Experimental Web Design

SVG's Interacting in relation to mouse movement using Trigonometry.

About me

My name is Roger, and I am in my Senior Year studying Computer Science at Eastern Washington University. (Class of Winter Quarter 2024)

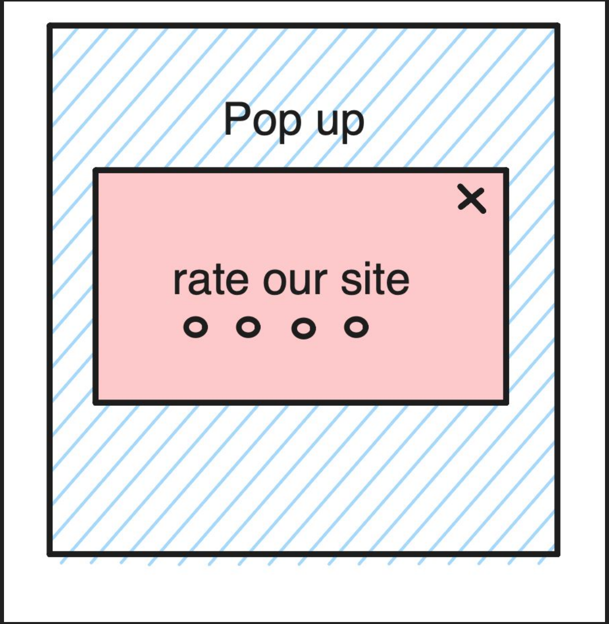
I enjoy playing the guitar and the pacific northwest.

Web design sparked my interest to study Computer Science.

I am always interested in how math fits in the world of web designl.

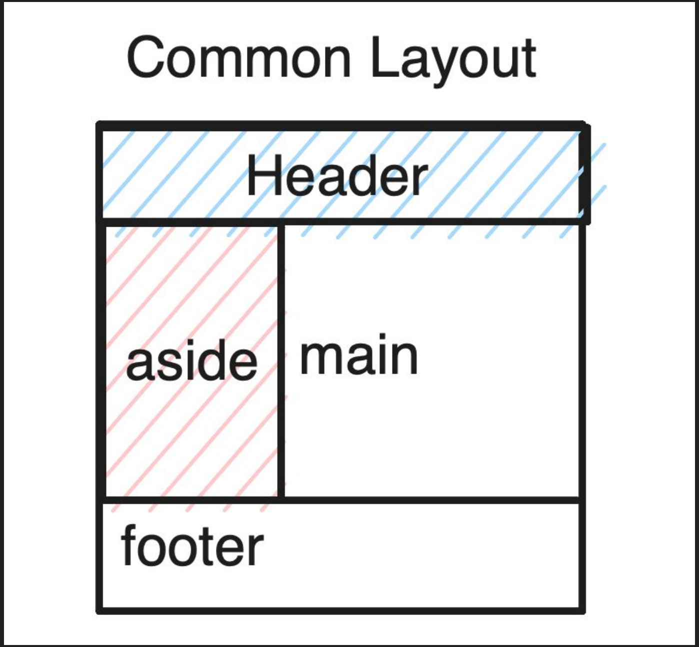
Websites employ user interaction

For example, you might get unwanted pop-ups.



Traditional Interaction

Help us read information



What is experimental web design?

Experimental web design is finding new way of interacting with websites/content in creative, untraditional, and useful ways.

Break norms of layout.

Interavity on scroll, hover, or click events.

3D Websites, using WebGL (libraries or build from scratch)

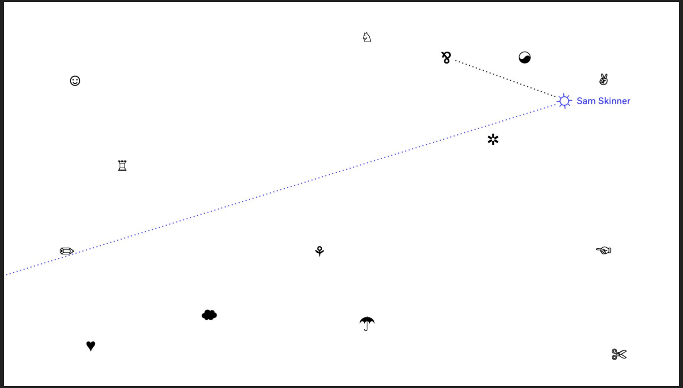
Websites may use Math :)

Always consider accessibility and how web effects might affect users.

