認知ゲーム実験（9）卒業研究より3

山上　暁

Cognitive Game Experiment (9): Some Graduation Studies 3

YAMAGAMI Akira

Abstract : The ninth report on the cognitive game experiments is concerning about the student graduation studies in 2009 year. The game experiment situations were produced on Windows XP PCs with programs of the HSP language. Three graduation studies were described.

(1) The first study examined the superiority of global features in the human letter reading. The 'local' letters were one of alphabets or hiragana Japanese letters. The 'global' letters (ex. H) were consisted from one of smaller 'local' letters (ex. S). The mean reaction times to identify the designated letter were longer in the conflict condition than those in the consistent- and the neutral- conditions in both letter sets.

(2) The second study examined the importance of the 'contextual' information in the human scene cognition. The mean recognition times in the 'jumble' condition, where 6 pieces of a whole scene photograph were mixed, were longer than those of in the 'normal (unjumbled)' condition.

(3) The third study examined the transformation axis effect in the mirror image tracing. The performance was measured as the time necessary to complete the tracing task, the number of deviation error and the total time of deviation. All data indicated the performance under the left-right transformation in the diamond figure condition was worse than in the circle and the square figure. These results showed the combination of the oblique figure element and the left-right transformation was most difficult.

Key Words : cognitive game experiment, superiority of global feature, importance of contextual information, mirror image tracing

2009年度卒業研究での認知ゲーム実験

2009（平成21）年度の卒業研究のうち認知ゲーム実験の形のものを，著者がここ数年ゲームを認知機能の実践フィールドとしてとらえて行なってきた「認知ゲーム実験」シリーズ（山上，2006–2010）の一部として紹介する。2009年度も心理学科での卒業研究での実験的研究や実験実習における基礎的心理学実験の中で認知機能を対象にして研究を実施した。認知心理学実験におけるゲームの利用という手法を取り入れた卒業研究のうち「森を見て木を見ず？」・「分割提示課題における画像知覚実験」・「鏡映描写における変換の種類と効果」の3研究を取り上げ，以下にその卒業研究3論文を紹介する。

1. 森を見て木を見ず？

太田侑子と香山久美は「森を見て木を見ず？」というタイトルの卒論（太田・香山，2010）を書き，その中で先行研究であるNavon（1977）の“Forest before trees”の拡張追試実験を行なった。Navon（1977）はアルファベットの大文字を用いて，例えばSという文字をHの字の形に配置して，大きく見たとき（グローバル条件）と小さい構成要素の文字を見たとき（ローカル条件）で正しく文字を読み取るまでの反応時間がグローバル条件の方が早いことを見出し，人間の近く認知機能におけるグローバル処理の優先性と不

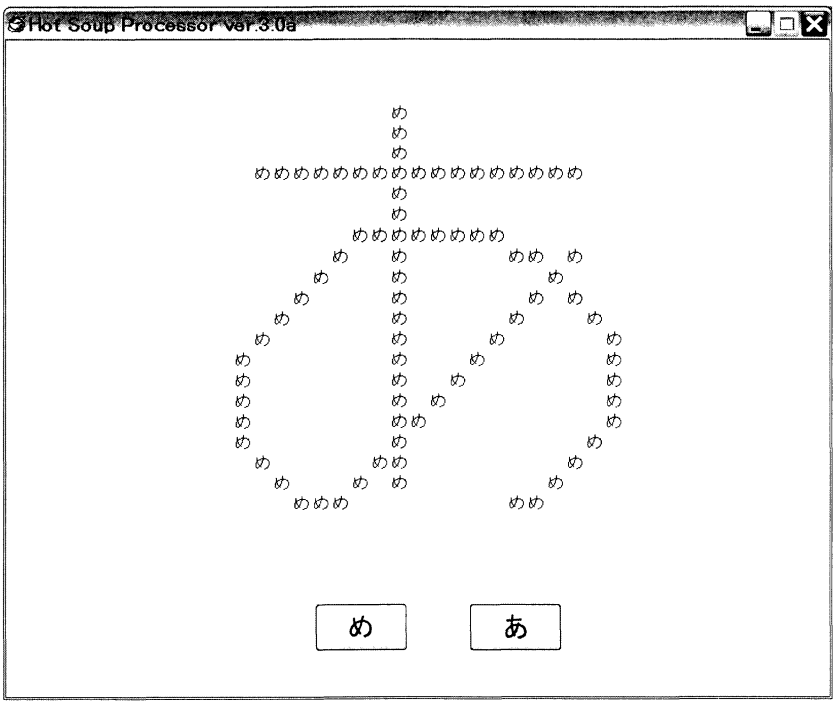

図1　「森を見て木を見ず？」実験の刺激の1例。グローバル条では「あ」，ローカル条件では「め」が正答となる。回答者は刺激文字の下にあるボタンに対応させたキーボード上のテンキーで答える。

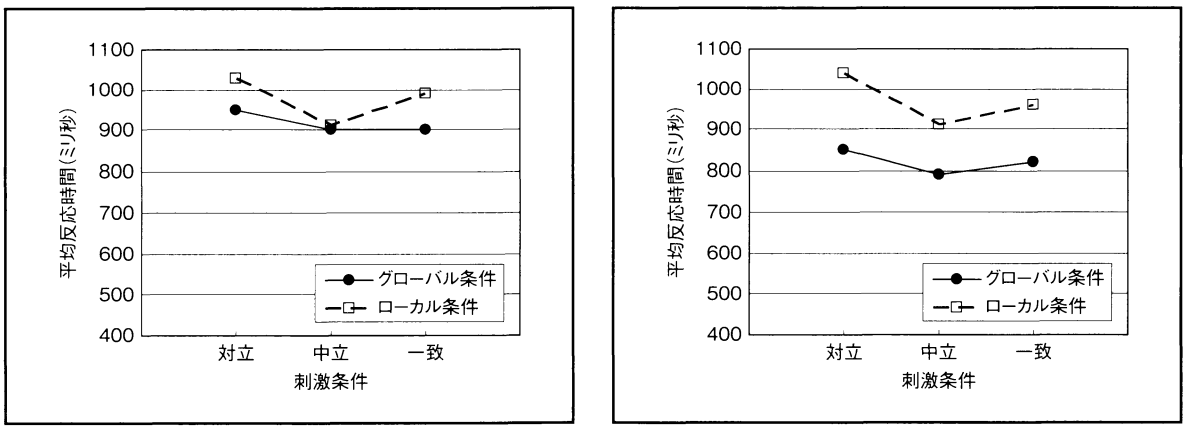

図2　（A）は第1群のグローバル条件先行群，（B）に第2群のローカル条件先行群の各条件での平均反応時間（ミリ秒）

可避性を主張した。太田と香山（2010）はこれをアルファベットの大文字に加えてひらがな文字（図1参照）を刺激として採用し，実験を行なった。

方法と手続

実験参加者は甲南女子大学学生30名。実験参加者を15名ずつの2群に分け，第1群グローバル条件を先に実施し第2群はローカル条件を先に行なった。刺激文字は「アルファベット条件」と「ひらがな条件」のそれぞれで，グローバル条件での読みとローカル条件での読みが異なる「対立条件」とそれらが同じ「一致条件」とローカルな構成要素としての文字のかわりに黒丸を配置した「中立条件」が組み合わされて作成された。実験参加者の課題は指定された条件内で指示された文字の読みを出来るだけ速く，かつ正確にキーによって反応することであった。

