Design and User Experience in Software for Medical & Psychological Research

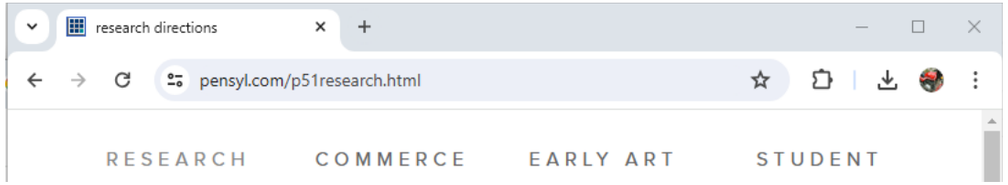
Russell Pensyl, Professor (ret.)

pensyl.com | Northeastern University | Boston, MA, USA

Much of the text in the paper is drawn from "InterAct, a history of interaction design, experience & interactive art."

pensyl@pensyl.com

Keywords: Vision/sensor; fusion, facial recognition; biometric data capture; emotion response detection; ultrasonic tracking for positioning & localization; mixed reality; physical computing.



Writing Samples

Design and User Experience in Software for Medical and Psychological Research

Facial Recognition and Emotion Detection In Environmental Installation

User Centric Content Delivery Using Biometric DataCapture and Intelligent Analysis

Robust Hybrid Tracking in Mixed Reality Environments

"interAct" – The history and analysis of interaction art and design.

A comprehensive history of interaction design, experience and media. Anticipated publish date – late 2024 or early 2025.

This text presents exhaustive survey of current digital, interactive artifacts created by modern cultures and a comprehensive history of cultural practices and behaviors that inform and illuminate their development. In this historiography, the survey includes not only development of software technologies over time but to looks at these from material culture, culture-historical archaeology and cultural ontological perspective where the objects of study are digital artifacts themselves. Digital artifacts are indicators of not only

full paper and slides at:
https://pensyl.com/p51research.html

Author's background:

1977-79 Studied: University of California at Los Angeles
Computer Science, Systems Analysis

1985 Bachelor of Fine Arts, Western Michigan University
Printmaking, Computer Science

1988 Master of Fine Arts, Western Michigan University
Multi-Media, Electronic Music

32 years – Professor, Chair, Vice Dean, Director, etc.

40+ years – producer | developer, media, interaction design, software design,
Clients: **Apple, IBM, Motorola, Intel, Disney. American Airlines, Adobe, Kodak, Palm Pilot, Pearson/Prentice Hall, Intelect, Inc., Highland Instruments, & others.**

https://www.pensyl.com/CVPensyl.pdf

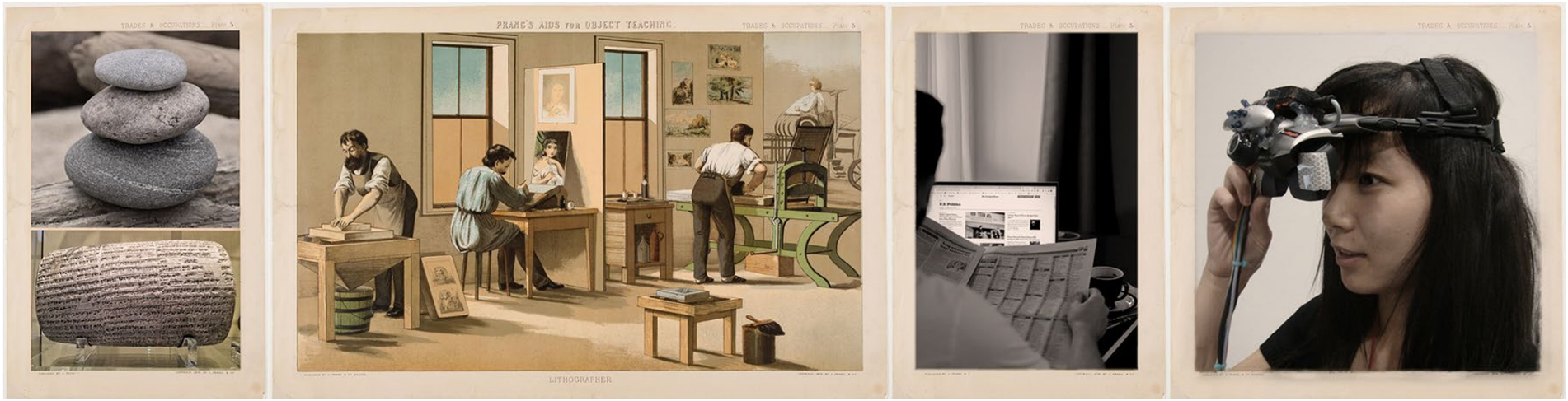
Author’s background:

Software design:

Facial recognition, emotion detection - Emota v4.0 ™

Mixed & Augmented reality

Gesture recognition, gait recognition, non-touch interfaces



https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

Author's background:

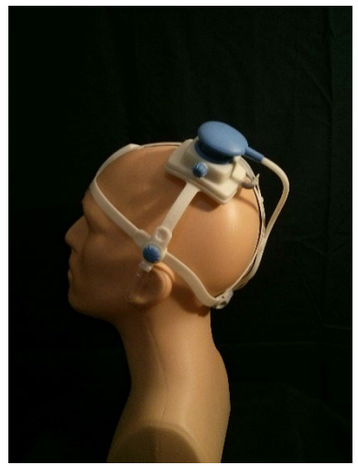
Software design:

Advance Vision-Sensor Fusion research/analysis

* *with researchers from MIT, Harvard & Boston Hospitals*

Device development

Electro-stimulation headset


https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

Introduction – interaction strategies, & user experience

Covered in the full paper:

Analysis Methods in Interaction Strategy & User Interface.

Design Process for development & Analysis.

Constraints & Affordances Analysis.

Structured Data.

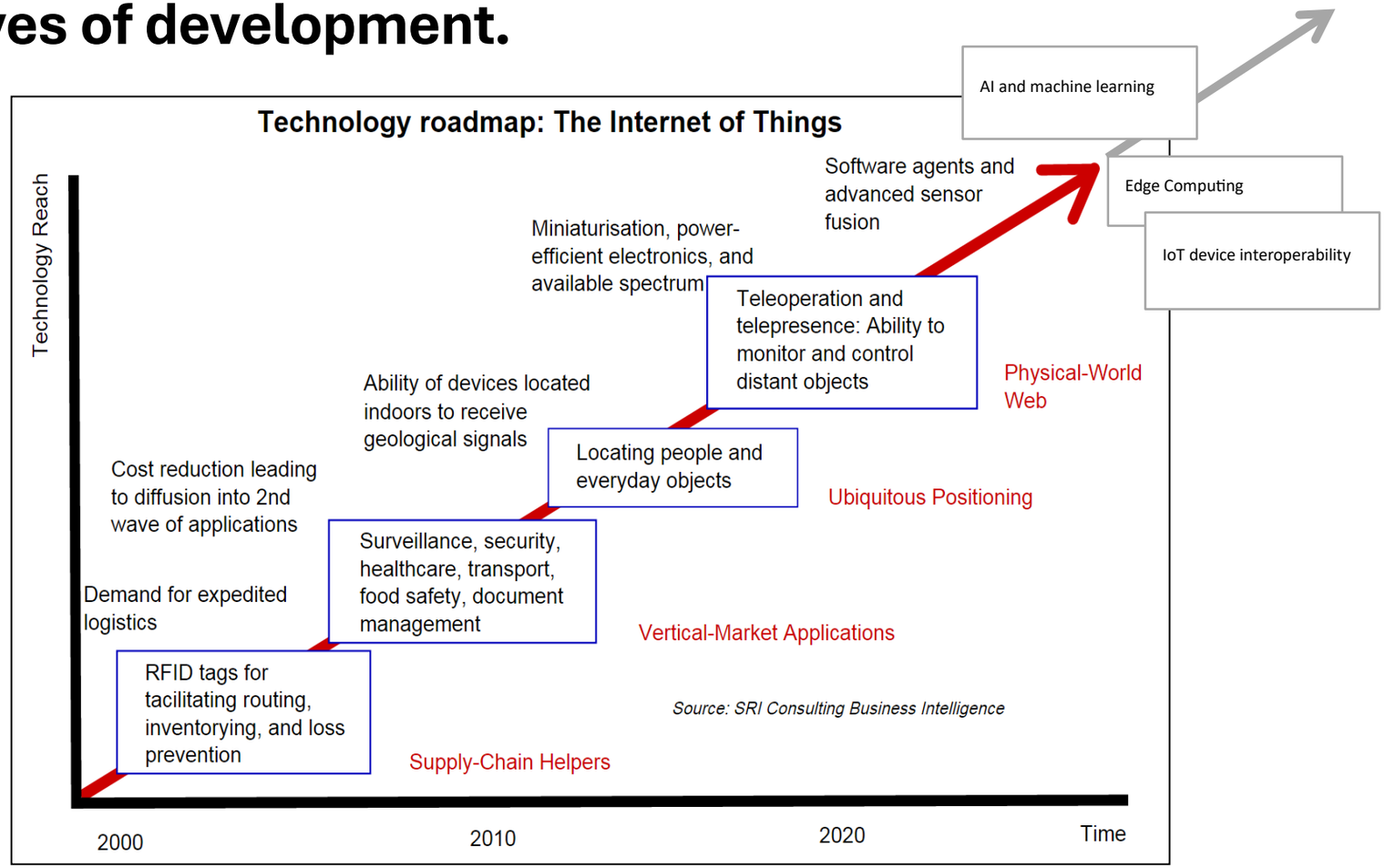
Facial recognition & Emotion Detection.

- As may be used as a data collection method for research purposes.

Computing systems function as a passive companion.

- Interaction strategies & experience affects data collection.

Introduction – current state of technology & extrapolation of the “next” waves of development.



https://en.wikipedia.org/wiki/File:Internet_of_Things.svg

Introduction – current state of technology & extrapolation of the "next" waves of development. Impacting sensor data collection

- Edge computing: data collection, processing & analysis
  Coser to the source, reducing latency & improving real-time decision-making.
- AI & machine learning
  Software agents act autonomously on behalf of humans.
- The IoT Consortium |Open Group **standards & guidelines**
  Device interoperability – similar to what Berners-Lee did for sharing o data content via standardized mark up, hypertext transfer protocol & uniform resource location.

Introduction – Considerations for software design in research data collection systems.

Competencies:

High level domain knowledge & expertise – Medical.
High level domain knowledge & expertise – Psychology – or other domains.

Software Engineering & Coding.

UI - User Interface.

UX - User experience design.

Introduction – Team-based work vs single expert

In team-based development - stakeholders speak & think with different conceptual frameworks that are almost different languages & linguistic syntaxes.

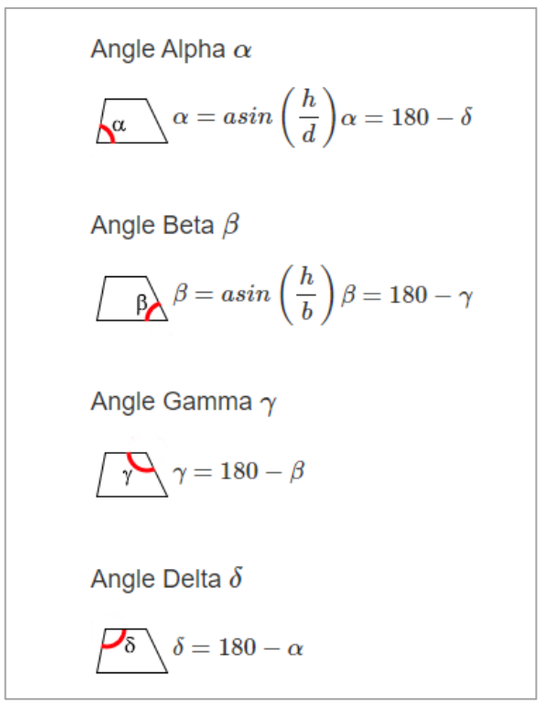
Possible - medical researcher has software design & engineering experience.

