Web based Augmented Reality

Equinor Virtual Summer Internship 2020, Case 09

Introduction

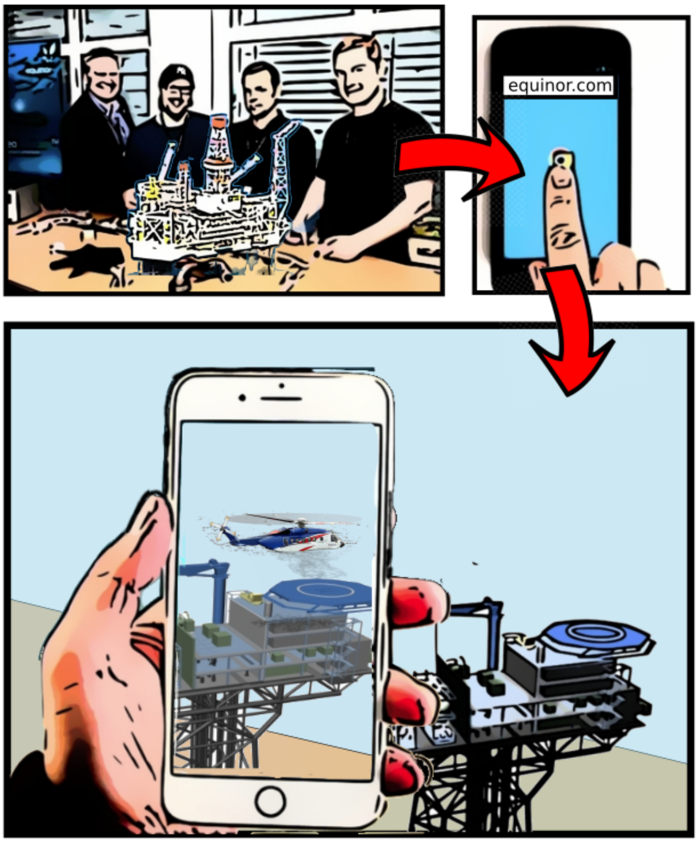

Figure 1: Cartoon showing the concept of augmented reality.

Augmented Reality (AR) is a technology that fuses Virtual Reality with the real world to construct an enhanced experience. AR is used in various arenas, but downloadable content and specific gear/software are often required. This generally makes AR less accessible, and the target group narrow. A desirable product would therefore be a web-based solution, where no installations are required, and a simple mobile device is sufficient. The main goal of case 09 is to present the main functionalities of web-based AR and evaluate if this concept is a viable choice for Equinor both in terms of development complexity, cost and convenience. To discover what is possible with the current web AR technology, we will try to solve a number of tasks to determine the quality of the current frameworks available. For example, one of the subgoals is to examine if AR could be used to get a sense of what large scale installations look like in true scale (for instance oil rigs), since these large installations are hard to visualize for people who have not seen them in real life.

Context

Equinor's development with augmented reality has been proven to be successful for saving time, reducing cost and increasing safety[1]. The current AR device used by Equinor is the Microsoft HoloLens. The HoloLens is a state-of-the-art device for AR, but it is very expensive and the accessibility for most Equinor employees is limited. Having a more accessible solution for AR could open up for more use-cases within Equinor, and therefore save time, money and increase safety. One example where this type of lightweight AR could be used is in fast prototype development, where the prototype could be enhanced by AR, showing different solutions or parts through a smartphone.

Strategic Aspects

Considering this as an innovative project, exploring the use of augmented reality is well aligned with Equinor's values. The value of being "courageous" implies that Equinor is curious and innovative. Exploring the use of an emerging technology such as augmented reality is a low-risk opportunity with potential high gains. One could argue that displaying a 3D model through AR can help understanding the tasks and risks better. In a report, PwC states that using AR technologies can increase engagement, understanding and emotion for visual communication[2]. In other words, the use of AR could support effective communication, which is another key value of Equinor.

Financial Aspects

With emerging IT such as augmented reality, concrete numbers for financial aspects are hard to determine beforehand. The future solutions and innovation this will generate is unknown, and it is hard to determine where costs will be saved. With our solution, the cost for hardware is negligible and the primary cost will be for software development. Looking at the HoloLens as an example, it has saved time and money in the construction phase of Johan Sverdrup[3]. An Equinor employee working with quality control states that using augmented reality increases the effectivity of the quality check by 4 times while also detecting more faults.