実験計画：実験デザインは3要因（2水準×2水準×10水準）計画で，第一の要因は被験者間比較要因で先行順序条件である「グローバル条件」・「ローカル条件」であり，第二の要因は被験者内比較要因としての2文字種の「アルファベット条件」・「ひらがな条件」であり，第三の要因は被験者内比較要因としての読み条件である「対立条件」・「一致条件」・「中立条件」であった。刺激は各文字種で6刺激用意したので，一人の実験参加者は2先行条件×2文字種×3読み条件×6刺激の合計72試行が各先行条件内でランダム順に行わ

れた。

結果と考察

図2（A）に第1群のグローバル条件先行群，（B）に第2群のローカル条件先行群の各条件での平均反応時間（ミリ秒）を示した。（A）のグローバル条件先行群ではグローバル条件では対立条件の反応時間が他の条件より遅く，ローカル条件では中立条件が1番早く答えられて対立と一致ではほぼ同じ平均反応時間となった。対立条件でグローバル条件の方が早かったことはNavon（1977）の結果に一致したが，一致条件でも同じ傾向となった。（B）のローカル条件先行群の各条件での結果では3つ読み条件すべてでグローバル条件の方が早い結果となった。このように，被験者間比較要因として「グローバル条件」・「ローカル条件」を設定すると，どちらを先にするかによって結果が異なったのは，実験中の学習の効果と見られる。また，「アルファベット」と「ひらがな」の刺激の比較では，全体的に「ひらがな」の方が反応時間が大きかった。

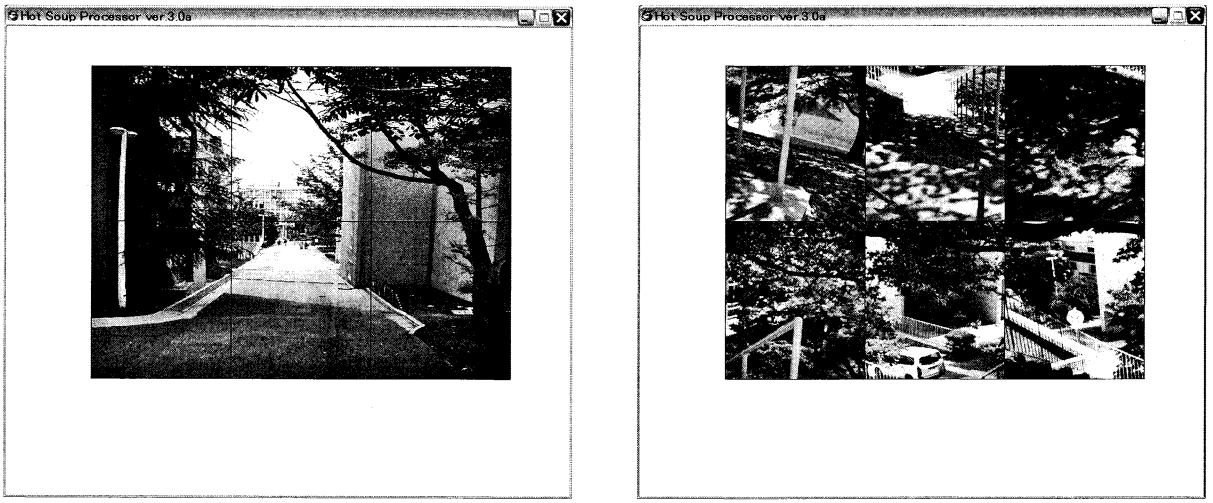

図3　分割提示課題における風景画像刺激。左が「ノーマル条件」で右が別の写真の「ミックス条件」の例である。実際の実験では100枚の個人宅の外見写真が用いられた。ここでの例は予備実験で使用した学内風景の一部である。

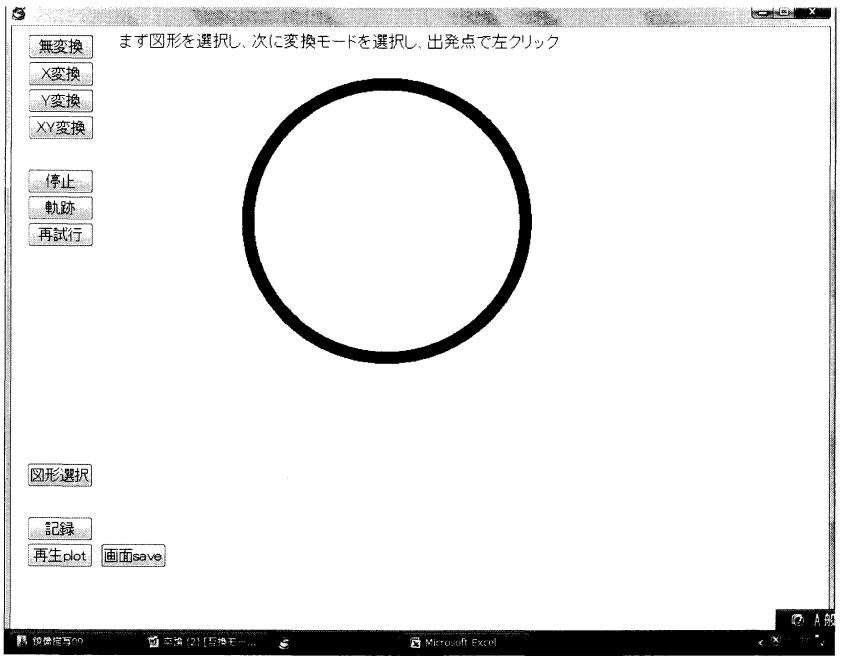

図4　鏡映描写の実験の「円」図形条件。変換条件は画面左上の4つのボタンで指定する。試行の出発点は円の黒い部分の真下方向から少し左の点で示されている。試行中にはマウスのポインタは表示されず，変換された座標値で緑のはっきり見える小さい点が表示された。

