

認知ゲーム実験 (8) 心的回転

山 上 暁

Cognitive Game Experiment (8): Mental Rotation

YAMAGAMI Akira

Abstract : As the eighth report on the cognitive game experiments, I developed a mental rotation experiment. The experiment situations were produced on a Windows XP PC with a program of the HSP language. The participant of the experiment judged whether the two perspective projection drawings of the block objects were same or different (mirror image) on the PC display. Three experimental factors were ; three kinds of the rotation axis (X, Y, Z), two kinds of object combination (same/different) and ten levels of the presentation angle difference (between 0–180° with 20° step) of the two block objects. Since the rotation axis was a between-subject variable, the numbers of participant were 23 under the X axis condition, 21 under the Y axis condition and 20 under the Z axis condition. Five repetition made $2 \times 10 \times 5$ trials for each participant. The mean reaction time (RT) data showed, (1) Main effects of three experimental variables were all significant on the RT data. (2) The ‘same’ trial RT means were the same among three axis conditions. The RT means of the ‘same’ trials were shorter than those of the ‘different’ trials under the X- and the Z-axis condition. (4) The RT means of the ‘same’ and the ‘different’ trials were increased almost linearly under all three rotations. The slopes of the liner regression equations of the ‘different’ trials were all smaller than those of the ‘same’ trials.

Key Words : cognitive game experiment, mental rotation, rotation axis effect

目 的

心的回転 (mental rotation) の実験は Shepard & Metzler (1971) によって行なわれた心的イメージについての実験である。われわれ自身の主観的観察からすれば、図1のような立体ブロックが2つ並んだ画像を見て、同じか違う (鏡映像である) かを問われた時には「頭の中でブロックを回転させて判断している」という感じがする。もちろんこのような内省の分析は重要だが、多義的であることも多い。彼らは実験で反応時間が2対象の呈示角度の差の一次関数となることを示した。平面回転でも、奥行き回転でもその性質は変わらず、あたかも3次元的世界の立体ブロックを等速に回転させているのと同じような結果が得られた。われわれの頭の中に外界対象の3次元アナログが存在

するかのようだ、というのが、彼らの主張であり認知心理学の一つの大きな成果とされるような実験になったのである。その後、さまざまな「心的操作」(mental operations) でそのような外界対象の3次元アナログの存在が主張された (Cooper & Shepard, 1973; 増井, 1983)。さらに認知心理学における「イメージ論争」として理論的展開もあった (邑本, 2005; 宮崎, 1998)。心的回転の実験が40年を経たこんにちにおいても、認知心理学実験実習のスタンダードテーマの一つとしていろいろなテキストで取り上げられている (浅野, 2008; 岩永, 1993; 桐田, 1999; 渡辺, 1985) のも、単にクイズのような課題としての面白さだけでなく、そうした認知心理学での理論的・歴史的意味を背景に持っているからであろう。

今回は「認知ゲーム実験」シリーズ (山上, 2006–2010) のひとつとしてこの「心的回転」の実験を採用

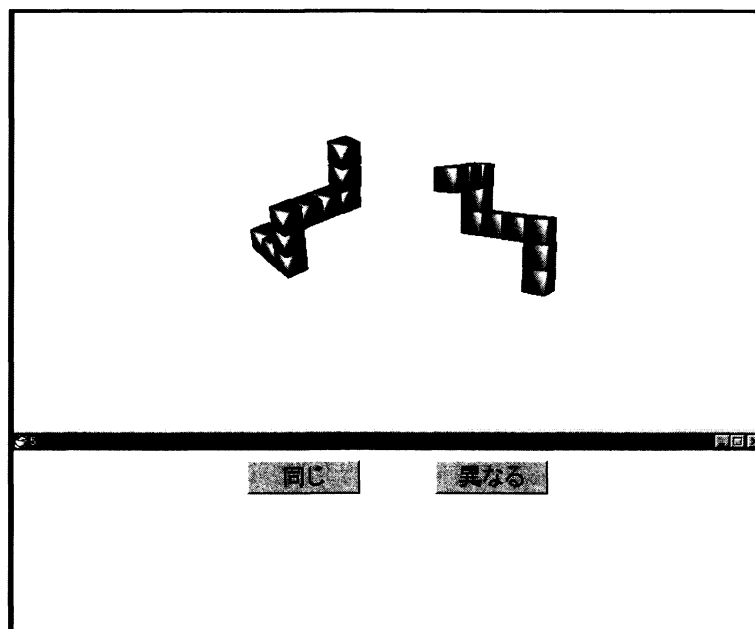


図 1 心的回転 (mental rotation) の実験の刺激図形の例

した。パーソナルコンピュータの能力が向上し、3Dゲーム用のソフトが容易に利用できるようになったおかげで、リアルタイムの3次元ブロックの2次元投映画像の呈示が実現した。単に Shepard & Metzler (1971) の実験の追試というだけでなく、回転を呈示空間での X 軸・Y 軸・Z 軸という 3 軸の条件を設定し、彼らが示していない「different」反応のデータも比較して、心的回転が行なわれる心的空間の特性も調べてみたい。

方 法

パーソナルコンピュータ (PC) 画面上に Shepard & Metzler (1971) が用いたような立方体を 10 個つなぎ合わせたブロック対象の 2 次元投影図を左右に 2 個呈示し、2 つの対象の角度差の関数としての反応時間を測定した (図 1 参照)。2 つの対象の角度差は彼らと同じく 0 度から 180 度までの 20 度ごとの 10 水準を用意した。実験参加者は心理学学科学学生 116 名で、各自が 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。各実験参加者はグループごとに回転軸の種類が決められた。1 回の実験では約 20 名が一斉に実験を行なった。Windows XP の PC のディスプレイ画面が常に実験参加者の正面にあるようにし、約 60 cm の観察距離をとった。甲南女子大学 5 号館の PC 教室で行ない、一人当たりの 1 セッションの所要時間は約 15 から 25 分であった。

刺激：PC 上で 3 次元対象を作成するプログラムであ

る 3 D Ace (大槻, 2005) で作成した図 1 のような立体ブロック対象をもとに、HSP 言語で作成した実験プログラムによって設定した呈示角度で 2 次元投影図としての刺激画像を試行ごとに作成させた。「同 (same) 条件」では 2 つの対象は同じもので左右の 2 対象の角度差が 0 度から 180 度まで 20 度ステップの 10 水準のうちの一つとなるようにそれぞれの対象の呈示角度が設定された。「異 (different) 条件」ではもとのブロックの鏡映としてのブロックの 2 次元投映図形が用いられた。3 次元対象回転の回転軸は画面に平行な X 軸と画面上でそれと直交する Y 軸および奥行き方向の Z 軸の 3 軸が用意された。

手続：実験参加者の課題は画面上の 2 つのブロック対象が同じ形である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試行ごとにランダムにした。刺激呈示から反応キーを押すまでの反応時間 (ミリ秒) と正誤が PC 内に自動的に記録された。

実験計画：実験デザインは 3 要因 (3 水準 \times 2 水準 \times 10 水準) 計画で、第一の要因は被験者間比較要因である回転軸の種類 (X 軸・Y 軸・Z 軸) であり、第二の要因は被験者内比較要因としての「同条件」と「異条件」であり、第三の要因は被験者内比較要因としての呈示角度差 (0 度から 180 度までの 10 水準) であった。回転軸の種類は実験参加者グループごとに決められたので、一人の実験参加者は 2 条件 \times 10 角度差を

