Art With Tech

Abstract:

The goal of this project is to explore art and math with computer science, and to see if people learning the concepts find it helpful. A lot of times in math and art, it is easier for people when they can visualize patterns, which is possible with programming languages such as Processing. I made programs with Processing(a Java based language) to model art and math. The topics included graphing sin cos tan patterns, fractals, stippling art(art made with different sized dots), and the main concept was the unit circle because it is a concrete math topic the majority of students learn. I made presentations to math and comp-sci classes at my school, and took the results from 2 surveys. The surveys looked at how many people knew the topics before, if they thought learning with code and visualizations was effective, and it asked specific questions over topics in the presentation. Looking at the survey data, I infer that people found graphics generated with comp-sci effective.

Table of Contents:

Statement of the Problem:

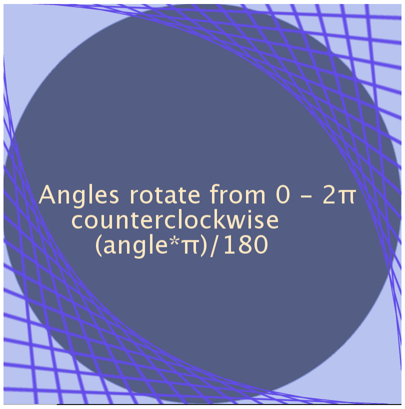
Would people find visuals created with code effective at teaching or solidifying a subject in art or math? There are a lot of ways to learn, and sometimes students don't pick up topics easily when taught the traditional way. With comp-sci, a lot of graphics and patterns can be generated using specific mathematical objects or equations. So the goal of this project is to find more effective ways to teach math and art and narrow the gap between art and math using comp-sci.

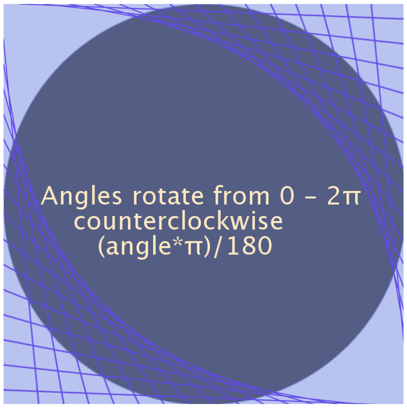
Description of the method:

I made programs about the unit circle, fractals, other visuals, and stippling art. I used code from a tutorial(reference and citation later) to make a sin cos tan program, modified it, and showed how when sin cos tan are interchanged, the patterns changed considerably.

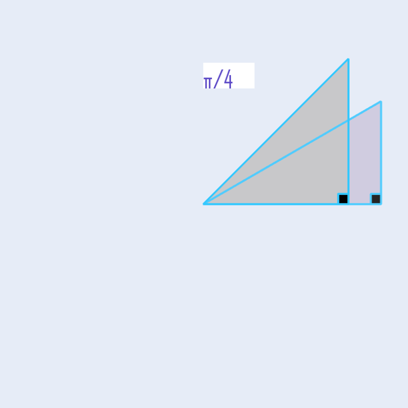
The unit circle program shows different ways angles of a unit circle can be visualized, how the unit circle is very symmetrical, and how it can be seen as a butterfly! When space is pressed, a counter increases, and tells the draw function to run a different value/set of instructions. It was hard to make a unit circle program from just lines and circles, and to animate and set ways for the circles and triangles to be drawn. Here are the transitions in the unit circle program:

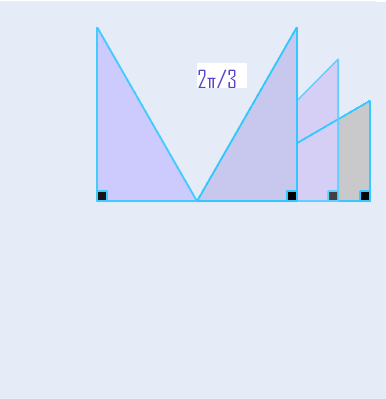
1. Shows a visual about the unit circle, and the purple graphic changes as the mouse moves left to right.

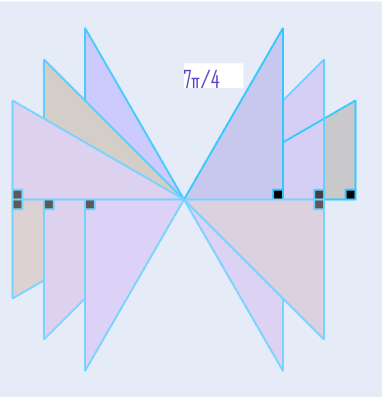




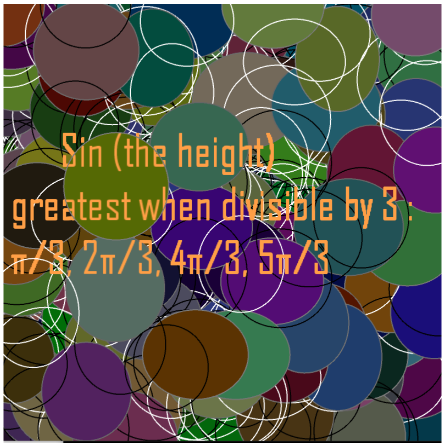
2. Shows triangles being drawn in unit circle markings and shows the angles to which they are compared to 2pi. It repeats drawing itself and starts from one triangle until space is pressed.

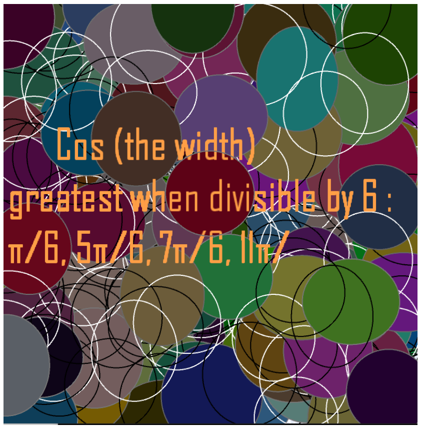


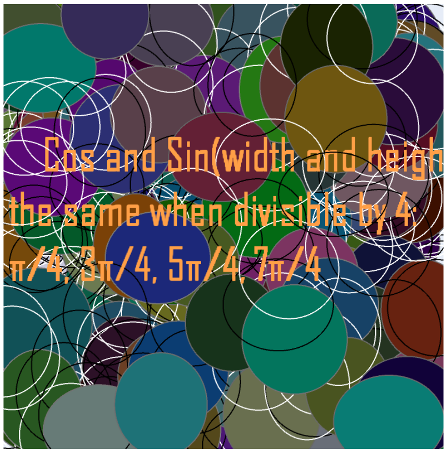




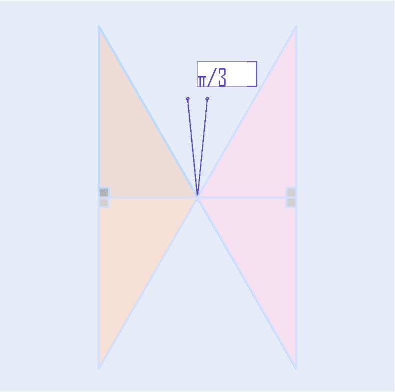
3. The background fills up with circles in random positions, and the graphics tell how to know if the height is the greatest, the width is the greatest, or they are equal based on a unit circle with markings.

