Interface Programming

24678805@edgehill.ac.uk

Coursework one

CALLUM MCLAUGHLIN

2022

Contents

[Task 1 2](#_Toc98365763)

[What is Human Computer Interaction? 2](#_Toc98365764)

[Chronology of HCI 2](#_Toc98365765)

[1980s – 1990s 2](#_Toc98365766)

[1990s – 2000s 2](#_Toc98365767)

[2000s – 2010s 2](#_Toc98365768)

[What is Perceptual User Interface? 3](#_Toc98365769)

[Techniques and Methods of PUI 3](#_Toc98365770)

[Feature Tracking 3](#_Toc98365771)

[Gesture Input 3](#_Toc98365772)

[Voice Commands 4](#_Toc98365773)

[Task 2 4](#_Toc98365774)

[Task 3 6](#_Toc98365775)

[Palm Detection Code 7](#_Toc98365776)

[Smile Detection Code 8](#_Toc98365777)

[Hand Detection Code 9](#_Toc98365778)

[Fist Detection Code 10](#_Toc98365779)

[Frontal Face Detection Code 10](#_Toc98365780)

[Task 5 11](#_Toc98365781)

[Introduction 11](#_Toc98365782)

[HCI Guidelines 11](#_Toc98365783)

[Innate and learned Gestures 12](#_Toc98365784)

[Static, Dynamic and Continuous gestures 12](#_Toc98365785)

[Reliability 12](#_Toc98365786)

[Design Appropriately for target audience 12](#_Toc98365787)

[Determine user intent and engagement 12](#_Toc98365788)

[Design for variability of input 12](#_Toc98365789)

[Iterate 13](#_Toc98365790)

[Nielsen's Heuristics 13](#_Toc98365791)

[Bibliography 14](#_Toc98365792)

# Task 1

## What is Human Computer Interaction?

Human Computer Interaction or HCI, is the study of how users interact with computing systems for human use and to what extent computers are or are not developed successfully for human interaction (CS.Bham, 2022).

HCI is important as a field of study and is essential when developing new technologies. Typical frustrations like not being able to figure out how to perform simple tasks on a system or device make HCI such an important part of technological developments. These problems and systems are studied, and solution are made to aid ease of use for the user.

## Chronology of HCI

HCI as an area of research and practice emerged in the early 1980s, initially as a speciality area in computer science (Carroll, 2021).

### 1980s – 1990s

During the 1980s HCI was focused on creating systems that were easy to learn and easy to use. Even back then there were endless possibilities for personal computing, but desktop computers were not very well designed and accessible to everyone at first.

The Desktop-Folder metaphor was part of a large effort to apply what are called “Mental Models” to the way we use computers. By transferring our physical office environment and eventually, larger world onto computer interfaces, we can more easily grasp how information is stored on desktops (X, 2017).

### 1990s – 2000s

In the 1990s, computer science students and the companies that were hiring them demanded HCI courses in University Curricula. As a result, Several major computer science departments have designated HCI research as a focus (Carroll, 2001).

During this era, the focus shifted from cognitive modelling to interaction design. As computers became communication tools, mental models could no longer explain the broad context of computer use. It became essential to explore external influences on how computers were used and to look at how interactions varied across different tools, software and organisations (X, 2017).

Email gained popularity during this time, meaning people weren’t just interacting with computers anymore, they were communicating with each other through them. There was a vastly growing interest in how computers were used to support communication and collaboration, signalling the rise of social and organisational computing.

### 2000s – 2010s

At this time, Apple came out with an innovative new interface in 2001 which was a wheel on their iPod music player. Throughout the early 2000s new mouse designs morphed them into all kinds of shapes and sizes. Keyboards were being redesigned with physical waves and ripples and functional shortcut keys were added. In general, HCI was driven by a need to make personal interaction with computers more ergonomic and efficient (DeWitt, 2019).

The launch of the iPhone in 2007 popularised touch as a new way of thinking about HCI. Interaction now needed to be multifaceted and fast, natural and personal. Touchscreens started becoming more common for personal devices.

We see ideas from each era of HCI influencing the work UI/Interface designers do today, these ideas co-exist, providing us with a rich knowledge to draw upon. It will be amazing to see how HCI evolves in the new decade and to see how the design evolves with it.

## What is Perceptual User Interface?

Perceptual User Interface or PUI, is an interface that allows a computer to “perceive”, interpret and respond approximately to the facial expressions, speech, gestures, movements and other perceptually based patterns of communication typical of the user (American Psychological Association, 2022).

The intent of PUI is to provide realistic, interactive encounters like those experience among people in the real world.

## Techniques and Methods of PUI

### Feature Tracking

Face Tracking detects and can also track the presence of a human face in a video. This technology can be incorporated into computer and mobile applications. Face Tracking Technology can be used online or offline.

