|  |
| --- |
| East Riding College |
| Software Engineering with HCI |
| Assignment 1 -Program Development |

|  |
| --- |
| Kai Tindall  Thursday, February 07, 2019 |

Contents

[1 Contents 1](#_Toc536783046)

[2 Table of Figures 3](#_Toc536783047)

[3 Introduction 4](#_Toc536783048)

[4 Task One 4](#_Toc536783049)

[4.1 Perception in HCI 4](#_Toc536783050)

[4.1.1 What is it? 4](#_Toc536783051)

[4.1.2 Colour 4](#_Toc536783052)

[4.1.2.1 Colour Theory 5](#_Toc536783053)

[4.1.2.2 Colour in Culture 7](#_Toc536783054)

[4.1.3 Pop-out and Patterns 8](#_Toc536783055)

[4.1.3.1 Pop-out 8](#_Toc536783056)

[4.1.3.2 Pre-attentive Processes 8](#_Toc536783057)

[4.1.3.3 Patterns 9](#_Toc536783058)

[4.1.3.4 Illusions 9](#_Toc536783059)

[4.2 Different Behaviour Models 10](#_Toc536783060)

[4.2.1 Predictive Models 10](#_Toc536783061)

[4.2.1.1 Hick-Hyman Law 10](#_Toc536783062)

[4.2.1.2 Keystroke-level Model 11](#_Toc536783063)

[4.2.1.3 Throughput 11](#_Toc536783064)

[4.2.1.4 Fitts’ Law 11](#_Toc536783065)

[4.2.2 Descriptive Models 12](#_Toc536783066)

[4.2.2.1 Key Action Model 12](#_Toc536783067)

[4.2.2.2 Buxton’s Three State Model 12](#_Toc536783068)

[4.2.2.3 Guiard’s Model 12](#_Toc536783069)

[4.3 Information Processing Including Human Information Processing and GOMS 13](#_Toc536783070)

[4.3.1 Human Information Processing 13](#_Toc536783071)

[4.3.2 GOMS 13](#_Toc536783072)

[4.4 The Application of HCI Principles for Specialist Uses 14](#_Toc536783073)

[4.4.1 Visual Disabilities 14](#_Toc536783074)

[4.4.2 Motor Disabilities 14](#_Toc536783075)

[4.4.3 Auditory Disabilities 14](#_Toc536783076)

[4.4.4 Cognitive Disabilities 14](#_Toc536783077)

[4.5 Health and Safety Considerations 15](#_Toc536783078)

[5 Task Two 15](#_Toc536783079)

[5.1 Explanation 15](#_Toc536783080)

[5.2 Choosing a HCI Within my Workplace 15](#_Toc536783081)

[5.3 Case Study 16](#_Toc536783082)

[5.3.1 FUNC\_1 16](#_Toc536783083)

[5.3.2 FUNC\_2 17](#_Toc536783084)

[5.3.3 FUNC\_3 17](#_Toc536783085)

[5.3.4 FUNC\_4 18](#_Toc536783086)

[5.3.5 HLTH\_1 19](#_Toc536783087)

[5.3.6 HLTH\_2 22](#_Toc536783088)

[5.4 Conclusion 23](#_Toc536783089)

[6 Works Cited 24](#_Toc536783090)

[7 Appendices 26](#_Toc536783091)

# Table of Figures

[Figure 1: Ling's cars website by Ling Valentine, Ling's cars, n.d. 4](#_Toc536783026)

[Figure 2: The Colour Wheel, by TigerColor 4](#_Toc536783027)

[Figure 3: Different Colour Schemes on the Colour Wheel, by Mihir Patkar 2014 5](#_Toc536783028)

[Figure 4: Warm and Cool Colours, by Mihir Patkar, 2014 6](#_Toc536783029)

[Figure 5: Orientation, Line Length / Width, Closure, by Albustin et al. 2010 7](#_Toc536783030)

[Figure 6: Types of Patterns, by Drew D, 2012 8](#_Toc536783031)

[Figure 7: Using 2D Shapes to Create 3D Shapes, by Drew D, 2012 9](#_Toc536783032)

[Figure 8: Graph of Hick-Hyman Law with Common Values of a and b 10](#_Toc536783033)

[Figure 9: Buxton's Three State Model Example of a Stylus, by William Buxton, 1990 11](#_Toc536783034)

[Figure 10: Guiard's Model of Bimanual Control 12](#_Toc536783035)

[Figure 11: Explanation of GOMS Terms 13](#_Toc536783036)

[Figure 12: Needs for the User-base 15](#_Toc536783037)

[Figure 13: Main Menu for Team Explorer 15](#_Toc536783038)

[Figure 14: Visual Studio Highlights Which File You're Currently Working On 16](#_Toc536783039)

[Figure 15: Visual Studio Pops Out The Line You're Currently Working On 17](#_Toc536783040)

[Figure 16: Automated Test Report Generated From Visual Studio 18](#_Toc536783041)

[Figure 17: Visual Studio with Text at 100% 19](#_Toc536783042)

[Figure 18: Visual Studio with Text at 195% 20](#_Toc536783043)

[Figure 19: Visual Studio with Light Theme 21](#_Toc536783044)

[Figure 20: Intellisense Predicting a Variable Name 22](#_Toc536783045)

# Introduction

In this document I shall be tackling the two distinct tasks outlined in the assignment brief. Task one being to talk about the principles of interface design, with task two being to justify selection of a HCI for a particular user group or environment.

# Task One

## Perception in HCI

### What is it?

“It’s what allows us to see and feel when using HCI” (D, 2012). What does that mean? Well it means it’s the colours, patterns, and objects that HCI uses to try and convey intuitivism. There are multiple methods and paradigms to design that alter the user’s perception of a HCI to try and improve that sense of innate knowledge about the product that sets the user at ease when using the HCI for the first time.

### Colour

Colour is an important factor in a HCI, they can convey emotion, be used to tap into cultural and biological mechanisms to influence the user. But you’ve got to know when and how to use them, some colours go well together and look really nice to the end user, making them want to continue using the product.