---

[1] "Echo — Equinors digitale tvilling - Equinor." 15 apr.. 2020, https://www.equinor.com/no/magazine/echo-equinors-digital-twin.html (opened 10/07/20)
[2] "The role of AR and VR in corporate communications." 28 jun.. 2017, https://www.digitalpulse.pwc.com.au/augmented-virtual-reality-corporate-communications/ (opened 10/07/20)
[3] "Echo — Equinors digitale tvilling - En interaktiv 3D-modell i ...." https://www.equinor.com/no/magazine/echo-equinors-digital-twin.html (opened 13/07/20)

Figure 2: Image showing an Equinor employee using HoloLens glasses when working with the digitalization of Johan Sverdrup.[4]

Risks

When identifying the risks involved in this case, we started by looking for upside potential. Some of these positive risks were:

- To create excitement towards AR and make way for more innovative projects
- To make AR more accessible for smaller projects
- To gain enough knowledge and skills to come up with new ideas related to AR
- To make an app able to recognize how robots behave and identify some of their specific actions

These upside risks are working as goals for the case work, and act as a big part of the desired result for the project as a whole. When establishing wanted outcomes, there is also a need to investigate the possible downside risks of executing the tasks. These risks can be related to the environment, society, safety etc. Since our case is purely software development, at least for the majority of the internship duration, it is difficult to see how our work could do any harm. When taking us interns into account, a probable risk would be getting harmed during testing of the application. When having placed large 3D models on GPS locations, one has to be careful when walking towards them and be cautious of traffic. Later, when approaching a finalized AR application product, other fatal risks could be present. An example could be if the application misclassifies a robot or misinterprets its current action, which could cause unknown consequences. Economical aspects are also part of the downside risks. If a lot of people are working on a project without achieving valuable results, there would be a considerable loss of money due to personnel salary. Larger economical losses are also possible, if errors cause some sort of expensive accident.

[4] "Echo — Equinors digitale tvilling - En interaktiv 3D-modell i ...." https://www.equinor.com/no/magazine/echo-equinors-digital-twin.html. (opened 13/07/20)

Proposed Solution

The case is divided into four parts, from now on referred to as Epic 1-4. These four parts are formulated to present different aspects of AR and scenarios where it is applicable to implement such technology. We have based our work on some well-known frameworks, such as:

- A-frame. A-frame is a web framework for building virtual reality (VR) experiences. A-Frame is based on top of HTML, making it simple to get started.[5] A-frame builds on top of three.js, which is a JavaScript library used for creating and displaying 3D computer graphics on the web.[6]
- AR.js. AR.js is a lightweight library for Augmented Reality on the Web, coming with features like Image Tracking, Location based AR and Marker tracking.[7]
- Node.js and Express.js for backend (processing player interactions and putting data into storage)
- Redis as a memory database

Epic 1 - Can web AR enhance physical 3d printed models?

The main goal of Epic 1 is to enhance a real-world object at points of interest through a web AR app. One method for achieving this is to populate the real-world model with markers of some kind at the points of interest. We explored multiple types of markers for this purpose. QR codes proved unsuitable as markers as the framework we used could barely recognize nor track them. Image tracking with custom images worked but proved lagging. Barcode markers (high contrast matrix-like figures) seemed to be the better out of the marker types available with the AR.js framework. Using barcode of size 3*3, it is possible to create 64 unique markers that the app can distinguish between. This gives the software developer the opportunity to create 64 unique interaction points in graphical user interface. Figure 3 shows the barcode for the number 0.

Figure 3: Barcode 0

[5] https://aframe.io/docs/1.0.0/introduction/ (opened 15/07/20)
[6] https://en.wikipedia.org/wiki/Three.js (opened 15/07/20)
[7] https://ar-js-org.github.io/AR.js-Docs/ (opened 15/07/20)

Figure 4: Image showing text appearing on top of barcode marker.

We printed 8 markers with the values 0-7 and put them on a guitar amplifier, but any box or other real-world object will do. For simple testing it is also possible to just point the camera at the markers, either printed or on a screen. We experienced that marker recognition does not work well with low contrast markers, nor with markers placed directly on a dark background. A solution is simply to use black/white markers with a white border around it, as seen in Figure 4. For the app to work as intended, the user will have to accept when it asks for access to camera, geolocation and device motion sensors. With the app we implemented, different actions will happen when pointing the camera towards the different markers:

- A simple text appears when recognizing a marker
- A simple 3D geometrical figure appears when recognizing a marker
- A 3D model appears when recognizing a marker
- A sound snippet plays when recognizing a marker

Figure 5: 3D model of a penguin appearing on top of barcode marker.