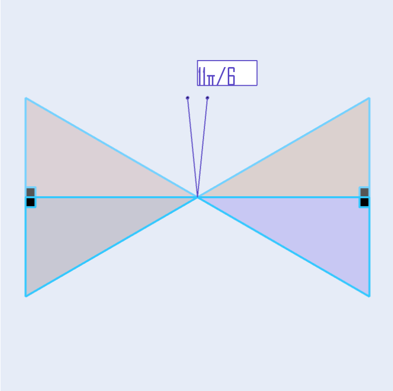


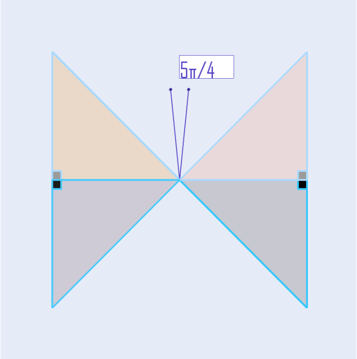




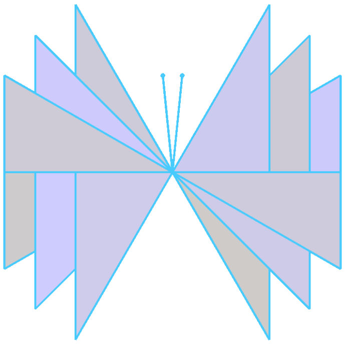
4. After each of the screens with the graphics of the circles filling up in the background, a “butterfly” is drawn with the angles showing how when, for example, sin is “something **π**/3” sin is the greatest, and the butterfly is the tallest. Same goes for when “something **π/6”** **c**os is the greatest, and the butterfly is the widest. When the mouse moves, the colors of the triangles being drawn change from brighter to darker shades.

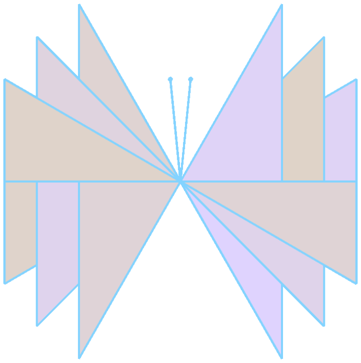


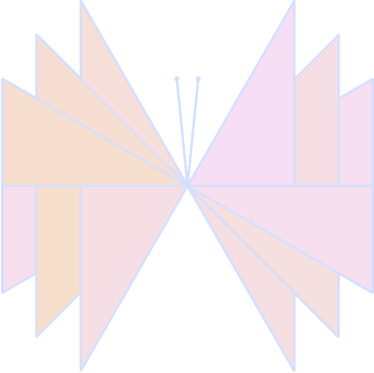




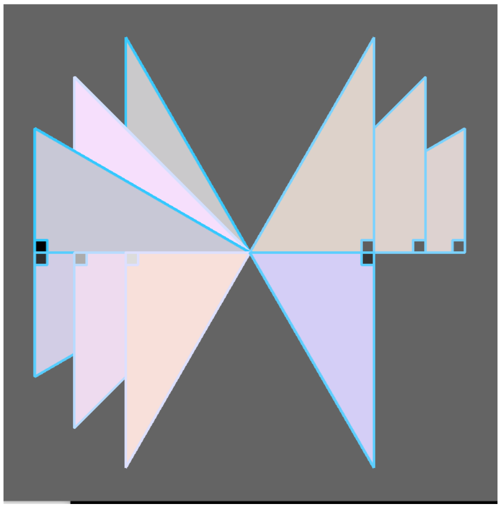
5.   A “butterfly” with all the unit circle markings.
(the colors change based on the position of the mouse).