However when done wrong colours can completely destroy the user’s view on the product, quite literally, take figure 1 for example; figure 1. (Ling Valentine n.d.) is a perfect example of using colour completely wrong. There are too many colours and none of them go with each other. They take user’s attention away from where it needs to be to everywhere else, and the user then has to really look for the options to proceed.

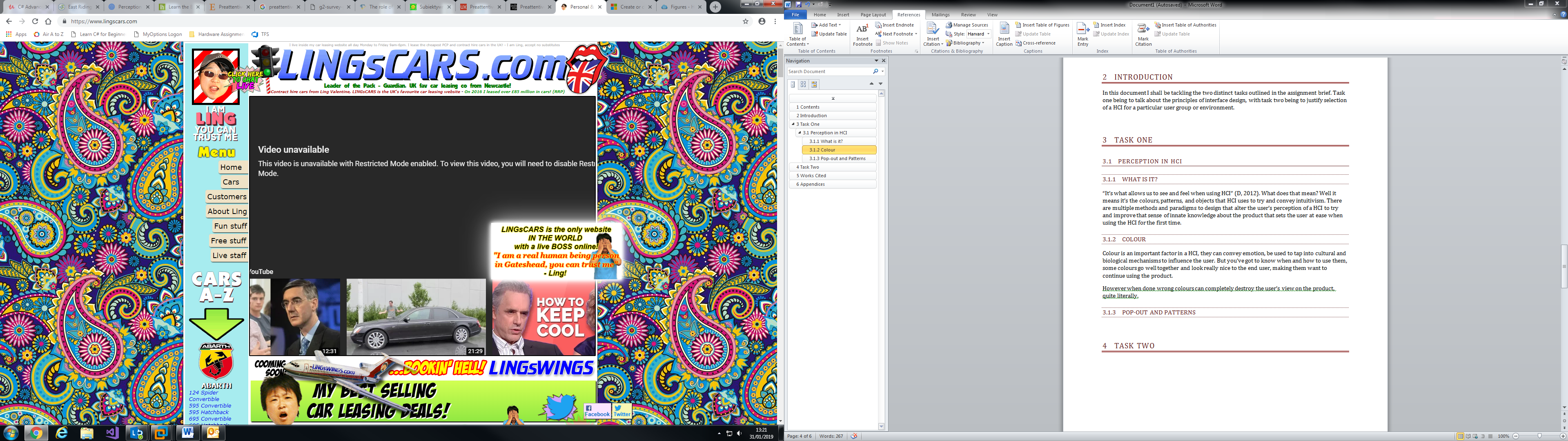


Figure : Ling's cars website by Ling Valentine, Ling's cars, n.d.

Why does this happen? Well it’s to do with evolutionary benefits. Albustin speaks about how “Finding the cherries among the leaves is much easier if we have colour vision” (Albustin, et al., 2010), hence why colour is so important for humans. There seems to be specific rules too which we can follow which humans seem to like. This area of study is called colour theory.

#### Colour Theory

Colour theory focuses around a concept called the “colour wheel”, you can see an example of the colour wheel in Figure 2 (TigerColor). Created by Sir Issac Newton in 1666, it’s designed so that any colours you pick from it will look good together.

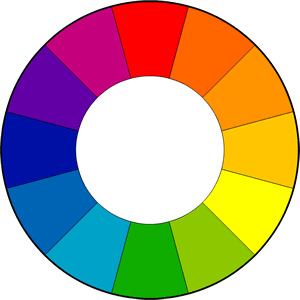


Figure : The Colour Wheel, by TigerColor

There are multiple different colour schemes to take from this colour wheel, Figure 3 (Mihir Patkar 2014) shows off the most popular.

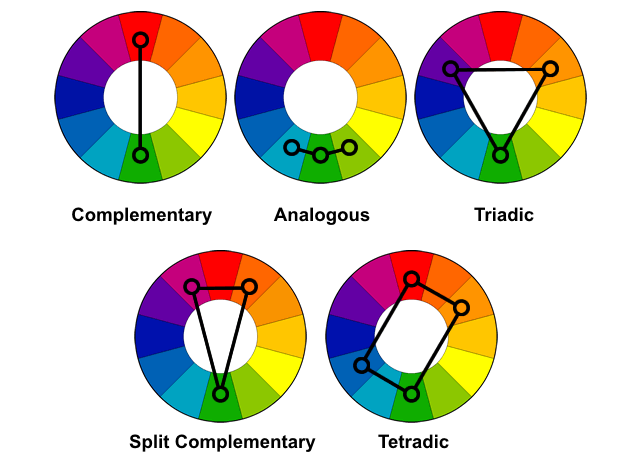


Figure : Different Colour Schemes on the Colour Wheel, by Mihir Patkar 2014

Complementary – These are two colours that are completely opposite in the colour wheel. These create very high contrast to stand out. Normally one colour is used as a background and the other is used as an accent colour.

Analogous – Any three colours that are next to each other are analogous, with these colour schemes it’s normally a good idea to uses tints instead of pure hues as they can be jarring. Also avoid mixing warm and cool colours. Warm and cool colours are shown in Figure 4 (Mihir Patkar 2014).

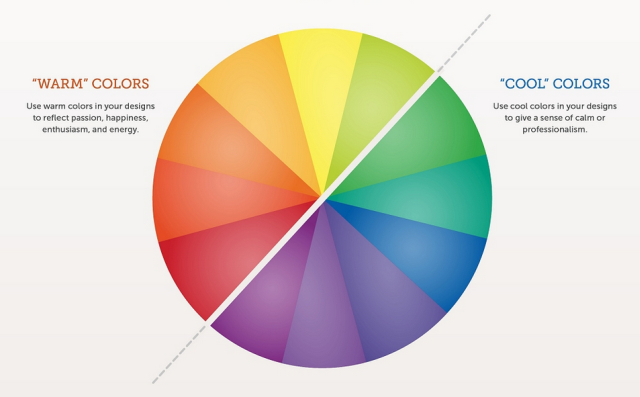


Figure : Warm and Cool Colours, by Mihir Patkar, 2014

Triadic – Any three colours, which are equally spaced apart on the colour wheel, are triadic. Triadic schemes are also very high contrast, and it normally used by letting one colour dominate and using the other two as accents.

