

HSP

ExpbyHSP

自己ペース読み（移動窓）課題

```
;-----BEGIN-----  
;Experiment 移動窓課題 for HSP3  
;Moving Window Reading Paradigm  
; copy left 2007-07-05 sugiura@nagoya-u.jp  
; 2013-06-28 HSP3 で動かないところを修正  
;  
;-----  
; ★データの記述★  
  
;sdim kaitou,2000      ; 文字列変数 data  
;sdim data,2000  
  
; 例のように引用符の中に英文を入れる。途中で改行しないこと。  
; このバージョンでは、画面の右側で折り返して提示する。  
  
kaitou = "No one would have believed in the last years of the nineteenth century that this world was  
being watched keenly and closely by intelligences greater than man's and yet as mortal as his own;  
that as men busied themselves about their various concerns they were scrutinised and studied, perhaps  
almost as narrowly as a man with a microscope might scrutinise the transient creatures that swarm and  
multiply in a drop of water."  
  
data = kaitou + " 右向き矢印で終了 " ; 問題の最後にこの文字列が出るように  
; さもないと、最後の単語で止まったままで  
; 終わった気になる人がある。  
  
; これ以降は修正する必要なし。  
  
; ただし、ログの記録に際して「★実際の実験ではコメントアウトする」の行は  
; 実際の実験ではコメントアウトしてコンパイルすること。  
;  
;-----  
; 全体的な環境設定  
  
wx = 800 ; 開くウィンドウの横  
wy = 600 ; 開くウィンドウの縦  
  
wxs = ginfo_dispx/2 - wx/2 ; ウィンドウの表示位置 X  
wys = ginfo_dispy/2 - wy/2 ; ウィンドウの表示位置 Y  
  
yoko = wxs  
tate = wys  
  
screen 0,wx,wy,1,wxs,wys ; メインウィンドウの作成  
  
; cls  
  
kiroku = str(kiroku) ; HSP3 に合わせて修正 2013-06-28  
kiroku = "" ; 変数名を空に  
notesel kiroku ; ログをメモリーノートパッドにとる  
  
;-----  
;cls  
  
; 読ませる英文を、単語に分けて配列「S」に入れる。  
  
; sdim S,20,500      ; 文字列型の配列  
; 変数名 S 「刺激」として提示される単語  
; 20 は、1単語の文字数の上限。  
; 500 は、単語の数の上限。  
  
tmoji = int(tmoji) ; HSP3 に合わせて修正 2013-06-28  
tmoji = 0  
w = str(w)      ; 文字列変数wの宣言 ; HSP3 に合わせて修正 2013-06-28  
j = 0  
repeat  
  getstr w,data,tmoji,' ' ; data のt文字目から ' '（スペース）までの単語を
```

```

        ; 切り出して文字列変数 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抜ける
    S(j) = w                  ; 単語を配列 S に入れていく
    j=j+1                    ; 配列の数 (単語の数) を増やしていく。
    tmoji=tmoji+strsize      ; 次の単語を先頭の何文字目から切り出すか
loop

    WordNum = j              ; 単語数を WordNum に入れる。

;-----
; 各単語を、その文字数分だけ下線にしたものを配列「U」に入れる。

cls

sdim U,20,WordNum           ; 文字列型の配列
    ; 変数名 U 各単語の文字数分の下線
    ; 20 は、1 単語の文字数の上限。
    ; WrodNum は、単語の数の上限。

tmoji=0
w = str(w)                  ; 文字列変数wの宣言 ; HSP3 に合わせて修正 2013-06-28
j=0
repeat
    getstr w,data,tmoji,' '  ; data の t 文字目から ' ' (スペース) までの単語を
    ; 切り出して文字列変数 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抜ける

    sl = strlen(w)           ; 単語の文字数を sl に
    repeat sl
        U(j) = U(j)+"_"      ; 単語の文字の数だけ下線を配列 U に入れていく
    loop

    j=j+1                    ; 配列の数 (単語の数) を増やしていく。
    tmoji=tmoji+strsize      ; 次の単語を先頭の何文字目から切り出すか
loop

;-----
; 実験用データ処理準備部分

;-----
; 最初に、下線だけで文章の全体のイメージを提示する。

fontsize=18
font " MS ゴシック ",fontsize

sdim Kasen,1000             ; 下線だけの表示をつくる

sdim W,20,WordNum           ; 文字列型の配列

nagasa = 0
tantan = 0
haba = 0
g = 0
repeat WordNum

    tantan = strlen(U(g))    ; 下線の数をもとに
    nagasa = (tantan + 2)*fontsize/2 ; フォントの大きさを計算
    haba = haba + nagasa     ; 提示する行の幅
    W(g) = U(g)              ; 下線の配列を変えずに別の配列で処理
    if wx < haba {           ; 画面の幅を超えたら
        W(g) = "\n\n" + W(g) ; 長い行は前に改行 2 回分をつける
        nagasa = 0           ; 行の幅の計算を 0 からやりなおす
        haba = 0
    }

    Kasen = Kasen+W(g)+" "    ; 改行記号入りの文字列として追加

    g = g + 1
loop

;-----
; 単語と下線による置き換えを、単語ごとに行ったものを別の配列に入れる

fontsize=18
font " MS ゴシック ",fontsize

sdim P,20,WordNum           ; 文字列型の配列

