

Dictation

汎用型

dic4

```
; dic4
; copyleft 2009-07-16 sugiura@nagoya-u.jp
; 2020-06-19 pack する際の、WAV ファイルの容量制限明記

; クローズにしたいテキストを、テキストファイルで、data.txt という名前で
; このプログラムと同じフォルダーに入れておく。
; 注意) 半角スペース一つを単語の区切りとみなすので、文末もスペースは一つに。
; 作成した結果は、out.txt という名前で保存される。
; 何語目を空欄にしたいかは、以下で、設定する。
;
; dic3 独立したプログラムとして動くように。
; 1 ) ダイアログでファイルを読み込むのをやめる => 6 )
; 2 ) 解答欄をつける
; 3 ) 正誤判断
; 4 ) 音声ファイルを再生
; 5 ) 音声ファイルも込みでコンパイルするバージョンも作成 => mp3 は無理。WAV ( 2MB 以下 ) に。
; 6 ) テキストファイルも込みでコンパイルするバージョンも作成
;
; dic4 での追加
; 1 ) 学習記録をとる
;
; 今後の課題
; 1 ) 問題文と音声ファイルの確認ダイアログ
;
; dic4 で追加
; dialog " 同じフォルダー内に問題文 data.txt と音声 oto.mp3 が必要です ", 3

nangome = 7 ; 何語目を空欄にするか

; 音声ファイルを読み込んでおく
mmload "oto.mp3",0

; ファイル読み込み
notesel text ; ノート用変数 text
noteload "data.txt" ; ファイルを読み込む

fontsize = 12 ; フォントの大きさ
font " M S 明朝 ", fontsize, 0

; -----
; 全体的な環境設定
wx=1000 ; 開くウィンドウの横
wy=860 ; 開くウィンドウの縦
screen 0,wx,wy,1 ; メインウィンドウの作成

pos 250, 200
mes " リスニングの練習です。単語を聞き取ってください。 "
```

```

mes ""
mes ""
mes ""
mes ""

pos 500, ,
button goto "はじめます", *letsgetstarted

mes ""
mes ""
mes ""
pos 200,500
mes "注意"
mes " 1 ) 問題文 data.txt と 音声ファイル oto.mp3 が同じフォルダ内に必要です。"
mes " 2 ) 7 単語目ごとに空欄になるように設定されています。"
mes " 3 ) 結果は out.txt というファイルに保存されます。"

stop

*letsgetstarted

/* ----- ここをやめる -----
; ファイル読み込み dialog コマンド 拡張子を指定し、引数 16 でファイル読み込み
dialog "txt", 16
; 読み込んだファイル名は refstr 変数に入る

notesel text ; ノート用変数 text
noteload refstr ; ファイルを読み込む
*/

;mes text

t = 0
w = ""

repeat
  getstr w, text, t, ' ' ; txt 文字列の t 文字目から ' ' (スペース) までの
                        ; 文字列を切り出して、変数 w に入れる。
  if strsize = 0 : break ; 文字サイズがゼロ、すなわち文字がなくなったら
                        ; break で抜ける。
  tango(cnt) = w ; 変数 w に入っている単語を、配列 tango に入れていく
  t = t + strsize ; 次の単語を先頭の何文字目から切り出すか計算
await
loop

haba = ginfo(12) ; ウィンドウの内のり幅

;mes haba

wait 10

tangosuu = length(tango) ; 配列の要素の数 = 問題数
premondaibun = "" ; 変数 mondaibun を文字型の変数に

hit = 0
seikai = ""

repeat tangosuu
  if (cnt + 1) % nangome == 0 { ; 商をとって、あまりが 0 だったら、、、
    ;----- 正解を配列に -----
    seikai(hit) = tango(cnt)
    hit = hit + 1 ; いくつめのヒットか記録
                  ; それを問題番号とする
  ; newtango(cnt) = "_____" ; そこんところは空欄に
  ;----- 単語の長さに応じて下線の数を -----