2. 分割提示課題における画像知覚実験

藤井由涼菜は「分割提示課題における画像知覚実験」（藤井，2010）というタイトルでBiederman（1972）の風景画像認知における文脈効果の実験の追試を行なった。Biederman（1972）は図3のような風景写真を横3列×縦2行に分割した刺激を用いて，それらをもとのような並びで呈示する「ノーマル条件」と並びをランダムにした「ミックス条件」で短時間呈示された直後に6分割画像のひとつのターゲット画像を呈示した。実験参加者にはそのターゲット画像が直前の6枚からなる全体画像の中にあったか，なかったかを出来るだけ速く，かつ正確に反応することを求めた。その結果ではミックス条件での正答率が低くなり，画像認知において当該部分のみでなく画像全体がもつ文脈の効果存在し，いわゆる「トップダウン処理」が行なわれていることが主張された。

方法と手続

実験参加者は甲南女子大学学生20名。刺激画像は実験者が撮った個人宅の外見のカラー写真100枚を画像加工ソフトで640×480ピクセルのモノクロにしたもの。ターゲット部分は先行画像の一部である場合を50試行，他の画像の一部である場合を50試行用意した。全体画像の呈示時間は1秒とし，合計100画像を各人で異なるランダム順で呈示した。実験参加者の課題はそのターゲット画像が直前の6枚からなる全体画像の中にあったか，なかったかを出来るだけ速く，かつ正確にキーボード上の反応キーを押して反応することであった。

実験計画：独立変数は刺激条件で被験者内比較の1要因2水準（「ノーマル条件」50枚と「ミックス条件」50枚）であり，各人の正答率と平均反応時間（ミリ秒）が従属変数であった。

結果と考察

実験参加者20名の正答数と反応時間を集計した結果，平均反応時間と平均正答数は「ノーマル条件」では1137ミリ秒と17.5個，「ミックス条件」では1673ミリ秒と8.0個で，どちらの指標でも「ノーマル条件」の優位性が認められた。

3. 鏡映描写における変換の種類と効果

古谷妙・松岡雅美は「鏡映描写における変換の種類と効果」（古谷・松岡，2010）において，山上（2006）が行なった鏡映描写の実験の図形条件を変更して拡張的追試実験を行なった。山上（2006）では鏡映描写課題における変換条件の効果を実験的に検討した。マウスを右に動かした時に画面上でポインタが左に動く場

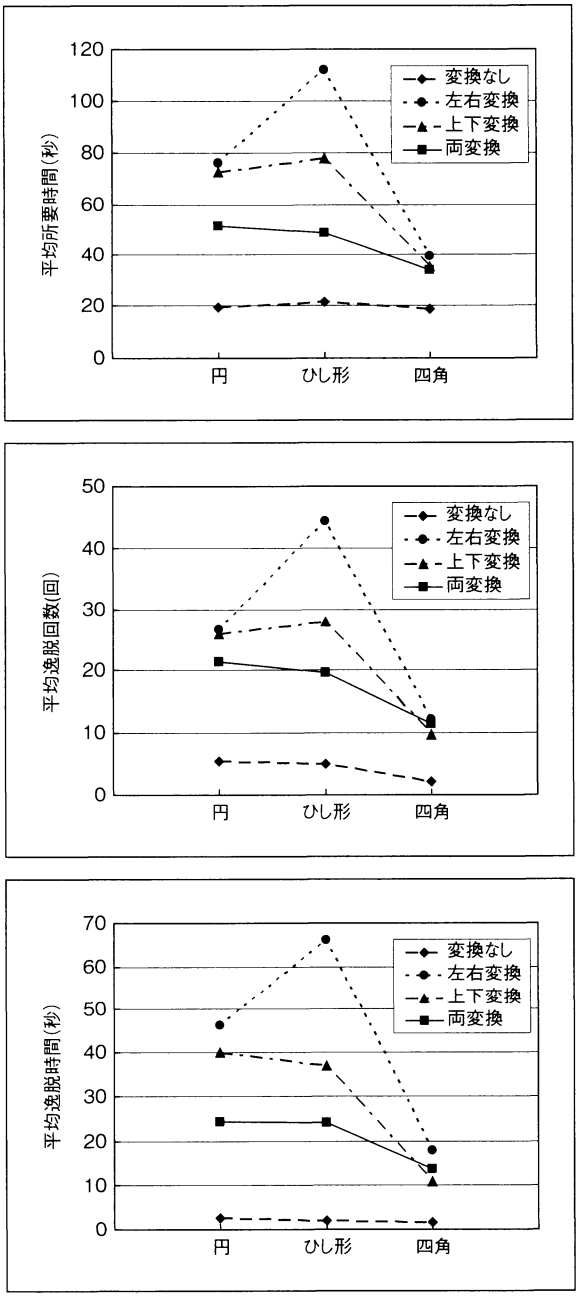

図5　鏡映描写の実験の結果。（上）には図形3条件を横軸にとった変換条件ごとの平均所要時間（秒）が示されている。同様に，（中）は平均逸脱回数（回），（下）は平均逸脱時間（秒）である。

合を「左右変換条件」，マウスを上に動かした時に画面上でポインタが下に動く場合を「上下変換条件」，その両方を同時に変換した「両変換条件」の3条件での遂行成績を比較した。トレースする課題の図形は画面上で幅6 mm，長さ100 mmの直線で，呈示方向は水平・垂直・右斜45度・左斜45度の4方向とした。33名の実験参加者の平均所要時間と平均逸脱時間を比べると，「左右変換条件」が一番難しく，その次に「上下変換条件」で，「両変換条件」が一番容易だった。直線図形の方向の効果は「左右変換条件」で一番顕著で，「左斜45度」>「右斜45度」>「垂直・水平」となり，「上下変換条件」でも同じ傾向であった。古谷・松岡（2010）は図形条件を少し複雑にして，円・ひし形・四角を課題図形にした（図4参照）。

方法と手続

実験参加者は甲南女子大学学生18名。変換条件は上で述べた3条件に統制条件として変換のない「無変換条件」を加えた4条件。トレースする課題の図形は円・ひし形・四角の3種で，「四角」はノートパソコンの15.4インチ液晶ディスプレイ画面上で幅4 mm，辺の長さ72 mmの黒色の正方形であり，「円」は正方形の4辺の合計と同じ長さの円周を持ち，「ひし形」は正方形を45度傾けたものを用いた。実験参加者の課題は指定された条件で課題図形の出発点から出発し，出来るだけ速く，かつ出来るだけ逸脱しないように課題図形の黒い図形部分をなぞり，出発点にもどることであった。

