Amber Dolezal

**Methodology and Procedure**

This section contains the same info as the Methodology and Procedure from the Test Plan.

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

**Methodology**

This AR remote rendering application will be tested utilizing the Hololens 1, Wi-Fi, a computer, and a bench-mark app (non-remote rendering).

As a team we were unable to achieve full integration with the object detection and unity application. Therefore, we were only able to conduct a test for each class separately, as well as testing all the classes combined within Unity minus the Object Detection input.

**Procedure**

**Testing Classes individually**

1. GameLogic
2. Create two variables, one for the total card count for the player and one for the dealer.
3. Change the variables so that you can test each possible outcome.
   * 1. Settings
        1. Surrender - Player hand = 0, Dealer hand = Any
        2. Player Busts - Player hand > 21, Dealer hand = Any
        3. Continue Playing - Player hand = Any, Dealer hand <= 16
        4. Player Wins - Player hand > Dealer hand, Dealer hand != 16 or Dealer hand > 21 and Dealer != 16
        5. Dealer Wins - Dealer hand > Play hand and Dealer hand != 22
        6. Push - Player hand = Dealer hand
4. Blackjack (probability)
   1. Launch a Jupyter notebook
      1. Open your terminal and navigate to the directory where you would like to save your notebook.
         1. Type the command jupyter notebook, the program will instantiate a local server at localhost: 8888 (or another specified port)
         2. A browser window should immediately pop up with the Jupyter Notebook interface, otherwise, you can use the address it gives you.
      2. Create a new notebook
         1. Go to New and select the Notebook you’d like to use or Click Upload a Notebook you want to use.
         2. Notebooks currently running will have a green icon, while non-running ones will be gray. To find all currently running notebooks, click on the Running tab to see the list.
      3. Run Cells
         1. Cells are how notebooks are structured and are the areas where you write your code.
         2. To run a piece of code, click on the cell to select it, then press SHIFT+ENTER or press the play button in the toolbar or you can utilize the Cell dropdown menu which has several options to run cells.
      4. After a Cells are ran
         1. The output of the cell’s code will appear below it. To stop running a piece of code, press the stop button.
      5. Example run
         1. In [1]: Entire blackjack class
         2. In [2]: game = blackjack()
         3. In [3]: game.newhand(shuffleafterhand=True)
         4. In [4]: game.cardplayed((9,1)) # Dealer gets 9

game.cardplayed((12,0)) # User gets Q

game.cardplayed((9,1)) # Dealer gets 9

Output:

Stand Probability = 22.0%

Hit Probability = 44.72%

Split? None

* + - 1. In [6]: game.cardplayed((12,0)) # User gets Q
      2. In [7]: game.getmoves()

game.printprobs()

Output:

Stand Probability = 87.95%

Hit Probability = 33.25%

Split? False

* + - 1. In [8]: game.newhand(shuffleafterhand=True)
      2. In [9]: game.cardplayed((9,1)) # Dealer gets 9

game.cardplayed((14, 0)) # User gets A

game.cardplayed((6,1)) # Dealer gets 6

game.cardplayed((14,0)) # User gets A

* + - 1. In [10]: game.getmoves()

game.printprobs()

Output:

Stand Probability = 80.72%

Hit Probability = 37.68%

Split? False

* + - 1. In [11]: for j in range(2,10):

for i in range(3):

game.cardplayed((j,2)) # extra tests

* + - 1. In[12]: game.getmoves()

game.printprobs()

Output:

Stand Probability = 41.83%

Hit Probability = 61.05%

Split? True

Currently, the detection works via a webcam & the built-in darknet detector demo [2]. This demo runs interference on the input webcam frames, and then draws bounding boxes on the output video feed frames. Prerequisites for the demo:

* Darknet [2] must be compiled.
* A trained weights file must be generated.
* A webcam is preferred but the detection does work on static images.

To run the demo:

1. Plug in the webcam into the host computer.
2. Position the webcam vertically above the cards to be detected and make sure the scene has plenty of light.
3. Navigate to the darknet folder containing the build files.
4. Run the following command from windows PowerShell:

**darknet** detector demo .\data\obj.data .\cfg\yolov4-tiny-obj3.cfg .\backup\yolov4-tiny-obj3\_20000.weights -c 0

**.\data\obj.data** → location of the custom object data file.

**.\cfg\yolov4-tiny-obj3.cfg** → location of the custom object configuration file.

**.\backup\yolov4-tiny-obj3\_20000.weights** → location of the trained WEIGHTS file.

**-c 0** → Use system webcam with ID=0

1. Observe the output in the video feed window with the bounding boxes drawn on. The identified cards will also appear in the command line window.

