BME Design-Spring 2023 - HAOCHEN WANG
Complete Notebook

PDF Version generated by

Sam Neuman

on

May 03, 2023 @11:08 PM CDT

Table of Contents

 Team contact Information

HAOCHEN WANG - Mar 01, 2023, 1:05 PM CST

Last Name	First Name	Role	E-mail	Phone	Office Room/Building
Block	Wally	Advisor	wfblock@wisc.edu	608/2659686	137 WIMR
Samuel	Poore	Client	poore@surgery.wisc.edu		WIMR Poore Lab
Shaffrey	Ellen	Client	shaffrey@wisc.edu		WIMR Poore Lab
Eftekari	Sahand	Client	eftekari@wisc.edu		
Neuman	Sam	co Leader, co Communicator, co BSAC, co BWIG, co BPAG	ssneuman@wisc.edu		
Wang	Haochen	co Leader, co Communicator, co BSAC, co BWIG, co BPAG	haochen.wang@wisc.edu		

Project description

HAOCHEN WANG - Mar 01, 2023, 1:09 PM CST

Course Number: BME 402

Project Name: An Accessible Virtual Reality Model of Surgical Microscopy for Microsurgical Practice

Short Name: VR Microsurgery

Project description/problem statement:

In this study, we propose a simple model for microsurgical practice that is comparable to commercial surgical micro-scopes while maintaining accessibility. Our design utilizes two Logitech webcams in an array with fixed horizontal disparity and angles of projections. The webcams are connected to a laptop computer via hardwire connection. Within Unity software, basic manipulations are made to position the two images in 3D virtual space before the images are streamed through a local server to an Oculus VR headset. The user is able to visualize the surgical specimen on the field with optimal ergonomics through a stable local connection with minimal lag.

About the client:

Poore lab focuses on developing novel strategies for peripheral nerve regeneration

Previous Lab Notebook Archive

HAOCHEN WANG - Mar 01, 2023, 1:16 PM CST

Title: Previous Lab Notebook Archive

Goals: To provide archive for the lab notebooks from previous semesters

Content:

See attachment below

HAOCHEN WANG - Mar 01, 2023, 1:11 PM CST

[Download](#)

BME_400-Lab_Notebook_Archive.pdf (9.23 MB)

2023/01/07 Team Meeting

HAOCHEN WANG - May 03, 2023, 9:40 PM CDT

Title: Team Meeting

Date: 01/07/2023

Content by: Haochen

Present: Sam and Haochen

Goals: Meet to establish goals for the semester

Content:

Team Meeting
Goals for the semester
1. Find way to access hardware address in Unity
2. Make android app or iPhone app so camera can be recognized as hardware
3. Find a better fabrication method for the housing and the mirrors

2023/01/18 Client Meeting 1

Title: Client Meeting

Date: 01/18/2023

Content by: Haochen

Present: Sam and Haochen

Goals: Meet with the client to establish the goal for the semester

Content:

1. Have Ellen and Sahand test on the prototype for the first stage
2. Not need IRBs
3. Set up a meeting for onsite testing for next week (Wednesday at 4 pm?)
4. Ideally have testing data a month before the end of the semester, 2 weeks before semester for presentation (have data ready on week 8)
5. Need a week for the final report
6. **Look for current ways for microsurgery training to be done (intro)**
7. A lot for method section

2023/01/18 Client Meeting 2

HAOCHEN WANG - May 03, 2023, 9:42 PM CDT

Title: Client Meeting

Date: 01/25/2023

Content by: Haochen

Present: Sam and Haochen

Goals: Meet with the client to receive feedback for the most recent build of Unity project

Content:

1. Need to have immersive build, bring the canvas closer
2. Manipulate the canvas with camera (zoom in and zoom out)
3. Make XR the parent of canvas, so that canvas moves along the head

2023/02/08 Client Meeting 3

Sam Neuman - Feb 27, 2023, 11:18 PM CST

Title: Client Meeting 1

Date: 02/08/2023 (02/27/2023)

Content by: Sam

Present: Sam and Haochen

Goals: Meet with the client to understand current errors in the program.

Content:

The following was our finalized code after this meeting:

```csharp
using System.Collections;

using System.Collections.Generic;

using TMPro;

using UnityEngine;

using UnityEngine.UI;


public class CameraScript : MonoBehaviour

{

    // Start is called before the first frame update

    public static int currentCamIndex = 1;

    WebCamTexture tex;

    public RawImage display;

    public TextMeshProUGUI startStopText;

    int cameraRequestedWidthRes = 1920;

    int cameraRequestedHeightRes = 1080;
```

```csharp
public void SwapCam_Clicked(){


    if (WebCamTexture.devices.Length > 0){


        currentCamIndex += 1;


        currentCamIndex %= WebCamTexture.devices.Length;


        if(tex != null)


        {


        StopWebcam();


        StartStopCamClicked();


        }


    }


}


public void StartStopCamClicked(){


    if (tex != null){


        StopWebcam();


        startStopText.text = "Start Camera";


    }


    else{
```

```csharp
            WebCamDevice device = WebCamTexture.devices[currentCamIndex];

            tex = new WebCamTexture(device.name, cameraRequestedWidthRes, cameraRequestedHeightRes);

            display.texture = tex;


            tex.Play();

            startStopText.text = "Stop Camera";

        }


}


public void StopWebcam()


{

        display.texture = null;

        tex.Stop();

        tex = null;


}



void OnEnable()


{

    StartStopCamClicked();

    currentCamIndex += 1;

    currentCamIndex %= WebCamTexture.devices.Length;
```

```
        }



}
```

Conclusions/action items:

This code relies on the tex assignment as null to operate. The public RawImageDisplay is a possible package library that may supercede old installments of Netcode and websocket client. This still throws an error that has to do with the recognition of the webcam images. Sahand also tested alignment of the two images in the canvas space.

2023/04/21 Client Meeting 4

HAOCHEN WANG - May 03, 2023, 9:36 PM CDT

Title: Client Meeting

Date: 04/21/2023

Content by: Haochen

Present: Sam and Haochen

Goals: Meet with the client to receive feedback for our final design

Content:

Issues:

1. Distortion from Logitech Webcam (from field of view 90 deg)
   a. Potential solution with software
2. 30 fps - noticeable lag and sometimes dragging down to 15 fps due to limits from raspberry
   a. But expensive to raise the fps up
3. Lighting condition
4. Biggest limitation: autofocus (need manual focus method)

Advantages:

1. Price point
2. Ability to have augmented reality (stereoscope)
3. Portable
4. High adjustable and customizable
5. Latency less than 100 ms

Prototypes are at a preliminary stage, these feedbacks will be helpful for bringing the design to the next stage.

Expenses:

Webcam C920 - $100 each
3D printed - $53 from maker space
Raspberry 4b (for 1080 p streaming) - $ 130
Oculus - $320 (including tax)
iPhone stand - $30 (clamps the chasey)

2023/03/03 Advisor Meeting 1

Sam Neuman - Mar 03, 2023, 12:13 PM CST

Title: Advisor Meeting

Date: 03/03/2023

Content by: Sam

Present: Haochen, Sam, Dr. Block

Goals: Complete the weekly advisor meeting.

Content:

Progress update:

Rolled Unity back to a previous version to help circumvent the updated Netcode package.

Need to download a set of libraries in Unity that may be out of date.

When tested on Oculus, the protocol is no longer accepted, so we are trying to restore it with an older version.

Dr. Block wishes he could visit New Orleans.

Conclusions/action items:

Our current design challenge is finding a solution to the lack of support from Unity to play on Oculus. We tried rolling back to a previous version to see if our desired packages are supported. Dr. Block recommended debugging over a large period of time during Spring Break.

2023/03/24 Advisor Meeting 2

Title: Advisor Meeting

Date: 03/24/2023

Content by: Haochen

Present: Haochen, Sam, Dr. Block

Goals: Record feedback for preliminary presentation

Content:

1. What microsurgery means - see things what cannot be seen with naked eyes
2. Why need stereo
3. Describe that the project will keep going

03/01/2023 Current Unity Builds

HAOCHEN WANG - Mar 01, 2023, 5:17 PM CST

Title: Current Unity Builds (as of 03/01)

Date: 03/01/2023

Content by: Haochen

Present: Sam, Haochen

Goals: To set up communication between Raspberry Pi and Oculus

Content:

Attachments are zip files of the current state of three notable build versions.