実験計画：実験デザインは被験者内比較要因2要因（4水準×3水準）計画で，第一の要因は変換条件である「左右変換条件」・「上下変換条件」・「両変換条件」・「無変換条件」であり，第二の要因は課題図形の「円・ひし形・四角条件」であった。一人の実験参加者は3変換条件×4図形条件の合計12試行を，順序効果が全体で相殺されるようあらかじめ計画された個人ごとの順で行なった。

結果と考察

図5は実験の結果で図形3種と変換4条件のデータを示している。図5の（上）では図形3条件を横軸にとった変換条件ごとの平均所要時間（秒）が示されている。（中）は平均逸脱回数（回），（下）は平均逸脱時間（秒）である。どの指標でも「ひし形」での左右変換条件が1番難しく，次に「ひし形」の上下変換と「円」の左右変換と上下変換，第3番目に「ひし形」と「円」の両変換という形になっている。垂直・水平成分だけの「四角」はどの変換でも同程度の値で，最も簡単であった。図6は各条件の軌跡をプロットした結果の1例であるが，左図のようにひし形図形での左右変換条件ではコーナーでも直線上でも逸脱が数多く見られ，それに従って所要時間や逸脱時間も長くなった。図6の右図の四角図形での上下変換条件ではそれに比べてずっと逸脱が少ないのがはっきり分かる。

ひし形図形では直線がすべて斜め成分であり，円図形も全ての部分で上下左右の斜め成分が連続していると考えると今回の実験も山上（2006）と同じく図形の斜め成分と上下変換という組み合わせが一番困難な条件であったことを示しているといえる。さらに「ひし形」では4つのコーナーを曲がる時に特に困難なようであった。この実験プログラムでは全ての課題達成プロセスが時間軸上の細かい単位で，座標値や時間データとして保存されている。図6のような逸脱軌跡プロ

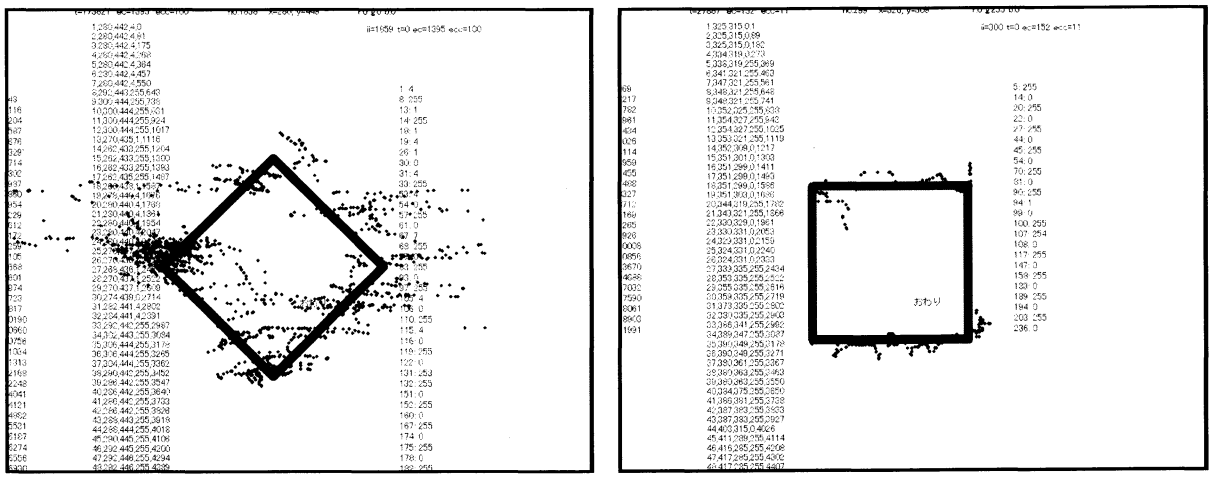

図6　鏡映描写の実験の軌跡をプロットした結果（画面上の数値は座標値や時間のデータである）。（左）はひし形図形での左右変換条件の例。（右）は四角図形での上下変換条件の例。

ットやそれらのデータをもとに非常にきめの細かいデータ分析が可能なので，そのような観点からの分析は今後も続けていこうと思う。

（注）本研究は甲南女子大学より平成20年度教育・学習方法等改善支援経費－教育・学習方法等の改善計画の「認知心理学実験におけるゲームの利用」として補助を受けた。

引用文献

Biederman, I 1972 Perceiving Real-World Scenes. Science 177, 77–80.

藤井由涼菜　2010　分割提示課題における画像知覚実験　甲南女子大学人間科学部平成21年度卒業論文（未公刊）

古谷妙・松岡雅美　2010　鏡映描写における変換の種類と効果　甲南女子大学人間科学部平成21年度卒業論文（未公刊）

Navon（1977）. Forest before trees : The precedence of global features in visual perception. Cognitive Psychology, 9, 353–383.

太田侑子・香山久美　2010　森を見て木を見ず？　甲南女子大学人間科学部平成21年度卒業論文（未公刊）

大槻有一郎　2005　12歳からはじめるHSP 3.0　ゲームプログラミング教室　ラトルズ

山上　暁　2006　認知ゲーム実験（1）鏡映描写　甲南女子大学研究紀要　人間科学編　**42** 7–11.

山上　暁　2007　認知ゲーム実験（2）神経衰弱ゲーム　甲南女子大学研究紀要　人間科学編　**43** 1–8.

山上　暁　2008　認知ゲーム実験（3）ストループ効果　甲南女子大学研究紀要　人間科学編　**44** 1–8.

山上　暁　2009 a　認知ゲーム実験（4）卒業研究より　甲南女子大学研究紀要　人間科学編　**45** 1–10.

山上　暁　2009 b　認知ゲーム実験（5）認知地図の整列効果　甲南女子大学研究紀要　人間科学編　**45** 11–19.

山上　暁　2010 a　認知ゲーム実験（6）視覚的注意分割　甲南女子大学研究紀要　人間科学編　**46** 95–103.

山上　暁　2010 b　認知ゲーム実験（7）卒業研究より　2　甲南女子大学研究紀要　人間科学編　**46** 105–112.

