ENTRY JANUARY 29, 2015

Created first prototypes for attachments

Instagram link

ENTRY FEBRUARY 1, 2016

Got openFrameworks working with Amazon Kindle fire with Android Studio. Ran examples.

ENTRY FEBRUARY 7, 2016

Image displayed on tablet responding to accelerometer data.

Here depending how the tablet is oriented the image orients accordingly.

EDIT - February 22, 2016

The orientation isn’t quite perfect as the image does not rotate based on orientation. Currently devices movements only changing x position. Does rotate draw position but not actual image.

(post video showing this)

ENTRY FEBRUARY 13, 2016
New prototype attachments designed for tablets. Features 360 rotation with cylinder rather than ball and socket, only allowing for simple rotation. New flat part will stick on to the top cylinder and attach to the back of the kindle case.

Instagram link

---

**Entry February 16, 2016**

*Programming*

Issues with using openFrameworks Android camera examples with Android Studio. Get a long error message. A look on the openFrameworks forum shows others have had similar issues. I have posted to say the same.
Back up plan is to look at pre-recorded video. I have been able to get pre-recorded video running on the kindle fire. So this is an option.

But currently the idea is to access the rear camera of the tablets. Moving tablets up and down and rotating will change the view of the video. The accelerometer will be used to create unexpected behaviour. For instance, show the video upside-down and to change the behaviour from static photo/ delayed video back to the live interaction.

**Frame building**
Looking at testing some wire grippers I have purchased. Will look at attaching these to the 3d printed attachments by creating wholes for the grippers to screw into. Tablets will still be able to rotate as the 3d printed part allows rotation independently of other parts.

**Tablets**
Looking at creating fake tablets to reflect just the glass/foil/mirrors. I can purchase glass replacement for kindle fire's. So this could be a solution as they would fit into the printed enclosures like the real tablets and look inline with the others.

---

**Entry February 22, 2016**

**Programming for tablets**
Looking for solution to change rotation of the image/videos displayed based on the orientation of the tablet. Maybe better to keep the orientation of the window set so as to only allow the image/video to rotate based on device’s orientation.

To keep the orientation of the window set there is a line in the AndroidManifest.XML file (openFrameworks>Examples>Android>)[in one of the examples].

```xml
android:name=".OFActivity"
android:label="@string/app_name"
//This Line
android:configChanges="orientation|screenSize"
android:screenOrientation="landscape">
<intent-filter>
```
Will need to use .getOrientation(); be able to rotate the image based on tablet orientation.

---

**Entry February 29, 2016**

Been working on rotation from discussion with Theo with taking rotation more simply. Mapping the y accel value and the degrees to a float value angle.

```java
float angle = ofMap(accel.y, 1, -1, 90, -90);
// (set anchor point)
image.setAnchorPoint(image.getWidth()/2, image.getHeight()/2);
ofTranslate(with/2, height/2);
ofRotate(angle);
```

This has allowed me to rotate the image from the centre of the screen by translating to the centre and setting the anchor point of the image to the centre.

**Video demonstration found here.**

Also low pass filter is required for reducing the fluctuations in the accelerometer values on the image.

```java
float currentAngle = 0;
float destAngle = 45;
currentAngle += (destAngle-currentAngle)*0.1;
```

Still unable to get camera to work completely successfully. Got camera examples running on a MotoX (2014) but crashed after a while. Maybe issue with android version in this case. The newer tablets may work better. Will need to get a camera app running in background and sending this over an IP address to my oF app with local host/LAN.

---

**Entry March 7**
Did some tests with IP cam with processing. When camrea is rotated this rotates the image. So therefore I have solved my issue. I will only need to use the accelerometer in the tablets to sense if the tablet has been moved. This will allow me to add my different behaviours such as video effects/random rotations.

I will look at using Camera Stream - WiFi IP Webcam which is a IP capture app available on the amazon app store.

Looking at trialing this with openFrameworks. The addon ofxIpVideoGrabber doesn't seem to be working.