- Version 5
    - The version with first successful canvas demonstration on Oculus
    - Now merged with codes to project each webcam to a dedicated RawImage object on canvas
    - Currently unable to grant Oculus the access to hardware connected to the laptop
- Sahand
    - Sahand's build that was previously successful in wirelessly connecting Raspberry Pi to Oculus via WebSocket
    - Merged partially to Version 5
    - Compatibility issues emerged with WebSocket functions, exploring new packages to resolve the issue
- Version 7
    - Basic build with Unity Render Streaming demos
    - Is working for transmission between laptop browser and Unity, but is apparently not stable
    - This is also an experimental package, which may cause more issues in the future
    - Has built-in server to mediate communication between the sender and the receiver
    - Investigating the code to better understand wireless streaming

Conclusions/action items:

Continue to work on the builds, but mainly focus on either using version 5 for wired connection or version 7 for wireless connection.

HAOCHEN WANG - Mar 01, 2023, 6:02 PM CST

Download

WorkStation-Ver7.zip (1.07 GB LA S3)

HAOCHEN WANG - Mar 01, 2023, 6:03 PM CST

[Download](#)

dario_20baseline_20webcam_20project_20_2_.zip (631 MB LA S3)

HAOCHEN WANG - Mar 01, 2023, 6:08 PM CST

[Download](#)

WorkStation-Ver5.zip (2.25 GB LA S3)

2023/05/03 Solution to project different images to each eye

HAOCHEN WANG - May 03, 2023, 7:27 PM CDT

Title: Current Unity Builds (as of 05/03)

Date: 05/03/2023

Content by: Haochen

Present: Sam, Haochen

Goals: To project different images to each eye

Content:

Changes are made to the last iteration of the Unity Project

1. Set up two canvases for the two raw images
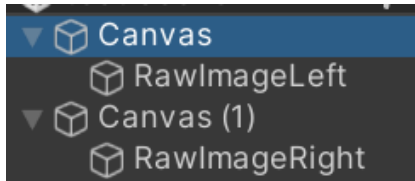
2. Under XRRig - Main camera, create two camera objects
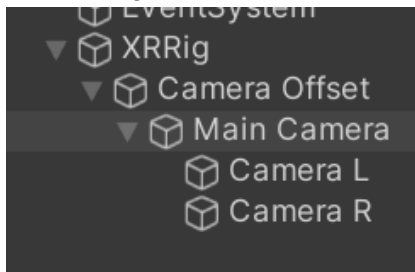
3. For each camera, set the target eye to the designated eye
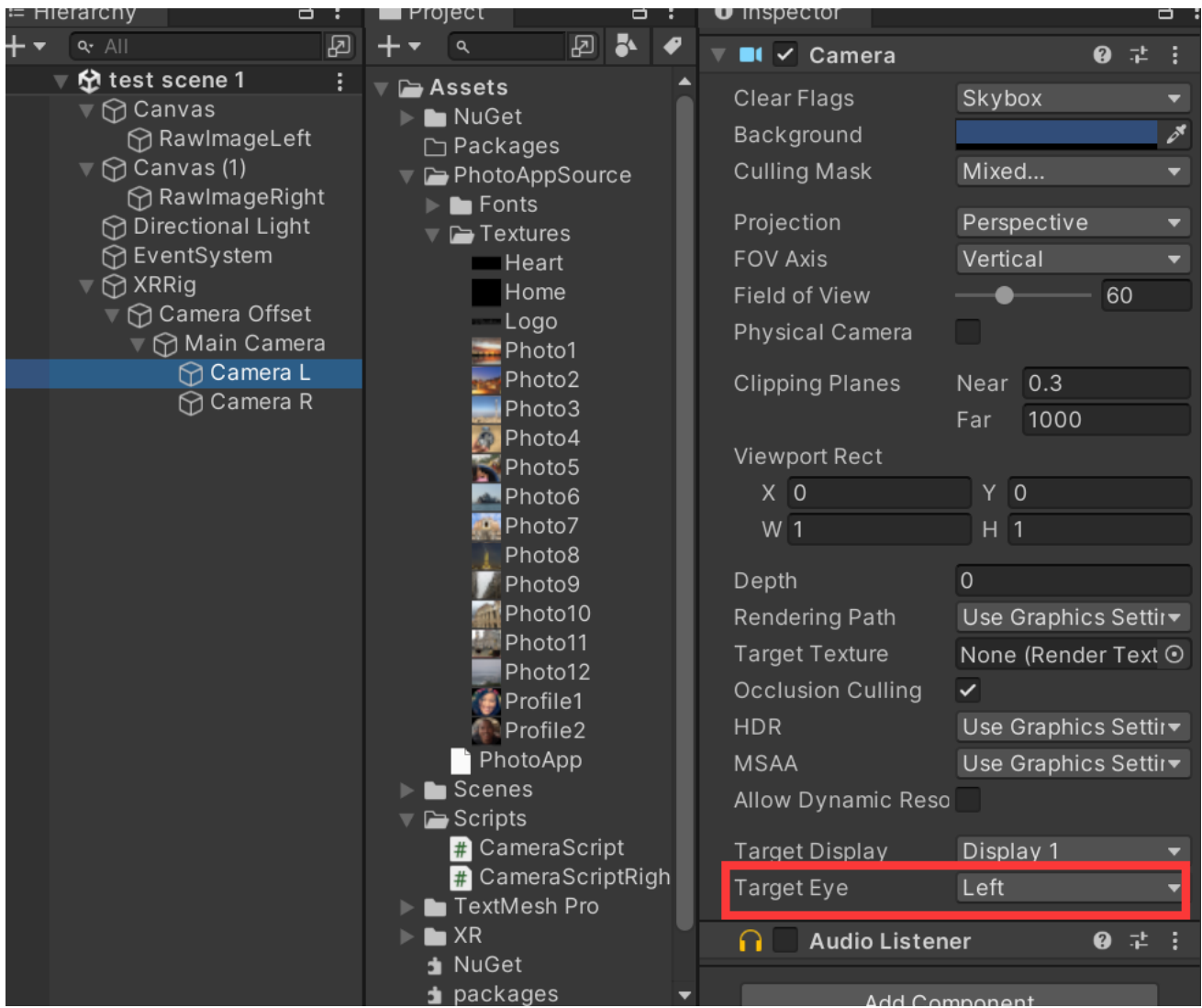
4. For each canvas, set render mode to "screen space - camera" and render camera to each sub-camera
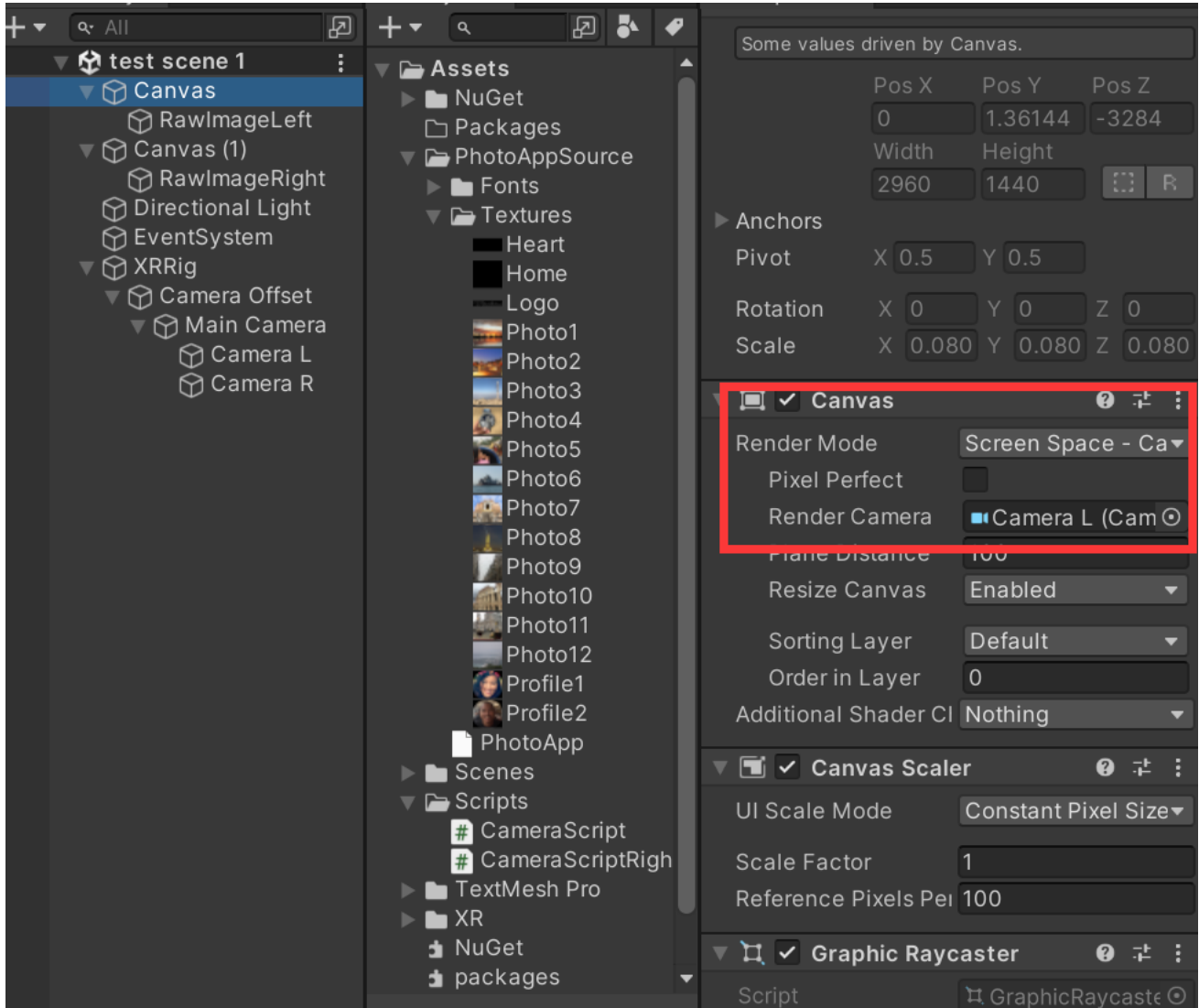
5. For each raw image, create/set a layer of its own
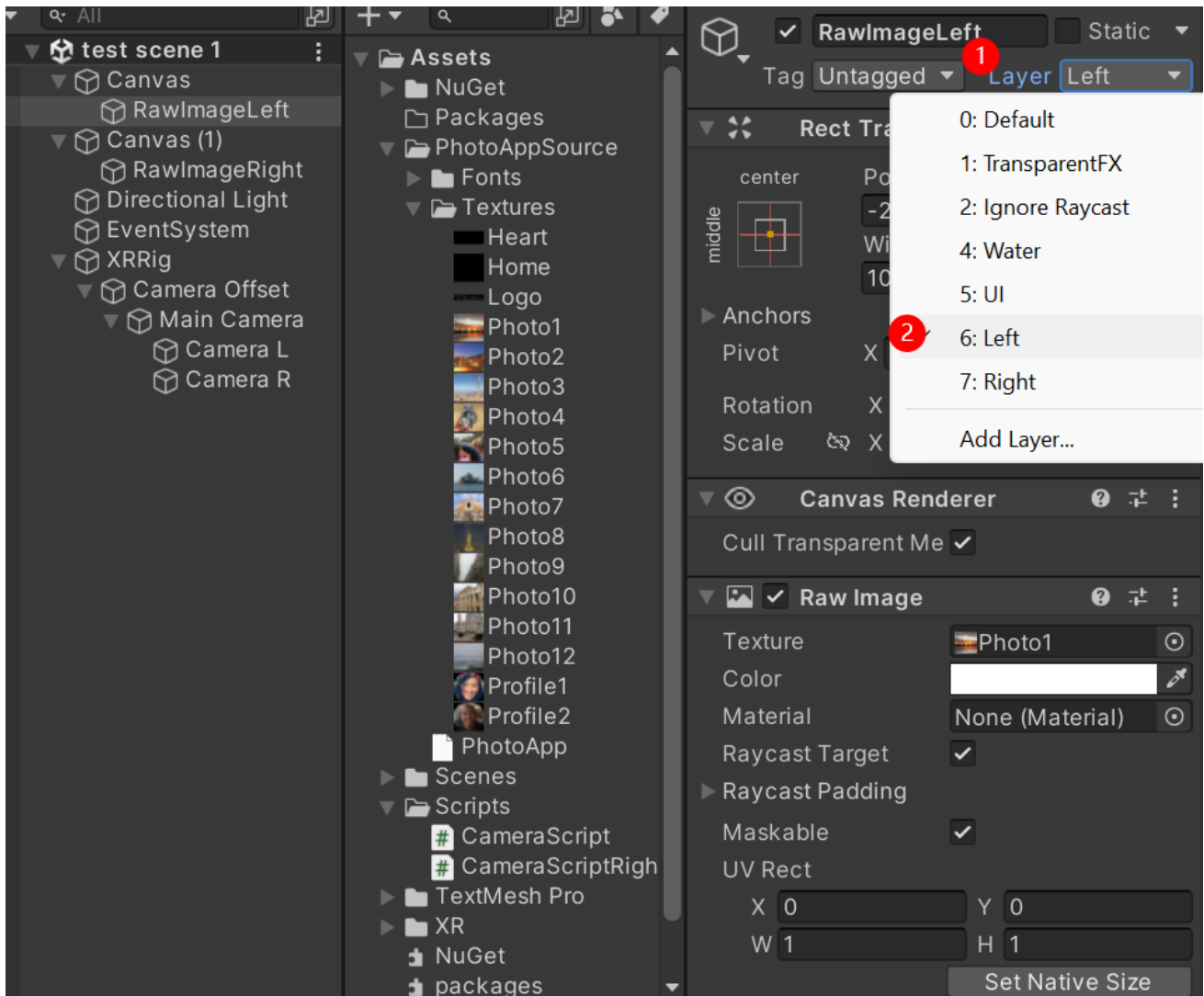
6. For each sub-camera, uncheck the layer from the opposite side
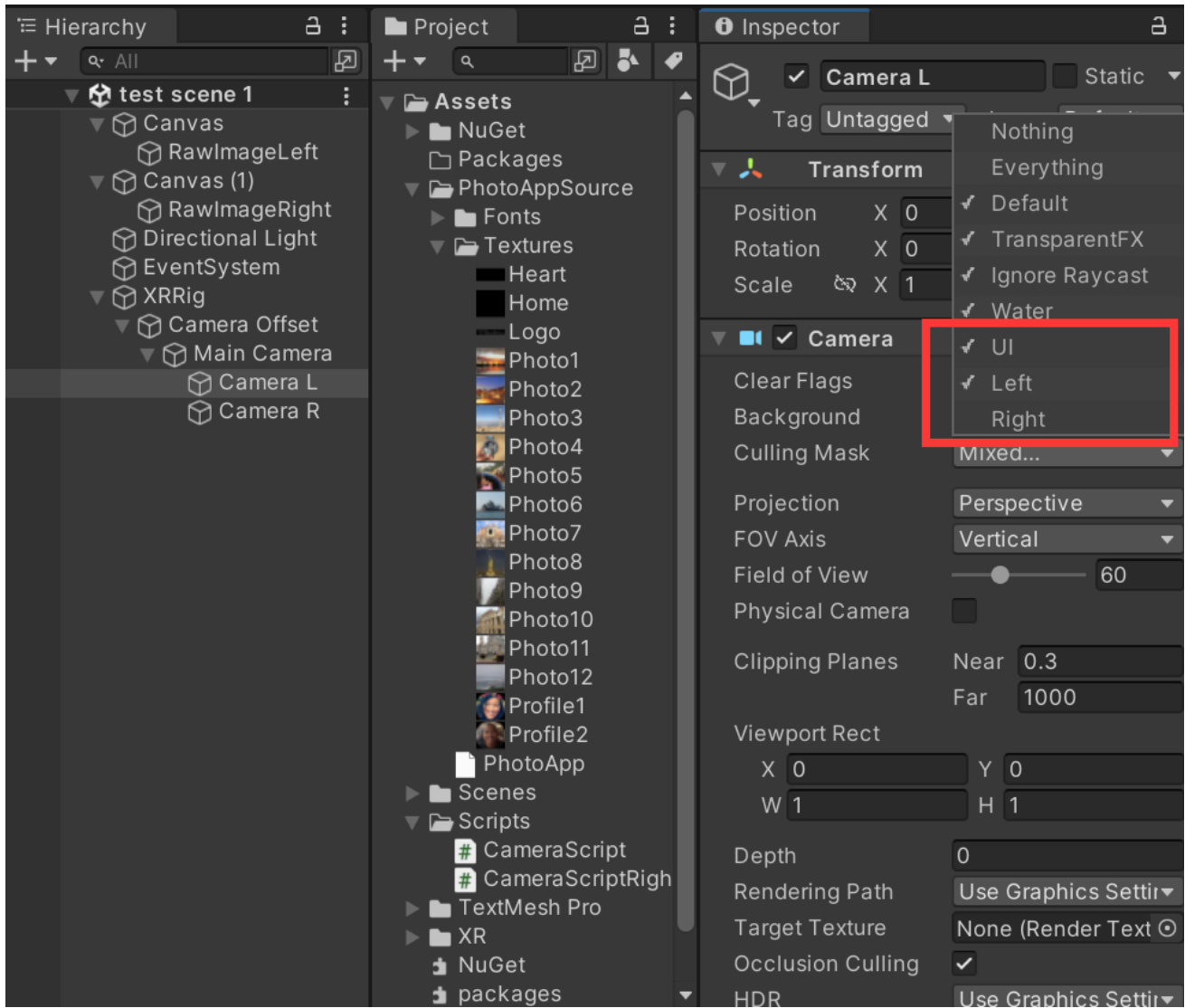
7. Then from the top menu, edit - project settings - XR plug-in management - oculus, set stereo rendering mode to Multi pass
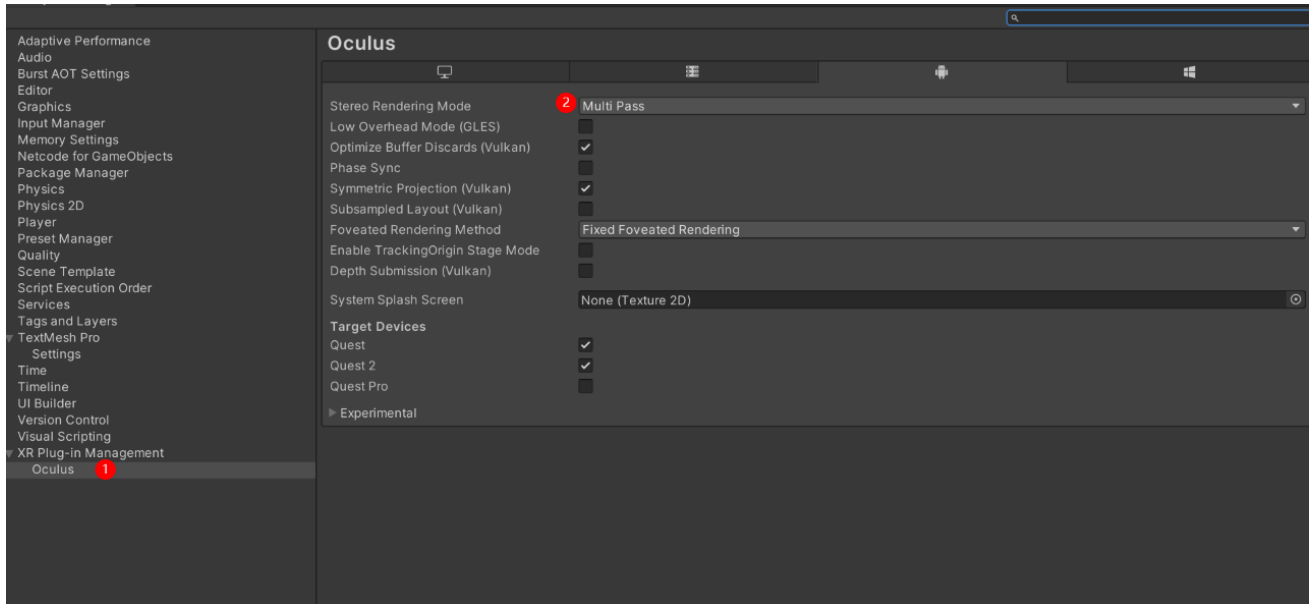
8. Now each eye piece on oculus should receive an image from only one of the two canvases, but just be aware that the recording function on oculus can only record image displayed to one of the eye pieces at a time

