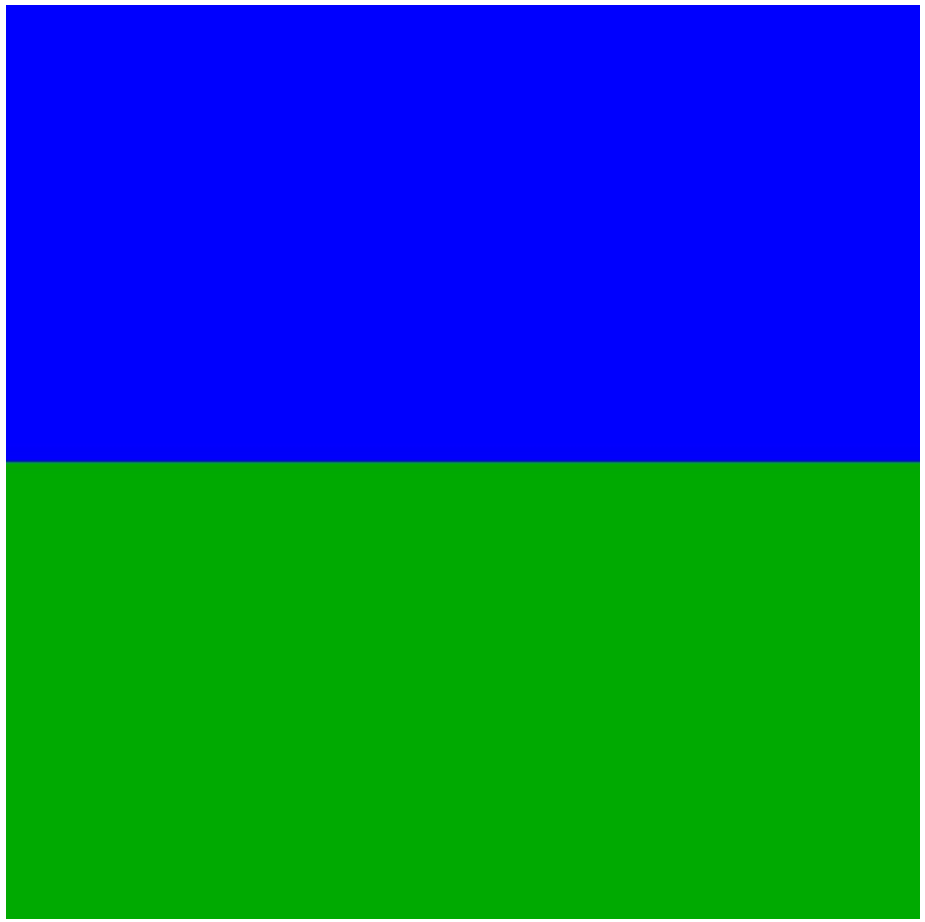
```java
public class MyHouse {
    public static void main(String[] args) {
        PennDraw.setCanvasSize(500, 500);
    }
}
```