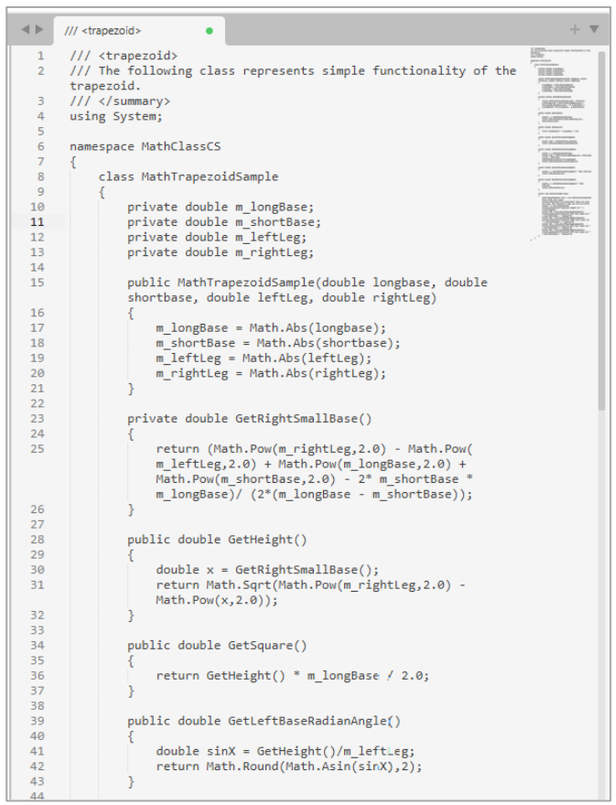
Often - software engineer or visual designer may have shared competencies.

Rare - one person has all three of these competencies.

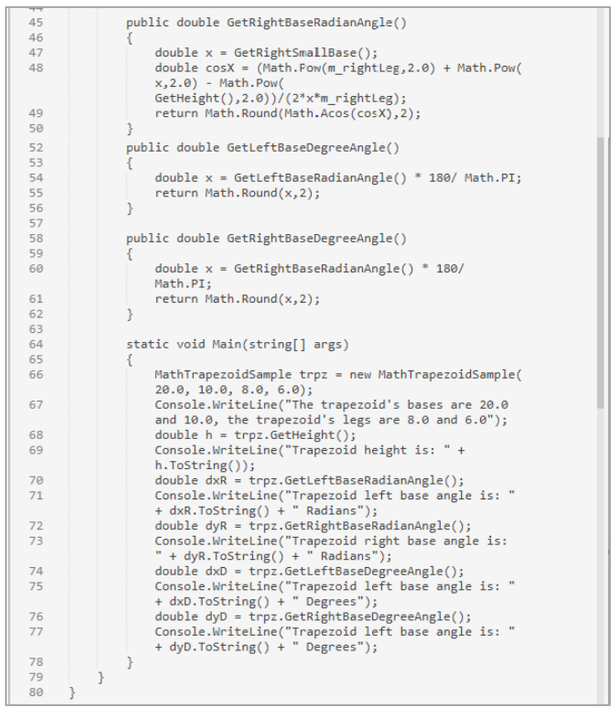
Introduction – software idiosyncrasies

Simple mathematical functions, such as for calculating angles in a trapezoid, can be easily processed by hand, becoming eighty lines of code in C#.





```
/// <trapezoid>
/// The following class represents simple functionality of the
trapezoid.
/// </summary>
using System;

namespace MathClassCS
{
    class MathTrapezoidSample
    {
        private double m_longBase;
        private double m_shortBase;
        private double m_leftLeg;
        private double m_rightLeg;

        public MathTrapezoidSample(double longbase, double
        shortbase, double leftLeg, double rightLeg)
        {
            m_longBase = Math.Abs(longbase);
            m_shortBase = Math.Abs(shortbase);
            m_leftLeg = Math.Abs(leftLeg);
            m_rightLeg = Math.Abs(rightLeg);
        }

        private double GetRightSmallBase()
        {
            return (Math.Pow(m_rightLeg,2.0) - Math.Pow(
            m_leftLeg,2.0) + Math.Pow(m_longBase,2.0) +
            Math.Pow(m_shortBase,2.0) - 2* m_shortBase *
            m_longBase)/ (2*(m_longBase - m_shortBase));
        }

        public double GetHeight()
        {
            double x = GetRightSmallBase();
            return Math.Sqrt(Math.Pow(m_rightLeg,2.0) -
            Math.Pow(x,2.0));
        }

        public double GetSquare()
        {
            return GetHeight() * m_longBase / 2.0;
        }

        public double GetLeftBaseRadianAngle()
        {
            double sinX = GetHeight()/m_leftLeg;
            return Math.Round(Math.Asin(sinX),2);
        }
```



```
        public double GetRightBaseRadianAngle()
        {
            double x = GetRightSmallBase();
            double cosX = (Math.Pow(m_rightLeg,2.0) + Math.Pow(
            x,2.0) - Math.Pow(
            GetHeight(),2.0))/(2*x*m_rightLeg);
            return Math.Round(Math.Acos(cosX),2);
        }
        public double GetLeftBaseDegreeAngle()
        {
            double x = GetLeftBaseRadianAngle() * 180/ Math.PI;
            return Math.Round(x,2);
        }

        public double GetRightBaseDegreeAngle()
        {
            double x = GetRightBaseRadianAngle() * 180/
            Math.PI;
            return Math.Round(x,2);
        }

        static void Main(string[] args)
        {
            MathTrapezoidSample trpz = new MathTrapezoidSample(
            20.0, 10.0, 8.0, 6.0);
            Console.WriteLine("The trapezoid's bases are 20.0
            and 10.0, the trapezoid's legs are 8.0 and 6.0");
            double h = trpz.GetHeight();
            Console.WriteLine("Trapezoid height is: " +
            h.ToString());
            double dxR = trpz.GetLeftBaseRadianAngle();
            Console.WriteLine("Trapezoid left base angle is: "
            + dxR.ToString() + " Radians");
            double dyR = trpz.GetRightBaseRadianAngle();
            Console.WriteLine("Trapezoid right base angle is:
            " + dyR.ToString() + " Radians");
            double dxD = trpz.GetLeftBaseDegreeAngle();
            Console.WriteLine("Trapezoid left base angle is: "
            + dxD.ToString() + " Degrees");
            double dyD = trpz.GetRightBaseDegreeAngle();
            Console.WriteLine("Trapezoid left base angle is: "
            + dyD.ToString() + " Degrees");
        }
    }
}
```

Introduction – software idiosyncrasies

Simple shadow removal formula for Haar Classification for facial recognition in standard math & translated into C# & Python:

$$f(x)=\begin{cases}\alpha K\log(x)+(1-\alpha)x & ,\ if\ x<127\\ \alpha[255-K\log(x)]+(1-\alpha)x & ,\ if\ x\geq 127\end{cases}$$

shadowAdjust.cs

```cs
using System;
using System.Math;

public class Program
{
    private static readonly double ALPHA = 0.64;
    private static readonly double K = 256;
    private static double varAdjust = 1;

    public static double Function(double x)
    {
        if (x >= 127)
        {
            return ALPHA * K * Math.Log10(x) + (1 - ALPHA) * x;
        }
        else
        {
            return ALPHA * (255 - K * Math.Log10(x)) + (1 - ALPHA) * x;
        }
    }

    public static void Main()
    {
        double x = 254;
        double functionX = Function(x);
        varAdjust = (double)x + ", f(x) = " + functionX;
        Console.WriteLine($"Input: x: {x}, f(x) = {functionX}");
    }
}
```

shadowAdjust.py

```py
from math import log10

ALPHA = 0.64
K = 256
varAdjust = 1

def function(x: float):
    if x >= 127:
        return ALPHA * K * log10(x) + (1 - ALPHA) * x
    else:
        return ALPHA * (255 - K * log10(x)) + (1 - ALPHA) * x

x = 254
function_x = function(x)
varAdjust = (f"Input: x: {x}, f(x) = {function_x}")
```

Introduction – parallel apps & libraries

Project retrieves data from an external source, or sensor, requires two or more applications running in parallel.

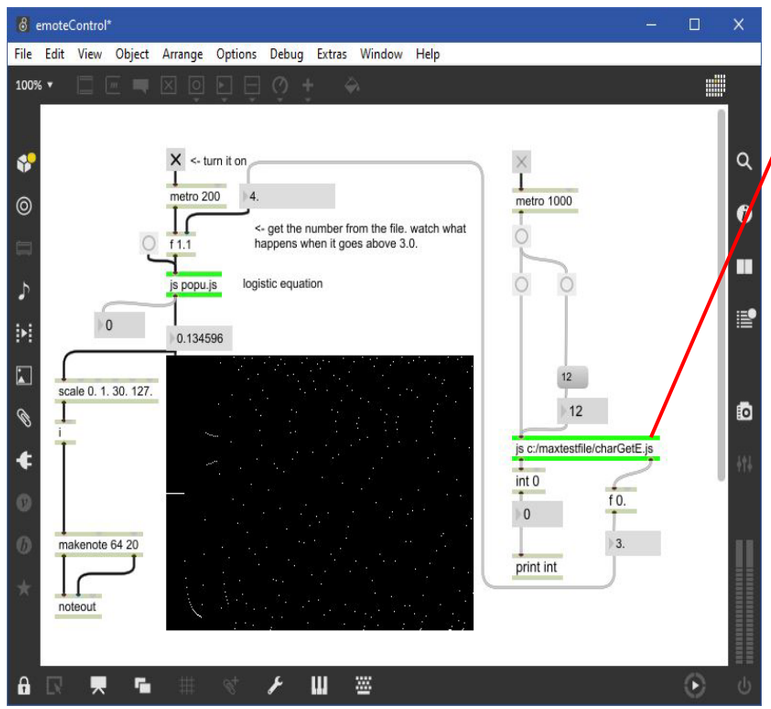


Figure16: Custom Javascript code affords communication between emoteControl Music app and Emota v3.0.

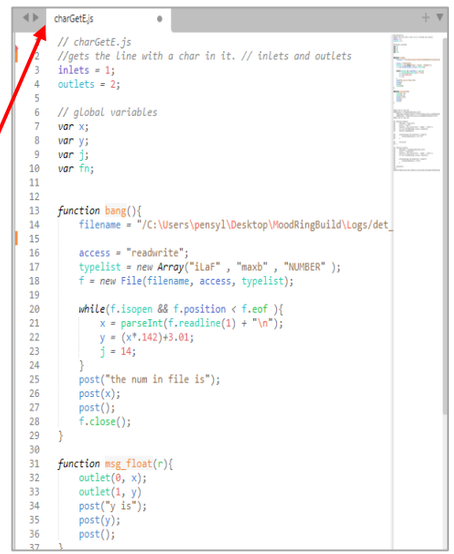


Figure 16: Javascript to pass data via serial ports.

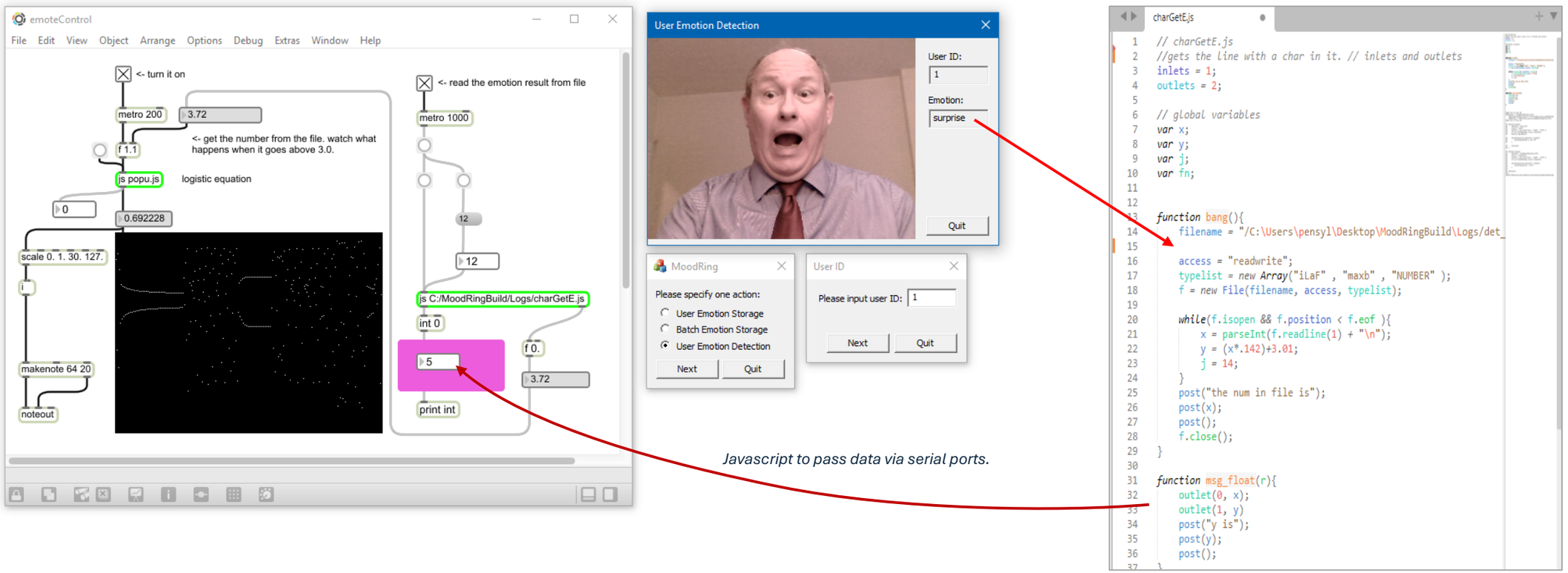
Serial communication between different applications.

facial/emotion detection app communicates via Serial port with a musical composition app.

