19-KelvinZimmerman

Item	Description	Points Possible	Points	Percent
1	My House: Three Rectangles	1	1	100%
2	My House: Two Lines	1	1	100%
3	My House: One Circle	1	1	100%
4	My House: One Text Label	1	1	100%
5	My House: Three Colors	1	1	100%
6	My House: Graphic Quality - Looks like a house	2	2	100%
7	My House: Interactive Feature - Click on graphic changes something	1	1	100%
8	Graphic Bar: Width of window is 400px	0.5	0.5	100%
9	Graphic Bar: Bar Count - Number of Bars is divisible by 6	1	1	100%
10	Graphic Bar: No Gaps between bars	2	2	100%
11	Graphic Bar: All bar widths within 1 of each other	1	1	100%
12	Graphic Bar: Color Progression	1	0.75	75%
13	Graphic Bar: No Bar Outlines	0.5	0	0%
14	Graphic Bar: Reduces Code Redundancy	1	0	0%
15	Answers Questions in Canvas Submission	1	1	100%
	Total	**16**	**14.25**	**89.06%**

My House

Item 1 – Three Rectangles ..... Points: 1/1

Item 2 – Two Lines ..... Points: 1/1

Item 3 – One Circle ..... Points: 1/1

Item 4 – One Text Label ..... Points: 1/1

Item 5 – Three Colors ..... Points: 1/1

Item 6 – Graphic Quality – Looks like a house ..... Points: 2/2

Item 7 – Interactive Feature – Click on graphic changes something .... Points: 1/1

Wow. I really like your idea for this. You didn't need to build the house drawing based on where the user clicked in the drawing, but that is a great idea!

Graphic Bar

Item 8 – Width of window is 400px ..... Points: 0.5/0.5

Item 9 – Bar Count – Number of bars is divisible by 6 ..... Points: 1/1

Item 10 – No gaps between bars ..... Points: 2/2

Item 11 – All bar widths are within 1 pixel of each other ..... Points: 1/1

Item 12 – Color Progression – horizontal and uniform ..... Points: 0.75/1

Your color progression never gets to the full green intensity. Your 6 green intensity values are 0, 42.5, 51.0, 63.75, 85.0, and 127.5. So you do progress uniformly between 0 and 127.5, but you need to progress from 0 to 255.

Item 13 – No Bar Outlines ..... Points: 0/0.5

Notice how your color bars have black outlines around them. If you call setWidth(0) on each bar before drawing it, you would eliminate these outlines.

Item 14 – Reduces Code Redundancy ..... Points: 0/1

Your code is written such that you have pre-determined exactly how many color bars you will use (6), pre-determined exactly how wide each bar will be (67), pre-determined what the 6 green intensity values will be (0, 255/6, 255/5, 255/4, 255/3, and 255/2).

Think about what you would have to change (and how many places in your code you would have to change) if you were asked to now use 12 color bars instead of 6. All your pre-computed values would have to be re-computed (outside of your program) and then you would have to edit the code to correct all those values and add 6 more rectangle variables. Suppose you were asked to use 48 bars or even 96 bars. What would happen to your code then?

Can you think of a way to compute the locations for the corners of each of your bars using the overall width of the window (400) and the number of bars you plan to draw? How about also computing the green intensity values based on the overall range available (256 values from 0 to 255) and the number of bars you plan to draw? If you set a variable named NUM_BARS to a value like:

```
NUM_BARS = 6
```

Then perhaps you could use a loop to go through and draw the bars like:

```
for bar_number in range(NUM_BARS):
    # figure out where this rectangle should go
    # figure out the green intensity level for this rectangle
    # set up the rectangle and draw it
```

Take a look at the document that I post in an announcement on Monday.

Item 15 – Answers Canvas Questions ..... Points: 1/1

General Comments