Using Augmented Reality to improve adherence to laboratory protocols and reduce user errors.

Ricardo Danza Madera
Handledare: Simon Gollbo

“Bakgrund”
This idea is a result of wanting to make working in a cluttered environment with strict protocols easier by using Augmented Reality technology. The development of the application will primarily be done with PrecisIT at their office in central Uppsala. Testing of the software will, if possible, be done at BMC. This should be the case given the fact that I have contact with students and teachers from the BMC campus.

Environments like laboratories use a very wide array of instruments and many of the tasks require adherence to strict protocols to, among other things, ensure the safety of the user/operator and ensure that there is no contamination of samples or bacteria cultures.

There are interesting examples of similar projects being developed by big companies such as GE Healthcare quite recently.
https://www.ge.com/reports/game-augmented-reality-helping-factory-workers-become-productive/
As well as a doctoral thesis done at Uppsala University:
http://uu.diva-portal.org/smash/record.jsf?pid=diva2%3A954225&dswid=1798

"Uppgiftsbeskrivning"
The goal of this thesis is to produce a working prototype that will use Augmented Reality in a laboratory environment. This working prototype should be able to follow an object in space and warn if that object is about to be used in a wrongful way. An example of this is when creating bacteria cultures. There are three items of interest during this process: a burner, petri dishes, and a metallic inoculating loop. When creating bacterial cultures on several petri dishes, the simplified procedure is as follows:

1. Dip the inoculating loop in the bacterial solution.
2. Swab the bacteria on the petri dish. Streaking is the technical term for this action.
3. Close the petri dish and put on a stack of completed petri dishes . (some sterilisation steps omitted)
4. Place the inoculating loop above the burner until the metal becomes almost white. (Sterilisation)
5. Wave the inoculating loop in the air until it cools down, otherwise it can kill bacteria in the next step.
6. Go back to step 1 and repeat the procedure for the remaining petri dishes.

The process described above is important to make sure that there is no cross-contamination of the cultures. The main goal, therefore, is to make sure that the handling protocols and procedures are executed correctly. For this goal to be accomplished, the application needs to be able to correctly identify when each step in the process has been carried out as well as warning if the user is about to skip a step.

Another part of the main goal is to be able to display basic information about the instrument on the screen after it has been picked up by the user. This can be for example: the last date the instrument was calibrated, whether the instrument has been sterilised or not, any important information or detail that should be known to the user before use. (Need more insights from the biologists and chemists regarding this).

"Tillvägagångssätt"
The main tools used for developing this will be simply a computer and a web camera which can be positioned so that it "sees" the entire workspace. The programming language used will be mainly C++ with the OpenCV library. OpenCV provides functionality that makes the

development of an AR applications more efficient. OpenCV is able to track physical objects without much problems, the C++ code also makes the code run relatively fast compared to other alternative languages. The result can be tested empirically, making sure that the program is able to complete the tasks outlined in the previous part. A qualitative aspect of this can be evaluated using user trials.

“Relevanta kurser”
Most of the general programming courses are relevant to this thesis, the list below contains courses that are especially important.
● Imperative and Object Oriented Programming Methodology.
● System Design.
● Database Design.
● Linear Algebra

“Avgränsningar”
The application will be developed for one device. To save time and focus on the core of this thesis, the application won’t be fine tuned to run on all possible Android devices. Another limitation is that it requires more sophisticated programming to keep track of objects if one faces the camera completely away from the workspace and then returns. Therefore, we assume that the camera will be focused on the workspace at all times, whether it be on a headset or a tripod.
Possible augmentations could be :
● Training mode: The user sees on the screen what the next step should be. This could be used for training.
● Voice activated commands. This would reduce the need to interact with fingers. Amazon has a service that can be used to this end. The set of steps described previously is just one of many protocols in a lab. It would be more comfortable for a user to activate a certain protocol by using keywords and no need to touch any contaminated surface.
● Storing information for each of the instruments in a database so that any user can access all the instruments in the lab.
The fact that these items are classed as “possible augmentations” implies that they are not required to be solved as part of the thesis project.

“Tidsplan”
Report writing will have time allocated every week process throughout the candidate thesis.
Week 13 -14: Research and program design. Meet with biologist and get more insight and feedback
regarding their protocols. More in depth look at software, frameworks and tools required for the implementation of the project. Testing Android Studio and ARCore. Write code to test these tools and
frameworks, this code need not be part of the thesis project but to understand possibilities and limitations.
Week 15-16: Start writing program, test it recognises objects with QR codes attached and can follow

such objects in space. Display information about the object.
Week 17-18: Recognise different actions. For example: Change the state of a tracked object when it
recognises that the object has been sterilised. Trigger a warning from the program.
Week 19: Full working prototype.

Week 20: Improvement of the prototype. Making the recognition of events more dynamic, working with
the possible augmentations mentioned before, or buffer in case implementation takes longer.
Week 21 - 22: Pure Writing