AR Chess interface

1st Mitchel Bloch
100742293
Design and Research

2nd Alan Dong
100752362
3D Modeling, UI assets

3rd Noah Glassford
100743060
Programming

***Abstract*—While we originally planned on developing a chess application that used computer vision and augmented reality, this was more out of scope than we realized. This paper will be looking at the paper prototyping techniques we used to get around the technical limitations and the evaluation methods that we used to evaluate our design.**

***Index Terms*—Augmented Reality, Computer Vision, Chess Engine**

I. Introduction

We chose a chess related project at the start of the semester due to the boom in popularity chess was experiencing, with the biggest online chess website chess.com growing from 7 million to 10 million users in less than a month. [1] Our project was planned to use a combination of augmented reality and computer vision in order to provide a more user friendly and more visual interface for chess engines such as stockfish. However, we experienced many problems with both computer vision and augmented reality and had to rely on paper prototyping. The main problem we experienced for computer vision was the extreme negative effect it had on performance, as well as finding a good way of converting the data into positions that could be used by unity. The main problem we experienced for augmented reality was inconsistency. While we intended to use paper markers to locate the chess board and pieces, the system was still extremely sensitive to any changes in lighting or camera angle. Augmented Reality also has an issue in Unity where things are quite floaty and not fully locked down to a position or rotation, so keeping the chess board lined up with the markers was also quite inconsistent.

II. Methods

For developing the app we used Unity with the Unity AR Foundation package and OpenCVSharp for computer vision. In order to track the chess board, we first tried to just detect a chessboard pattern using the image detection tools that are built into the AR Foundation package. This worked sometimes, but not consistently enough to provide a good user experience. We also tried using a marker specifically made for AR, and while this one worked fairly well for tracking, it was also the size of a full sheet of paper and visually did not match a chess board at all, so that implementation wouldn't really work that well. We also weren't able to use computer vision in as large a scope as intended, but this was due to performance issues when running computer vision code on mobile devices. In order to try and replace this computer vision code we created markers for the pieces for the AR Foundation package to track, however, as mentioned in the introduction, these markers were very inconsistent and very sensitive to lighting conditions and camera angles. In the early prototype stage, Alan and Mitchel tested the prototype app as actors. They each played 2 games on chess.com which uses the same chess engine that we planned to use in our AR chess app. During the in class playtest, the AR detection was not working properly, so the users were just told the information that the app would have given to them. Later on, we created small paper markers to place on the board to communicate the information in the more visual way the app intended to. We intended for these markers to work in AR, however sensitivity to lighting and camera angle made this very impractical.

III. Results

For the app, we were able to get a chessboard to render in AR and track to a marker, however we were not able to track the position of the pieces or do an AR overlay. Unfortunately, those two things were the major technological component of our app. However we were able to recreate very similar functionality to our proposed design using a paper prototype. The app was tested in class by classmates who were then asked to fill out a survey. Due to the issues we encountered with the app we had to use Noah as a proxy for the Stockfish software to suggest moves. It is unclear but assumed that due to this the results we got from testing were skewed. We had an average overall SUS score of 71.5 which is considered above average, a 10 score average on ease of learning and an 11.5 score average on prior experience needed, meaning that most people can just pick up the app and use it without much issue. Our lowest average score was cumbersome while using, this was most likely due to how difficult it was to get AR to recognize the board and display the image properly on top of the board. As for PQ, we get a total score of 401/475, broken down into Realism 138/175, Possibility to act 93/100, Quality of interface 68/75, Possibility to examine 63/75, Self-evaluation of performance 39/50. We got a high score in possibility to act as the user has a lot of choices of which pieces to move from the suggestions. The highest demand for the TLX was physical demand at 10.5, as both lifting and positioning the phone to point at the chessboard can take a lot of physical effort. With the user having to do this a lot while using the app, it could become tiresome to them.

IV. Conclusion

We have found that the app design was easy to use and get into from our SUS and TLX data, the app design has a simple and easy to use interface and a new user can get used to it in a short period of time. We also found that AR improves learning capabilities in various other education applications [2], including chess. Due to the way that AR visualizes all the information while also being interesting enough to keep the user entertained and motivated enough to keep using the application. During the testing process we also found that The AR recognition relies deeply on the camera aligning with the chess board, as well as having good lighting in the room. This led us to using markers, which allows a more accurate tracking for the app, however it is still not enough to make the AR reliable enough to be used for the app.

References

[1] https://www.chess.com/blog/CHESScom/chess-is-booming-and-our-servers-are-struggling

[2] Izquierdo, J. L., Alfonso, M. R., Zambrano, M. A., Segovia, J. G. (2019). Mobile application to encourage education in school chess students using augmented reality and m-learning. RISTI: Revista Ibérica de Sistemas e Tecnologias de Informação, 2019(22), 120–133.