Institute of Socio-Economic Planning
Discussion Paper Series
No. 604

HyperScaling in Image Research

by

MATSUDA,Noriyuki*,and NAMATAME,Miki**
**Institute of Socio-Economic Planning, Univ. of Tsukuba*
***Design Research, Tsukuba*

Sep, 30,1994

The authors are thankful to Mr.SENO,Makoto for his exellent technical assistance.

Contact:
MATSUDA,Noriyuki
Associate Professor
Inst. of Socio-Economic Planning,Univ.of Tsukuba
1-1-1 Tennoudai,Tsukuba,Ibaraki 305
JAPAN

tel: (0298) 53-5170 fax:(0298)55-3849
email:mazda@shako.sk.tsukuba.ac.jp

Image Research Report

1章　特徴

1－1　特徴

HyperScalingのスタック群の特徴を以下に示す.

○　HyperTextによるanchoring機能

○　反応記録をフィールド"record"(反応保持のために設定したフィールドはすべてこの名前を与えられる.）保存し共有化を図る.

○　また上記のことにより測定項目の増設，削除が容易におこなえるようになった.

○　操作の統一性を保っている．例えばボタンの表示に関しては"showUp",カードの再構築，設定には"reset"というメッセージを使う.

○　カードの移動は，被験者が"OK","NEXT"などのボタンを押す事によって行われる．これにより被験者にある程度の主導権を感じされられる.

○　操作はボタンオブジェクトへのマウス操作である．クリック，ドラッグ，ホールドダウンなどが用いられる.

このレポートでは実例として，化粧品メーカーのブランド商品（スキンケアが主）の消費者評価への適用を紹介する.

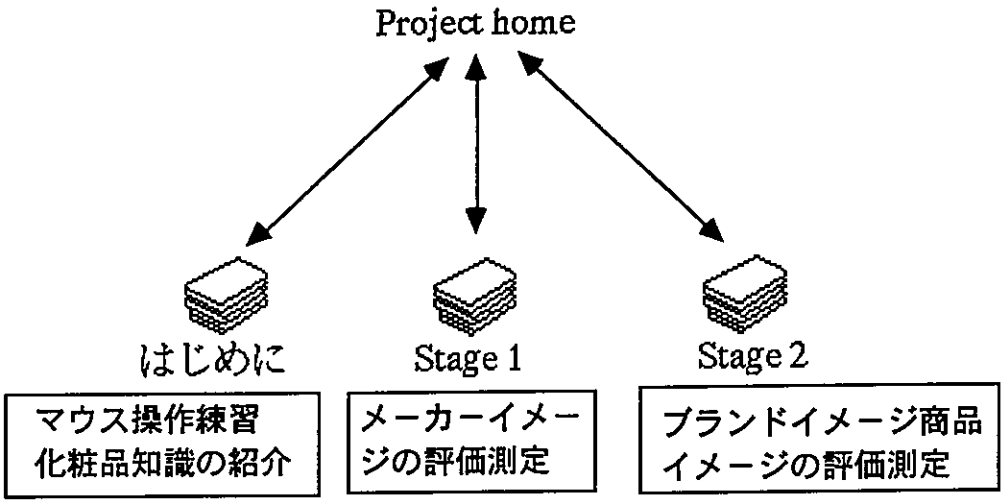
2章　スタック群について

2－1　構成

Image researchのスタック群の目的は，化粧品に対するイメージの調査である．

このprojectの基点は，"Project home"である．ここにはprojectの作動に必要なハンドラやタスクスタックに共通なハンドラが書き込まれている．

ホームスタックの下に３つのタスクスタック（"はじめに"，"stage1"，"stage2"）がある．



projectの作動に必要なハンドラには，環境設定，各タスクスタックへの移動や被験者の反応記録の書き出し外部ファイルの設定などがある．

```
on prepare
   prepareImageLists -- gCIList,gBIList

  --disable cd btn "OK"
...
...
  hide cd btn "name cover"
end prepare
```

projectの準備用ハンドラ
gCIList: メーカー名リスト登録
gBIList: ブランド名リスト登録

各タスクスタックに共通なハンドラとしては，データの作成や外部ファイルへの書き出しなどがある．

外部ファイルへの書き出しハンドラ
gOutFile: 反応記録用外部ファイル名

```
on writeOut what, atEnd
  global gOutFile

  open file gOutFile

  do "write what to file  gOutFile " & atEnd
  close file gOutFile
end writeOut
```

2-2　進行

カードの移動は，被験者が"OK","NEXT"などのボタンを押す事によって行われる．これにより被験者にある程度の主導権を感じされられる．これらのボタンは初期状態では"disable"となっている．所定の条件が満たされたとき各ボタンには"showUp"というメッセージが送られ"enabled"となる．

これに対応したハンドラを以下に示す．

```
on showUp
  enable me
  show me
  if me is 0
  then blink the name of me,2,5

  add 1 to me
end showUp
```

2－3-1　反応記録用外部ファイル

"project home"で反応記録用外部ファイルが作成さる．各タスクスタックからはそれに書き出しが行われる．ファイル名は，"gOutFile"というグローバル変数に入れられるので，他のスタックから参照可能である．タスクスタックでの被験者の反応は，ホームスタックのハンドラ（transferValueなど）をとおして定型化されこのファイルに書き出される．

以下に反応記録用外部ファイルの一例を示す．

```
Name: kawasaki/ female/ Date & Time: 8/12/94 10:55:14 AM

#Task stack: はじめに / 8/12/94 10:55:18 AM

!Stimulus: skincare/ cd: 3 / type: PICT / 10:57:17 AM
Method: Cross / 2 scale(s) / Defalut values: 5.0,5.0
>Scale 1valTags10.0,6.4
^Cross 1
 start 9136.0/5.1,4.8 lap 353.0
5.1,4.8 at 10974.0/10.0,6.4 lap 428.0
>Scale 2valTags7.2,8.9
^Cross 2
 start 11754.0/7.2,8.9 lap 370.0

#Task stack: stage 1 / 8/12/94 10:58:55 AM

!Stimulus: カネボウ/ cd: 2 / type: text / 11:01:22 AM
Method: Cross / 2 scale(s) / Defalut values: 5.0,5.0
>Scale 1 Cross 2.3,8.0
^Cross 1
gDefaultValue start 25524/2.3,8.0 lap 498.0
>Scale 2 Cross 5.1,5.1
^Cross 2
gDefaultValue start 26255/5.1,5.1 lap 466.0
・・・
・・・
```

projectが開かれたときに書き込まれる部分

各タスクスタックが開かれたときに書き込まれる部分

カード全体の情報，初期値，scalingmethodなど

被験者の反応記録

2－3－2　反応記録の保持と共有

各タスクスタックでの被験者の反応は"record"というカードフィールドに書き込まれ保持される．反応を他スケールで利用したい場合は，このフィールドを参照する．フィールドに書き出す事によってグローバル変数を利用しなくても共有する事ができる．更新されない場合も前回のものが残っているので，それを利用できる．

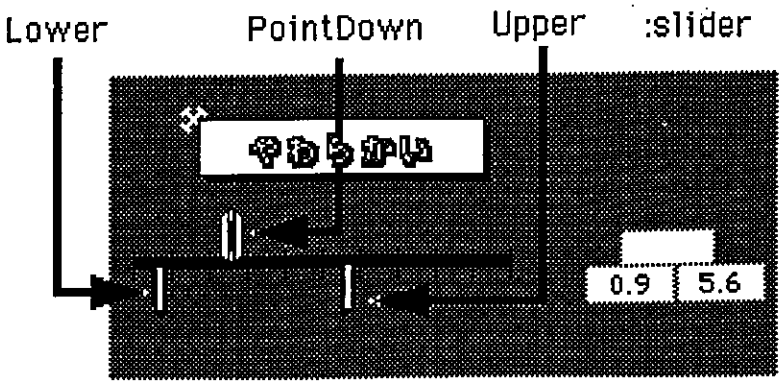
2－4　グローバル変数について

このprojectスタックにおけるグローバル変数はすべて"g"からはじまる．

2－5　Scaling Methodについて

このprojectでは 1 dimensional Scalingと2 dimensional Scalingの2種類のScaling Methodが用いられる． 1 dimensional Scalingでは以下の6つの組み合わせが設定されている．

