***Steps that can be taken to make the current app/project better***

* *Remember that any additional features we decide to add or if we decide to work on improving a current feature in the project, it should be backed up by claims (for example a research paper or an article) that support the fact that adding/improving that said feature will improve the learning experience for the user so we can write down why we worked on that specific feature in both the documentation and the research poster*

***Software Aspect of the Project***

1. **Implementing Video Playback to improve the learning experience**:

* Video of how to use the 3D printer on top of a marker object in a plane (set the plane as a target and configure relevant tracking parameters by visiting Machyne)
* Import the Unity VideoPlayer Package and other UI packages as required
* Add UI (such as buttons) to help the user control the video playback

The following video shows how we could exactly implement this feature.

[Video Playback Unity 2020 & Vuforia 8 Tutorial in 5 minutes! | Augmented Reality Apps](https://youtu.be/PyBDJvs0y1Q)

***Why should we implement this feature?*** (relevant to the Research Poster and documentation in the future)

*The following Research Paper showed that video-based learning can be more engaging and can facilitate learning by providing visual and auditory cues that can aid in comprehension.*:

<https://www.uky.edu/~gmswan3/544/9_ways_to_reduce_CL.pdf>

*Mayer, R. E., & Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. Educational Psychologist, 38(1), 43-52.*

It also found that college students who learned from video lectures scored significantly higher on exams compared to students who learned from written lectures.

**I’ve also included a literature review of this specific research paper under the folder “Literature Review,” further explaining why we might want to implement this feature and also includes implementation of other features. I basically explain the concept of “cognitive load” and how focusing on that aspect for the project can help improve the learning experience.**

1. **Further User Interaction Options, once the user sees the video or goes through the tutorial of operating the 3D printer using the AR headset, to enhance the learning experience:**

These are the following UI’s that come to my mind and something we could implement in our project:

* **Quizzes :**

Help to engage users and assess their understanding of the 3D printer or any other equipment at Machyne. Include MCQ’s, true/false questions, or short answer questions to test users' knowledge on how to operate the machine and provide them with feedback on their performance. Personalized Feedback can help motivate the user and provide them with guidance on how they can improve their learning. We could also implement a Progress Tracking Feature to help users to see how far they have come. This data could also be saved on the server and accessed based on user login and password (so basically we could implement a login feature that saves the users data -> this could also solve the security aspect of the project). We could also create a Progress bar or other visual indicators to show users how far they have progressed in learning to operate the machine. We could also gamify (making it more gamelike and teaching the user about a topic in a fun interactive way) the app by including rewards and the heart system if the user gets an answer on, for example, the quiz wrong. Once again, this will only be relevant to the user if we set up a log-in system and save the data on the server so it can be retained once the user logs back in

*How could we do it?*

The following video shows how we could implement the quiz feature *(this example is a 2-D one but we can convert it into 3D that the user can interact with in an AR setting by selecting “AR” as the project type in Unity and then adding 3-D content on a plane by using Blender or Maya):*

[How to make a Quiz Game with Multiple Choices in Unity](https://youtu.be/G9QDFB2RQGA)

[How to make a Quiz Game in Unity (E01. UI) - Tutorial](https://youtu.be/g_Ff1SPhidg)

We will need to create a script, which will include variables to store the quiz questions and answers, as well as handling user input and updating the UI elements. . We also may want to create a system for keeping track of the user's score and providing feedback on their performance.

