

認知ゲーム実験 (6) 視覚的注意分割

山 上 暁

Cognitive Game Experiment (6): Visual Attention Distribution

YAMAGAMI Akira

Abstract : As the sixth report on the cognitive game experiments, I developed a game experiment to examine the performance of visual attention. The game situations were produced on a Windows XP PC with a program of the HSP language. Ten small square objects were moved on the PC screen in one of eight directions for a short time. Observer's task was to pursue multiple objects (the number of 'target' varied from 1 to 5) which were indicated before their motion. When the motion was ceased, one of 10 objects (included the targets) was flashed and the observer was asked whether it was the one of the targets indicated before motion. The first experimental factor was the number of target and the second one was the speed of target (slow /medium/fast). The third one was the learning effect which was measured with the difference between the first and the second experimental session performances within each speed conditions. The mean reaction time (RT) of 32 observers in each of all 6 experimental sessions indicated, (1) the means of RT increased with the number of target, (2) they decreased with the speed of motion and (3) the learning effect was greatest in the slow speed condition. The mean number of the correct reaction (CR) of 32 observers in each of all 6 experimental sessions indicated (4) the means of CR decreased with the number of target, (5) they decreased with the speed of motion and (6) the learning effect was not observed with this measure.

Key Words : cognitive game experiment, visual attention, multiple object tracking

目 的

一度に多くの対象を目で追うのはむずかしい。目の動き（眼球運動）という意味では一度に一方向にしか目を動かすことができない。しかし、比較的狭い視野範囲のなかではいくつかの対象に注意をほらうことが出来る。例えば混雑した駅の中で歩いている時には前や横の人々の動きを察知しながら、自分の歩く方向や速度を調節している。車を運転する場合にも、周りの車や信号や自転車や歩行者などの多くの対象の動きを読みながら事故の起こらないようにしなければならない。また、ゲームになれた人は画面に表示される多くの情報を一瞬の間に読み取ってゲームの進行に役立っている。このように、われわれは日常的に注意を分割して行動しなければならない場面も多い。今回の実験

は「認知ゲーム実験」(山上, 2006-2009)のひとつとして開発されたもので、視覚的注意分割課題における追跡対象の数の効果の実験的検討という形で、多くの動く対象に注意を払う場合に対象の数がどのような効果を持つのかを調べるのが目的である。

注意 (attention) は第一にひとつの事柄に選択的に「注意する」(selective attention)「集中する」(focal attention)「焦点をあてる」(attention focusing)という意識 (consciousness) の焦点化・集中を意味する。他方ではそれと逆に、同時に二つ以上の事柄に意識を分割する (divided attention) という側面を持つ。視覚的注意 (visual attention) の場合には多くの視覚的対象のひとつをじっと見つめる (凝視する) という場合が前者の焦点的注意 (focal attention) に対応するし、多数の視覚刺激に注意を分割する事態は後者と考えられる。多くの対象に注意を分割する場合には、ひとつずつの対

象の処理（系列処理, serial processing）を短時間でつぎつぎと移動させている（時間的分割処理, time sharing processing）のか、実際に同時に多くの対象を処理（空間的分割）している（並行処理, parallel processing）のかが問題となる。注意の働きには「意識が狭く深くひとつに集中する」という働きと「意識を拡散的に広く浅く保つ」という両極端があり、通常はその中間段階をさまざまな要因の影響を受けて変動（注意の変動）していると考えられる。ナイサーは焦点的注意（focal attention）と前注意的過程（preattentive process）という区別でこの2つの処理方式に対応した注意の機能を区別した。他方、注意は意識的に注意を向けるという意味での「能動的注意」と突然大きな音がした場合などのように注意が自然に向くあるいは惹きつけられるという意味での「受動的注意」の区別も存在する。

注意をどのようなものとして捉え、どのような働きと見るか（注意のモデル）はさまざまなものが提案されてきているが、アンダーソンは視覚的注意を空間を動きまわるスポットライトの比喩で述べている。また、カーネマンは注意のモデルをコンピュータ情報処理の比喩を使って次のような形で述べている。人が情報処理を行うためにはその対象に注意し、処理資源をつぎ込む必要があり、それを注意資源（attentional resources）と名づけた。注意資源には一定の処理容量限界（capacity limit）があり、人間はその範囲内でうまく物事を処理していく認知活動（cognitive activities）

を行っている。単純な情報処理では注意資源の多くを使う必要がないので、同時に他の対象に注意を分割して情報処理を行うことができる。ひとつの情報処理の課題が複雑な場合にはより多くの注意資源が必要であり多くの心的努力（mental effort）が必要となり、他の課題を同時に行うことができないのはもちろん、その課題にもエラーが多くなったり、反応時間が長くなったりする。

さらに注意の時間的変化を考えると、処理の自動化が問題となる。はじめは同時に多くのことができなかつたことでも経験や練習を重ねるとほとんど注意を払わずに無意識的に（unconsciously）できるようになる。シュナイダーとシフリンはこのような注意を必要としない自動処理（automatic processing）と注意を必要とするコントロール処理（controlled processing）を区別し、文字検出実験で、はじめはコントロール処理であったものが試行回数が増大によって（2100 試行後）自動処理に移行することを実験的に示した（菊池・八木, 2007；小谷津, 1981；塩入, 2007；渡辺, 1999）。

今回の実験では多数の対象を追跡する視覚的課題である多物体追跡（multiple object tracking；MOT）法を用いて、追跡すべき対象の数が対象の認知にどのように影響するかを調べる。これまでの研究（Pylyshyn & Storm, 1988）から対象の数が多くなると、(1) 反応時間が長くなり、(2) 正答数が減少する、という予想（仮説）が成り立つ。