リスト 1　HSP 言語による鏡像描写実験のプログラム

```
1  //-------------------------------------------------------------------------------------
2  // (HSP3)        鏡像描写09-3.hsp
3  //                                    yamgama akira   09601-                MSPゴシック(11)
4  //-------------------------------------------------------------------------------------
5  #include "tmanage3.as" :// 測定には gmsec(), 待機には sleep を使う
6  #pack   "tmanage3.as" :// 実行ファイル作成
7  //-------------------------------------------------------------------------------------
8  :// ズームが必要な図形はバッファに取り込んでおく
9    //buffer 2: gosub *sikaku1:// バッファ2 に多色正方形
10   //buffer 3: gosub *vline:// バッファ3 に垂直棒
11   //buffer 4: picload "sikaku2.jpg":
12   //buffer 5: picload "star3-1(BW).bmp"://バッファに画像
13   //buffer 6: picload "spiral1-0.jpg":// "spiral1-0.jpg":
14   //mmload "start2.wav",7,0 :// 逸脱警告音の用意
15 //-------------------------------------------------------------------------------------
16 *PREPARATION :xx0=1000:yy0=800:screen 0,xx0,yy0,,160,00 :cls : randomize
17   ii=0: objmode 2 :z=18: font "sytemfont",z : objsize 120,24
18   bb0=200: bb1=20: bb2=36: bb3=bb0+90: bb4=bb3+90: bb5=bb4+120
19 //-------------------------------------------------------------------------------------
20 *dimension : nn=0: zz=99: zzz=99999// 配列変数の宣言
21   dim NN,zz: :dim MN,zz : dim par1,zz : dim par2,zz
22   dim Q1,zz : dim SS,zz : dim R99,zz  : dim R9,zz : dim MN,zz: sdim bname,zz
23   dim px,zzz :dim py,zzz :dim rt,zzz :dim bb1,zzz :// 位置と色の記録
24
25 //goto *start0:
26 goto *namae  :// et="t": rn=0: name="": goto *start0
27 stop//-------------------------------------------------------------------------------------
28 *rand :// 1 から N1 まで N9 個ずつのランダム数列(配列ss()に)(今回不使用)
29   N1=24 :N9=1:i=0: rpp=0:  :rx=630
30 repeat : p=rnd(N1)+1 :Q1(p)=Q1(p)+1
31   if Q1(p) > N9 : continue
32   i=i+1: rpp=rpp+1 : SS(rpp)=p:pos rx,i*z :mes str(rpp)+"   :" :pos rx+30,i*z :mes p :
33   if rpp >= N1*N9 :  :break
34 loop : pos 30,460 : goto *testpattern:
35 //-------------------------------------------------------------------------------------
36 *testpattern :color 0,0,0 : X=400:Y=320 : L= 100 : //cls: テストパターン
37   CIRCLE X-L,Y-L,X+L,Y+L,0: LINE X-2*L,Y,X+2*L,Y:LINE X,Y-2*L,X,Y+2*L
38   X=xx0-1:Y=yy0-1:LINE 0,0,X,0: LINE X,0,X,Y:LINE X,Y,0,Y: LINE 0,Y,0,0
39   goto *namae ://pos 30,500 : BUTTON "next",*start0: stop
40 stop//-------------------------------------------------------------------------------------
41 *namae:name="":pos 30,10:mes "name=":pos 86,10:input name:objsel 2://被験者番号入力
42   //pos  30,bb1+bb2*1+60 : BUTTON "OK"   ,*start0://*buttons
43   pos 450,bb1+bb2*1 : BUTTON "start",*start0 :// *jj1
44  stop://pos 450,bb1+bb2*3 : BUTTON "ボタン選択",*jj0
45 *jj1 :rn=1: goto *rand ::: *jj0 :rn=0: goto *testpattern
46 stop//-------------------------------------------------------------------------------------
47 *figureset :zm1=1: :// ボタン設定/出発点の座標(x9,y9)は手動測定する
48   if nn=1: x0=280: y0=  60: x9=417: y9=403: pos x0,y0: picload "maru2.jpg",1:
49   if nn=2: x0=260: y0= 250: x9=272: y9=446: pos x0,y0: picload "daiya2.jpg",1:
50   if nn=3: x0=320: y0= 300: x9=327: y9=313: pos x0,y0: picload "sikaku2.jpg",1:
51   :// 画像ズーム貼り付け(座標x0,y0):// ボタン設定/出発点の座標(x9,y9)
52   :// button no. / set position(xy) / size(xy) / zoom ratio / start point(xy)
53   if nn=4: buf=4:x0=320:y0=  300:fx=599:fy=599:zm2=1:x9=327:y9=313:gosub *zoom1://
54   if nn=5: buf=5:x0=280:y0=  60:fx=599:fy=599:zm2=2:x9=337:y9=205:gosub *zoom1://
55   if nn=6: buf=6:x0=280:y0= 160:fx=599:fy=599:zm2=2:x9=337:y9=205:gosub *zoom1://spiral1
56   if nn=7: an0=00.0: x9=401: y9=197: w0=160: h=160: gosub *ang: //square
57   if nn=8: an0=00.0: x9=401: y9=197: w0=20:  h=300: gosub *ang: //vertical
58   if nn=9: an0=45.0: x9=401: y9=197: w0=20:  h=300: gosub *ang: //oblique(45)
59   if nn=10:an0=90.0: x9=401: y9=197: w0=20:  h=300: gosub *ang: //horizontal
60   if nn=11:gosub *ring1
61   if nn=12:an0=45.0: x9=401: y9=197: w0=160: h=160: gosub *ang: //diamond
62 return: //----------------------------------------------------------------------------
63 *buttons: cls: ii=ii+1: // 刺激題番号ボタン行列を表示する(smallest version)
64   repeat zz: bname(cnt)=str(cnt):loop:// ボタンの名前(図形)
65   bname(1)="maru2": bname(2)="daiya2":   bname(3)="sikaku2"  :
66   bname(4)="4":
67   bname(5)="5":     bname(6)="6": bname(7)="7" :  bname(8)="8":
68   bname(9)="9":   bname(10)="10": bname(11)="11":  bname(12)="12":
69  // bname(4)="/sikaku2":
70  // bname(5)="星型":     bname(6)="渦巻き": bname(7)="四角形" :  bname(8)="垂直線":
71  // bname(9)="右45度":   bname(10)="水平線": bname(11)="リング":  bname(12)="ひし形":
72  //if ii=1:tt1=gettime(3)*60*60*24+gettime(4)*60*24+gettime(5)*60+gettime(6)://開始秒
73
74   font "sytemfont",20:color 000,000,000 ://pos 30,30:mes "< 図形番号 >"://番号ボタン式
75   pos 10,0 : mes "name= " + name:pos 20,100:mes "<<   図形の選択   >>"
```