組み合わせ
Point
Interval
Both
Point > Both
Interval > Both
Point > Interval



1 dimensional Scaling

スライドボタンのスクリプト

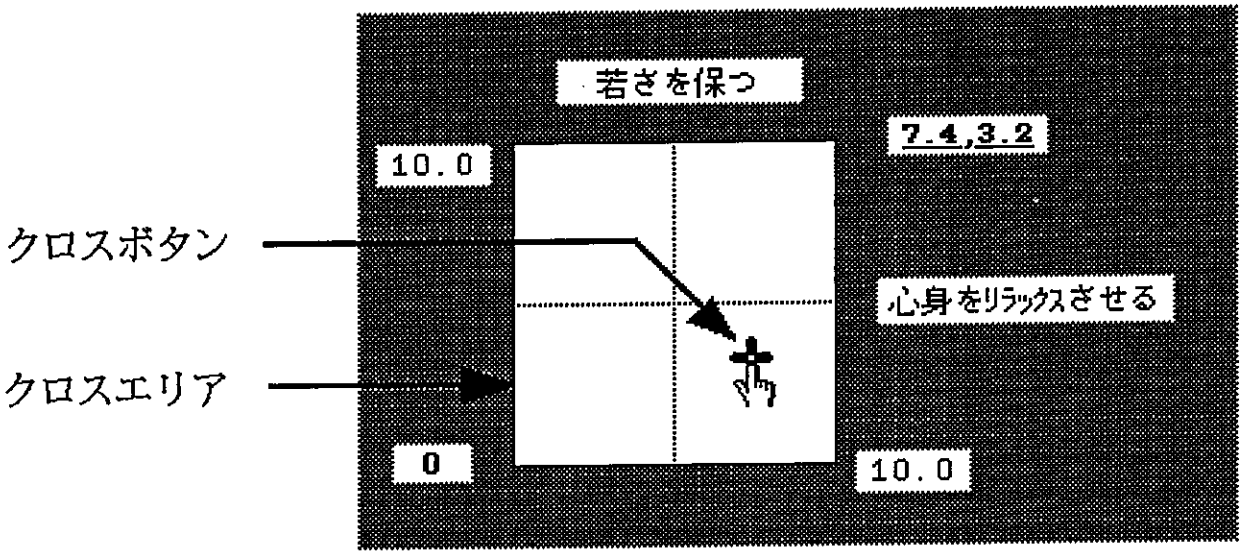
```
on mouseDown
  get the short name of me
  recordStartTime it
  put (the mouseH - the right of me) into adjustH

  slideLower adjustH,0
  recordFinishTime it
end mouseDown
```

"recordStartTime" "recordFinishTime"でマウスダウン，マウスアップの時間を記録．ボタンの動きは"slideLower"ハンドラで制御されている．

---

slideLower等のハンドラは付録P.29におさめてある．

2 dimensional Scalingではクロスのみとなっている



2 dimensional Scaling

クロスボタンのスクリプト

```
on mouseDown
  put the short name of me into myName
  set the numberFormat to "0.0"
  recordStartTime myName

  get the loc of me
  put the mouseH - (item 1 of it) into adjustH
  put the mouseV - (item 2 of it) into adjustV

  slideCross adjustH,adjustV
  recordFinishTime myName
end mouseDown
```

"recordStartTime", "recordFinishTime"でマウスダウン，マウスアップの時間を記録．ボタンの動きは"slideCross"ハンドラで制御されている．

"SlideCross"他関連あるハンドラは付録P.29におさめている．

3章　各スタックについて

3-1　Project　homeスタックについて

このスタックを起動する事によってImage researchがスタートする．このスタックを基点にユーザは３つのタスクスタックを順に起動していくことになる．このスタックが起動されてからの動きを大まかに見ていくと次のようになる．

（１）関係スタックの使用開始

（２）Image researchの初期条件の設定
　　各リストの作成
　　コントロールメニューの作成

（３）ユーザからのデータ入力

（４）"OK"ボタンのハンドラ実行

（５）はじめにへ移行

ハンドラを以下に示す

```
on startUp
   start using stack "utilities"                    （１）に当たる部分
  start using this stack                                  同上

  set the top of card window to 40
  prepareMenu                                          （２）に当たる部分
  prepare                                                 同上
  prepareNameFld "anyone"                                 同上
  select line 1 of cd fld "Person"
  pass startUp -- to the message hieracrchy
end startUp
```

―――― 以下にprepareMenu，prepare，prepareNameFld "anyone"についてそれぞれ説明する．

フィールド"Control"のデータを使ってメニューアイテムとメッセージのリストを作成

```
on prepareMenu
  toLockScreen
  push cd
  go to cd 2 of stack "project home"

  createMenu
  pop cd
  toUnlockScreen
end prepareMenu
```

以下ﾌｨｰﾙﾄﾞ "Control"参照

Special menus

menuItem: menuMsg
menuMsg, when omitted,
= menuItem w/out space

menu 'Control'

Scaling method: scalingMethod
cd attributes: cdAttributes

Reset This cd: resetThisCd
Reset All cds: resetAllCds

Resize btns

Initial setup

```
on prepare
  prepareImageLists -- gCIList,gBIList
  disable cd btn "OK"
  show cd btn "OK"
  put 0 into cd btn "OK"

  put "はじめに,Stage 1,Stage 2,プレゼント申込み" into list
```

"OK"ボタンをdisableに設定し，"0"を入力する．これは，カウンターとしての機能である．

```
  repeat with i=1 to the number of items of list
    disable cd btn (item i of list)
  end repeat
  hide cd btn "name cover"
end prepare
```

"はじめに，Stage 1,Stage 2,プレゼント申込み" への各ボタンを disable に設定する

```
on prepareNameFld who
  put "Person" into obj
  if cd fld obj is empty
  then   put who into cd fld obj
end prepareNameFld
```

カードフィールド"Person"の内容を調べ，空であれば"anyone"を入力する．

ここまでの準備が終了した時点で次にユーザから氏名の入力がある．入力後ユーザが"OK"ボタンを押すことにより，"OK"ボタンのハンドラが実行される．そのハンドラを見ていく．