2023/05/03 Final Materials and Expenses

HAOCHEN WANG - May 03, 2023, 7:24 PM CDT

All items are provided by the clients:

Table 1. Itemized expenses associated with design construction

Item	Quantity	Unit Price ($USD)
Logitech C920 Webcams	2	100
3D Printed Chassis	1	53
Oculus Quest 2	1	320
Raspberry Pi 4b	1	130
Phone Boom Arm	1	30
Total		733

2023/05/03 Future Testing Plans

Sam Neuman - May 03, 2023, 10:40 PM CDT

Title: Plans for future testing

Date: 3 May 2023

Content by: Sam

Present: Sam and Haochen

Goals: Record some potential avenues for testing the device.

Content:

We have two things that we want to prove in testing this device. The first goal is to compare the use of this device with the use of a commercial surgical microscope and unassisted surgery. We hope to recruit trained microsurgeons and randomly assign them to perform each of the three trials in random order and also use the SMaRT scale to evaluate their user experience. The second goal is the evaluate the potential of this device as a training modality. We hope to recruit untrained medical students and have them do a standard 40-hour course on either the VR system or a commercial microscope and then perform microsurgery procedures on a chicken breast.
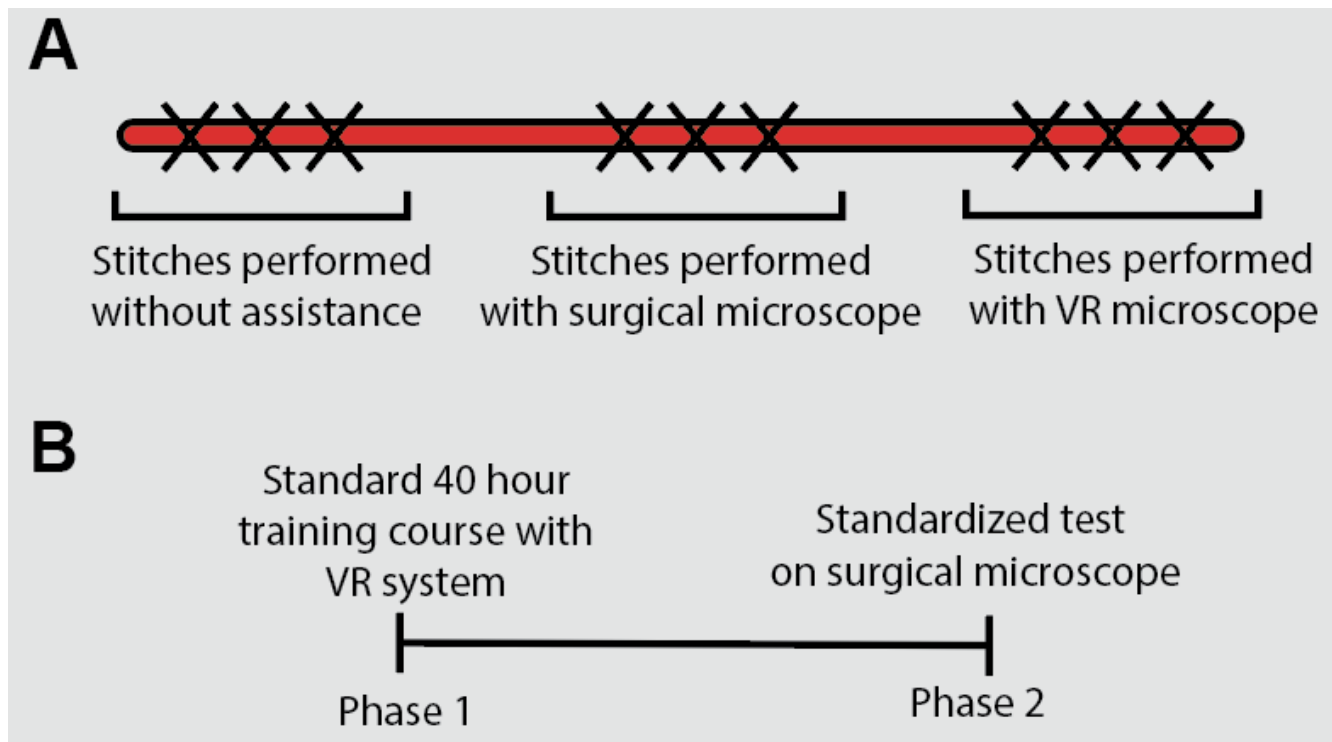
Fig. 5. Testing of the design concept will occur in two rounds. A. Methods intended to evaluate the potential for the VR system as a replacement for commercial surgical microscopes. The VR system will be compared against surgical microscope-assisted and unassisted suturing by the performance of three sutures on a practice suture pad by groups of trained attending plastic surgeons. The order in which each trial is completed will be randomized. B. Methods intended to evaluate the VR system as a training tool for untrained medical students. Medical student participants will undergo a standardized 40 hour training course on either the VR system or a commercial microscope and will then be asked to perform a standardized test on a surgical microscope. This test will involve anastomosis procedures on a preserved chicken breast.

