Andrew Richardson – Teesside University p4053489

Computing Project Eyestrain reduction application that dynamically adjusts monitor settings with a light sensor

Table of Figures

Figure 1 IPS diagram 8
Figure 2 Circuit diagram 11
Figure 3 WMI not supported error 15
Figure 4 Monitor Adjustment and eye strain 18
Figure 5 Eyestrain and time spent on computers 19
Figure 6 Eyestrain frequency bar chart 19
Figure 7 Eye strain and frequency of breaks 20

Contents

Introduction 3
Abstract 3
Research and analysis 3
  Aim 4
  Causes of eye strain 4
  Health effects of blue light 5
Displays 6
  Types of displays 7
    Cathode Ray Tube 7
    Liquid Crystal Display 7
    Light Emitting Diode 8
    Organic Light Emitting Diode 8
  Display Properties 8
    Brightness 8
    Contrast 9
    Gamma 9

Resolution 9
Refresh rate 9
Design 10
Implementation 11
API explanation 12
Sine wave variation 12
Technologies considerations and costing 13
Setbacks 15
Testing 18
Breaks 19
Display 20
Law and Ethics 20
Reflective evaluation 22
Limitations and further development 22
Evaluation of testing 23
Skills developed 23
References 24

Introduction

In a study by Agarwal (Agarwal, et al., 2013) 53.8% of subjects tested that used a computer for 6 hours or more at work reported suffering from eye related problems. This reduced to 40% when they took regular breaks and 30% when brightness adjustment was used. In another study (Bhanderi, et al., 2008) 46.3% of subjects of 419 subjects had asthenopia during or after computer work. The cost of diagnosis and medical treatment has been estimated to be $2 billion (Kanitkar, et al., 2005). As the use of tablets and smartphones has increased people are now spending more time looking at a screen. It is believed that in addition to taking normal steps to alleviate asthenopia or eye strain monitor adjustment was found to reduce problems reported that a desktop app can be created to automatically adjust the monitor settings according to the ambient lighting in the room.

Abstract

The aim of the project is to write a desktop app to dynamically alter screen settings from a light sensor for the Linux Operating System and to investigate eye strain in computer users, if screen use can lead to more serious eye conditions and how the app can facilitate the easing of symptoms. Other means of reducing eye strain are also to be researched.

Research and analysis

Eye strain affects office workers and general computer users as daily screen time has increased with ownership of smartphones and tablets (Consultancy, 2017) The report goes on to say that in 2012 52% of UK adults owned or had regular access to a smartphone which had risen to 85% in 2017, during the same period tablet usage and ownership had increased from 16% to 68%. Laptop ownership remained relatively stable from 73% to 78%

Aim

The objective of the project is to identify causes of eye strain, investigating the technologies used, and to develop an app to help reduce them and to recommend other user practices that the operator can use to maintain healthy vision.

This learning outcomes of this project aim to improve Java programming skills and understanding of the UNIX command line, in addition to gaining understanding of Arduino microcontrollers and the basics of circuit design.

Causes of eye strain

Computer users blink less often leading to dry eyes (Portello & Rosenfield, 2010). Eyestrain usually goes away after taking breaks but can be persistent or could be a sign of an underlying condition. Eye strain can lead to discomfort, headaches, difficulty in concentrating, diplopia or double vision, increased sensitivity to light or photophobia, blurred vision.

The best way to avoid eye strain is to take regular breaks and look far into the distance. Lights and sunlight coming in through a windows and bright overhead lighting can also be a problem. Glare reflected from the computer screen is a contributing factor, this can be alleviated by using an anti-glare screen and hood over the monitor and using a matt computer screen. A screen with the highest resolution possible should be used (Specsavers, n.d.). It is recommended to match the brightness of the monitor with the surroundings and use a larger text size. Shorter wavelength visible light causes more eyestrain than longer wavelength colours, the monitor should be adjusted to have a lower amount of blue light as blue has the shortest wavelength whilst red is on the opposite side of the visible light spectrum.

Monitors emit blueish light that is thought to be harmful to sleep. The brightness levels can be turned down, but colour temperature levels are not usually controllable by the user.

The Seegehalli paper (Seegehalli, 2016) identifies screen flicker as being associated with visual fatigue, this improves with higher refresh rates. Cathode Ray Tube displays refresh the entire screen that is constantly redrawn as an electron beam steered using magnetic fields is scanned across a phosphor panel, hitting each dot that illuminate and then fade within a fraction of a second requiring a refresh every 50 times a second. CRT displays have a significant flicker that is bad for eye strain, but they have been replaced with LCD and LED displays. LCD displays use cold cathode fluorescent lamps or CCFL that have a glow after they are turned off, the glow is more significant than LED resulting in reduced flicker. Reducing the brightness can also increase eye comfort but this is often done with Pulse Width Modulation which is a way of dimming using a digital signal by turning the pixel on and off with larger gaps to reduce brightness. Running PWM at higher frequencies reduces the flicker effect.

The author of the Seegehalli paper theorises that keeping the screen at a constant brightness and distance from the user means that the eye muscles responsible for focusing have no opportunity to relax which is why reading books strains the eyes less than moving pages when reading a book. One proposed solution in the Seegehalli paper is to vary brightness according to a pattern such as a sine wave or from a randomiser, among others. From running the program, the brightness from the screen or luminance was not observed to change much as the illuminance in the room mostly there could be some variation based on the automatic setting.