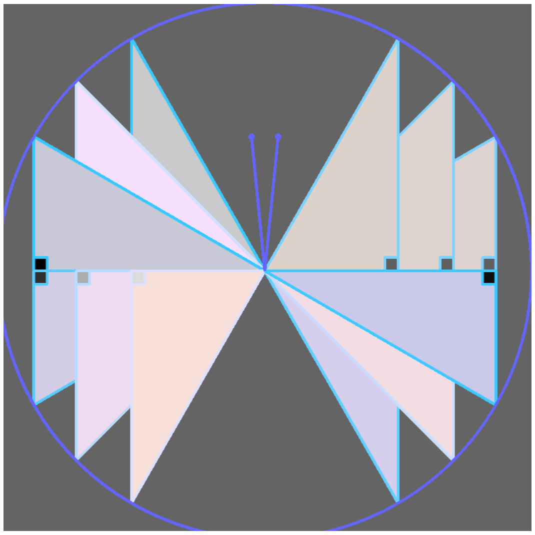




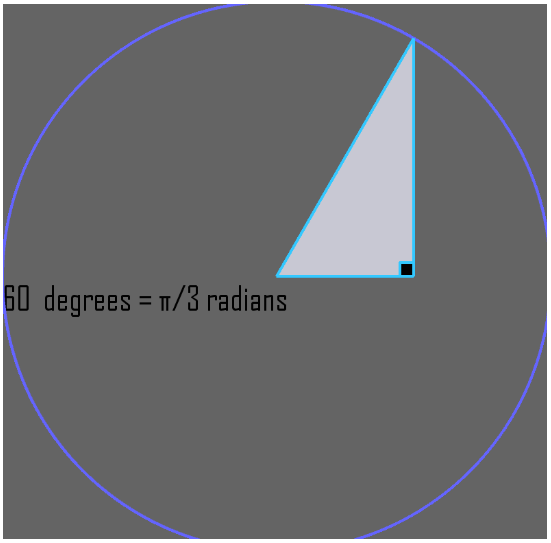


6. When space is pressed, triangles matching up with unit circle markings are drawn.

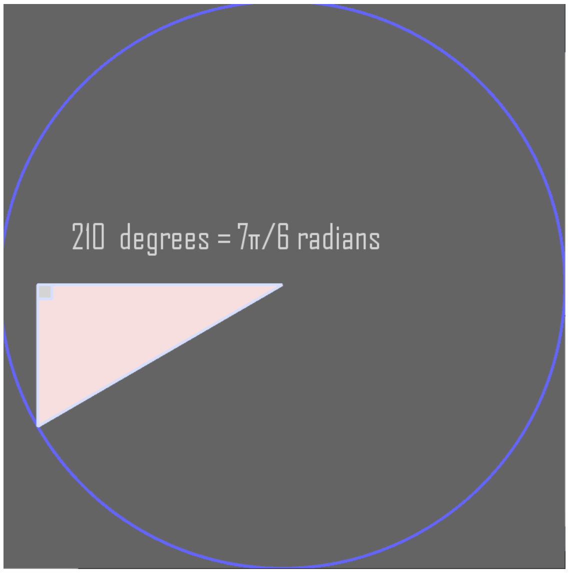




7. After that, when space is pressed angle conversion to **π** and reverse is shown.



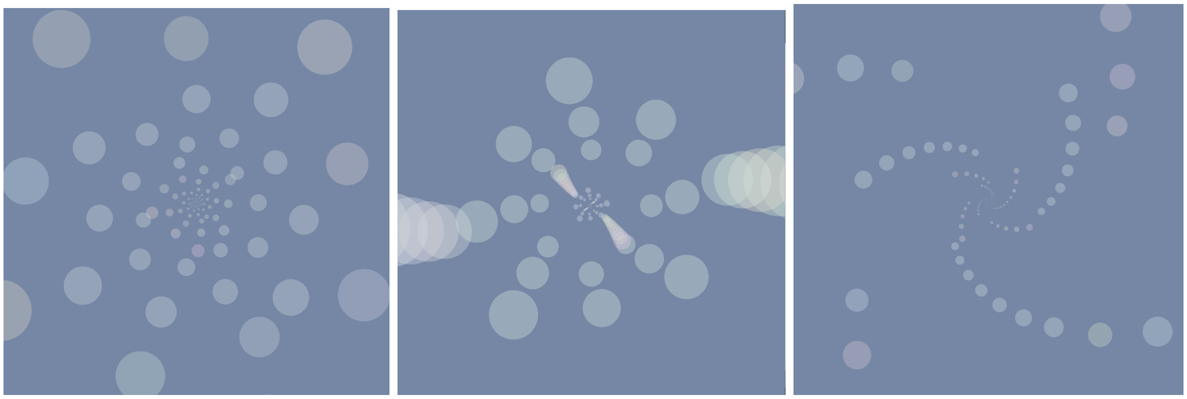
60 degrees = π/3 radians



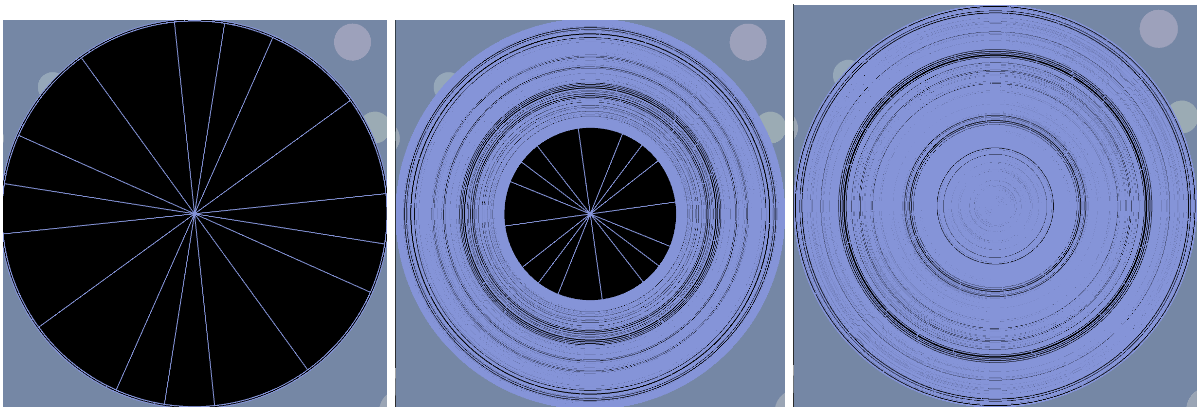
210 degrees = 7π/6 radians

Other Programs:

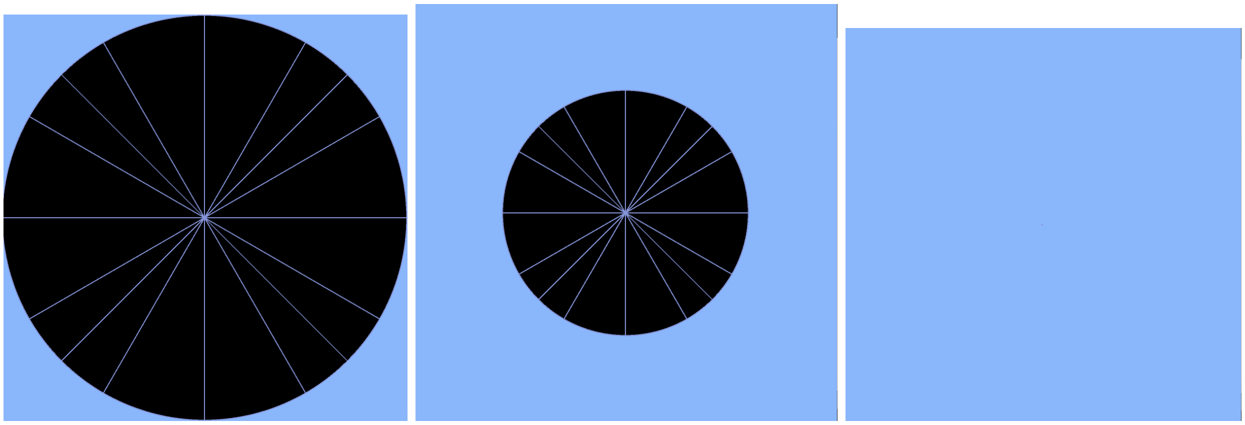
1. A graphic showing how the size of spinning circles originating at the center changes when the mouse moves .



2. In Processing, the "draw" function is called 60 times a second, and if the background is not erased, it can be seen that the program runs a vast amount in a short amount of time.

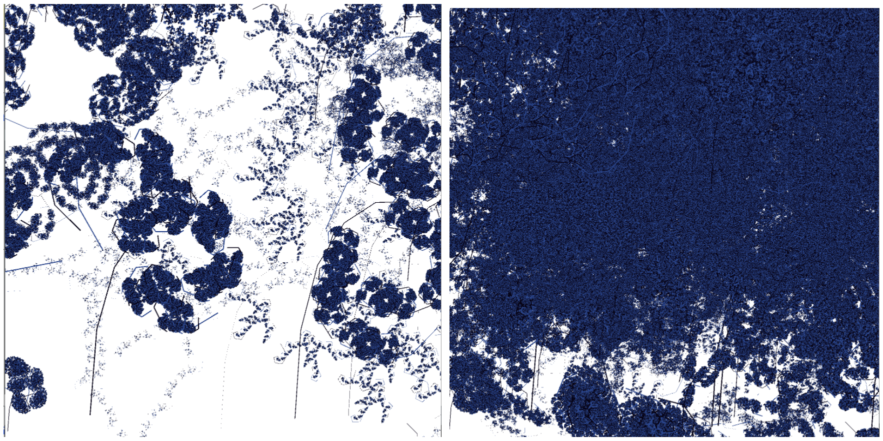


3. When the background is erased every time draw is called:



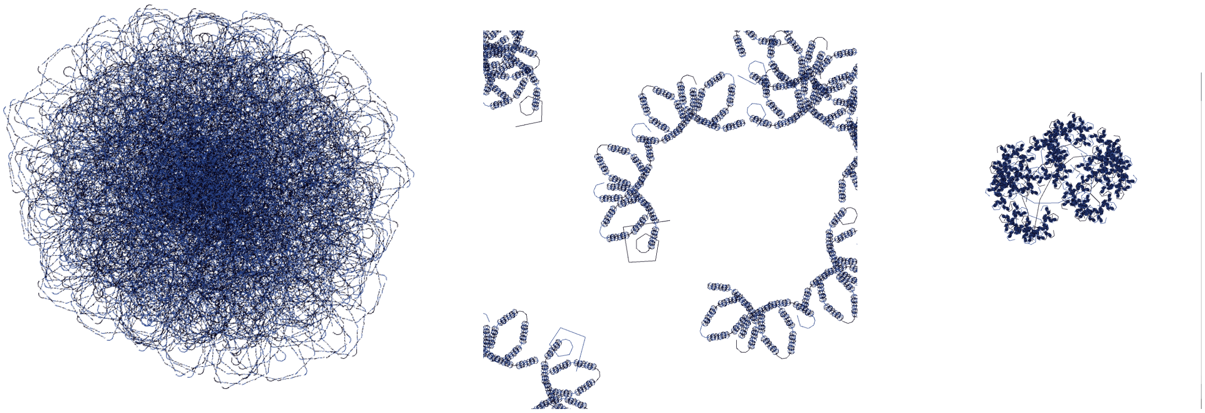
Fractals:

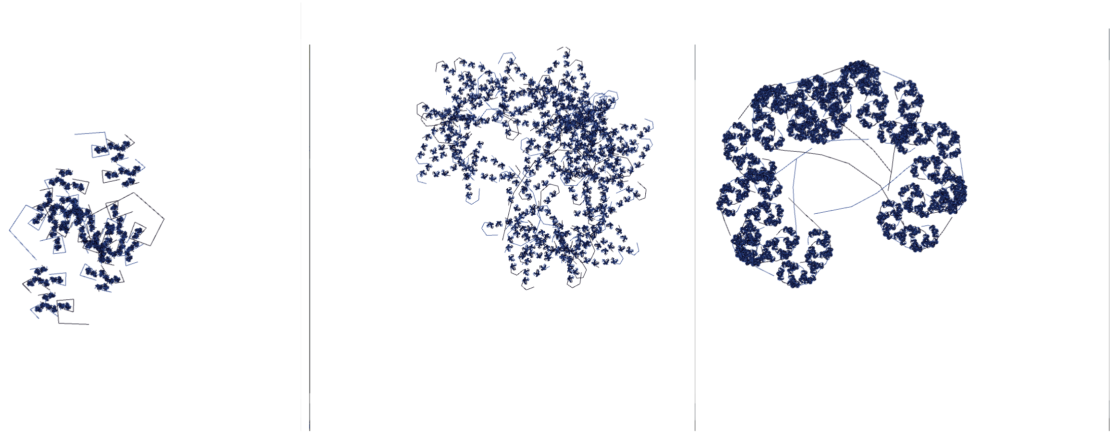
1. They use the same amount of recursion, and start with the same length, but the angle changes randomly, to fill up the screen.



2. When the mouse is moved, the angle of the fractals changes:

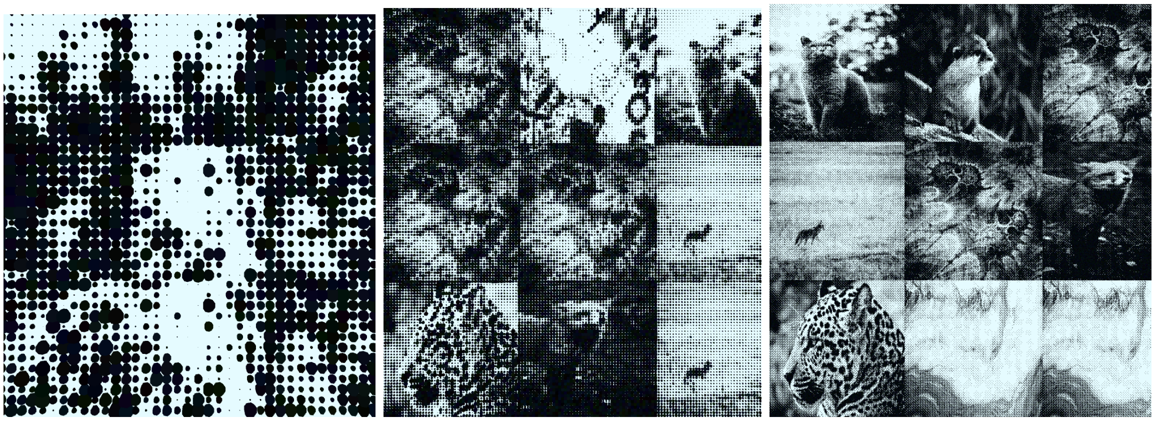
The original length of the segment is the same each time, but different patterns are created based on the angle the segment is redrawn.

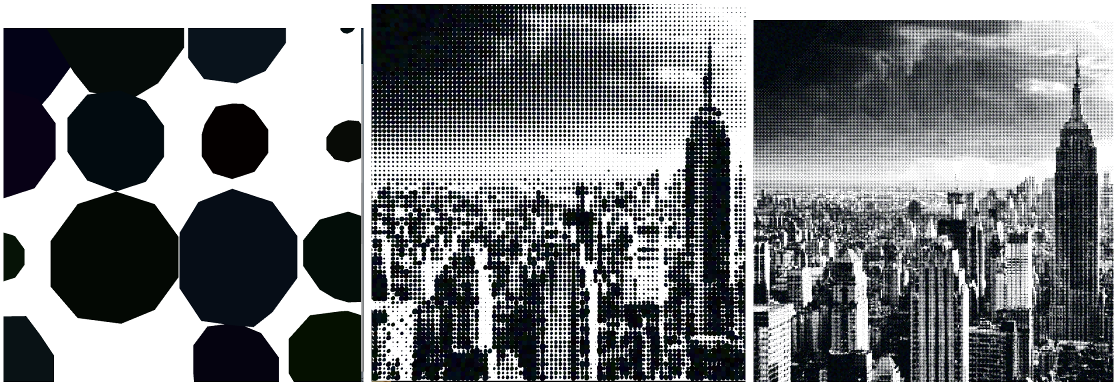


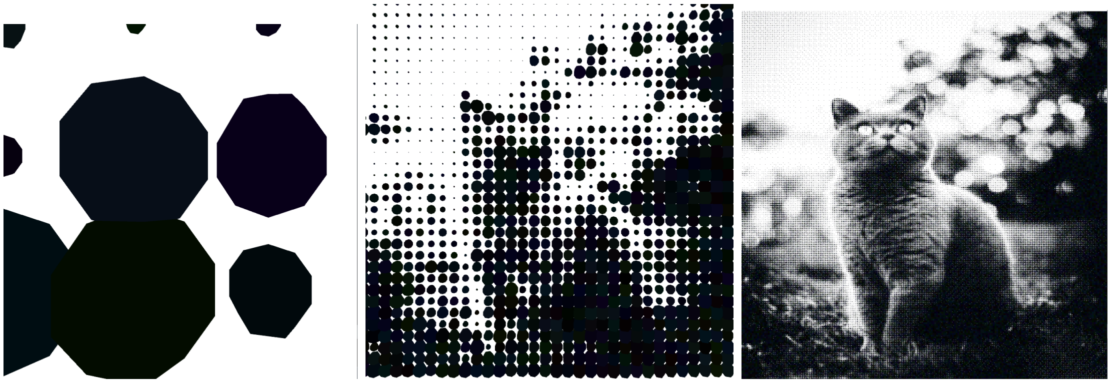


For the art concepts(the goal of the project was to model both art and math concepts and see if people understood or could learn from them better) I did stippling art, properties of colors, and positive and negative space. Stippling art is art with dots, and the size of the dots is decided by how light or dark an area is. If an image is divided into a very few divisions, then the image won't be clear. If it is divided into a lot, up close the image looks like a pile of dots, but far away it appears and looks realistic.