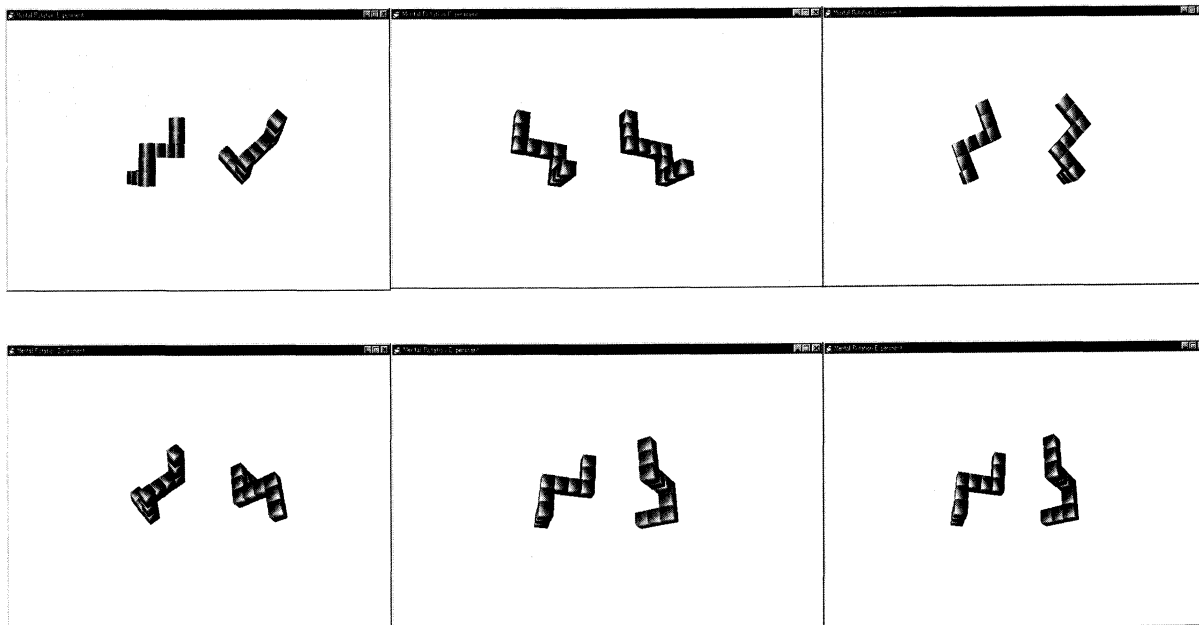


図2 3つの回転軸条件での刺激例。左から X 軸回転・Y 軸回転・Z 軸回転で、上の段が「同条件」で、下の段が「異条件」の刺激図形例である。

それぞれ 5 回ずつ測定するため合計 100 試行がランダム順で行われた。始めの 60 試行が終了すると「休憩」の時間となり、各自が適当な休憩（1-2 分）の後に残りの 40 試行が実施された。6 つの実験グループで各軸条件に 2 グループを割り当てて各条件で約 40 名の実験を行なった。

結果と考察

6 つの実験グループで各回転軸条件に 2 グループを割り当てて各条件で約 40 名の実験を行なった。100 問中の正答数が 91 以上の者のデータを集計し分析にした。各自の正反応のうち反応時間（ミリ秒）の長い

ものを 5 つ削除した残りのデータのうち、正答のデータのみを各人の角度差条件で最大 5 回の繰り返しについて、異同条件と角度差条件の 20 個の平均と SD を算出し、各人の個人データを整理した。基準をクリアした実験参加者の数は X 軸回転条件で 40 人中 21 名、Y 軸回転条件で 39 人中 20 名、Z 軸回転条件で 37 人中 23 名であった。

このデータから各回転軸条件別に「同条件」と「異条件」に分けて、角度差の関数としての平均反応時間をプロットしたのが図 3 である。この平均反応時間データに 3 要因（被験者内比較 1 要因と被験者間比較 2 要因の混合計画）の分散分析を行なった。第一の回転軸の種類の変異要因 ($F(2,61) = 5.06, p$

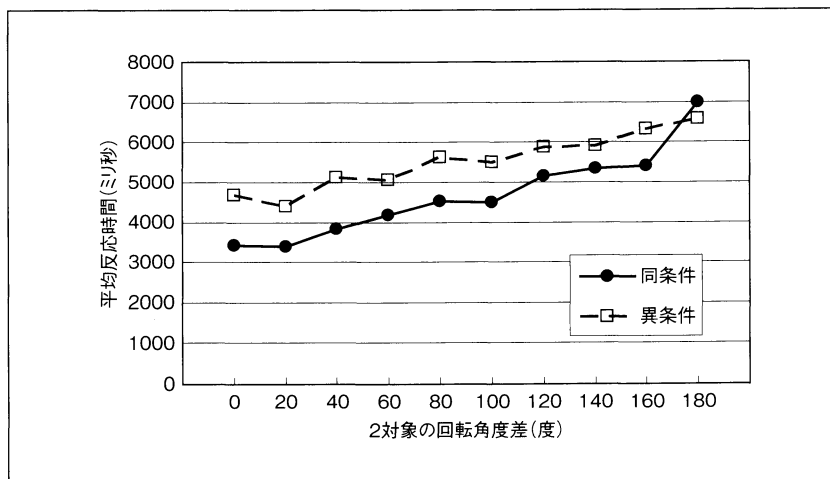


図 3-1 X 軸回転での 2 対象の回転角度差（度）の関数としての平均反応時間（ミリ秒）

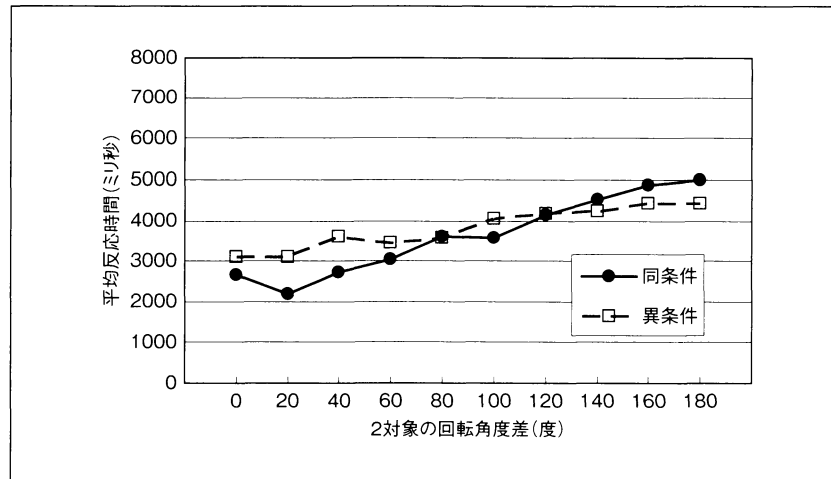


図3-2 Y軸回転での2対象の回転角度差(度)の関数としての平均反応時間(ミリ秒)

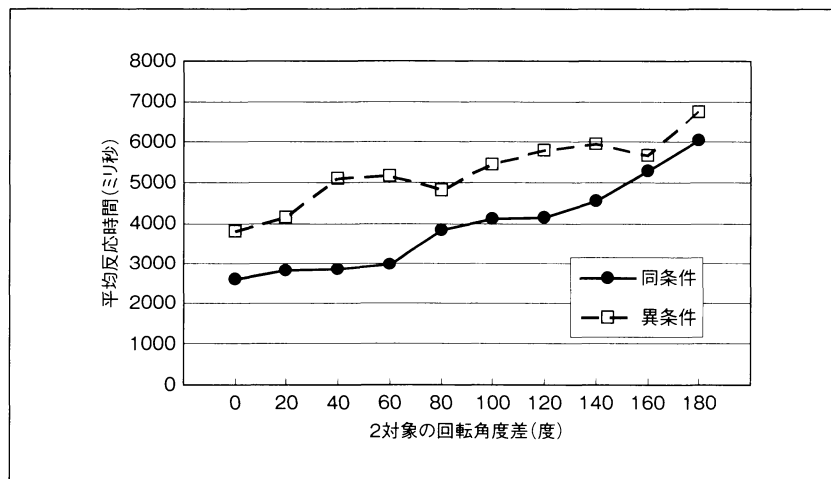


図3-3 Z軸回転での2対象の回転角度差(度)の関数としての平均反応時間(ミリ秒)