Health effects of blue light

High frequency blue light has the highest energy of visible light compared to green and red light. Exposing our eyes to blue light causes the lens in our eyes to become opaque in a condition called cataracts which needs surgery to replace the lens with an artificial one, however, there is no evidence that blue light from screens is harmful as the levels are too low (Sheppard & Wolffsohn, 2018). Cataracts are also associated with lifestyle such as diet and smoking, in a Taiwan study (Shih, et al., 2014) sitting for prolonged periods of over 7 hours a day was found due to be a risk factor for cataracts which could be a problem for people spending long periods spent seated at a computer. Blue light also damages the light detecting layer of cells at the back of the eye called the retina that could over time lead to vision problems including age-related macular degeneration although most damaging blue light is from the sun from which higher levels are given out, less than 1% emitted from computer screens is thought to be harmful (Royal National Institute of Blind People, 2017). In a study blue light exposure from the sun was only found to be harmful when combined with low concentrations of antioxidants in blood plasma which can be remedied with dietary changes (Fletcher, et al., 2008).

Blue light is useful for regulating our circadian rhythm which is the bodies natural wake and sleep cycle. Exposure to blue light in the morning inhibits the brain's pineal gland production of the sleep hormone melatonin. As a result, room exposure to blue light before bed in the room was found in a Chamberlain study (Chamberlain, et al., 2011) to disrupt sleep and may create a similar affect for computer and mobile device users.

The design of apps and web pages is a factor in readability with higher colour contrast being easier to read than low contrast. Luminance as well as colour contrast appeared to be a factor in a study (Shieh & Lin, 2000)

A study of subjects with a random number wearing short wavelength blocking glasses found that in a two-hour duration computer task subjects wearing the high block glasses reported significantly less eye strain in comparison to subjects wearing none blocking glasses (Lin, et al., 2017). Tests were conducted on an Apple iPad 3 by testing subjects on white and sepia background colours, with the latter with decreased blue light emission. In a questionnaire most subjects reported having decreased symptoms when using the sepia mode with reduced blue light effective radiance enabled (Isono, et al., 2013). Both studies only focused on short term use and did not investigate long term use.

In conclusion whilst of blue light from monitors is not harmful to long term ocular health and does not cause serious conditions including AMD and cataracts, it can cause problems in the short term through eye strain discomfort and sleep disorders. As this is a new area of research and the use of tablets and smart phones leading to an increase in screen time is a new phenomena there is not yet enough data to establish a long term link.

Displays

Different types of displays and monitor properties were reviewed as the program aims to adjust properties and there is also the question on whether different display types are better regarding eye care.

Types of displays

Cathode Ray Tube

Cathode Ray Tubes use a vacuum tube from which a stream of negatively charged subatomic particles called electrons is given out by a negatively charged cathode that are focused and accelerated by positively charged anodes to steer the electron beam towards a phosphor coated screen that gives off visible light. Three beams are used to combine red, green and blue to give a full colour image. CRT displays are bulky in comparison to newer flat panel display technologies and use more power as the image must be refreshed many times a second. CRT monitors and televisions have fallen out of use and are no longer manufactured.

Liquid Crystal Display

LCDs use organic crystalline molecules that are twisted when electric fields are applied to allow light through a polarizer. LCDs do not generate light, so a backlight is required. There are two types of LCD: Twisted Nematic and In Plane Switching. The newer IPS displays offer better colour reproduction and better viewing angles but have longer response times than TN, consume more power and have higher manufacturing costs. (TN Panel, n.d.) The main physical difference between TN and IPS is that in TN the electrodes are on opposite sides of the two glass substrates with the electric field moving from one end to another whilst in IPS the electric field moves from one TFT to another.

Vertical Alignment (VA) panels were developed later to have the advantages of TN and IPS panels with better viewing angles than TN and superior response time to IPS but are still outmatched by TN displays (Hale, 2018). VA have higher contrast ratios than both IPS and TN as they are better able to block light from the backlight.

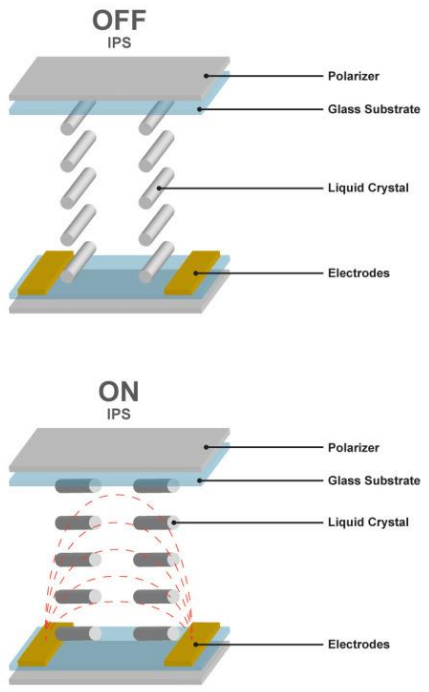


Figure 1 IPS diagram

Light Emitting Diode

Light Emitting Diodes are inorganic semiconductors that give off light, so unlike LCD displays no backlight necessary making them more energy efficient than an LCD which also avoids the problem of having to block out the backlight giving them darker blacks.

Organic Light Emitting Diode

OLED displays use organic semiconductors that offer better colour reproduction than inorganic semiconductors used in LED displays. They can be curved and are more power efficient. OLED displays are a newer technology that is more expensive to manufacture than other types of display.

Display Properties

Displays have different settings that are explained in this section.