```

```

kasen = strlen(tango(cnt)) ; 単語の文字数
sen = ""
;-----
;   あまり短すぎる場合は、せめて3文字分
if kasen < 3 : kasen = 3
;-----
repeat kasen
    sen = sen + " "
await
loop
newtango(cnt) = "" + hit + ")" + sen

}
else{
    newtango(cnt) = tango(cnt)    ; それ以外はもとのまま
}

premondaibun = premondaibun + " " + newtango(cnt)

await
loop

;mes premondaibun

;   ここまでで、一旦、下線を入れた文章作成。
;-----
;   これ以降で、画面の幅に折り返し表示

*here
cls

pos 40,,
mes " 空欄を聞き取って、解答欄に記入してください。 "
mes ""
mes ""

t = 0
w = ""

repeat
    getstr w, premondaibun, t, ' ' ; 文字列の t文字目から ' ' (スペース) までの
    ; 文字列を切り出して、変数 wに入れる。
    if strsize = 0 : break ; 文字サイズがゼロ、すなわち文字がなくなったら
    ; breakで抜ける。
    mondaitango(cnt) = w ; 変数 w に入っている単語を、配列 に入れていく
    t = t + strsize ; 次の単語を先頭の何文字目から切り出すか計算
await
loop

;-----

mondaibun=""

kaisuu = length(mondaitango)

out = ""
notesel out

repeat kaisuu
    mojisuu = 0
    nagasa = 0
    pre = ""

    pre = mondaibun + " " + mondaitango(cnt); つなげてもとの文に戻す
    mojisuu = strlen(pre) ; 文の文字数
    nagasa = mojisuu * fontsize ; 文字列のピクセル数

;----- 最後の一文の例外処理 -----
if cnt == kaisuu -1 {
    last = mondaibun + " " + mondaitango(cnt)
    mes last
    noteadd last
}

```

```

;-----
if haba > nagasa {                                ; 文字列の長さより画面の幅が多い場合
    mondaibun = mondaibun + " " + mondaitango(cnt); 問題文に単語を足していく
}
else{                                              ; そうでないときには、
    mondaibunout = mondaibun + " " + mondaitango(cnt) + "%n" ;
    mes mondaibunout
    noteadd mondaibunout
    mondaibunout = ""
    mondaibun = ""
}

await
loop

pos 800,,
mes " 総語数：" + tangosuu
noteadd ""
noteadd " 総語数：" + tangosuu
noteadd ""
noteadd ""

; 音声ファイル操作ボタン
sousa = ginfo_cy

pos 50, sousa
button gosub "Play", *speech

pos 110, sousa
button gosub "Stop", *pause

pos 380, sousa
button gosub "Check", *check

pos 550, sousa
button gosub "Answer", *okotae

pos 610, sousa
button gosub "Clear", *clear

pos 750, sousa
button gosub "Finish", *owari

;----- 入力欄一覧 -----

kaitouichiX = ginfo_cx
kaitouichiY = ginfo_cy ; ginfo_cy は、pos 命令により設定されたカレントポジションの Y 座標
sdim nyuryoku          ; 文字型配列の宣言
*bottom

pos kaitouichiX, kaitouichiY
repeat hit
    bangou = cnt +1
; nyuryoku(cnt) = "" ; これでも文字型は宣言できるが、中身が空になる

pos 150,,
hora = "" + bangou + ")"
mes hora
objmode 2
; kaitouichiZ = ginfo_cy - 16 ; フォントを指定されているものに
; takasa = ginfo_mesy ; フォントサイズではダメ
kaitouichiZ = ginfo_cy - takasa ; ginfo_mesy は文字の高さの値

pos 180, kaitouichiZ