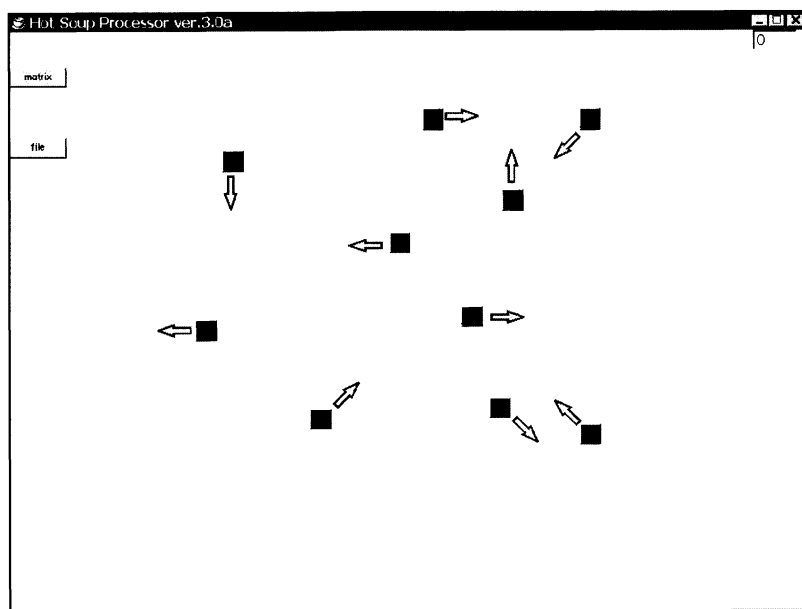


図 1 多物体追跡課題事態（10 個の灰色四角がいろいろな方向に同時に移動）の例（図中の矢印は移動の 8 方向を示す。実験時には表示されない。）

方 法

PC画面上に10個の小さな四角形図形がランダムな位置に提示され、同時にいろいろな方向に動く多物体追跡課題事態をHSP言語によるプログラムで作成した。実験参加者は心理学科学生32名で、各自1台のPCを用い、教示以外は自動化された実験として実施した。Windows XPのPCを用い、ディスプレイ画面は常に被験者の正面にあるようにした。ディスプレイの前の約60cmの距離に実験参加者が座った。甲南女子大学5号館のPC教室で行った。一人当たりの1セッションの所要時間は約4~7分であった。

刺激：PC画面の中心部分に5×5の格子座標を想定し、そこからランダムに選ばれた10の座標点から10個の小さな灰色四角形が1.2~1.8秒間動く。移動方向は45度刻みの8方向からランダムに選ばれた1方向であった。移動の前に10個の移動四角形のうち1個から5個がランダムに指定されて、青く囲まれ3回点滅することで「ターゲット」（追跡対象）となる。ターゲットを含む10個が移動したあとに、10個のうちの一つがランダムに選ばれて今度は赤く囲まれる。回答者の課題はこの最後の赤くマークされたものが、はじめのターゲットのひとつだったかどうかをできるだけ間違えないようにできるだけ速く「イエス」か「ノー」のどちらかのキーを押して答えること（強制選択法）であった。反応はキーによって行い、「イエス」・「ノー」キーの右手と左手への割り当ては回答者ごとにランダムにした。反応キーと刺激が赤くマークされてからキーを押すまでの反応時間（単位はミリ秒）が記録された。

実験計画：実験デザインは、3要因（5水準×3水準×2水準）の被験者内比較デザインであった。第一の要因は追跡すべき対象の数（1~5）であり、第二の要因は10個の対象の移動速度（低・中・高速度）、第三の要因は3つの速度別セッションを1ブロックとして、2ブロック行うことによる練習要因（1回目と2回目）であった。ひとつのセッションでは追跡対象の数が1つから5つまでの5水準がランダムに設定され、10個の移動対象の最初の位置と移動方向はランダムに選択されて、5水準をそれぞれ10回ずつ測定するため5条件×10回の合計50試行がランダム順で連続して行われた。ひとつ目のブロックが低・中・高速度の順で実施された後、10分の休憩の後にもう一度ずつ低・中・高速度の順でもう1ブロック行われ、

全部で2ブロック6セッションが実施された。

手続：(1) 各セッションでの課題は、はじめに1から5個の間で指示された対象を含10個の移動対象がとまったときに、指示された1個がはじめに指示されたものの1つかどうかを判断し、キー反応することであった。あまり深く考え過ぎないことや回答（反応）は、能力や性格に関係することもあるが、今回はその観点からは分析しないことなどを教示として最初のセッションの前に伝えた。

(2) コンピュータプログラムの操作手順に従って実験を進めた。はじめに数回の練習試行を行い課題の理解を確認すると同時に反応の仕方に慣れさせた。

(3) 個人別にPCが発生する擬似乱数にしたがって50試行を連続して行った。刺激の提示順と反応キーと反応時間（ミリ秒単位）が記録され、実験の結果データは自動的にエクセルファイルとして保存された。

(4) この1セッション分を上述べた形で6セッションが実施された。

教示：「これは複数の動く対象の追跡の実験です。実験の課題は、はじめに1から5個の間で指示された対象を含む10個の移動対象がとまったときに、指示された1個がはじめに指示されたものの1つかどうかを判断し、「イエス」か「ノー」のキーを押すことです。どちらかはっきりしない場合もどちらかに決めて必ずどちらかのキーを押してください。あまり考え込まず直感的にして下さい。この実験はあなたの能力や性格を調べようとするものではありませんが、集中してやって下さい。また、もし気分が悪くなったらすぐに中止して申し出て下さい。画面までの距離は約60cmにして、できるだけいつも同じところから見るようにして下さい。何か質問はありますか？この実験ではやっている途中では質問などで中断できませんので、分からないことがあったら何でもいいですから、今のうちに質問して下さい。（質問などがあればそれに答えてから）それでは練習を始めます。」

結果と考察

32名の全6セッションの反応時間（RT、ミリ秒）と正答数（CR）のデータの平均を、速度別に1回目と2回目のグラフで示したのが図2で（A）（B）（C）は低速・中間・高速条件の平均反応時間（ミリ秒）、（D）（E）（F）は平均正答数である。

まず、移動速度（3水準）×回数要因（2水準）×対象の数（5水準）の3要因被験者内比較デザインの分

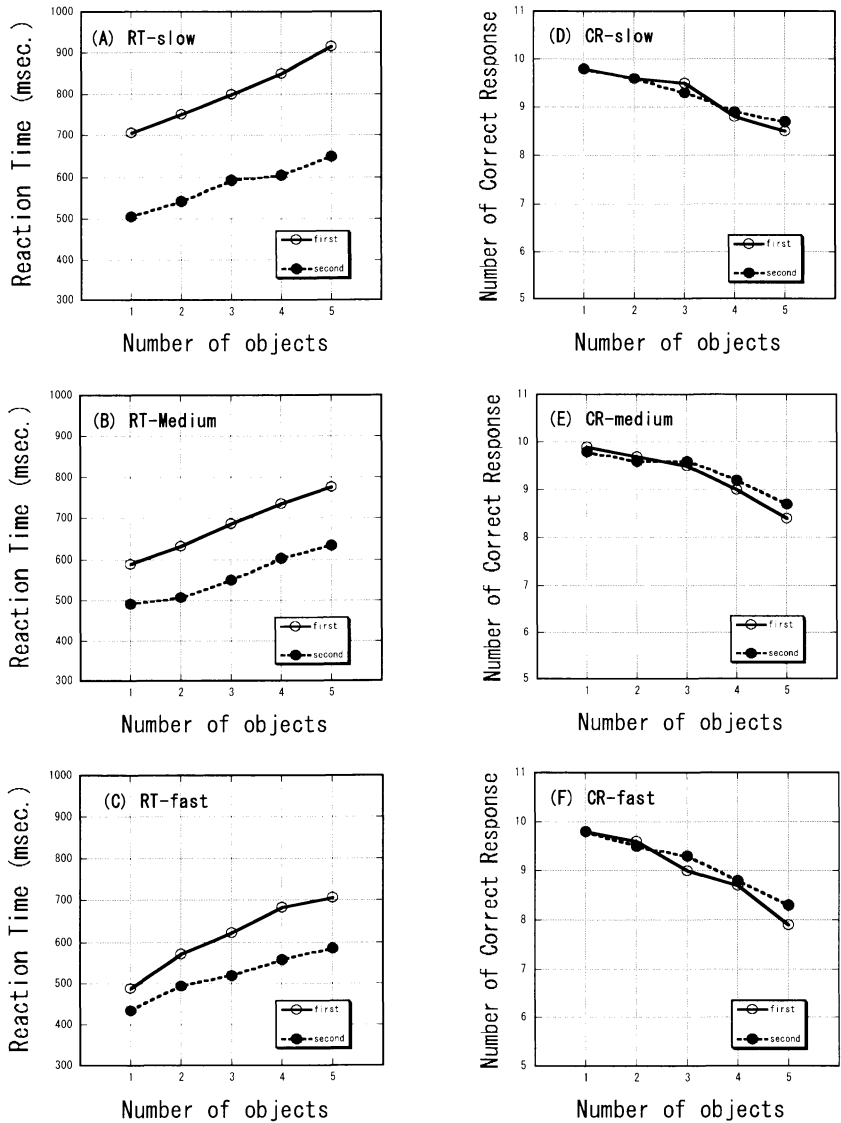


図2 32名の反応時間(RT, ミリ秒)と正答数(CR)の平均を速度別に1回目と2回に分けて示したグラフ。(A)(B)(C)は低速・中間・高速条件の平均反応時間(ミリ秒),(D)(E)(F)は平均正答数である。

散分析を反応時間の平均について行った。その結果、速度 ($F(2,62) = 58.852, p < 0.01$), 回数 ($F(1,31) = 62.511, p < 0.01$), 対象数 ($F(4,124) = 99.087, p < 0.01$) の3要因の主効果と、速度×回数 ($F(2,62) = 11.267, p < 0.01$), 回数×対象数 ($F(4,124) = 5.036, p < 0.01$) の一次交互作用が有意となった。さらに下位検定の結果、速度×回数の交互作用では1回目には3つの速度水準間にすべて有意差が見られたが、2回目には低速と高速条件のみに認められた(5%有意水準)。また、回数×対象数の交互作用では1回目と2回目の平均値の間にすべて有意差があり、それぞれの5つの対象数条件間のすべてに有意差が認められた(5%有意水準)。2次の交互作用は有意とはならなかった。つまり、平均反応時間の結果は、(1)どのセッションでもターゲットとなる対象数が多いほど

反応時間は増大し(対象個数効果)、(2)各速度条件内の2回目の平均反応時間は1回目のものより短くなり(学習効果)、(3)低速条件・中速度条件・高速条件の順で反応は早くなり(速度効果)、(4)低速移動条件の1回目の第1セッションと2回目の第6セッションの差(学習効果)は高速移動条件の場合よりも大きいことが示された(速度効果)。

次に、同じ3要因被験者内比較デザインの分散分析を正答数の平均について行った。その結果、速度 ($F(2,62) = 6.147, p < 0.01$) と対象数 ($F(4,124) = 45.239, p < 0.01$) の2要因の主効果のみが有意となった。この主効果の下位検定の結果では速度の主効果の低速と中間速度の間に差がなく、対象数の主効果では対象数1と2,2と3の間に有意差がないほかはすべて有意差が認められた(5%有意水準)。つまり、平均

正答数の結果は、(4) どのセッションでもターゲットとなる対象数が多いほど正答数は減少し(対象個数効果)、(5) 各速度条件内の1回目の平均正答数は2回目と差(学習効果)がなく、(6) 高速移動条件平均正答数は中・低速度移動条件の場合よりも少ないことが示された(速度効果)。

今回の実験の結果を要因別にまとめると、以下のようになる。

(1) 対象個数の効果：ターゲットとなる対象数が多いほど反応時間は増大し、正答数は減少したので実験仮説は2つとも支持された。

(2) 移動対象の速度の効果：低速度条件・中速度条件・高速条件の順で反応は早くなり、高速移動条件平均正答数は中・低速度移動条件の場合よりも少ない。ただし今回の実験では低速度条件・中速度条件・高速条件の順でセッションが組まれたので、実験順序効果と交絡している可能性がある。分離するためには今後の分析が必要である。

(3) 学習効果：各速度条件内の2回のセッションの成績の差としての学習効果については、平均反応時間では低速移動条件の1回目の第1セッションと2回目の第6セッションの差(学習効果)は高速移動条件の場合よりも大きいことが示されたが、平均正答数ではどの速度条件でも差が無かった。

おわりに

今回の認知ゲーム実験では多物体追跡(MOT)法という新しい実験パラダイム(齋木, 2007)による視覚的注意に関する実験を行った。ゲームの側から見ると、このように注意を分割し短期の視覚的記憶にもとづいて特定の反応をするという状況はさまざまなゲームではもっとも基本的なゲーム遂行の成分といえよう。もちろん、攻撃してくる敵の数が多ければゲームが難しくなるのは当たり前だが、その「難しくなる」というのが、今回の実験結果から反応時間と正答数(正しく敵を見分けること)の両方にかかわっていることや、敵の数が1人から2人になったからといって急に難しくなるのではなく、5人までは徐々に困難度

が増していくということがいえるような結果となった。このように、「認知ゲーム実験」(山上, 2006-2009)ではゲーマーがいわば「当たり前」としていることの内容が分析的に捉えられ、認知的活動の様相が明確になっていくことが特徴であろう。

今回の実験パラダイムを発展させていろいろな図形特徴(例えば色彩や形態)を実験変数とすることにより、視覚的注意の働きをさまざまな側面から分析できる(Klieger, Horowitz, & Wolfe, 2004)が、今回のデータはそのための基本的なデータともなりうる。

(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

- 菊池 正・八木善彦 2007「注意の基礎的理論」大山・今井・和気・菊池(編)新編 感覚・知覚ハンドブック 3章「注意と感覚・知覚」誠信書房 Pp.48-57.
- Klieger, S. B., Horowitz, T. S., & Wolfe, J. M. 2004 Is Multiple Object Tracking Color Blind? Vision Science Society
- 小谷津孝明 1981「注意」新版心理学事典 誠信書房 Pp.580-582.
- 渡辺めぐみ 1999「注意」心理学辞典 有斐閣 Pp.582-583.
- Pylyshyn, Z. W., & Storm, R. W. (1988) Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, 3, 179-197.
- 齋木 潤 2007「物体、属性に対する注意」大山・今井・和気・菊池(編)新編 感覚・知覚ハンドブック 3章「注意と感覚・知覚」誠信書房 Pp.64-72.
- 塩入 諭 2007「視覚的注意」塩入(編)視覚2 第8章 朝倉書店 Pp.183-224.
- 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)認知地図の整理効果 甲南女子大学研究紀要 人間科学編 45 11-19.

リスト1 HSP言語による視覚的注意分割実験プログラム

```

1 //-----
2 // (HSP3) MOT実験09-6.hsp ( Multiple Object Tracking Experiment)      MSPゴシック(11)
3 #packopt name "MOT実験09-6":// ctrl+F9->.exe      (yamagamai akira 09223-)
4 // 再認実験 + 再生実験
5 //-----
6 #include "imagebtn0.hsp": #pack "imagebtn0.hsp"://ボタンに画像を貼り付け+実行ファイル
7 #include "tmanage3.as" : #pac "tmanage3.as" :/// ミリ秒測定 gmes()+実行ファイル
8 buffer 1: picload "BRWG.bmp" :// バッファ 1 に画像
9 mmload "start2.wav" ,1.0:// mmload "リズム1.wav",1
10 //-----
11 *setGamen1:// 初期画面
12 //xx=ginfo_dispx:yy=ginfo_dispy: yy0=double(yy):screen 0,xx,yy,0.0://full size
13 xx0= 900: yy0= 660: screen 0,xx0,yy0,,120,20 :cls : randomize
14 objmode 2 :font "sytemfont",18 : objsize 220,30: color 000,000,000: // 使用画面枠
15 line 0,0,xx0,0:line xx0-1,0,xx0-1,yy0-1:line xx0-1,yy0-1,0,yy0-1:line 0,yy0-1,0,0
16 tt1=gettime(3)*60*60*24+gettime(4)*60*60+gettime(5)*60+gettime(6):// 開始秒測定
17 //-----
18 i=199: j=9: dim A,i,j: dim bbn,i,j: dim ccr,i,j: dim rrt,i,j: dim rt2,i,j
19 sdim pdd,i: sdim SSB,i: dim rs,i: dim nk,: dim px9,j: dim py9,j: : dim tn,i
20 dim bx,i: dim by,i: dim tg,i: dim nnn,i: dim RR,10 :dim SG,i:dim RR,i
21 //-----
22 //CD=0: ot=1: tm=3000: FB=1: et="e": goto *matrix1:// 再認のテスト
23 //RL=0:spd=6:goto *NNNN // 反応右手がno、左手がyesの場合( RL=1 はその逆 )
24 pos 30,30 : BUTTON " R/L0 ( 1= yes ) ( 3= no ) ",*rl0:
25 pos 30,80 : BUTTON " R/L1 ( 3= yes ) ( 1= no ) ",*rl1:stop
26 *rl0: RL=0:goto *rlrl
27 *rl1: RL=1:goto *rlrl
28 *rlrl:pos 30,230 : BUTTON " 条件1 " ,*cn1:
29 pos 30,280 : BUTTON " 条件2 " ,*cn2:
30 pos 30,330 : BUTTON " 条件3 " ,*cn3:stop
31 *cn1:spd=6
32 *cn2:spd=4
33 *cn3:spd=2: //spd=26 :// マシン別速度調整値
34 *NNNN: N999=50: // 1セッションの試行回数
35 clrobj 0,4
36 //-----
37 pos 30,60: name2="": mes "name": input name2: name2="S"+str(name2)://被験者番号入力
38 pos 30,120 : BUTTON " 再認実験 " ,*kkk1: // オブジェクト1:// CD=0
39 //pos 30,180 : BUTTON " 再生実験 " ,*kkk1: // オブジェクト2:// CD=1
40 pos 30,240 : BUTTON " テスト " ,*kkk2
41 pos 30,300 : BUTTON " 練習/デモ " ,*kkk3
42 ////////////////////////////////////////////////////////////////////tm=30: stop //:フィードバック設定・制限時間3秒
43 tm=3000: stop //:フィードバック設定・制限時間3秒
44 *kkk1 :CD=stat-1 :et="e": FB=1: ot=1: goto *matrix1: // 再認 CD=0 再生 CD=1
45 *kkk2 :CD=2 :et="t": FB=1: ot=1: goto *matrix1
46 *kkk3 :CD=9 :et="e": FB=1: ot=1: cls :goto *instruction
47 stop://-----
48 *instruction : color 220,220,160: boxf 00,00,xx0,yy0: color 00,00,00
49 pos 60,20: font "sytemfont",48:objsize 200,36: mes "< 動体追跡実験 >": mes
50 font "sytemfont",22: mes "実験の方法": mes
51 mes "これから画面上に10個の小さな四角形が画面上で動きます。"
52 mes "そのうち、はじめにマークされたものに注意してください。"
53 mes "すべてが止まった時に、最初にマークされたのがどれかをクリックします。"
54 mes "あるいは、最初にマークされた物かどうかを回答します。"
55 mes "まずはじめに練習しましょう。":pos 480,400: BUTTON " 練習 " ,*kkk7
56 stop: *kkk7: cls: goto *matrix1
57 stop://-----
58 *start0: font "sytemfont",18: objsize 160,24:mes "CD="+CD
59 pos 30,400: BUTTON "フィードバックあり" ,*kFB1
60 pos 30,440: BUTTON "フィードバックなし" ,*kFB0: stop
61 *kFB1: FB=1: goto *time0:
62 *kFB0: FB=0: goto *time0:
63 stop://-----
64 *time0:ot=0 //: 制限時間設定
65 pos 30,500: BUTTON "制限時間なし" ,*matrix1
66 pos 30,540: BUTTON "制限時間あり" ,*ari : stop
67 *ari :tm=3000: ot=1: goto *matrix1 :// 制限時間3秒
68 stop://-----
69 *matrix1:// 座標マトリクス作成と枠表示
70 //spd= 4:
71 //goto *skip1
72 if et="t": spd= 10: goto *skip1
73 repeat 5: pos 460,220: font "sytemfont",66:mes 5-cnt: wait 99:// カウントダウン
74 color 255,255,255: boxf 00,00,xx0,yy0: color 00,00,00: loop
75 *skip1:cls: md=0: : color 255,255,255: boxf 0,0,xx0,yy0// 全面消去
76
77 color 00,00,00:font "sytemfont",10j=0
78 repeat 5,i:i=cnt:repeat 10,i+j+1:NO(j)=i:mes str(j)+" "+str( NO(j)): loop: loop// 配列
79 repeat 5,i:i=cnt:repeat 2,0j=cnt:repeat 5,1,k:cnt:p=p+1:NA(p)=j:
80 pos 100,0+p*10:mes str(p)+" "+str( NA(p)):loop: loop: loop// 配列
81 //wait 100://stop

```

1. 各種準備
画面セット
変数配列

2. モード選択
反応設定
速度条件

3. 教示用意

4. フィードバック
制限時間設定

5. 座標マトリクス

```

82
83 x1=160: y1=10: h0=80: v0=80://外枠原点
84 x0=x1+h0 :y0=y1+v0:vn=5: hn=5: h1=h0: v1=v0: hh=h1+10: vv=v1+10:// 矩形サイズと間隔
85 font "sytemfont",12: pos xx0-60,0: if md=0: input nn,50,24:// nn 表示(オブジェクト0)
86 :repeat vn,1:i=cnt:repeat hn,1:j=cnt: nn=j+(hn)*(i-1) // ボタン番号 1 から
87 : objmode 2: color 000,000,000 :font "sytemfont",12 // ボタンの文字サイズ
88 : bx(nn)=x0+(j-1)*hh :by(nn)=y0+(i-1)*vv://objprm 0,nn // nn 変更表示
89 : ///objsize h,v :pos bx(nn),by(nn): button goto str(nn)*kkk222 // ボタン表示
90 : color 120,120,120: boxf bx(nn),by(nn),bx(nn)+h1,by(nn)+v1:// 四角表示
91 : color 255,255,255: boxf bx(nn)+1,by(nn)+1,bx(nn)+h1-1,by(nn)+v1-1:// 四角白抜き
92 : color 000,000,120: pos bx(nn),by(nn): if et="t": mes nn:// テスト用番号表示
93 :loop: loop: color 000,255,000
94 // 移動範囲枠(緑色)
95 line x1,y1,x1+(hn*hh)+2*h0,y1:line x1+(hn*hh)+2*h0,y1,x1+(hn*hh)+2*h0,y1+(vn*vv)+2*v0
96 line x1+(hn*hh)+2*h0,y1+(vn*vv)+2*v0,x1,y1+(vn*vv)+2*v0:line x1,y1+(vn*vv)+2*v0,x1,y1
97 if et="t":c1=80:c2=120:else:c1=40:c2=120://操作ボタンオブジェクト数一定のため消せない
98 if md=0: pos 0, 40: button "go",*plot1 // オブジェクト1
99 if md=0: pos 0, c1: button "matrix",*matrix1:// オブジェクト2
100 if md=0: pos 0, c2: button "file",*file1 // オブジェクト3
101 //-----
102 *rand0: pd="": // 1-50 が 1 個ずつの乱数列 RN(cnt): // 試行数分乱数
103 repeat 100,1:RN(cnt)=0:Q1(cnt)=0:loop: // 乱数列クリア
104 N1=50: N9=N999/N1: i=0: r=0: rx=30: z=12: color 000,120,000: // *-----GREEN
105 :repeat : p=rd(N1)+1 :Q1(p)=Q1(p)+1: if Q1(p) > N9 : continue
106 : i=i+1: r=r+1: RN(r)=p: pos rx,160+i*z: if et="t": mes str(r)+" ":
107 : if i<=30:pos rx+30,160+i*z: if et="t":mes p :
108 : if i>30 and i< 66: pos rx+ 90,160+(i-30)*z:if et="t": mes p :
109 : if i>=66 and i<101: pos rx+150,160+(i-65)*z:if et="t": mes p :
110 : pos 60,600:font "sytemfont",12 :pd=pd+str(p)+",",// 試行数分乱数の記録
111 : if r >= N1*N9: break:
112 :loop: cc=0: ii=0: color 000,120,000: font "sytemfont",12: if et="t": mes pd:
113 goto *plot1
114 stop
115 //-----
116 *rand1: pd="": // 1-5 が 20 個ずつの乱数列 JS(cnt): // 何個を動かすかの配列試行分
117 repeat 100,1:JS(cnt)=0:Q1(cnt)=0:loop: // 乱数列クリア
118 N1=5: N9=N999/N1: i=0: r=0: rx=30: z=12: color 000,000,000: // ****-----BLACK
119 :repeat : p=rd(N1)+1 :Q1(p)=Q1(p)+1: if Q1(p) > N9 : continue
120 : i=i+1: r=r+1: JS(r)=p: pos rx,160+i*z: if et="t": mes str(r)+" ":
121 : if i<=30:pos rx+30,160+i*z: if et="t":mes p :
122 : if i>30 and i< 66: pos rx+ 90,160+(i-30)*z:if et="t": mes p :
123 : if i>=66 and i<101: pos rx+150,160+(i-65)*z:if et="t": mes p :
124 : pos 60,600:font "sytemfont",12 :pd=pd+str(p)+",",// 移動個数配列の記録
125 : if r >= N1*N9: break:
126 :loop: cc=0: ii=0: color 000,000,000: font "sytemfont",12: if et="t": mes pd:
127 //-----
128 *rand2: pd2="": // 1 or 0 が 50 個ずつの乱数列 RON(cnt): // 再認指示の配列試行分
129 repeat 100,1:RON(cnt)=0:Q1(cnt)=0:loop: // 乱数列クリア
130 N1=2: N9=N999/(N1*5): i=0: r=0: rx=30: z=12: color 000,000,255: // *****
131 repeat 5,0: j=cnt
132 :repeat: k=cnt: p=rd(N1) :Q1(p)=Q1(p)+1: if Q1(p) > N9 : continue:// N9=5
133 : //i=i+1: r=r+1: RON(r)=p: pos rx+200,160+i*z: if et="t": mes str(r)+" ":
134 : i=i+1: r=r+1: RON(j*10+r)=p: pos rx+200,160+i*z: if et="t": mes str(r)+" ":
135 : //if i<=30:pos rx+230,160+i*z: if et="t":mes p :
136 : //if i>30 and i< 66: pos rx+ 290,160+(i-30)*z:if et="t": mes p :
137 : //if i>=66 and i<101: pos rx+350,160+(i-65)*z:if et="t": mes p :
138 : pos 90,520:font "sytemfont",12 :pd2=pd2+str(p)+",",// 再認指示配列
139 : if r >= N1*N9: break:
140 :loop: cc=0: ii=0: //color 000,000,255: font "sytemfont",12: if et="t": mes pd2:
141 pd2=pd2+" ":i=0: r=0: repeat 10,0:Q1(cnt)=0:loop:loop
142 color 000,000,255: font "sytemfont",12: if et="t": mes pd2://-----BLUE
143 pos 200,0:color 255,000,000
144 repeat 5,0: j=cnt:repeat 10,1: i=cnt:pos 300+50*j,i*10:mes RON(j*10+i):loop:loop://----RED
145 STOP
146 //-----
147 *rand33: pd3="": // 1 or 0 が 50 個ずつの乱数列 RON(cnt): // 再認指示の配列100試行分
148 //repeat 100,1:RON(cnt)=0:Q1(cnt)=0:loop: // 乱数列クリア
149 //N1=2: N9=N999/(N1*5): i=0: r=0: rx=30: z=12: color 000,000,255:
150 repeat 50,1: j=cnt
151 //repeat: k=cnt: p=rd(N1) :Q1(p)=Q1(p)+1: if Q1(p) > N9 : continue:// N9=5
152 : //i=i+1: r=r+1: RON(r)=p: pos rx+200,160+i*z: if et="t": mes str(r)+" ":
153 : i=i+1: r=r+1: RON(j*10+r)=p: pos rx+200,160+i*z: if et="t": mes str(r)+" ":
154 : //if i<=30:pos rx+230,160+i*z: if et="t":mes p :
155 : //if i>30 and i< 66: pos rx+ 290,160+(i-30)*z:if et="t": mes p :
156 : //if i>=66 and i<101: pos rx+350,160+(i-65)*z:if et="t": mes p :
157 : pos 90,520:font "sytemfont",12 :pd2=pd2+str(p)+",",// 再認指示配列
158 : if r >= N1*N9: break:
159 :loop: cc=0: ii=0: //color 000,000,255: font "sytemfont",12: if et="t": mes pd2:
160 pd2=pd2+" ":i=0: r=0: repeat 10,0:Q1(cnt)=0:loop://loop
161 color 000,000,255: font "sytemfont",12: if et="t": mes pd3:
162 repeat 5,0: j=cnt:repeat 10,1: i=cnt:pos 300+50*j,i*10:mes RON(j*10+i):loop:loop:
163 //-----

```

6. 移動対象表示

7. 試行順序の乱数

8. 対象個数の乱数

9. ターゲット有無の乱数

10. テスト用の乱数


```

243 if CD=1 :stop :// 再認(CD=0)対象指示の色づけと反応キー入力//iparam-48
244 RON(ii)=NA(RN(ii)): JS(ii)=NO(RN(ii))
245 if RON(ii)=1:bnn=rnd(JS(ii))+1:else:bnn=rnd(10-(JS(ii)))+JS(ii)+1://1-10指示ボタン指定(bnn)
246 n=bnn: dx=24: dy=dx: wz=3: color 0,255,0:pos 0,0:font "sytemfont",12: mes t1-t7
247 color 255,000,000: boxf px9(n)-wz-1,py9(n)-wz-1,px9(n)+dx+wz,py9(n)+dy+wz: // 正方形
248 // キー入力待ちと制限時間チェック
249 *key0: ot9=0:key=0:getkey key,49: if key =1: key=1: goto *key1:// キーの1
250 : getkey key,51: if key =1: key=3: goto *key1:// キーの3
251 getkey key,57: if key =1: key=9: pos 100,0: font "sytemfont",66: mes "STOP":stop:// キーの9
252 : getkey key,97: if key =1: key=1: goto *key1:// テンキーの1
253 : getkey key,99: if key =1: key=3: goto *key1:// テンキーの3
254 : t2= gmsec() : if t2-t1>tm: ot9=9: r9=99: goto *record1: //制限時間
255 : wait 2:goto *key0: // onkey *key1: onclick *key1: :// 反応待ち
256 *key1: r9=key: goto *record1:// 反応キー記録
257 stop://-----
258 *record1:pos 0,0: font "sytemfont",12 // 結果表示・記録 //オーバータイムなら反応は99
259 rs(cc)=r9: // RL=0では反応右手が3(NO)、左手が1(YES)( RL=1 はその逆 )
260 if RL=0 and r9=3: rs(cc)=0:
261 if RL=0 and r9=1: rs(cc)=1:
262 if RL=1 and r9=1 :rs(cc)=0:
263 if RL=1 and r9=3 :rs(cc)=1:
264 if ot9=9: rs(cc)=99: //オーバータイムなら反応は99
265 if et="t":mes "RL="+str(RL)+" "+r9="+str(r9)+" "+str(rs(cc)): // キー変換チェック
266 if CD=1: bnn=stat-3:// 再生の場合のクリック反応ボタン番号
267 t3= gmsec() :// if CD=0 and et="e":end:
268 ccc=cc: objprm 0,ii // 試行数カウンター(ii)
269 tcc(ii)=tcc(ii)+1: jj=tcc(ii) // 試行内反応数カウンター(jj)
270 if CD=0: tcc(ii)=1: jj=1 // 再認実験では1試行1反応
271 tn(ii)=rd: // ターゲット数
272 A(ii,jj)=tg(nk(bnn)) // ターゲットマーク 正答番号
273 bbn(ii,jj)=nk(bnn) // 反応ボタン番号
274 nnn(ii)=bnn
275 // 再認(CD=0) の場合は反応一致か、再生(CD=1) の場合はターゲットマークが1か
276 if CD=1 : rs(ii)=1
277 if A(ii,jj)=rs(ii): k=k+1: ccr(ii,jj)=1: else: ccr(ii,jj)=0: // 正誤(1/0)
278 if ot9=9: ccr(ii,jj)=0: //オーバータイムなら判定は0
279 rrt(ii,jj)=t3-t1: rt2(ii,jj)=rrt(ii,jj) // 試行開始からのRT(ミリ秒)
280 if jj=1: rt2(ii,jj)=rrt(ii,jj)-rrt(ii,jj-1): // 前の反応からのRT2(ミリ秒)
281 pos 20,160: font "sytemfont",36: mes str(k)+" "+str(rd): // ターゲット個数表示
282 color 255,255,255: boxf 20,160, 120,300:color 255,000,000
283 pos 20,160:font "sytemfont",66 :
284 if FB=1: if CD=1: mes str(k)+" "+str(rd): else: mes str(k)//スコア表示
285 if jj>rd: if CD=1: mes "finished":// クリック数終了通知
286 // 試行数+試行内クリック数+ターゲットマーク+反応ボタン番号+正誤+RT+RT2
287 // 試行+刺激番号+ターゲットマーク+反応ボタン番号+正誤+RT+RT2
288 color 255,255,255: boxf 0,0, 400,30:font "sytemfont",20: color 000,000,000