Split Complementary – This scheme uses three colours, it takes one colour and then matches it with the two colours next to its complimentary colour. This scheme is great for those just starting out because it’s difficult to mess up because you get contrasting colours that aren’t as diametrically opposite as complementary colours.

Tetradic – Uses four colours in the form of two sets of complementary colours. This is the hardest scheme to balance. It offers more variety than any other scheme but if all colours are used in equal amounts then the scheme will not look balanced.

#### Colour in Culture

Colours mean different things in different cultures, so when creating HCI it’s important to take into consideration where it’s going to be used, and if anything needs to be altered for localisation purposes.

For example, Stewart (Stewart, n.d.) talks about how “Red symbolizes love and passion in many countries in North and South America, and Europe.”, and then goes on to contrast that with a point of “Red in African cultures symbolises death and grief.”. Important differences that need to be understood when creating a HCI that has the potential to be used outside of the country it was created in.

### Pop-out and Patterns

#### Pop-out

What is pop-out? Well it’s very simple really, it’s using techniques to create elements that grab your attention on the HCI, there are a few methods that you can use to do this, and they all utilise your pre-attentive processes, or in other words, what you see before you even think about it.

#### Pre-attentive Processes

##### Categories

Albustin (Albustin, et al., 2010) outlines the four categories of pre-attentive processing as: Colour; Form; Spatial position; and Motion. We can incorporate elements of these different categories in our HCI to direct the user’s attention to central points within our design. For instance if your webpage has a submit button where it’s essential it’s pressed, you could utilise any of these methods to make it stand out to be sure the user doesn’t miss it.

“Colour seems to be the most important factor when it comes to pre-attentive processing” (Albustin, et al., 2010). This should mean it’s the most effective and important to get right, so before trying to tackle any other of the categories, one should try and master colour first.

Form is very simple but also very powerful. Essentially it’s just how an object looks; if it looks different from all the objects around it, it commands attention.

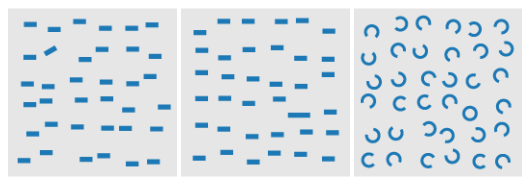


Figure : Orientation, Line Length / Width, Closure, by Albustin et al. 2010

Spatial position is where the element is, this includes all axes, spatial and rotational. So for example you may have three elements that have the same form, but one of them is in a different spatial position to the other two, that would be the element that demands the attention.

And finally, motion is the fourth pre-attentive category. Mainly used for animation to “highlight changes in a dataset across user-selected data axis” (Albustin, et al., 2010) motion is very intuitively processes as we deal with it every day and expect certain properties about it.

##### The Role of Pre-attentive Processes

The role of pre-attentive processes in HCI is definitely an important one, as confirmed by (Michalski & Grobelny, 2008) where they mention “The presented studies confirm that the pre-attentive visual processing mechanisms may play an important role and should be explicitly incorporated into the HCI field”. This shows that pre-attentive processes definitely have a part to play in intuitive HCI and should therefore be utilised wherever possible.

Its specific role though is to attract people’s attention and form natural, intuitive links the user can understand and use very easily.

#### Patterns

Patterns are often used as a good way to indicate users that certain things have similar functionality without explicitly telling them. Using similarity, proximity, closure, and continuation we can group objects together without having to physically tell the user these objects all have similar functionality. Figure 6 (Drew D, 2012) explains and gives examples of these concepts.

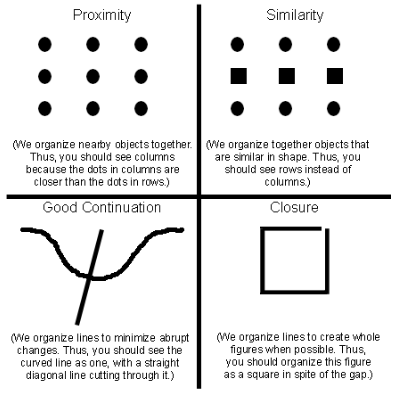


Figure : Types of Patterns, by Drew D, 2012

#### Illusions

Illusions are elements such as using 2D shapes to create the illusion of it being 3D. as you can see in Figure 7 (Drew D, 2012), this is quite effective and the images almost look real. Using this technique to create icons, symbols, etc. is very powerful as it lets the user combine real world common knowledge to the HCI, which will dramatically decrease learning time as they will already have an innate understanding of what the concept of the symbols may mean.

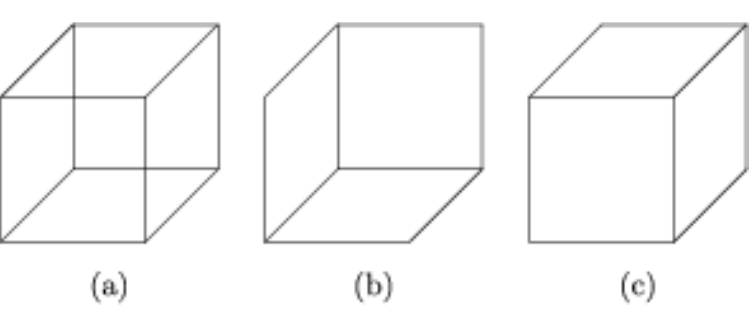


Figure : Using 2D Shapes to Create 3D Shapes, by Drew D, 2012

## Different Behaviour Models

In HCI, there are different models, sorted into the predictive and descriptive sets, which we can use to model human behaviour around the interfaces. We can then use this to alter the design of the interface to make it more user friendly.

### Predictive Models

Predictive models are used in HCI to predict how a user will react to an interface, without having to conduct research, which would potentially postpone a release day. This is confirmed by Mackenzie (MacKenzie, 2003) when he states “In human-computer interaction, predictive models allows metrics of human performance to be determined analytically without undertaking time-consuming and resource-intensive experiments”.