Conclusions/action items:

We have two goals for our proposed future testing plans. The first is the evaluate how the VR system compares to a commercial microscope and unassisted surgery. The second is to evaluate the potential of the VR system as a potential training modality. Through testing, we also hope to

have participants rate our device using the SMaRT scale to guage how the user experience is for them.

03/01/23 Iterations of Unity Project by 03/01

Title: Iterations of Unity Project by 03/01

Date: 03/01/23

Content by: Haochen Wang

Present: N/A

Goals: To fix the compatibility issues with current Unity project build

Content:

1. OnEnable()
    1. Used in all build versions
    2. Add OnEnable function to automatically start the camera at run
2. Full screen when launch and head tracking canvas
    1. Used in all build versions
    2.
    3.
3. Project Unity onto Oculus via Quest Link
    1. Applied to all projects
    2. Cannot recognize the device using wired connection, even with quest link official cable
    3. Wireless connection through mobile hotspot had successful device recognition, but the desktop client app requires dedicated graphic card as advanced as NVIDIA 20 and 30 series, which defeats the purpose of this project as using affordable and portable devices for training.
    4. Will serve as a backup plan
4. Wireless casting via hotspot on Raspberry Pi 4
    1. Ultilizes scrcpy package
    2. Original post on reddit: https://www.reddit.com/r/OculusQuest/comments/ege6qx/the_ocupi_wireless_nas_and_casting_using_a/
    3. Code archived for future review in the next post
    4. Was not tested due to lack of access to client's Raspberry Pi
5. Direct wired connection between webcam and Oculus
    1. Requires adapter (about $ 20)
    2. The author did not mention how the system was set up, but we could ask if this is the approach to take
    3. Original video on YouTube: https://youtu.be/qpTlmogYRnc
    4. Note that the author is calling for action to replace hardware connection with software design, indicating that the solution to our own project will be also applicable for other purposes that lack a valid solution
6. websocket-sharp
    1. Applied to build version 5
    2. Complied to run with C# for the WebSocket protocol
    3. Still under testing, the API appears not supporting video transmission, will dig into that
    4. Official page: http://sta.github.io/websocket-sharp/ (installed via NuGet)
7. Unity Render Streaming
    1. Native package that allows streaming Unity working environment to a browser within the network
    2. Since previous work showed success to capture via webcam and project to Unity desktop client, this will work fine
    3. Tested with templates, found no issue for projecting to browser on the laptop
    4. Browser in oculus cannot visit the streaming server, appears to be issues with my internet settings
    5. This is promising and seems like a backup plan
    6. Official page: https://docs.unity3d.com/Packages/com.unity.renderstreaming@3.0/manual/index.html

8.

Conclusions/action items:

 Continue with Unity Render Streaming and WebSocket protocols

HAOCHEN WANG - Mar 01, 2023, 5:09 PM CST

Original post on reddit:

Posted by u/[deleted] 3 years ago

The OcuPi: wireless NAS and casting using a Raspberry Pi 4 for the Quest

Discussion
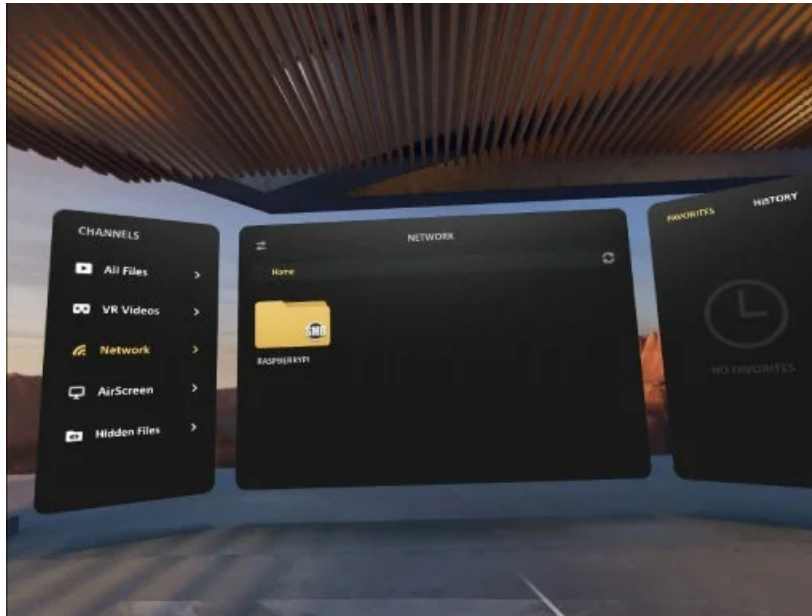
I created a 5GHz hotspot and a wireless NAS for my Quest using a Raspberry Pi 4 (4GB model), and a few people have asked how I did it, so here it is. I'm using it with a usb wifi adapter
, but you can use the onboard wifi chip if you like. Just change a few things in the steps below. I'm also able to cast to the Pi, using scrcpy , but it's only video at the moment...I bought a $20 sound card for the Pi, and plan on connecting a bluetooth audio receiver to the line-in of the sound card, then connect a bluetooth audio transmitter to the Quest, then stream the audio over bluetooth. I'm still putting the unit together, so I'll post that up when I'm done.

But in the meantime, this will give you a 5GHz hotspot and wireless NAS for the Quest.

Parts needed:

Raspberry Pi 4 4GB

Raspberry Pi 4 usb-c power supply

Heat sinks

4TB WD MyPassport drive

16GB microSD

You can also buy a Raspberry Pi 4 4GB starter kit, that comes with a case for the Pi. Otherwise, search for a Raspberry Pi 4 case, or mount in a Tupperware container or something. I'm using a gutted out atx power supply box
    to house it all up. I call it The OcuPi. :)

Once you have all the parts, then download the Raspbian operating system, and follow below:

----

  1. Grab the latest Raspbian image:

https://downloads.raspberrypi.org/raspbian_full_latest

2) Write the image to a microSD card, 8GB minimum.

https://sourceforge.net/projects/win32diskimager/

3) Copy these instructions to a text file, then copy to the microSD so you can transfer to the Pi's desktop once booted, so you can then copy and paste the instructions into the terminal.

4) Look for the config.txt on the microSD, and add the following to the bottom of the file:

max_usb_current=1

#over_voltage=4

#arm_freq=1750

#gpu_freq=600

dtoverlay=pi3-disable-wifi

dtoverlay=pi3-disable-bt

disable_overscan=1

* If you have heatsinks and a fan, then uncomment over_voltage, arm_freq, and gpu_freq to overclock from 1.5 GHz to 1.75 GHz. Also, I disabled the onboard wifi and bluetooth, as I'm using an external wifi adapter with a higher data rate than the onboard chip, and I don't need the bluetooth, as I'm using a bluetooth audio receiver for the casting. If you prefer to use the onboard wifi and bluetooth for a keyboard or something, then just comment them out in the config, or delete them altogether.

5) Once you've saved the file, safely eject the microSD card, insert into the Pi, connect your keyboard/mouse/dongle, connect your monitor, connect your ethernet cable, and power up the Pi. It can take up to a minute for the first boot, as it resizes the system partition to use the entire microSD, then you're greeted at the desktop with an installer GUI. Set your preferences, then let it do it's update. It may give a time out error or some other repository issues, but we're going to manually update everything anyway, so no worries.

6) Once the update's finished, choose to reboot. Then back at the desktop, plug your usb stick in, open it up, and copy the readme to the desktop. Then click the eject icon on the right side of the taskbar, and eject the usb stick.

Then click the black terminal icon on the left side of the taskbar, and c&p the following commands...everything 'after' the $ sign:

7) disable screen saver:

$ sudo nano /etc/xdg/lxsession/LXDE-pi/autostart

* Edit to look like below:

u/lxpanel --profile LXDE-pi

u/pcmanfm --desktop --profile LXDE-pi

u/xscreensaver -no-splash

#Disable screensaver:

u/xset s noblank

u/xset s off

u/xset -dpms

point-rpi

* Then save: [CTRL]+[O], [Enter] then [CTRL]+[X], [Enter]

$ sudo reboot

8) Update system if needed:

$ sudo apt update

$ sudo apt upgrade

$ sudo apt dist-upgrade

$ sudo apt autoremove

$ sudo apt autoclean

9) set cpu to 'performance' mode:

$ sudo apt install cpufrequtils

$ echo 'GOVERNOR="performance"' | sudo tee /etc/default/cpufrequtils

$ sudo systemctl disable raspi-config

$ cd /sys/devices/system/cpu

* for the next 4 steps, c/p the first line, press enter, then use the page up key to recall the command you just entered, then use the page left key to edit the 'cpu0/' to 'cpu1/', then 'cpu2/', then finally 'cpu3/'. It's quicker than c/p'ing all 4 lines, but just make sure to press enter after each.