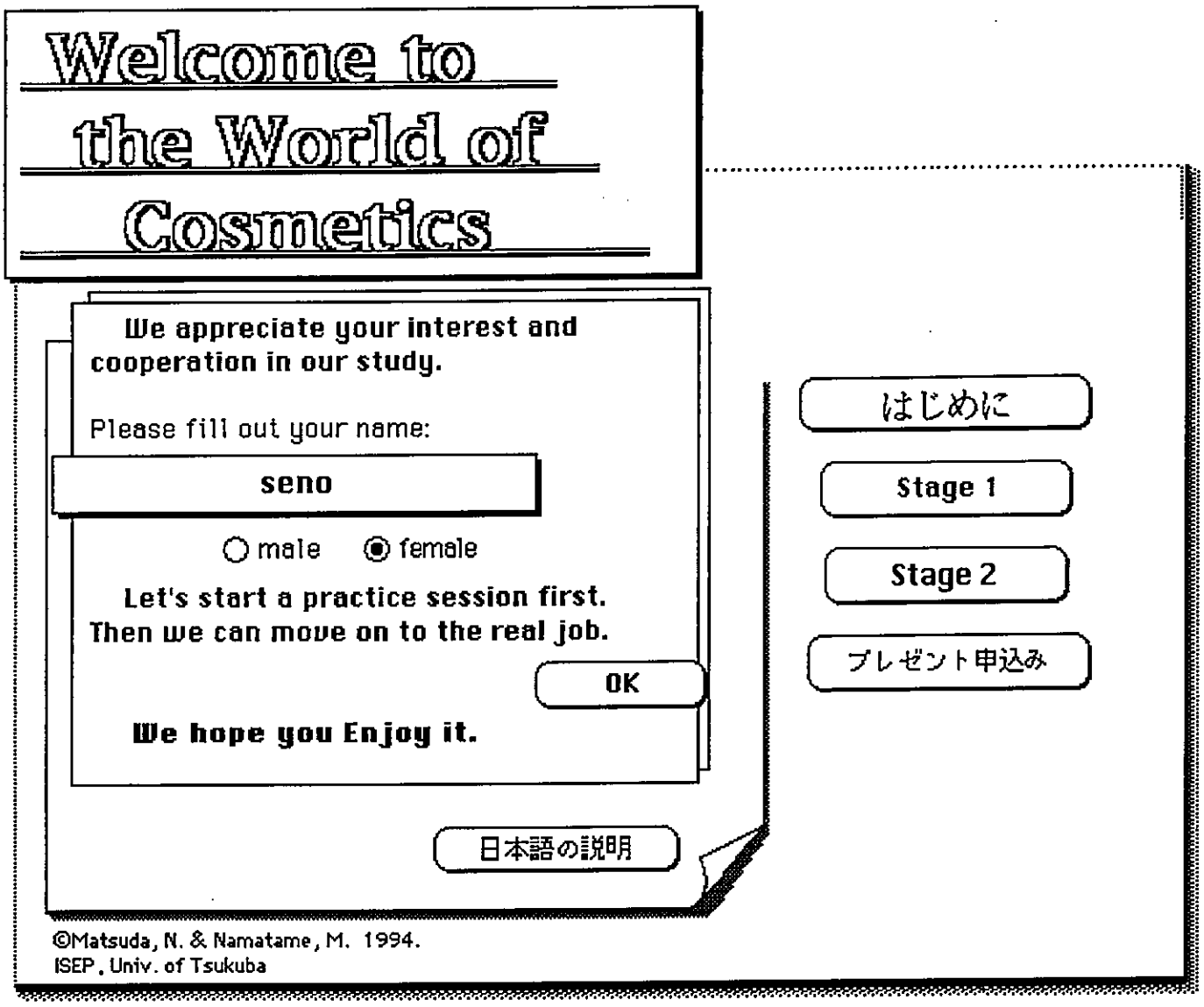
"OK"ボタンのスクリプト

```
on mouseUp
  global gTaskList  -- set in bg script

  transferNameSex  -- in cd script

  put the loc of me into xyxy
  get the rect of cd btn "はじめに"
  set the loc of me to item 1 of it,item 4 of it
  hide me
  send "showUp" to cd btn "はじめに"

  set the loc of me to xyxy
  --  enable cd btn "Stage 1"
end mouseUp
```

被験者氏名と実験日を反応記録ファイル名にし，ファイルを作成する．ファイル名は，グローバル変数gOutFileでやり取りされる．

"showUp"メッセージをカードボタンに送る

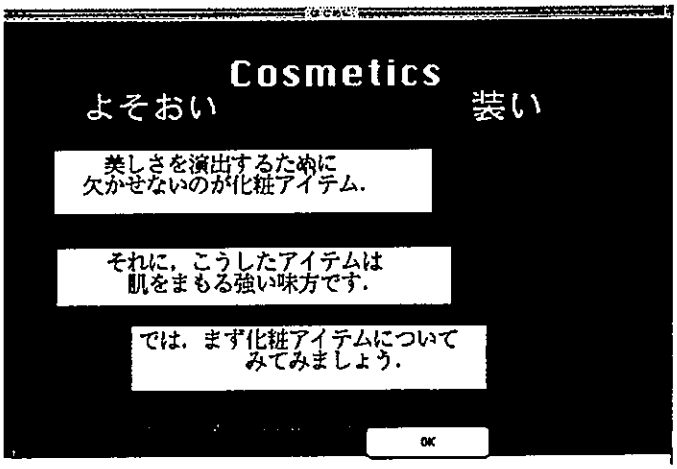
このときの画面状態を次に示す


Welcome to the World of Cosmetics
We appreciate your interest and cooperation in our study.
Please fill out your name:
seno
male
female
Let's start a practice session first. Then we can move on to the real job.
OK
We hope you Enjoy it.
日本語の説明
はじめに
Stage 1
Stage 2
プレゼント申込み
©Matsuda, N. & Namatame, M. 1994.
ISEP, Univ. of Tsukuba

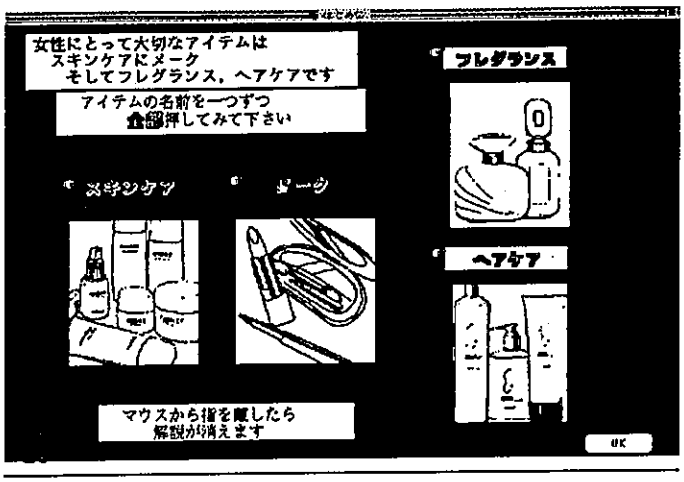
3－2　”はじめに”スタックについて

”はじめに”スタックは，被験者にマウス操作を覚えてもらうためのものである．これは，4枚のカードで構成されている．

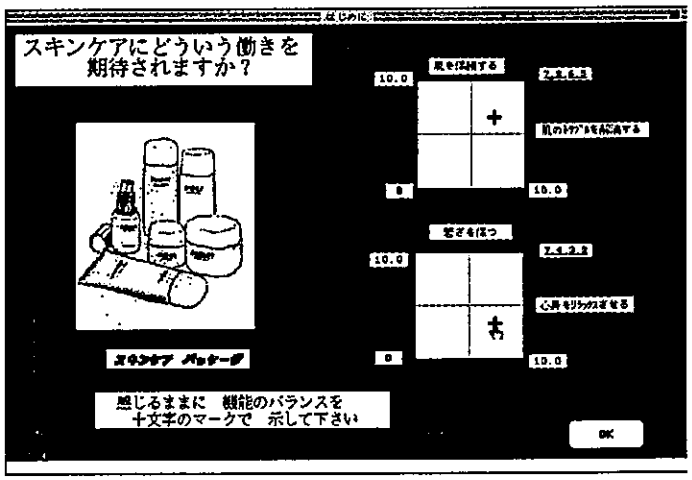
カード１ ----オープニング
カード２ ----アイテム説明（マウスクリックの練習）
カード３ ----スキンケアへの期待（２D scalingの練習）
カード４ ----価格評定(１ D scalingの練習)



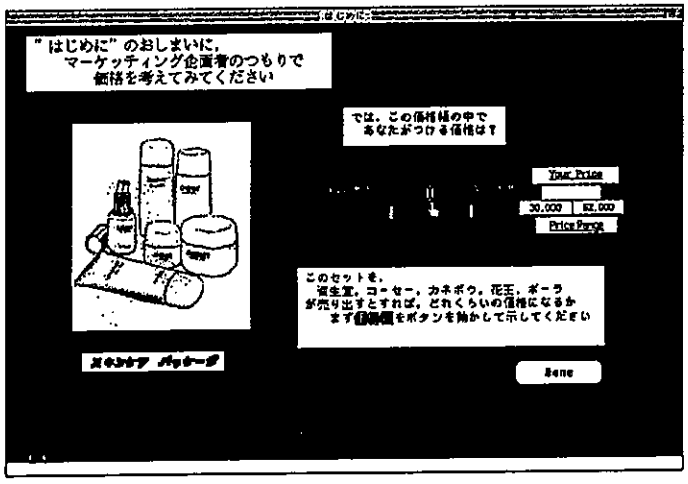
カード１



カード２



カード３



カード４

スタックにおけるカードの流れを見ていく．

```
on openStack
  recordTaskName the short name of this stack

  pass openStack
end openStack
```