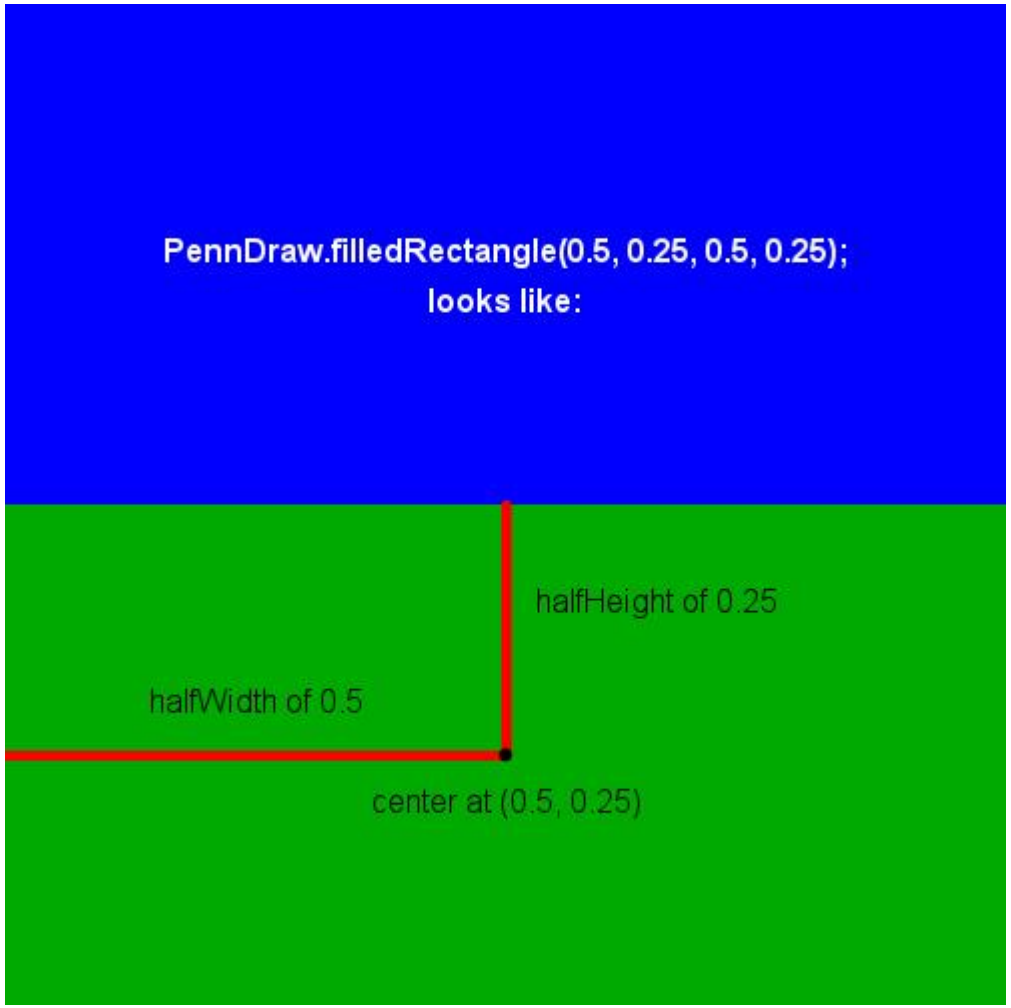
```java
public class MyHouse {
    public static void main(String[] args) {
        PennDraw.setCanvasSize(500, 500);

        PennDraw.clear(PennDraw.BLUE);
    }
}
```



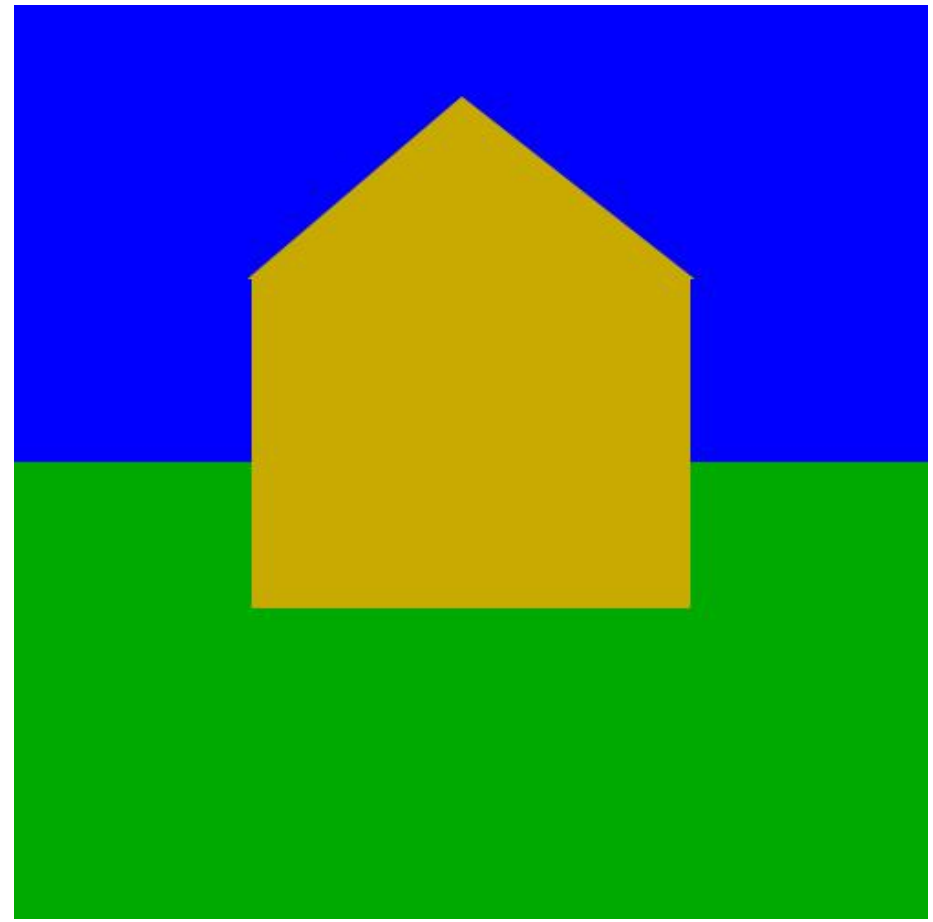
```java
public class MyHouse {
    public static void main(String[] args) {
        PennDraw.setCanvasSize(500, 500);

        PennDraw.clear(PennDraw.BLUE);

        PennDraw.setPenColor(0, 170, 0);
        PennDraw.filledRectangle(0.5, 0.25, 0.5, 0.25);
    }
}
```



