

# WPF アプリケーションによる作業課題アプリの作成について

松原 篤

製作技術課

## 1 はじめに

グラフィカルユーザインターフェース(Graphical User Interface : GUI)は、(人の)体性感覚器からポインティングデバイスを通じてコンピュータ画面上に配列された部品に指令を送ることにより、目的の操作を遂行することができるインターフェースであり、携帯端末や遠隔操作における技術の基盤となっている。GUI を実装したアプリケーションは、高水準かつオブジェクト指向型のプログラミング言語により開発がなされており、開発言語は、デバイスに搭載された OS との相性や処理速度・計算量(Complexity)に鑑み、目的に応じて選定される。

本報告では、GUI アプリケーションとして、「短期記憶(Short-term Memory)」と「想起(Recall)」を行うことをねらいとした作業課題アプリの作成について述べる。作業課題としては、制限時間を設定しつつ、① 画像とそれに対応した名称、② 1 文字(英文字)ごと、無作為に選出し構成された「意味をもたない(Non-Semantic)」文字列、をそれぞれ題材としたもの(2種類のアプリ)であり、言語はWPF アプリケーションを開発ツールとする Visual Basic (ver. 2019)とした。

なお、上記画像および文字列を扱った作業課題アプリを、以降、それぞれ「作業課題 1」および「作業課題 2」と呼称する。また、作成においては、学生への研究資源および開発指導とした「勉強会」という形態の下、実施した。

## 2 WPF アプリケーション<sup>1)</sup>

WPF アプリケーション(Windows Presentation Foundation)は、① ボタンやテキストボックスといった設定や属性、またイベントに関わる機能をもつ各種コントロール(オブジェクト)、② 図形の描画に関わるグラフィクス、③ 映像・音声処理、④ 構成に関わる各種レイアウトなど、ユーザインターフェースを開発するための各種機能を揃えたフレームワークである。WPF では、デザイン画面については、XAML(Extensible Application Markup Language)と呼ばれるマークアップ言語を用いて設計され、配置した各種コントロールの機能については、Visual Basic や C#等のオブジェクト型指向言語を利用することにより、実装される。また、プログラムの実行は「.Net Framework」と呼ばれるプラットフォームの下で可能である。

## 3 作業課題 1 について

画面設計に関しては、① 作業に関する注意事項等を示した「イントロダクション」、② 制限時間内に記憶と想起(解答)が遂行できる「作業課題提示」、③ 課題に対する結果および成績(正解率)を表示する「結果」、④ 課題遂行者(以降、作業者)に対する順位付けを示した「ランキング」の合計 4 つの画面(Window クラス)から構成される。また、各画面を表示するタイミングは、上記①～④に対し昇順としている。

### 3.1 イントロダクション画面

画面は、注意事項を表示するための TextBlock、課題に関する種別(カテゴリ)を選択することができる ListBox、作業課題提示画面に移行するための Button など、各種コントロール(クラスのオブジェクト)により構成している(図 1 参照)。レイアウトに関しては、Grid クラスにより、行・列に関するプロパティ(Row/ColumnDefinitions)を通じて各コントロールの要素的な配置を実現している。

作業課題となる画像データの読込については、ListBox により選定されたカテゴリ名に対応するフォルダが自動的

に参照され、フォルダ内の画像データファイルが、出題(課題)数分、ランダムかつ重複することなく抽出される仕組みとなっている。

### 3.2 作業課題提示画面

図 2 に示す作業課題提示画面では、上段に、① データ設定(Data Setting)、作業開始(Start)、ランキング(Ranking)、プログラム終了(End)を行うための Button、② 制限時間や指示(記憶/想起)に関する表示、ならびに作業名の入力が可能な TextBox、などの各種コントロールを配置している。

中段では、画像と、それに対応した名称が上下に位置する形をデータセットと呼称する場合、データセットを横並びに 5 つ表示し、それを 2 段設けることにより、合計 10 個のデータセットが表示できるよう、Grid 分割している(画像:グレー領域、名称:橙色、空色領域)。また、最下段(桃色領域)では、画像データの引用先情報を表示できるようにしている。

### 3.3 課題提示に関する仕掛け

作業課題におけるプロトコルの骨子としては、① 作業開始(Start ボタン)後、最初の課題(10 個の画像とその名称)に対し、指定された制限時間内に記憶をする。次に、② 記憶した画像データのみが表示されるため、想起を行いながら、制限時間内にそれぞれの画像に対応する TextBox に名称を入力する。ただし、想起時に提示される画像データは、記憶時と異なり、ランダムに配置されるようにしている。以上、上記①および②の作業を 1 課題とし、それを繰り返すことにより、作業課題では、作業者は複数回の課題を、画像が重複して提供されることなく遂行することができる。

タイマーについては、名前空間「System.Windows.Threading」に属する DispatcherTimer クラスを利用する。本クラスは、WPF アプリケーション上で動作可能なタイマークラスであり、設定時間ごとに呼び出されるメソッド(サブ関数)の内部(別スレッド)から作業課題提示画面に配置された各種コントロールへのアクセスが行えるよう、デリゲートシステムを容易に、かつ自動的に構築できるクラスである。本作業課題では、2 つのタイマーを用意し、記憶時と想起時でその都度、動作のスイッチングを行っている。

課題間、あるいは記憶、想起間の画像(および名称)の切替えは、表示/非表示の機能をもつ Grid クラスの Children プロパティの Add/Remove メソッドにより実現している。メソッドを働かせる要素は、画像データの場合は、BitmapImage クラスによりビットマップ形式に変換された画像データをソースとした Image クラスのオブジェクト(コントロール)であり、名称は前記 TextBox クラスのオブジェクトとなる。

### 3.4 結果およびランキング画面に関して

結果画面については、出題数分、すべての課題が終了次第、図 3 のような構成の下、中央部に大きく位置する TextBox に、① 全課題数に対する平均正解率、② 課

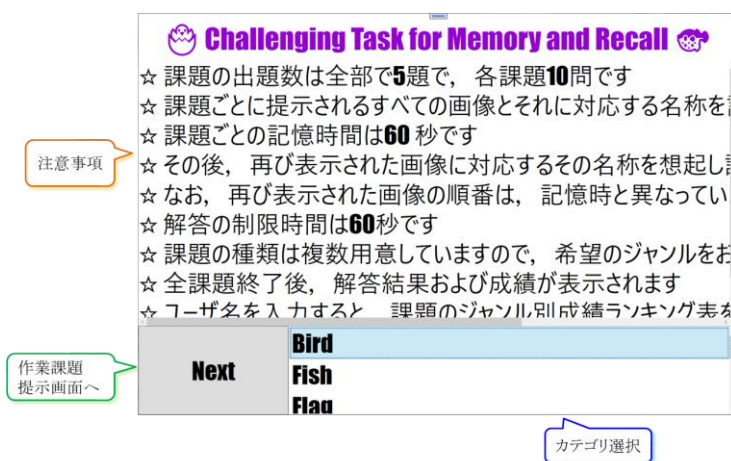


図 1. イントロダクション画面

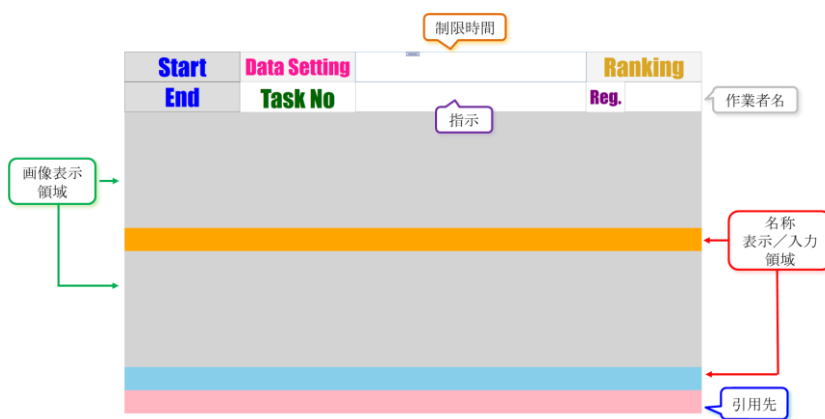


図 2. 作業課題提示画面

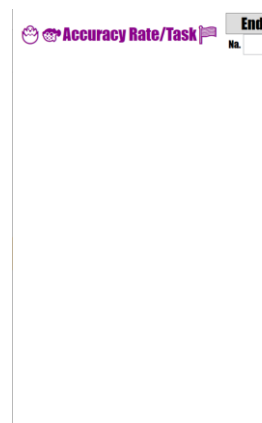


図 3. 結果画面

題ごとの正解率, ③ 課題ごとの内容として正解, 回答, 正誤判定, といった結果に関する情報が表示される. これらの情報は統計分析等で再利用できるよう, ファイルに保存している.

ランキング画面については, 詳細は割愛するが, 作業者の順位は, 該当するカテゴリに対する過去の作業者の成績(平均正解率)を考慮し位置付けられる.

### 3.5 作業例

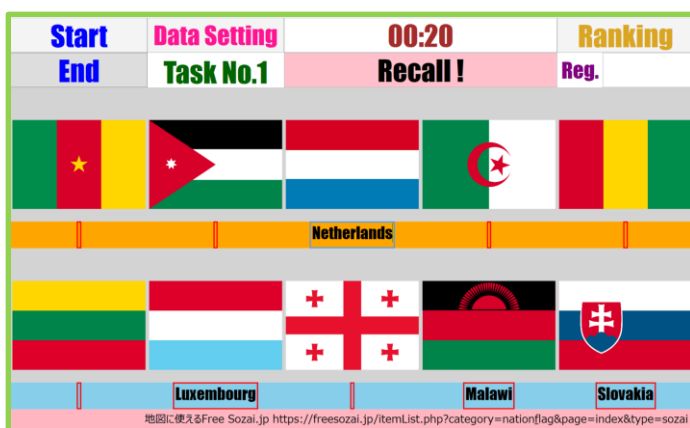
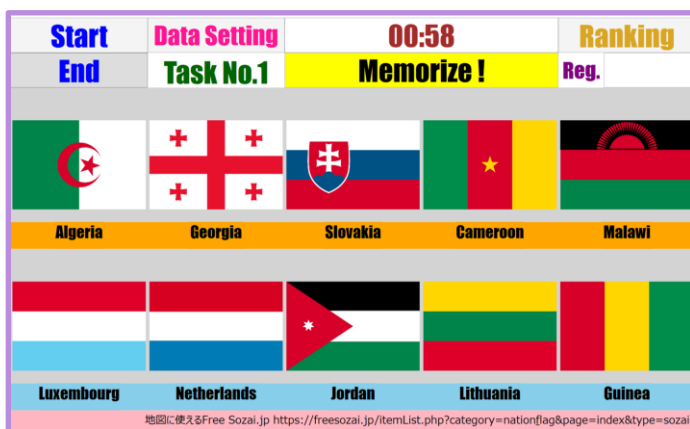
作業課題の設定条件としては, 出題(課題)数 5, 1 課題の画像数 10, 記憶および想起に関する制限時間は共に 60 秒とし, 作業者は著者である. 条件下では, 各カテゴリに格納されている画像データ 100 個に対し, 合計 50 個(1 課題 10 画像×課題数 5)のデータが, ランダム, かつ重複することなく選出される.

図 4 はカテゴリとして「Flag(国旗) 2)」を選択したときの, 作業に関する事例である. 図の 2 つの課題例(課題 1 および 4)において共に, 上図が記憶(Memorize)を, 下図が想起時に解答(Recall)を, それぞれ遂行している状態を示しており, 特に解答に関しては, 課題 1 および 4 において, 「Netherlands」および「Philippines」と, それぞれ入力した直後を示している.

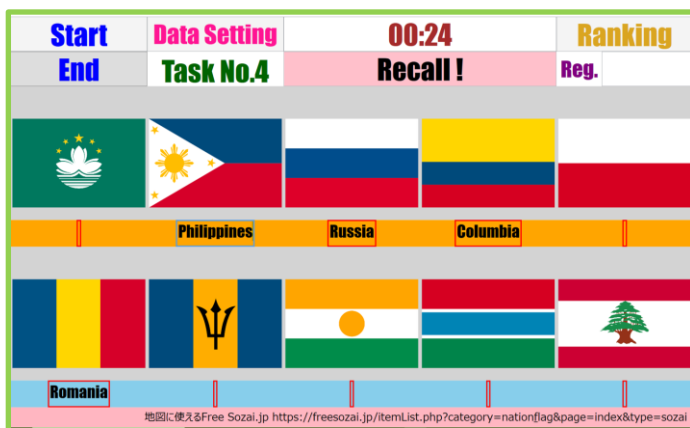
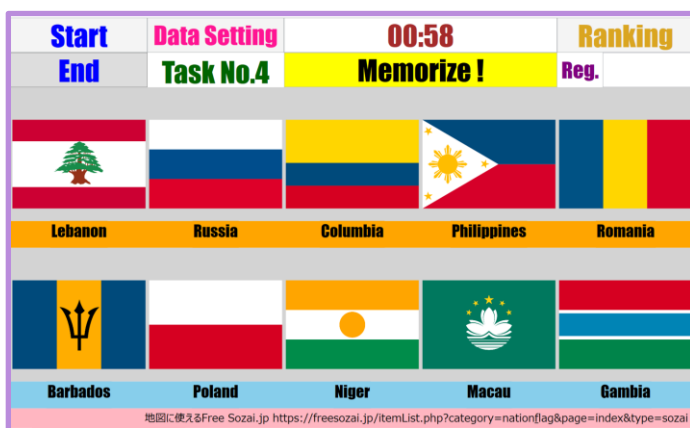
図 5 は, すべての課題を遂行した後, 表示される結果画面(表示は一部)であるが, 参考まで, 5 つの課題に対する平均正解率は 48%であり, 内訳として課題ごとの正解率[%]はそれぞれ 50,

Accuracy Rate/Task		End
No.	Test	
Average of Accuracy Rate: 48.00%		
Task No.1; Accuracy Rate:50%-----		
Image No.1:	Cameroon Cameroon	<input type="radio"/>
Image No.2:	Jordan Jordan	<input checked="" type="radio"/>
Image No.3:	Netherlands Netherlands	<input type="radio"/>
Image No.4:	Algeria Alge	<input checked="" type="radio"/>
Image No.5:	Guinea	<input checked="" type="radio"/>
Image No.6:	Lithuania	<input checked="" type="radio"/>
Image No.7:	Luxembourg Luxembourg	<input type="radio"/>
Image No.8:	Georgia	<input checked="" type="radio"/>
Image No.9:	Malawi Malawi	<input type="radio"/>
Image No.10:	Slovakia Slovakia	<input type="radio"/>
Task No.2; Accuracy Rate:50%-----		
Image No.1:	Ireland Ireland	<input type="radio"/>
Image No.2:	Singapore Singapore	<input type="radio"/>

図 5. 結果画面(Flag)



(a) 課題 1



(b) 課題 4

図 4. 作業課題(Flag) [上/下 : 記憶/想起]

50, 50, 40, 50 であった。

また、別のカテゴリ「Bird(鳥類)<sup>3)</sup>」を選択したときの、作業課題提示および結果の画面例を、それぞれ図 6 および図 7 に示す。



図 6. 作業課題(Bird:課題3)[上/下 : 記憶/想起]

図 6 は 3 番目の課題例であり、想起時においては、「ベニマシコ」と入力・変換した直後の状態を示している。結果として、5 つの課題に対する平均正解率は 62%，課題ごとの正解率[%]はそれぞれ 60, 70, 60, 60, 60 であった。

なお、Flag(国旗)の課題については、Bird(鳥類)の場合と異なり、解答を英語表記で行うものにしたが、画像認識や単語のスペリングといった複数の能力を必要とするため、課題設定としては、比較的難易度が高いように思われる。



図 7. 結果画面(Bird)

#### 4 作業課題 2 について

画面設計については、「作業課題 1」を大筋準拠したものとなっているが、① 注意事項を示す「イントロダクション」と、② 結果やランキングの表示も含む「作業課題提示」の 2 画面構成として簡素化している。

##### 4.1 課題に関する仕掛け

作業課題は、英文字(大小 26 文字→合計 52 文字)において、1 文字ずつ、重複することなくランダムに選出し構成された「意味をもたない(Non-Semantic)」文字列としている。ただし、「重複がない」とは、1 課題の文字列において、同一の文字が含まれないことであり、課題ごとでは含まれる可能性があることを意味している。

また、想起時の解答は、提示された課題の文字に対応するキーを単に入力して遂行できるものではなく、先に報告した、入力に活用したキーの(英)文字に対し ASCII コードにしたがい割り当てられた

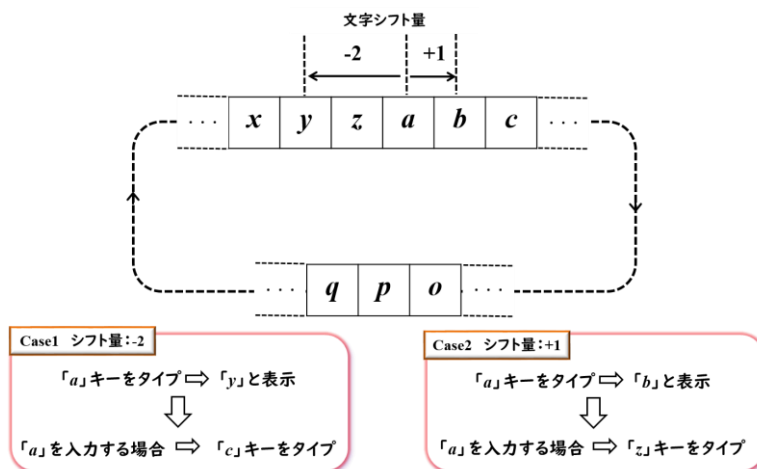


図 8. 文字シフトの原理

数値を、四則演算を通じて変化させ、入力キーと異なる文字を出力させる「文字シフト」機能<sup>4)</sup>を考慮する必要があるものとしている(大文字「A」～「Z」はコード値「65」～「90」に、小文字「a」～「z」は「97」～「122」にそれぞれ対応)。

文字シフトの原理を図8に示す。図のように、英文字に対し序列に従う両端の文字(小文字の場合は「a」と「z」)については、隣接関係を構築することにより、文字間に Cyclic な関係性を施すと共に、文字の「シフト量」は入力キーに対し出力を前後の文字にシフトさせる量のことであると定義した上で、シフト量の設定は大文字と小文字で区別をしている。例を挙げると、大文字、小文字のシフト量をそれぞれ、 $-2$ 、 $+1$ とした場合、文字「A」を出力するには「C」キーを、文字「a」を出力するには「z」キーを、それぞれ入力として活用しなければならない。特に小文字の例では、「z」キーのコード値 122 にシフト量 $+1$ を加えたものから、英小文字の総数 26 を差し引くことにより、コード値 97 の文字「a」の出力を実現している。したがって、文字列「aZY」と出力する場合は、「zBA」と入力すればよい。

タイマーは、作業課題 1 の場合と同様、Dispatcher Timer クラスを活用するが、1つのクラスオブジェクトの時間経過にしたがい、記憶および想起、それぞれの時間帯を区分して設けることにした。

結果の表示については、作業者のモチベーション向上の効果も期待し、現在進行形の課題に対し記憶時に直前の課題の結果を表示する仕組みとした。なお、前課題の終了後の結果やランキングの表示形式については、作業課題 1 と同等な形となっている。

#### 4.2 作業例

設定条件としては、課題数5、各課題の文字列の長さをそれぞれ、3, 3, 4, 4, 5 とし、シフト量はそれぞれ(大, 小) $=(-1, +1)$ ,  $(+1, -1)$ ,  $(+1, -2)$ ,  $(-2, +2)$ ,  $(+1, -2)$ として、難易度を段階的に上げるようにした。また、1課題の持ち時間を 60 秒とした上で、記憶および想起(解答)の制限時間をそれぞれ 20 秒、40 秒と区分した。

図9は作業例(課題2および5)であるが、作業課題1のときと同様、各課題の上図が記憶時を、下図が想起(解答)時の状態をそれぞれ示している。課題2については、シフト量は大文字が $+1$ 、小文字が $-1$ のため、解答としては「kNm」と入力することにより、課題通りの出力「jOl」が実現している様子がわかる。一方、課題5については、課題

<b>Start</b>	Limited (s)	59	Ranking
<b>End</b>	AccountName	<input type="text"/>	
<b>Info.</b>	Please memory the presented letters		
<b>Task</b>	jOl		
<b>Ans.</b>	<input type="text"/>		<b>Result</b> Task1: gZ Answer1: gZ Judgment: Correct!

<b>Start</b>	Limited (s)	14	Ranking
<b>End</b>	AccountName	<input type="text"/>	
<b>Info.</b>	Please typing the memorized letters following the shift rule Shift information Upper: 1 Lower: -1		
<b>Task</b>			
<b>Ans.</b>	jOl		<b>Result</b>

(a) 課題 2

<b>Start</b>	Limited (s)	60	Ranking
<b>End</b>	AccountName	<input type="text"/>	
<b>Info.</b>	Please memory the presented letters		
<b>Task</b>	cdjTN		
<b>Ans.</b>	<input type="text"/>		<b>Result</b> Task4: GqoE Answer4: GqoE Judgment: Correct!

<b>Start</b>	Limited (s)	1	Ranking
<b>End</b>	AccountName	<input type="text"/>	
<b>Info.</b>	Please typing the memorized letters following the shift rule Shift information Upper: 1 Lower: -2		
<b>Task</b>			
<b>Ans.</b>	cdiTN		<b>Result</b>

(b) 課題 5

図9. 作業課題(文字列) [上/下 : 記憶/想起]

「cdjTN」に対する解答「cdiTN」は、3 番目の小文字に対し、小文字のシフト量「-2」を考慮し、本来は「1 (エル)」キーを活用し文字「j」を出力させなければならないところを、誤って「k」キーにより文字「i」を出力させたため、不正解となった状態である。また、記憶時において、図では右下の箇所、課題 2 に対しては課題 1 の、課題 5 に対しては課題 4 のそれぞれ結果が共に正解として出力されている様子も伺える。

全体の結果としては、全 5 問に対し 3 問が正解で、課題 3 及び上記 5 が不正解であったため、正解率は 60%であった(図 10 参照)。

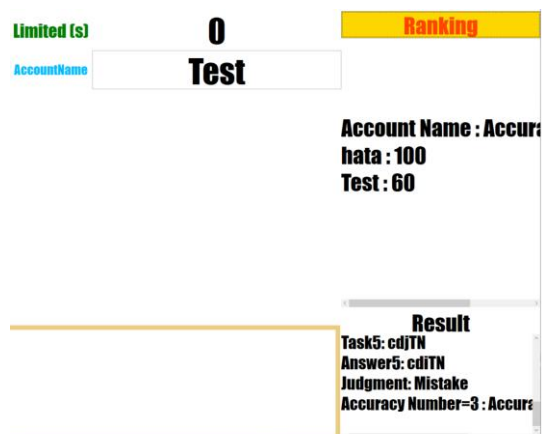


図 10. 結果(一部)およびランキング表示

以上、2 つの作業課題の難易度の設定に関しては、課題コンテンツの提示法や制限時間といった諸条件を、目的に応じて調整すればよいと思われる。

## 5 おわりに

本報告の作業課題アプリの作成については、著者と研究生 2 名の計 3 名とした座組において、「アプリ開発勉強会」と題して実施した。実施期間は本年度 4 月～12 月(計 30 回:3 時間程度/回)である。学生諸君は WPF アプリケーションの開発だけでなく Visual Basic 言語によるプログラミングも未経験であったため、進行については、まずは、GUI の定義から開始し、① 各種コントロールとイベント処理、② 複数ウインドウ画面の導入と起動、③ レイアウトに関して～分割及びコントロールの格納、④ タイマークラス、⑤ 入出力インターフェースなどに触れつつ、作成に対する基本的な関連事項の習得に努めた。

次に、作業課題ごとに核となる遂行事項として、作業課題 1 では、① 画像データや関連情報の読み込み、② 各クラスオブジェクト(コントロール)の動的配置、③ 画像データの選出と提示(記憶/想起)法などがあり、作業課題 2 では、① 文字列(Non-Semantic)の生成、② 文字シフトの考え方と実装法等が、また作業課題の共通事項として、結果の表示と保存などが挙げられるため、この辺りの具体的な実装については、各学生の研究内容に適合したプランに基づき、適宜、サンプル的、かつ種となるような試行型のプログラムを作成し検証を行いながら段階的(step by step)に遂行した。

最後に、勉強会という形態に賛同を戴き、作業課題アプリの開発に取り組んで戴いた、本大学大学院創成科学研究科西藤聖二准教授の研究グループである電気電子情報系専攻 修士 1 年生 畑山明莉氏(担当:作業課題 1)、ならびに同 1 年生 吉永直樹氏(同:作業課題 2)には、この場を借りて御礼を申しあげると共に、今後、作業課題アプリを一層有益なものに拡張・発展されていくことを期待する次第である。

## 参考文献

- 1) Microsoft .Net のドキュメント Windows Presentation Foundation (.NET Framework) :  
<https://docs.microsoft.com/ja-jp/dotnet/desktop/wpf/?view=netframeworkdesktop-4.8&preserve-view=true>
- 2) 地図に使える Free Sozai.jp : [https://freesozai.jp/itemList.php?category=nation\\_flag&page=index&type=sozai](https://freesozai.jp/itemList.php?category=nation_flag&page=index&type=sozai)
- 3) 日本の野鳥識別図鑑 : <https://zukan.com/jbirds/>
- 4) 松原篤, 生体信号に影響を与える作業課題の開発, 山口大学技術部技術報告集, 第 15 巻, pp.26-29 (2015)