#### Hick-Hyman Law

The Hick-Hyman law relates to the reaction time of users, when presented with differing amounts of stimuli. This law gives the mathematical equation of:

Where and are constants defined by values determined empirically, is reaction time, and is the amount of stimuli. With common values of and , figure 8 is a graphical representation of the Hick-Hyman law.

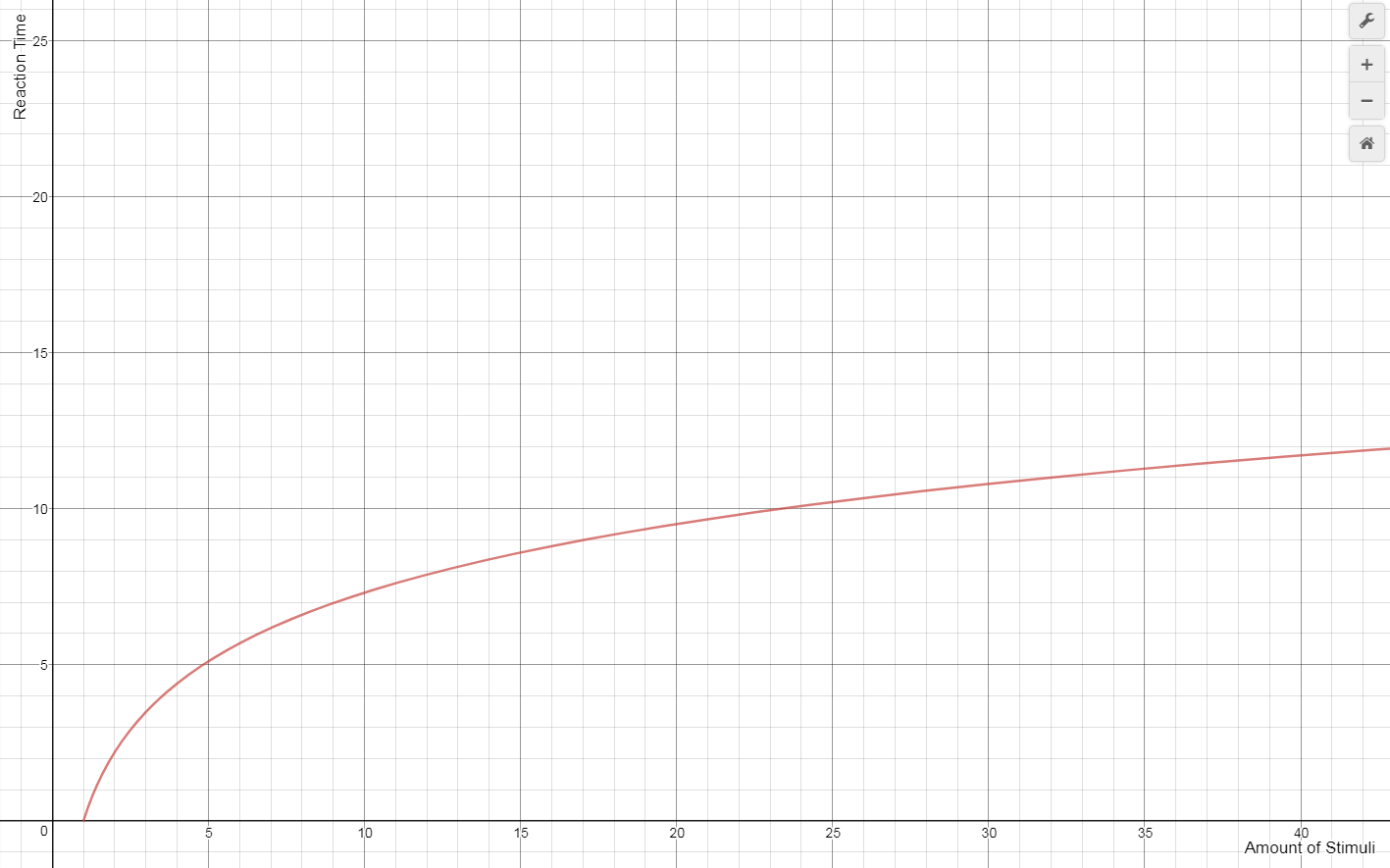


Figure : Graph of Hick-Hyman Law with Common Values of a and b

This law is a very good yard-stick at judging whether or not your menu is too deep. It’s been proven using this law that “breadth should be favoured over depth in hierarchical menus” (MacKenzie, 2003).

#### Keystroke-level Model

The Keystroke-level model focuses on low level actions such as pressing a key on a keyboard, moving the mouse, clicking a button, etc. First described by Card, Moran, and Newell in the early 1980s, it’s main goal is to determine how long users actually take to use the HCI through hardware.

The model attempts to predict the time a user will take to reach an outcome based on which actions they must complete to get there. A great description of the method can be found [here](http://www.cs.loyola.edu/~lawrie/CS774/S06/homework/klm.pdf) (Kieras, 2001)

#### Throughput

Throughput is mainly linked to the output that is being given by the system. Used to measure speed that the HCI takes to process a command made by the user, the value of this can vary dramatically when given long-winded or sophisticated tasks to complete.

This is a very good measure for adequacy of the system’s physical specifications for the task as it is “showing you the computers processing speed and allowing you to determine yourself if it needs to be upgraded or not” (RyanGlover, 2013).

#### Fitts’ Law

In short terms, Fitts’ law “is used to predict how quick a user will be able to move to a specific area based on how far away it is, and how large the object you are trying to reach is” (RyanGlover, 2013). How this is done is by using the mathematical formula known as Fitts’ Law, defined as:

Where is movement time, and are empirically determined constants, is the amplitude, which is the distance of the centre of the target from the starting location, and is the target width. (Goktürk, n.d.).

### Descriptive Models

Descriptive models are just as important as predictive models; however don’t normally yield empirical data or any measure of user performance. “Simply put, descriptive models provide a framework or context for thinking about or describing a problem or situation.” (MacKenzie, 2003).

#### Key Action Model

The key action model is very simple, it models how keys on a keyboard should perform. KAM defines three categories of keys: symbol keys, executive keys, or modifier keys (MacKenzie, 2003).