```

```

input nyuryoku(cnt), 200,24,0
inputID(cnt) = stat

await
loop

repeat length(inputID)
objprm inputID(cnt), nyuryoku(cnt)
loop

; dic4 での追加
noteadd "%t" + "No." + "%t" + " 正解 " + "%t" + " 解答 "
stop

;----- 正誤判断 -----

sdim nyuryokunew

*check
atteru = 0

; dic4 での追加
noteadd "-----"

repeat hit
; dic4 での追加
monme = cnt + 1
noteadd "%t" + monme + "%t" + seikai(cnt) + "%t" + nyuryoku(cnt)

if nyuryoku(cnt) == seikai(cnt){ ; 入力と正解があっていたら
atteru = atteru + 1 ; あってる数を足していく

}else{ ; そうじゃなかったら
nyuryoku(cnt) = "" ; 入力欄を空に
}
}
await
loop

; dic4 での追加
noteadd "-----"

dialog "" + atteru + " 問あってます。",0
goto *bottom

return

;notesave "out.txt"

*owari
cls
pos 500, 400
mes "" + atteru + " 問あってました。 "

; dic4 での追加
noteadd "-----final answer-----"
noteadd "" + atteru + " 問あってました。 "
noteadd "-----"
repeat hit
dainan = cnt +1
noteadd "%t" + dainan + "%t" + seikai(cnt) + "%t" + nyuryoku(cnt)
await
loop

; dic4 での追加 結果の表示
; mes out
; notesel out
notesave "out.txt"

wait 180
end

;-----sub-----

;----- 音を鳴らすのととめるのと -----
*speech
mplay 0
return

```

```

*pause
mmstop
return

;----- 正解一覧 -----
*okotae
hora = ""
repeat hit
    bangou = cnt +1
    hora = hora + " " + bangou + ")" + "%t" + seikai(cnt) + "%n"

; noteadd hora

await
loop
    objmode 2 ; フォントを指定されているもの
    tateichi = kaitouichiY
    pos 500, tateichi
    mesbox hora, 200, takasa * (hit + 1), 0, , ; hora の最後の改行分 1 行多い
    messageboxID = stat

return

*clear
objprm messageboxID, "" ; メッセージボックスの内容を空に
return

```

実例実行ファイル

- 上の dic4 を修正し、
 - 音声ファイル bliss.wav を組み込み、
 - テキストファイル bliss.txt を組み込み、
 - 3 語目を順に空欄にするように設定した。
 - 結果は、out.txt ファイルに保存される。
 - 実行ファイルとしてコンパイルしたので、以下のファイルのみで実行可能
 - 「実行ファイル」をダウンロードして実行しようとする、Windows が警告を出します。

Windows によって PC が保護されました

Microsoft Defender SmartScreen は認識されないアプリの起動を停止しました。このアプリを実行すると、PC が危険にさらされる可能性があります。

[詳細情報](#)

実行しない

- ・「詳細情報」をクリックし、
- ・「実行」してください。

素材の出典：VOA

- ・ <https://learningenglish.voanews.com/a/english-in-a-minute-ignorance-is-bliss/5411849.html>