To create a progress bar, we will need to create a Canvas and a Slider in the Unity Editor. You can use the Slider component to create a horizontal or vertical progress bar, and you can use the Rect Transform component to set the size and position of the progress bar. We will also need to create a script to control the progress bar’s current value and maximum value, as well as handling the user input and updating the UI Elements. We can add the progress bar to the AR scene by creating a {GameObject} to represent the progress bar and then attach the C# script to it which controls the progress bar to handle user input and update UI elements.The following tutorial might help too

[EASY Unity Progress Bar Tutorial + Particles! [2019]](https://youtu.be/UCAo-uyb94c)

[Login and register system in Unity (UI) - Unity UI tutorial](https://youtu.be/PIA-4BUJfo0)

***Why should we implement this feature?*** (relevant to the Research Symposium Poster and documentation in the future)

According to the following research papers, quizzes can be an effective way to promote learning and retention of information:

* Barkley, E. F., Cross, K. P., & Major, C. H. (2014). Collaborative learning techniques: A handbook for college faculty. San Francisco, CA: Jossey-Bass.

<https://download.e-bookshelf.de/download/0002/5216/40/L-G-0002521640-0003712114.pdf>

* Kulhavy, R. W. (1979). Student self-testing: An investigation of the testing effect. Journal of Educational Psychology, 71(4), 523-532.
* Maki, P. L., & Sergiovanni, T. J. (1982). A case for the provision of feedback. Educational Leadership, 40(2), 63-67.
* Hwang, G. J., Chen, N. S., & Tsai, C. C. (2014). The effects of collaborative learning with augmented reality on problem-solving ability. Educational Technology & Society, 17(1), 325-336.
* Lin, C. H., Chen, H. Y., & Chiu, C. M. (2015). The effects of collaborative learning with augmented reality on problem-solving skills, critical thinking skills, and engagement. Educational Technology & Society, 18(3), 271-283.

Quizzes with feedback can help to enhance the learning experience in an augmented reality (AR) setting by engaging users and providing them with specific guidance on how to improve their understanding of the material. According to research, quizzes can be an effective way to promote learning and retention of information, and providing feedback on quiz performance can increase students' self-esteem and confidence, leading to higher levels of engagement and achievement (Barkley, Cross, & Major, 2014).

One study found that quizzes in an AR environment improved students' learning outcomes and engagement, compared to traditional classroom instruction (Lin et al., 2015). Another study found that quizzes in an AR environment increased students' motivation and self-directed learning, compared to traditional lecture-based instruction (Hwang et al., 2014)

Furthermore, a study found that students who took self-graded quizzes after reading a textbook had better retention of the material than students who did not take quizzes (Kulhavy, 1979). Another study found that students who received feedback on their quiz performance had higher scores on a final exam than students who did not receive feedback (Maki & Sergiovanni, 1982).

By engaging users, providing them with specific feedback, and encouraging them to actively engage with the material, quizzes with feedback can help to improve learning and retention of information.

*(add more research papers supporting how implementing the progress bar feature and security log-in information can help motivate the user learn and improves the learning experience in an AR environment)*

* **Collaborative learning:**

Collaborative learning can help to engage users and encourage them to work together to solve problems or complete tasks. In an AR learning experience, you can create collaborative elements such as group challenges to encourage users to work together and learn from each other, consequently receiving feedback.

The Summer 2021 team suggested using the Cloud Anchoring technique in order to allow multiple users to interact with AR in unity. We could do this by setting up a cloud anchor service provided by cloud-based AR platforms such as ARCore Cloud Anchors or ARKit Cloudanchors. We could then implement this cloud anchor functionality into our project by setting up a cloud anchor functionality in the real world and then sharing the achor’s ID with the other users. We will also need to implement tracking and calibration functionality to ensure that the virtual objects remain stable and aligned with the real world. This is **really hard** but doable and would make the functionality and use case of our project so much better. It could even be useful to make a fully virtual object of the machine the user wants to use so they can use it at the comfort of their own homes before it arrives or they physically use it.

[Unity AR Foundation - How To Use Google AR Cloud Anchors?](https://youtu.be/EuKW9RjORdk)

***Why should we implement this feature?*** (relevant to the Research Symposium Poster and documentation in the future)

According to the following research papers, collaborative learning has been shown to be effective at promoting learning and retention of information, particularly in an augmented reality (AR) environment:

**Both of the research papers cited can be found above in the section explaining quizzes**

One study found that collaborative learning in an AR environment improved students' ability to solve complex problems, compared to traditional lecture-based instruction (Hwang et al., 2014). Another study found that collaborative learning in an AR environment increased students' problem-solving skills, critical thinking skills, and engagement, compared to traditional classroom instruction (Lin et al., 2015).