---

**Entry March 8**

Designed and printed new prototype for attachement. Rotation part not working quite right yet. More experimenting needed. But here the sides of the base part have been extended to a rectangle. Here the wire grippers will be drilled in to hold on to the steal wire.
Entry March 10

After posting issue on git hub about issues with ofxIpVideoGrabber got issues resolved. Had been an issue with my project generator. However unable to get that working with my spesific IP cam as the addon only supports axis cgi ip cam streams or ones using mjpeg streaming. I am now trying to look in to ip cam that will work with openFrameworks in some other way.
Entry March 11

Got acess to GDS to build frame. Found some good bits of wood that will be used to build my frame. Looking at it being around 1.5mx1.5m. I think a square format could look quite good to display the tablets.

**Issue with tablets** - Amazon seem to have discontinued the fire 7”. They currently have a 6–8 week dispatch time. They now seem to be pushing their new Amazon fire 6HD. But this is £79.99. May have to look at buying tablets second hand on ebay.

---

Entry March 14

Got to GDS this moring to look at the wood again. I found another peice of wood that would measure the corret length. 1.5mx1.5m seems to be a good size. I then discovered that the timber was not flat and bowed. I went up to the Fine Art wood lab to see if this could be resolved, it couldn’t. But I have found a supplier that could provide good flat timber that would be ideal.

They seem to have two low cost ideal pices.

**Timber 1**

This is quite an unfinished wood, quite rough. But I think it would be good to be quite plane and simple without seeming too sterile. It has an ideal width and comes in good lengths that could be cut.

**Timber 2**

This peice is quite well finished, a bit more expensive. I’m also not sure whether that it will sit well with the mental steal wire.

I think I will look to build my first version at leat out of Timber 1, leaving it quite plane. I also looked in to making it a well structured frame by making 45 degree cuts in the wood so it would be strong and look seemless. I can also inforce this with metal L brackets for extra strength.
I will get the wood by the end of this week.

---

Entry March 15

SOME SUCCESSES!

I managed to get the addon ofxIpVideoGrabber working with my own IP stream. I did this from an app IP Webcam that I found on the Google play store. I tested this with an android phone then managed to get the apk to install it on to the Amazon kindle. This worked and I ran the app using ofxIpVideoGrabber on a computer to test receiving it.

I had to fiddle with getting the of app written on the computer to compile with android. Bit of trial and error. Including selecting correct part of my code was working and adding that to the android project while keeping the necessary android components. Sometimes a few make file errors. But got there. I then got it compiled but the video wasn't displaying.

```
03-15 22:35:23.777 27600-28008/cc.openframeworks.CPdebugIP
E/IPVideoGrabber: Exception : [http://192.168.0.3:8080/video]: I/O
error: Permission denied 03-15 22:35:23.808 27600-
27631/cc.openframeworks.CPdebugIP E/IPVideoGrabber:
[http://192.168.0.3:8080/video] Connection retries exceeded,
connection connection failed. Call ::reset() to try again.
```

I found out it wasn't connecting to the video because the android app requires the permission to access the internet. This can be altered in the AndroidManifest.xml file with the added line:

```
<uses-permission android:name="android.permissionINTERNET"></uses-
permission>
```

I should have remembered these permissions from my entry February.

I also got the top menu bar to go away and display the app in full screen with the line in the Application section of the xml file:

```
android:theme="@android:style/Theme.Black.NoTitleBar.Fullscreen"
```
The video streams very slowly with any video resolution above 640. I have got it working at 320x240. It works very smoothly albeit low resolution.

---

**Entry March 17**

**Tablets**

Excited at opportunities that using an IP cam have allowed. This along with the accelerometer’s in the tablets could allow for further actions on the viewers interaction.

I can split the action based on the interaction into different possibilities. For instance:

- **50%** - Rotates how you would expect

- **25%** - Rotation becomes random based on if the tablet senses movement, it will pick a different rotation value.