* Custom Javascript

Introduction – parallel apps & libraries

External source, or sensor, requires two or more parallel applications.



Introduction – parallel apps & libraries

Single applications – in current high-level languages (C#, C++, Python) use "libraries" – such as a math library.

shadowAdjust.cs

```
1  using System;
2  using System.Math;
3
4  public class Program
5  {
```

/// <trapezoid>

```
1  /// <trapezoid>
2  /// The following class represents simple functionality of the
   trapezoid.
3  /// </summary>
4  using System;
5
6  namespace MathClassCS
7  {
8      class MathTrapezoidSample
```

Arduino Mega or Mega 2560

alogFirmata.ino ● AnalogFirmata.ino ●

```
1  /* Firmata is a generic protocol for communicating with microcontrollers
2   * from software on a host computer. */
3  #include <Servo.h>
4  #include <Firmata.h>
5
6  /* servos */
7  Servo servo9, servo10; // one instance per pin
8  /* analog inputs */
9  int analogInputsToReport = 0; // bitwise array to store pin reporting
10 int analogPin = 0; // counter for reading analog pins
11 /* timer variables */
12 unsigned long currentMillis;     // store the current value from millis()
13 unsigned long previousMillis;    // for comparison with currentMillis
```

Each of these incur a "cost" in processing time.

Internal & external data point collection

One important aspect of data collection – as well as for many other applications – is the frequency response rate of the data collection. Devices are:

- CPU
- USB Bus transmission speed
- Hard-wired Sensors (EEG, EKG)
    - Pressure, weight, gait, tensor, angle sensors, etc.
- Wireless Sensors (Bluetooth or Wi-Fi) (ex. FitBit, Gyroscopic Accelerometers, etc.)
- Vision systems (AR/VR, Kinect, Custom coded webcam)

External devices for data point collection

Short list of external devices for data collection

WIMP – keyboard, mouse, touchpad
Eye tracking,
Gyroscopic accelerometers angle/rotation,
Motion sensors -Fitbit, Nike Fuelband, etc.
EEG/Brainwave/Pulse sensors,

Vision systems for Spatial positioning,
Vision systems for Facial recognition.

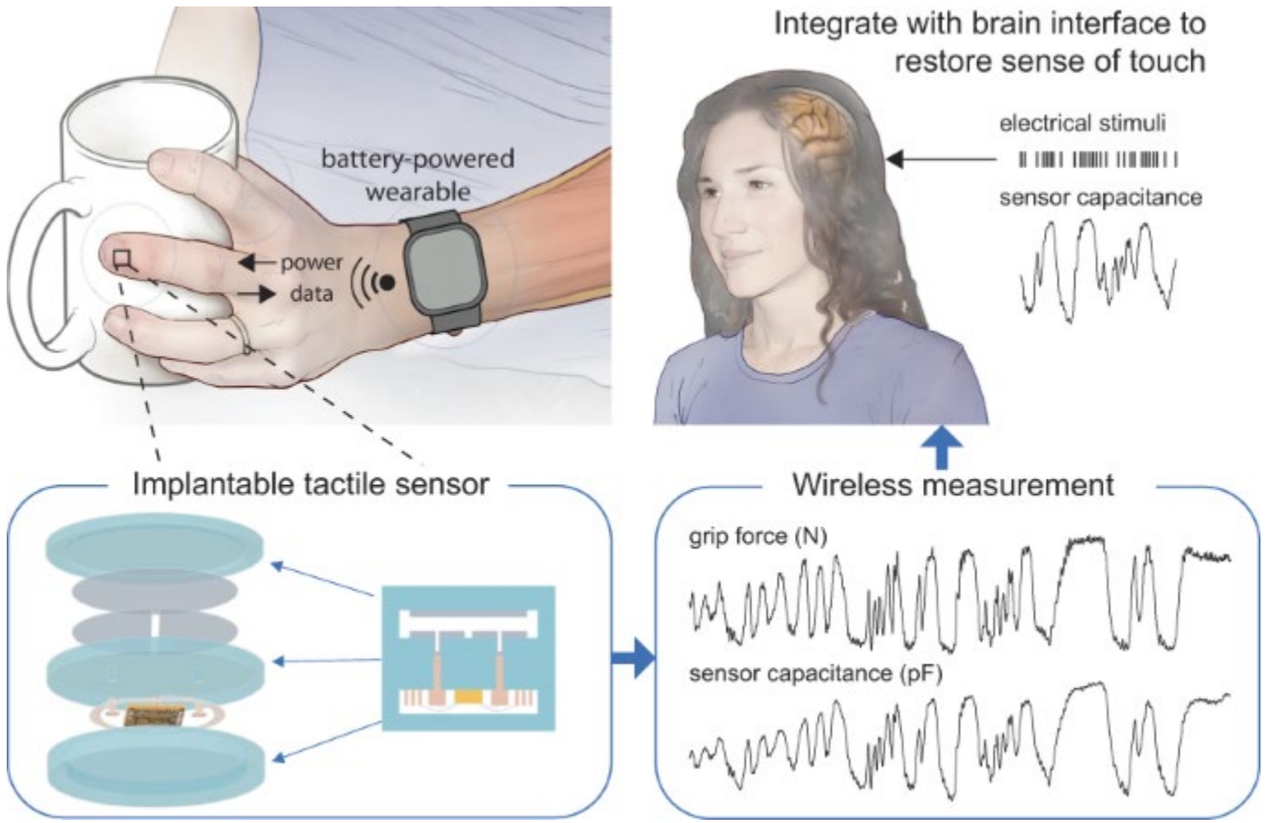
Ultrasonic tracking
Touch screen (capacitance) Sensors,
Gravity sensors, Compass,
Galvanic Skin Response sensors,

Vision systems for Motion detection,

https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

External devices for data point collection

One important aspect of data collection – each sensor or data point collection incurs a cost in compute cycles.



https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

External devices for data point collection

One important aspect of data collection – each sensor or data point collection incurs a cost in compute cycles.

https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

External devices for data point collection

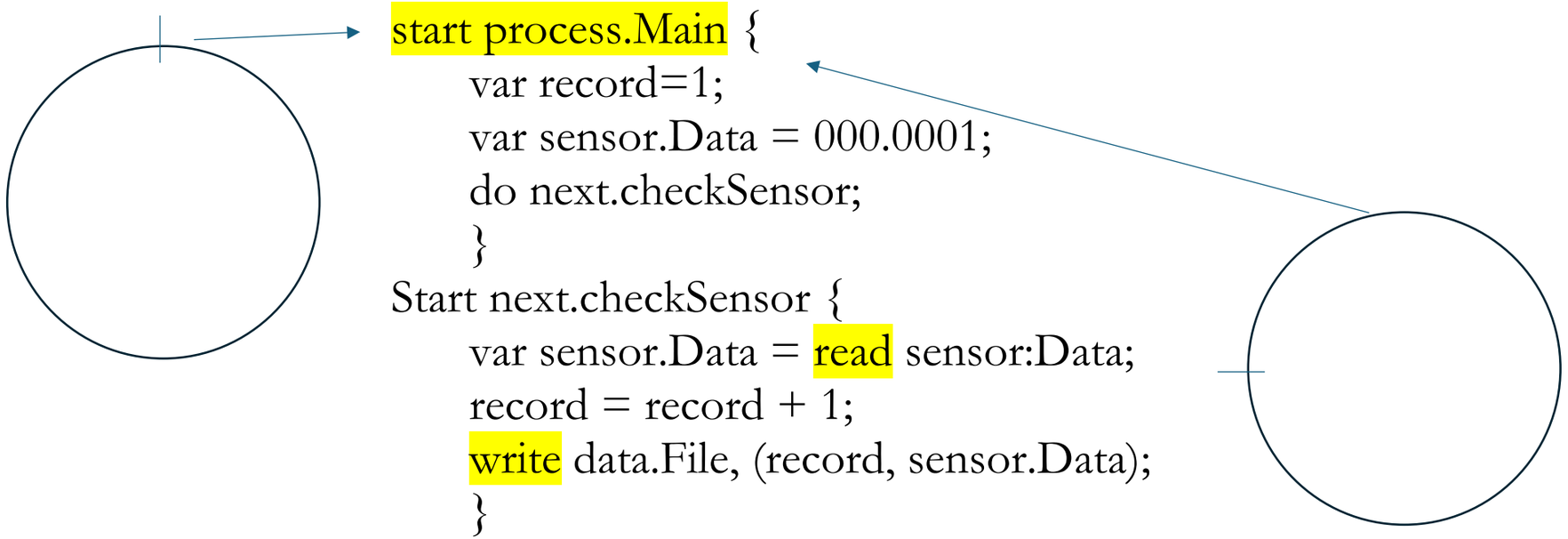
One important aspect of data collection – each sensor or data point collection incurs a cost in compute cycles.

Writing the date incurs another cost in computer cycles.

How it works - cost of processing using external devices

To an understanding of how a computer works, download John Scott's book, "But How Do It Know?"

A simple analogy: analog clock & appointments. At each 12 o'clock, CPU polls base instruction set & tells components in the CPU or other to do something.



```
start process.Main {
      var record=1;
      var sensor.Data = 000.0001;
      do next.checkSensor;
      }
Start next.checkSensor {
      var sensor.Data = read sensor:Data;
      record = record + 1;
      write data.File, (record, sensor.Data);
      }
```

How it works - cost of processing using external devices

An Arduino microcontroller is useful & FUN tool to learn basic coding & control of devices. Many people use these for self-built Home Automation systems

```
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH = on)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off (LOW = off)
  delay(1000);                      // wait for a second
}
```

* http://www.arduino.com

How it works - cost of processing using external devices

Sampling frequencies of devices used for data collection.

Is there a clock?

Can it be reliably used

Synchronization and reconciliation of sampling frequency in real time vs post capture re-processing.

How it works - cost of processing using external devices

- CPU, “system” clock managed by the hardware bios & operating system.
- Code in a high-level languages like C# or C++ can collect an event in “system time” as a data point.
- Sensors or other external input devices used to collect data may or may not have a clock.
- Some devices simply stream data as fast as their internal components run. Accessing device clock is not available in the device API.

How it works - cost of processing using external devices

Cost in “time” for processing – each sensors or process add “time” to collection. Discrepancies in data point time of collection.