1. 準備
画面セット
変数配列

2. 乱数作成
テストパターン

3. 開始ページ

4. 図形貼付
微調整

5. 各種ボタン
設定

```
76    repeat zz:stim(cnt)=0:ans(cnt)=0:bn(cnt)=0:bn2(cnt)=0:rt(cnt)=0:yn(cnt)=0:loop
77    x0=24 :y0=40:vn=4: hn=3: h=120: v=40: hh=h+7: vv=v+7  :// 原点とボタンサイズと間隔
78  repeat vn,1:i=cnt:repeat hn,1:j=cnt:  bn=j+(hn)*(i-1) :// ボタン番号 1 から
79    objmode 2: color 000,000,000 :font "sytemfont",20  :// ボタンの文字サイズ
80    objsize h,v :pos x0+(j-1)*hh,y0+(i-1)*vv+h:p1(bn)=0:button bname(bn),*response2
81  loop : loop: font"sytemfont",20: pos 500,560: objsize 160,40: BUTTON "記録",*file1
82    stop: *response2: bn2(stat+1)=1: nn=(stat+1): mes str(nn): gosub *start0:
83  stop://-----------------------------------------------------------------------------------------
84  *zoom1:// zm1=copied   zm2=original(1:2で半分)
85    pos x0,y0:gzoom fx*zm1,fy*zm1,buf,0,0,fx*zm2,fy*zm2,0: return:  // ズーム貼り付け
86  return://---------------------------------------------------------------------------------------
87  *start0: cls : font "sytemfont",20: bx1=20: by=33: by0=10// スタートの図とボタンの設定
88    tt1=gettime(3)*60*60*24+gettime(4)*60*60+gettime(5)*60+gettime(6)://開始秒
89    gosub *figureset:color 000,000,000: // ボタンでトレース図形を選択
90    pos 130,08: mes "まず図形を選択し、次に変換モードを選択し、出発点で左クリック"
91    objsize 80,30: font "sytemfont",18: bx1=20: by=33: by0=10: // ボタンの設定
92
93    pos bx1,by0+by*0    :BUTTON goto  "無変換"   ,*xy:   // 無変換
94    pos bx1,by0+by*1    :BUTTON goto  "X変換"    ,*xy:   // X変換
95    pos bx1,by0+by*2    :BUTTON goto  "Y変換"    ,*xy:   // Y変換
96    pos bx1,by0+by*3    :BUTTON goto  "XY変換"   ,*xy:   // XY変換
97    pos bx1,by0+by*5    :BUTTON goto  "停止"     ,*stop1: // 停止
98    pos bx1,by0+by*6    :BUTTON gosub "軌跡"     ,*trace1:// トレース
99    pos bx1,by0+by*7    :BUTTON goto  "再試行"   ,*again1:// 再試行
100   //pos bx1,by0+by*9    :BUTTON gosub "vertical" ,*vt: //
101   //pos bx1,by0+by*10   :BUTTON gosub "45 deg."  ,*ob: //
102   //pos bx1,by0+by*11   :BUTTON gosub "90 deg."  ,*hz: //
103   //pos bx1,by0+by*12   :BUTTON gosub "rotation" ,*rot://
104   //pos bx1,by0+by*13   :objsize 40,30: an0=30.0: input an0: objsize 80,30
105   //pos bx1+45,by0+by*13: mes "deg"
106   //pos bx1,by0+by*14   :BUTTON gosub "sikaku"   ,*sikaku
107   //pos bx1,by0+by*15   :BUTTON gosub "dia"      ,*dia
108   pos bx1,by0+by*16   :BUTTON gosub "図形選択"   ,*figure1: //
109   //pos bx1,by0+by*17   :BUTTON gosub "ring"     ,*ring1://
110   pos bx1,by0+by*18   :BUTTON goto  "記録"     ,*file1://
111   pos bx1,by0+by*19   :BUTTON gosub "再生plot"  ,*fileplot1://
112   pos bx+110,by0+by*19 :BUTTON gosub "画面save" ,*save1://
113   color 0,255,0: pr1=4: pr2=4: circle x9-pr1,y9-pr2,x9+pr1,y9+pr2:// 出発点の表示
114   objsel 11: stop::: *again1: goto *start0
115 stop://-----------------------------------------------------------------------------------------
116 *xy: cnd=stat+0: repeat:wait 3:getkey ky1,1:if ky1=1: break ://mouse left to start1
117    loop :
118  tt1=gettime(3)*60*60*24+gettime(4)*60*60+gettime(5)*60+gettime(6)://開始秒
119 goto *trial1
120 stop://-----------------------------------------------------------------------------------------
121 *save1: dialog "bmp",17,".bmp(拡張子)をつける": bmpsave refstr: return:// 画面画像保存ダイアログ
122 stop://-----------------------------------------------------------------------------------------
123 *figure1: color 255,255,255: boxf 000,000,xx0,yy0: gosub *buttons:// 白でクリア
124 return//----------------------------------------------------------------------------------------
125 *vline: color 000,000,000: x0=00: y0=0: h0=300: w0=20: boxf x0,y0,x0+w0,y0+h0:// 黒で描く
126   x9=10:y9=02:pr1=4:pr2=4:color 0,255,0:circle x9-pr1,y9-pr2*0,x9+pr1,y9+pr2*2://緑出発点
127 return//----------------------------------------------------------------------------------------
128 //*vt :  an0=00.0: gosub *ang :return
129 //*ob :  an0=45.0: gosub *ang :return
130 //*hz :  an0=90.0: gosub *ang :return
131 //*rot:  an0=an0 : gosub *ang :return
132 return//----------------------------------------------------------------------------------------
133 *ang: // 垂直線回転サブルーチン//pi=57.3 :an0=90.0:// ここの両方を小数点表示することが必要
134  :// x5,y5 は回転 0 度の場合の図形の左上座標値
135    w=160: h=160:
136 x5=300: y5=200: x5=360+(w0/2): y5=200+(h0/2)
137    pos x5,y5: color 255,0,0: circle x5,y5,x5-5,y5-5: //test point(x5,y5 in red)
138    an1=(an0/57.3): pos 120,40: color 0,0,0: mes str(an0)+" ("+an1+" )"
139 if nn= 7 or nn=12: gmode 0,w0,h0: pos x5+w0/2,y5+h0/2: grotate 2,0,0,an1:// sikaku
140 if nn= 8 or nn= 9 or nn=10: gmode 0,w0,h0: pos 300,300: grotate 3,0,0,an1:// vline
141 x9=x5+(w0/2)+sin(an1)*(h0/2)-pr1: y9=y5+(h0/2)-cos(an1)*(h0/2)-pr2
142 circle x9+pr1*0,y9-pr2*0,x9+pr1*2,y9+pr2*2
143 return//----------------------------------------------------------------------------------------
144 *ring1:color 255,255,255: boxf 000,000,xx0,yy0:// 白でクリア
145    color 000,000,000: x0=300: y0=200: w=20: s=200:
146    color 000,000,000: circle x0,y0,x0+s,y0+s:// 黒で描く
147    color 255,255,255: circle x0+w,y0+w,x0+s-w,y0+s-w:// 黒で描く
148    x9=400 :y9=200: pr1=4: pr2=4:color 0,255,0::// 緑で出発点で描く
149    circle x9-pr1,y9-pr2*0,x9+pr1,y9+pr2*2: return
150 stop://-----------------------------------------------------------------------------------------
151 *sikaku1: // buffer 2 の四角 // 地色の違う正方形枠図形を描く(菱形はこれを回転する)
152 color 255,255,255: boxf 110,000,xx0,yy0:// 白でクリア
153 ://上の辺
154 x0=20: y0=00:// x0=640:y0=10
155   //x0=300: y0=200:// x0=640:y0=10
156   ax1=x0: ay1=y0:aL1=40: aL2=60: aL3=30: aw=20
157   ay2=ay1+aw: ax2=ax1+aL1: ax3=ax2+aL2: ax4=ax3+aL3:
158   color 200,100,101: boxf ax1,ay1,ax2,ay2:
159   color 060,120,102: boxf ax2,ay1,ax3,ay2
160   color 120,120,003: boxf ax3,ay1,ax4,ay2
```