Symbol keys represent symbols, such as letters, numbers, or punctuation. Executive keys are bound to a function within the system, these are keys such as ENTER, F1, or ESC. Modifier keys don’t do anything themselves but instead alter how the other keys react, examples include ALT, CTRL, and SHIFT.

#### Buxton’s Three State Model

Buxton’s three state model is a representation of graphical input devices, such as a mouse or trackpad.

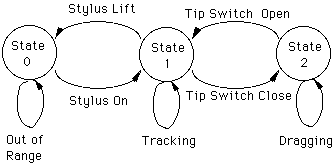


Figure : Buxton's Three State Model Example of a Stylus, by William Buxton, 1990

Figure 9 (William Buxton, 1990) shows how to change states using the stylus, users will get used to the change of states to perform specific actions, so when a new device comes along and they try to change the signature of the state changes to get to the same action, users find this challenging.

#### Guiard’s Model

Guiard’s model tries to descriptively model bimanual control, this means it looks at how the two hands interact with each other, and the different roles between the preferred and non-preferred hand.

|  |  |
| --- | --- |
| Hand | Role and Action |
| Non-Preferred | * Leads the preferred hand * Sets the spatial frame of reference for the preferred hand * Performs coarse movements |
| Preferred | * Follows the non-preferred hand * Works within established frame of reference set by the non-preferred hand * Performs fine movements |

Figure : Guiard's Model of Bimanual Control

Guiard’s model should be used by designers to make sure that people who use either hand are accommodated for, because if someone who’s right handed finds something intuitive to use but someone who’s left hand doesn’t, well then if that HCI was placed somewhere where the majority of people are left handed then that HCI is going to be very difficult to use most of the time.

## Information Processing Including Human Information Processing and GOMS

Often forgotten is the fact that humans themselves are part of the HCI (They are the H after all), This means that their own processing power needs to be taken into consideration when designing HCIs; after all, the whole point of a HCI is to process things for humans quicker than they would be able to without the HCI, if the HCI takes longer than the human then it’s pointless.

### Human Information Processing

How humans process information has been a topic of study for years, with HCI however we’re mostly concerned with how the human mind deals with memory and absorbing information.

Humans generally collect and use information in these four steps (HCI, 2011):

* Absorb the information
* Analysis the information
* Use the information
* Do something with the information

You can draw many parallels from human memory to computer storage. Where the human’s short term memory would be equivalent to the computer’s RAM, and long term memory would be the computer’s storage device (such as a hard drive).

The human mind works similarly to a computer, firstly the brain will input the data with the hardware, which is then processed using the software, the information is then stored and then used to make decisions.

### GOMS

GOMS stands for Goals, Operators, Methods, and Selection rules. It’s a model that can be used to allow users and developers to understand, see, and perform tasks that are undertaken in a HCI.

|  |  |
| --- | --- |
| Term | Definition |
| Goals | What the user wants to accomplish from the system where in order to complete a goal they need to use the HCI. |
| Operators | Actions that are physically taken and performed in order to complete a given goal |
| Methods | A set of tasks that are needed to be put in sequence in order to complete the goal. |
| Selection rules | Rules that people take in order to choose between the different methods of completing the given goal. |

Figure : Explanation of GOMS Terms

## The Application of HCI Principles for Specialist Uses

Sometimes users may have some kind of disability that will have an effect on their use of the HCI, this means that certain measures may need to be taken to make sure that people are accommodated for in the HCI they’re trying to use. I have spoken about some examples below.

### Visual Disabilities

Blind users need to use text-to-speech software when using HCIs, it’s worth noting that if there is a fair amount of text, to structure your text in a way that blind users may be able to get a good overview from hearing the sub-titles alone. Nielsen (Nielsen, 1996) goes on to explain, when using HTML tags, to use <h1>, <h2>, and <h3> tags appropriately.

It also talks about ALT attributes to provide alternative text for images, this means that if you’re blind, instead of just being told the caption, you get a description of the image so that you can try to visualise the image.

Some people aren’t completely blind too; some people just have reduced eyesight and just require text to be made clearer. To accommodate for this, HCI designers need to use clear fonts and size their text relative to the settings the user has chosen.

### Motor Disabilities

Nielson (Nielsen, 1996) also talks of users who cannot make fine movements with their hands may also need special consideration when designing HCI. Because of this, HCI design should have an option to enlarge buttons to make fine movements a lot less common and unnecessary.

### Auditory Disabilities

Deaf, or near deaf, users may also need features implemented into the HCI, such as a transcript for any video or sound content within your HCI.

### Cognitive Disabilities

Interaction Design (Interaction Design Foundation, 2017) talks about designing for the learning impaired, some general techniques include making simple, uniform, screens; this just means keeping screens simple to avoid confusion, keeping language simple, and distractions to the minimum.

They also talk about keeping lists small and in a logical order, especially if they link to other parts of the HCI. Nielsen (Nielsen, 1996) backs this up when he talks about simplified navigation of websites.

## Health and Safety Considerations

HCI can be incredibly useful in a health and safety environment for a variety of reasons. First of all, you can utilise colour and pop-out techniques to really grab people’s attention to safety issues, by having red flashing lights with loud noises it’s quite hard to not realise there is an issue, which is why HCI is sometimes very important in safety.

Secondly, computers can monitor many more elements and variables in such a faster time than humans ever could that they can detect problems in ways humans never could, which is why the HCI needs to convey effectively to the end-user these issues, because humans may sometimes miss them.

It’s important to realise though that the HCI needs to be done correctly, for example, if you’re going to use flashing lights to convey danger, you may need to think about if someone has epilepsy and if they would have an epileptic fit if those lights were to go off, rendering them helpless to escape the danger.

# Task Two

## Explanation

In task two, I will be choosing a HCI from my workplace and creating a case study surrounding it.

## Choosing a HCI Within my Workplace

The HCI I have chosen is a piece of software called Visual Studio 2017, this application is a software IDE (Integrated Development Environment) that helps software engineers write code and perform tests.