- **25%** - The image displayed on the tablet changes to one of the others. Or the image on another tablet changes to the camera feed of the tablet the viewer is interacting with. This can be accomplished by pre-loading the IP address for all the IP cam feeds into the XML file. I then could look at sending OSC commands to force the other tablet to change the IP cam feed to the other tablet.

This adds to the complexity of the kaleidoscopic pattern of all the tablets. It will be fun to play with all these variables I they can be tweaked to allow for this. Because of this I need to get the other tablets so I can conduct these test.

**3D Printing**
New Rotation part

Printed the new rotation part with much thicker infil. 60% resolution and 4 shell infill. This is so it can withstand holding the tablet.
I have now redesined this slightly to take off the top tip and extend the lower thicker area to increase the strength further. Along with this I created new plate that will stick to the kindle case allowing the tablet to attatch to the attctment so the tablet can rotate. The slight hole to allow the rotation part to fit in was done by using the boolean modifier in blender, where you can interset objects to join them togther or cut into a object.

Need to create holes in the base part to fit the screws of the wire grippers.

---

**Entry March 18**

Created holes in the base part, again done with the boolean modifier.

---

**Entry April 14**
Tablets have arrived!!

All tablets are here. Will begin loading on the OF app to test.

Been working on the frame today. Created the frame. Have realised that the frame may not be strong enough for the mentioned steal wire. Been trying to put in wooden braces for extra support. But the metal brackets are not strong enough. May need to get thicker wood and stronger brackets.
Entry April 15
I decided overnight that it would be best to build a new frame with stronger, thicker wood. I also needed stronger L brackets to strengthen the frame. So that day I built a new frame and then in addition I added extra support from corner wooden parts (seen in first image) cut with 45 degree angles. I have also cut holes for my metal steal wire.
Entry April 18

PROGRAMMING

I was having some difficulty with gradle with my android development. Need to be using 2.2. I have discovered but android studio has been forcing updates on the gradle. I edited this in the android project folder > gradle > wrapper > gradle-wrapper.properties and edited the distribution url

distributionUrl=https://services.gradle.org/distributions/gradle-2.2-all.zip

Need to keep it this way to avoid disruptions with my program running on the tablet.
I have now got two tablets switching video between themselves now ([videoLink](#)). I am using the IP WebCam set up on both tablets with the APK to install on to android. In the xml file I have added the IP address of both video streams. Then with a touch down event I have triggered a for loop to load in the next camera in from the IP address list. Continuing to touch the screen results in cycling through the streams. I will keep it like this for the final exhibition or switch this randomly based on accelerometer values or OSC.

```cpp
void ofApp::touchDown(int x, int y, int id){
  for (std::size_t i = 0; i < NUM_CAMERAS; ++i)
  {
    ofRemoveListener(grabbers[i]->videoResized, this, &ofApp::videoResized);
    std::shared_ptr<Video::IPVideoGrabber> c = std::make_shared<Video::IPVideoGrabber>();
    IPCameraDef& cam = getNextCamera();
    Poco::URI uri(cam.getURL());
    c->setURI(uri);
    c->connect();
    grabbers[i] = c;
  }
}
```

Calls the function `getNextCamera()`; choosing the next ip cam in the xml file.
3D PRINTING

Been doing some 3d printing today with the new design. First attempt failed because it was not scaled enough and wasn’t using the 4 shell thickness. Reprinting these now. **Update 19th April** - New prints were a success, printed at correct specifications.

**Entry April 20**

**Printing kindle cases**

Due to not being able to find a suitable case in order to glue to the rest of the attachment I am having to print kindle cases. This comes with the advantage of being able to have the attachment plate printed on to the case, saving having to glue resulting in a messy look.

First print failed due to it being slightly too small and the supports had not printed correctly meaning that it was not as strong as would have been liked. Making some adjustments in a program called MeshMixer. This allows for real world measurements to be used and objects can be adjusted for optimum 3D printing.