Feature tracking enables the development of technologies such as face analysis and facial recognition. When it comes to facial analysis, face tracking makes it possible to follow a particular face as it moves within a video stream, count the number of people in a video frame or love video stream, determine the direction in which a face is looking and recognise the facial expressions and perform sentiment analysis (SightCorp, 2022).

Advantages of face tracking include better security, easy integration and automated identification however the disadvantages are huge storage requirements, vulnerable detection and potential privacy issues.

### Gesture Input

Gesture Input or Gesture recognition is technology that uses sensors to read and interpret, usually hand movements, as commands. In some cars, this capability allows drivers and passengers to interact with the vehicle, usually to control the infotainment system without touching any buttons or screens.

A gesture recognition system starts with a camera pointed a specific three-dimensional zone, capturing frame-by-frame images of hand positions or the position of whatever is being tracked. These images are then analysed in real time by a computer and machine learning technologies which then translates the hand motions or gestures into commands, based on a predetermined library of signs (Mobility Innsider, 2021).

Advantages of gesture recognition are that gestures are quick to perform and lots of complex commands could be combined into one simple gesture, a lot of gestures are universal which is highly advantageous for breaking the language barrier in some ways however the disadvantages of gesture recognition is that they could be easily mistaken for different gestures, sometimes there are multiple gestures for the same thing or ones that look similar when performed and unless a highly trained AI is interpreting them, it could be hard to differentiate them.

### Voice Commands

Speech recognition is the ability of a machine or program to identify words spoken aloud and convert them into readable text or commands. Rudimentary speech recognition software has a limited vocabulary and may only identify words and phrases when spoken clearly but an advanced and highly trained AI algorithm can interpret voice commands with ease and execute them.

A good example of speech recognition in today’s technology would be Alex, Cortana, Google Assistant and Siri. All of these are trained AI algorithms that can interpret voice commands from the user and execute those commands accordingly (GetSmarter, 2019).

Advantages of voice commands being that talking is a lot faster than typing, voice commands are more efficient, and it also helps those who have difficulty with reading and sight. The disadvantages being there is the possibility of misinterpretation due to a person’s accent, the cost to set up is not cheap and background noise interference can hinder the effectiveness.

## Task 2

The second task was to create an image using OpenCV and draw several shapes on the image. For this task, a polygon, rectangle, square, circle, ellipsis and name were drawn on top of the rendered image. Below is a screenshot of the final rendered image with the shapes drawn on it:

A picture containing graphical user interface

Description automatically generated

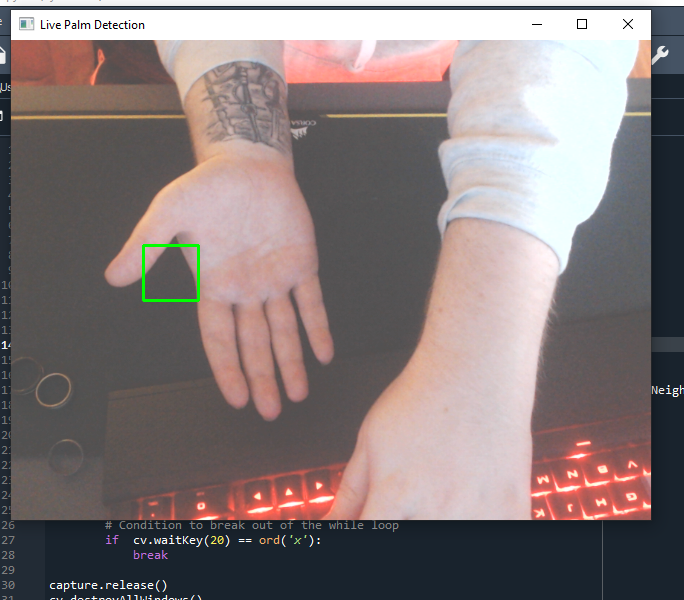
As can be seen in the screenshot of the code below, OpenCV is imported, and a colour image is loaded using the cv2.imread function. Then it was just a simple task of using the OpenCV draw functions for different shapes and defining their size and position and colour through the function parameters. A font was also defined in line 22 for the name “Callum McLaughlin” to be drawn over the image. Line 26 then displays the new image with the shapes drawn over it and the window is resized to 1024x1024 pixels in line 27. Line 29 onwards handles displaying the image until the escape key is pressed and when it is, the function cv2.imwrite() saves the image to the specified folder (OpenCV, 2022).

Text

Description automatically generated

## Task 3

The third task was to implement a programme for recognising different body parts and poses such as Front face, Smile, Palm open, fist and whole hand. This task was relatively simple to complete as the code is almost the same for each script other than the imported algorithm. Below is an example of the palm detection code working, although admittedly the drawn square is a little off the mark.