1. When the mouse moves from left to right, the subdivisions of the brightness being taken increases or decreases to make clearer or less detailed images.

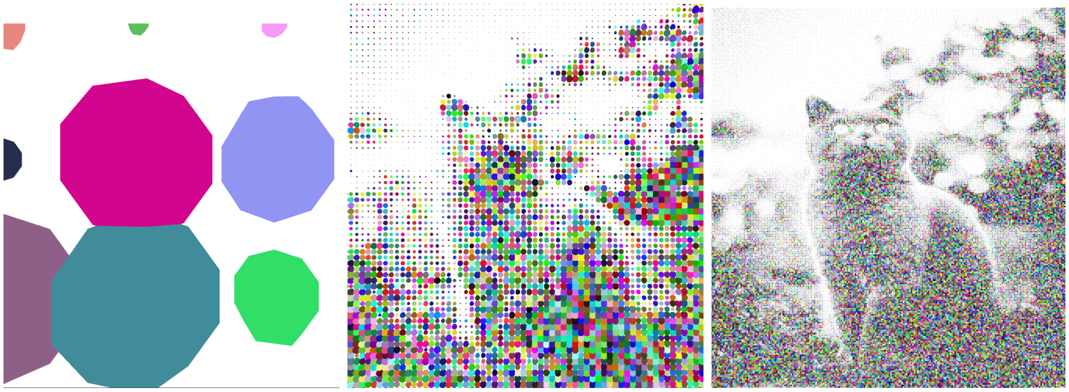






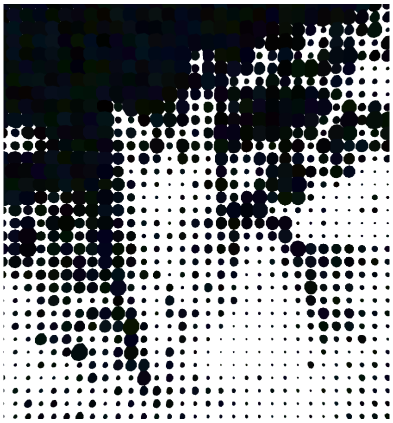
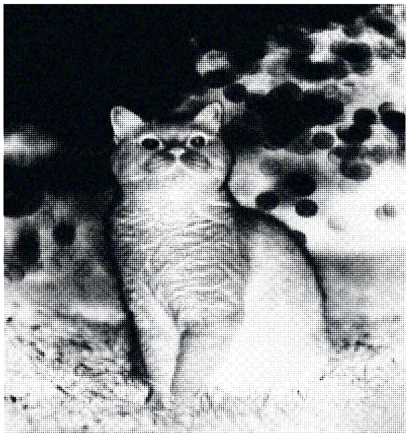
It doesn’t matter what the color of the dots is, necessarily, because they are taking up space to show brightness. For example in a black and white movie, if there is a vase, the vase might have been split up into thousands of random colors, but the camera would only see the vase. This is the same in reverse, the random colors still show the outline.

2. Stippling art with random colors:



Positive and negative space is also the outline of the image. If the light is changed to dark, and dark to light, then the outline of the image would still appear.

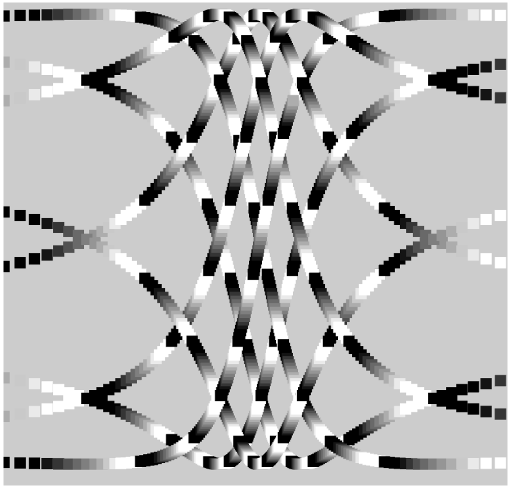
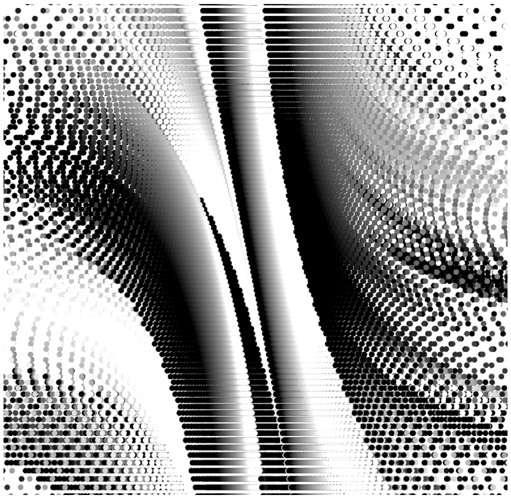
3. Stippling art with the dots being white instead of dark.

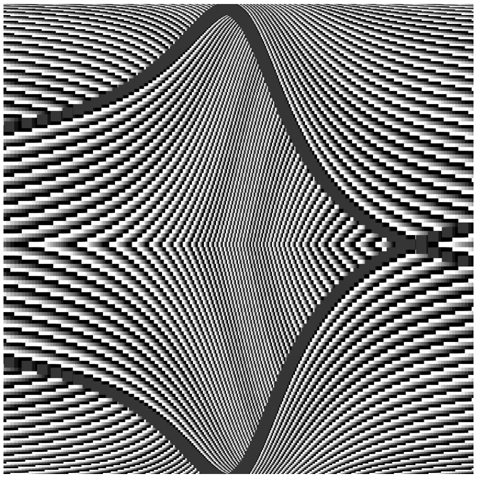
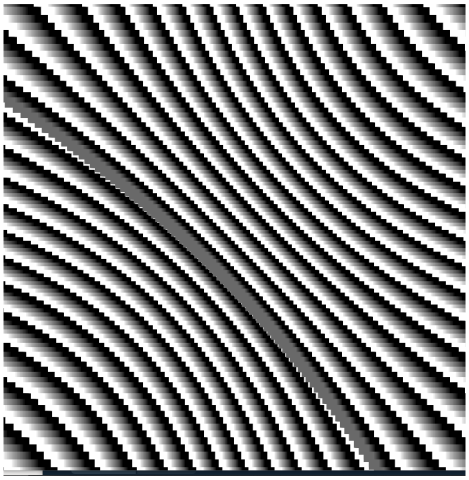



I made changes to a sin cos tan tutorial in Processing to show how tan, cos, sin, when interlaced with each other to create patterns, and those patterns are very different from one another. Just changing sin to cos, or vice versa, creates a big difference!
Credits go to:

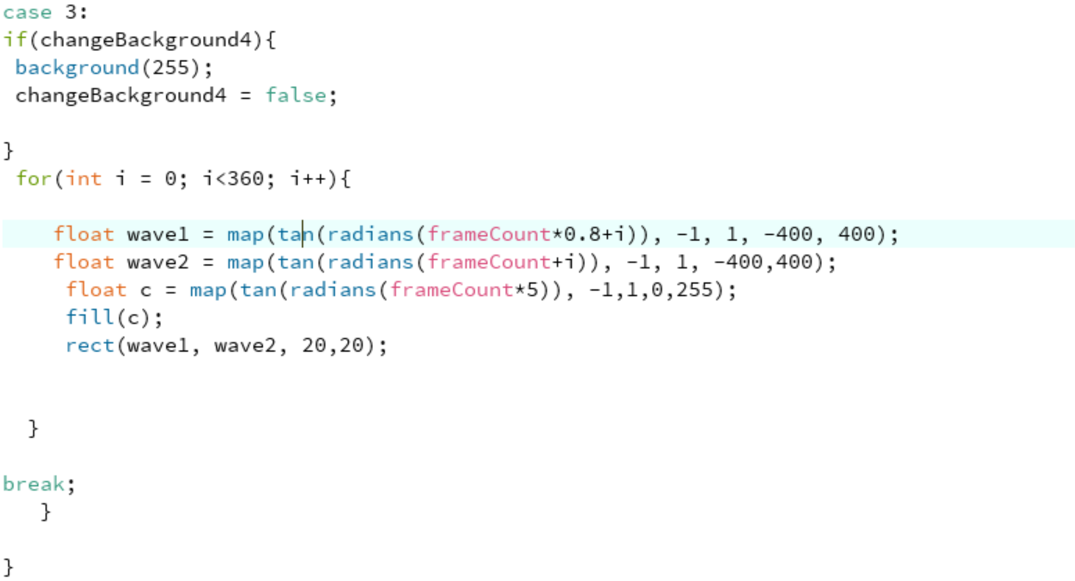
Rodenbroker, T. R. [tim rodenbröker creative coding]. (2021, June 17). *Exploring Wave-Figures in Processing* [Video]. YouTube. https://www.youtube.com/watch?v=Th7m7QEeUbI&t=947s

1. Almost the same code showing patterns with sin cos tan, but sin cos tan is interplaced with each other.

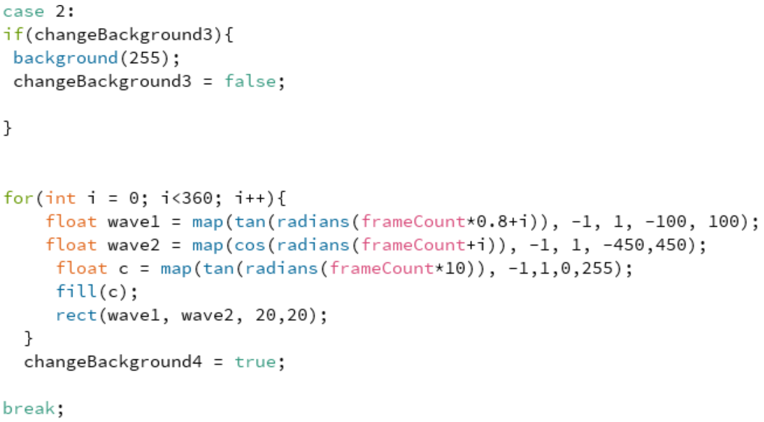



2. Code showing how tan changed to cos creates a huge difference when looking at the patterns they produce.



```
case 3:
if(changeBackground4){
 background(255);
 changeBackground4 = false;

}
 for(int i = 0; i<360; i++){

    float wave1 = map(tan(radians(frameCount*0.8+i)), -1, 1, -400, 400);
    float wave2 = map(tan(radians(frameCount+i)), -1, 1, -400,400);
     float c = map(tan(radians(frameCount*5)), -1,1,0,255);
     fill(c);
     rect(wave1, wave2, 20,20);


  }

break;
   }

}
```

VS:



```
case 2:
if(changeBackground3){
 background(255);
 changeBackground3 = false;

}


for(int i = 0; i<360; i++){
    float wave1 = map(tan(radians(frameCount*0.8+i)), -1, 1, -100, 100);
    float wave2 = map(cos(radians(frameCount+i)), -1, 1, -450,450);
     float c = map(tan(radians(frameCount*10)), -1,1,0,255);
     fill(c);
     rect(wave1, wave2, 20,20);
  }
  changeBackground4 = true;

break;
```

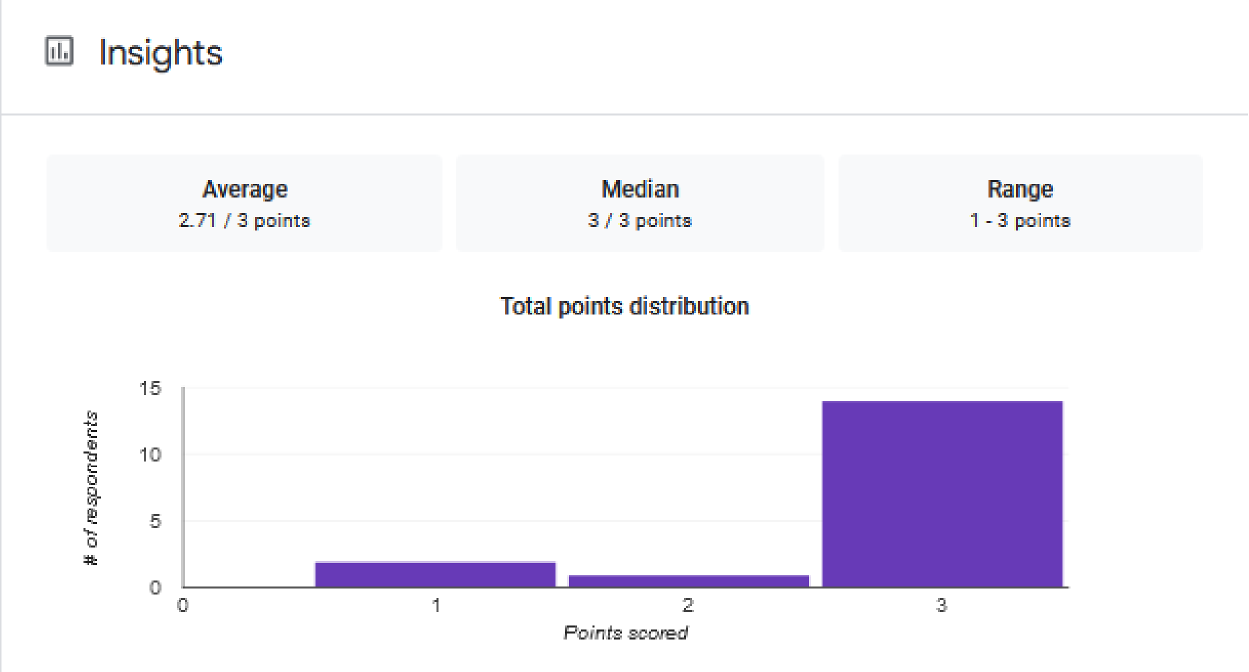
When the programs were done, I did a presentation to 2 computer science classes, and 3 math classes at my school with them. I presented the tan cos sin programs, unit circle programs, stippling art programs, and other programs to the students. Then I took survey data from students in 2 classes to see if the presentation of the programs I did was “effective” and if they learned anything. (The survey was anonymous).