Brightness

Brightness is the intensity of luminance from a monitor. Luminance is light emitted from a light source which in this case a monitor, is measured in cd/m^2. Ambient lighting from all directions in a room or illuminance is measured in lux. (U.S. Department of Energy, 2012)

Contrast

Contrast is the difference in luminance between the brightest white and the dimmest black parts of the display.

Gamma

Gamma is the non linear relationship between luminance and light values of the image. Gamma must be corrected as displays do not produce the same light intensity as the input voltage. The red, green and blue in colour images. The curves are altered for colour correction. In the project the gamma for blue and green has been decreased to lower the amount of higher frequency light that is given out without affecting the colour too noticeable (Berger, n.d.). Differing RGB gamma values affect the hue of the image.

Resolution

Resolution is the number of pixels in a screen which is measured by the number of horizontal by the number of vertical pixels. The total number of pixels can be calculated by multiplying this number together. A monitor may have 1920x720 pixels.

Refresh rate

The refresh rate is the number of times the image can be updated per second, it is measured in frequency per second or Hertz, abbreviated as Hz.

Design

As the need for matching the luminance of the display with the illuminance of the ambient light in the room, light sensor hardware is required with a linux app with some means of communicating this data from the light sensor to the computer to dynamically adjust these settings. Blue light has been identified as being a problem for screen users so the app must also find a way to reduce this. As users may wish to make adjustments based on their own preferences the app should be customizable. A fluctuating brightness level should be available based on the brightness bias.

A prototype Linux app using off the shelf electronics prototyping hardware will be used for demonstration purposes.

A way of reducing higher frequency light must achieved without affecting the colour contrast so that it has a negative impact the readability which comes into user experience or UX.

The Web Content Accessibility Guidelines 2 (WCAG 2) specifies colour contrast conformance requirements for foreground text and background colours. For small text the minimum contrast ratio is 4.5:1 to achieve the lowest conformance rating of A. Colour can easily be checked from hexcodes using tools provided by webAIM that compare the contrast ratio between a foreground and background colour returning the contrast ratio and whether it passes the WCAG AA or AAA standards. It is more difficult to measure if the screen properties have been altered. The WCAG 2 guidelines are more tolerant for larger text insofar as colour contrast ratios are concerned.

Implementation

Using a Light Dependent Resistor (LDR), alternatively named a photoresistor on an Arduino to give out a voltage level on an analogue input on the board that is then sent to the Java program that can be approximated as a brightness value reading. The desktop Java application then acts as an interface to automatically control the screen settings.

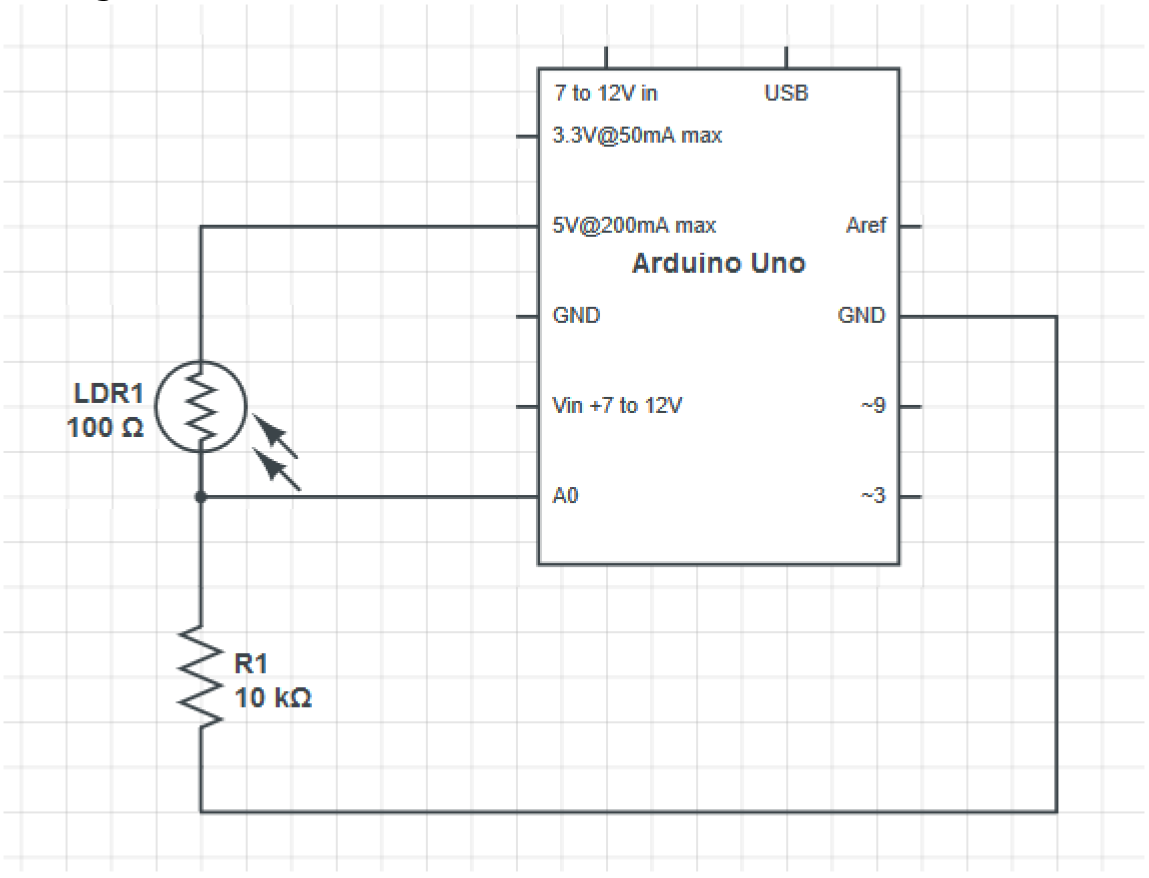


Figure 2 Circuit diagram

A user set bias to increase or decrease the brightness from the automatic level can be applied. Gamma is the intensity of white light, the mixing of red, green and blue light gives colour temperature. Presets can be enabled to control the colour temperature at a static level to decrease the amount of high frequency light at the blue end of the visible light spectrum.