$ sudo sh -c "echo performance > cpu0/cpufreq/scaling_governor"

$ sudo sh -c "echo performance > cpu1/cpufreq/scaling_governor"

$ sudo sh -c "echo performance > cpu2/cpufreq/scaling_governor"

$ sudo sh -c "echo performance > cpu3/cpufreq/scaling_governor"

* Then check to see the changes:

$ cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor

10) Disable the swapfile, as it slows down the microSD, and is pointless when the RPi4 has 4GBs of RAM now, instead of just 1GB the RPi3B+ had.

```
$ cd
```

```
$ sudo swapoff -a
```

```
$ sudo update-rc.d -f dphys-swapfile remove
```

```
$ sudo rm /var/swap
```

```
$ sudo apt purge dphys-swapfile dphys-config
```

```
$ sudo nano /etc/fstab
```

* If /swapfile'....' is still in the list, then comment out the line:

```
#/swapfile
```

* Then save: [CTRL]+[O], [Enter] then [CTRL]+[X], [Enter]

```
$ sudo reboot
```

11) Then open termianl again, and complete the following:

* Check to see swapfile was removed:

```
$ free
```

* It should say:

```
Swap: 0 0 0
```

* Then check cpu GOVERNOR set to 'performace' stuck:

```
$ cpufreq-info
```

* Check to see all cores are set to:

current policy: frequency should be within 600 MHz and 1.50 GHz.

The governor "performance" may decide which speed to use within this range.

current CPU frequency is 1.50 GHz.

* If so, then connect wifi adapter and continue. If choosing to use the onboard wifi, skip step 12, and go straight to 13.

12) Install wifi drivers:

$ wget
https://gist.githubusercontent.com/kmonsoor/59c47d05d8c6cb397c15cd5eba7909e8/raw/7e04bd3dd4d76eb21eca9f602247d757b89648da/RPi-install-wifi.sh

$ sudo mv RPi-install-wifi.sh
    /usr/bin/install-wifi

$ sudo chmod +x /usr/bin/install-wifi

$ sudo install-wifi

13) Create hotspot:

$ wget -q https://git.io/voEUQ
    -O /tmp/raspap && bash /tmp/raspap

* Answer [Y] to everything, don't just hit [enter], as the very first option defaults to [N].

* After reboot, open browser:

IP address: 10.3.141.1

Username: admin

Password: secret

* Select Configure hotspot

- Wireless Mode: 802.11a - 5 GHz

- Channel: 13

- Security

- Security Type: WPA+WPA2

- Encryption Type: TKIP

- PSK: whatever you like

- Advanced

- WiFi client AP mode: Enabled

- Maximum number of clients: 10 (defaults to 2007 clients)

- Country Code: your country

* Save settings

$ sudo reboot

$ sudo nano /etc/dhcpcd.conf

interface wlan0

static ip_address=192.168.4.1/24

nohook wpa_supplicant

* Then save: [CTRL]+[O], [Enter] then [CTRL]+[X], [Enter]

$ sudo nano /etc/dnsmasq.conf

interface=wlan0 # Use the require wireless interface - usually wlan0

dhcp-range=192.168.4.2
    ,192.168.4.20   ,255.255.255.0   ,24h

* Then save: [CTRL]+[O], [Enter] then [CTRL]+[X], [Enter]

```
$ sudo nano /etc/hostapd/hostapd.conf


driver=nl80211

ctrl_interface=/var/run/hostapd

ctrl_interface_group=0

ssid=OcuPi

hw_mode=a

wmm_enabled=0

macaddr_acl=0

interface=wlan0

wpa=2

auth_algs=1

wpa_key_mgmt=WPA-PSK

wpa_passphrase=passphrase

wpa_pairwise=TKIP

rsn_pairwise=CCMP

country_code=CA

ignore_broadcast_ssid=0


# N

ieee80211n=1

require_ht=1

ht_capab=[MAX-AMSDU-3839][HT40+][SHORT-GI-20][SHORT-GI-40][DSSS_CCK-40]


# AC

ieee80211ac=1

require_vht=1

ieee80211d=1

#ieee80211h=0

ieee80211h=1

#vht_capab=[MAX-AMSDU-3839][SHORT-GI-80]
```

```
vht_capab=[MAX-AMSDU-11454][MAX-A-MPDU-LEN-EXP7][SHORT-GI-80]

vht_oper_chwidth=1

channel=44

vht_oper_centr_freq_seg0_idx=50
```

* Then save: [CTRL]+[O], [Enter] then [CTRL]+[X], [Enter]

$ sudo iw reg set YourCountryCode

* ex: sudo iw reg set CA

11) Configure NAS storage:

* Find the idVendor and idProduct of the external drive:

$ sudo dmesg -C

* Attach external drive

$ dmesg

* [422.849481] usb 2-2.1: New USB device found, idVendor=1058, idProduct=25e2, bcdDevice=40.05

$ sudo nano /boot/cmdline.txt

* Add usb-storage.quirks=idVendor:idProduct:u to beginning of the line, with a space between.

* Example:

```
usb-storage.quirks=1058:25e2:u dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1
root=PARTUUID=5c0b60a3-02 rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait quiet splash
plymouth.ignore-serial-consoles
```

* Then save: [CTRL]+[O], [Enter] then [CTRL]+[X], [Enter]

```
$ sudo reboot
```

```
$ sudo apt install gparted ntfs-3g samba synaptic
```

```
$ sudo mkdir /home/shares
```

```
$ sudo mkdir /home/shares/public
```

```
$ sudo chown -R root:users /home/shares/public
```

```
$ sudo chmod -R ug=rwx,o=rx /home/shares/public
```

```
$ sudo nano /etc/samba/smb.conf
```

* Scroll down to '##### Authentication #####' and add below: (use [CTRL]+[W], enter search string, hit [enter])

```
security = user
```

* Now scroll/search to [homes] and add:

```
read only = no
```

* Then scroll to very bottom and add:

```
[public]
comment = public storage
path = /home/shares/public
valid users = u/users
force group = users
create mask = 0660
directory mask = 0771
read only = no
```

* Then save: [CTRL]+[O], [Enter] then [CTRL]+[X], [Enter]

```
$ sudo /etc/init.d/smbd restart
```

```
$ sudo smbpasswd -a pi
```

* Enter new password and verify

$ sudo blkid

* Look for your external storage, ie: /dev/sda1

* Copy the UUID='number'

$ sudo umount /dev/sda1

$ sudo mkdir /home/shares/public/disk1

$ sudo chmod -R ug=rwx,o=rx /home/shares/public/disk1

$ sudo mount /dev/sda1 /home/shares/public/disk1

$ sudo nano /etc/fstab

* Add the following:

UUID=Your-UUID-Number /home/shares/public/disk1 auto defaults,user,nofail 0 2

* ex: UUID=D65099E95099D09B /home/shares/public/disk1 auto defaults,user,nofail 0 2

* Then save: [CTRL]+[O], [Enter] then [CTRL]+[X], [Enter]

$ sudo reboot

* Upon reboot, open home folder..that should put you in /home/pi. Select home from left panel, or click the up arrow to the left of the address bar, then open the shares folder. Check to see all your videos appear. Full directory is /home/shares/public/disk1.