表1 「同条件」と「異条件」での各軸回転の平均反応時間データの回転角度差(度)に対する直線回帰の勾配と切片の値と決定係数

同条件	勾配	切片	決定係数	異条件	勾配	切片	決定係数
X軸回転	17.2	3118.7	0.90	X軸回転	11.2	4502.4	0.93
Y軸回転	15.9	2201.3	0.95	Y軸回転	8.2	3066.4	0.94
Z軸回転	18.4	2261.9	0.94	Z軸回転	13.3	4052.5	0.86

<0.01), 第二の要因の被験者間比較要因の「同条件」と「異条件」($F(1,549) = 18.8, p < 0.01$), 第三の要因の被験者間比較要因である呈示角度差 ($F(9,5492) = 51.1, p < 0.01$) の三つの主効果は有意となった。回転軸の種類と異同条件の交互作用 ($F(2,61) = 3.42, p < 0.05$) と異同条件と角度差の交互作用 ($F(9,549) = 4.12, p < 0.01$) が有意となり, 回転軸の種類と角度差の交互作用 ($F(18,549) < 1, n.s.$) と3要因の2次の交互作用 ($F(18,549) < 1, n.s.$) は有意とはならなかった。有意となった一つ目の回転軸の種類と異同条件の交互作用をグラフにすると図4のようになった。下位検定(5%有意水準)の結果, 「同条件」では3つ

の回転軸の種類で平均反応時間に有意差が見られず, 「異条件」ではX軸とZ軸では差がなく, Y回転条件での平均反応時間が他の2軸よりも有意に短いことが示された。またX軸とZ軸では「同条件」の平均反応時間が「異条件」よりも有意に短いことが示された。つまり, 角度差条件を込みにしたY回転条件では「同条件」と「異条件」の差がなく, 他の2軸の場合より早く反応が出来, X軸とZ軸では「異条件」よりも「同条件」の反応が早かった。もう一つの有意な交互作用である異同条件と角度差の交互作用(図5)についての下位検定(5%有意水準)の結果, 単純主効果が有意となった「同条件」では, 10水準を0

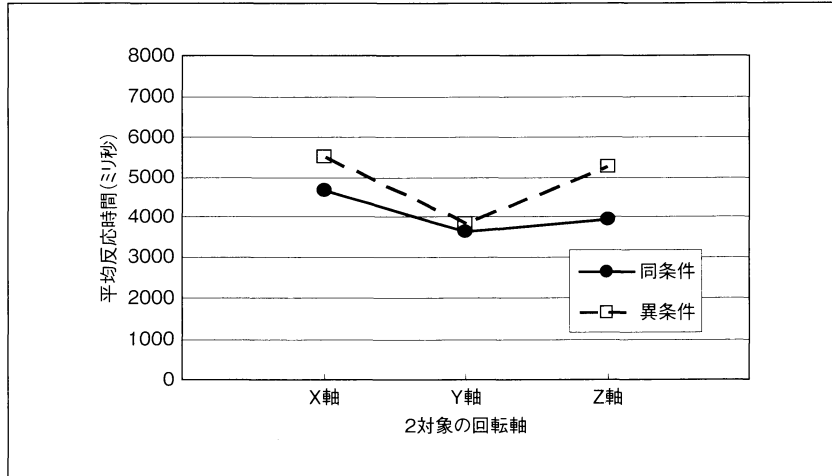


図4 角度差を込みにした回転軸の関数としての平均反応時間(ミリ秒)

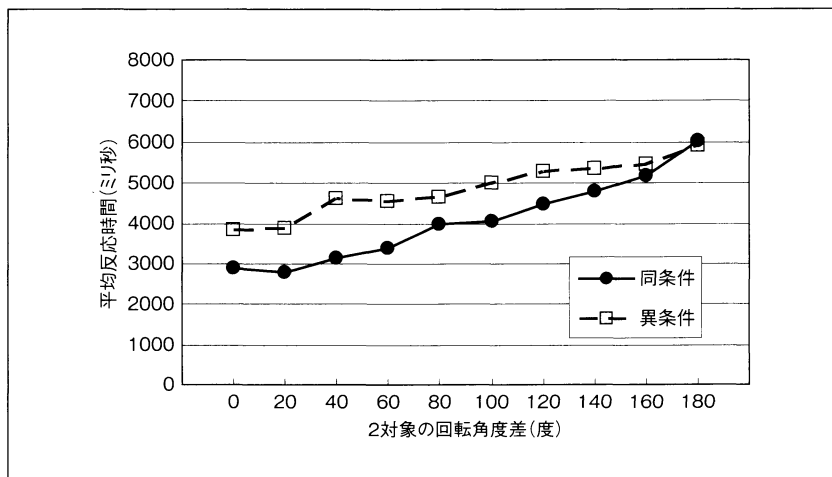


図5-1 3軸を込みにした2対象の回転角度差(度)の関数としての平均反応時間(ミリ秒)

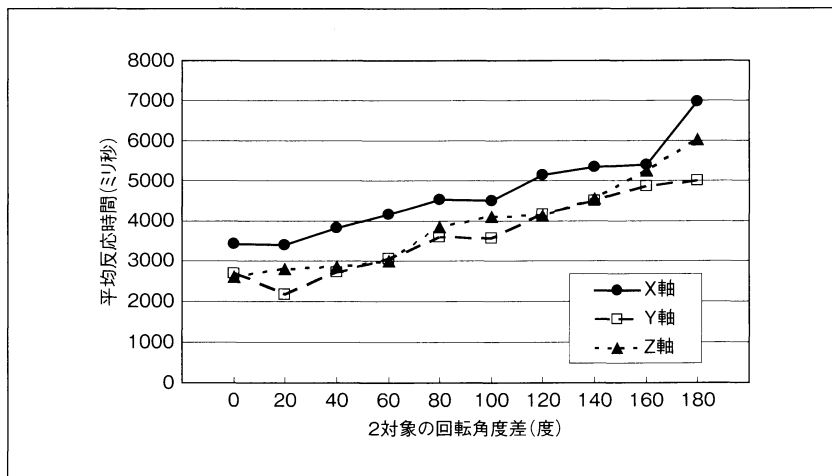


図5-2 3軸別の2対象の回転角度差(度)の関数としての「同条件」平均反応時間(ミリ秒)

度の1から180度の10で表記すると(1-2-3-4)と(5-6-7)と(8-9)と10の4つのグループ間で有意差が見られ、同じく単純主効果が有意となった「異条件」では、(1-2)と(3-4-5-6-7)と(8-9-10)の3

つのグループ間で有意差が見られた。また「同条件」は全体として「異条件」より1秒ほど早く反応され、(9)と(10)以外では有意差が見られた。

心的回転の反応時間の分析は「同条件」について行

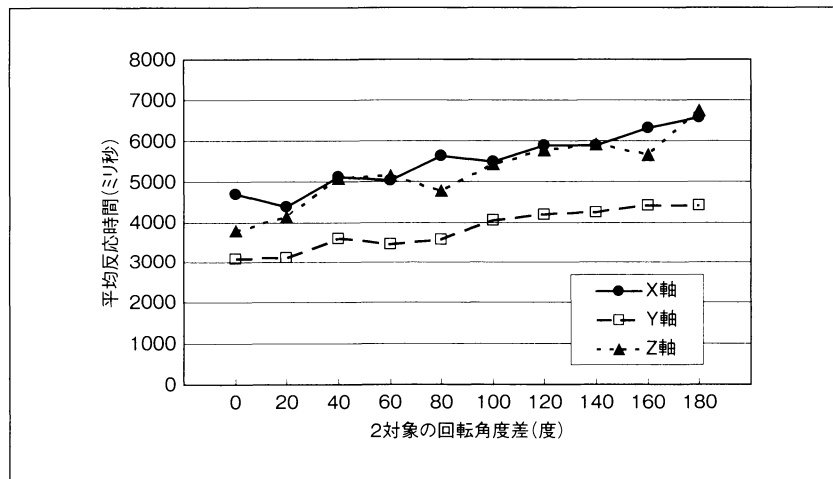


図 5-3 3軸別の 2 対象の回転角度差 (度) の関数としての「異条件」平均反応時間 (ミリ秒)

