

This week, our task was to explore more on markers and multiple markers. My task was to do experimentations with markers of different size and do measurements and see what is the minimum possible marker size.

Before making various markers of various sizes, I tested how well the original Hiro marker worked with a basic setup. A 12.5 cm x 12.5 cm Hiro marker would be recognized by AR.js and display a model from up to a distance of 2 meters. This distance was also affected by the amount of light in the room and the quality of the camera used. When the room gets darker, the performance was highly affected and the distance was shorter. Also when I used my phone to test the markers, I had to move closer to the markers to make it work even though my phone has a better camera than my PC. This is probably due to the blur effect caused by the phone's inability to focus on paper on such close distance.



Figure 1. The hiro marker

Then I tested the optimum distance for Hiro markers of different sizes. Markers with size 70% and 50% of the original size were quite useful and worked without any problem but they are still very huge. It is possible to fit 8 markers in an A4 page with markers of 50% width and height of the original size which is 6.25 x 6.25. I also tried with markers of size 4.00 x 4.00, 2.66 x 2.66 and 1.33 x 1.33. The smaller 2 worked only if they are held to the camera with a distance of 20 to 40 centimeters which I believe is close to useless. However the marker with size 4.00 x 4.00 is useful up to a distance of 1 to 1.5 meter and yields reliable results. Considering these observations, I believe a square marker with an edge size of 4.00 to 6.00 centimeters are optimal. Also during the tests I observed that when 2 or more instances of a marker is present, AR.js doesn't display any model on neither of them most of the time which is I believe something nice to know.

I also tried to solve the clicking problem that happened to me on mobile testing which is clicking to an object doesn't register most of the time but if it registers the onclick handlers that I created for the objects fired 3-5 seconds later which I believe is a major problem. To fix this issue I tried to

emulate clicking on an object by clicking its marker which would cause the marker to go off camera and reenter the camera after the physical touch to the marker happened. I made the marker emit an onclick event when that occasion happens. The results are somewhat reliable and certainly more performant on mobile.

This is my work for the week. I have pushed all my work to github in case you might want to check it.

Here is the link:

<https://github.com/uygaruyaniksoy/ucd-aha>