* To link contents from multiple directories into 1 shared directory, do the following for each directory:

$ ln -s /BASE/* /TARGET/

sudo ln -s /home/shares/public/disk1/BASE/* /home/shares/public/disk1/TARGET/

12) Digital clock settings: (right-click clock)

%a %d %b %y %I:%M %p

13) Install SSH and VNC:

$ sudo raspi-config

* scroll down to '5 Interfacing Options'

* select P2 SSH

* scroll down to '5 Interfacing Options'

* select P3 VNC

* select finish

$ sudo reboot

14) Install adb and scrcpy for Quest casting:

$ sudo apt install ffmpeg libsdl2-2.0-0 make gcc git pkg-config meson ninja-build libavcodec-dev libavformat-dev libavutil-dev libsdl2-dev adb

$ wget https://github.com/Genymobile/scrcpy/releases/download/v1.9/scrcpy-server-v1.9.jar

$ git clone https://github.com/Genymobile/scrcpy

$ cd scrcpy

$ meson x --buildtype release --strip -Db_lto=true -Dprebuilt_server=/home/pi/scrcpy-server-v1.9.jar

$ cd x

$ ninja

```
$ sudo ninja install
```

* do a final cleanup then reboot

```
$ sudo apt autoremove
```

```
$ sudo apt autoclean
```

```
$ sudo reboot
```

* casting:

```
$ sudo dmesg -C
```

* connect Quest to usb port

```
$ dmesg
```

* copy the idVendor and idProduct IDs:

```
[ 98.212841] usb 1-1.2.2: new high-speed USB device number 5 using xhci_hcd
[ 98.355756] usb 1-1.2.2: New USB device found, idVendor=2833, idProduct=0186, bcdDevice= 4.04
[ 98.355761] usb 1-1.2.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 98.355765] usb 1-1.2.2: Product: Quest
[ 98.355769] usb 1-1.2.2: Manufacturer: Oculus
[ 98.355772] usb 1-1.2.2: SerialNumber: 1234567890ABCD
```

```
$ sudo nano /etc/udev/rules.d/51-android.rules
```

```
SUBSYSTEMS=="usb", ATTR{idVendor}=="2833", ATTR{idProduct}=="0186", MODE="0666", GROUP="plugdev", OWNER="
<pi>"
```

```
$ sudo udevadm control --reload-rules
```

```
$ sudo reboot
```

```
$ adb devices

List of devices attached

* daemon not running; starting now at tcp:5037

* daemon started successfully

1234567890ABCD unauthorized


* accept connection in Quest


$ adb devices

List of devices attached

1234567890ABCD device


* copy IP at end of string:


$ adb shell ip route

10.0.0.0/24
    dev wlan0 proto kernel scope link src 10.0.0.4



* disconnect Quest for wireless mode


$ adb connect 10.0.0.4:5555



connected to 10.0.0.4:5555




$ scrcpy -f -c 1000:1000:110:220 -m 720 -b 15M
```

That's it. It takes a few steps, but it goes by fairly quickly. Once the hotspot's configured, you connect an ethernet cable, then connect the Quest to the SSID (OcuPi). Then open SkyboxVR, and select Network, then select Raspberry Pi. Then shares, then public, then whatever you called your videos folder. After that, Skybox scans the videos and creates

thumbnails. Unfortunately it doesn't keep the thumbnails, and will rescan and recreate the thumbnails every time you start it up again. It's fairly quick, so it's not that bad. It would be better if it kept the images in cache or something though, like the pc version does.

To make this portable, I'm using an Anker PowerCore 10000 PD battery pack, along with a 6" USB-C charging cable
    ...the 3ft cable the Anker came with results in a low power indicator popping up on the Pi, meaning it wasn't getting the full 3A supply. But the shorter cable works fine.

I've also set mine up to use a 120GB SSD for the OS, but it's a few more steps involved, required a powered
USB to SATA adapter
    , and a powered USB3.0 hub   , which means 2 power supplies in total. It can still be run off the battery pack though, using the battery's 2.4A USB-A port. I'll post those steps as well if anybody wants them.

Enjoy. :)

edit: the powered hub is for the WD MyPassport, not the SSD. It's not exactly necessary, it just means the Pi gets it's full power for itself and any dongles. The Pi, dongles and WD MyPassport will run fine on the single Raspberry Pi power adapter. But with the extra wifi adapter, and bluetooth audio receiver, I figured it best to power the drive separately. Totally up to you, and depends on your setup.

edit #2: I forgot to mention about VNC...if you want to control/access the Pi from a tablet or smartphone, then install
VNC Viewer
    from the Google Play store to your device. The VNC we enabled in Raspbian is the same app. You can also sideload the app to the Quest, then control the Pi directly from the Quest if you need to, or want to edit a photo or something before using the browser to upload somewhere. But mostly, it's the best way to turn the Pi off properly through the start menu, rather than just unplugging it, risking data corruption.

Also, if you just want the NAS and not the casting, then a Raspberry Pi 3B+ or earlier will also do the trick. They just don't have enough RAM for casting. I should also mention that for casting, you connect an HDMI cable from the Pi to the TV, or PC monitor, then selecting the input.

Cheers.

2023/01/31 Competing Device: Pocket Suture

Sam Neuman - Feb 27, 2023, 10:43 PM CST

Title: Competing Device: Pocket Suture

Date: 01/31/2023 (02/27/2023)

Content by: Sam

Present: Sam

Goals: Document a potential competing design

Content:

This design is not currently mass-market but appears to be driven by a startup company. The design is basically a stand for a smartphone that allows the phone to be angled downward so that the surgeon can practice suturing. There doesn't appear to be any associated software with the project. The focal distance is roughly 2 hand lengths (30 cm). No stereotopic vision can be achieved as there is only one camera.

Conclusions/action items:

This is a very simple design consisting of a smartphone stand that allows the phone to be angled down. The surgeon would need to hunch over to view the screen of the phone. Additionally, no stereotopic vision can be achieved as only one camera can be used at a time. This device included no software and is simply a stand to hold a phone.

Sam Neuman - Feb 27, 2023, 10:43 PM CST

https://www.pocketsuture.com/trainer

Sam Neuman - Feb 27, 2023, 10:43 PM CST

From their website:

2023/02/11 Academic publication on smartphone surgery system

Sam Neuman - Feb 27, 2023, 10:51 PM CST

Title: Competing Design: academic publication on a system utilizing a smartphone for microsurgery practice

Date: 02/11/2023 (02/26/2023)

Content by: Sam

Present: Sam

Goals: Document an academic project involving microsurgery practice systems

Content:

This device's design appears very similar to the previous smartphone stand design. The author's goal with this publication was to quantify how well the design worked in comparison to a surgical microscope. The authors used both experienced microsurgeons (attending surgeons) and medical residents training in microsurgery for their experiments. They devised multiple trials involving a standardized but varying number of sutures to be completed, and the time and quality of the suturing was quantified for comparison. Surgery was performed on fresh chicken breast. Finally, the authors quantified the number of errors made during the surgery trials. Through some complicated statistical analysis, the authors concluded that their proposed design was statistically equivalent to the professional surgical microscope.

Conclusions/action items:

This publication utilized a simple stand to hold a smartphone so that the viewer would look down at a magnified projection of the sutures. The design involved no software, and stereotopic vision could not be achieved as only one camera perspective could be used. Through statistical analysis, the authors concluded that their setup was equivalent to the use of a commercial surgical microscope. Due to this design's simplicity, it cannot be used as a standard for our design concept or comparisons, but it is a good proof-of-concept design to consider while moving forward.

Sam Neuman - Feb 27, 2023, 10:51 PM CST

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8795636/

From the paper:

2023/02/18 Possible C# Script solutions

Sam Neuman - Feb 27, 2023, 11:04 PM CST

Title: Possible C# code solution for iWebSocketClient Error

Date: 02/18/2023

Content by: Sam

Present: Sam and Haochen

Goals: Document a single possible solution for a compile error in C#

Content:

Below is a possible workaround in the code:

```csharp
bool started = false;

tex = 1;

void Update() {

    if (started == false) {

        StopWebcam();

        tex = null;

        StartStopCamClicked();

    }

}
```

Conclusions/action items:

It's my suspicion that there may be a closed infinite loop in the code. If the script does less than one cycle and this function is at the end, then redeclaring tex and non-null will allow StopWebcam to be called. Parallel processing will allow StartStopCamClicked to be called before the infinite loop in StepWebcam is completed (if it exists), which should hopefully allow this to be solved. I sent this to Haochen to test.