Xrandr, part of X Window System or X11 is used to change the brightness and gamma that can be entered manually into the terminal for fixed settings as:

```
xrandr -output DP-0 --brightness 0.8 -gamma 1:0.9:0.7
```

In the Java program the terminal commands can be executed using the runtime class and the command entered as a String inside the exec method with values concatenated into the string. This must be done inside a try-catch block.

```
Runtime.getRuntime().exec("xrandr --output " +
videoCardOutput + " --brightness " + brightnessSetting + "
--gamma " + gammaRSetting + ":" + gammaGSetting + ":" +
gammaBSetting);
```

The app uses the jSerialComm API to get the information from the light sensor as a voltage level. A user bias is added to the voltage level to allow the user to adjust the level to their preferences. Java does not have the functionality to communicate with an arduino over its serial to USB port interface so a third party API is required.

API explanation

An Application Programming Interface is an additional library to extend the functionality of a programming language. Many languages do not have a built in graphical library, C++ does not. Users wishing to write a graphical application in this language must use QT or another API available. Java has no 3D graphical library so openGL is a popular choice. APIs may also be used to build apps for a service such as Twitter's Ads API.

A user bias slider has been added to the user interface that allows the user to adjust display brightness from the default automatic value based on their needs and preferences. If it is set to zero only the automatic brightness level is used.

To prevent the Xrandr value being set out of bounds by the user bias, that is over 1.0 or a negative value an if statement containing Boolean operators was used in the conditional parameters so that if the it would default to the automatic value only.

Sine wave variation

As the aforementioned Seegehalli paper proposed using varying brightness to prevent the muscles responsible for focusing from becoming tired from remaining in a fixed position for a long period of time. A sine wave was incorporated using the sine function in the math class in the standard Java library, this is added to the user bias that could be toggled on and off by the user with a checkbox. The same conditional parameters also contained an AND logical operator so that the user bias and sine wave would stay within the limits. A for loop was used and the sine wave was calculated to keep it within the limits of xrandr brightness value. However, this reduced values and caused the brightness to change significantly, making the screen flicker, which as previously discussed is undesirable. A further problem was that the loop that generated the sine wave was resource intensive and caused noticeable slowdown of the program and the machine in general.

As there are many very similar values sent by the Arduino per second the code has been altered to create an array that sends out an average every time the array has been populated.

Technologies considerations and costing

The aim was to keep costs as low as possible due to financial restraints. Netbeans 8.2 was chosen for development because it is free to copy and distribute but cannot be changed under the General Public Licence Version 2. C# would have been unsuitable as it is more for Windows, Python was strongly considered as it has the Tkinter API which the developer has found easier to use than Swing for creating Graphical User Interfaces but Java was chosen instead because of the developer's familiarity with the language, it is write once run anywhere and has a built in graphical library which is useful for creating user controls.

Java is a C-based interpreted language; the Java Virtual Machine runs on the software layer of the operating system and the Java program is compiled into an intermediary language called bytecode before being compiled a second time into assembly that can be understood by the operating system. There is a JVM for the major operating systems (Windows, Linux, macOS). The second compilation differs depending on which JVM is being used. The interpretation process during interpretation means that more memory and processing resources are used than an application than a compiled language such as C++. Java used a JIT or just-in-time compiler that optimises the process, so methods used less often are compiled later or excluded from compilation altogether. (IBM Knowledge Center, n.d.)

The third party jSerialComm API by Fazecast was used to communicate with the Arduino was used because it was simple to install and freely available. The electrical components were cheap to obtain as only three jumper wires, a breadboard a resistor and photoresistor was required.

An Arduino Uno board was the most expensive purchase although at £20 they are relatively cheap and many third party clones are available. Arduino operates under a creative commons licence and allows clones to be sold on the condition that they are not sold under the name Arduino or -duino suffix. An official Arduino was used in this project because it was already in inventory and reportedly many clones suffer from reliability issues. An alternative is to build the microcontroller by using the design available on the Arduino site and buying the components off the shelf. As the project uses hardware that also requires setup beyond what a consumer is reasonably capable of it is only suitable for prototyping.

A raspberry pi was chosen to demo the project as this is more cost effective and more portable than a computer at around £30.

X Window System was developed in the 1980s at MIT as a bitmap windowing system for UNIX-like operating system including Linux (Shields, 2019) and is now maintained by the X.Org foundation. The windows and input devices such as keyboard and mouse are handled by the X server. Applications running on X are clients. The X server can run both local and remote programs. Window Managers (WM) such as Enlightenment handle the appearance of a desktop environment (DE) such as Cinnamon or GNOME or may be used standalone. A lot of window managers (WM) are integrated into a DE (arch linux, n.d.).

Wayland has been developed as a replacement display server for X and currently ships with major linux distributions as the default with support from many common Window Managers. Wayland does not yet have support for altering the brightness or colour gamma of the display. The new display server provides legacy support for X.