なわれてきているので、今回の実験では 3 軸の回転の反応時間の間には差が認められないということになった。心的回転実験での回転軸の効果を検討した先行研究ではさまざまな結果が出ており (大倉, 1993), さまざまな条件に左右されるようだ。

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) 「回転軸の種類」と「異同条件」および「2 対象呈示角度差」の 3 つの要因の主効果は有意であった。
- (2) 回転軸要因の効果を見ると (図 4), 「異条件」では Y 軸回転は他の 2 軸よりも反応されたが, 「同条件」では 3 軸で差はなかった。Y 軸回転では「異条件」と「同条件」ではほぼ同じ反応時間で, X 軸回転と Z 軸回転ではどちらでも「同条件」の方が「異条件」よりも早く反応がされた。
- (3) 2 対象の呈示角度差が大きくなるほど「異条件」でも「同条件」でも反応時間はほぼ直線的に増大したが, 「異条件」の勾配の方が緩やかな形になった (図 5)。

おわりに

今回の認知ゲーム実験では認知心理学実験の古典的ともいえる「心的回転実験」を行なった。ゲームとしてやってみると頭の中で回転させるのが得意な人と不得意な人がいるのがよくわかる。1 試行ずつじっくり考えてする人とぱっとすばやく判断する人はどんなゲームや認知実験でも分かれるが, この心的回転課題ではよりはっきりと分かれていたようだ。この個人差の特性を課題やそれに対応すると考えられる認知的要素ごとに分析してみるのも面白いだろうなという気にな

る。課題や条件をシステムティックに変えてゲームをしてもらおうと様々な認知的要素で個人差のプロフィールが描けそうである。これまでの「認知ゲーム実験シリーズ」(山上, 2006-2010) を有機的に組み合わせると個人差を検討してみるのも面白そうだ。

(注) 本研究は甲南女子大学より平成 21 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

- 浅野昭祐 2008 心的回転 (メンタルローテーション) 兵頭・須藤 (編著) 認知心理学基礎実験入門 八千代出版 pp.119-128.
- Cooper, L. & Shepard, R. 1973 Chronometric studies of the rotation of mental images. In W. Chase (Ed.), *Visual Information Processing*. New York, NY: Academic Press. pp.135-142.
- 岩永 誠 1993 心的回転 利島・生和 (編著) 心理学のための実験マニュアル 北大路書房 pp.166-169.
- 桐田隆博 1999 反応時間計測法 海保・加藤 (編著) 認知研究の技法 福村出版 pp.73-78.
- 増井 透 1983 メンタル・イメージ 自然 1983 年 9 月号 26-35.
- 邑本俊亮 2005 イメージ論争 森・中條 (編) 認知心理学キーワード 有斐閣 pp.98-99.
- 宮崎清孝 1998 イメージは創造をささえるか - イメージ論争再び - 丸山 (編著) 心理学のなかの論争 ナカニシヤ出版 pp.121-148.
- 大倉正暉 1993 空間の 3 次元軸における心的回転 人間科学年報 (甲南女子大学人間科学研究会) 第 18 号 17-32.
- 大槻有一郎 2005 12 歳からはじめる HSP 3 D ゲームプログラミング教室 ラトルズ
- 笹岡貴史 2010 メンタルローテーションとその仕組み

- 乾・吉川・川口(編) よくわかる認知科学 ミネルヴァ書房 pp.60-61.
- Shepard, R. N. & Metzler, J. 1971 Mental rotation of three-dimensional objects. *Science*, 171, 701-703.
- 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.
- 横澤一彦 1995 心的イメージの操作 行場次郎(編) 認知心理学重要研究集 1 誠信書房 pp.172-175.
- 渡辺武郎 1985 心的回転(第14章) 中谷(監) パーソナルコンピュータによる心理学実験入門 ブレーン出版 pp.159-173.

リスト1 HSP言語による心的回転実験プログラム

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78	<pre> //----- // (HSP3) メンタルローテーション実験10-5(回転軸選択式) (yamagami akira 1030 MSPゴシック(11) #packopt name "Rotation10-5.hsp":// ctrl+F9->.exe //----- #include "hgimgx.as" :#pack"hgimgx.as" :// hgimgx #include "tmanage3.as" :#pack"tmanage3.as" :// ミリ秒測定gmes() // 注意:フォルダ内に"joyp.as"+ "IniEdit.as"+ "m.joystick.as" の3つが必要 #include "joyp.as" :#pack"joyp.as" :// joyp.asモジュールを使用 psb_joysset "padset.ini" :設定の読み込みと反映 padid = stat :設定を反映させたゲームパッドのID //----- *begin: xx0= 800: yy0= 500: screen 0,xx0,yy0,200,00: hgini: // 対象提示画面 title "Mental Rotation Experiment":// タイトル設定 cls: objsize 200,36: objmode 2: font "sytemfont",18 : screen 1,xx0,200,200,yy0+00: cls: randomize :// コントロール・表示画面 //----- *dimension : zz=999:// 配列変数の宣言 dim yn,zz : dim rt,zz : sdim pict,zz : dim ans,zz : dim rrr,zz : dim rt9,zz dim bn,zz : dim bnn,zz : sdim sss,zz : sdim p,zz : sdim pp,zz : sdim rr,zz dim Q,zz : dim A,zz : dim bbn,zz : dim rrt,zz : sdim stt,zz : dim bbt,zz dim bbt,zz : dim grr1,zz : dim grr2,zz : sdim xyz,zz: pic="": ii=0: sss1="": sss2="": sss3="": sss4="": name2="": // 初期値 tttt=120: ii=0: FB=1: tm=9999: sss5="": repeat 24,1 : p(cnt)="": loop:// tttt=120 exp //N1=100: et ="t":FB=1:tm=5000:rb="r":goto *rand1:// ----- test //----- gsel 0: objsize 200,36: // 対象提示画面 bx0=20:by0=60:pos bx0,by0:mес "name":input name2:name2="S"+str(name2)//:被験者番号入力 pos bx0,180 : BUTTON "実験" ,*kkk1 pos bx0,240 : BUTTON "練習" ,*kkk2 pos bx0,300 : BUTTON "テスト" ,*kkk3 pos bx0,360 : BUTTON "教示" ,*kkk4 bx0=400: objsize 100,36: pos bx0,180 : BUTTON " X 軸回転 " ,*kkk5 pos bx0,240 : BUTTON " Y 軸回転 " ,*kkk5 pos bx0,300 : BUTTON " Z 軸回転 " ,*kkk5 : stop //----- *kkk1: CD=1: et ="e": goto *rand1 *kkk2: CD=2: et ="e": goto *rand1 *kkk3: CD=3: et ="t": goto *rand1 *kkk4: CD=9: et ="e": goto *instruction *kkk5: ch123=stat-4 :mes ch123: stop stop://----- *instruction : //color 200,200,120: boxf 00,00,xx0,yy0: color 00,00,120 pos 60,20: font "sytemfont",36: mes "心的回転実験10": mes font "sytemfont",18: gg=250: pos gg,150: mes "反応の方法": mes pos gg,200: mes "これから画面上に2つのブロック物体が表示されます。" pos gg,250: mes "同じものか異なるものかをボタンを押して答えてください。" pos gg,300: mes "出来るだけ正確にすばやく判断してください。" pos gg,350: mes "まずはじめに練習してみましょう。"//:pos 480,400: BUTTON "開始",*kkk7 stop :*kkk7 :goto *rand1 ://*start0 //: goto *trials stop://----- *rand1 :// 1 から N1 まで N9 個ずつのランダム数列(配列SS()に) if (ch123 !=1)and (ch123 !=2) and (ch123 !=3) :goto *begin: gm=10: gp=20: gk=5: // 水準数・ステップサイズ(角度)・くりかえし数 N1=gm*2*gk :N9=1:i=0: rpp=0: z=16: font "sytemfont",z :// -----100試行 repeat : p=rnd(N1)+1:Q1(p)=Q1(p)+1 if Q1(p) > N9 : continue i=i+1: rpp=rpp+1 : SS(rpp)=p pos xx0-90,i*z :mes str(rpp)+" : " :pos xx0-50,i*z :mes p : if rpp >= N1*N9 : :break loop : if et ="t": wait 100: goto *trials: else : goto *trials: stop://----- *start0 : font "sytemfont",18: objsize 160,24 pos 30,520 : BUTTON "フィードバックあり" ,*kFB1 pos 30,580 : BUTTON "フィードバックなし" ,*kFB0 :stop *kFB1 :FB=1: goto *button0: *kFB0 :FB=0: goto *button0: stop stop://----- *button0 ://: 番号ボタン式と乱数列番号指定式がある if et != "e": name2="test" *bango :cls : nn="": //乱数列番号指定式 font "sytemfont",18:objsize 120,24: pos 0,60:mes " RND No.=":pos 86,60:input nn: pos 0,120 : BUTTON " next " ,*selesct1: objsel 0 :stop *selesct1: tm=0: nn=int(nn): if nn=0 :goto *bango pos 0,200 : BUTTON "制限時間なし" ,*selesct0 pos 0,250 : BUTTON "制限時間あり" ,*ari : stop *ari :tm=20000:goto *selesct0 :// 制限時間20秒 stop://----- </pre>	1. 準備 画面セット 変数配列
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78	<pre> gsel 0: objsize 200,36: // 対象提示画面 bx0=20:by0=60:pos bx0,by0:mес "name":input name2:name2="S"+str(name2)//:被験者番号入力 pos bx0,180 : BUTTON "実験" ,*kkk1 pos bx0,240 : BUTTON "練習" ,*kkk2 pos bx0,300 : BUTTON "テスト" ,*kkk3 pos bx0,360 : BUTTON "教示" ,*kkk4 bx0=400: objsize 100,36: pos bx0,180 : BUTTON " X 軸回転 " ,*kkk5 pos bx0,240 : BUTTON " Y 軸回転 " ,*kkk5 pos bx0,300 : BUTTON " Z 軸回転 " ,*kkk5 : stop //----- *kkk1: CD=1: et ="e": goto *rand1 *kkk2: CD=2: et ="e": goto *rand1 *kkk3: CD=3: et ="t": goto *rand1 *kkk4: CD=9: et ="e": goto *instruction *kkk5: ch123=stat-4 :mes ch123: stop stop://----- *instruction : //color 200,200,120: boxf 00,00,xx0,yy0: color 00,00,120 pos 60,20: font "sytemfont",36: mes "心的回転実験10": mes font "sytemfont",18: gg=250: pos gg,150: mes "反応の方法": mes pos gg,200: mes "これから画面上に2つのブロック物体が表示されます。" pos gg,250: mes "同じものか異なるものかをボタンを押して答えてください。" pos gg,300: mes "出来るだけ正確にすばやく判断してください。" pos gg,350: mes "まずはじめに練習してみましょう。"//:pos 480,400: BUTTON "開始",*kkk7 stop :*kkk7 :goto *rand1 ://*start0 //: goto *trials stop://----- *rand1 :// 1 から N1 まで N9 個ずつのランダム数列(配列SS()に) if (ch123 !=1)and (ch123 !=2) and (ch123 !=3) :goto *begin: gm=10: gp=20: gk=5: // 水準数・ステップサイズ(角度)・くりかえし数 N1=gm*2*gk :N9=1:i=0: rpp=0: z=16: font "sytemfont",z :// -----100試行 repeat : p=rnd(N1)+1:Q1(p)=Q1(p)+1 if Q1(p) > N9 : continue i=i+1: rpp=rpp+1 : SS(rpp)=p pos xx0-90,i*z :mes str(rpp)+" : " :pos xx0-50,i*z :mes p : if rpp >= N1*N9 : :break loop : if et ="t": wait 100: goto *trials: else : goto *trials: stop://----- *start0 : font "sytemfont",18: objsize 160,24 pos 30,520 : BUTTON "フィードバックあり" ,*kFB1 pos 30,580 : BUTTON "フィードバックなし" ,*kFB0 :stop *kFB1 :FB=1: goto *button0: *kFB0 :FB=0: goto *button0: stop stop://----- *button0 ://: 番号ボタン式と乱数列番号指定式がある if et != "e": name2="test" *bango :cls : nn="": //乱数列番号指定式 font "sytemfont",18:objsize 120,24: pos 0,60:mes " RND No.=":pos 86,60:input nn: pos 0,120 : BUTTON " next " ,*selesct1: objsel 0 :stop *selesct1: tm=0: nn=int(nn): if nn=0 :goto *bango pos 0,200 : BUTTON "制限時間なし" ,*selesct0 pos 0,250 : BUTTON "制限時間あり" ,*ari : stop *ari :tm=20000:goto *selesct0 :// 制限時間20秒 stop://----- </pre>	2. モード選択
43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78	<pre> *instruction : //color 200,200,120: boxf 00,00,xx0,yy0: color 00,00,120 pos 60,20: font "sytemfont",36: mes "心的回転実験10": mes font "sytemfont",18: gg=250: pos gg,150: mes "反応の方法": mes pos gg,200: mes "これから画面上に2つのブロック物体が表示されます。" pos gg,250: mes "同じものか異なるものかをボタンを押して答えてください。" pos gg,300: mes "出来るだけ正確にすばやく判断してください。" pos gg,350: mes "まずはじめに練習してみましょう。"//:pos 480,400: BUTTON "開始",*kkk7 stop :*kkk7 :goto *rand1 ://*start0 //: goto *trials stop://----- *rand1 :// 1 から N1 まで N9 個ずつのランダム数列(配列SS()に) if (ch123 !=1)and (ch123 !=2) and (ch123 !=3) :goto *begin: gm=10: gp=20: gk=5: // 水準数・ステップサイズ(角度)・くりかえし数 N1=gm*2*gk :N9=1:i=0: rpp=0: z=16: font "sytemfont",z :// -----100試行 repeat : p=rnd(N1)+1:Q1(p)=Q1(p)+1 if Q1(p) > N9 : continue i=i+1: rpp=rpp+1 : SS(rpp)=p pos xx0-90,i*z :mes str(rpp)+" : " :pos xx0-50,i*z :mes p : if rpp >= N1*N9 : :break loop : if et ="t": wait 100: goto *trials: else : goto *trials: stop://----- *start0 : font "sytemfont",18: objsize 160,24 pos 30,520 : BUTTON "フィードバックあり" ,*kFB1 pos 30,580 : BUTTON "フィードバックなし" ,*kFB0 :stop *kFB1 :FB=1: goto *button0: *kFB0 :FB=0: goto *button0: stop stop://----- *button0 ://: 番号ボタン式と乱数列番号指定式がある if et != "e": name2="test" *bango :cls : nn="": //乱数列番号指定式 font "sytemfont",18:objsize 120,24: pos 0,60:mes " RND No.=":pos 86,60:input nn: pos 0,120 : BUTTON " next " ,*selesct1: objsel 0 :stop *selesct1: tm=0: nn=int(nn): if nn=0 :goto *bango pos 0,200 : BUTTON "制限時間なし" ,*selesct0 pos 0,250 : BUTTON "制限時間あり" ,*ari : stop *ari :tm=20000:goto *selesct0 :// 制限時間20秒 stop://----- </pre>	3. 教示
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78	<pre> *rand1 :// 1 から N1 まで N9 個ずつのランダム数列(配列SS()に) if (ch123 !=1)and (ch123 !=2) and (ch123 !=3) :goto *begin: gm=10: gp=20: gk=5: // 水準数・ステップサイズ(角度)・くりかえし数 N1=gm*2*gk :N9=1:i=0: rpp=0: z=16: font "sytemfont",z :// -----100試行 repeat : p=rnd(N1)+1:Q1(p)=Q1(p)+1 if Q1(p) > N9 : continue i=i+1: rpp=rpp+1 : SS(rpp)=p pos xx0-90,i*z :mes str(rpp)+" : " :pos xx0-50,i*z :mes p : if rpp >= N1*N9 : :break loop : if et ="t": wait 100: goto *trials: else : goto *trials: stop://----- *start0 : font "sytemfont",18: objsize 160,24 pos 30,520 : BUTTON "フィードバックあり" ,*kFB1 pos 30,580 : BUTTON "フィードバックなし" ,*kFB0 :stop *kFB1 :FB=1: goto *button0: *kFB0 :FB=0: goto *button0: stop stop://----- *button0 ://: 番号ボタン式と乱数列番号指定式がある if et != "e": name2="test" *bango :cls : nn="": //乱数列番号指定式 font "sytemfont",18:objsize 120,24: pos 0,60:mes " RND No.=":pos 86,60:input nn: pos 0,120 : BUTTON " next " ,*selesct1: objsel 0 :stop *selesct1: tm=0: nn=int(nn): if nn=0 :goto *bango pos 0,200 : BUTTON "制限時間なし" ,*selesct0 pos 0,250 : BUTTON "制限時間あり" ,*ari : stop *ari :tm=20000:goto *selesct0 :// 制限時間20秒 stop://----- </pre>	4. 乱数列作成
62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78	<pre> *start0 : font "sytemfont",18: objsize 160,24 pos 30,520 : BUTTON "フィードバックあり" ,*kFB1 pos 30,580 : BUTTON "フィードバックなし" ,*kFB0 :stop *kFB1 :FB=1: goto *button0: *kFB0 :FB=0: goto *button0: stop stop://----- *button0 ://: 番号ボタン式と乱数列番号指定式がある if et != "e": name2="test" *bango :cls : nn="": //乱数列番号指定式 font "sytemfont",18:objsize 120,24: pos 0,60:mes " RND No.=":pos 86,60:input nn: pos 0,120 : BUTTON " next " ,*selesct1: objsel 0 :stop *selesct1: tm=0: nn=int(nn): if nn=0 :goto *bango pos 0,200 : BUTTON "制限時間なし" ,*selesct0 pos 0,250 : BUTTON "制限時間あり" ,*ari : stop *ari :tm=20000:goto *selesct0 :// 制限時間20秒 stop://----- </pre>	5. フィードバック 設定
68 69 70 71 72 73 74 75 76 77 78	<pre> *button0 ://: 番号ボタン式と乱数列番号指定式がある if et != "e": name2="test" *bango :cls : nn="": //乱数列番号指定式 font "sytemfont",18:objsize 120,24: pos 0,60:mes " RND No.=":pos 86,60:input nn: pos 0,120 : BUTTON " next " ,*selesct1: objsel 0 :stop *selesct1: tm=0: nn=int(nn): if nn=0 :goto *bango pos 0,200 : BUTTON "制限時間なし" ,*selesct0 pos 0,250 : BUTTON "制限時間あり" ,*ari : stop *ari :tm=20000:goto *selesct0 :// 制限時間20秒 stop://----- </pre>	6. モード設定


```

79 *select0: pos 0,300 : BUTTON "ボタン式",*button00
80 //pos 0,350 : BUTTON "乱数列式",*randomNo : stop
81 *button00 :rb="b":cls :goto *trials
82 stop://-----
83 repeat 100,1 :// 1から100まで
84 r= strmid (rr(nn),cnt*3-3,2) :// 文字列として読み込み
85 rs(cnt)= int(r):if et!="e": mes rs(cnt) :// 数値として取り出し
86 loop :goto *trials
87 stop://-----
88 *assignment: sk=btn:// 刺激番号で異同と角度差を割り当てる
89 //gm=10: gp=20: gk=5: // 水準数・ステップサイズ(角度)・くりかえし数
90 ssk=sk: // ssk= 刺激番号(オリジナル :// 50回以下は「same」条件(gd=0)
91 gd=0: if sk>gm*gk :sk=sk-gm*gk: gd=1 :// 51回以上は「different」条件(gd=1)
92 if gk*00+1<= sk and sk<= gk*01 : ang=00*gp:// 1から5までなら0度
93 if gk*01+1<= sk and sk<= gk*02 : ang=01*gp:// 6から10までなら20度
94 if gk*02+1<= sk and sk<= gk*03 : ang=02*gp://
95 if gk*03+1<= sk and sk<= gk*04 : ang=03*gp://
96 if gk*04+1<= sk and sk<= gk*05 : ang=04*gp://
97 if gk*05+1<= sk and sk<= gk*06 : ang=05*gp://
98 if gk*06+1<= sk and sk<= gk*07 : ang=06*gp://
99 if gk*07+1<= sk and sk<= gk*08 : ang=07*gp://
100 if gk*08+1<= sk and sk<= gk*09 : ang=08*gp://
101 if gk*09+1<= sk and sk<= gk*010: ang=09*gp://
102 return: // ***ここから試行ループ開始*****
103 *trials: ot=0:color 0,0,0: // オーバータイムのフラグ
104 gsel 0: cls: hreset: gsel 1: cls: // 両画面クリア
105 //mes1="":repeat N1,1:mes1=mes1+ str(cnt)+":"+str(SS(cnt))+":":loop: mes mes1: stop
106 ii=ii+1: if ii=1 t00= gmsec(): //経過時刻も記録 ii は試行カウンター
107 objsize 160,36: objmode 2:font "systemfont",18 :
108 if ii=1: ttt1=gettime(3)*60*60*24+gettime(4)*60*60+gettime(5)*60+gettime(6)://開始秒
109 if ii=1: ttt1= str(gettime(0))+":"+str(gettime(1))+":"+str(gettime(3))
110 if ii=1: ttt1= ttt1 +":"+str(gettime(4))+":"+str(gettime(5))+":"+str(gettime(6))
111 if ii=61 : BUTTON "休憩時間",*prepare1: stop:// 60 試行で休憩入れる
112 if ii > N1 : goto *FILE1
113 goto *prepare1:// ボタン選択でない場合
114 //-----
115 font "systemfont",20:color 00,00,00 :pos 30,30:mes "< 刺激番号 >://刺激番号ボタン式
116 // 刺激番号ボタン行列を表示する (smallest version)
117 repeat zz:stim(cnt)=0:ans(cnt)=0:bn(cnt)=0:rt(cnt)=0:yn(cnt)=0:loop
118 x0=24 :y0=40:vn=10: hn=10: h=60: v=40: hh=h+7: vv=v+7 :// 原点とボタンサイズと間隔
119 repeat vn,1:i=cnt:repeat hn,1:j=cnt: bn=j*(hn)*(i-1) :// ボタン番号 1 から
120 objmode 2: color 000,000,000 :font "systemfont",36:// ボタンの文字サイズ
121 objsize h,v :pos x0+(j-1)*hh,y0+(i-1)*vv+h:p1(bn)=0:button str(bn),*prepare1
122 loop :loop:font "systemfont",18 :pos xx0-200,yy0-60:objsize 160,40:BUTTON "記録",*FILE1
123 stop : *response2:bn2(stat+1)=1:nn=(stat+1):mes str(nn):gosub *trials
124 stop://-----
125 *prepare1: //cls:
126 if rb="b": btn=stat+1: ://if rb="r":
127 btn=SS(ii): gosub *assignment: // 刺激番号 ----- *assignment
128 gsel 1:cls :// コントロール・表示画面にボタンをセット
129 if et="e"and ii=1: pos 710,100: objsize 60,36: BUTTON "start",*start1: stop
130 *start1: clobj 0,0: objsize 120,36: objmode 2: font "systemfont",26
131 ma="同じ": mb="異なる": gg=10: ///aa=ma: bb=mb
132 // 左右のボタンにランダムに反応文字を割り当て
133 bt=int(rnd(2)): bbt(ii)=bt: py3=550: key=0
134 if bt=0: aa=str(ma): bb=str(mb):// bt=0 で正順
135 if bt=1: aa=str(mb): bb=str(ma):// bt=1 で逆順
136 pos 250,gg : BUTTON aa,*response:
137 pos 450,gg : BUTTON bb,*response:font "systemfont",18
138 if et="t": pos 710,gg : objsize 60,36: BUTTON "start",*FILE1
139 if et="t"and ii=1 : pos 010,gg : objsize 120,36: BUTTON " random trial", *objects
140 if et="t": pos 710,60 : objsize 120,36: BUTTON " next", *trials:
141 if et="t": pos 710,gg : objsize 60,36: BUTTON "FILE",*FILE1
142 if et="t" and (ii > 1) :goto *objects
143 if et="p": pos 710,gg+160: font "systemfont",10: objsize 60,26: BUTTON "はじめから",*begin9:
144 font "systemfont",26: if et="e": goto *objects
145 stop: *begin9: end :// はじめから
146 stop://-----
147 *objects:// 刺激図形提示
148 CD=0: CC=0: dd=1: motion=0: ss=1: //step=0.1:
149 step=0.174532925f ://(10 degrees, (3.14159265*2)/36 )
150 gsel 0: font "systemfont",36 :// 対象提示画面
151 cls: pos xx0/2-15,yy0/2-30: mes "X": //-----// 凝視点(時間 tttt)
152 wait tttt: font "systemfont",18: gsel 1: // コントロール・表示画面
153 color 255,255,255: boxf 0,0,xx0,100: color 0,0,0: // 文字消去
154 gsel 0: CC=gd:// 対象提示画面 + 異同選択
155 //CC=int(rnd(2)):// CC=0 (same), CC=1(different) 異同ランダム選択
156 ch=ch123: //ch =int(rnd(3))+1: :// ch 1-3 (x,y,z)どの軸で回転するか
157 rr1=int(rnd(19)): dd1=rr1: step1=step*rr1: // 0-180 度の中からランダム選択/(右対象)
158 //rr2=int(rnd(19)): dd2=rr2: step2=step*rr2: // 0-180 度の中からランダム選択/(左対象)
159 rr2=int(ang/10)+rr1: dd2=rr2: step2=step*rr2 // 0-180 度の中からランダム選択/(左対象)
160 grr1(ii)=rr1: grr2(ii)=rr2:// 物体角度の記録
161 both=1: rx1=0: ry1=0: rz1=0: rx2=0: ry2=0: rz2=0
162 //-----