```

```

; 変数名 P 単語か文字数分の下線
; 20 は、1単語の文字数の上限。
; 3200 は、単語の数の上限。

sdim Bun,1000
sdim R,1000,WordNum

;HSP のバージョン2系は、Perl の配列に関する便利な関数、splice, join がない。

e = 0
repeat WordNum      ; 生成する文の数だけ繰り返す

    f = 0
    repeat WordNum    ; 文の中の要素の数だけくり返す

        if f > e : P(f) = U(f) ; 何番目の文の何番目の要素かをみて同じ場合だけ
        if f = e : P(f) = S(f) ; 単語を出して、それ以外は下線を出す
        if f < e : P(f) = U(f) ; それを配列 P に入れていく

    f = f + 1

loop

;-----
; 画面の幅に応じて折り返すように

nagasa = 0
tantan = 0
haba = 0
g = 0
repeat WordNum

;   strlen tantan,P(g)
   tantan = strlen(P(g)) ; HSP3 に合わせて修正 2013-06-28
   nagasa = (tantan + 2)*fontsize/2
   haba = haba + nagasa

   if wx < haba {
       P(g) = "\n\n" + P(g) ; 長い行は前に改行をつける
       nagasa = 0
       haba = 0
   }

   Bun = Bun+P(g)+" "

   g = g + 1

loop

R(e) = Bun      ; 出来上がった Bun を配列 R に入れていく

Bun = ""        ; Bun を空に

;-----

e = e + 1
loop

;-----

mouse -1        ; マウスカーソルを非表示に
                ; 後で、mouse で戻すように

;-----
; 説明の画面の提示

cls

pos 100,150      ; 画面上、左上隅から、
                ; 右50、下150に表示

font "MS ゴシック", 32, 1
mes " Moving Window Reading "
mes " "

font "MS 明朝", 16

mes " 画面に単語の数だけ下線が表示されます。"
mes " 矢印キーを押すと、単語が「一つずつ」表示されます。"
mes " "
mes " 右向きの矢印「→」のキーを押すと、次の単語が表示されます。"

```

```

mes " 左向きの矢印「←」のキーを押すと、前の単語に戻ります。"
mes " "
mes " 先に読み進むときは、右向きの矢印をどんどん押してください。"
mes " "
mes " 読み終わるまでの時間を計るので、出来るだけ速く読んでください。"
mes " "
mes " 最後は、もう一度、右向きの矢印を押して終了します。"
mes " "
mes " 使い方がわかったら、右向きの矢印を押してください。"

repeat      ; 反応を待つ

    stick res    ; 入力されたキー情報の取得

    if res&4 : goto *ready

await 1: loop

;-----
*ready

cls

    fontsize=18
    font " MS ゴシック ",fontsize

    pos 20, 20

mes " 右向きの矢印を押してはじめてください。"
mes " "
mes Kasen      ; 下線だけのものを提示

repeat      ; 反応を待つ

    stick res    ; 入力されたキー情報の取得

    if res&4 : goto *mon01 ; 右向き矢印が押されたら開始

await 1: loop      ; リピートで反応を待つ場合は
    ; await 1 を入れる

;-----
;
;               メインルーチン
;-----

i=0

*mon01
cls

    fontsize=18
    font " MS ゴシック ",fontsize

    pos 20, 20

mes " "      ; 最初の下線だけのページのレイアウトと
mes " "      ; 合わせるために上に 2 行分の空行を

mes R(i)      ; 刺激の提示

gosub *RTs      ; 反応時間の計測開始

osu=""      ; 何が押されたかの変数を空に
res=""

repeat      ; 反応を待つ

    stick res    ; 入力されたキー情報の取得

    if res&4 {
        osu=">" ; 4 は右向き矢印 (→) キー
        gosub *RTe ; 計測を停止し
        b = i      ; 何番目の単語か
        b = str(b) ; 文字列型に変換してから記録
        noteadd b+"\t"+S(i)+"\t"+jikan+"\t"+osu
        ; ログに追加記録
        i = i + 1
        break      ; loop からぬける
    }