* **Improving the aesthetics of the UI (e.g. the text) to make the app look more professional:**

To improve this aspect of the project, we should research design principles and guidelines for creating effective and aesthetically pleasing UIs. (we should look at research papers and articles that provide evidence-based recommendations for designing user-centered interfaces. Some useful design principles to consider include usability, legibility, simplicity, and consistency.)

Use the grid system + animations and transitions

We could also use layout and composition to organize our UI elements more efficiently.

We could also not allow the user to press the next button (which moves to the next instruction) until they have acted on the instruction that is displayed on top of the 3-D printer (image detection?)

Lastly, we could have a question mark for each page that the user can click if they want to see further instructions.

1. ***Including more machinery and equipment in the app for the user to learn within the makerspace:***

* Test and research whether the same user performs better or worse in some machines then others, which could consequently result in research findings on why this happens (external factors, the position of the machine or is it just naturally harder to learn how to use?)
* Will need manuals of the other machines to implement this feature.
* This can be done by expanding the library of prefabs within unity

1. **Including a fully virtual machine that the user interacts with instead of their being a physical machine with just pop-ups and UI’s next to it:**

* Can be useful if we want the user to pass a certain quiz score (as mentioned earlier) for them to progress to use the physical machine
* May reduce the chance of the user causing damage to the machine, resulting in a more fruitful learning experience

1. **Publishing the app in the appstore or playstore:**

* May provide better usability testing as the app would be accessible to a large amount of users
* Can be a great opportunity to implement a remote update system to allow the administrators to add features and iron out bugs. This can be done by setting up a version control system such as Git, a build server that automates the build server and creating a script that automatically checks for updates within the server (might have to use an API for this)

1. **Improve smoothness of application**

* This can be done by optimizing the assets through reducing the number and size of textures, models, and other resources.
* Also insure that all the assets are properly optimized for the mobile devices.
* Use the Unity Profiler -> *The Unity Profiler is a powerful tool that allows you to identify performance bottlenecks in your AR application. You can use the Profiler to see how much time is being spent on different tasks, such as rendering, physics, and AI, and to identify areas of your AR application that may be causing performance issues.*
* Use a lightweight AR SDK
* Enable Multithreading -> *Unity's multithreading capabilities can help to improve the smoothness of your AR application by allowing certain tasks to be performed in parallel. You can enable multithreading in your AR application by using the Unity Job System or the Burst Compiler.*
* Use Occlusion Culling -> *Occlusion culling is a technique that allows you to reduce the number of objects that are being rendered by the GPU (graphics processing unit). This can help to improve the smoothness of your AR application by reducing the workload on the GPU. You can enable occlusion culling in your AR application by using the Occlusion Culling window in the Unity Editor.*
* The previous team also recommended removing initial popup animation for component popups. (I don’t know what this means until I try out the app)

1. **Implementing Voice Commands to interact with the UI**

* This could certainly make our project “cooler” but we still have to look for research papers that support the fact that adding this feature will improve the learning experience for the user so we can support why we added this feature in both the documentation and the poster.
* We could do this by setting up the Unity Microphone class, which provides access to the system's microphone device.
* Create a C# script to both handle and process voice commands
* Might need collaboration with the hardware team as well

***Hardware Aspect of the Project***

1. **The previous team thought of using the following DIY Augmented Reality Headset:**

[**cheApR**](https://www.instructables.com/CheApR-Open-Source-Augmented-Reality-Smart-Glasses/)

* They never did though as it was hard, so communicate with Sevinch whether she can work on this and whether it can even work well with the software.