```

7. 刺激番号
割り当て

8. 初期設定

9. 刺激番号
ボタン式

10. 初期設定
ボタン設定

11. 刺激図形
提示

```

163 // スクリーン0(対象提示画面)設定・開始
164 clscolor $00FFFF; clscolor $eeeeee; setfont 16,16,12,1:// 背景色設定
165 mxload "Block-1R2": block1 = stat
166 modelshade block1, 2,0
167 //(右対象)//mxload "Block-1L2": block2 = stat
168 if CC=0: mxload "Block-1R2": block2 = stat
169 if CC=1: mxload "Block-1L2": block2 = stat
170 modelshade block2, 2,0
171 //gsel 0: // スクリーン0(対象提示画面):カメラ設定
172 cammode CAM_MODE_LOOKAT://cammode CAM_MODE_NORMAL
173 //selcpos: objsetf3 0.0f, 0.0f, 24.0f; カメラ座標//----同じ見え方にするため24.0f
174 selcpos: objsetf3 0.0f, 0.0f, 24.0f; カメラ座標//----同じ見え方にするため24.0f
175 selcint: objsetf3 0.0f, 0.0f, 0.0f ; 注視点座標
176 selcang: objsetf3 0.0f, 0.0f, 0.0f ; カメラ角度
177 //object配置:// 他軸の傾き入れる
178 bbb=int(rnd(2))//( Y 軸回転の場合には「bbb=0 手前傾き bbb=1 後方傾き」)
179 regobj BLK1, block1: //left(左対象)
180 selpos BLK1: objsetf3 -4.0f, 0.0f, 0.0f
181 if ch=1 and bbb= 0: selang BLK1: objaddf3 0.0f, 0.4f, 0.0f // ch=1 X 軸回転の場合
182 if ch=1 and bbb= 1: selang BLK1: objaddf3 0.0f,-0.4f, 0.0f // y=0.4f 少し傾けた
183 if ch=2 and bbb= 0: selang BLK1: objaddf3 0.4f, 0.0f, 0.0f // ch=2 Y 軸回転の場合
184 if ch=2 and bbb= 1: selang BLK1: objaddf3 -0.4f, 0.0f, 0.0f // x=0.4f 少し傾けた
185 if ch=3 and bbb= 0: selang BLK1: objaddf3 0.0f, 0.0f, 0.4f // ch=3 Z 軸回転の場合
186 if ch=3 and bbb= 1: selang BLK1: objaddf3 -0.4f, 0.0f, 0.0f // x=0.4f 少し傾けた
187 regobj BLK2, block2: //right(右対象)
188 selpos BLK2: objsetf3 4.0f, 0.0f, 0.0f
189 if ch=1 and bbb= 0: selang BLK2: objaddf3 0.0f, 0.4f, 0.0f // ch=1 X 軸回転の場合
190 if ch=1 and bbb= 1: selang BLK2: objaddf3 0.0f,-0.4f, 0.0f // y=0.4f 少し傾けた
191 if ch=2 and bbb= 0: selang BLK2: objaddf3 0.4f, 0.0f, 0.0f // ch=2 Y 軸回転の場合
192 if ch=2 and bbb= 1: selang BLK2: objaddf3 -0.4f, 0.0f, 0.0f // x=0.4f 少し傾けた
193 if ch=3 and bbb= 0: selang BLK2: objaddf3 0.0f, 0.0f, 0.4f // ch=3 Z 軸回転の場合
194 if ch=3 and bbb= 1: selang BLK2: objaddf3 -0.4f, 0.0f, 0.0f // x=0.4f 少し傾けた
195 color 0,0,0:lr=9: t1= gmsec(): //if tm != 0 : goto *timelimit
196 *DRAWLOOP2:// 対象表示ループ
197 // 2対象の一方を1軸ずつ指定してステップ回転させるルーチン
198 selang BLK1: // 左対象 //step1=step: step2=step: dd1=step1: dd2=step2
199 if ch=1: objaddf3 step1, 0.0f, 0.0f: rx1=rx1+dd1: goto *g22://*hgd2
200 if ch=2: objaddf3 0.0f, step1, 0.0f: ry1=ry1+dd1: goto *g22://*hgd2
201 if ch=3: objaddf3 0.0f, 0.0f, step1: rz1=rz1+dd1: goto *g22://*hgd2
202 *g22: selang BLK2: // 右対象
203 if ch=4 or (both=1 and ch=1): objaddf3 step2, 0.0f, 0.0f: rx2=rx2+dd2: goto *hgd2
204 if ch=5 or (both=1 and ch=2): objaddf3 0.0f, step2, 0.0f: ry2=ry2+dd2: goto *hgd2
205 if ch=6 or (both=1 and ch=3): objaddf3 0.0f, 0.0f, step2: rz2=rz2+dd2: goto *hgd2
206 *hgd2: hgdraw:// goto *pass0: //*hyouji1
207 gsel 1: font "sytemfont",18:// コントロール・表示画面
208 if et="t": pos 10,60: mes "ssk=" +str(ssk)+"sk=" +str(sk)+"gd=" +str(gd)+"ang=" +str(ang)
209 if ch=9: chh=ch: title str(ii): xyz=""
210 if CC=0: cc4="(same)": else: cc4="(different)"
211 color 255,255,255: boxf 000,80,xx0,200: color 0,0,0: gg=140: // -----表示
212 if et="e": goto *pass1
213 pos 020,gg: mes "X1="+str(rx1) : pos 020,gg+30: mes "X2="+str(rx2)
214 pos 100,gg: mes "Y1="+str(ry1) : pos 100,gg+30: mes "Y2="+str(ry2)
215 pos 180,gg: mes "Z1="+str(rz1) : pos 180,gg+30: mes "Z2="+str(rz2)
216 pos 300,gg: mes "CC="+str(CC)+" "+cc4: pos 300,gg+30: mes "ch="+str(chh)
217 xyz=str(CC)+" "+str(rx1)+" "+str(ry1)+" "+str(rz1)+" "+str(rx2)+" "+str(ry2)+" "+str(rz2)
218 pos 450,gg:mes "STEP1="+str(int(dd1+10)/10):pos 450,gg+30:mes "STEP2="+str(int(dd2+10)/10)
219 *pass1: gsel 0: hgsync 1: // 対象提示画面
220 // 反応入力待ち-----
221 if motion=0: ch=9:// 静止の場合はループの中で変化条件(ch=1-6)以外に設定(ch=9)
222 ot=0: key=0: // キー入力待ちと制限時間テック
223 *key20: ot9=0: key=0:getkey key,49: if key =1: key=1: goto *key21:// キーの1
224 keyname="": getkey key,51: if key =1: key=3: goto *key21:// キーの3
225 getkey key,57: if key =1: key=9: cls: pos 100,0: font "sytemfont",66: mes "STOP":stop
226 : getkey key,97: if key =1: key=1: goto *key21:// テンキーの1
227 : getkey key,99: if key =1: key=3: goto *key21:// テンキーの3
228 //ジョイスティック反応 (サンワサブライJY-P56US 使用)-----
229 JStick key, 0x800000-1, padid
230 if et="t" and padid = -1: //
231 if key& (131072*1): keyname = "7": key=1: goto *key21
232 if key& (131072*2): keyname = "8": key=3: goto *key21
233 if key& (131072*4): keyname = "9": key=1: goto *key21
234 if key& (131072*8): keyname = "10": key=3: goto *key21
235 if et="t": pos 0,30 : mes keyname
236 goto *DRAWLOOP2:// 対象表示ループ
237 // -----
238 // t2= gmsec() : if t2-t1>tm: ot=1: r9=99: goto *response://制限時間
239 //await 2:goto *response:// onkey *key1: onclick *key1: // 反応待ち
240 *key21: r9=key: cls: mes r9: goto *response:// 反応キー記録(1 or 3)
241 stop://-----