"#Task stack:"に続いてスタック名，日付，時間が反応記録用外部ファイルに書き込まれる．

カード1のスクリプト

```
on openCard
  showTitleBtns
  wait 30 ticks
  showNotes
  wait 20 ticks
  send "showUp" to cd btn "OK"

  pass openCard
end openCard
```

隠しておいたタイトル用のボタンやカードフィールドを順次画面上に持ってきている．カード1はオープニングのカードなので被験者は次のカードへ進むために1度ボタンを押す以外は何もしない

カード2のスクリプト

```
on openCard
  global gMisc

  put "スキンケア,メーク,フレグランス,ヘアケア" into gMisc
  put "0,0,0,0" into cd btn "OK" -- as a counter
  wait 50 ticks
  lock screen show cd fld "note B"
  unlock screen with visual effect dissolve
  wait 30 ticks
  show cd fld "note C"

  pass openCard
end openCard
```

ボタン"OK"には"0,0,0,0"が入力されている．このカードには被験者によって操作される4つのボタンがある．各ボタンには例えばi番目のボタンがおされたときボタン"OK"のアイテムiに1加えよというスクリプトがある．

ボタンのスクリプト

```
on mouseUp
  add 1 to item 4 of cd btn "OK"
  set the hilite of me to true
  hide cd fld (the short name of me)
  OKorNot -- in cd script
end mouseUp
```

```
on OKorNot
  get cd btn "OK" -- initially set to "0,0,0,0"
  if min(it) = 1
  then send "showUp" to cd btn "OK"
end OKorNot
```

"OKorNot"では，ボタン"OK"のminを取り，それが0か否かですべてのボタンがクリックされたがどうかを判別している．すべてのボタンがクリックされたと判別したら，"showUp"というメッセージをボタン"OK"に送る．

```
on mouseStillDown
  get the short name of me
  show cd fld it at the loc of cd btn ("haircare")
end mouseStillDown
```

マウスがおされている間は，各項目についての情報を提示する．マウスがボタンから離れると，フィールドは閉じる．

```
on mouseLeave
  if the optionKey is not down
  then hide cd fld (the short name of me)
end mouseLeave
```

カード3スクリプト

```
on openCard
  global gScalingMethod

  put "Cross" into gScalingMethod
  ...
  ...
  put "0,0" into cd btn "OK"

  pass openCard
end openCard
```

Scaling Methodを指示するグローバル変数"g Scaling Method"には"cross"が入力されている．ボタン"OK"に"0,0"を入力しているのは，カード2と同じ様にカウンターとして利用するためである．

```
on closeCard
  global pctName

  nOfScales
  transferScalingInfo
  transferValues
  disable cd btn "OK"

  pass closeCard
end closeCard
```

"nOfScales"は，カード内のスケールの数を調べグローバル変数"gNofScales"に代入する．

"transferScalingInfo"は，反応記録用外部ファイルの頭書部分のデータを作成する．

"transferValues"は，被験者の反応を定型化する．その後メッセージ"writeOut"を発生して，これらのデータを反応記録用外部ファイルに書き出させる．

クロスボタンのスクリプト

```
on mouseUp
  add 1 to item 1 of cd btn "OK"
  ...
  ...
end mouseUp
```

カード４のスクリプト

```
on openCard
  global gPctName,gScalingMethod

  ・・・
 showPCT "skincare"
  ・・・
  put "interval > both" into gScalingMethod
  ・・・

  pass openCard
end openCard
```

Scaling Methodは"interval > both"指定されている．

showPCT "skincare"では，ファイル"skincare"をpictureコマンドで開いている．このハンドラでは，同時に位置や同名のピクチャーウィンドウが存在

スライドボタンのスクリプト

```
on mouseDown
  ・・・
  slideLower adjustH,0
  ・・・
end mouseDown
-----------------------------------------------------------
on slideLower adjustH,valAdjust
  global gIndex,gLend,gRend,gLlimit,gRlimit

  ・・・
  toShowPointDownOrNot "Upper"&& gIndex
end slideLower
-----------------------------------------------------------
on toShowPointDownOrNot partner
  global gScalingMethod,gIndex

  put "PointDown" && gIndex into obj
  if gScalingMethod contains "Both" ッ
  and the visible of cd btn obj is false and cd fld partner is not empty
  then
    ・・・
    ・・・
   send "showUp x,empty" to cd btn obj
  end if
end toShowPointDownOrNot
```

スライドボタンのスクリプトの流れで特徴的なのは"slide…"である．これよりさらに"toShowPointDownOrNot"メセージが発生される．

"toShowPointDownOrNot"ハンドラでは以下の２項目について真偽を求める．

1. ScalingMethodに"both"のオプションがとられているか
2. 対となるスライドボタンの数値フィールドがemptyでないか（つまり被験者によって操作されている）か

共に真のとき "PointDown"スケールを開き "showUp"を送る．

所定の条件が満たされるとボタン"OK"にメッセージ"showUp"が送られenableになる.

ボタン"OK"のスクリプト

```
on mouseUp
  visual effect iris open slowly
  go to stack "Project Home"
  disableEnable "はじめに","Stage 1"
end mouseUp
```

"Preject Home"スタックに戻り"disableEnable"ハンドラ ("Project Home"スタックのスタックスクリプトにある.) でタスクスタック選択ボタン"はじめに"をdisableにして"Stage 1"をenableにする.

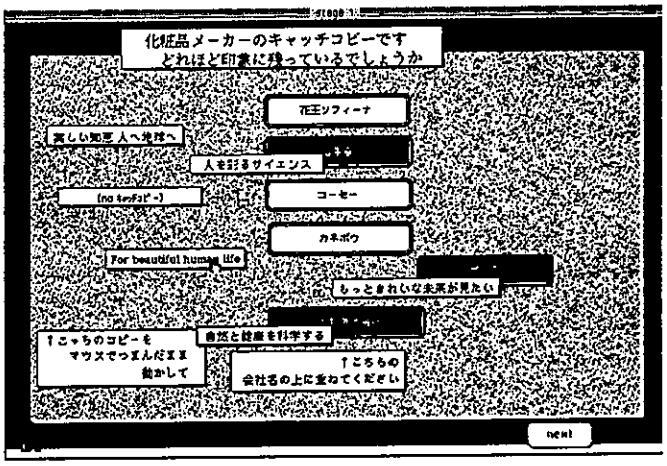
3-3 "Stage 1"スタックについて

"Stage 1"スタックは, 化粧品メーカー, ブランドにたいするイメージの測定を行うためのタスクスタックである. 3枚のカードから構成されている.

カード1 ----キャッチコピーと化粧品メーカーのマッチング
カード2 ----メーカーに対するイメージ測定 (2 Dimensional Scaling)
カード3 ----スキンケアの商品ブランドの提示

カード1について

このカードでは, 化粧品メーカーとそれらが掲げているキャッチコピーとのマッチングを被験者によっておこなってもらう. 初期状態では, 画面左側にキャッチコピー, 中央にメーカー名もしくは商品ブランド名のボタン, フィールドを配置している. カードが開かれた後デモが始まる. それから被験者はキャッチコピーフィールドをドラッグしてマッチングを行う.