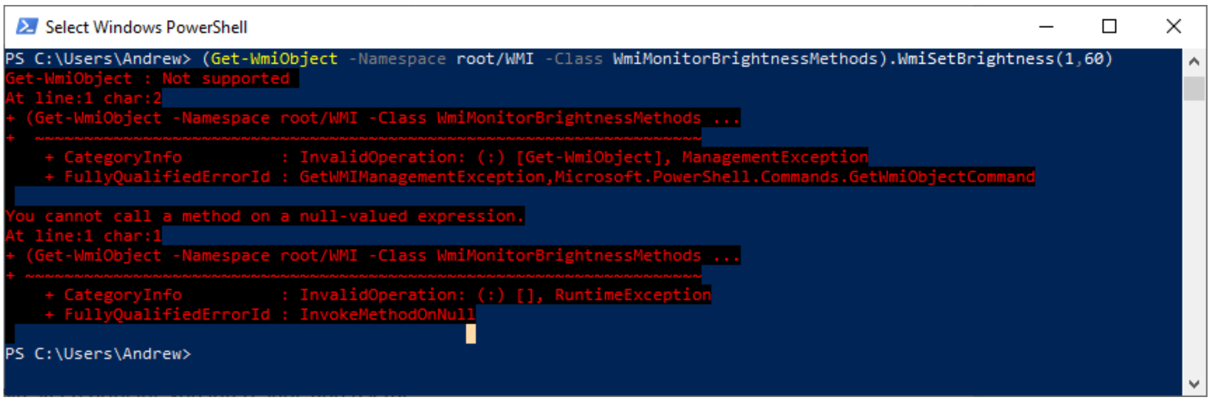
X Window System contains the X Resize, rotate and reflect extension that is a command line tool that can be used to adjust screen resolution but in this project is useful for changing the screen brightness and gamma.

Setbacks

Xbacklight was found to be only usable on supported devices that allowed the user to the user to control the monitor backlight brightness from the laptop keyboard and some all-in one computers. Another component of X11 was chosen instead, xrandr that stands for resize and rotate. X11 still has legacy support in its successor, Wayland although this will not always be the case. Wayland does not currently have the functionality to adjust brightness and colour temperature at present.

Operating system detection has been incorporated to allow the program to be improved so that it can run on multiple platforms. Like xbacklight, WMI shell commands were found to only work on supported devices including laptops. The PowerShell command that used WmiSetBrightness method to set the brightness to 60% was attempted only to throw a Wmi not supported error:

```
(Get-WmiObject -Namespace root/WMI -Class
WmiMonitorBrightnessMethods).WmiSetBrightness(1,60)
```



Select Windows PowerShell

```
PS C:\Users\Andrew> (Get-WmiObject -Namespace root/WMI -Class WmiMonitorBrightnessMethods).WmiSetBrightness(1,60)
Get-WmiObject : Not supported
At line:1 char:2
+ (Get-WmiObject -Namespace root/WMI -Class WmiMonitorBrightnessMethods ...
+  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (:) [Get-WmiObject], ManagementException
    + FullyQualifiedErrorId : GetWMIManagementException,Microsoft.PowerShell.Commands.GetWmiObjectCommand

You cannot call a method on a null-valued expression.
At line:1 char:1
+ (Get-WmiObject -Namespace root/WMI -Class WmiMonitorBrightnessMethods ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (:) [], RuntimeException
    + FullyQualifiedErrorId : InvokeMethodOnNull

PS C:\Users\Andrew>
```

Figure 3 WMI not supported error

There were some delays in waiting for components to arrive including the photoresistors and the USB to TTL Serial Converter. Some problems were also experienced with the Arduino malfunctioning when connecting to the USB interface that was later solved by fully inserting the USB cable. jSerialComm that has Linux support was chosen as the API to send the voltage data from the arduino to the Java application running on the PC. The API was installed by going into the context sensitive menu on Libraries in the project in Netbeans and adding the JAR by navigating to it in the file explorer. With jSerialComm-2.4.0.jar included in the project libraries the API then functioned correctly. Whilst waiting for components the code is being attempted before it can be tested with the hardware.

```
sp.setComPortParameters(9600, 8, 1, 0);
```

The setComPortParameters method set the baud rate which is the rate a signal changes per second, measured in hertz, number of data bits, stop bits and parity. In my program the baud rate has been set to 9600, there are 8 data bits per word set in newDataBits, 1 stop bits in a word and no parity error detection is used so it is set to 0 for NO_PARITY.

The Serial Transistor-Transistor Logic to USB was found to be no longer required as serial to USB is already included on the microcontroller board, simplifying the circuit and reducing the cost and size of the sensor. The fixed 10 kiloohm resistor after the variable resistance photoresistor voltage can be calculated using

Voltage output = R/(R+10K)*VCC

R-resistance of variable resistor or photoresistor

VCC – power supply voltage which is 5V

After the photoresistor there is an ADC (Analogue Digital Converter).

To convert the analogue reading to voltage the sensor value is multiplied by the reference voltage which is 5.0 and divided by 1023.0. This is because the Arduino's analogue inputs give a reading of between 0 and 1023 which must be divided by the circuit's initial voltage to give the true potential difference reading.

Voltage = ADC * (Vr/1023.0)

As written in code:

```
float voltage = sensorValue*(5.0/1023.0);
```

Most people are familiar with base 10 numbers also called decimal or denary. Computers run on binary base two that is converted to hexadecimal, base 16 and then denary which are readily understood by users. The analogue input is an unsigned binary number that has a maximum denary value of 1023 and can be expressed in binary as:

$0011\ 1111\ 1111_2$

In hexadecimal which can be used a short hand way of writing binary numbers this can be written as:

$3FF_{16}$

1024	512	256	128	64	32	16	8	4	2	1
	1	1	1	1	1	1	1	1	1	1

As there are 10 1s it can be deduced that it is a 10 bit number.

