Austin Hobbs

**Problem Description**

The problem should be outlined, and all problem requirements and engineering parameters should be discussed. This portion of the report includes summarized information from the Engineering Specification Report.

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

Originally, the issue to be solved was a wireless communication & power consumption problem. We have an AR device (the Microsoft HoloLens) and we wanted to improve the runtime & battery life by rendering our game of choice on the host computer wirelessly. Ideally, since the rendering will not be done on the hololens, we expected an improvement. The following lists our current requirements.

**Project Requirements**

The following is a list, in order of importance according to the sponsors, requiring the remote rendering device to:

* accompany a set of documents specifying every piece of software and its functionality
* allow the AR/VR device will connect to a wifi-router in order to communicate to the host computer
* perform the app rendering and processing will be done on the host computer
* improve runtime and battery life
* allow the app rendering and processing will be done on the host computer
* identify the playing cards that a person is holding
* determine the value of the cards that are given to the player
* determine the value of the cards held by the dealer
* display the value of the cards that are on the table
* display the statistics for next move that the player should perform
* determine if the player should hit, split, or stand for their next move

**Parameters and Target Values**

To ensure each of the project requirements are met, verifiable engineering parameters and accompanying target values are identified as follows. This remote rendering device will:

* be implemented using a HoloLens
* have a total system latency will be under two seconds
* improve battery life by at least TBD
* display the probability of success for each move the player could perform
* display a graphic and play a sound when player hits a card count of twenty-one
* play a sound when player wins the game

For this project, the object detection was done separately from the game logic. We successfully got object detection working correctly & the game logic to successfully read the probabilities from the game logic script in Unity.

Here, we’ll go over what requirements we were able to get done, and what we didn’t get to.

**Completions**

* *The AR/VR device will connect to a Wi-Fi router in order to communicate to the host computer* ▶ We successfully connected the HoloLens to our provided router and successfully setup communication (outside of a game)
* Determine if the player should hit, split, or stand for their next move ▶ In our Unity game, we successfully were able to detect what should happen next via the probability script. The UI renders correctly, and successfully updates the on screen probabilities with a given set of dealer & player cards.
* Determine the value of the cards held by the dealer ▶ YOLO was able to successfully identify playing cards, however once integrated, the code would need to differentiate cards held by the dealer from cards held by the player.
* Display the probability of success for each move the player could perform ▶ This is done via the probability script, and is successfully printed to the screen.
* Display a graphic when player hits a card count of twenty-one ▶ An overlay (coins) was made to show when a player hits 21 or wins.
* Display the statistics for next move that the player should perform ▶ Statistics are handled through the probability script.
* Play a sound when player wins the game ▶ After a lot of messing about with Unity, we finally got sound to play correctly when a player wins.
* Identify the playing cards that a person is holding ▶ YOLO successfully identifies cards in the webcam scene.
* Determine the value of the cards that are given to the player ▶ YOLO outputs the value it sees of each card it successfully identifies.
* Display the value of the cards that are on the table ▶ YOLO identified the cards on the table if a webcam is situated above the table.
* Accompany a set of documents specifying every piece of software and its functionality ▶ Documents were created (here with this Final document) and on GitHub outlining our current work.

**Failures**

* Implemented using a HoloLens ▶ We unfortunately could not get the integration between all parts to work as one unit, so this was not able to be implemented on the HoloLens.
* Total system latency will be under two seconds ▶ In this case this requirement fails since the system is no longer on the HoloLens itself.
* Improve runtime and battery life ▶ In this case this requirement fails since the system is no longer on the HoloLens itself.
* Perform the app rendering and processing on a host computer ▶ In this case this requirement fails since the system is no longer on the HoloLens itself.

From above, we found ~73.33% of the requirements were met.