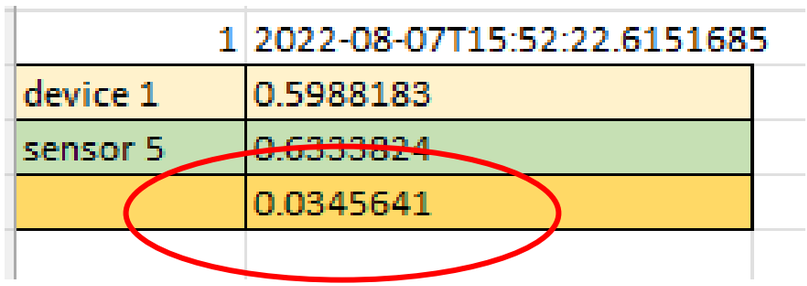
FrameCount	SystemTime	DeviceTimeStamp	TimeElaps	Trigger	DeviceFra	TriggerCo	data poin	X(Millime	Y(Millimet	Z(Millimet	Confidenc	TimeElaps	Sample ra	AverageTimeElapsedInterval(millsec)
16985	2022-08-07T15:52:22.598818	1664700222	10.4281	0		0	collect po	-135.14	103.403	1275.5	Medium	10.4281		
1	2022-08-07T15:52:22.616430	18258940	2.7022	0	912	0	0.0075	0.0075	-0.0032	-0.4443	-0.389	0.8546	2.7022	
1	2022-08-07T15:52:22.629326	18279221	15.5987	0	913	0	0.0053	-0.0117	-0.0064	-0.3189	-0.3641	0.9085	15.5987	
1	2022-08-07T15:52:22.620382	18279316	6.6545	0	913	0	-0.0032	-0.0011	-0.0021	-0.752	-0.4171	0.5244	6.6545	
1	2022-08-07T15:52:22.623296	18279942	9.5689	0	913	0	0	-0.0053	0.0085	-0.041	-0.6811	0.7509	9.5689	
1	2022-08-07T15:52:22.633382	18279800	19.655	0	913	0	0.0032	0.0021	0.016	-0.7289	-0.4146	0.5992	19.655	
1	2022-08-07T15:52:22.625255	18279747	11.5292	0	913	0	-0.0032	-0.0085	0.0021	-0.1609	-0.6312	0.7256	11.5292	
1	2022-08-07T15:52:22.617406	18279793	3.6795	0	913	0	0.0064	-0.0085	-0.0011	0.4687	-0.8797	0.0181	3.6795	
1	2022-08-07T15:52:22.621343	18279615	7.6173	0	913	0	-0.0011	0.0075	0.016	0.0314	-0.7375	0.6972	7.6173	
1	2022-08-07T15:52:22.6262895		12.568	0		0	13.1546	24.7571	11.7453	13.1687	4.35401	-2.4826	62.8257	12.568
1	2022-08-07T15:52:22.6151685		1.4481	0		0	30954	63455	65535	65535	FALSE	FALSE	1.4481	
device 1	0.5988183													
sensor 5	0.6333824													
	0.0345641													
35304	2022-08-07T16:12:17.645884	2859433555	15.9219	0		0	collect po	-139.46	94.1217	1290.22	Medium	15.9219		
1	2022-08-07T16:12:17.659486	70839053	4.3626	0	3541	0	0.0117	0.0053	-0.0096	-0.4413	-0.3838	0.8484	4.3626	
1	2022-08-07T16:12:17.670457	70859270	15.3348	0	3542	0	-0.0032	-0.017	-0.0064	-0.3225	-0.3696	0.9084	15.3348	
1	2022-08-07T16:12:17.674373	70859904	19.251	0	3542	0	0	-0.0021	-0.0053	-0.7476	-0.42	0.5299	19.251	
1	2022-08-07T16:12:17.668507	70859113	13.3837	0	3542	0	0.0075	0.0011	0.0128	-0.0331	-0.6819	0.75	13.3837	
1	2022-08-07T16:12:17.668337	70860011	13.2065	0	3542	0	0.0021	-0.0032	0.0128	-0.7218	-0.4147	0.6052	13.2065	
1	2022-08-07T16:12:17.674279	70859275	19.1568	0	3542	0	0.0011	-0.0096	-0.0043	-0.1536	-0.6313	0.7266	19.1568	
1	2022-08-07T16:12:17.659458	70859623	4.3344	0	3542	0	-0.5323	0.0085	-2.5593	0.4626	-0.816	-0.222	4.3344	
1	2022-08-07T16:12:17.655472	70840001	0.3475	0	3541	0	-0.0075	0.0085	0.0277	0.0217	-0.7374	0.6952	0.3475	
1	2022-08-07T16:12:17.6625357		7.418	0		0	12.3505	21.0989	12.0984	14.2155	3.81799	-1.4328	59.7632	7.418
1	2022-08-07T16:12:17.657178	2.0595		0		0	32539	65535	65535	65535	FALSE	FALSE	2.0595	
device 1	0.6458843													
sensor 4	0.6685072													
	0.0226229													

https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

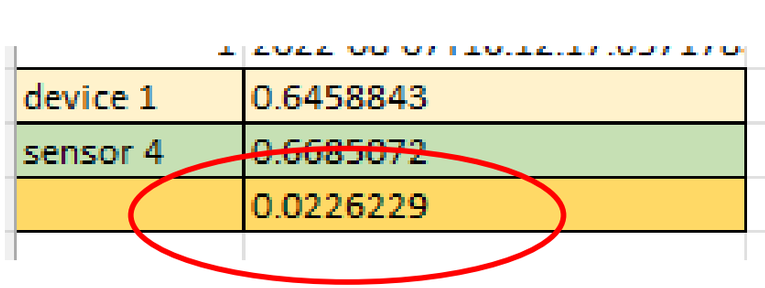
How it works - cost of processing using external devices

Cost in “time” for processing – each sensors or process add “time” to collection. Discrepancies in data point time of collection.

Human reaction time average: forty milliseconds.



	1	2022-08-07T15:52:22.6151685
device 1		0.5988183
sensor 5		0.6333824
		0.0345641



device 1	0.6458843
sensor 4	0.6685072
	0.0226229

Sophisticated algorithms are needed to compensate for variations in time of collection.

But: if done at processing time – incurs another cost in time.

How it works - cost of processing using external devices

Factors affecting Cost in “time” for processing.

Data loss in mid-stream collection.

Gyroscopic-Accelerometers may have programmable frequencies.

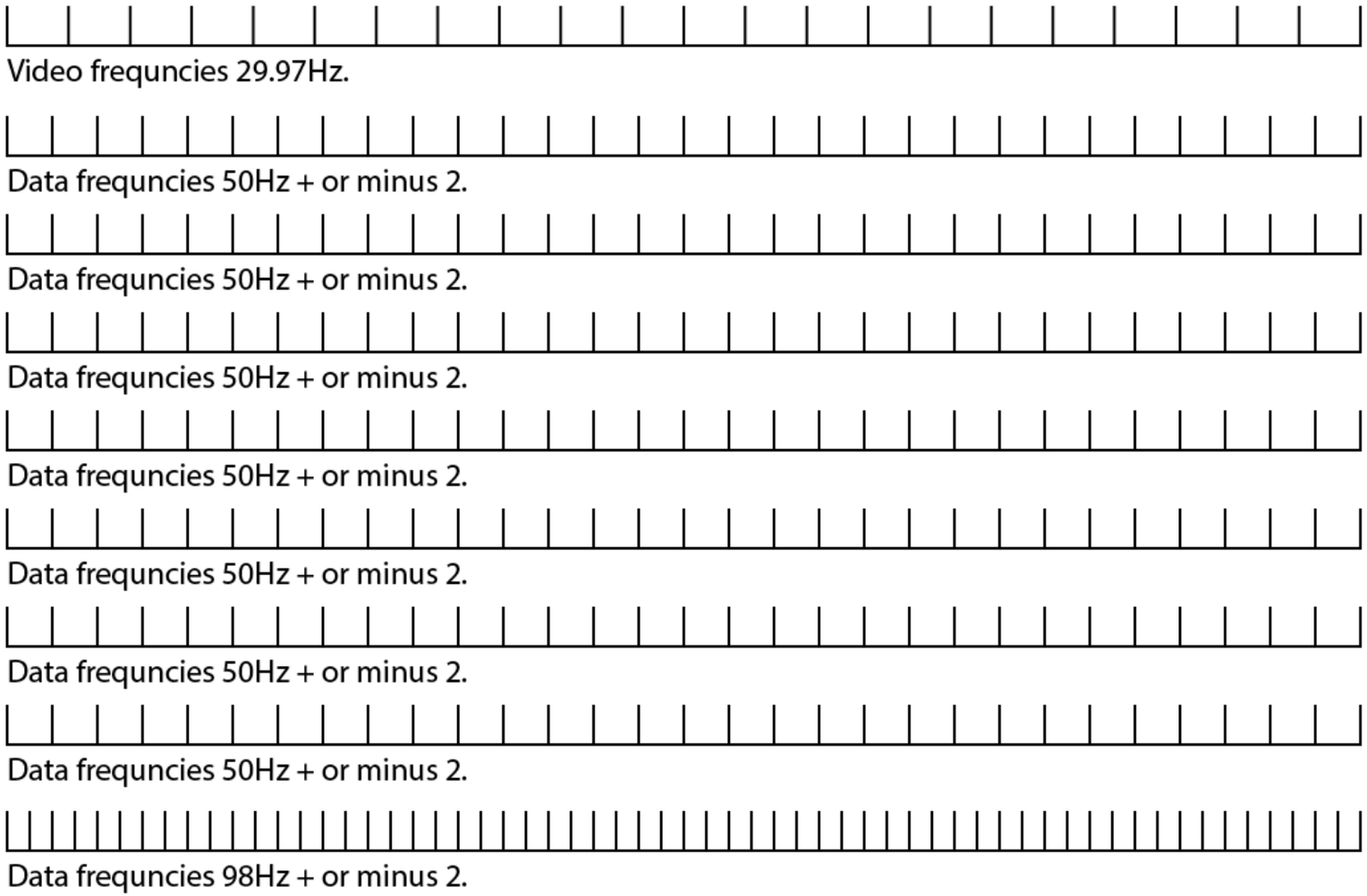
Trade-off in faster collection vs data loss.

Synchronization of data streams

How it works - cost of processing using external devices

Ideal is NO Data loss or “drop-out.”

Synchronization of data streams

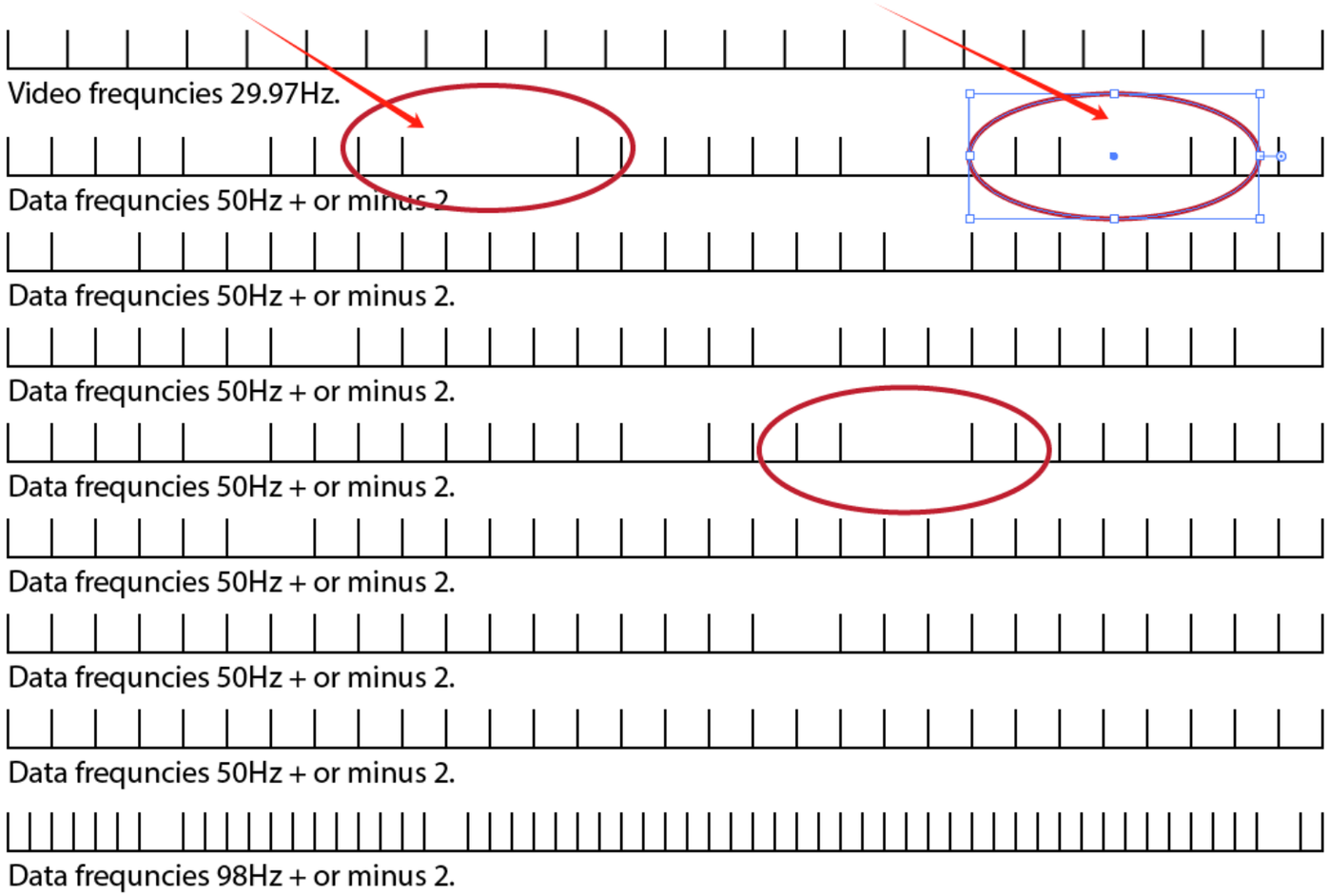


How it works - cost of processing using external devices

Actual = Data loss or "drop-out."

Testing data drop-out in midstream

Threshold of acceptable loss?



How it works - cost of processing using external devices

Factors affecting Cost in “time” for processing.

- Each additional sensors adds “time” to collection.
- Wireless sensors operate in common consumer 2.4 gigahertz frequency range
- Sensor buffer storage may overflow resulting in data loss.
- Interference in the frequency range.
- Time of flight & delivery of data from sensor to dongle is “probabilistic.”
  - Results in possible extreme variations in system time vs device time.
  - Some devices may not use a device clock.
    - Correction of synchronization requires post-processing.
      - In write of data file?
      - In post-collection data analysis?

How it works - cost of processing using external devices

Best practice: collect data in RAM buffer in raw data, simplest delimitation.

On stop of collection – dump (write) to data file.

Formats for data files:

CSV vs XML

CSV is considered a flat structure of data format.

CSV is significantly smaller than XML, requiring less processing power.

How it works – stand-alone turn-key systems vs custom designed

Extensibility? Adaptability?

Emotiv headset

EEG signal – analog

to digital – 128 | 256 Hz

GE CASE Exercise EKG 12 streams @ 60Hz

User Experience – User interface considerations

User Experience – User interface considerations

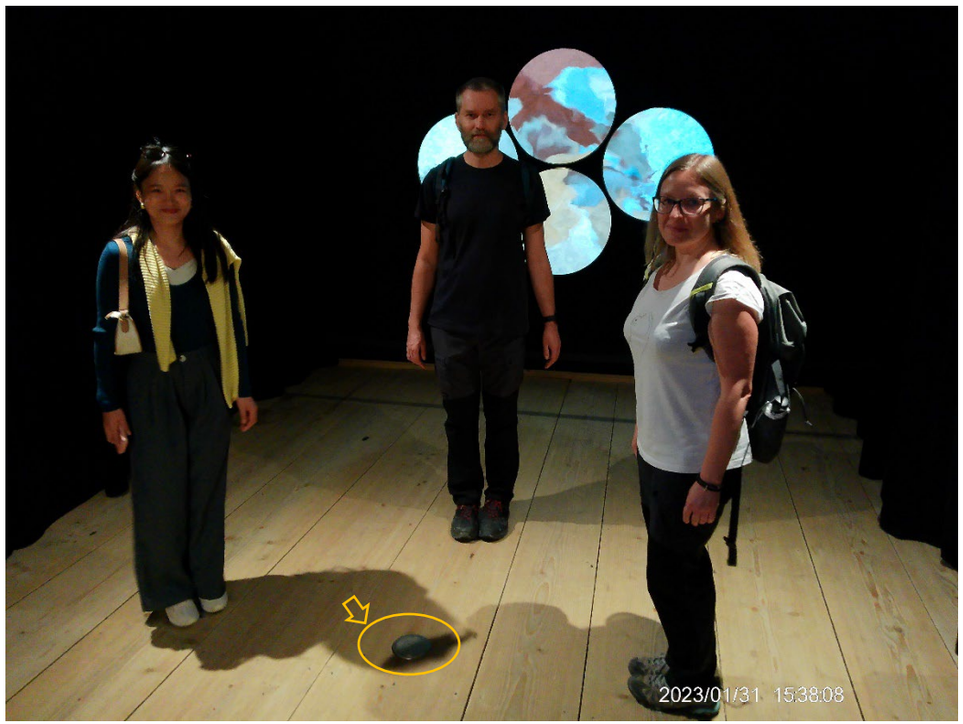
- Who is using the system?
- For what is the system used?
- Where & how are the systems to be used.

User Experience – User interface considerations

- Flow, affordances & sequences of actions.
  - Observation adjustment & refinement of UI & flow.
- What data needs to be visible in capture time.
- Post-process reliability & evidence successful data capture in sessions.
- Who are the users – MDs, expert lever technicians? Training of operators.
- Clarity of UI elements & affordances.

User Experience – User interface considerations – for ALL

The best user interface & experience is undoubtedly the one with the least actions that acts as a passive companion.



Interactive system to play five section of a full orchestra – Mozart's birth house, Salzburg, AT.

Sensors on the floor affords up to five participants to instantiate the five sections of one of Mozart's pieces.

User Experience – User interface considerations – for ALL

The best user interface & experience is undoubtedly the one with the least actions that acts as a passive companion.

With the advent of functional AI systems, feature rich, hidden, or transparent applications as the norm.

What may be possible: Project Milo[1], a demonstration of a "controller-free:" depth sensing & pattern recognition technologies. Milo is an example of emergent behavior of a virtual character that learns, adapts, & adjusts behaviors to the cues from a human actor.

---

[1] https://www.techradar.com/news/gaming/microsoft-says-molyneux-s-milo-is-only-a-tech-demo-699668

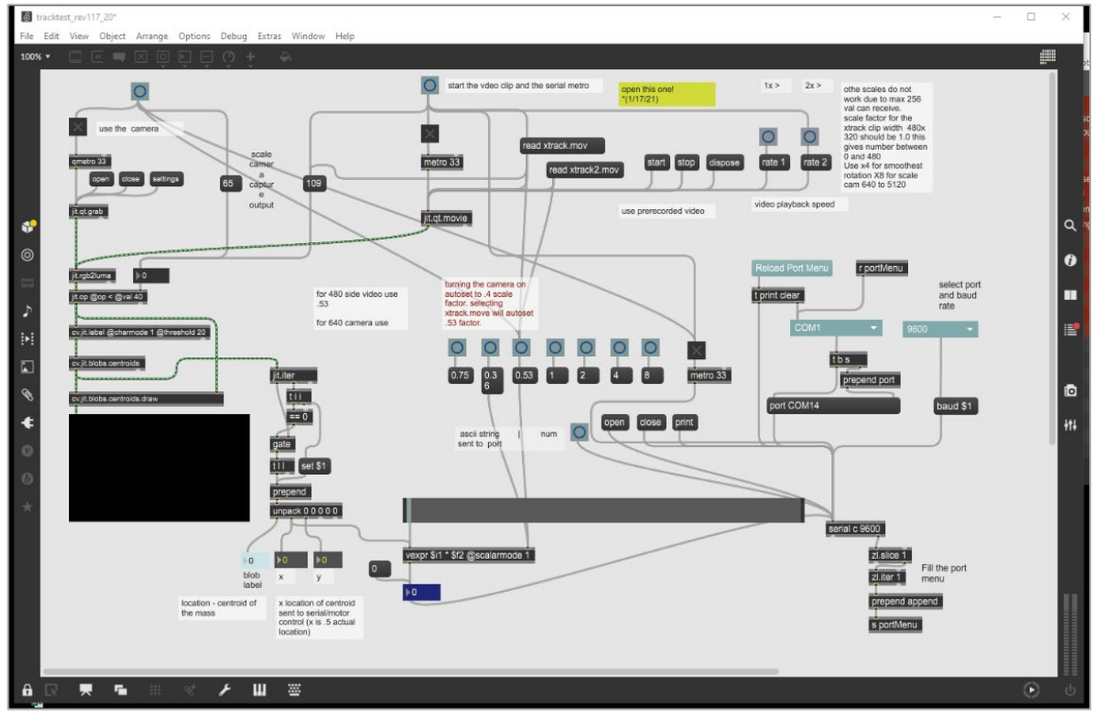
https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – for WHO

PI-Researcher has expert level software engineering & will conduct all experiments & data capture.

The need for an easy to understand & use interface is lessened.

Actual "presentation" mode interface



Raw production & programming in MAX MSP.

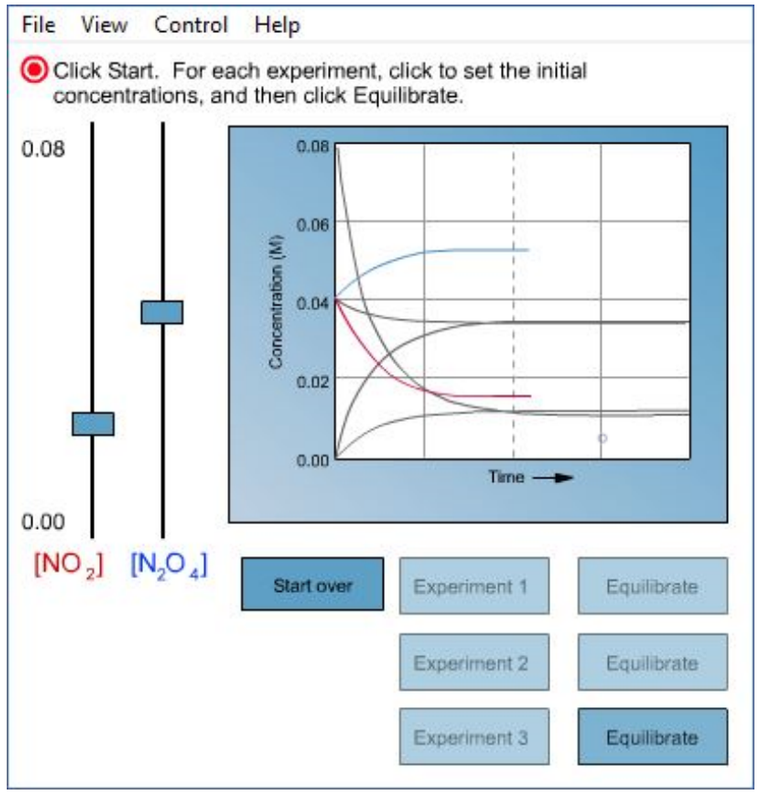
https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – for WHO

PI Researcher has expert level software engineering & will conduct all experiments & data capture.

Or

Novice level participant with zero knowledge nor desire to learn to use the system.



Most software needed for research applications exists between these two extremes.

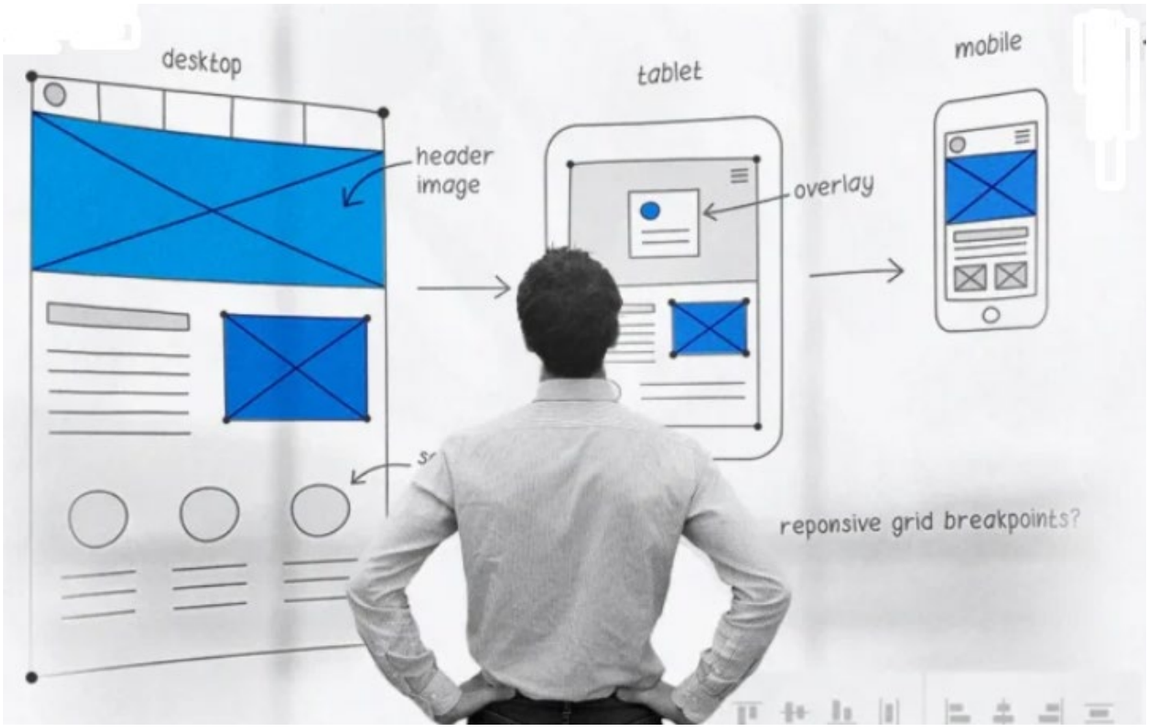
The interface is less of a concern the closer and more expert the user is to the process.

Novice chemistry student - interactive tutorial simulates a chemical reaction: UI structure and interaction strategy **must** provide appropriate affordances.

Clarity of UI elements and affordances is imperative.

https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – for WHO



Constraints & Affordances

Flow & sequences of actions.

Clarity of UI elements and affordances.

What data needs to be visible in capture time?

Don Norman: The Design of Everyday Things.

https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – for WHO

Constraints:

Technological – actions limited by what the technology will allow.

Primarily affects software engineering BUT may have profound consequences for the PI in obtaining usable results.

Physical – due to materials and human capabilities and limitations

Environmental – space, location, wireless, etc.

Cultural - expectations of behavior of the system and the participant.

Affordances:

The *affordances* of the environment are what it *offers* the animal, what it *provides* or *furnishes*, either for good or ill. ... It implies the complementarity of the animal and the environment.

Gibson, The Theory of Affordances.

User Experience – User interface considerations – tools

Raw command line interface

Windows Forms and Console UI

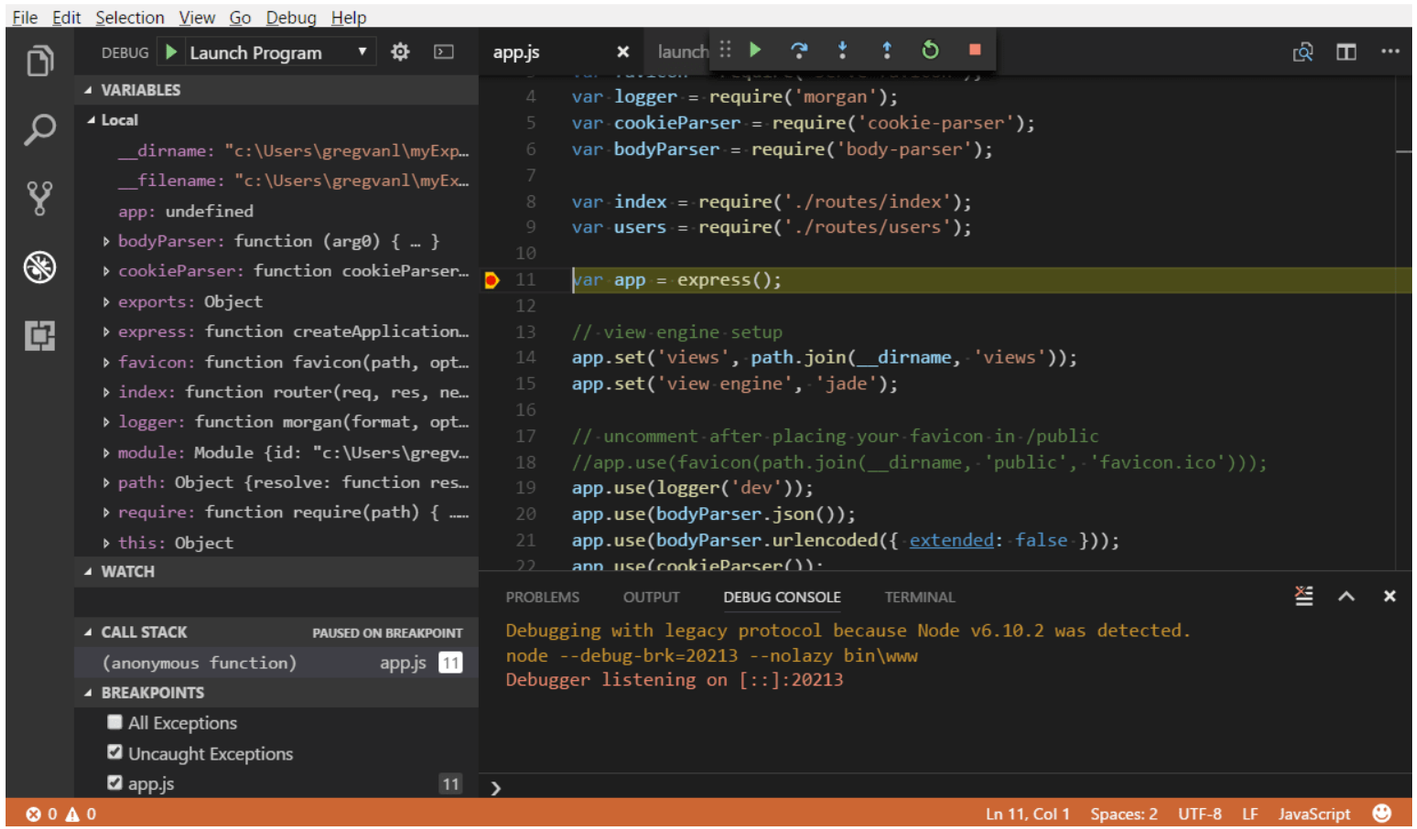
Windows Presentation Foundation

XAML UI

User Experience – User interface considerations – WHAT | WHERE

Raw command line interfaces

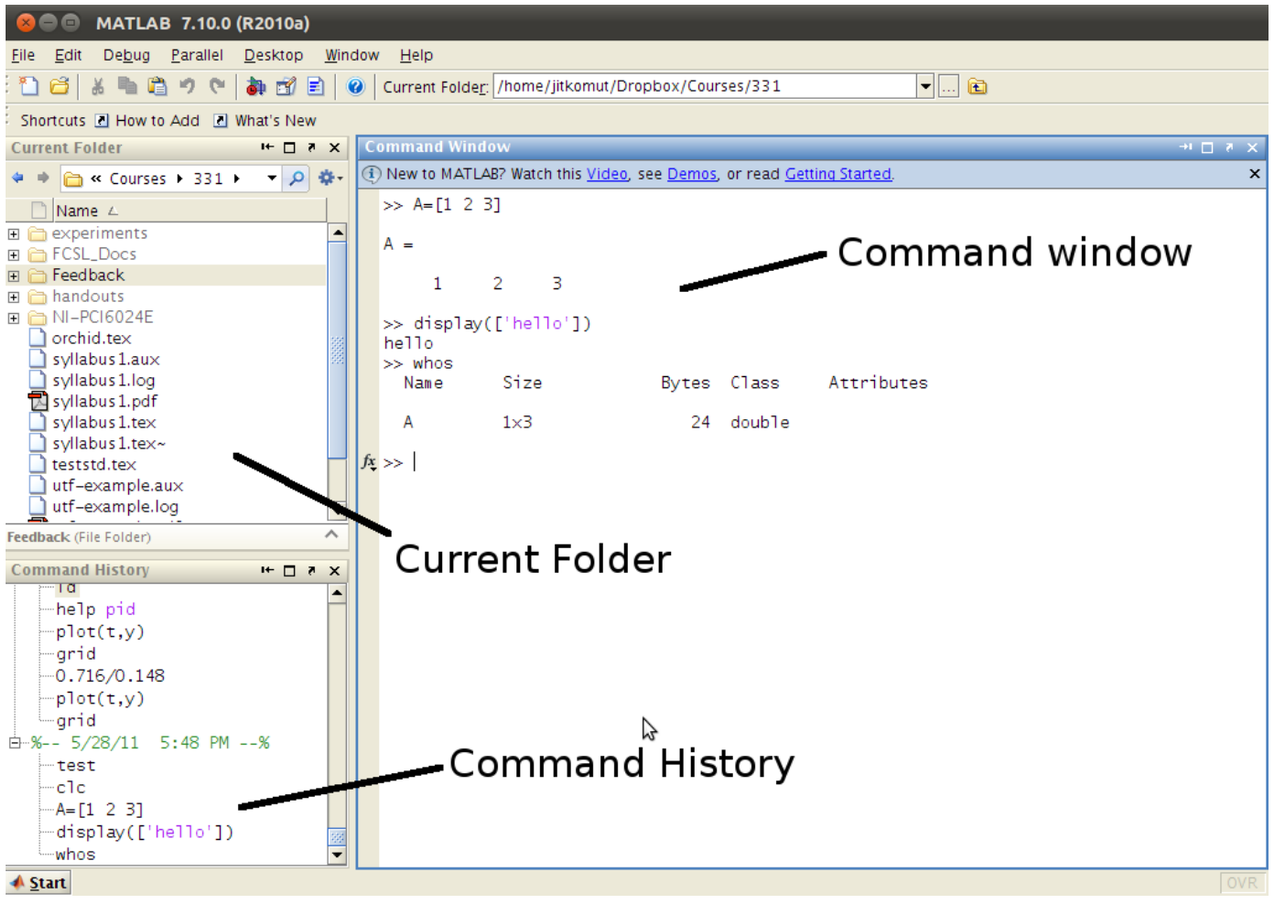
Node.JS in Visual studio



https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – WHAT | WHERE

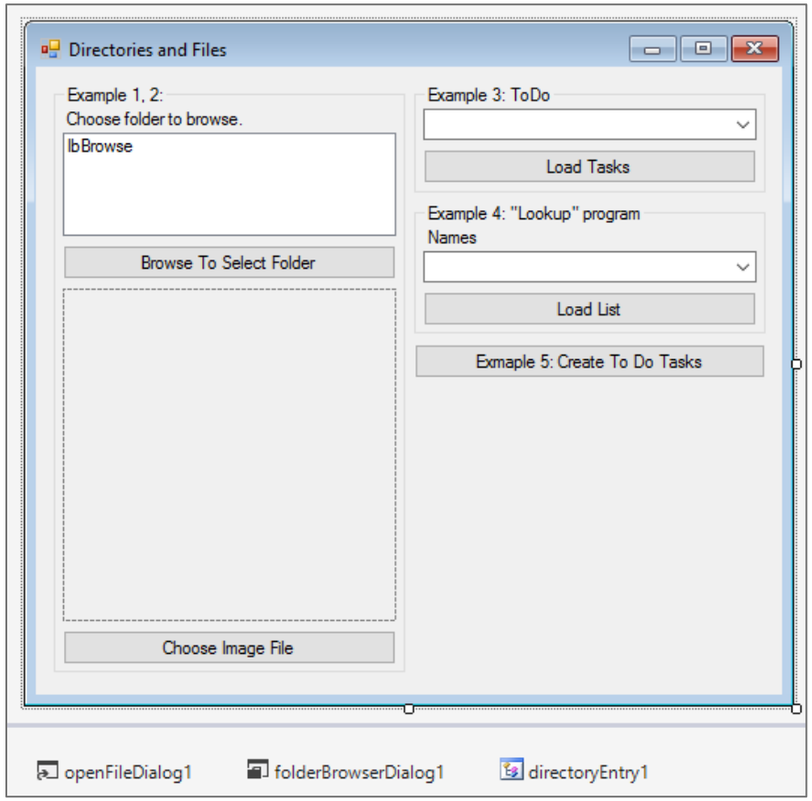
Raw command line interface – MATLAB



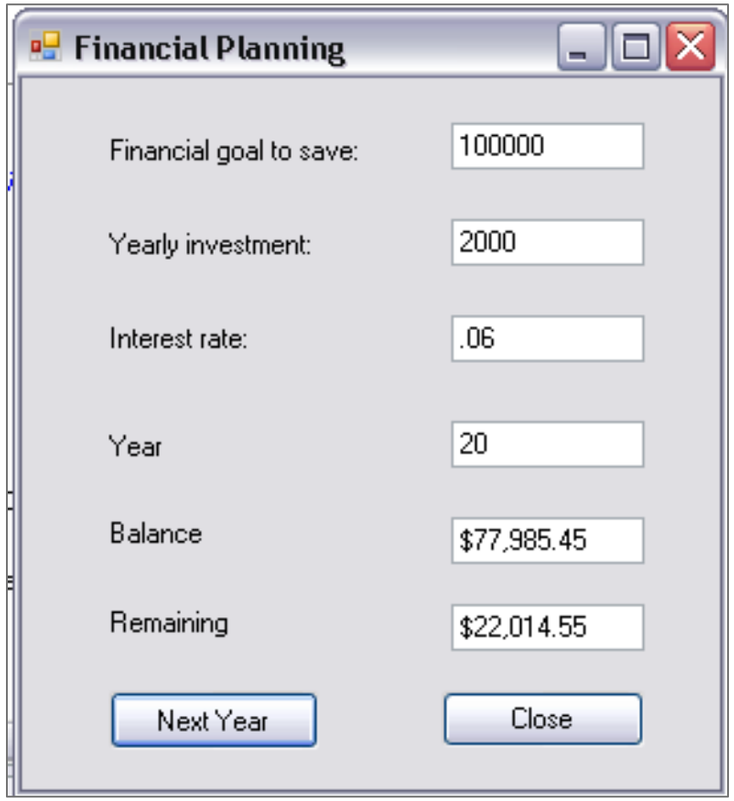
https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – WHAT | WHERE

