Stefano, Austin

Software description

All parts needing to be manufactured must be modeled in a solid-modeling program such as SolidWorks and appropriately dimensioned such that sponsors can fabricate more parts in the future. Isometric views and orthographic projections of each part should be produced. Assembly drawings should also be included. Standard engineering practices for labeling and dimensioning drawings should be followed, including a logical part and assembly numbering system. For ECE only groups, this requirement can be relaxed or we can ask for assistance from the MAE Department.

As appropriate, circuitry should be modeled using an appropriate modeling tool. Schematics shall be included with appropriate documentation so that sponsors can modify the part in the future.

---------------------------------------------------------------------------------------------------------------------

The AR/VR Blackjack project relies heavily on software. Software is being built to run an AR game on the Microsoft Hololens. The software is built by using the Unity Editor. With the Unity Editor, it is possible to build a game for the Microsoft Hololens featuring graphic elements for the UI, as well as the logic required to run the game correctly. This logic includes the math required to calculate the probabilities of winning each round of Blackjack.

So far into the project, code has been obtained from a github repository to be able to perform a TCP connection via a python script, which allows communication between Python programming and the game app for the Hololens. Details on this process is that there is a Python script: “FinalServer.py '' which creates the server via a wi-fi connection. It does this by importing the module “socket”. This allows the creation of socket objects which help communicate and create the TCP server with the computers ip address, and the desired port number.

To run this Python script successfully, it is required to run on a Linux OS. This is due to conflicts of the use of file handles in Windows. Thus, Oracle VM Virtualbox will be used to run a virtual machine which has Ubuntu 16.04.7 installed to run this script.

The Unity Editor has its own C# script: “UnityClient.cs” which serves as a client to the server which allows it to receive the data being transmitted by the created Python server. This data is suspected to be the results of the probability math algorithm.

Other software that is being created for this project is the Unity Program solution, and scripts which help control Game Objects, as well as the Game Objects themselves. Game Objects allow the creation of the AR camera, UI, text for UI, and music being used for the AR/VR Blackjack assist tool. Furthermore it allows the use of Barracuda to be able to launch the object detection algorithm: “YOLO”. To do this a UNNX file provided by the github repository is used to launch YOLO, and to train the algorithm by launching a python script which performs the training for the algorithm to detect playing cards correctly. To train the algorithm, many photos of different playing cards must be fed in.

Finally, to put everything together, a C# script will be built to be able to send the object detection data over to the card probability python script: “blackjack.py”. The blackjack.py script is responsible for calculating whether it is more beneficial to stay or hit in the game of Blackjack. It does this with the help of importing the package “numpy”, which helps by easily creating arrays/tuples, and to easily perform math functions like finding the max, or the sum of these arrays. Additionally, it allows the creation of arrays filled with 0’s. Using numpy makes it easier to write the probability algorithm.