**Testing Unity Project**

In this section we will be testing all the classes together within Unity without the Object Detection input.

1. Within the Start function of the GameLogic class contained in the Unity project, change the values that are added to the argumentList. Use values that will give you each possible outcome.

Note: The values added after “p” will be added together to get the Player’s hand and every value after “d” will be added together to get the Dealer’s hand.

Settings

* + 1. Surrender - Player hand = 0, Dealer hand = Any
    2. Player Busts - Player hand > 21, Dealer hand = Any
    3. Continue Playing - Player hand = Any, Dealer hand <= 16
    4. Player Wins - Player hand > Dealer hand, Dealer hand != 16 or Dealer hand > 21 and Dealer != 16
    5. Dealer Wins - Dealer hand > Play hand and Dealer hand != 22
    6. Push - Player hand = Dealer hand

1. After the changes have been made and saved in the GameLogic class. Run the Unity project by pressing the play button at the top middle of the screen.

**Setting up HoloLens to use Windows Device Portal**

1. Power on your HoloLens and on the device.
2. Use the Bloom on HoloLens to launch the main menu.
3. Gaze at the Settings tile and do an air-tap gesture on HoloLens.
4. Select the Update menu item.
5. Select the For developers menu item.
6. Enable Developer Mode.
7. Scroll down and enable Device Portal.
8. Set up the Windows Device Portal so you can deploy apps to this HoloLens over Wi-Fi, select Pair to generate a pairing PIN. Leave the Settings app at the PIN popup until you enter the PIN into Visual Studio during your first deployment.

**Connecting over Wi-Fi**

1. Connect your HoloLens to Wi-Fi.
2. Look up your device’s IP address by either:
   1. Going to Settings > Network & Internet > Wi-Fi > Advanced Options.
   2. Going to Settings > Network & Internet and selecting Hardware properties.
   3. Using the “What’s my IP address?” voice command.
3. From a web browser on you PC, go to https://<YOUR\_HOLOLENS\_IP\_ADDRESS>
   1. The browser will display the following message: “There’s a problem with the website’s security certificate” because the certificate, which is issued to the Device Portal, is a test certificate. You can ignore this certificate error for now and continue.

**Installing a certificate**

1. In Windows Device Portal, navigate to the Apps manager page.
2. In the Deploy apps section, select Install Certificate.
3. Under Select certificate file (.cer) used to sign an app package, select Choose File and browse to the certificate associated with the app package that you want to sideload
4. Select Install to start the installation.

**Installing the benchmark/non-remote rendering application**

1. When you’ve created an app package from Visual Studio, you can remotely install it onto your device from the generated files.
2. In Windows Device Portal, navigate to the Apps manager page.
3. In the Deploy apps section, select Local Storage.
4. Under Select the application package, select Choose File and browse to the app package that you want to sideload.
5. Check the respective boxes if you want to install optional or framework packages along with the app installation and select Next.
6. Select Install to start the installation.
7. Once the installation is complete, go back to the All apps page on your HoloLens and launch your newly installed application.

**Instantaneous System Performance Readings**

1. From the Windows Device Portal home page on the Microsoft HoloLens, Select the Views from the menu on the left.
2. Then select Apps.
   1. Ensure that the only app that is currently running is the one being tested.
   2. End any unnecessary processes that are running.
3. From the Windows Device Portal home page on the Microsoft HoloLens, Select the Performance from the menu on the left.
4. To see the instantaneous system performance, select System Performance.
   1. Log the following metrics periodically during the test:
      1. **SoC power:** Instantaneous system-on-chip power usage, averaged over one minute
      2. **System power:** Instantaneous system power usage, averaged over one minute
      3. **Frame rate:** Frames per second, missed VBlanks per second, and consecutive missed VBlanks
      4. **GPU:** GPU engine usage, percent of total available
      5. **CPU:** percent of total available
      6. **Memory:** Total, in use, committed, paged, and non-paged

**Performing a trace**

1. On the left, navigate to Performance > Performance Tracing
2. Choose an available profile or select Custom profiles > Browse then use this profile to analyze CPU performance:

<https://aka.ms/CPUProfileforDevicePortal>

1. Click Start Trace.
2. The HoloLens is now recording a trace. Make sure to trigger the performance issues that you want to investigate, and then select Stop Trace.
3. The trace will be listed at the bottom of the webpage. Select the disk icon at the right-hand side to download the ETL file.

You now have an ETL file that you can either open directly in WPA or send to someone else.