289 if et="e" and CD=0: pos 010,0: mes ii: pos 040,0: mes r9
290 //if et="t": pos 010,0: mes ii: pos 040,0: mes jj : pos 60,0: mes "no="+tn(ii)
291 if et="t": pos 010,0: mes ii: pos 040,0: mes RN(ii) : pos 60,0: mes "no="+tn(ii)
292 if et="t": pos 120,0: mes "tg="+A(ii,jj)+ " bn="+str(nnn(ii))+ " "+str(bbn(ii,jj)):
293 if et="t": pos 280,0: mes "rs="+rs(ii) : pos 340,0: mes "cr="+ ccr(ii,jj)
294 if et="t" and ccr(ii,jj)=1: color 0,0,255: pos 400,0: mes "YES": color 0,0,0
295 if et="t" and ccr(ii,jj)=0: color 255,0,0: pos 400,0: mes "NO!": color 0,0,0
296 if et="t": pos 480,0: mes rrt(ii,jj): if CD=1: pos 560,0: mes rt2(ii,jj)// 再生時
297 if et="t": pos 540,0: mes str(bnn)+" "+str(nk(bnn))+ " "+str(tg(nk(bnn)))+ " "+str(RN(ii))
298 if jj>rd: pos 520,0: if CD=1: mes "finished": //
299 if cc>=N999 :mes "END":goto *file1:stop
300 if CD=0 or CD=9: wait 99: goto *plot1:// 自動継続モード
301 stop://-----
302 *file1:font "sytemfont",12: color 255,000,000 // データ記録をCSV形式で保存する
303 name=name2 + "-"6 CD="+str(CD)+" RL="+str(RL)+" spd="+str(spd): //赤で表示
304 n=0: b=10: bb=18: ppp = "": ppdd="": sss="": kkk=0: CR="¥r": // CRLF="¥n¥r"
305 tt7=gettime(3)*60*60*24+gettime(4)*60*60+gettime(5)*60+gettime(6):// 終了秒測定
306 tt8= str(tt7-tt1): // 全所要時間秒測定 // tt9 は記録ファイルのヘーダー
307 tt9= name + CR +str(gettime(0))+ " "+str(gettime(1))+ " "+str(gettime(3))
308 tt9= tt9 + " "+str(gettime(4))+ " "+str(gettime(5))+ " "+str(gettime(6))
309 tt9= tt9 + " (" +tt8+ " sec.": //+CR +ターゲット個数配列=" + pd
310 tt9=tt9+CR+"pd2="+pd2
311 iii="i"+" "+NO"+" "+mv"+" "+A"+" "+nn"+" "+bn"+" "+rs"+" "+cr"+" "+RT"
312 sss = tt9 + CR +CR+ iii+ CR: // sss は全試行分の記録ストリング
313 :repeat ccc,1 :i=cnt ://1 試行分のデータを累積
314 : :repeat tcc(i),1: j=cnt :n=n+1:
315 : : ppp=ppp+str(i)+" "+str(RN(ii))+" "+str(tn(ii))+" "+str(A(ii,jj))+" "+str(nnn(ii))+" "+str(bbn(ii,jj)):
316 : : ppp=ppp+" "+str(rs(ii))+" "+str(ccr(ii,jj))+" "+str(rrt(ii,jj))+" "+str(rt2(ii,jj))+ CR
317 : : sss= sss + ppp // ppp は1クリック分の記録ストリング
318 : : pos 150,10+n*12: ppp=ppp: mes ppp :ppp = "": // ppp をクリア
319 : :loop: ppdd=ppdd+str(i)+ " "+ str(JS(ii))+",8888,"+SSB(i)+CR://1 試行分のランダム数列
320 : :loop:sss=sss+CR+ppdd://1 試行分のランダム数列の累積記録を付加
321 mes "-----" +CR : mes sss : notesel sss :if CD=9: stop: // 全記録ストリング
322 if et="e":
323 notesave name+".csv":// 実験時のみ記録ファイル作成
324 if et="e": pos 500,500:sysfont: mes "おわかりました。おつかれさま。":
325 stop://-----

```

14. 反応
入力待ち
(ループ)

15. 反応入力
処理
データ記録
反応画面表示

16. データ
ファイル保存