I have chosen this HCI because it’s a very common interface that myself and my colleagues work with almost every day, so with it being so common, I thought it a good HCI to really break down and take a look at.

This HCI is predominantly used by software engineers working with languages based on the .NET framework, such as C#, F#, and IronPython. The environment will also affect how they use the HCI too; in my workplace there is a heavy emphasis on teamwork, robust code, and professionalism.

This of course alters the needs of the user-base; a solitary software engineer will have different needs than a team of them, after all.

There are a number of common health issues to take into consideration when talking about software engineers, because of the hazards of the job, some people may have difficulties with their eyesight after years of looking at a computer monitor, and others may also have repetitive strain injuries (RSI), which will also need to be taken into consideration when looking at the HCI of Visual Studio.

Therefore, the main needs of the user-base are as follows:

|  |  |
| --- | --- |
| Reference | Requirement |
| Function | |
| FUNC\_1 | Shall have clean team integration |
| FUNC\_2 | Shall have expert options that increase functionality speeds |
| FUNC\_3 | Shall be clear what is being worked on |
| FUNC\_4 | Shall have clear indicators of code robustness |
| Health | |
| HLTH\_1 | Shall have options to reduce stress on the eyes |
| HLTH\_2 | Shall have options to reduce how much is typed to reduce stress for people with RSI |

Figure : Needs for the User-base

## Case Study

I believe that Visual Studio 2017’s HCI meets the needs of the user-base within my workplace. In this section I will be going through each requirement I set out in Figure 12 and justifying why it has been met.

### FUNC\_1

Visual Studio includes team integration through the “Team Explorer”, which is a menu which can connect your projects from sources such as GitHub, TFS, or Git into Visual Studio. The main menu for this team explorer looks like this:

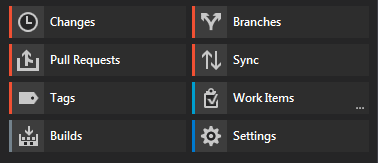


Figure : Main Menu for Team Explorer

As you can see, it combines many principles in HCI perception to create this intuitive button layout. Using parts from colour theory and patterns, the user automatically understands certain things from this interface.

Colour theory is utilised very well in this instance, first of all, the colours pop out of the black background, making it easier to see the white text and icons. Also, as this application is marketed to a heavily US and European audience, whose culture considers the colour red with feelings of danger (Cousins, 2012), I feel it only appropriate that that options relating to version control (Saving of your work) have been marked with a red accent to indicate that if not looked at, there could be severe mistakes made.

Work items and Settings have also been marked with blue accents; this can represent wisdom (ColorWheelPro, n.d.) and serenity (Stewart, n.d.). I find this works well, because when checking your work items, you want to feel calm, so you can think about solutions freely, and while changing settings, you may feel smart when you’re personalising your settings profile.

Visual Studio also makes use of patterns to allow the user to know exactly which items are grouped together, as illustrated in Figure 6: Types of Patterns, by Drew D, 2012, elements that look similar, are grouped together without us even thinking of it. So once the user processes that one button is for version control, they automatically know the other buttons marked with a red accent are too.

Visual Studio also utilises Fitts’ Law in an attempt to reduce movement time, as each button has been made wider than is necessary to try and reduce the fraction of within Fitt’s Law. This makes the buttons a bit easier to navigate to quickly, and reduces user stress.

### FUNC\_2

When professionals use products, such as the user-base in this case study, they tend to want to increase their throughput (see 4.2.1.3 for more details on throughput), one way of doing this is to implement hotkeys. Which just so happens to also be the method Visual Studio has chosen.

We can use the Keystroke-level model to estimate how much faster using a hotkey is for a certain action within Visual Studio, for this example I have chosen creating a new file in the solution.

The hotkey for creating a new file in visual studio is CTRL+N by default, this is two button clicks. Taking the method laid out by Kieras (Kieras, 2001), and taking the user to be an expert typist (they are a professional after all), this gives a sequence of “KK”, which is 0.24 seconds.

To perform this action without using a hotkey, the user must first move their cursor to the solution, then right click, add, add new file, left click. This results in a sequence of “PBBPPBB”, which is 3.7 seconds. This is over 15 times slower than using the hotkey.

Visual Studio definitely hits FUNC\_2, as there is a vast array of hotkeys outlined by visualstudioshortcuts (VisualStudioShortcuts, n.d.); which, as proven, reduce functionality time considerably.

### FUNC\_3



Figure : Visual Studio Highlights Which File You're Currently Working On

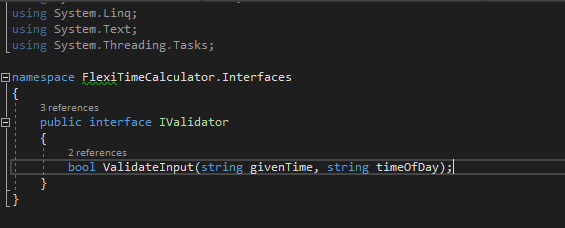


Figure : Visual Studio Pops Out The Line You're Currently Working On

Visual Studio uses colour theory throughout the application to focus the user’s attention in strategic places. One such place is allowing the user to see where they’re working, and in which file. As illustrated in Figure 14: Visual Studio Highlights Which File You're Currently Working On, the currently selected file pops against the others. This grabs your attention and you can immediately place which file you’re currently looking at.

The use of blue is very similar to what I mentioned in 5.3.1, meaning wisdom. This is obviously the kind of mood someone working in a professional environment will want.

It uses more pop-out in Figure 15: Visual Studio Pops Out The Line You're Currently Working On, where it places two bands above and below the current line the user is working on, this is to quickly centre your eyes back onto where they’re needed whenever you look away and look back again.

This uses the closure pattern illustrated within Figure 6: Types of Patterns, by Drew D, 2012, because your mind wants to see completed shapes, it will imagine the two lines being a rectangle that encases the line you’re working on, drawing more attention to it.

### FUNC\_4

Code robustness is of great importance to software engineers, which is why they need an easy way to be able to quickly see a measure of this robustness. One way of doing this is by running tests, which is facilitated within Visual Studio.