2023/02/21 Finalized Design Concept

Sam Neuman - Feb 27, 2023, 11:14 PM CST

Title: Finalized Design Concept
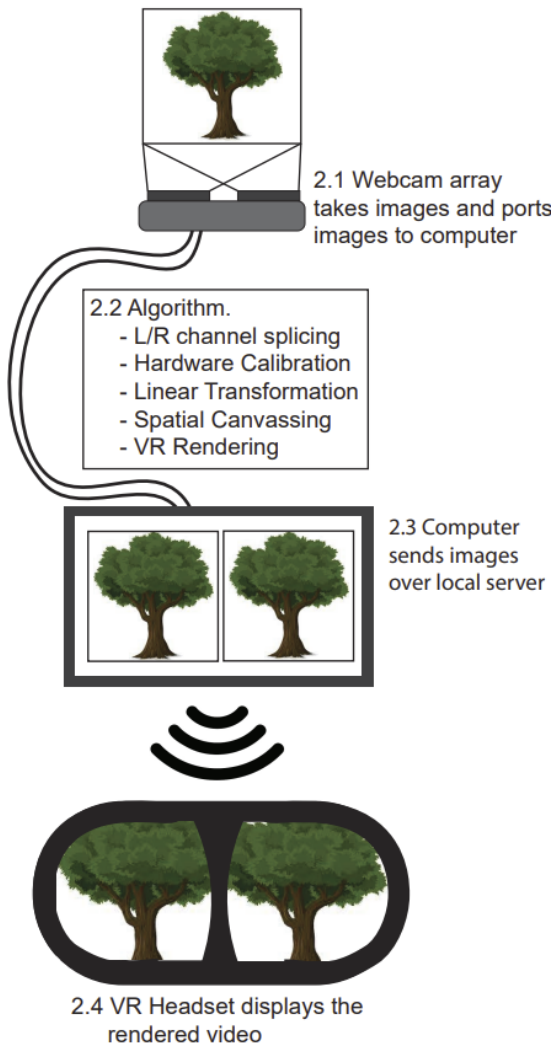
Date: 02/27/2023

Content by: Sam

Present: Sam and Haochen

Goals: Document our finalized design concept for the semester

Content:

Our design can be conceptualized in four main parts. In the first part, the two webcams take an image of the object from two different perspectives; these perspectives allow for stereotopic vision to be achieved. Next, the images are ported through a wired connection to a computer, where a basic algorithm is used to manipulate the images. This algorithm involves splicing of the left and right channel hemispheres, calibration of the hardware and a linear transformation if needed, canvassing of the images in the 3D vector space, and rendering in VR. After this, the computer transmits the manipulated canvas to the Oculus VR headset. Finally, the canvas can be viewed by the microsurgeon through the VR headset.

2.1 Webcam array takes images and ports images to computer

2.2 Algorithm.
- L/R channel splicing
- Hardware Calibration
- Linear Transformation
- Spatial Canvassing
- VR Rendering

2.3 Computer sends images over local server

2.4 VR Headset displays the rendered video

Conclusions/action items:

This design concept is very similar to the previous semester with the replacement of the mirror array with an array of two webcams. Simply, the webcams are placed at different positions to acheive stereotopic vision. The images produced by the webcams are manipulated by the computer before they are ported to the Oculus VR headest. The user can then view the two images on a 3D canvas after necessary manipulations have been performed.

2023/03/13 RasperryPI IP recognition

Sam Neuman - Mar 13, 2023, 6:03 PM CDT

Title: Method for RasperryPi IP recognition

Date: 13 March 2023

Content by: Sam

Present: Haochen and Sam

Goals: Propose a method for easy IP recognition of the RasperryPi

Content:

For the trouble the happens when the RasperryPi IP changes when rebooting:

(This is coded in rust but use cypython{} to run at top of Rust script)

```python
import subprocess

import smtplib

from email.mime.text import MIMEText

import datetime

import os


# Account Information

##################################################################


to = 'recipientemail@server.com' # Email to send to.

gmail_user = 'senderaddress@gmail.com' # Email to send from. (MUST BE GMAIL)

gmail_password = 'emnopqoodnzcvkir' # Gmail app password.

smtpserver = smtplib.SMTP('smtp.gmail.com', 587) # Server to use.


smtpserver.ehlo()  # Says 'hello' to the server

smtpserver.starttls()  # Start TLS encryption
```

```python
smtpserver.ehlo()

smtpserver.login(gmail_user, gmail_password)  # Log in to server

today = datetime.date.today()  # Get current time/date

arg='ip route list'  # Linux command to retrieve ip addresses.

# Runs 'arg' in a 'hidden terminal'.

p=os.popen("ifconfig | grep 'inet '").read()

#data = p.communicate()  # Get data from 'p terminal'.

print("p=")

print(p)

print("END p")

iplist = os.system("ifconfig | grep 'inet '");

print("iplist = "+str(iplist))

print("*****************")

print(iplist)

# Creates the text, subject, 'from', and 'to' of the message.

msg = MIMEText("IP Addresses:" + "\n" + str(p))

msg['Subject'] = 'IPs For RaspberryPi on %s' % today.strftime('%b %d %Y')

msg['From'] = gmail_user
```

```
msg['To'] = to


# Sends the message


smtpserver.sendmail(gmail_user, [to], msg.as_string())


# Closes the smtp server.


smtpserver.quit()
```

Conclusions/action items:

This may be a solution to the problems with RasperryPi restablishing a new IP every time it reboots. On the RasperryPi script, this can just be pasted above the existing Rust. I'm not sure if the python will be recognized with the given Rust package. If it doesn't, use the cypython{} script to enclose this. Sahand mentioned that this might not be necessary.

2023/05/03 Final Design Update

Sam Neuman - May 03, 2023, 10:33 PM CDT

Title: Some updates on the final design

Date: 3 May 2023

Content by: Sam

Present: Sam

Goals: Record some figures that I'm putting in the final report

Content:

Attached below.

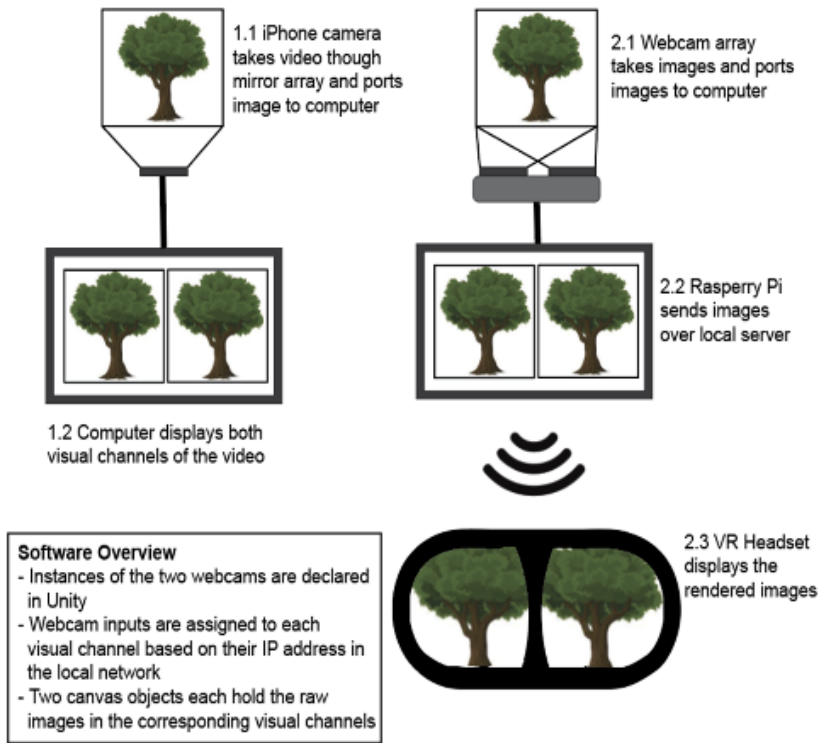Fig 1. Sahand using the system to do some suturing.

Fig 2. The two phases of design progress. Phase 1 was last semester, Phase 2 is this semester.

Conclusions/action items:

These figures are intended to be part of the final report. I wanted to include them in this notebook so there is some reference. Haochen and I have been updating these figures over this semester and the last semester. The figure on the array of the webcams in the chassis may need to be updated further.

2014/11/03-Entry guidelines

John Puccinelli - Sep 05, 2016, 1:18 PM CDT

Use this as a guide for every entry

- Every text entry of your notebook should have the **bold titles** below.
- Every page/entry should be **named starting with the date** of the entry's first creation/activity, subsequent material from future dates can be added later.

You can create a copy of the blank template by first opening the desired folder, clicking on "New", selecting "Copy Existing Page...", and then select "2014/11/03-Template")

Title: Descriptive title (i.e. Client Meeting)

Date: 9/5/2016

Content by: The one person who wrote the content

Present: Names of those present if more than just you (not necessary for individual work)

Goals: Establish clear goals for all text entries (meetings, individual work, etc.).

Content:

Contains clear and organized notes (also includes any references used)

Conclusions/action items:

Recap only the most significant findings and/or action items resulting from the entry.

2014/11/03-Template

John Puccinelli - Nov 03, 2014, 3:20 PM CST

Title:

Date:

Content by:

Present:

Goals:

Content:

Conclusions/action items: