

重定ゼミナール

タイピングゲーム・ソフトウェアの開発

4年J組 01G0912

角田友昭

<目次>

1 . はじめに	3
2 . 作品のねらいと差異性	4
3 . ユーザーが自ら自由な単語を表示させて、それをタイプできる利点	5
4 . 開発ツール	5
5 . タイピングゲームの仕様	8
6 . ソースコード解説	14
7 . 課題点	33
8 . 参考文献	34

## 1. はじめに

コンピュータは現在、日常生活のうえで欠かすことのできないものである。ほんの数十年前までは、コンピュータを使うのは研究者、専門家、企業であり、コンピュータを使える、ただそれだけで羨望の眼差しを受けた。しかし現在ではコンピュータを使うのが当たり前であり、時と場合によってはコンピュータを利用しないと不利な立場、不便な場合になりうる。会社の給与計算、給与明細は手書きからコンピュータ処理に変わり、大学生のレポートも手書きからコンピュータを利用して提出されるようになった。また、その就職活動においてもコンピュータを通じてインターネットで行われる時代であり、コンピュータを使えない学生にとっては就職活動でさえ満足にできない時代である。そして、コンピュータ学習は今や、高等学校、小学校でも行われるようになっている。しかしコンピュータを利用する事が当たり前になった現在でも、少なからずコンピュータを使えない人がいる。そのようなコンピュータ初心者の方々にコンピュータを利用して、コンピュータの利便性や必要性を感じてほしい。そしてコンピュータに少しでも興味を持ってくれたら私は嬉しいと思ったため、本研究をタイピングゲーム・ソフトの開発にした。

実は私自身、コンピュータが大嫌いだった。大学のコンピュータの講義も憂鬱でレポートも手書きで提出していた。ワードソフトも使いこなすことができず、手書きの方が早かったのである。しかし今後のためにワードぐらいは学習しようと思い、タイピングソフトを利用した。その結果、ブラインドタッチができるようになり、さらにコンピュータへの学習意欲、関心が深くなっていた。そして今では、ソフトウェアを利用するだけでなく、プログラミング言語を用いてソフトウェアを開発するまでになった。

このようなことからコンピュータ学習の第一歩として、多くの人は、まずキーボードから文字を打つタイピングから学習を始める。このタイピングはコンピュータを利用する上で最も基本的で重要なことである。本研究ではその基本的なタイピング学習用ソフトを自ら開発する。これによってタイピング技術の向上をはかり、その後のコンピュータ学習をスムーズに行えるようタイピングを学習しコンピュータ学習の基礎を作る。

## 2. 作品のねらいと差異性

上の<はじめに>でも述べたとおり、コンピュータ初心者におけるタイピング技術の向上を目的とするソフトウェアの開発である。そして、そこから少しでもコンピュータに関心をもってもらい、その後のスムーズなコンピュータ学習に発展していければよいと思っている。

確かに現在、世の中にはさまざまなタイピングゲームが存在する。フリーソフトとして手軽に手に入り利用できるものや、Web 上で楽しめるもの。また、市販されているソフトなどは圧倒的な視覚的効果で普通のシューティングゲームをやっているような感覚になるものでさえ存在する。これら、さまざまなタイピングゲームで共通することは、タイプする文字や単語が画面に表示され、その文字や単語に従ってユーザーがタイピングしていく形式である。一般的なタイピングゲームはそれを開発する人がタイプする文字や単語をあらかじめ決めて表示される。ユーザーにとってみれば、タイプする文字や単語が最初から決められていて、それを打ついかなければならない。そこで私は今までにない新しい形のタイピングゲームとして、ユーザーが自身でタイプしたい単語や文字を恣意的に表示させ、それをタイプするタイピングゲームを開発する。そこで今までになかった新しいタイピングゲームを完成させ、他のタイピングゲームとの差異化を図ろうと思った。このことによる利点は後述する。

### 3. ユーザーが自ら自由な単語を表示させて、それをタイプできる利点

単語データベースを txt ファイルにすることによって、ユーザーが自由に単語データベースを操作できるようにした。これによってユーザーは自分がタイプしたい単語や文字を自由に表示させ、それをタイプしていくことを可能にする。

タイピングは人それぞれの、打ち方、癖がある。特にローマ字入力の方法において、「じ」を「zi」や「ji」と入力したり、「しゅ」を「shu」や「syu」と入力したりする。この場合、まったく初めてタイピングを学習する人にとってはそれほど気にならないことであるが、ワードソフトなどである程度ローマ字入力を学習した人にとっては、それなりに自分のタイピングの癖がある。そのため自分が好きな入力方法でタイピングできるように、単語データベースを txt ファイルにし、ユーザーが自由にそれを操作できるようにした。

一般に販売されているものやフリーソフトのタイピングゲームにおいて、確かにユーザー自身が好むローマ字入力に対応したタイピングゲームが存在する。ゲームのオプション画面で、自分が好むローマ字入力方法を設定できるゲームや、設定に関係なくプログラムがユーザーの入力方法の全てに対応しているゲームなどがある。しかし本研究のタイピングゲームは入力方法だけでなく、タイプする単語や文字さえもユーザーの思い通りになる。

### 4. 開発ツール

本研究の開発に使用した開発ツールについて説明する。

#### Visual Basic.NET 概要

##### .NET とは

Microsoft 社が 2000 年 7 月に発表したネットワークベースのアプリケーション動作環境を提供するシステム基盤。同社の「Windows DNA」戦略をさらに進化させたもの。インターネットを含むネットワーク上に散在したアプリケーションが自らの機能を「サービス」として公開し、各種の端末から利用するための基盤となるソフトウェアや記述言語・プロトコルなどの規約の衆望を構築することを目的としている。.NET に対応した端末は Java 仮想マシンのようなソフトウェアの動作環境が搭載され、OS の種類に関係なくサービスを受けられるようになる。動作環境となる基盤のソフトウェアは携帯電話や PDA のような小型のコンピュータ向けのものからパソコンやサーバ

などの高性能のコンピュータ向けのものまで OS や機種に関わらず幅広く用意され、様々な言語で開発されたサービスを呼び出して利用することが出来るようになっている。また、各サービスはネットワーク上の他のサービスと連携することができ、単機能のサービスをいくつもつなげて大規模なサービスが構築できるようになる。

Windows 向けの基盤ソフトウェアとしては同社から「Microsoft .NET Framework」が提供されており、この上で各種の.NET 対応サービス・ソフトウェアが利用可能となっている。また、同社のポータルサイト「MSN」では.NET の先駆けとなる再利用可能なサービスの提供を始めており、サービス利用者の個人情報の管理や認証を行なう「Passport」サービスなどが利用可能となっている。アプリケーションの記述言語は特定しておらず、既存の C++ や Visual Basic、VBScript、Jscript などに加え、同社が新たに開発した C#」言語などが利用可能となっている。.NET に対応した言語には名称の末尾に「.NET」の語が付されており、Visual Basic .NET のように従来のバージョンとは記法などが大幅に変更された言語もある。また、サードパーティによって従来のプログラミング言語を.NET に対応させた開発環境が提供されている。

#### .NET Framework とは

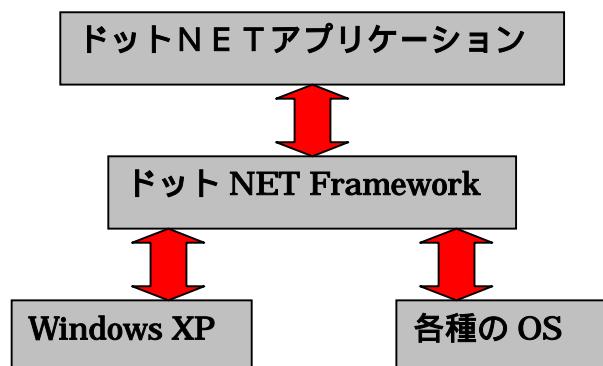
Microsoft .NET 対応アプリケーションの動作環境であり、クラスライブラリの集合体という形で提供される。.NET 対応アプリケーションソフトは、コンパイル時に.NET Framework 用の中間コードに変換され、.NET Framework のランタイムが中間コードを再コンパイルし、OS ごとのネイティブコードに変換して動作する。したがって、.NET 対応アプリケーションは.NET Framework がインストールされている環境であれば、OS などに依存せず動作することができる。これは Java 仮想マシンに近い性格を持つが、.NET Framework 自体は特定の開発言語に依存しない。Microsoft 社が提供する主要な開発言語である Visual Basic、C++、C#などのほか、サードパーティがこれ以外の開発言語を.NET Framework に対応させることもできる。.NET Framework はランタイムとして動作するため、.NET 対応アプリケーションを動作させる環境では必ずインストールが必要になる。Windows 98/NT 4.0 以降の Windows に対して.NET Framework が用意されているが、この他の OS 上で動作する.NET Framework を作成することも可能とされている。

## Visual Basic.NET とは

Visual Basic.NET は.NET Framework 上で動作する.NET アプリケーションソフトの 1 つである。従来の Basic は、MS-DOS 上で動作する通常のプログラム開発を行うためのプログラム言語であるが、Visual Basic はこれとは異なり Windows 用のソフトを開発するためのプログラム言語である。

## Visual Basic.NET の特徴

- ・ 画面操作の基本的な部分を VB.NET が自動的に作成するため、従来、プロの開発者でしか作成できなかった Windows ソフトをコンピュータプログラミングの初心者でも容易に作成できる。
- ・ マイクロソフト社の Quick Basic をベースにしているので、Basic を使用したことがあれば容易にプログラムを作成できる。
- ・ アニメーションの作成、データベースの作成など応用範囲が広い
- ・ Excel や Access などと組み合わせて使用することにより、Excel のマクロ作成など、これらのソフトの機能を拡大できる。

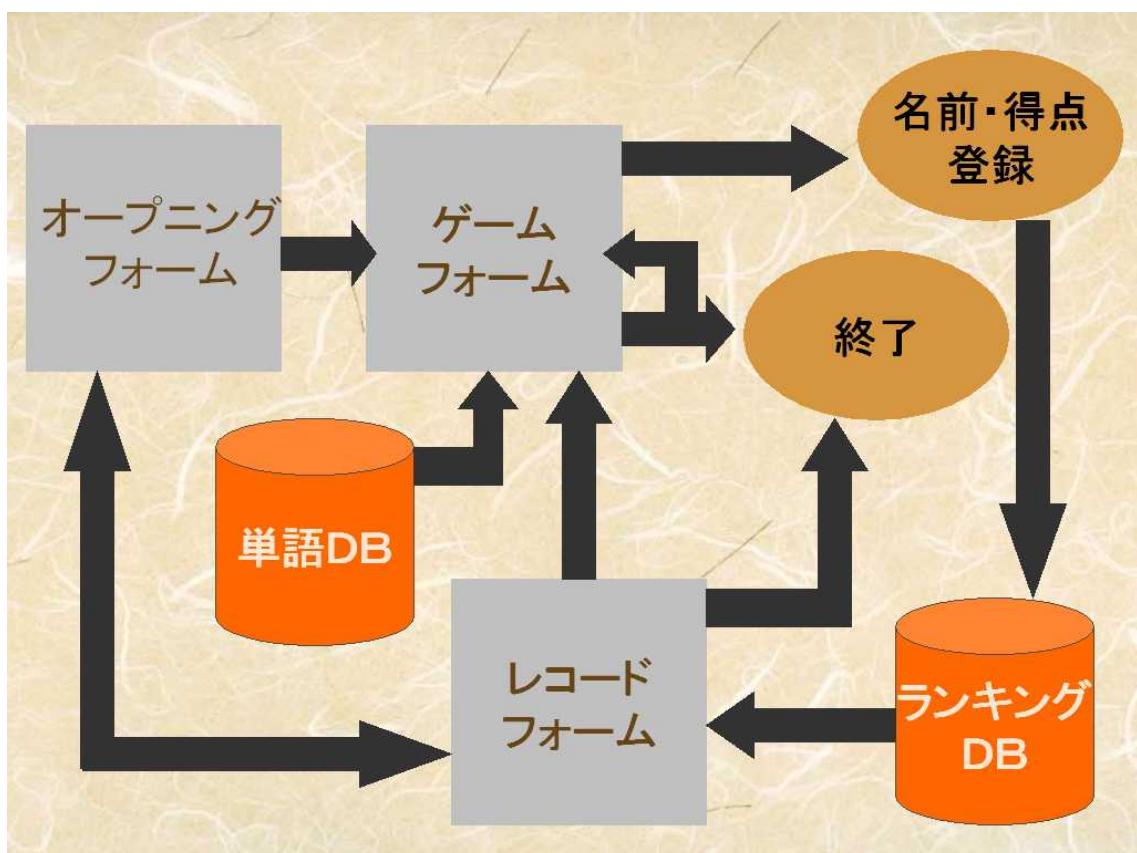


## 5. タイピングゲームの仕様

本研究、タイピングゲームは3つのフォームと3つのデータベースから構成されている。フォームはそれぞれゲームのオープニング画面を表示するオープニングフォーム、ゲームの本体であるゲームフォーム、上位5名の得点と名前を表示するレコードフォームである。

データベースはタイプする文字や単語を格納しておく単語データベース、上位5名の得点を格納する得点データベースと名前データベースである。

オープニングフォームからはランキングフォーム、ゲームフォームへと移ることができ、ランキングフォームは得点上位5名の名前と得点をデータベースから読み込み、それぞれ表示する。オープニングフォームからゲームフォームに移行すると、ユーザーが入力した単語データベースから単語を読み込み、それを表示させる。



まず実行ファイルを実行させるとオープニングフォームが現れ、「SUPER TSUNO TYPE 2005」というタイトルが一文字ずつ表示される。



「RECORD」ボタンを押すとレコードフォームが表示され、これまでの得点上位 5 名の名前と得点のランキングが確認できる。



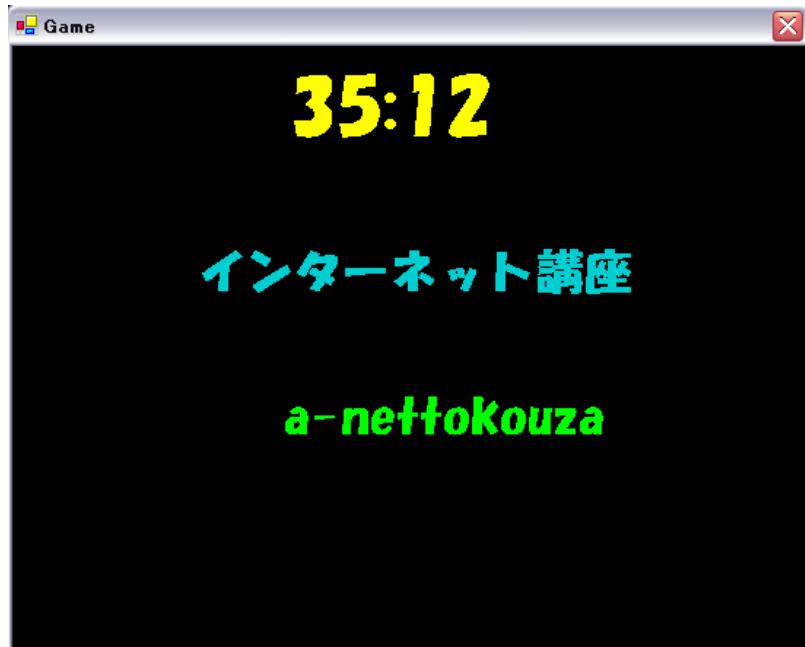
このレコードフォームがロードされると、名前データベースと得点データベースからそれぞれの順位が読み込まれ表示される。

「OK」ボタンを押すとオープニングフォームに戻る。

オープニングフォームの「START」ボタンを押すとゲームフォームが表示され、単語データベースから、ユーザーが入力した単語を読み込み、表示させる。



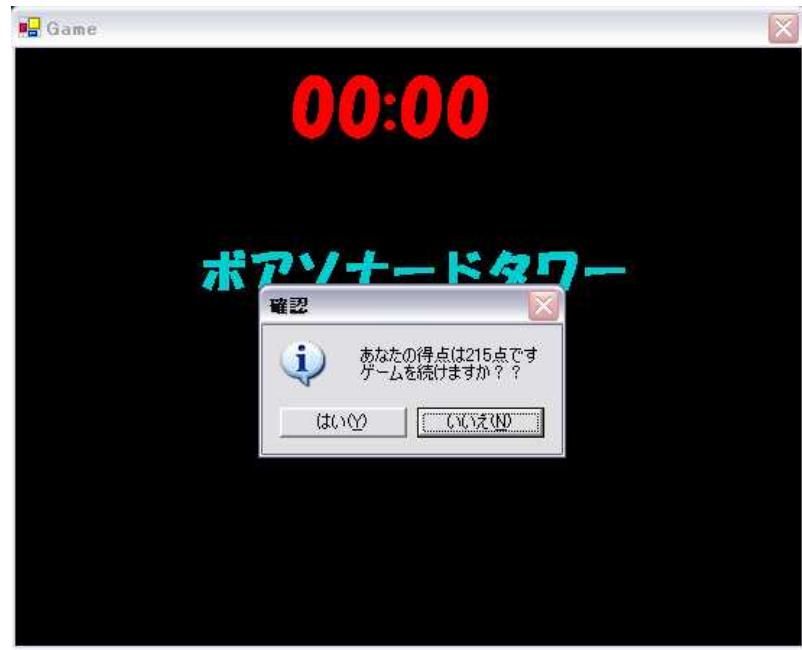
どのキーにもかかわらず、最初にキーがタッチされたら 40 秒のタイマーが発生すし、表示されているローマ字どおり入力していく。  
一つのローマ字をタイプするごとに得点 1 が加算されていく。



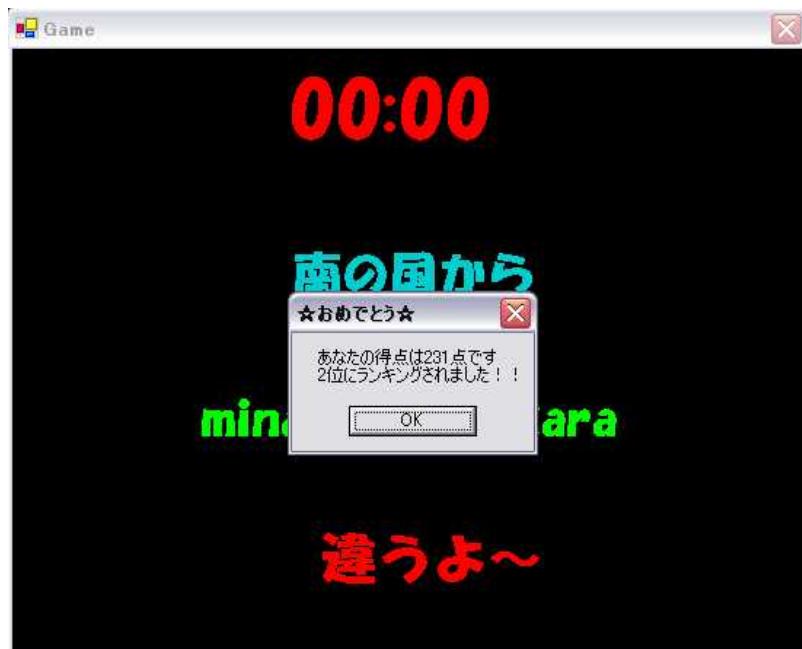
表示された単語を打ち終わるとウィンドウが上下左右に揺れて、次の単語が表示される。  
ミスタイプすると「違うよ～」という文字が表示される。  
残り時間 5 秒を切るとタイマー表示が赤く変わる。

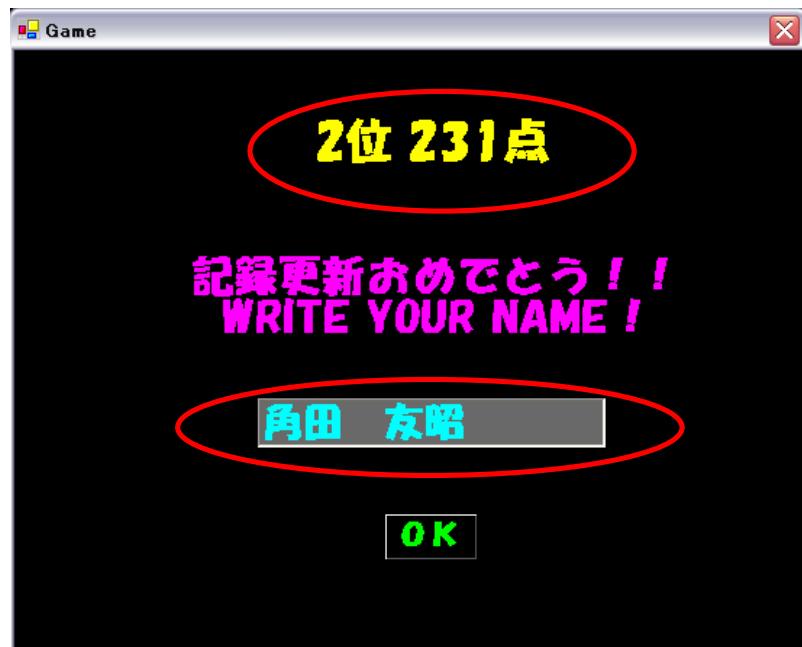


タイマーが0秒になつたらゲーム終了となる。得点が過去の上位5名よりも低かった場合、メッセージボックスに得点が表示され、ゲーム継続を問われる。「はい」ボタンを押せばゲーム継続。「いいえ」ボタンを押せばゲーム終了。



また、得点が過去の上位5名よりも高かった場合、メッセージボックスに得点と順位が表示される。「OK」ボタンを押すと名前書き込み欄が表示される。





名前を書き込み、「OK」ボタンを押すと得点データベースと名前データベース得点と名前がそれぞれ似登録され、レコードフォームが表示される。



ゲームを続ける場合は「CONTINUE」ボタン、終了する場合は「END」ボタンをおす。

## 6 . ソースコード解説

全てのソースコードを解説すると膨大な量になってしまうため、その中でも核となる部分、私なりに工夫した部分を解説する。また、ソースコードにおけるコメントや解説は緑色で記述する

### オープニングフォーム

```
Module module1
    Public Open As Opening
End Module
```

ゲームフォームやレコードフォームからこのオープニングのグローバル変数を参照する。標準モジュールにグローバル変数 Open を宣言しておく

```
Public Class Opening
    Inherits System.Windows.Forms.Form

    Dim word As String = "SUPERTSUNOTYPE 2 0 0 5"    オープニングの文字を格納
    Dim TimerCounter1 As Integer = 0
    Dim TimerCounter2 As Integer = 0
    Dim WordCounter As Integer = 1
```

Public FormCounter As Integer フォーム間を参照するためのグローバル変数

### Waveファイルを鳴らす

```
Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA"
    (ByVal lpszSoundName As String, ByVal uFlags As Integer) As Integer
    Dim SND_ASYNC As Short = &H1S
    Dim SND_NODEFAULT As Short = &H2S
```

### Private Sub Timer1\_Tick

オープニングフォームが表示されて 3 秒間何も起こらない処理  
タイマー 1 のEnabledはFormロード時からTrueにしてある  
インターバルは1000に設定

```
    TimerCounter1 = TimerCounter1 + 1
```

フォームをロードして3秒経ったらTimer2が作動する

```
If TimerCounter1 = 3 Then
    Timer2.Enabled = True
    Timer1.Enabled = False
End If
End Sub
```

Private Sub Timer2\_Tick

フォームに「SUPER TSUNO TYPE 2005」を一字ずつ表示していく処理  
インターバルは 180 に設定

```
TimerCounter2 = TimerCounter2 + 1
```

ワードカウンターが 5 以下の処理

```
If WordCounter <= 5 Then
    If Label1.Text = "" Then
        Sound()
        Label1.Text = Mid(word, WordCounter, 1)
        WordCounter = WordCounter + 1
    Else
        Sound()
        Label1.Text = Label1.Text & Mid(word, WordCounter, 1)
        WordCounter = WordCounter + 1
    End If
```

カウンターが 10 以下の処理

```
ElseIf WordCounter <= 10 Then
    If Label2.Text = "" Then
        Sound()
        Label2.Text = Mid(word, WordCounter, 1)
        WordCounter = WordCounter + 1
    Else
        Sound()
        Label2.Text = Label2.Text & Mid(word, WordCounter, 1)
        WordCounter = WordCounter + 1
    End If
```

カウンターが14以下の処理

```
ElseIf WordCounter <= 14 Then
    If Label3.Text = "" Then
        Sound()
        Label3.Text = Mid(word, WordCounter, 1)
        WordCounter = WordCounter + 1
    Else
        Sound()
        Label3.Text = Label3.Text & Mid(word, WordCounter, 1)
        WordCounter = WordCounter + 1
    End If
```

カウンターが18以下の処理

```
ElseIf WordCounter <= 18 Then
    If Label4.Text = "" Then
        Sound()
        Label4.Text = Mid(word, WordCounter, 1)
        WordCounter = WordCounter + 1
    Else
        Sound()
        Label4.Text = Label4.Text & Mid(word, WordCounter, 1)
        WordCounter = WordCounter + 1
    End If
End If
```

レコードボタン表示、スタートボタン表示

```
If TimerCounter2 = 23 Then
    Sound()
    Button2.Visible = True
ElseIf TimerCounter2 = 25 Then
    Sound()
    Button1.Visible = True
    Timer2.Enabled = False
```

```
End If
```

```
End Sub
```

サウンドプロシージャ

Bin ファイル内にある Fire.WAV を読み込んで鳴らす

```
Sub Sound()
    Dim ret As Integer
    ret = sndPlaySound(CurDir() & "\Fire.WAV", SND_ASYNC Or SND_NODEFAULT)
End Sub
```

ゲームフォーム呼び出し

```
Private Sub Button1_Click
    Dim newf As New Game
    newf.Show()
    Visible = False
End Sub
```

レコードフォーム呼び出し

```
Private Sub Button2_Click
    フォーム間のグローバル変数を 1 にする
    FormCounter = 1

    Dim newf As New Record
    newf.Show()
    Visible = False
End Sub
```

```
Private Sub Opening_Load
```

オープニングフォームのインスタンス (Me) をOpenに格納することで、グローバル変数

Openを通して、ゲームフォームやレコードフォームからこのオープニングのグローバル  
変数を参照できるようにする

```
Open = Me  
End Sub  
End Class
```

## ゲームフォーム

```
Dim x As Integer = 0    タッチ数  
Dim y As Integer = 0    label1のtextの数  
Dim z As Integer = 1    label1のtextを1文字ずつ取ってくる  
Dim ransuu As Integer  
Dim tensuu As Integer = 0  
Dim time As Integer = 4000  
Dim keycount As Integer = 0  
  
Dim counter1 As Integer = 0    ウィンドウの縦の動き  
Dim counter2 As Integer = 0    ウィンドウの横の動き
```

```
Dim TypeWord(500000) As String    タイプする単語
```

```
Dim n As Integer    データベースの変数  
Dim yoso As Integer    配列の要素を決定する  
Dim LoopCount As Integer = 0    ループ文をカウントする変数
```

```
Dim DataLoopCount As Integer = 1  
Dim score(5) As Integer  
Dim NewName As String  
Dim RecordName(5) As String    データベースの上位5の名前  
Dim NameJun As Integer
```

データベースからタイプする単語を読み込んで配列に格納する処理

```
Private Sub Form1_Load
```

```

n = FreeFile()
FileOpen(n, "Word.txt", OpenMode.Input)

Do While Not EOF(n) ファイルエンドまでファイルをリードする
    yoso = yoso + 1 配列の要素(1)～(最後)まで格納できる
    Input(n, TypeWord(yoso))
Loop
FileClose(n)

tangohyouji()
End Sub

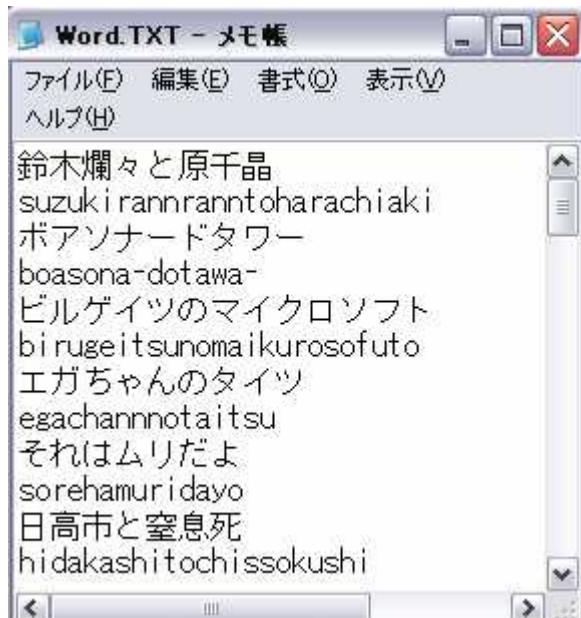
```

Word.TXT の日本語、ローマ字がそれぞれ配列 TypeWord に格納されることになる

TypeWord(1) = “鈴木爛々と原千晶”  
 TypeWord(2) = “suzukirannrannto～”  
 TypeWord(3) = “ボアソナードタワー”  
 TypeWord(4) = “boasona-dotawa-”

⋮ ⋮ ⋮ ⋮

TypeWord(yoso) = “～～～～～～～”



単語表示プロシージャ

Sub tangohyouji()

Randomize()  
 ransuu = Math.Floor(Rnd() \* yoso) + 1 配列の要素数を使用

label1には必ずローマ字が表示され、label5には必ず日本語が表示される処理  
 ローマ字が格納されているのは要素数が偶数の配列  
 日本語が格納されているのは要素数が奇数の配列

整数変数ransuuを直接配列の要素に使用

```
If ransuu Mod 2 = 0 Then ransuuが偶数の時
    label1.Text = TypeWord(ransuu) ローマ字表示
    Label5.Text = TypeWord(ransuu - 1) 日本語表示
Else ransuuが奇数の時
    label1.Text = TypeWord(ransuu + 1) ローマ字表示
    Label5.Text = TypeWord(ransuu) 日本語表示
End If
```

label1に表示されたローマ字の数を数えて変数yに格納

```
y = Len(label1.Text)
End Sub
```

ローマ字をタップして正解を判定する処理

```
Private Sub Form1_KeyPress
    タイマーを発生させる
    最初のキータッチが発生したらタイマーも発生
    keycount = keycount + 1
    If keycount = 1 Then
        Timer1.Enabled = True
    End If
```

label1に表示されたローマ字のZ番目の1文字とキータッチが同じなら、正解の処理

```
If Microsoft.VisualBasic.Mid(label1.Text, z, 1) = e.KeyChar Then
    FireSoundを鳴らす
    Dim ret As Integer
    ret = sndPlaySound(CurDir() & "\Fire.WAV", SND_ASYNC Or
    SND_NODEFAULT)
```

label1に表示されたZ番目の文字を1つずつ消していく

```
Mid(label1.Text, z, 1) = " "
```

タッチ数の計算と得点

```
x = x + 1
```

```
tensuu = tensuu + 1
```

「違うよ～」表示を隠す

```
Label3.Text = ""
```

次の文字を取ってくる

```
z = z + 1
```

Else 不正解時の処理

```
Label3.Text = "違うよ～"
```

```
End If
```

表示された単語の数と得点が同じになったら次の単語を表示する

```
nextword()
```

```
End Sub
```

Nextwordプロシージャ

```
Sub nextword()
```

```
If y = x Then
```

表示された単語を打ち終わったらウィンドウが揺れる

ウィンドウを揺らす処理としてタイマー 2 , 3 を発生させる

```
Timer2.Enabled = True
```

```
Timer3.Enabled = True
```

それぞれの変数を元の値に戻す

```
x = 0
```

```
z = 1
```

label1に次の単語を表示する

```
tangohyouji()
```

```
End If
```

```
End Sub
```

```
Private Sub Timer2_Tick
```

インターバル60に設定

ウィンドウのたての動き

```
counter1 = counter1 + 1
```

counter1が奇数の時

```
If counter1 Mod 2 = 1 Then  
    Top = Top + 12
```

Else counter1が偶数の時

```
    Top = Top - 12  
End If
```

counter変数を2で割った余りを出すことによって、ウィンドウを左右に揺らす  
これによって1つのタイマーでウィンドウを左右にすらすことができる

counterが4になったら0に戻してタイマー2を止める

```
If counter1 = 4 Then  
    counter1 = 0  
    Timer2.Enabled = False  
End If
```

```
End Sub
```

このTimer2の処理と同じようにTimer3を使ってウィンドウを縦に揺らす

残り時間が00:00になった時の、得点比較、データベース更新の処理

```
Private Sub Timer1_Tick
```

インターバルを10に設定し

```
    time = time - 1
```

残り5秒になったらテキストの色を赤くする

```
If time = 500 Then  
    Label8.ForeColor = Color.Red
```

```
End If
```

残り0秒になって得点をデータベースと比較、更新

```
If time = 0 Then
```

```
    Timer1.Enabled = False
```

データベースから得点を読み込む

```
n = FreeFile()
```

```
FileOpen(n, "record.txt", OpenMode.Input)
```

```
Do
```

```
    Input(n, score(DataLoopCount))
```

```
    DataLoopCount = DataLoopCount + 1
```

```
Loop Until DataLoopCount = 6
```

```
FileClose(n)
```

```
DataLoopCount = 1
```

データベースから名前を読み込む

```
n = FreeFile()
```

```
FileOpen(n, "RecordName.txt", OpenMode.Input)
```

```
Do
```

```
    Input(n, RecordName(DataLoopCount))
```

```
    DataLoopCount = DataLoopCount + 1
```

```
Loop Until DataLoopCount = 6
```

```
FileClose(n)
```

```
DataLoopCount = 1
```

点数とデータベースを比較

```
*****
```

1位の時の処理

```
*****
```

```
If tensuu > score(1) Then
    score(5) = score(4)
    score(4) = score(3)
    score(3) = score(2)
    score(2) = score(1)
    score(1) = tensuu
```

NameJun = 1

得点のデータベースの上書き更新

```
DataRewrite()
```

MsgBox表示

```
Dim ret As MsgBoxResult = MsgBox("あなたの得点は" & tensuu & _
"点です" & vbCrLf & "1位にランキングされました！！" & _
, MsgBoxStyle.Exclamation & MsgBoxStyle.OKOnly, " おめでとう"
")
```

名前書き込みフォーム呼び出し

```
If ret = MsgBoxResult.OK Then
    NameWrite()
End If
```

\*\*\*\*\*

2位の時の処理

\*\*\*\*\*

```
ElseIf tensuu > score(2) Then
    score(5) = score(4)
    score(4) = score(3)
    score(3) = score(2)
    score(2) = tensuu
```

NameJun = 2

得点のデータベースの上書き更新

DataRewrite()

MsgBox表示

```
Dim ret As MsgBoxResult = MsgBox("あなたの得点は" & tensuu & _
"点です" & vbCrLf & "2位にランキングされました！！" & _
, MsgBoxStyle.Exclamation & MsgBoxStyle.OKOnly, " おめでとう
")
```

名前書き込みフォーム呼び出し

```
If ret = MsgBoxResult.OK Then
    NameWrite()
End If
```

\*\*\*\*\*

3位の時の処理

\*\*\*\*\*

```
ElseIf tensuu > score(3) Then
    score(5) = score(4)
    score(4) = score(3)
    score(3) = tensuu
```

NameJun = 3

得点のデータベースの上書き更新

DataRewrite()

MsgBox表示

```
Dim ret As MsgBoxResult = MsgBox("あなたの得点は" & tensuu & _
"点です" & vbCrLf & "3位にランキングされました！！" & _
, MsgBoxStyle.Exclamation & MsgBoxStyle.OKOnly, " おめでとう
")
```

名前書き込みフォーム呼び出し

```
If ret = MsgBoxResult.OK Then
    NameWrite()
End If
```

\*\*\*\*\*

4位の時の処理

\*\*\*\*\*

```
ElseIf tensuu > score(4) Then
    score(5) = score(4)
    score(4) = tensuu
```

NameJun = 4

得点のデータベースの上書き更新

DataRewrite()

MessageBox表示

```
Dim ret As MsgBoxResult = MsgBox("あなたの得点は" & tensuu & _
    "点です" & vbCrLf & "4位にランキングされました！！" & _
    , MsgBoxStyle.Exclamation & MsgBoxStyle.OKOnly, " おめでとう"
")
```

名前書き込みフォーム呼び出し

```
If ret = MsgBoxResult.OK Then
    NameWrite()
End If
```

\*\*\*\*\*

5位の時の処理

\*\*\*\*\*

```
ElseIf tensuu > score(5) Then
    score(5) = tensuu
```

```
NameJun = 5
```

得点のデータベースの上書き更新

```
DataRewrite()
```

MsgBox表示

```
Dim ret As MsgBoxResult = MsgBox("あなたの得点は" & tensuu & _
"点です" & vbCrLf & "5位にランキングされました！！" _
, MsgBoxStyle.Exclamation & MsgBoxStyle.OKOnly, " おめでとう
")
```

名前書き込みフォーム呼び出し

```
If ret = MsgBoxResult.OK Then
    NameWrite()
End If
```

\*\*\*\*\*

ランキング外の時

\*\*\*\*\*

```
ElseIf tensuu < score(5) Then
    Dim ret As MsgBoxResult = MsgBox("あなたの得点は" & tensuu & _
"点です" & vbCrLf & "ゲームを続けますか？？" _
, MsgBoxStyle.Question & MsgBoxStyle.YesNo, "確認")
```

ゲーム続行

```
If ret = MsgBoxResult.Yes Then
    tangohyouji()
    それぞれの変数を元の値に戻す
    x = 0
    y = 0
    z = 1
    tensuu = 0
    time = 4000
```

```
keycount = 0

Label8.ForeColor = Color.Yellow

表示された単語の数と得点が同じになって次の単語を表示する
nextword()

Else
    ゲーム終了
    End
    End If
    End If
End If

End Sub
```

名前書き込みフォーム呼び出しプロシージャ

```
Sub NameWrite()
```

```
Label8.Visible = False
Label5.Visible = False
label1.Visible = False
label1.Text = ""
Label3.Visible = False

Label6.Visible = True
TextBox1.Visible = True
Button1.Visible = True
Label9.Visible = True
If NameJun = 1 Then
    Label9.Text = "1位 " & tensuu & "点"
ElseIf NameJun = 2 Then
    Label9.Text = "2位 " & tensuu & "点"
ElseIf NameJun = 3 Then
```

```

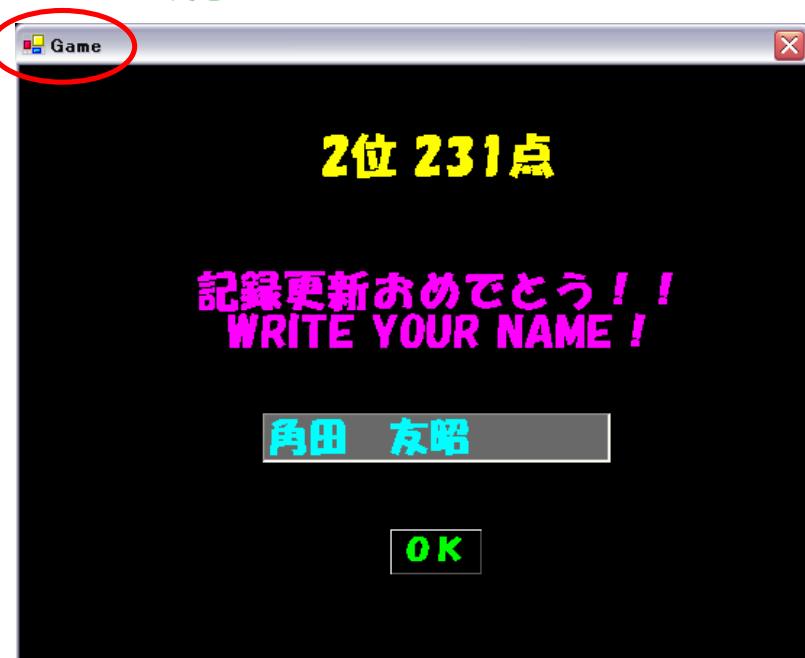
Label9.Text = "3位 " & tensuu & "点"
ElseIf NameJun = 4 Then
    Label9.Text = "4位 " & tensuu & "点"
ElseIf NameJun = 5 Then
    Label9.Text = "5位 " & tensuu & "点"
End If

```

End Sub

名前書き込みフォームはただ、ゲームフォームにTextBox、Label、ButtonのVisibleをTrueにするだけである

特別に新しいフォームを用意していない



Private Sub Button1\_Click

フォーム間のグローバル変数を 2 にする

Open.FormCounter = 2

NewName = TextBox1.Text

TextBox1.Text = ""

If NameJun = 1 Then

```

    RecordName(5) = RecordName(4)
    RecordName(4) = RecordName(3)
    RecordName(3) = RecordName(2)
    RecordName(2) = RecordName(1)
    RecordName(1) = NewName

    ElseIf NameJun = 2 Then
        RecordName(5) = RecordName(4)
        RecordName(4) = RecordName(3)
        RecordName(3) = RecordName(2)
        RecordName(2) = NewName

    ElseIf NameJun = 3 Then
        RecordName(5) = RecordName(4)
        RecordName(4) = RecordName(3)
        RecordName(3) = NewName

    ElseIf NameJun = 4 Then
        RecordName(5) = RecordName(4)
        RecordName(4) = NewName

    ElseIf NameJun = 5 Then
        RecordName(5) = NewName

    End If

```

名前のデータベース上書き更新

```

n = FreeFile()
FileOpen(n, "RecordName.txt", OpenMode.Output)

```

Do

```

    WriteLine(n, RecordName(DataLoopCount))
    DataLoopCount = DataLoopCount + 1

```

Loop Until DataLoopCount = 6

```
FileClose(n)
```

```
DataLoopCount = 1
```

レコードフォーム呼び出し

```
Dim newf As New Record
```

```

newf.Show()
Visible = False
End Sub

```

## レコードフォーム

```
Private Sub Record_Load
```

オープニングフォームのグローバル変数FormCounterを参照してレコードフォーム上のボタンの配置を操作する

オープニングフォームからレコードフォームを呼び出した時

```
If Open.FormCounter = 1 Then
```

```
    Button3.Visible = True
```

```
Else ゲームを終了してランキング後、レコードフォームを呼び出した時
```

```
    Button1.Visible = True
```

```
    Button2.Visible = True
```

```
End If
```



オープニングフォームから呼び出した時

ゲームを終了してランキング後に呼び出した時

データベースから得点を読み込む

```
n = FreeFile()
```

```
FileOpen(n, "record.txt", OpenMode.Input)
```

```

Do
    Input(n, score(loopcount))
    loopcount = loopcount + 1
Loop Until loopcount = 6

FileClose(n)
loopcount = 1

データベースから名前を読み込む
n = FreeFile()
FileOpen(n, "RecordName.txt", OpenMode.Input)

Do
    Input(n, RecoName(loopcount))
    loopcount = loopcount + 1
Loop Until loopcount = 6

FileClose(n)
loopcount = 1

それぞれの名前と得点を表示
Label1.Text = "1位 " & RecoName(1)
Label2.Text = "2位 " & RecoName(2)
Label3.Text = "3位 " & RecoName(3)
Label4.Text = "4位 " & RecoName(4)
Label5.Text = "5位 " & RecoName(5)
Label6.Text = score(1) & "点"
Label7.Text = score(2) & "点"
Label8.Text = score(3) & "点"
Label9.Text = score(4) & "点"
Label10.Text = score(5) & "点"

End Sub

```

## 7 . 課題点

### (インターフェース)

当初の計画であった3D表示が果たせていない。確かにタイピングゲームとしてはある程度、完成した形になっているが、ビジュアル面での効果はかなり乏しい。多くのユーザーを引き付けるためには、やはり圧倒的な視覚効果が必要であると実感している。

### (設定機能の充実)

本研究のタイピングゲームは私が決めた一つの設定の下でしかゲームをすることができない。例えば、ゲーム時間、難易度、ミスタッチ数、分速タッチ数など各種機能をユーザーの好みによって設定できるよう、今後も開発を続けていこうと思う。これらのことが、結局は幅広いユーザーの獲得へと繋がっていくのだと思う。

### (プログラム)

プログラムの書き方自体に課題点が多くある。例えばfor文などを用いてプログラムをスリムにできるところが多くある。さらに書法を学習してプログラムのスリム化を追及していくなければならない。そうすることによって、プログラム全体のサイズを小さくすることができる。プログラムが膨大な量になった場合など、それ自体の処理速度に関わってくると考えられるので、プログラムをなるべくスリムにするよう心がけなければならない。

また、ゲーム中におけるプログラムの課題点として、得点が上位5位以内にランキングされた時、名前を記入している間にウィンドウの終了ボタンを押してしまうと、得点だけがデータベースに上書きされてしまう問題点が挙げられる。

当初の計画からすると本研究のタイピングゲームは完成形ではないので、以上の課題点をふまえて開発を続けていきたいと思う。

今回の卒業制作を開発するにあたって、重定先生、山本先生にはプログラムや全体の仕様、設計などにおいて、さまざまご指導いただき大変ありがとうございました。また同ゼミ生にも大変お世話になり感謝しております。2年間ありがとうございました。

## 8 . 参考文献

河西朝雄『VB.NET 基礎学習 Bible』東京、技術評論者、2003年。

山本昌弘 / 重定如彦『例題でわかる Visual Basic.NET』東京、東京電機大学出版局、2004年。

藤田伸二『Direct X + VB.NET ではじめるゲームプログラミング』東京、翔泳社、2002年。

若山芳三『学生のための Visual Basic.NET』東京、東京電機大学出版局、2004年。