```

12. 対象提示画面
設定・開始

13. 対象提示
左対象
右対象
コントロール

14. 反応
入力待ち
(ループ)

```

242 *response: if key=0: r9=stat: // (キーが押されない時)画面の反応ボタンは左が0、右が1
243   if key=1: r9=0 : // 左キー(1)は0に変換
244   if key=3: r9=1 : // 右キー(3)は1に変換
245   bnn=r9: t2= gmsec(): // 反応キー番号 // 結果表示・記録
246   gsel 1: // コントロール・表示画面
247   if et="e": clobj 0,1: // 反応ボタンを消す // pos 0,0: mes "bnn="+str(bnn): stop
248   if ot=1: bnn=99: // オーバータイムの反応番号
249   // 正誤判定のための正解ボタン選定
250   if bt=0 :an=0: // ボタンが正順なら an=0
251   if bt=1 :an=1: // ボタンが逆順なら an=1
252   //-----
253   *record1: // original 刺激番号は btn
254   ccc=ccc+1 : // 試行数カウンタ
255   bbtn(ccc)=btn : // original 刺激番号
256   Q(ccc)=gd : // 「same」条件(gd=0)
257   A(ccc)=ang : // 角度差
258   bbn(ccc)=bnn : // 反応ボタン番号(r9)
259   if bt=1 and bbn(ccc)=0: bnn=1: bbn(ccc)=1: goto *p9: //-----反応逆転
260   if bt=1 and bbn(ccc)=1: bnn=0: bbn(ccc)=0:
261   *p9: rrt(ccc)=t2-t1 : // RT(ミリ秒)
262   if gd=bnn :pnt=1 :else: pnt=0
263   stt(ccc)=str(ma)+","+str(mb)+","+aa+","+bb
264   ccr(ccc) =pnt : // 正誤 ( 1=right 0=wrong )
265   xyz(ccc)=xyz : // 回転数値情報
266   //-----
267   *nfb: gsel 1: // コントロール画面 + 結果表示 + フィードバック
268   font "sytemfont",20: color 000,000,000 :pos 300,0: gg=100
269   // "試行数"+","+刺激番号"+","+異同"+","+反応"+","+反応時間"+","+スコア" ---- 表示
270   // et = "t": // test
271   if et = "t": pos 010,gg: mes ccc: pos 050,gg: mes "no."+str(btn)
272   if et = "t": pos 120,gg: mes "gd="+str(Q(ccc)): pos 180,gg: mes "ang="+str(A(ccc))
273   if et = "t": pos 270,gg: mes "R="+str(bbn(ccc)): pos 330,gg: mes rrt(ccc):
274   if et = "t": pos 400,gg: mes "j="+str(ccr(ccc))
275   if et = "t": pos 460,gg: mes "bbt="+str( bbt(ccc)):
276   if et = "t": pos 520,gg: mes "rr="+str(grr1(ccc))+str(grr2(ccc))
277   color 255,000,000 : //フィードバック
278   if et = "t" and ccr(ccc)=1 : pos 740,gg: mes "right" :
279   if et = "t" and ccr(ccc)=0 : pos 740,gg: mes "wrong" :
280   if et = "t": pos 410,60: mes xyz: color 000,000,000
281   // et = "e": // test
282   // if et="t": pos 710,60 : objsize 120,36: BUTTON "next" ,*trials:
283   if et = "t": stop: goto *trials:
284   if et = "e": wait tttt : goto *trials: // 自動モード試行間隔時間
285   stop: //-----
286   *FILE1: // 配列変数(1次元)をCSV形式で保存する
287   gsel 0: wiki1="": wiki2="": // 対象提示画面
288   cls :font "sytemfont",16 : color 255,000,000 : // 赤で表示
289   b=10: bb=18: CR="¥r": // CRLF="¥n¥r"
290   tt7= gettime(3)*60*60*24+gettime(4)*60*60+gettime(5)*60+gettime(6): // 終了秒
291   tt8= str(tt7-tt1): name2=name2 + "-" + CD+"-"+str(ch123): // 全所要時間秒測定
292   tt9= name2 + ":" + ttt1+">>" + CR
293   tt9= tt9 + str(gettime(0))+str(gettime(1))+str(gettime(3))
294   tt9= tt9 + str(gettime(4))+str(gettime(5))+str(gettime(6)) : // 終了時刻
295   tt9 = tt9 + " Finished (" + tt8+ " sec." ) + "CD=" + CD: nnn="no.": ppp=""
296   iii="試行数"+","+刺激番号"+","+異同"+","+角度"+","+反応"+","+RT"+","+判定"+","
297   iii=iii+"ボタン配置"+","+物体角度": // "+","文字順"+",""+ボタン順"
298   ssss = tt9 + CR +CR+ iii+ CR: :kkk=0
299   repeat ccc,1
300   ppp = ppp+str(cnt)+","+bbtn(cnt)+","+Q(cnt)+","+ A(cnt)+","+ bbn(cnt)+""
301   ppp = ppp+rrt(cnt)+","+ccr(cnt)+","+bbt(cnt)+","+xyz(cnt)+","+grr1(cnt)+","+grr2(cnt)
302   pp(cnt) = ppp : ssss = ssss + pp(cnt) +CR: pos b+000,bb*cnt:
303   wiki1= str(cnt)+","+str(Q(cnt))+","+str(A(cnt))+","+str(bbn(cnt))+","
304   wiki2= str(rrt(cnt))+","+str(ccr(cnt))+","+str(bbt(cnt))+","+str(xyz(cnt))
305   +","+str(grr1(cnt))+","+str(grr2(cnt))
306   mes wiki1+wiki2 :pos b+300,bb*cnt: mes pp(cnt) :ppp=""
307   loop: pos 20,bb*(ccc+1): sss=tt9+CR+nnn+CR+sss+CR+rrt: mes ssss : notesel ssss
308   if et="e":
309   notesave name2+".csv": // 実験時のみ記録ファイル作成
310   font "sytemfont",66 : color 000, 255,000: pos 60,240 : mes "終わりです。お疲れさま。"
311   stop: //-----

```

15. データ記録
反応画面表示

16. データ
ファイル保存