### Palm Detection Code

Text

Description automatically generated

As you can see the script is very simple, the webcam is used as the video source, the classifier is assigned to a variable and a loop repeats the drawing of the shape around the palm every time it detects it. The loop only ends whenever the programme is closed, when the ‘x’ key is pressed.

This code is the same for each other classifier. I have attached screenshots of the code below for reference.

### Smile Detection Code

Text

Description automatically generated

### Hand Detection Code

Text

Description automatically generated

### Fist Detection Code

Text

Description automatically generated

### Frontal Face Detection Code

Text

Description automatically generated

## Task 5

**Write a set of guidelines you think are appropriate to use when developing application that involve gesture-based interactions.**

As a starting point, it would be useful to review the Kinect for Windows guidelines and compare/contrast these with Neilsen’s heuristics (links/downloads on Blackboard).

Is there any overlap? Is there anything suggested by Neilsen that would make sense if it was implemented in a gesture-based interface?

Are there guidelines for mobile devices with touch screens that suggest otherwise?

### Introduction

In order to create a functionally sound and efficient interface for a user, there are several HCI considerations to be made before the design phase. User involvement in development stages can benefit the finished product greatly as the design can have iterations. Having multi-disciplinary design teams is also a very good approach as you could have specialist in Computer Science, Psychology, Ergonomics, Engineering etc to enhance the user experience and create an interface that incorporates all the essential features and comforts that a good Interface must have (Behera, 2022).

It is essential that the design team understands the specific context of us for the interface as well as the specific user group and organisation requirements. Iterations and designs must then be evaluated at the end of development against the requirements by having users test the interface.

In short, HCI requirements outline the main objectives being that the interface must have good usability, universality and usefulness which is achieved by good planning and design, following closely along the users needs, being committed to requirements analysis and thorough testing.

### HCI Guidelines

The Kinect for Windows overall design principles outlines that the best user experiences are context-aware, and the UI should adapt as the distance between the user and the sensor changes as well as the sensor responding to different stimulus in the environment it is monitoring, such as an increased number of users. It also mentions a very good point about the range of motion of the user. As the user is further away, they will have a wider range of motion but as they get closer, they could have more complex tasks and gestures performed (Microsoft, 2022).

There are several input methods, and some are the best choice for a specific task but could eb the worst for another, it can be a fine line in understanding this and its why HCI guidelines are so important. The simpler interactions are, the more confident the user can be in using them which leads to smoother inputs to the sensor. If the gestures are complex and leave room for error, the user could get confused and not perform the gesture correctly, the best way is the simple way with gesture commands.

Now the guidelines that have been created will be discussed below.

### Innate and learned Gestures

Designing for innate gestures that most people will already be familiar with can give the interface a huge boost in usability and efficiency as well as creating ones that the user will need to learn and memorize.

Creating gestures that are simple and effective, easy to remember for everyone, allows the user to have a better experience with the interface and eventually after some time, interaction will become seamless if the gestures are basic and well designed.

### Static, Dynamic and Continuous gestures

Regardless of whether a user knows a gesture by heart or not, the gestures that can be designed can range from a single stationary pose to a more prolonged motion of a part of the body such as the hand.

Having poses or gestures that are meaningful and universal allows them to be easily remembered and makes the process of learning the interface navigation seamless.

### Reliability

When designing any interface, you must design for reliability, it should be one of the top priorities. Without reliability the application can feel unresponsive and difficult to use which can be frustrating for the users.

If the gesture is too circumscribed or complex there will be fewer “false positives” but it might be frustrating or difficult for the user to perform whereas if the gesture is too unspecific or simple, it will be easier to perform but might have a lot more false positives or conflicts with other gestures. That is why it is essential to strike a medium between complex and simple.

### Design Appropriately for target audience

Regardless of how the gestures are defined, developers must make sure that they keep their target audience in mind so that the gestures work for the heigh ranges, physical and cognitive capabilities of the target users.

There must be frequent usability tests conducted and developers must ensure they test the full range of intended user types.

### Determine user intent and engagement

The interface must be able to determine when a user has intent on interacting with it. This is a key issue and is very difficult to do well. Unlike other devices that can review input, those which require explicit contact from a user or only track a small area of their body, a system such as Kinect for Windows sees a person holistically.

The challenge is to detect when the user intends to interact with the interface correctly and avoid detecting “false positives”.

### Design for variability of input

Users who have previous experience on other interfaces that may be like your own can affect how they interact with your application. You must keep in mind that just because on person performs a gesture a certain way, does not mean everyone will, there could be slight deviations such a waving from the wrist, the elbow or with their whole arm.

All of these are wave gestures that should all be picked up by an interface. If it proves too difficult to allow the user to have multiple ways of performing the gesture, it may be best to give the users clear instruction about the exact gesture that the interface requires.