Windows Forms



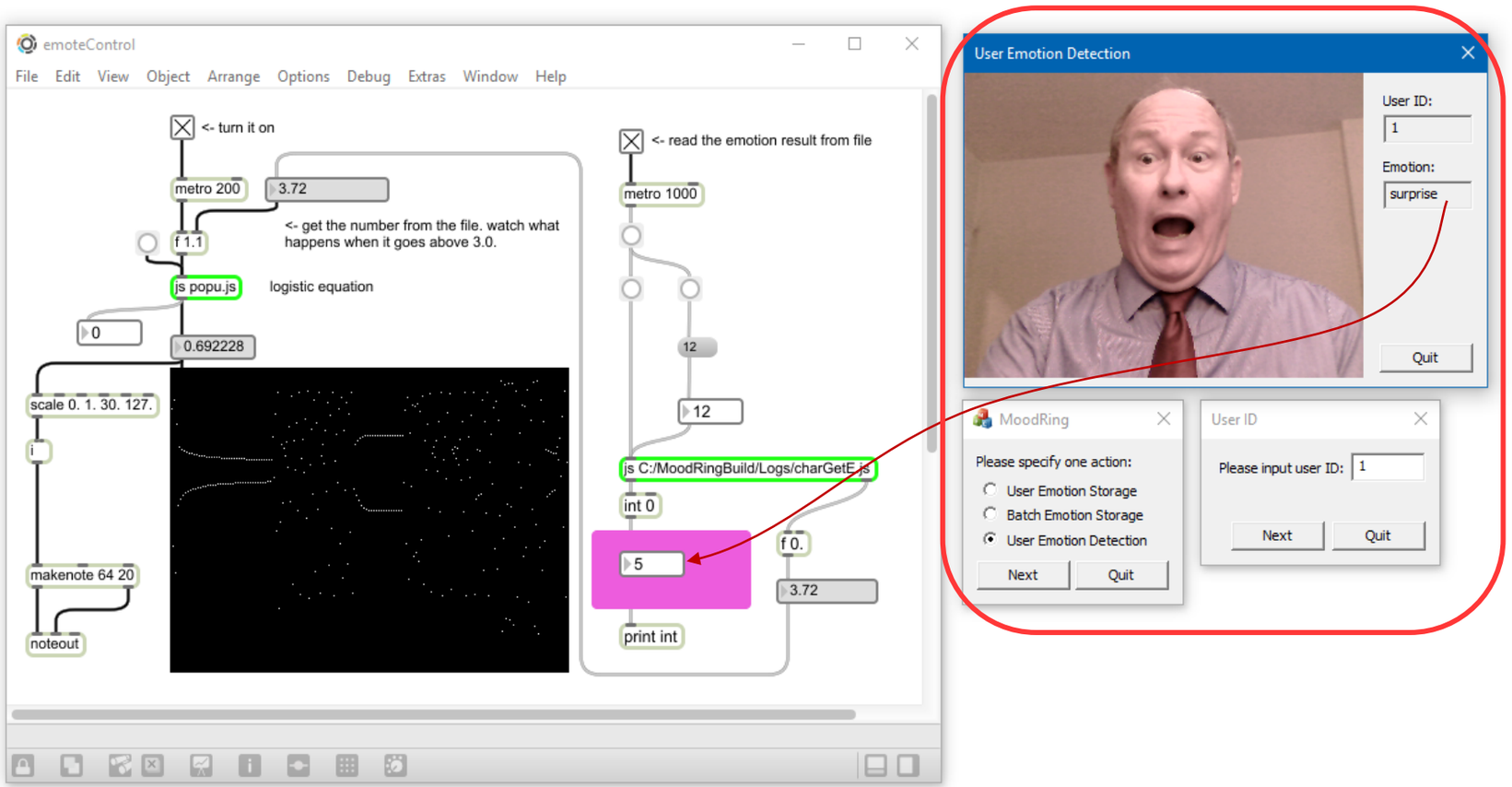
Console UI



https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – WHAT | WHERE

Windows Forms



https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – WHO | WHERE

Windows Presentation Foundation – XAML UI

For application environments used by many participants with varying degrees of expertise AND cultural backgrounds.

Medical practitioners

Clinicians

Graduate Students

Hired assistance

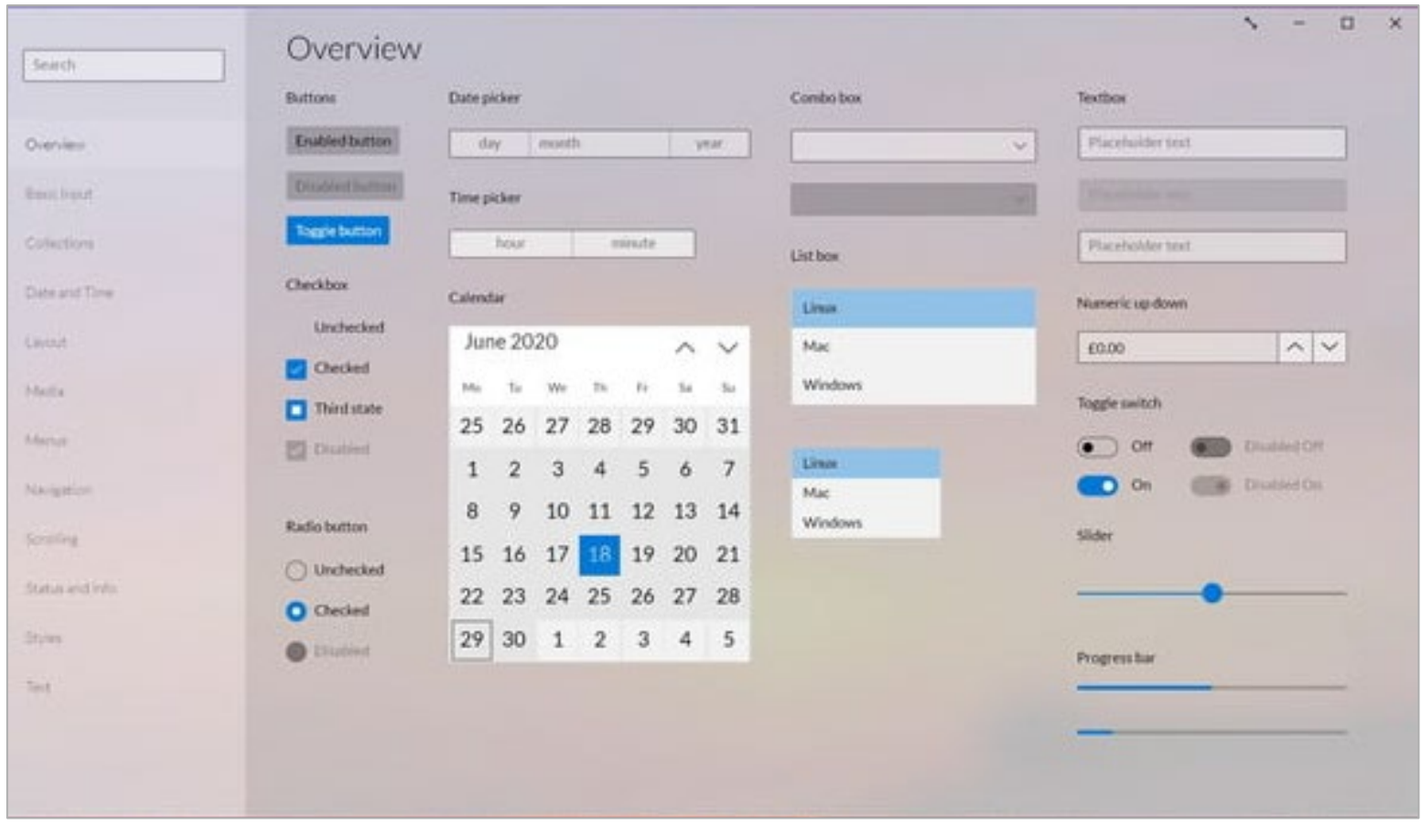
Average consumers

Allows for complete integration of code with user interface.

* *Requires both software engineering and user interface design skill sets*

User Experience – User interface considerations – WHO | WHERE

Windows Presentation Foundation – XAML UI – **BENEFITS**



Anyone with Web design HTML skills *can* master XAML

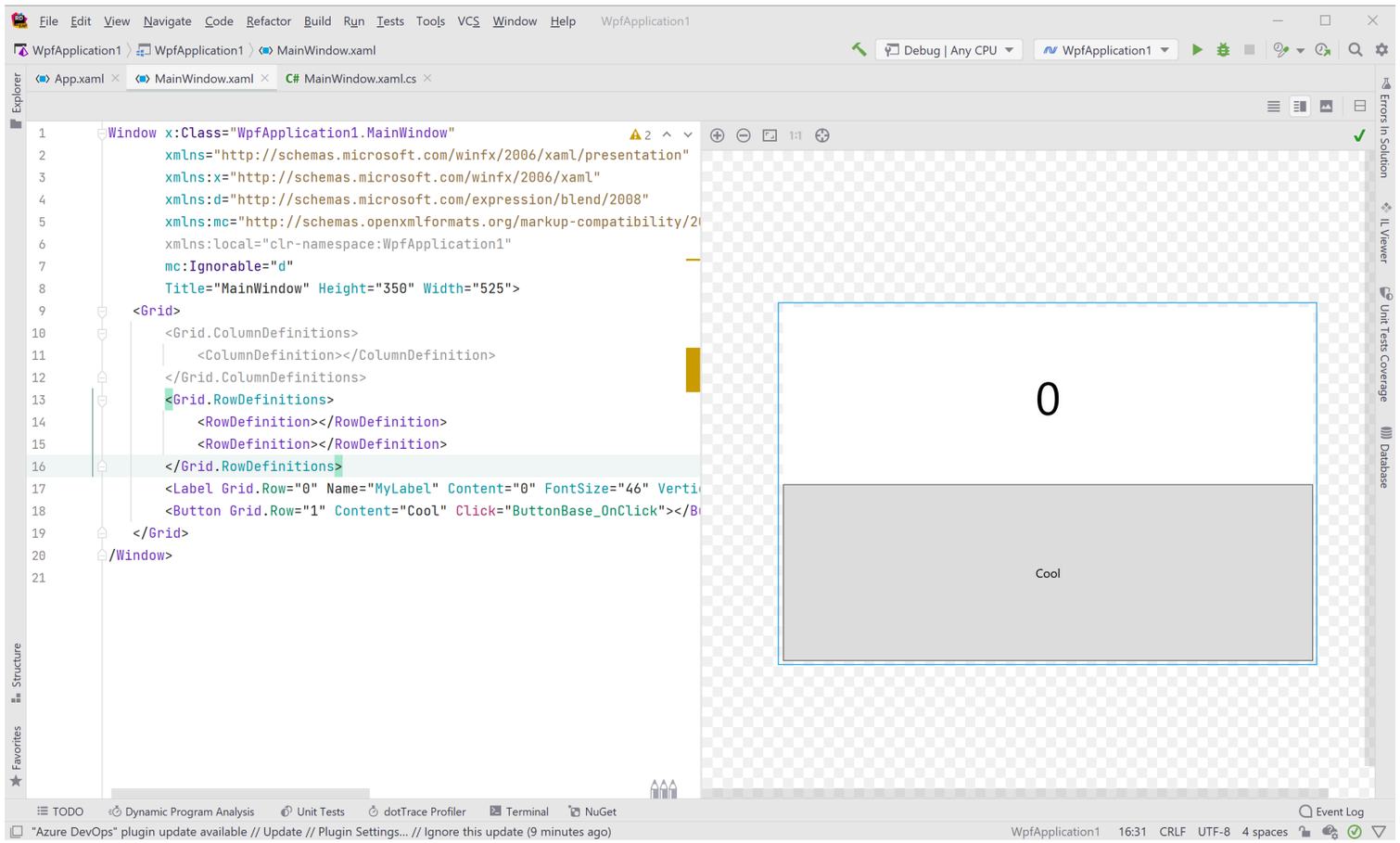
Similar syntax to HTML/CSS

Rich library of customizable interface elements

https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – WHO | WHERE

Windows Presentation Foundation – XAML UI – **BENEFITS**



Extensive libraries

Customizable for specific uses.