Results From the Study:

From the survey results, I can inference that students thought the presentation of graphics was effective.

Survey results from the math class(Honors Algebra II):

This shows graphics from the survey I gave to the Honors Algebra class.



Insights

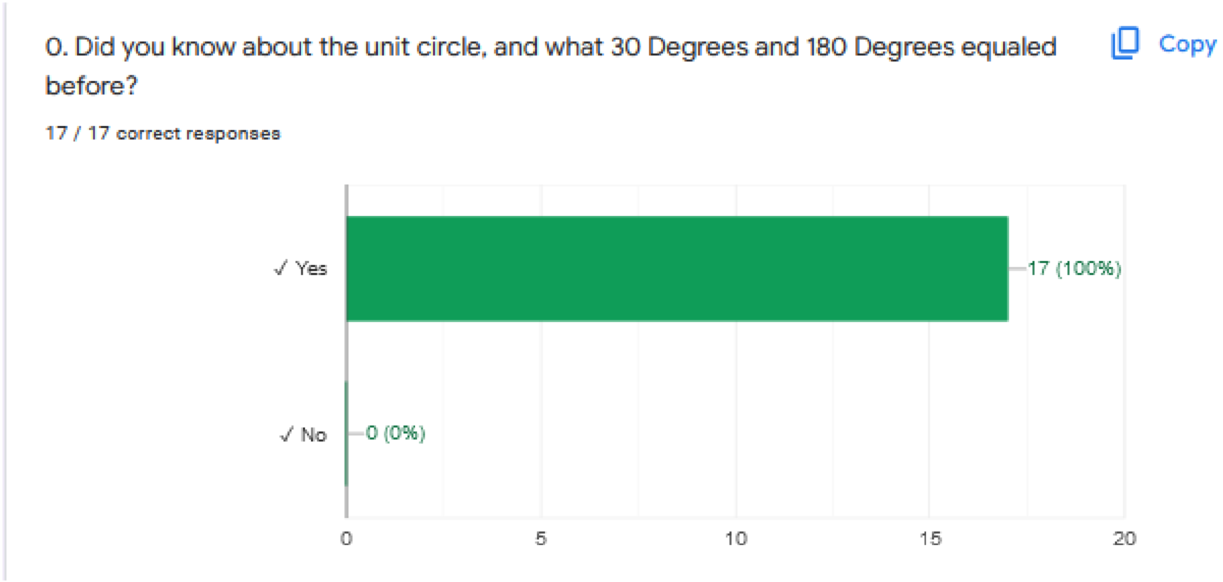
Average
2.71 / 3 points

Median
3 / 3 points

Range
1 - 3 points

Total points distribution

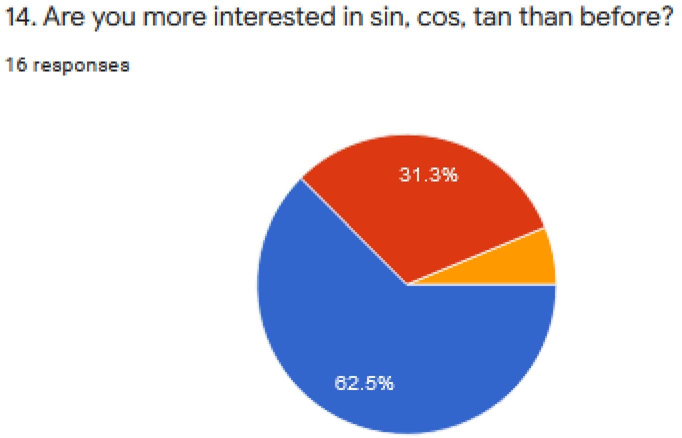
Most people in the math class got the answers to unit circle questions right. 14/17 got all questions right, 1/17 got 2 out of the 3 questions right, and 2/ 17 got one of the questions right.



0. Did you know about the unit circle, and what 30 Degrees and 180 Degrees equaled before?

17 / 17 correct responses

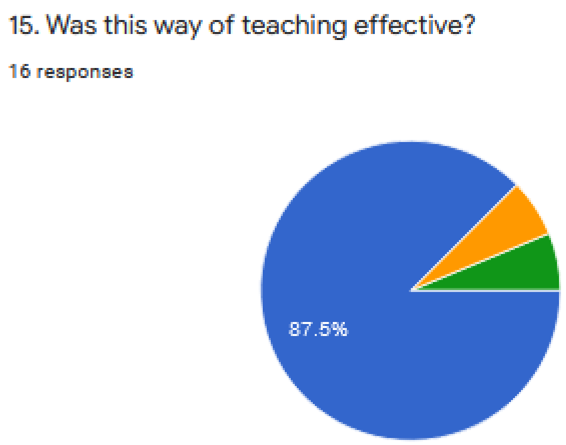
Everyone in the class said they knew what the unit circle markings equaled before. (So the data from the past graphic should be data from before the presentation).



14. Are you more interested in sin, cos, tan than before?

16 responses

Most people were more interested in sin cos tan than before. 62.5% were, 6.3% middle area, and 31.3% no.