スタックスクリプト

```
on openStack
  global gCIList
```

gCIList(CI=corporation-Identity)にはこのスタックで頻繁に使われる化粧ブランド名が入力される.
"recordTaskName"ハンドラでは, タスクスタック名(="Stage 1")を引き数に反応記録用外部ファイルにデータ (スタックの頭書き)を書き出す.

```
  put "花王ソフィーナ,資生堂,コーセー,カネボウ" into gCIList
  recordTaskName  the short name of this stack
```

```
  pass openStack
end openStack
```

カード１のスクリプト

```
on openCard
  global gMisc,gCIList

  put "0,0,0,0" into line 1
          of cd btn "next"
  put empty into gMisc

  ・・・
  send "move" to cd fld "ポーラ"
  ・・・

  pass openCard
end openCard
```

"gMisc"には openCard時のキャッチコピーフィールドのlocationが書き込まれる．これを利用してcloseCard時にフィールドをもとの位置に再配置する．ボタン"next"に"0,0,0,0"を書き込むのはこれまで同様カウンターとして利用するためである．

デモ部分

フィールド関係のスクリプト

キャッチコピーフィールドをmouseDownするとメッセージ"shiftCatchCopy"が発生する．それに対応するスクリプトが以下に示すものである．

```
on shiftCatchCopy
  set the partNumber of the target to the number of cd parts
  ・・・
  hitOrNot the short namr of target
  ・・・
end shiftCatchCopy
```

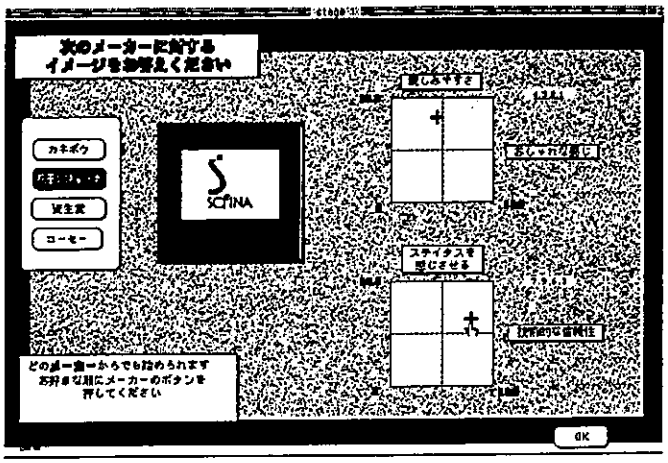
```
on hitOrNot what
  global gCIList

  if the mouseLoc is within
      the rect of cd btn what
  then
    get true
    renewNext(what)
  else get false
  set the hilite of cd btn what  to it

  if hitAll() is true and the enabled of cd btn "next" is false
```

"renewNext"では，フィールドとボタンがマッチしており,かつそれらがgCIListに含まれるものであるときボタン"next"に１を加える作業が行われている．"hitAll"関数では，ボタン"next"を参照してすべてのボタン，フィールドのマッチんグが行われたかどうかを判定している．行われていたら"true"をかえす．

```
  then send "showUp" to cd btn "next"
end hitOrNot
```

カード２について

このカードでは，化粧品メーカー，ブランド（”カネボウ，コーセー，花王ソフィーナ，資生堂”）についてのイメージの測定を行う．被験者は，どの銘柄からでもはじめる事ができるし，どの順序で答えても良い．被験者によってすべての銘柄に解答が行われて時点で"showUp"メッセージがボタン"OK"に送られる．ここで得られたデータはスタック"Stage 2"で使用される．



カード２のスクリプト

```
on openCard
  global gMisc,gPct

   . . .
  writeOwnRecord "","" -- in this bg
   . . .
end openCard
```

反応を記録するための前準備として"writeOwnRecord"でファイルに頭書きを作る．

ブランド選択ボタンのスクリプト

```
on Clselected myName
  global gPct,gNofScales


  disable the target
  set the hilite of
      the target to true
  send "showPct myName"
      to cd fld "wFrame"
  send "coverUncover" && "up"
      to cd btn "Cover"
  . . .
  . . .
  send "count myName" to cd btn "OK"
  resetScales
  enableScaleBtns true
end Clselected
```

選択されたボタンはdisableとなり，ハイライトとなる．

"showPct myName"によって選択されたブランドの商品写真のピクチャーウィンドウを作る．

"coverUncover"では，選択ボタン群を被験者から見えないようカバー（フィールドによって）している．

反応を記録するためのメッセージを発生した後ボタン"OK"にメッセージ"count myName"をおくる．これはボタンのカウンターに１を加えている．その為のハンドラは，”"OK"ボタンのスクリプト”（にある）を参照して欲しい．

次にスケールを初期状態に戻し("resetScales")そのスケールをenable(メッセージ"enableScaleBtns true"部分）に設定する．

ここで被験者への準備が終わる．

ボタン"OK"のスクリプト

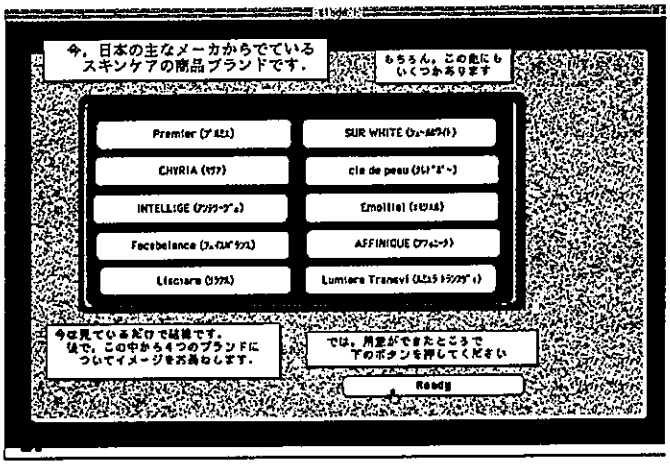
```
on count obj
  global gCIList

  put itemN(obj,gCIList) into N
  add 1 to item N of me
end count
```

関数"itemN(obj,list)"はスタック"utilities"に記述されているものである．"obj"が"list"の何番目のアイテムであるかを返してくる．

スケール関係のスクリプトはP.を参照

カード３について

このカードでは，被験者に対しスキンケアの商品ブランドを提示する．このカードのねらいはこのタスクスタックに続く"Stage 2"での測定にたいする注意の喚起である．提示する銘柄は日本国内のブランド，メーカーのもの１０種類である．被験者は，提示された情報を確認した後ボタン"READY"をクリックするよう指示される．被験者がクリックすると画面は"project home"スタックに戻る．



カード３のスクリプト

特筆すべきスクリプトが無いので省略する．

3－4 "Stage 2"スタックについて

"Stage 2"スタックは，スキンケア４銘柄（プルミエ，キリア，アンテリージェ，フェイスバランス）についてのイメージの測定を行うためのタスクスタックである．被験者は"Stage 1"でブランドのロゴと商品名を提示されたが，ここで続いて商品のロゴを確認する事になる．さらにカードが進むに連れモノクロの商品パッケージ，カラーのパッケージと商品特有の情報が増えていくよう設定されている．

使用される ScalingMethodは，"matching"，"1 dimensional hyper scaling"，"2 dimensional hyper scaling"の３種類である．12枚のカードで構成されている．

カード1-------宝石(12個）と商品ロゴとのマッチング
カード２ ------カード3 ～6での測定の導入部分
カード３～6 ---カード１の測定結果を初期設定としたイメージ測定
カード７ ------カード8 ～11での測定の導入部分
カード8 ～11 - -今までの測定結果を踏まえた上での最終測定
カード12 -----最後の画面

スタック，バックグラウンドスクリプトについて

このスタックの他タスクスタックとの相違は，データ用外部ファイルからのデータ読み込み作業があるという点である．カード１およびカード３からの測定で宝石のピクチャーデータ，イメージデータ（初期設定とするもの）を外部ファイルから取り込むので，それに必要なハンドラがスタック，バックグラウンドに書き込まれてある．