DictationBliss.exe

・
・

```
; dic4 修正 dic5bliss  
; copyleft 2009-07-16 sugiura@nagoya-u.jp  
; copyleft 2020-06-19
```

```
#pack "bliss.wav"  
#pack "bliss.txt"
```

```
nangome = 3 ; 何語目を空欄にするか
```

```
; 音声ファイルを読み込んでおく  
mmload "bliss.wav",0
```

```

; ファイル読み込み
notesel text          ; ノート用変数 text
noteload "bliss.txt"   ; ファイルを読み込む

fontsize = 12          ; フォントの大きさ
font " M S  明朝 ", fontsize, 0

;-----
; 全体的な環境設定

wx=1000      ; 開くウィンドウの横
wy=860      ; 開くウィンドウの縦

screen 0,wx,wy,1      ; メインウィンドウの作成

pos 250, 200

mes "Dictation You Like"
mes ""
mes ""
mes ""
mes ""
mes ""

pos 600, ,
button goto "Let's begin!", *letsgetstarted

mes ""
mes ""
mes ""
pos 200,500
mes "Note:"
mes "    The sound and text are from VOA."
mes "    https://learningenglish.voanews.com/a/english-in-a-minute-ignorance-is-bliss/5411849.html"
mes ""
mes "    This program is created by sugiura@nagoya-u.jp"
mes "    Visit https://sugiura-ken.org/wiki/wiki.cgi/exp?page=CALLbyHSP"

stop

*letsgetstarted

t = 0
w = ""

repeat
    getstr w, text, t, ' ' ; txt 文字列の t 文字目から ' ' (スペース) までの
                           ; 文字列を切り出して、変数 w に入れる。
    if strsize = 0 : break ; 文字サイズがゼロ、すなわち文字がなくなったら
                           ; break で抜ける。
    tango(cnt) = w         ; 変数 w に入っている単語を、配列 tango に入れていく
    t = t + strsize        ; 次の単語を先頭の何文字目から切り出すか計算
await t
loop

haba = ginfo(12)          ; ウィンドウの内のり幅

;mes haba

wait 10

tangosuu = length(tango) ; 配列の要素の数 = 問題数

premondaibun = ""        ; 変数 mondaibun を文字型の変数に

hit = 0

```



```

seikai = ""
repeat tangosuu
    if (cnt + 1) % nangome == 0 {      ; 商をとって、あまりが0だったら、、、
        ;----- 正解を配列に -----
        seikai(hit) = tango(cnt)
        hit = hit + 1                ;   いくつめのヒットか記録
        ;   それを問題番号とする
    ;   newtango(cnt) = " _____ " ;   そこんところは空欄に
;----- 単語の長さに応じて下線の数を -----
    kasen = strlen(tango(cnt)) ; 単語の文字数
    sen = ""
    ;-----
    ;   あまり短すぎる場合は、せめて3文字分
    if kasen < 3 : kasen = 3
    ;-----
    repeat kasen
        sen = sen + " _ "
    await
    loop
    newtango(cnt) = "" + hit + ")" + sen

}
else{
    newtango(cnt) = tango(cnt)      ;   それ以外はもとのまま
}

premondaibun = premondaibun + " " + newtango(cnt)

await
loop

;mes premondaibun

;   ここまでで、一旦、下線を入れた文章作成。
;-----
;   これ以降で、画面の幅に折り返し表示

*here
cls

pos 40,
mes "(Fill in the blanks as you listen.)"
mes ""
mes ""

t = 0
w = ""

repeat
    getstr w, premondaibun, t, ' '      ;   文字列の t 文字目から ' ' (スペース) までの
    ;   文字列を切り出して、変数 w に入れる。
    if strsize = 0 : break                ;   文字サイズがゼロ、すなわち文字がなくなったら
    ;   break で抜ける。
    mondaitango(cnt) = w                  ;   変数 w に入っている単語を、配列 に入れていく
    t = t + strsize                        ;   次の単語を先頭の何文字目から切り出すか計算
    await
    loop

;-----

mondaibun=""

kaisuu = length(mondaitango)

out = ""
notesel out

repeat kaisuu
    mojisuu = 0
    nagasa = 0
    pre = ""