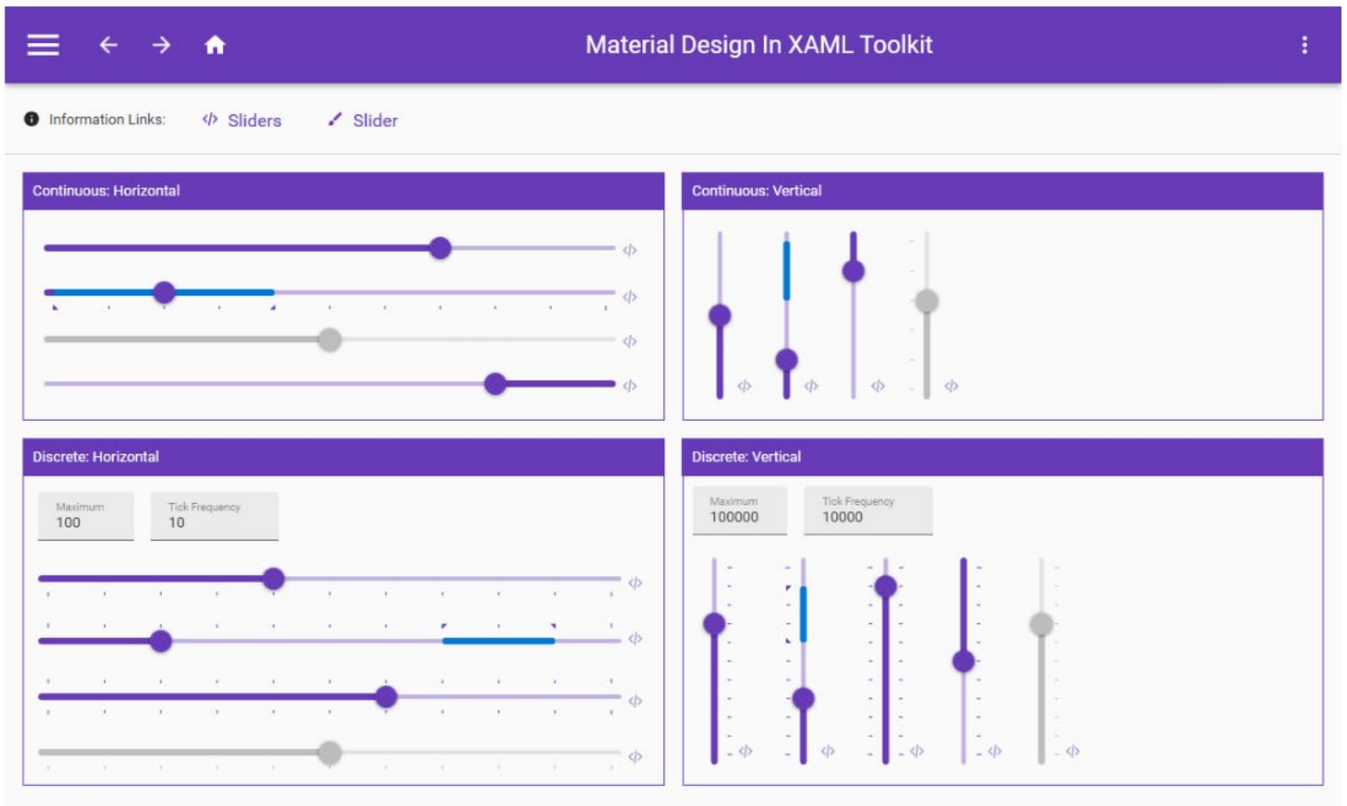
Icon set/font icons

Multiple language sets

https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – WHO | WHERE

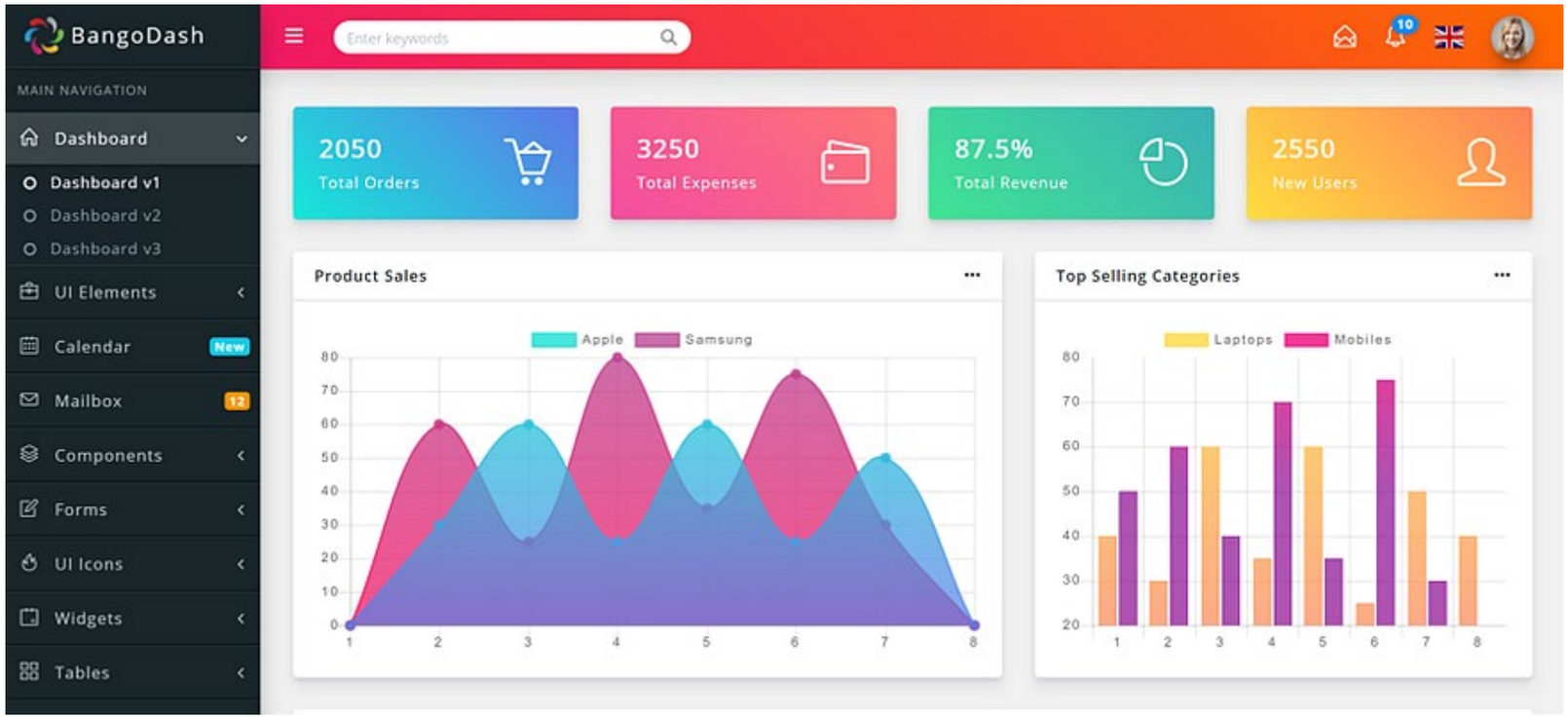
Windows Presentation Foundation – Material Design in XAML.



https://uxplanet.org/top-50-dashboard-ui-kits-and-templates-in-2019-8583e41b775d

User Experience – User interface considerations – WHO | WHERE

Windows Presentation Foundation – Material Design in XAML.



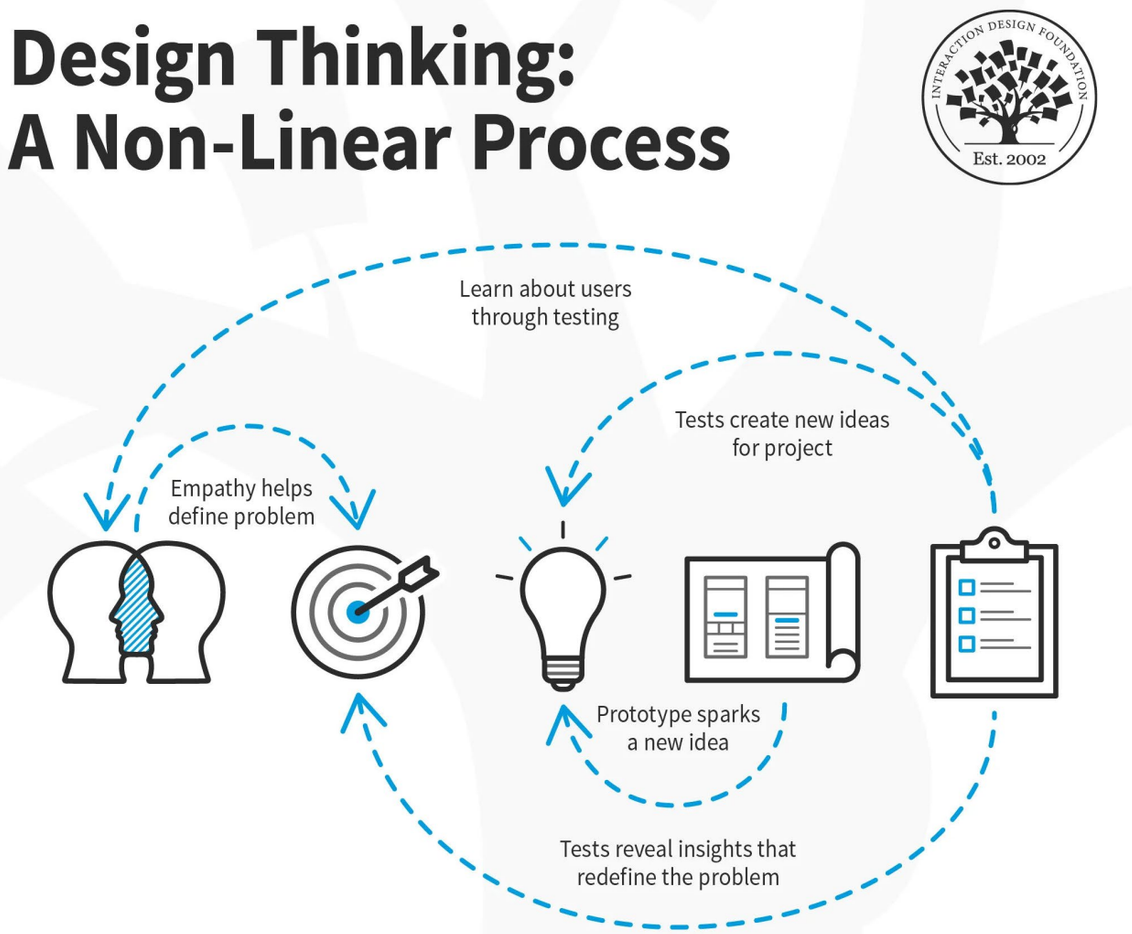
https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

User Experience – User interface considerations – WHO | WHERE

Windows Presentation Foundation – Material Design in XAML.

https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

Conclusions – Recommendations



Prototype on paper - to alleviate the preconceived notions of development team

Non-experts may perceive the affordances of the interface differently than the team expects

Site visits and observation

Test and refine interface

https://pensyl.com/p51research.html | https://pensyl.com/Design%20and%20User%20Experience%20in%20Software%20for%20Medical.pdf

Conclusions – Recommendations

Work with a competent team of experts

- Software engineers
- Coders
- User Interface designers

In university environments, graduate students in Computer Science *may* be accessible.

Upper lever undergraduate Design students and Graduate students will have the skills in UI and UX to assist.