また，フィールドによるデータ管理手法の拡張の試みとしてカードの初期設定値，使用されるScaling Methodに関する情報等を各カードに設定した情報フィールドに書き込むという手法を取り入れている．このフィールドは，"cd attributes"（図3.4.1)と"Scaling Method"（図3.4.2)であり通常はinvisibleとなっている．"control"メニュー(P.10参照)の"Scaling Method","cd attributes"を選ぶとvisibleとなる．これに関するスクリプトも同様に書き込まれている．

NofScales: 10
Values: 0.0-10.0
Default values: 5.0
Stimulus type: PICT
Reset on close: no

図3.4.1

Scaling method
Point
Interval
Both
Point > Both
Interval > Both
Point > Interval

図3.4.2

スタックスクリプトについて

```
on openStack
  global gPCTFolder,gBIList

  . . .
  enable menu "Control"
  . . .
  put "jewelry photos:" into gPCTFolder

  pass openStack
end openStack
```

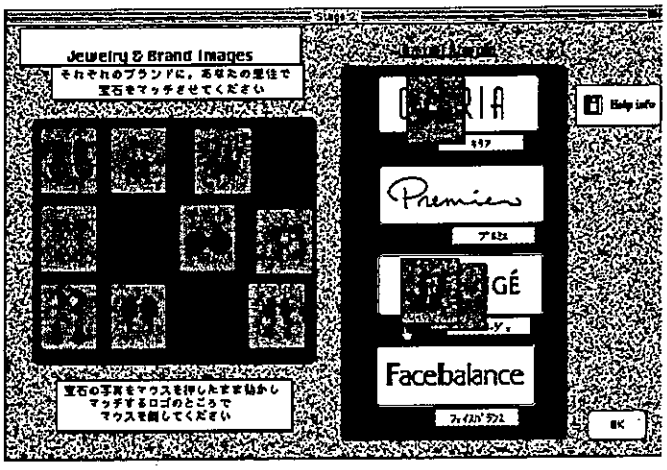
" enable menu "control"" でdisable だった"control"メニューをenableに設定している．また，このカードで使う宝石のピクチャーデータを読み込むための前準備として put "jewelry photos:" into gPCTFolder がある．

```
on openCard
  global gScalingMethod
  global gNofScales

  if there is a cd fld "Stimulus name"
  then
    put the selectedText of cd fld"scaling
method" into gScalingMethod
   end if
  pass openCard
end openCard
```

カードがオープンする時そのカードが測定用のカードかどうか判定し，そうであれば"Scaling Method"情報保持用フィールド" scaling method"からその情報（設定状態）をグローバル変数gScalingMethodに入れている．

カード１について

このカードではスキンケアのロゴと宝石とのマッチングを行ってもらう．ロゴは４銘柄，宝石は12種類である．宝石ピクチャーは画面左の黒枠の中，ロゴは右側に配置される．被験者は宝石ピクチャーをドラッグしてマッチングを行う．複数の宝石をロゴとマッチングさせる事も可能だが最後にクリックした宝石が測定値となる．被験者はボタン"infomaton"をクリックするとこのことを知るすべてのロゴに宝石がマッチングされた時点でボタン"OK"にメッセージ"showUp"が送られがenableとなる．



ここで測定されたデータは，フィールド"record"に書き込まれ他のカードから参照，共有される．なおフィールド"record"は通常invisibleになっており"control"メニューから"show Cd Fld Record"を選択するとvisibleとなる．また．被験者はボタン"infomaton"をクリックすると情報を得る事ができる．

カード1のスクリプト

```
on openCard
  global gMisc,gBlList

    . . .
   repeat with i=1 to the number of
                        items of gBlList
    put 0 into item i of cd btn "OK"
  end repeat
    . . .
  showGems
end openCard
```

ボタン"OK"に"0,0,0,…"を代入しているのは，カウンターとして利用するためである．
"showGems"では，宝石のピクチャーのリストを作成し"lineUpGems"でウィンドウを指定の場所に開いている．指定の場所に開く際に同名のフィールドのlocationを利用している．

宝石ピクチャーに対応すスクリプト

```
on  mouseDownInPicture  wName,xyClick
    . . .
   repeat until the mouse is up
    . . .
  gemAndBrand wName,the mouseLoc
end mouseDownInPicture
```

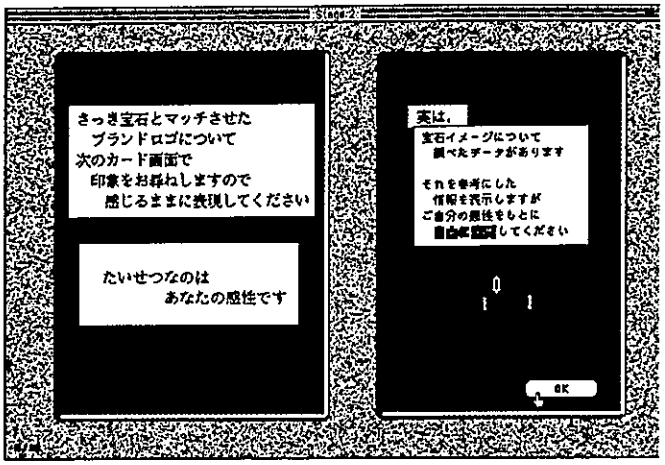
```
on gemAndBrand wName,xyUp
  global gBlList,gMisc
  put false into anyHit
  repeat with i=1 to
              the number of items of gBlList
    get item i of gBlList
    if xyUp is within the rect of cd btn it
    then
      blink the name of cd btn it,1,"1 ticks"
      put true into anyHit
      send "register wName,it,i" to cd btn "OK"
      exit repeat
    end if
  end repeat
   if anyHit is false and wName is in gMisc
  then send "unregister wName" to cd btn "OK"

end gemAndBrand
```

1. ピクチャーがロゴボタンの領域内でマウスアップされたか．真ならば2に進み，偽ならば4に進む
2.1 既に他のロゴにマッチングされていたかどうか
2.2 マッチングされていたらカウンターから1引く
3. マッチングされたロゴに対応するカウンターに1を加え終了
4. 既に他のロゴにマッチングされていたかどうか
5. マッチングされていたらカウンターから1引いて終了

以上

カード２について

このカードでは，これからの測定の説明が行われます．カードオープン時には画面左側の”さっき宝石と・・・”とかかれたフィールドと，画面右側の”実は，”とかかれたフィールドのみが表示されている．その後，はじめに右側の，次に左側のフィールドが表示されていく．すべてのフィールドが表示されるとボタン"OK"がenableとなる．

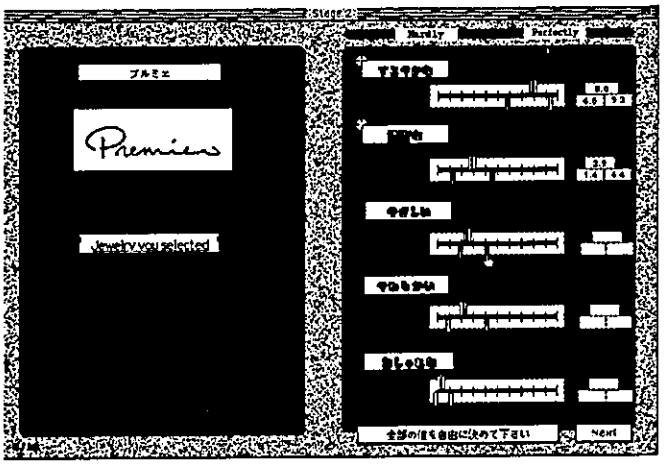


カード２のスクリプトについて

特徴的なスクリプトがないので省略する．

カード３～６について

このカードでは，化粧品（スキンケア）のイメージの測定を行う．測定される項目は，”すこやかな，清潔な，やさしい，やわらかい，おしゃれな”の５項目である．Scaling Methodは1 Dimensional Hyper Scalingである．画面左上方にロゴを下方にパッケージのピクチャー（但し被験者が"Jewelry you selected"の部分でマウスボタンを押している間だけ表示する)を，右側にはスケールを配置している．スケールの初期値は，宝石とのマッチングの結果を利用している．各宝石は，そのイメージを測定しておりデータ用外部ファイルとして用意されている．そこからデータを読み込みスケールのデフォルトとする．被験者がすべての項目に反応するとボタン"NEXT"がenableとなる．各スケールにたいする被験者の反応の有無はボタン"Mark"をカウンターとして利用して判定している．