Experimental Interaction

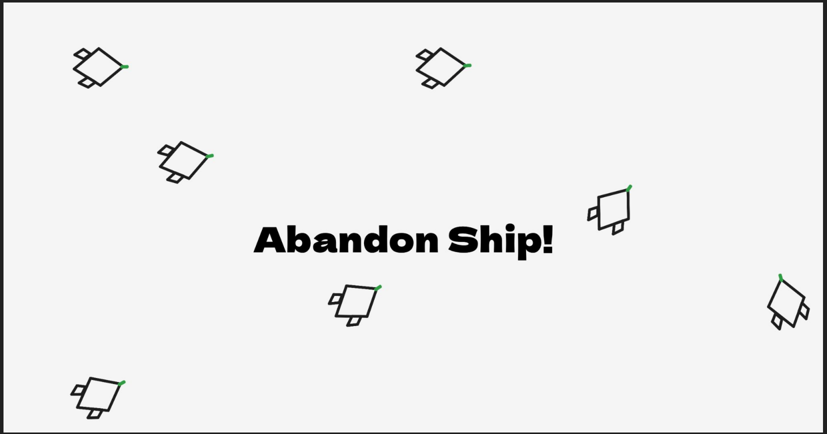
samskinner by Jake Dow-Smith

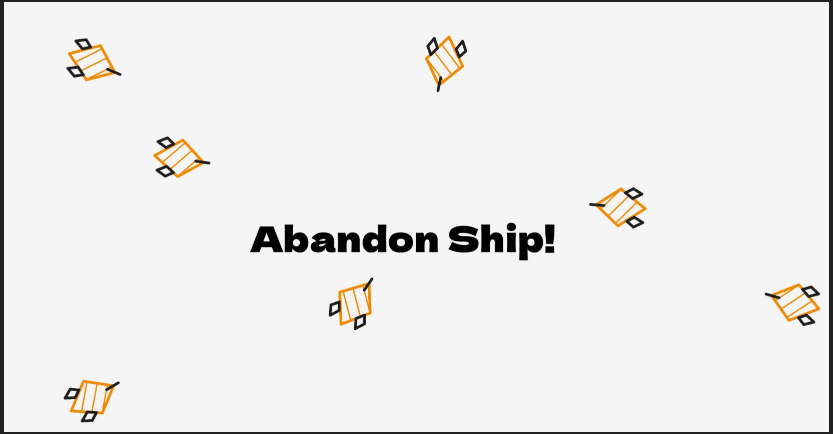
http://samskinner.net



What we will create

Arrows that point and follow the cursor location. We will use Trigonometry! Math is useful hooray!

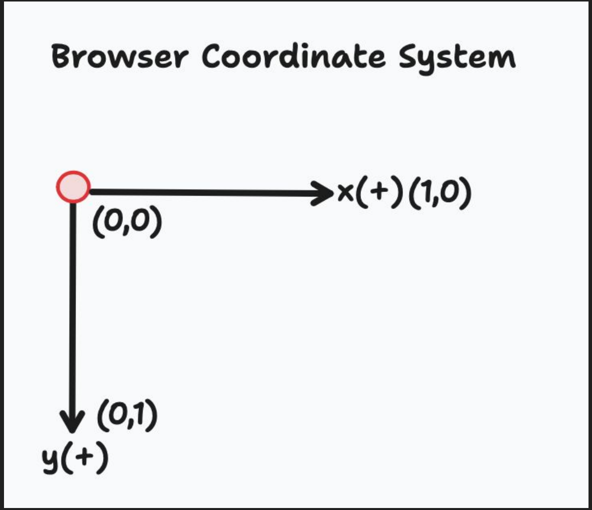




There are several things to consider

The browser coordinate system is different than cartesian coordinates because the y-axis are switched. However, the x-axis is the same!

Browser System: positive y-axis is downwards, and negative y-axis is upwards.



(Drawn by Roger)

There are several things to consider

We will not be working with a regular unit circle since we will use the built-in Math.atan2(y,x) function.

Returns: angle in Radians

Between [-Π,0]

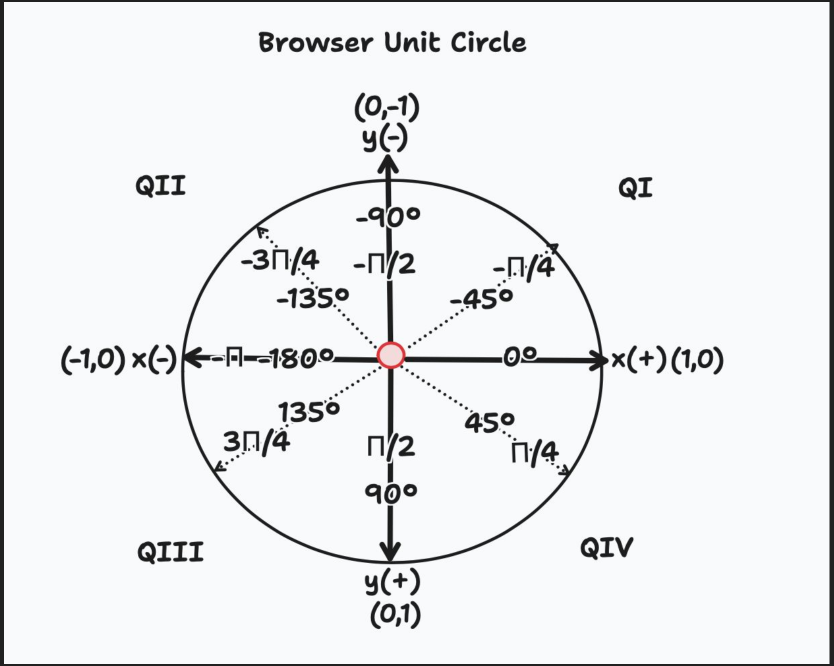
But, we will convert to degrees to make

CSS rotations possible with JavaScript.