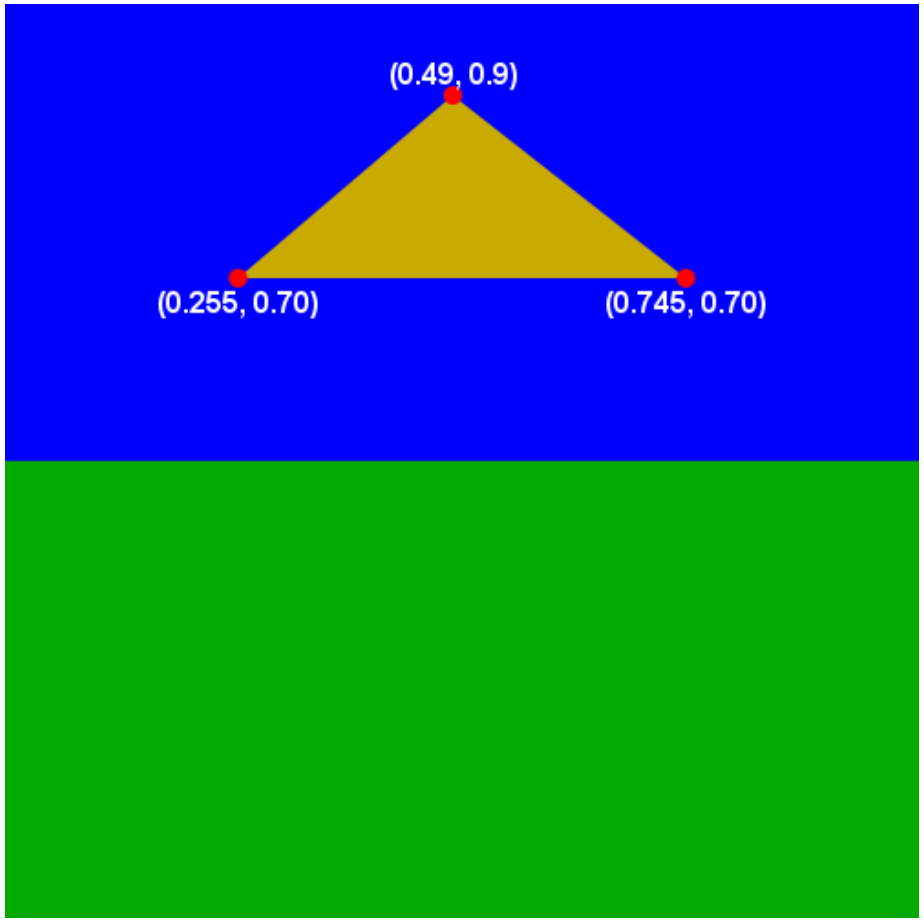
How?


PennDraw.filledRectangle(0.5, 0.25, 0.5, 0.25);
looks like:
halfHeight of 0.25
halfWidth of 0.5
center at (0.5, 0.25)

```java
public class MyHouse {
    public static void main(String[] args) {
        PennDraw.setCanvasSize(500, 500);

        PennDraw.clear(PennDraw.BLUE);

        PennDraw.setPenColor(0, 170, 0);
        PennDraw.filledRectangle(0.5, 0.25, 0.5, 0.25);

        PennDraw.setPenColor(200, 170, 0);

        PennDraw.filledPolygon(0.255, 0.70, 0.745, 0.70, 0.49, 0.90);

        PennDraw.filledRectangle(0.5, 0.52, 0.24, 0.18);
    }
}
```



Building that roof, explained:



```java
public class MyHouse {
    public static void main(String[] args) {
        PennDraw.setCanvasSize(500, 500);

        PennDraw.clear(PennDraw.BLUE);

        PennDraw.setPenColor(0, 170, 0);
        PennDraw.filledRectangle(0.5, 0.25, 0.5, 0.25);

        PennDraw.setPenColor(200, 170, 0);
        PennDraw.filledPolygon(0.255, 0.70, 0.745, 0.70, 0.49, 0.90);
        PennDraw.filledRectangle(0.5, 0.52, 0.24, 0.18);

        PennDraw.setPenRadius(0.005);
        PennDraw.setPenColor(PennDraw.BLACK);
        PennDraw.polygon(0.255, 0.70, 0.745, 0.70, 0.49, 0.90);
        PennDraw.rectangle(0.5, 0.52, 0.24, 0.18);
    }
}
```