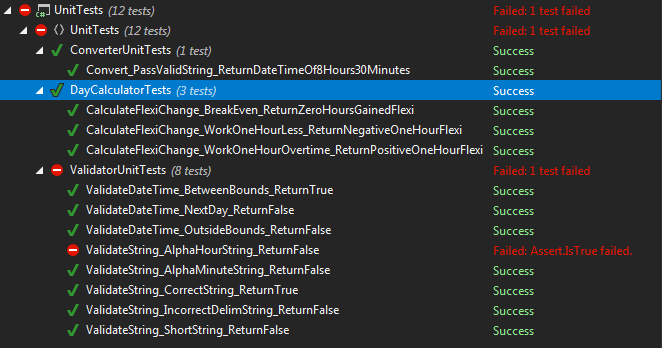


Figure : Automated Test Report Generated From Visual Studio

As you can see it’s very clear which tests have passed and which failed. They achieve this again by using colour, the green and red pop against each other so you can easily tell the difference, this is because on the colour wheel (as illustrated in Figure 2: The Colour Wheel, by TigerColor) green and red are opposite and therefore have high contrast (TigerColor, n.d.).

It’s also very intuitive what each colour means too, as I mentioned in 5.3.1 and 4.1.2.2, Visual studio was primarily developed for a US and European audience, and because of that culture, they associate different colours to different meanings. With green being good, and red being bad.

Also, because Visual Studio gives you icons next to the tests too, if a user has difficulty reading, they can still tell by the clear icons that the tick means it passed, whereas the red circle means it failed.

All in all it’s a very clean and intuitive interface to work with, and I definitely think it passes FUNC\_4.

### HLTH\_1

Visual Studio needs to accommodate for users with poor eyesight, as it is a very common issue among software engineers. Fortunately, Visual Studio combats this twofold, first of all it allows users to increase the size of the text, and allows users to change the theme of the program to allow the user to customise and pick whichever option feels the easiest on the eyes.

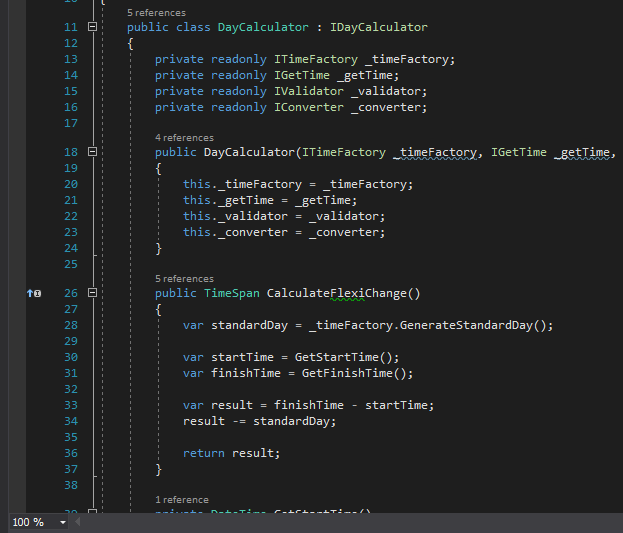


Figure : Visual Studio with Text at 100%

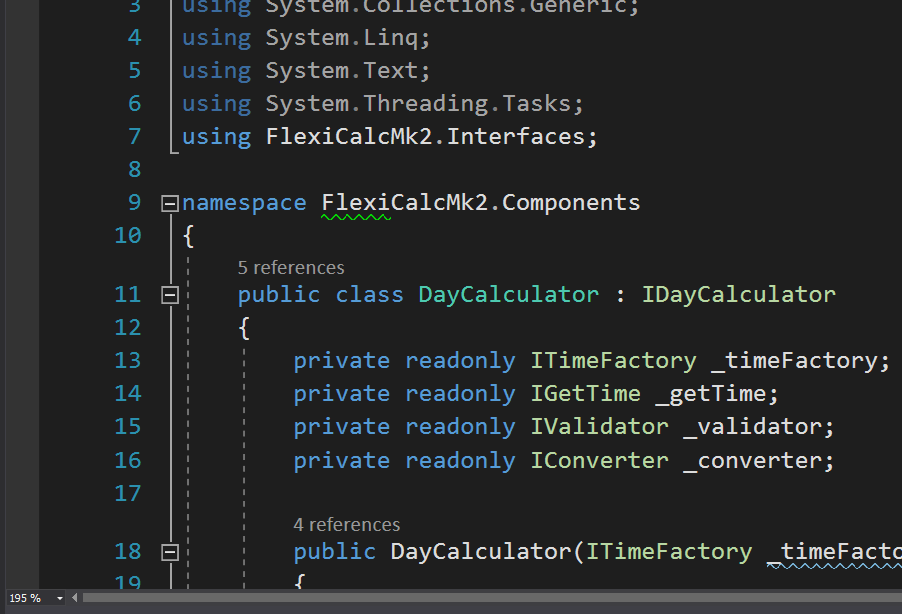


Figure : Visual Studio with Text at 195%

As you can see, just by pressing CTRL and scrolling, Visual Studio allows the user to adapt the test to suit their needs, making it easier to read.

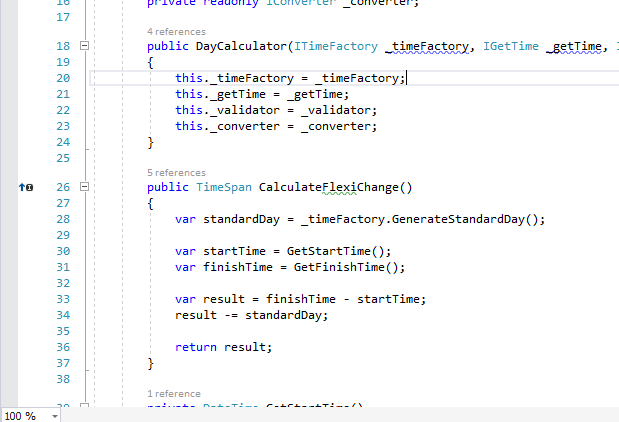


Figure : Visual Studio with Light Theme

Also, as you can see in Figure 19: Visual Studio with Light Theme, Visual Studio offers themes to customise the appearance of your working environment. Some people prefer the lighter theme over the darker one, personally I prefer dark but that doesn’t matter. People can choose what’s easiest on their eyes.