6. 条件設定
割り当て
ボタン配置

7. 各種
サブルーティン

8. 各種図形
描画

```
:// 右の辺
  bw=20: bx1=ax4-bw: by1=ay2: bL1=40: bL2=60: bL3=30:
  bx2=bx1+bw: by2=by1+bL1: by3=by2+bL2: by4=by3+bL3:
  color 120,000,004: boxf bx1,by1,ax4,by2:
  color 180,000,005: boxf bx1,by2,ax4,by3:
  color 240,000,006: boxf bx1,by3,ax4,by4:
:// 下の辺
  cw=20: cx1=bx1: cy1=by4: cL1=40: cL2=50: cL3=40:
  cy2=cy1-cw: cx2=cx1-cL1: cx3=cx2-cL2: cx4=cx3-cL3:
  color 190,090,127: boxf cx2,cy2,cx1,cy1:
  color 200,120,008: boxf cx3,cy2,cx2,cy1:
  color 120,120,009: boxf cx4,cy2,cx3,cy1:
:// 左の辺
  dw=20: dx1=ax1-dw: dy1=ay1:dL1=40: dL2=60: dL3=30
  dx2=dx1+dw: dy2=dy1+dL1: dy3=dy2+dL2: dy4=dy3+dL3:
  color 145,000,110: boxf dx1,dy1,dx2,dy2:
  color 200,220,011: boxf dx1,dy2,dx2,dy3:
  color 100,120,012: boxf dx1,dy3,dx2,dy4:
  w0=dw+aL1+aL2+aL3: h0=aw+bL1+bL2+bL3: x8=dx1+(w0/2): y8=dy1+(h0/2)
:// 出発点
  x9=x0-dw :y9=y0: pr1=4: pr2=4:color 0,255,0::// 緑で出発点で描く
  circle x9+pr1*0,y9-pr2*0,x9+pr1*2,y9+pr2*2: return
stop://-------------------------------------------------------------------------------
*trial1: ec=0: ecc=0: x0=x9: y0=y9: t0=gmsec()::// 開始時刻(ミリ秒)
// tt1=gettime(3)*60*60*24+gettime(4)*60*24+gettime(5)*60+gettime(6)://開始秒
  //mouse -1://マウスカーソル消す
  repeat,1: ii=cnt: color 255,255,255 :boxf 0,0,800,60:// 表示クリア
:// マウスポインタ点座標
  mx=mousex: my=mousey: rt(cnt)=gmsec()-t0 :// マウスポインタ点座標と時間記録
:// マーカー(座標変換点)の色取得
  dx=mx-x0: dy=my-y0:// 出発点からのポインタの移動量
  if cnd=0 :px(cnt)=mx    : py(cnt)=my    : pget px(cnt),py(cnt): // 無変換
  if cnd=1 :px(cnt)=x0-2*dx: py(cnt)=my    : pget px(cnt),py(cnt): // X変換
  if cnd=2 :px(cnt)=mx    : py(cnt)=y0-2*dy: pget px(cnt),py(cnt): // Y変換
  if cnd=3 :px(cnt)=x0-2*dx: py(cnt)=y0-2*dy: pget px(cnt),py(cnt): // XY変換
  rr1=ginfo_r:gg1=ginfo_g :bb1(cnt)=ginfo_b :// 座標変換点のオリジナル色記録
:// マーカー(座標変換点)表示
  color 000,255,0: gosub *plot: wait 5:// マーカーの色+ マーカー表示時間
  color rr1,gg1,bb1(cnt) : gosub *plot:// 座標変換点のオリジナル色に戻す
:// 逸脱時処理(bb1(cnt)=青成分で判定、bb1(cnt)=255は白地)
  if bb1(cnt)=255: color 255,0,0: pos 300,0: mes"bb1:"+str(bb1(cnt)):// 逸脱時処理
 //if bb1(cnt)=255: color 255,0,0: boxf 300,0,660,60: // 逸脱警告(赤)
  if bb1(cnt)=255:color 255,0,0:font "sytemfont",36:mes "逸脱しています":// 逸脱警告(赤)
  if bb1(cnt)=255: ec=ec+1://mmplay 7 ://逸脱警告音 : 累積逸脱時間(白でのカウント数)
  if bb1(cnt)=255 and bb1(cnt-1)!=255: ecc=ecc+1: // 累積逸脱回数(白以外から白へ)
:// 座標等表示
  color 0,0,0:  font "sytemfont",16
  pos 120,0: mes "t="+str(rt(cnt))+"  ec="+str(ec)+"  ecc="+str(ecc)
  pos 380,0: mes "no."+str(cnt)  +" :  x="+str(px(cnt))+ ", y="+str(py(cnt))
  pos 600,0: mes "r:" +str(rr1)  +" g:"  +str(gg1)  +" b:"  +str(bb1(cnt))
  wait 3: gosub *key1:loop // 座標等表示時間
stop//--------------------------------------------------------------------------------
*trace1: bx=610: font "sytemfont",16: color 0,0,255 :// トレース(軌跡)+結果表示(変化点のみ)
  pos bx,30:mes " ii="+str(ii)+" t="+str(rt(ii))+" ec="+str(ec)+" ecc="+str(ecc)
  repeat ii,1: pp=0: bbb=0: color 0,0,255://color 120,120,120
  if bb1(cnt)!=bb1(cnt-1): i=i+1: pos bx+60,i*z+42: bbb=1
  if bbb=1:mes str(cnt)+": "+str(bb1(cnt)):pos bx0,i*z+42:mes str(rt(cnt))
  color 000,255,000:gosub *plot:loop: return// トレースの色
stop//--------------------------------------------------------------------------------
*plot :pr1=3: pr2=3:// マーカーのサイズ
  circle px(cnt)-pr1,py(cnt)-pr2,px(cnt)+pr1,py(cnt)+pr2
return//-------------------------------------------------------------------------------
*key1: getkey ky,27 : if ky = 1: goto *stop1: // esc key
      getkey ky,40 : if ky = 1: goto *stop1: // down key
      getkey ky,2  : if ky = 1: goto *stop1: // mouse right
      stick  ky,512: if ky=512: goto *stop1: // mouse right
return
*stop1:
tt7=gettime(3)*60*60*24+gettime(4)*60*60+gettime(5)*60+gettime(6):// 終了秒測定
 mouse : ii=ii+1: stop
//:stop-------------------------------------------------------------------------------
```