(Drawn by Roger)

[Read about atan2 function]

[MDN read about CSS rotations]



There are several things to consider

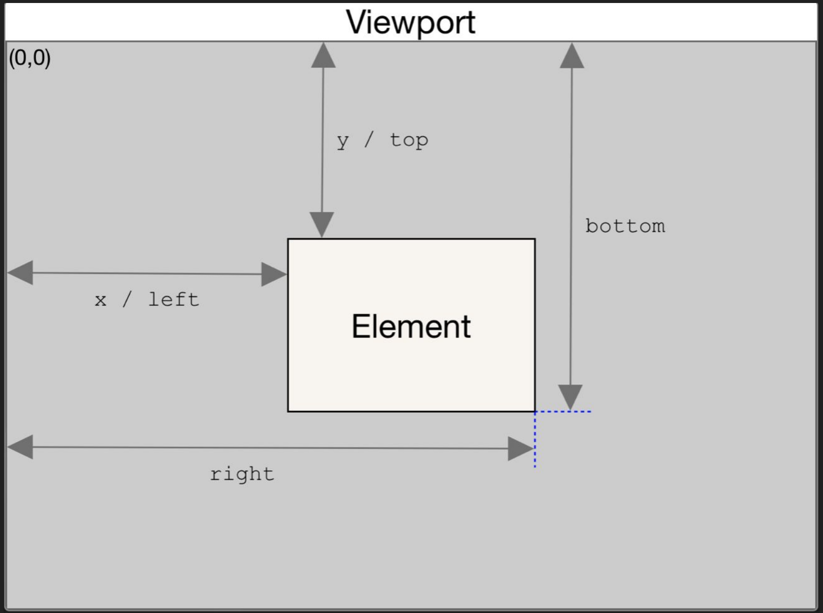
The center of an image can be found using the getBoundingClientRect() left and top properties plus the image’s width and height divided by two respectively.

Afterwards, we calculate an image’s center minus current mouse position. This ensures proper tracking rotation within [-180°, 0°] for any image no matter where we are hovering on the browser window.

An equation: convert radians to degrees

degrees = radians x 180 / Π

getBoundingClientRect()



Provides the size of element, and it’s position relative to the viewport.

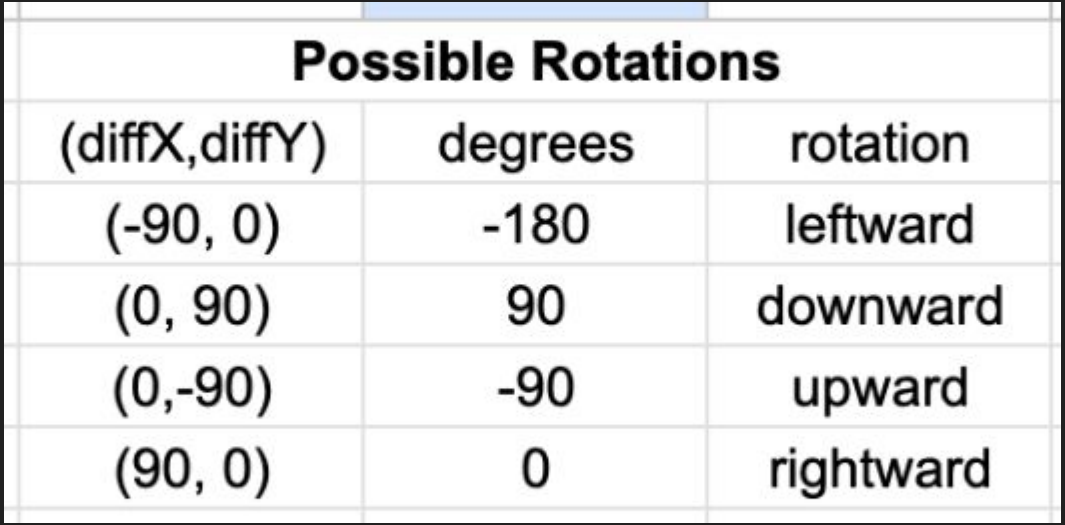
Image by Mozilla Developer Network: [read about getBoundingClientRect]

Possible rotations with the difference equation

At any given point our images will be within these rotation positions.

Result:

Images will spin and point directly at the cursor location.



Possible Rotations		
(diffX,diffY)	degrees	rotation
(-90, 0)	-180	leftward
(0, 90)	90	downward
(0,-90)	-90	upward
(90, 0)	0	rightward

(Calculated on a spreadsheet)

In conclusion

Thanks for attending!

Math can be a useful for web design!

Websites do not have to be boring.

What’s something we can bring from our surroundings and employ it as interactivity?

Code and Slides

Let’s connect:

LinkedIn