### HLTH\_2

RSI (Repetitive Strain Injury) is common amongst people that type all day; this may be because of many different reasons, bad posture when typing, not taking adequate rests every now and again, etc. This means that Visual Studio should really have some kind of feature that reduces the need to type, and it does! Intellisense is a feature of Visual Studio that tries to predict what you’re going to type and then autocomplete it for you, this means you can have things like very descriptive variable names without having to worry about mistyping the name or worrying how long it is.

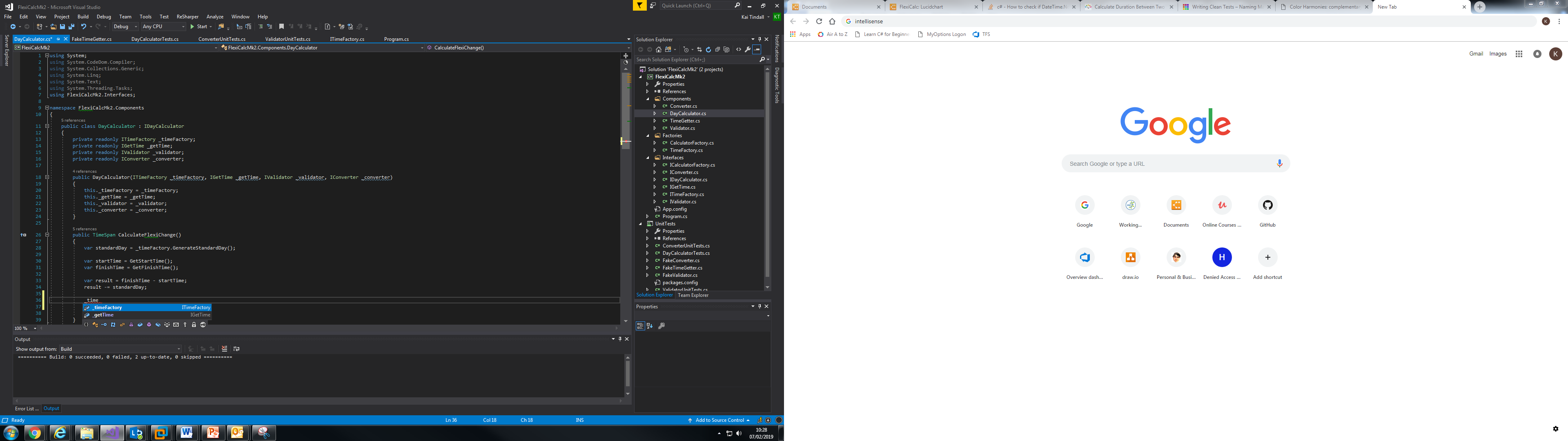


Figure : Intellisense Predicting a Variable Name

This allows users that may have a RSI to be able to type less while maintaining the same amount of code being written. This is great for them, because it means that there’s less pain involved.

## Conclusion

In conclusion, I believe Visual Studio 2017 to have an adequate HCI for the user-base within my workplace, for the reasons I have listed above.

# Works Cited

Albustin, A., Bacheitner, S., Djerdjizi, A. & Hollerit, B., 2010. *Pre-Attentive Processing,* s.l.: s.n.

ColorWheelPro, n.d. *Colour Meaning.* [Online]   
Available at: http://www.color-wheel-pro.com/color-meaning.html  
[Accessed 6 February 2019].

Cousins, C., 2012. *Colour and Cultural Design Considerations.* [Online]   
Available at: https://www.webdesignerdepot.com/2012/06/color-and-cultural-design-considerations/  
[Accessed 6 Feburary 2019].

D, D., 2012. *Perception within HCI.* [Online]   
Available at: https://prezi.com/k99gxbgkldi-/perception-within-hci/  
[Accessed 31 January 2019].

Goktürk, M., n.d. *The Glossary of Human Computer Interaction.* [Online]   
Available at: https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/fitts-s-law  
[Accessed 5 February 2019].

HCI, 2011. *Information Processing.* [Online]   
Available at: http://blog2.robtroster.com/2011/05/information-processing\_11.html  
[Accessed 5 February 2019].

Interaction Design Foundation, 2017. *Key Consideration for User Experience for the Disabled.* [Online]   
Available at: https://www.interaction-design.org/literature/article/key-considerations-for-user-experience-for-the-disabled  
[Accessed 5 February 2019].

Kieras, D., 2001. *Using the Keystroke-Level Model to Estimate Execution Times.* Michigan: s.n.

MacKenzie, S. I., 2003. *Motor Behaviour Models for Human-Computer Interaction.* San Francisco: Morgan Kaufmann.

Michalski, R. & Grobelny, J., 2008. The Role of Colour Preattentive Processing in Human-Computer Interaction Task Effency. *International Journal of Industrial Ergonomics,* I(38), pp. 321-332.

Nielsen, J., 1996. *Accessible Design for Users With Disabilities.* [Online]   
Available at: https://www.nngroup.com/articles/accessible-design-for-users-with-disabilities/  
[Accessed 5 February 2019].

RyanGlover, 2013. *Fundamental Principles of HCI Design.* [Online]   
Available at: http://principlesofhci.blogspot.com/  
[Accessed 5 February 2019].

Stewart, S., n.d. *Color Meanings in Different Cultures.* [Online]   
Available at: https://study.com/academy/lesson/color-meanings-in-different-cultures.html#/lesson  
[Accessed 1 February 2019].

TigerColor, n.d. *Color Harmonies.* [Online]   
Available at: http://www.tigercolor.com/color-lab/color-theory/color-harmonies.htm  
[Accessed 7 February 2019].

VisualStudioShortcuts, n.d. *Visual Studio 2017 Keyboard Shortcuts: The Complete List.* [Online]   
Available at: http://visualstudioshortcuts.com/2017/  
[Accessed 6 February 2019].