```

```

if res&1 {
    osu="<" ; 1は左向き矢印 (←) キー
    gosub *Rte ; 計測を停止し
    b = i
    b = str(b)
    noteadd b+"\t"+S(i)+"\t"+jikan+"\t"+osu
    ; ログに追加記録
    i = i - 1 ; ひとつ戻る

    if i<0 : i + 1 ; 戻り過ぎないように
;    goto *mon01 ; 前に戻る (repeat と goto をあわせたらネストが
;    ; 深すぎるというエラーが出たのでやめた。
    break
}

await 1: loop ; repeat は、loop まで
; await を入れないと無限ループ

cls ; 画面をクリアー

if i<j : goto *mon01 ; 問題数の上限まで繰り返す

;-----
*owari ; プログラムの終了

mouse ; マウスカーソルを表示 (元に戻す)
pos 50,50
font "MS 明朝",32
mes "これで終わります。"
mes " "
font "MS 明朝",12
mes "#・単語・ミリ秒・方向" ; ★実際の実験ではコメントアウトする

mes kiroku ; 記録の表示 (確認のため) ★実際の実験ではコメントアウトする

; 記録の日時をログファイル名に付ける
yy = gettime(0) : yy = str(yy)
mm = gettime(1) : mm = strf("%02d",mm) ; 一桁の場合は0がつくように2桁で
dd = gettime(3) : dd = strf("%02d",dd)
hh = gettime(4) : hh = strf("%02d",hh)
mn = gettime(5) : mn = strf("%02d",mn)
ss = gettime(6) : ss = strf("%02d",ss)

itu = "kiroku"+yy+mm+dd+"-"+hh+mn+ss+".txt"

mes "保存ファイル名は"
mes itu
mes " "

bsave itu,kiroku ; ファイルに保存

button "End", *owari2

stop

*owari2

end ; プログラムの終了

;----- サブルーチンは最後に書く
;-----
*RTs ; リアクションタイム計測開始

hajime = 0 ; まず変数を空に

ji = 0
hun = 0
byou = 0

```

```

miri = 0

ji = gettime(4); : ji = strf("%02d",ji)
; hun = gettime(5) : fun = strf("%02d",hun)
hun = gettime(5); : hun = strf("%02d",hun)
byou = gettime(6); : byou = strf("%02d",byou)
miri = gettime(7); : miri = strf("%02d",miri)

hajime = (hun*60000)+(byou*1000)+miri

return

*RTe      ; リアクションタイム計測終了

kokomade = 0      ; 変数を空に
jikan = 0

jie = 0
hune = 0
byoue = 0
mirie = 0

;gettime jie,4      ; ここのところの変数名を開始時のものと変えた
;gettime hune,5
;gettime byoue,6
;gettime mirie,7      ; ミリセカンドのレベルで調べる

jie = gettime(4); : jie = strf("%02d",jie)
hune = gettime(5) ;: fune = strf("%02d",hune)
byoue = gettime(6) ;: byoue = strf("%02d",byoue)
mirie = gettime(7); : mirie = strf("%02d",mirie)


kokomade = (hune*60000)+(byoue*1000)+mirie

if ji < jie : kokomade = kokomade + 3600000 ; もし時間をまたいだ場合は
; 1時間分のミリセカンドを足す

jikan = kokomade - hajime ; 引き算をしてかかった時間を計算

return

;-----
;-----END-----

```