### Iterate

Designing and gesture and getting it to feel natural and right to the user might take several iterations. Creating parameters for anything that you can and make sure to constantly test it. Iterations are proven to be an invaluable tool in rapid development of applications because you can constantly tweak, assess and repeat until you have what you intended on creating.

## Nielsen's Heuristics

When reviewing both the Kinect for Windows guidelines and Nielsen’s 10 heuristics, there were a few that stood out in Nielsen’s Heuristics that were not as heavily focused on or even mentioned in the Kinect for Windows Guidelines.

One of these being error prevention and that instead of having good error messages that tell the user what went wrong, the interface should be designed so that it prevents a problem from occurring in the first place.

Eliminating error-prone conditions or checking for them and presenting the user with confirmation options before they commit to the action seems like a better design philosophy that is mentioned in Kinect for Windows’ Guidelines (Heurio, 2022).

Another difference between the two is that Nielsen’s Heuristics also mentions aesthetic and minimalist design as a key guideline which Kinect for Windows simply does not cover in as much detail if at all.

Nielsen’s Hesutics say that dialogues should not contain information which is irrelevant or not needed and that every additional unity of information in a dialogue completes with the other relevant units of information and diminishes their overall visibility which can lead to confusion and frustration for the user.

# Bibliography

American Psychological Association, 2022. *APA Dictionary of Psycology - Perceptual User Interface.* [Online]   
Available at: https://dictionary.apa.org/perceptual-user-interface  
[Accessed 03 03 2022].

Behera, A., 2022. *Learning Edge - Edgehill - CIS3149/3118 – Interface Programming - Week 8 Lecture.* [Online]   
Available at: https://learningedge.edgehill.ac.uk/ultra/courses/\_278894\_1/cl/outline  
[Accessed 15 03 2022].

Carroll, J., 2001. *The Evolution of Human-Computer Interaction.* [Online]   
Available at: https://www.informit.com/articles/article.aspx?p=24103#:~:text=In%20the%201990s%2C%20computer%20science,comprehensive%20undergraduate%20texts%20have%20appeared.  
[Accessed 12 03 2022].

Carroll, J. M., 2021. *Human Computer Interaction - Brief Intro.* [Online]   
Available at: https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro  
[Accessed 23 02 2022].

CS.Bham, 2022. *Human Computer Interaction - Teaching.* [Online]   
Available at: https://www.cs.bham.ac.uk/~rxb/Teaching/HCI%20II/intro.html  
[Accessed 04 03 2022].

DeWitt, D., 2019. *THE EVOLUTION OF HUMAN-COMPUTER INTERACTION AND DIGITAL SIGNAGE.* [Online]   
Available at: https://www.sixteen-nine.net/2019/10/25/the-evolution-of-human-computer-interaction-and-digital-signage/  
[Accessed 08 03 2022].

GetSmarter, 2019. *Applications of Speech Recognition.* [Online]   
Available at: https://www.getsmarter.com/blog/market-trends/applications-of-speech-recognition/  
[Accessed 14 03 2022].

Heurio, 2022. *Nielsen's 10 Usability Heuristics.* [Online]   
Available at: https://www.heurio.co/nielsens-10-usability-heuristics  
[Accessed 14 03 2022].

Microsoft, 2022. *Kinect For Windows.* [Online]   
Available at: https://learn-eu-central-1-prod-fleet01-xythos.content.blackboardcdn.com/5b74134a6d3cc/1176893?X-Blackboard-Expiration=1647486000000&X-Blackboard-Signature=h3ULLeOJLWjXttEzOTBJOHyxK3sCK9S7AqdQSMMW47M%3D&X-Blackboard-Client-Id=309029&response-cache-control  
[Accessed 12 03 2022].

Mobility Innsider, 2021. *What is Gesture Recognition?.* [Online]   
Available at: https://www.aptiv.com/en/insights/article/what-is-gesture-recognition#:~:text=Gesture%20recognition%20is%20technology%20that,touching%20any%20buttons%20or%20screens.  
[Accessed 12 03 2022].

OpenCV, 2022. *Image File Reading and Writing.* [Online]   
Available at: https://docs.opencv.org/3.4/d4/da8/group\_\_imgcodecs.html#gabbc7ef1aa2edfaa87772f1202d67e0ce  
[Accessed 12 03 2022].

SightCorp, 2022. *What is Face Tracking Technology?.* [Online]   
Available at: https://sightcorp.com/knowledge-base/face-tracking/  
[Accessed 11 03 2022].

X, L., 2017. *A Brief History of Human-Computer Interaction (HCI).* [Online]   
Available at: https://blog.prototypr.io/the-rise-of-human-computer-interaction-hci-823dd6286e1d  
[Accessed 11 03 2022].