The sine wave was implemented in the method as calculated from a loop but was later cut out of the program for reasons explained in implementation.

```
//    public float sineWave()
//    {
//        double sineOfBias = 1;
//        float sineOfBiasAsFloat = 20;
//        //float sineOfBiasAsFloat;
//
//        for(int i=0;i<20;i++)
//        {
//        sineOfBias = Math.sin(i);
//        sineOfBiasAsFloat = (float)sineOfBias;
//        }


//        return sineOfBiasAsFloat;
//     }
```

Xrandr was found not to run on raspberry pi running Raspbian, even when X11 was installed. The exact cause is unknown but this is suspected to be due to xrandr not being supported by its graphics processor unit.

Testing

Testing was done as an intercept survey about their computer usage habits and experience of eyestrain if applicable to find the prevalence of computer related eyestrain. The survey undertaken had 26 respondents.

The sampling was not random and may be over representative of eye strain sufferers as they may be more likely to respond to a survey about a condition they suffer from. As the survey was done online all the respondents can be assumed to be computer literate.

50% of respondents said they experienced eyestrain when using a computer. 34.6% of respondents said they spent between 6-8 hours a day on a computer, with 30.8% or 1 fewer subject spending 8 hours or more. 73.1% of respondents used a computer for work.

The survey was created in Google Forms and shared for responses on various social media platforms. Pivot tables in Google Sheets were used for statistical analysis.

44.4% of users that experienced eyestrain did not adjust the monitor settings. 52.94% of affected users adjusted their monitors. This could mean that it is better to use factory settings for eye strain, alternatively users that suffer eye strain are more likely to adjust their monitors in an effort to ease their symptoms. The margins in such a small data set vary just by one which is not statistically significant, a larger sample size is needed to get any meaningful results.

COUNTA of Do you adjust the monitor settings?	*Do you adjust the monitor settings?*	
Do you experience eye strain when using a computer?	No	Yes
No	5	8
Yes	4	9

Figure 4 Monitor Adjustment and eye strain

Comparing eye strain and length of time spent daily on a computer shows that the group that spent between 6-8 hours were most likely to suffer from eyestrain at 38.46% of all groups. The group that was least likely to suffer from eyestrain spent 8 hours or more a day at 46.15% of the all groups and only 25% of that group suffered from eyestrain. None of these groups reported no eyestrain. 100% of the 2-4 hour group respondents suffered eye strain.

COUNTA of How long do you typically spend on the computer most days?	*How long do you typically spend on the computer most days?*					
Do you experience eye strain when using a computer?	8 hours or more	Between 2-4 hours	Between 4-6 hours	Between 6-8 hours	Less than 2 hours	Grand Total
No	46.15%		15.38%	30.77%	7.69%	100.00%
Yes	15.38%	23.08%	15.38%	38.46%	7.69%	100.00%
Grand Total	**30.77%**	**11.54%**	**15.38%**	**34.62%**	**7.69%**	**100.00%**

Figure 5 Eyestrain and time spent on computers

Breaks

15.38% of respondents that always took breaks did not experience eye strain. 46.15% of users that reported eyestrain sometimes took breaks. Over half at 53.85% respondents that did not report symptoms took breaks rarely. This shows that users that experienced eye strain took breaks more often. The group that always took breaks did not report eyestrain.