To enhance the interaction with the user, we implemented a cursor. When pointing at a marker with the cursor and then clicking, an event can be triggered. We added these features to the app:

- When clicking on a marker a text box appears/disappears
- When clicking on a marker, followed by clicking on another marker, a straight line is drawn between them
- When clicking on a marker, the visibility of the 3D model present on top of the marker is adjusted. For example, that it turns transparent, or partially transparent.
- When performing touch gestures on a marker, the 3D model on top of it can be rotated and scaled.

Watch the video named “Epic1.mp4” in this[8] google drive folder to see how the web app works.

[8] https://drive.google.com/drive/folders/1-D7dxKpEvbziOmaLRB-FqdZaG-fOBfFc?usp=sharing

Epic 2 - Can web AR provide users with a sense of true scale of industrial equipment?

The main goal of Epic 2 is to explore if large 3D models can be displayed in the app, if it can give a sense of true scale, and if we can create points of interest in the experience. The 3D model could be placed both with relative position and with GPS locations. To add animation to 3D models and interaction between models is also desired.

Figure 6: Screenshot from the web app showing 3D models placed at GPS locations.

Already made large 3D models can be downloaded for free from the internet, for instance from www.sketchfab.com. We chose to use files of the filetype glTF which has several advantages, such as adding animations easily, compatibility with A-frame and that it is widely used by programmers globally. We have been using a program called Blender to edit glTF models, which can be downloaded for free from www.blender.org. This program is also excellent for adding animations to 3D models. We were able to place the glTF models at given GPS locations, and they were fairly stable. But the accuracy when using GPS is not always the best. This is often due to reflections of signals and noise when inside houses or near buildings. To correct this, we implemented a Kalman filter. A Kalman filter uses data and mathematical models, in addition to the current measurements to calculate an estimate of what the current state should be. This method is often used for noisy and unstable systems and is a powerful tool. We were able to see improvement in stability when using the Kalman filter in our application.

We applied two different ways to introduce animations to the 3D models. The first one is pure HTML code using A-frame components. This is sufficient for simple animations where the model moves as a whole. This could be that it moves from a start position to an end position in a straight line, rotations or being scaled. When wanting more complex animations, embedded animations in the glTF model is a suitable choice. It is possible to run the animation a fixed number of times, a fixed time interval, in an infinite loop etc. It is also possible to pause and unpause the animation, for instance by clicking on the 3D model. We further developed some of the functionalities presented in Epic 1 to be applicable for GPS placed objects as well. This includes playing sound snippets when clicking and drawing

Figure 7: Screenshot from the web app showing a straight line being attached to two spheres.

straight lines between 3D models in motion. To see these functionalities, watch the video named "Epic2.mp4" in this[9] google drive folder. To see how the app can show large scale objects, watch the video named "Epic2.5.mp4" in the same google drive folder. This video shows the Eiffel Tower being attempted to be placed in the real world fixed at a GPS location. It is possible to see some drifting and instability, which is due to the properties of GPS signal handling.

To improve the inaccuracy issue, another solution was tried out. Some points in the model were programmed so that when the user is within a given radius of that point, the GPS location of the phone is manipulated to be fixed at that point. This improves the user experience in several ways, for instance that the correlation between the camera view and the virtual models were much more stable. This solution is especially suited for applications where lookout points are desired. We also implemented altitude manipulations, so that when walking to a lookout tower both the GPS location and the altitude are fixed to the top of the tower. We experienced that using a phone in landscape mode gives the best stability. We believe this to be the case due to sensor placements and configurations in the device. When holding the phone vertically looking upwards on tall models the models are glitching and jumping around, which is not the case in landscape mode. This could deviate between different phones.

Epic 3 - Can multiplayer AR work with web technologies?

The main goal of Epic 3 is to find out if it is possible to provide a multiplayer AR experience through web technologies. We wanted to see if we can send personalized objects to other people and play sounds when something happens.

An example of multiplayer in use: Someone is presenting a new prototype where a 3D model is displayed through AR. There are five people watching the AR presentation through their phone. If multiplayer mode was possible, the presenter could change the AR experience for everyone at once whenever the presenter wanted to. Without multiplayer mode, the attendants would have to do the changing themselves, which would be harder to synchronize and would result in a worse experience.

Multiplayer mode adds a new layer of complexity because we would need to have a module that keeps track of what all the users are doing and also send synchronized requests to them.