9. 試行ループ

10. 軌跡プロット

11. キー入力

```
*file1:// 配列変数(1次元)をCSV形式で保存する
  name=name + "-" + CD :font "sytemfont",16 : color 255,000,000 ://赤で表示
  b=10 : bb=18: ppp = "" : sssss="": kkk=0: CR= "¥r": // CRLF="¥n¥r"
 // tt7=gettime(3)*60*60*24+gettime(4)*60*60+gettime(5)*60+gettime(6):// 終了秒測定
  tt8= (tt7-tt1): // 全所要時間秒測定　// tt9 は記録ファイルのヘーダー
  tt9= name + CR+str(gettime(0))+"-"+str(gettime(1))+"-"+str(gettime(3))
  tt9= tt9  +" "+str(gettime(4))+":"+str(gettime(5))+":"+str(gettime(6))
  tt9= tt9  +"( "+str(tt8)+" sec.) "+CR+"CD=" + CD
  iii="no."+","+"x"+","+"y"+","+"bb1"+","+"time"
  sssss = tt9 + CR +CR+ iii+ CR: // ssss は全試行分の記録ストリング
repeat ii,1 ://全試行の1行分のデータを累積
  ppp= ppp+str(cnt)+","+str(px(cnt))+","+str(py(cnt))+","+str(bb1(cnt))+","+str(rt(cnt))+CR
  sssss= sssss + ppp:// ppp は1試行分の記録ストリング
pos 150,10+cnt*16: mes ppp :ppp = "" :loop :// ppp をクリア
  mes "-------------" +CR : mes sssss  : notesel sssss
  if et ="e" :
  notesave name+".csv"://  実験時のみ記録ファイル作成
  pos 500,500: sysfont: mes "おわり" :
stop://-------------------------------------------------------------------------------------
*fileplot1: objsize 80,30: zz=999:// 配列変数の宣言
  sdim pp,zz: dim yn,zz  : dim RT,zz  :: dim CR,zz  : dim stim,zz : dim ans,zz :
  sdim w,256: row=54: cell=10:// カラム数(cell)ずつ(row)行分を読み取る
  sdim Q,zz,row+1,cell+1: dim A,zz,row+1,cell+1: //sdim A,zz,row+1,cell+1
//-------------------------------------------------------------------------------------------
*filename: ssss="": repeat 5,1: CR(cnt)=0: RT(cnt)=0: loop
  sss9="":sdim txbuf,32000: notesel txbuf
  dialog "csv",16: FF=refstr: mes FF: noteload FF: // ファイル名がフルパスで入る
  FF=strmid(FF,76,15):// 必要なファイル名にカット(66は調整必要、ディレクトリによる)
  font "sytemfont",12 : color 255,000,000 :// 赤で表示
//-------------------------------------------------------------------------------------------
*getitems1:// 1行単位で1行目はcellセルを文字配列に取り込む
repeat row,1: c1=cnt: mm="": i=0:j=0:  c0=cnt ://c0=cnt*2-2
  noteget w,c0://mes w:// 変数wに1行目ゲット
  repeat cell,1:getstr Q(c1,cnt),w,i,',':i+=strsize:mm= mm+Q(c1,cnt)+", ":loop:
loop: font "sytemfont",12: pp=100
//pos 500,120:button goto "next0" , *bwait0: stop : *bwait0: cls // 止めて値をチェック
//データ読み込み//配列Q(1行目から)をm文字配列A(4行目のデータから)に読込み
  rr=9999:ii=rr: m="": color 255,000,000 :// 赤で表示
repeat rr,1: r=cnt: color 255,000,000:// red
pos 0,0
  repeat 5,1: c=cnt:pos 300,c*20+(rr-1)*100:mes r+"-"+c
 A(r,c)=int(Q(r+4,c)):
m= m+str(A(r,c))+", ": loop
  m="": pos pp+020,000+r*12: px(r) =int(A(r,2)): py(r) =int(A(r,3)):
  if px(r)=0 and px(y)=0: break:// X,Y座標値ともゼロで停止
loop: //mes "data end":stop: return
  pos bx1,by0+by*20  :BUTTON gosub  "データ表示"   ,*datashow
  pos bx1,by0+by*21  :BUTTON gosub  "データプロット",*dataplot:
stop: //-------------------------------------------------------------------------------------
*datashow: //データ表示
repeat rr,1: r=cnt:
  color 000,000,255 : pos pp+0220,000+r*12: mes r ://blue
  px(r) =int(A(r,2)): pos pp+0260,000+r*12: mes px(r)
  py(r) =int(A(r,3)): pos pp+0320,000+r*12: mes py(r)
  if px(r)=0 and px(y)=0: break:// X,Y座標値ともゼロで停止
  col(r)=int(A(r,4)): pos pp+0380,000+r*12: mes col(r)
  tt(r) =int(A(r,5)): pos pp+0440,000+r*12: mes tt(r)
loop: mes "data end": return
stop: //-------------------------------------------------------------------------------------
*dataplot: //データプロット
repeat ii,1: pp=0: bbb=0: color 0,0,255://color 120,120,120
  if bb1(cnt)!=bb1(cnt-1): i=i+1: pos bx+60,i*z+42: bbb=1
  if bbb=1:mes str(cnt)+": "+str(bb1(cnt)):pos bx0,i*z+42:mes str(rt(cnt))
  color 000,255,000:gosub *plot:loop: return// トレースの色
stop: //-------------------------------------------------------------------------------------
```

12. データ記録ファイル作成

13. データ記録ファイル設定データファイルから読込みプロット

14. データ表示・プロット