```

```

pre = mondaibun + " " + mondaitango(cnt); つなげてもとの文に戻す
mojisuu = strlen(pre) ; 文の文字数
nagasa = mojisuu * fontsize ; 文字列のピクセル数
;----- 最後の一文の例外処理 -----
if cnt == kaisuu -1 {
    last = mondaibun + " " + mondaitango(cnt)
    mes last
    noteadd last
}
;-----

if haba > nagasa { ; 文字列の長さより画面の幅が多い場合
    mondaibun = mondaibun + " " + mondaitango(cnt); 問題文に単語を足していく
}
else{ ; そうでないときには、
    mondaibunout = mondaibun + " " + mondaitango(cnt) + "¥n" ;
    mes mondaibunout
    noteadd mondaibunout
    mondaibunout = ""
    mondaibun = ""
}

await
loop

pos 800,,
mes "Words: " + tangosuu
noteadd ""
noteadd "Words: " + tangosuu
noteadd ""
noteadd ""

; 音声ファイル操作ボタン

sousa = ginfo_cy

pos 50, sousa
button gosub "Play", *speech

pos 110, sousa
button gosub "Stop", *pause

pos 380, sousa
button gosub "Check", *check

pos 550, sousa
button gosub "Answer", *okotae

pos 610, sousa
button gosub "Clear", *clear

pos 750, sousa
button gosub "Finish", *owari

;----- 入力欄一覧 -----

kaitouichiX = ginfo_cx
kaitouichiY = ginfo_cy ; ginfo_cy は、pos 命令により設定されたカレントポジションの Y 座標

sdim nyuryoku ; 文字型配列の宣言
*bottom

pos kaitouichiX, kaitouichiY
repeat hit

```

```

    bangou = cnt + 1
;   nyuryoku(cnt) = ""           ; これでも文字型は宣言できるが、中身が空になる

    pos 150,
    hora = "" + bangou + ")"
    mes hora
    objmode 2
;   kaitouichiZ = ginfo_cy - 16   ; フォントを指定されているものに
    takasa = ginfo_mesy          ; フォントサイズではダメ
    kaitouichiZ = ginfo_cy - takasa ; ginfo_mesy は文字の高さの値

    pos 180, kaitouichiZ
    input nyuryoku(cnt), 200,24,0
    inputID(cnt) = stat

await
loop

    repeat length(inputID)
    objprm inputID(cnt), nyuryoku(cnt)
    loop

; dic4 での追加
    noteadd "%t" + "No." + "%t" + " 正解 " + "%t" + " 解答 "
stop

;----- 正誤判断 -----

sdim nyuryokunew

*check
    atteru = 0

; dic4 での追加
    noteadd "-----"

    repeat hit

; dic4 での追加
    monme = cnt + 1
    noteadd "%t" + monme + "%t" + seikai(cnt) + "%t" + nyuryoku(cnt)

    if nyuryoku(cnt) == seikai(cnt){ ; 入力と正解があっていたら
        atteru = atteru + 1          ; あってる数を足していく
    }else{ ; そうじゃなかったら
        nyuryoku(cnt) = ""          ; 入力欄を空に
    }
}
await
loop

; dic4 での追加
    noteadd "-----"

    dialog "" + atteru + " correct answers",0
    goto *bottom

return

;notesave "out.txt"

*owari
cls
pos 500, 400
mes "The number of correct answers: " + atteru

; dic4 での追加
    noteadd "-----final answer-----"
    noteadd "" + atteru + " correct answers"
    noteadd "-----"
    repeat hit
        dainan = cnt + 1
        noteadd "%t" + dainan + "%t" + seikai(cnt) + "%t" + nyuryoku(cnt)
    await
    loop

; dic4 での追加 結果の表示

```

```

; mes out
notesel out
notesave "out.txt"

wait 380
end

;-----sub-----

;----- 音を鳴らすのととめるのと -----
*speech
mmplay 0
return

*pause
mmstop
return

;----- 正解一覧 -----
*okotae
hora = ""
repeat hit
    bangou = cnt + 1
    hora = hora + " " + bangou + ")" + "%t" + seikai(cnt) + "%n"
; noteadd hora

await
loop
    objmode 2 ; フォントを指定されているもの
    tateichi = kaitouichiY
    pos 500, tateichi
    mesbox hora, 200, takasa * (hit + 1), 0, , ; hora の最後の改行分 1 行多い
    messageboxID = stat

return

*clear
objprm messageboxID, "" ; メッセージボックスの内容を空に
return

```