[9] https://drive.google.com/drive/folders/1-D7dxKpEvbziOmaLRB-FqdZaG-fOBfFc?usp=sharing

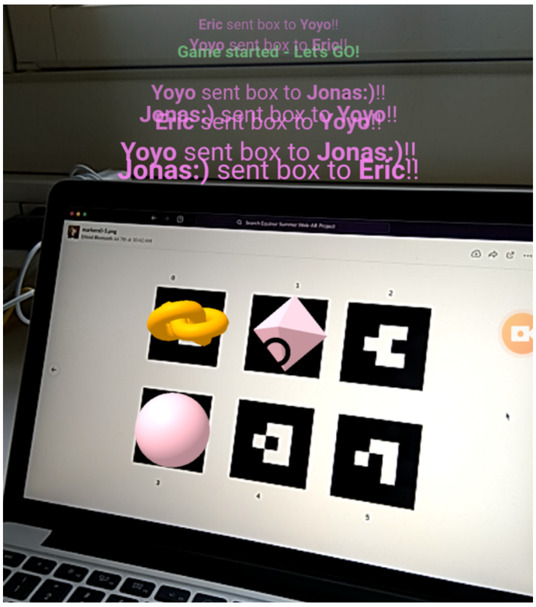

Figure 8: Screenshot from the Epic 3 game showing update messages and different 3D objects appearing on top of barcode markers.

To test the multiplayer concept, we decided to create a game similar to Whac-A-Mole[10], where one would have 3d models that were sent to another player when you clicked on them. The 3d models would be displayed on top of markers (which were previously discussed in Epic 1).

At this point we had good knowledge about the capabilities of a-frame. But to have multiplayer mode we needed a server which stored and processed data from all players that were connected to the app. After consulting with our mentor, some time went into setting up a backend server together with a memory database. At the same time, work was spent setting up a user interface for interacting with all the players models.

In the end, we managed to create a multiplayer AR game which fulfilled the main goals. In the game, you can type in a username and select a personal object. The objects are displayed on top of barcode markers that you film with your mobile camera. If you click on an object (by touching your screen), it will be sent to another player and displayed at one of their markers. When you receive a model there will be a sound announcing the new object. The game ends when one player runs out of objects on their marker board (the fastest player wins). Watch the video named “game.mp4” in [this](https://drive.google.com/drive/folders/1-D7dxKpEvbziOmaLRB-FqdZaG-fOBfFc?usp=sharing)[11] google drive folder to see a gameplay.

Epic 4: Can web AR be used for creating multiuser virtual meetings?

With Epic 4 we wanted to see if we could develop the multiplayer functionality from Epic 3 further. Our proposed solution of Epic 3 made us able to interact with each other. Our main goal in Epic 4 was to be able to “see” other users and interact with them as if they were physically in the same room as yourself. Epic 4 might be a more complex task than Epic 3 since we need to keep track of all users in relation to each other. Still, with the experience we have made and what we have learned so far, we thought it was a natural next step to implement a virtual meeting.

In this Epic, we created a virtual meeting where all users are able to see the other users in the form of an avatar. If two users in reality are far away from each other, they will still be in close proximity of each other in the virtual meeting. If one user moves, then the other users will also see him move in

[10] https://en.wikipedia.org/wiki/Whac-A-Mole (opened 16/07/20)
[11] https://drive.google.com/drive/folders/1-D7dxKpEvbziOmaLRB-FqdZaG-fOBfFc?usp=sharing

the virtual meeting. The users' positions are simply translated into relative positions close to each other.

As of today, 24/07/20, we have not implemented any possibilities of user interaction yet. However, our goal is to implement proof of concept interactions between users. One example of such interactions could be to make the users able to send text messages to each other. We also hope to implement the possibility of speaking (through the microphone) with other users who are within a close proximity in the virtual meeting

Potential use cases for multiplayer AR are already discussed in Epic 3. Epic 4 adds the possibility of "being together" virtually. One potential use case for this functionality, especially in times when most employees are at their home offices, is virtual meetings. The possibility of seeing each other as if they were close by is clearly a different experience than e.g. a phone meeting. Another possible use case is to combine an AR presentation (already discussed in Epic 3) with the virtual meeting. Using the technology this way can enhance the presentation and possibly make a better experience for the users.