カードスクリプト

```
on openCard
  global LimL,LimR,gStartTime
  global pctMvyName,gPCT

  put cd fld "Stimulus name" into brand
  showBIPCT brand
  placeGemPCT line 2 of cd fld "gem"

  nOfScales
  setDefaultVals brand
  setUpCounter
  resetThisCd cd fld "default record"

  put the ticks into gStartTime

  pass openCard
end openCard
```

"showBIPCT"では，対象となるスキンケアの商品名ロゴをピクチャーとして表示している．１行上で変数"brand"にその値が代入されている．

brandplaceGemPCT line 2 of cd fld "gem"では商品とマッチングされた宝石のピクチャーを用意している．しかし，この時点では表示されない．

nOfScalesではスケールの数をグローバル変数"gNOfScales"に代入します．

setDefaultVals brand では，変数"brand"を引き数として宝石とのマッチングの結果から用意されたデフォルト用の数値をフィールド"default"に書き込んでいる．

setUpCounter では，ボタン"NEXT","MARK 1,2,3,…"に"0,0,0,・・・"を代入し，hiliteをfalseにしている．

resetThisCd cd fld "default record"では，データ用フィールド"default record"からデータを読み込みスライドボタンをその位置に設定している．

このカードでは，"1 Dimensional Hyper Scaling"を用いて測定が行われるが，それについてのスクリプトは付録P.29を参照して欲しい．

ここでは，スクリプトの解読を助けるために，スクリプト中に出てくるフィールドについて述べておく．フィールドの位置は次のページ図3.4.2を参照して欲しい．

StimulusName --------対象となるスキンケアの商品名.
Mark 1 ----------スライドボタンが操作されたかどうかをを判別する
カウンターの役目
term 1 ----------項目名
point down 1 --------代表値をあらわす.
lower 1 ----------最低値をあらわす.
upper 1 ----------最高値をあらわす.

gem ----------クリックすると被験者がマッチングした宝石のピクチャーを表示する.
note----------注意書
default record -------マッチングから得られた値をここに格納しdefaultとして利用する.
response record------被験者の反応をここに格納し共有を可能とする.
wFrame----------商品名のロゴのピクチャーを表示するときその座標を与えるために使用する. これによって再構成や訂正が簡単になる. (スクリプトについては, P.31「付録-2 ピクチャーの表示について」を参照)
gemFrame----------マッチングした宝石のピクチャーを表示するときその座標を与えるために使用する. 上と同じ効果がある.

各フィールドの名前(隠れているものも含む)

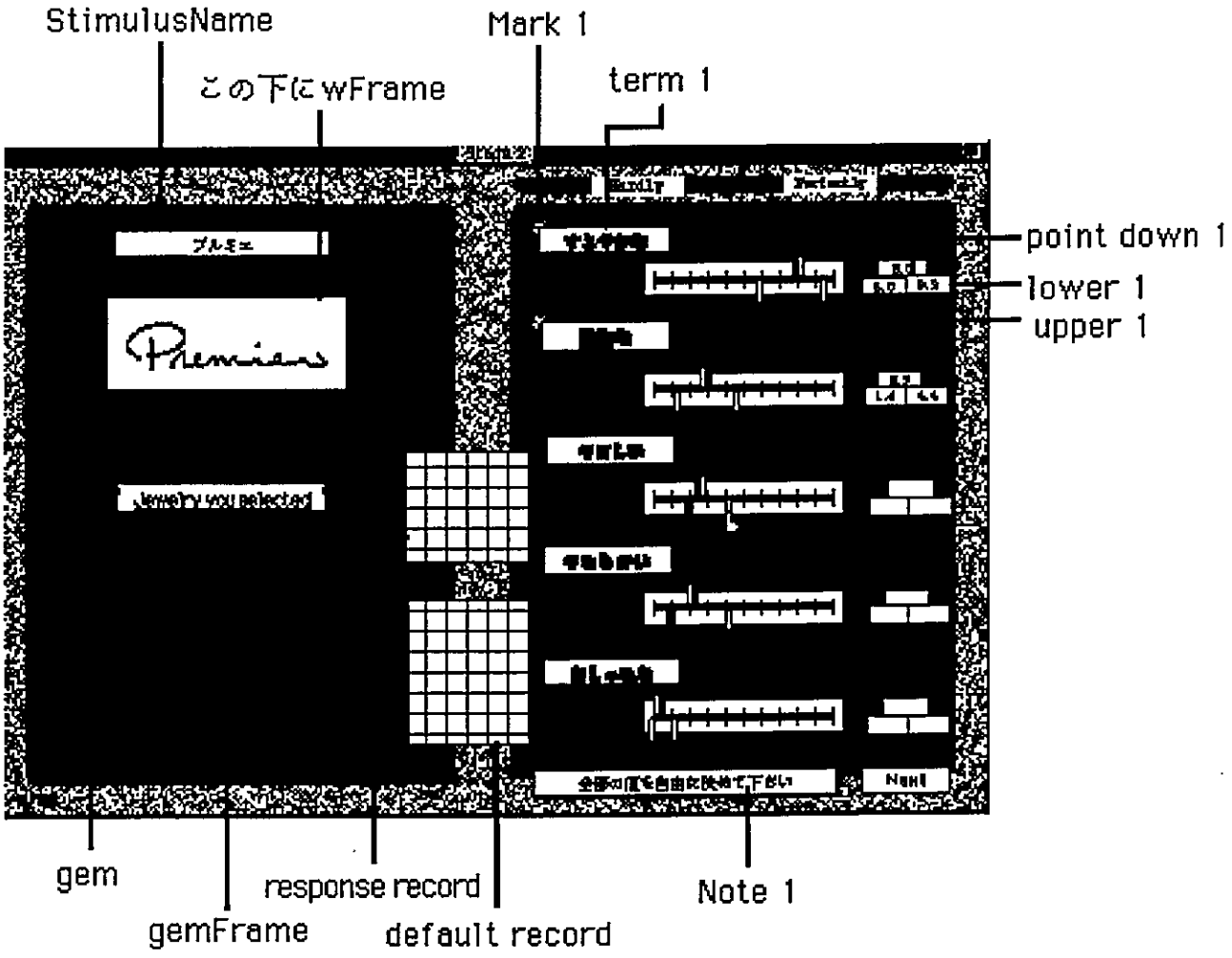


図3.4.3

カード７について

このカードでは，測定の対象となっている４銘柄のスキンケアのパッケージを被験者に提示する．しかしこの段階ではモノトーンのピクチャーを用いる．被験者へのインパクトを段階的にするためで，次のカードで個別にカラーのピクチャーを見せていく．

またこのカードには今までの測定記録を収集する記録用フィールドがあり，被験者の反応記録が整理されこの先の測定で利用される．

カードスクリプト

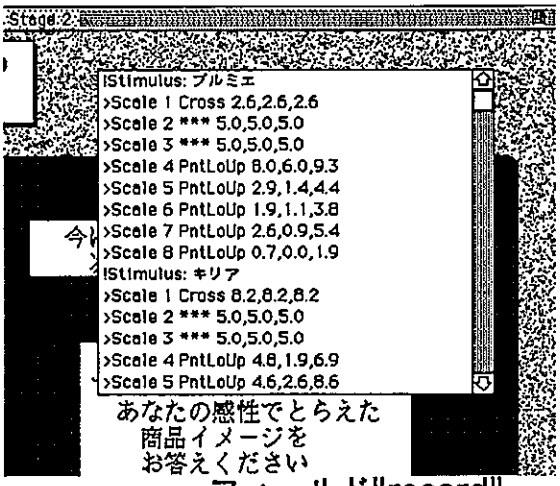
```
on openCard
  ・・・
  send "mouseUp" to cd fld "record"
  ・・・
  pass openCard
end openCard
```

