This week, I tried to enhance the performance of the application from the last week. The application marker layout is in Figure 1 and it had a problem with marker detection i.e. even though the camera hold still and lighting didnt change, objects over the markers might disappear from the view for a frame or two and reappear afterwards.

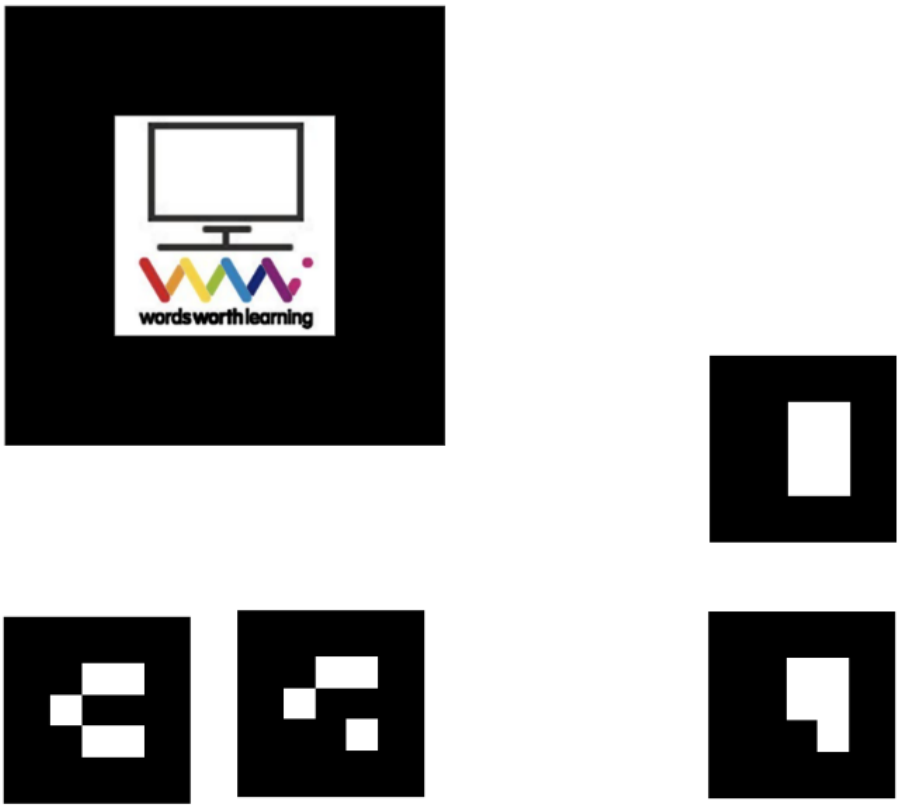

Figure 1

I had this bug while testing on my mobile phone most of the time and performance on my laptop is actually OK. To fix the issue on mobile, I tried multiple markers instead of the WWL marker in Fig. 1. I tried more complex ones and very simple ones. Also I tried the Hiro marker and ARjs since they are very well used among the tutorials, I thought they might perform better, meaning they won't disappear for brief moments. However this was not the case. All the different markers work pretty normal when they are all on their own but they had flashing problems as soon as I tested them with barcode markers that I used for hand gestures.

Upon further research, I stumbled upon a variable in a-frame's pattern recognition code where a variable named ***minConfidence*** is used to limit the recognition of patterns. A-frame tests this value with a variable that shows how confident it is about there is a marker on the view or not. I lowered this treshold value and I believe I got somewhat better results comparing to the ones before but it was still far from useable.

Finally I switched the WWL marker with another barcode marker which I got the best result. As far as I understand from examining the aframe-ar.js code, pattern recognition and barcode recognition are two very different process and I believe barcode markers are very much better on performance than pattern markers. During my final test I had 3 quick disappear and reappears on a single marker in 1 minute and the other 4 markers were all still(they didn't flash even for once)

Judging from this and the previous week's tests I believe this is the most useable way of markers. My final and most performant marker layout is shown in Figure 2.

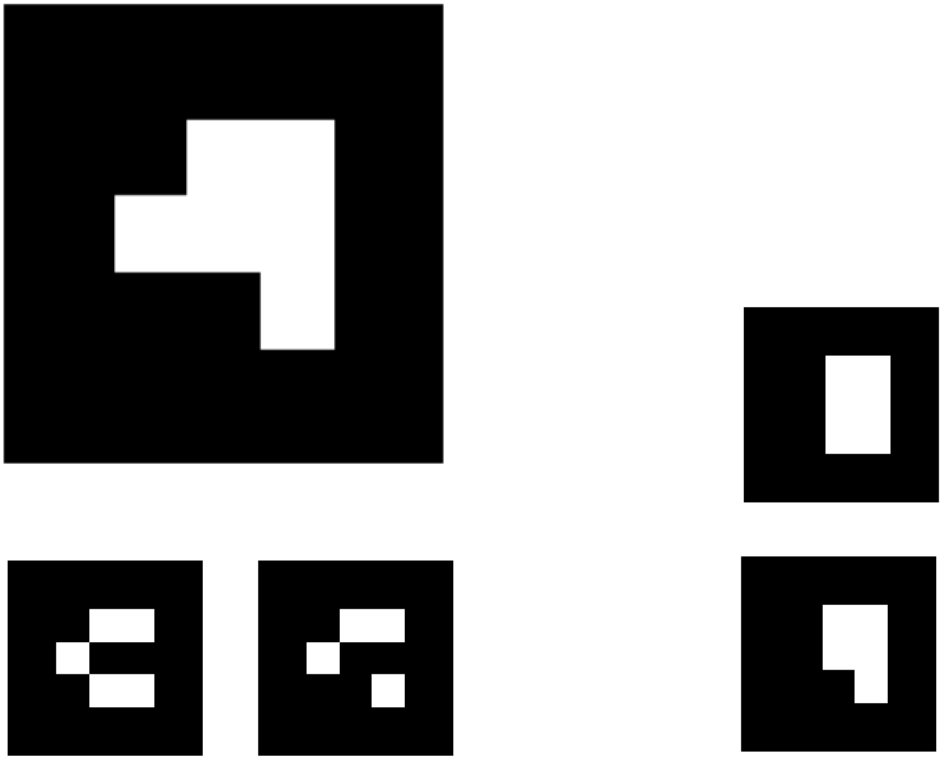

Figure 2.