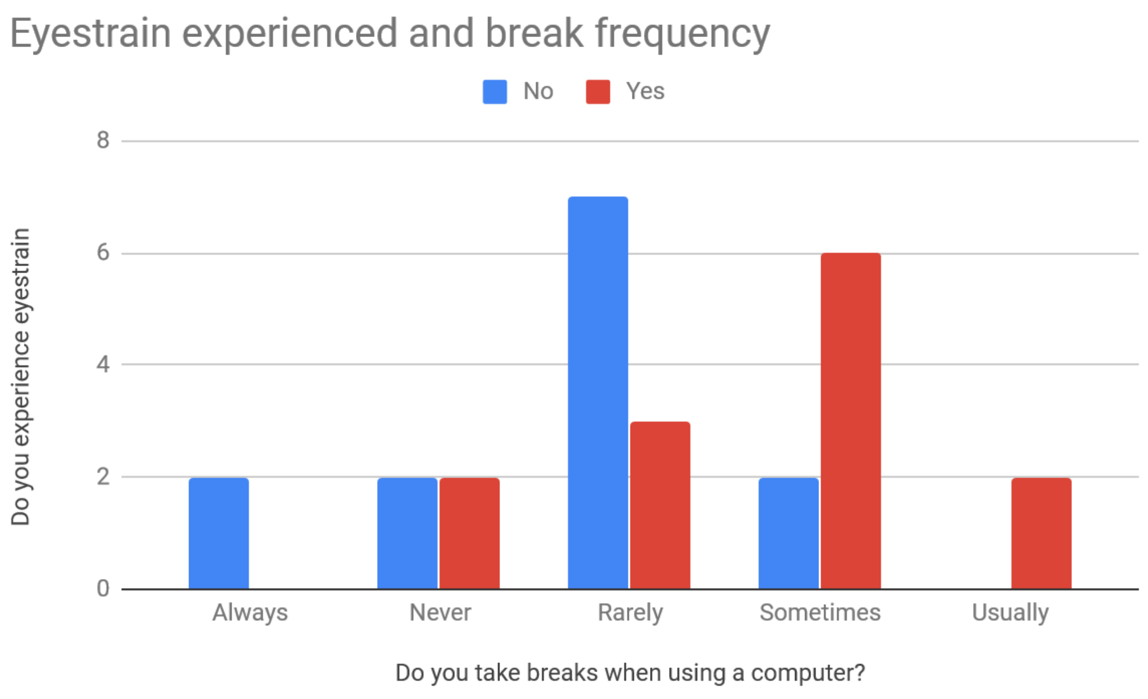


Figure 6 Eyestrain frequency bar chart

COUNTA of Do you experience eye strain when using a computer?	*Do you take breaks at least every hour when using a computer?*				
Do you experience eye strain when using a computer?	Always	Never	Rarely	Sometimes	Usually
No	15.38%	15.38%	53.85%	15.38%	
Yes		15.38%	23.08%	46.15%	15.38%
Grand Total	**7.69%**	**15.38%**	**38.46%**	**30.77%**	**7.69%**

Figure 7 Eye strain and frequency of breaks

Display

Type of display, resolution and frequency was asked but the same types were used by a majority of respondents or the user lacked the knowledge of their setup to answer the question, responding with don't know. Again, in the Seegehalli paper it has been suggested that higher resolutions are better for eye strain however most users had the same 1920x1080 displays or specification close to that so there was not enough data to draw any meaningful conclusions. Respondents were also asked about the refresh rate of their monitors to see if that made a difference however 60Hz was the most prevalent at 34.6% and 61.5% said that they did not know.

30.8% of respondents said that they used an LCD monitor and 34.6% used LED, with the same percentage responding that they did not know. As previously mentioned, CRT has a flickering effect however as this is obsolete technology none of the respondents reported using this type of VDT. The survey did not specify the different types of LCD display technologies (TN, IPS, VA). This could have been specified further with logic that asked respondents that replied LCD and then asked them to specify further. There may be some confusion as to the type of monitor owned as some LCD panel monitors are sold as LED as they have an LED backlight. The survey did not take into account that some users may be using dual monitor setups or multiple devices with mixed panel types.

Law and Ethics

In the UK workers mean working hours is calculated over a reference period which is usually 17 weeks. Although there are exceptions for some jobs the maximum average working hours is 48 hours a week but as this is a mean working hours can vary as long as the mean remains equal to or less than 48 hours. Some jobs have reference periods that vary. If an employee always works, the average on weekdays they work 9.6 hours a day. Considering that over half of subjects in the Agarwal study experienced eyestrain from working over 6 hours this means that over half of employees could experience eye strain.

Employees may be asked by their employer to opt out of the average working week to work longer hours, but they cannot be penalised for doing so and consent can be withdrawn at any time with notice.

Employers must fund eye tests for employees that use a computer daily for work when requested in accordance with the Health and Safety Display Screen Equipment regulations 1992.

For the survey to comply with the Data Protection Act only data relevant to the project was collected, it will not be used for any other purpose and it is only stored for as long as it is needed. Personal details were not required so they were not collected, making the survey anonymous. General Data Protection Act or GDPR states that clients must opt in to having their data stored and can opt out at any time. EU GDPR.ORG defines personal data as any information relating to an identifiable person who can be identified directly or indirectly (EU GDPR.ORG, n.d.). No personal identifiable details are stored when conducting the surveys making the research GDPR compliant.

Reflective evaluation

The aim of the project was originally to create an interface to control Xrandr that would adjust the brightness based on the time of day. This was too narrow for a final year project and from reading that luminance should be changed according to illuminance it was instead changed to a brightness sensor.

Limitations and further development

Although the project was mainly intended to be for Linux, as that platform has a smaller user base early in development it had some functionality to detect the operating system and be cross platform to run on Windows as well as that has the largest user base. This was abandoned as shown in the setbacks section, the shell commands could not be executed successfully.

There were problems getting a list of user ports and video outputs and they have to be hardcoded in the prototype. These were originally meant to be shown in a drop down menu.

A way of reducing blue light was found by reducing the blue gamma although no specific data was found on what they should be. Changing the colour gamma settings often made text unreadable.

A problem with automatic brightness is if using the computer in a completely dark room the brightness would be set to zero making the screen completely unreadable. This can be solved by raising the brightness bias slider. A more permanent solution would have been to introduce a higher lower limit.

One obstacle that was overcome was preventing the program from changing the brightness to out of range.

Every time a change was made in the program a previous version has been kept. If this was done again the versioning feature in Netbeans would be used and a separate set of versions would be created for only when new features were added or removed, as many versions have the same functionality but with bug fixes. Basic version control was used by numbering filenames. Bug fixes would also have been logged.

The program is only effective at limiting problems arising from monitor settings. It is not effective at easing problems associated with eye muscles including convergence insufficiency.

Evaluation of testing

The age groups and gender of participants was not queried which could have revealed some findings about which demographics are most affected by eyestrain. As the product required hardware, it was not possible to find a testing group as it could not be easily distributed and required specialist skills to setup. Only single user performance testing and code review was done over the course of development. The next step in testing project would be to conduct a pilot study with test users for functionality, bug testing and effectiveness.

Skills developed

Further knowledge of Java was developed in writing multi threaded code, choosing a correct third party API and exploring more elements of the Swing library such as JSliders, using the Linux terminal is now better understood. Basic circuit designed skills and Arduino programming in the C script were developed which was done quite easily due to prior experience in Java.