カードフィールド "record" にメッセージ"mouseUp"を送ると今までの測定記録（被験者の反応記録）が収集されこの後の測定に利用するために再構成される．

ここでデータの再構成についても少し詳しく解説する．

フィールド"record"のスクリプト

```
on mouseUp
  pickupCIAnchors
  collectOwnBIRecord
  combineBoth
end mouseUp
```



フィールド"record"

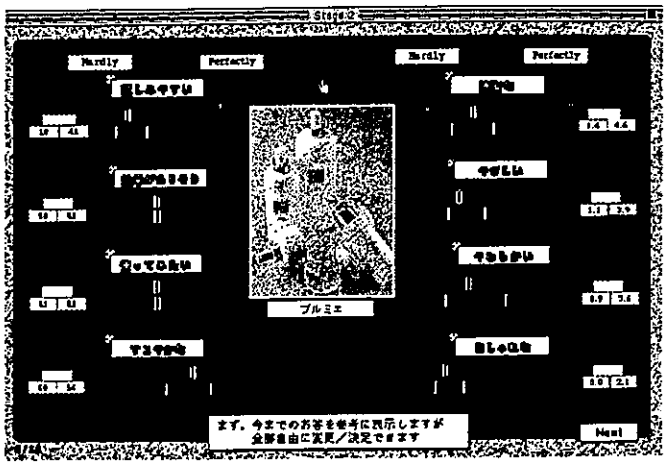
"pickupCIAnchors" では，"stege 1"で測定したメーカーへのイメージの結果を読み込んでそのうち必要なものだけを残している．

"collectOwnBIRecord "では，"stage 2"で測定した商品名ロゴに対するイメージの結果を読み込んでいる．

"combineBoth"では上の"pickupCIAnchors" "collectOwnBIRecord"で用意したデータをまとめフ・ールド"record"に書き出している．

カード8〜11について

このカードでは，スキンケアの商品イメージの総合的な測定（評価）を被験者におこなってもらう．被験者はここではじめて商品パッケージのカラーピクチャーを見ることになる．使用されるScaling Methodは"1 Dimensinal Hyper Scaling"である．測定項目は８項目（親しみやすい，効果がありそう，使ってみたい，すこやかな，清潔な，やさしい，やわらかい，おしゃれな）あり，このうち第1,4,5,6,7,8項目については，これまでの測定で得られた被験者の反応値をdefaultとして提示する．画面中央に商品パッケージのピクチャーを配置し，その左右に４項目づつスケールを配置した．すべての項目に反応が見られた時点でボタン"Next"にメッセージ"showUp"が送られenableとなり次のカードにすすめるようになる．

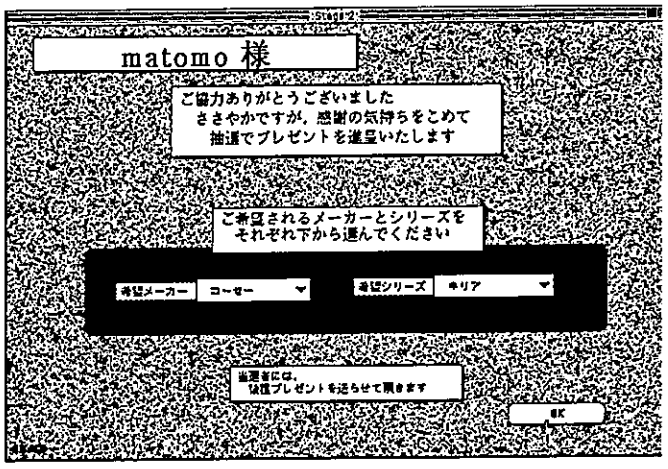


スクリプトについて

このカードでは他の"1 Dimensinal Hyper Scaling"を使用した測定用のカードと大差ないので省略する．

カード12について

このカードで測定は終了する．被験者はプレゼントの申し込みを行う．被験者が希望メーカー，シリーズを選ぶとボタン"OK"がenableとなる．これを押すと測定は終了しスタック"project home"に移る．



スクリプトについて

特筆すべきスクリプトがないので省略する．

以上で各カードについてのスクリプトの解説は終わりである．

付録　開発者のために

1　Hyper Scaling のスクリプト

Hyper Sclingのスクリプトを以下に示す.

```
"1 Dimensional"
 BUTTON: "PointDown"
 on mouseDown
  put "Point" && last word of the short name of me into obj
  recordStartTime obj  -- as fldName
  put (the mouseH - item 1 of the loc of me) into adjustH

  set the numberFormat to "0.0"
  slidePointDown adjustH,1
  recordFinishTime obj

  countMarker obj  -- cd script
 end mouseDown

 on mouseDoubleClick
  global gX,gStartTime

  put "time: " & timeNow() & return after me
 end mouseDoubleClick

 on showUp x,value
  put item 2 of the loc of me into y
  show me at x, y
  set the numberFormat to "0.0"
  put value into cd fld remove("Down",the short name of me)
  set the numberFormat to "0"
 end showUp

"2 Dimensional"
BUTTON"cross"
on mouseDown
 put the short name of me into myName
 recordStartTime myName
 set the numberFormat to "0.0"
```

```
  get the loc of me
  put the mouseH - (item 1 of it) into adjustH
  put the mouseV - (item 2 of it) into adjustV

  slideCross  adjustH,adjustV
  recordFinishTime  myName
  send "ShowUp" to cd btn "OK"
end mouseDown

on showUp xy,value
  show me at xy
  put  value into cd fld the short name of me
  --  blink the name of me,1,"10 ticks"
end showUp

FIELD"cross"
on mouseUp
  if the shiftKey is down
  then
    areaSize 120,120
    drawRectangle
    centerLines the loc of me
    choose browse tool
    set the autoHilite of me to false
  end if
end mouseUp

on areaSize wide,high
  set the hilite of me to false
  set the width of me to wide  -- must be an even number
  set the height of me to high
end areaSize

on drawRectangle
  choose the rectangle tool
  set lineSize to 1
  set pattern to 1
  set Filled to true
  drag from the topLeft of me to the bottomRight of me
end drawRectangle
```

```
on centerLines center
 choose the line tool
 brokenLine  the left of me,the right of me,item 2 of center,"a,b"
 brokenLine  the top of me,the bottom of me,item 1 of center,"b,a"
end centerLines

on brokenLine a,aLimit,b,ab
 repeat while a<aLimit
  add 2 to a   -- evry other pixel
  do "click at ab"  -- a,b  or b,a
 end repeat
end brokenLine
```

2　ピクチャーの表示について

ピクチャーの表示に関するスクリプトの例を幾つかを以下に示す．ピクチャーを思い通りの場所に提示するには，提示用のフィールドを利用するよい．

```
on showBIPCT brand
 put brand && "BI" into pctName
 if there is a window pctName
 then close pctName

 put "logo photos:" into folder
 picture fullName(folder & pctName),,shadow,false,,true
 put the width of window pctName into wide
 put the height of window pctName into high
 get the loc of cd fld "wFrame"
 add -(wide div 2) to item 1 of it
 add -(high div 2)  -14 to item 2 of it
 show window pctName at it
end showBIPCT

on placeGemPCT pctName  -- to be shown by appropriate action
 global gPct

 put line 2 of cd fld "gem" into gPct
 if there is a window gPct
 then close gPct
 put gPct into pctName

 put "gem photos:" into folder
 picture fullName(folder & pctName),,shadow,false,2,true
```

```
  put the width of window pctName into wide
  put the height of window pctName into high
  get the loc of cd fld "gemFrame"
  add -(wide div 2) to item 1 of it
  add -(high div 2)   to item 2 of it
  set the topLeft of window pctName to it
end placeGemPCT
```