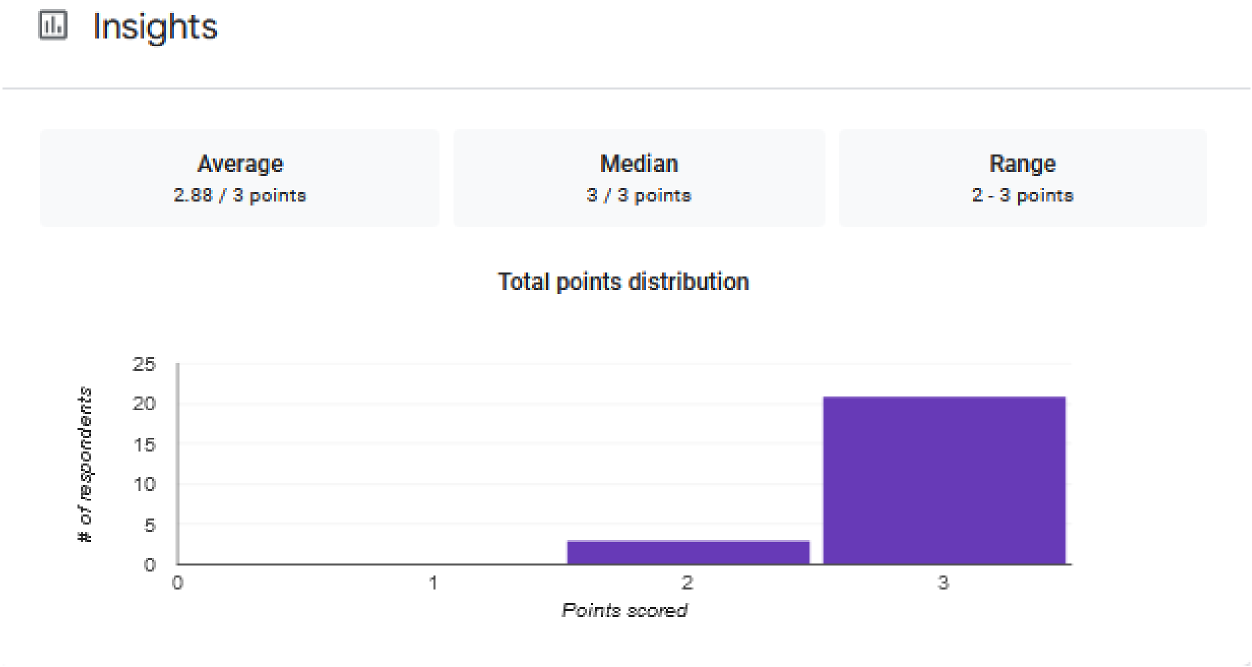

15. Was this way of teaching effective?

16 responses

87.5 % of people said it was effective, 6.3% kind of, and 6.3% other.

The surveys were anonymous, so this data shows that people thought graphics with comp-sci were effective. The math class knew about the unit circle, so data from how many people got it right should not be taken .

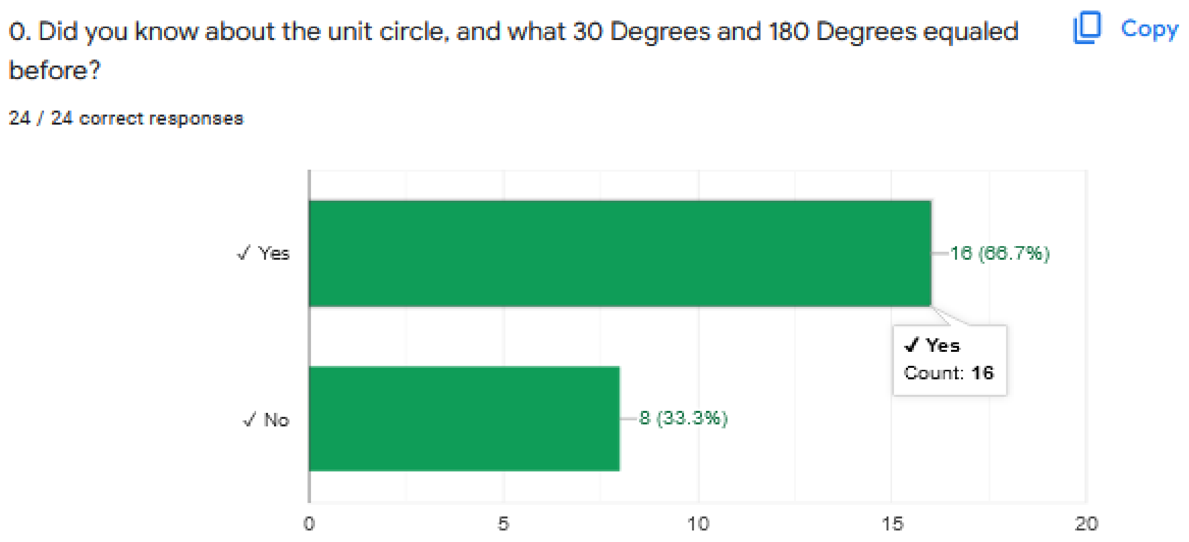
In the Computer Class:


Insights

Average	Median	Range
2.88 / 3 points	3 / 3 points	2 - 3 points

Total points distribution

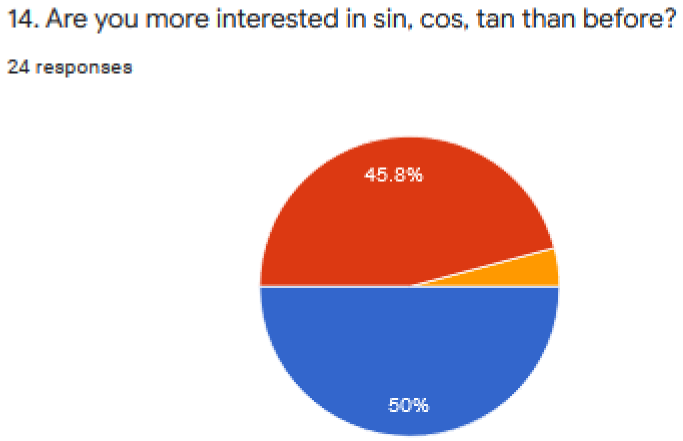
Most people got the answers to unit circle questions right in the computer science class.



0. Did you know about the unit circle, and what 30 Degrees and 180 Degrees equaled before?

24 / 24 correct responses

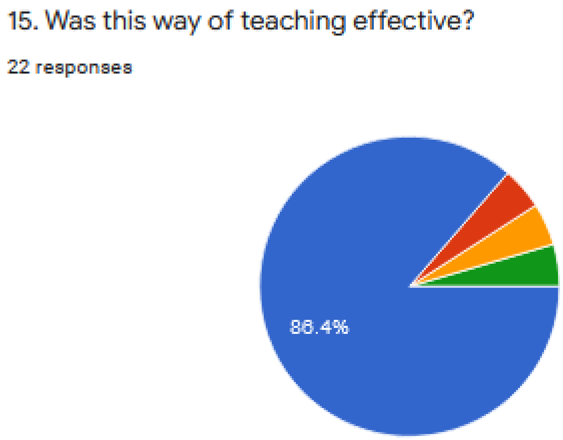
⅓ of the people did not know the answer to the questions before, meaning that they learned something from the presentation, because less than ⅓ of people got one of the questions wrong.



14. Are you more interested in sin, cos, tan than before?

24 responses

50% were more interested in the unit circle, 45.8% were not, and 4.2% were other.



15. Was this way of teaching effective?

22 responses

86.4% of people thought it was effective, 4.5% said no, and 9% was kind of or other.

Most people in the comp-sci class thought the presentation was effective, most people got the multiple choice questions about the unit circle right(even though ⅓ of the people said they didn't know the answer before the presentation), showing that at least in some way the graphics generated with Processing were effective.

References:

Rodenbroker, T. R. [tim rodenbröker creative coding]. (2021, June 17). *Exploring Wave-Figures in Processing* [Video]. YouTube. https://www.youtube.com/watch?v=Th7m7QEeUbI&t=947s

Most Significant Achievement:

I was really happy that the majority of people said teaching with computer graphics was effective. I've never really worked on such a big project before, and I was really happy that the programs worked in the end, and that I was able to generate visuals that I liked with code. :)

Future Work:

I would like to continue this project in the future. I can make visualizations of other math/art programs, and explore how graphics generated with comp-sci can be useful. I would like to improve the program, and give a survey to a larger group of people.

Acknowledgments:

Thank you to Mrs. Lozano for always helping me and giving me invaluable suggestions. I’m really happy I worked on this project, and without her I don’t think I would have.

Thank you also to Mrs. Lozano again for letting me present to her comp-sci classes to gather data.

Thank you also to Mrs. Larson for letting me come into her math classes to get data.