References

Agarwal, S., Goel, D. & Sharma, A., 2013. Evaluation of the factors which contribute to the ocular complaints in computer users. *Journal of clinical and diagnostic research: JCDR,* 7(2), p. 334.

arch linux, n.d. *Window Manager.* [Online]
Available at: https://wiki.archlinux.org/index.php/window_manager
[Accessed 22 March 2019].

Berger, R. W., n.d. *Why do images appear darker on some displays? An explanation of monitor gamma.* [Online]
Available at:
http://web.mit.edu/jmorzins/www/gamma/rwb/gamma.html
[Accessed 7 March 2019].

Bhanderi, D. J., Choudhary, S. & Doshi, V. G., 2008. A community-based study off asthenopia in computer operators. *Indian J Ophthalmol,* 56(1), pp. 51-55.

Chamberlain, K. et al., 2011. Exposure to Room Light before Bedtime Suppresses Melatonin Onset and Shortens Melatonin Duration in Humans. *The Journal of Clinical Endocrinology & Metabolism,* 96(3), pp. E463-E472.

Consultancy, 2017. *UK smartphone penetration continues to rise to 85% of adult population.* [Online]
Available at: https://www.consultancy.uk/news/14113/uk-smartphone-penetration-continues-to-rise-to-85-of-adult-population
[Accessed 7 March 2019].

EU GDPR.ORG, n.d. *GDPR FAQs.* [Online]
Available at: https://eugdpr.org/the-regulation/gdpr-faqs/
[Accessed 7 March 2019].

Fletcher, A. et al., 2008. Sunlight Exposure, Antioxidants, and Age-Related Macular Degeneration. *Arch Ophthamol,* 126(10).

Hale, B., 2018. *TN vs IPS vs VA: Which is the Best Monitor Display for Gaming?.* [Online]
Available at: https://techguided.com/display-panel-types-tn-ips-va/
[Accessed 13 March 2019].

IBM Knowledge Center, n.d. *The JIT compiler.* [Online]
Available at:
https://www.ibm.com/support/knowledgecenter/en/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/jit_overview.html
[Accessed 5 March 2019].

Isono, H. et al., 2013. *The effect of blue light on visual fatigue when reading on LED-backlit tablet LCDs,* Tokyo: International Display Workshops 2013.

Kanitkar, K., Carlson, A. N. & Y, R., 2005. *Ocular Problems Associated With Computer Use.* [Online]
Available at: https://www.reviewofophthalmology.com/article/ocular-problems-associated-with-computer-use
[Accessed 13 March 2019].

Lineback, N., n.d. *X11 - X Windowing System.* [Online]
Available at: http://toastytech.com/guis/remotex11.html
[Accessed 11 March 2019].

Lin, J. B., Gerratt, B. W., Bassi, C. J. & Apte, R. S., 2017. Short-Wavelength Light-Blocking Eyeglasses Attenuate Symptoms of Eye Fatigue. *Investigative ophthalmology & visual science,* 58(1), pp. 442-447.

Portello, J. K. & Rosenfield, M., 2010. Effect of Blink Rate on Computer Vision Syndrome. *Investigative Ophthalmology & Visual Sciences,* 51(13), p. 950.

Royal National Institute of Blind People, 2017. *Can blue light cause AMD?.* [Online]
Available at: https://www.rnib.org.uk/nb-online/blue-light-amd
[Accessed 13 March 2019].

Seegehalli, P. J., 2016. Digital eye strain reduction techniques: a review. *International journal on computer science and engineering,* 8(3), pp. 94-95.

Sheppard, A. L. & Wolffsohn, J. S., 2018. Digital eye strain: prevalence, measurement and amelioration. *BMJ Open Ophthalmology.*

Shieh, K. & Lin, C., 2000. Effects of screen type, ambient illumination, and color combination on vdt visual performance and subjective preference. *International Journal of Industrial Ergonomics,* 26(5), pp. 527-536.

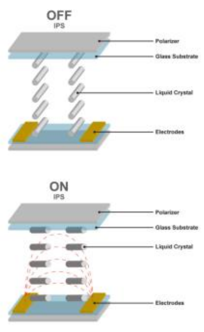
Shields, I., 2019. *Learn Linux 101: Install and configure X11.* [Online]
Available at: https://developer.ibm.com/tutorials/l-lpic1-106-1/
[Accessed 22 March 2019].

Shih, Y.-H., Chang, H.-Y., Lu, M.-I. & Hurng, B.-S., 2014. Time trend of prevalence of self-reported cataract and its association with prolonged sitting in Taiwan from 2001 and 2013. *BMC Ophthamol,* 14(128).

Specsavers, n.d. *Computer eye strain.* [Online]
Available at: https://www.specsavers.co.uk/eye-health/computer-eye-strain-symptoms-and-solutions
[Accessed 24 January 2019].

TN Panel, n.d. *TN Vs. IPS Vs, VA.* [Online]
Available at: https://www.tnpanel.com/tn-vs-ips-va/
[Accessed 8 March 2019].

U.S. Department of Energy, 2012. *Analysis of room illuminance and televisions with automatic brightness control: energy efficiency program for consumer products: television sets,* Washington: U.S. Department of Energy.

Image	URL
 OFF IPS Polarizer Glass Substrate Liquid Crystal Electrodes ON IPS Polarizer Glass Substrate Liquid Crystal Electrodes	https://images.pcworld.com/images/article/2012/03/fig-4-ips-11336066.jpg