Discussion

One of the objectives of case 09 was to develop a web app that is easily accessible, regardless of devices and software available. Nevertheless, we experienced some issues with compatibility. For instance, Apple does not allow the Chrome web browser to support all functionalities required for AR to work properly. And "old" phones, in our case an Android device from 2017, might not support AR functionality out of the box. In our case we managed to solve this by tweaking developer settings on the phone, but most people might not be able to make it work this way.

Enhancing physical objects through markers works well. There are big differences between different types of markers though, and we found that barcode markers have the best performance. Image tracking has a tendency to lag, but it does have other use cases. For example, with image tracking it is possible to take a picture of a building and use that as a "marker", so that the building will be recognized when you are looking at it at the right angle and light. This might be a way to have "invisible" markers in a controlled environment.

When placing 3D models with GPS coordinates instead of markers, they can often appear slightly unstable or misplaced. This is because of how GPS works, and there are several factors causing this behaviour. E.g. when inside or near buildings, the walls and other obstacles will reflect the GPS signals and accumulate signal noise. Therefore, the app works better in large and open areas, such as parking lots or football fields. This is also preferable since some of the models are quite large, requiring enough space to walk about safely.

Multiplayer mode works great, and the limitations are only on the network capabilities. Our implementations show that there is good potential in using multiplayer experiences in things like meetings and presentations.

We tried to make the app compatible with Google Cardboard. When having pure VR, thus having a 3D model without using a camera to interact with the real world, the app works with Google Cardboard. But there is an issue converting the camera recording into the correct display format. This is something that could be further developed to create an even better experience for the app user. Using the app in regular portrait or landscape mode works perfectly fine.

Conclusion

Web AR shows great promise for driving lightweight and accessible AR experiences using a smartphone. It is both more convenient and cheaper than hardware-based AR experiences (like HoloLens). There are many features available and it is possible to create complex AR experiences by using a simple framework like A-frame.

However, at the current state of web AR, there are some compatibility and performance concerns. The compatibility is easy to fix by just getting a compatible smartphone and browser. But the performance concern is due to limitations in network and hardware. Therefore, experiences that require a lot of computing might not work optimally in web AR.

If you are willing to pay for webAR technologies, there are also libraries like https://8thwall.com, which provides more features and better performance than the current free open source alternatives.

Epics

Epic	Evaluation	Notes
1 - Can web AR enhance physical 3d printed models?	Yes	Possible to display stable models and text on barcode markers. As well as play sounds on POI.
2 - Can web AR provide users with a sense of true scale of industrial equipment?	To some extent	It is hard to get a true sense of scale due to the inaccuracy in gps and the limitations in comparing the model to real life objects. But the experience is not horrible, and the interaction between objects and POI works well.
3 - Can multiplayer AR work with web technologies?	Yes	It works well. Pretty much anything in the a-frame scene is modifiable through a multiplayer experience. Easily replicable to presentations, for example.
4 - Can web AR be used for creating multiuser virtual meetings?	Yes	Works well.

Features

Feature	Evaluation	How
Enhance physical objects with text and models	Yes	Markers, events
Play sound at point of interest	Yes	Markers, GPS, events
Show true scale	To some extent	GPS
Multiplayer	Yes	Backend, database

Methods (A-frame, ARjs)

Method	Evaluation	Notes
Image tracking	Lagging	
QR code tracking	Not recognizing well	
Barcode tracking	Adequately	Best performance of the markers.
Location-based AR (GPS)	Ok	The objects jump around a bit because of the GPS inaccuracies. This was improved with two methods: Kalman filter and viewpoint spots.

Compatibilities

What	Model/Type	Notes
Smartphone	Android	Need to check if the smartphone has ARCore. Works with Chrome browser.
	iPhone	Need to check if the smartphone has ARKit. Works with Safari.
Browser	Chrome	Need to enable webXR flags in chrome://flags.
	Safari	Works well.

Future

The AR/VR industry is changing fast. It’s fairly safe to say that we can expect more web-specific features and libraries in the years to come. As well as improvement in the compatibility issues. More and more smartphones will be compatible by default, and web browsers will go from having experimental AR modes, to running AR by default. This will make web AR very accessible, and that is where the power lies. Instead of needing dedicated hardware, AR experiences will be available to anyone with a smartphone.

In regard to performance, with the emergence of WebAssembly it is now possible to compile efficient languages like C++ for the web. Thus, the future of web AR will likely have better performance, which makes it possible to use features that now are running poorly on the web (image tracking, for example).

All in all, the future of web AR looks promising.

Appendix

- https://github.com/equinor/eit-web-ar