

付録

各プログラム共通の注意点: “...” は行が続いていることを示す。そのため gedit など で編集する際は省いて記述する。

図形提示プログラム (Grating.m)

```
ListenChar(2);
AssertOpenGL;
%OS で共通のキー配置にする
KbName('UnifyKeyNames');

%%%%%%%%%図形提示プログラム
%縞模様が左に傾いていたら F, 右に傾いていたら J を押す

FlushEvents('keyDown');
Filename='result.dat'; % 結果は result.dat というファイルに出力される

%変数
ITI=1;
NofTr=10; %試行数
oriCond = [1 2 1 2 1 2 1 2 1 2]; %1: 右, 2: 左

%提示画面の設定
whichScreen = max(Screen('Screens'));
[window, ScreenRect] = Screen('Openwindow', whichScreen);
[FontName, FontNumber] = Screen('TextFont', window, 'Takao Pゴシック');

%刺激の大きさと注視点の設定
defRect=CenterRect(SetRect(0,0,200,200), ScreenRect);
fixRect=CenterRect(SetRect(0,0,3*2,3*2), ScreenRect);

%色, 背景色の指定
black = BlackIndex(window);
white = WhiteIndex(window);
gray = (black + white) / 2;
inc=white-gray;

%縞刺激: ow1 と ow2 を刺激として用いる
[x,y]=meshgrid(-100:100,-100:100);
ori=[30 -30];
for i = 1:2;
    ow(i)=Screen('OpenOffScreenWindow', window, gray);
    angle=ori(i)*pi/180;
    f=0.03*2*pi;
    a=cos(angle)*f;
    b=sin(angle)*f;

    m=exp(-(x/50).^2)-(y/50).^2).*sin(a*x+b*y);
    Screen('PutImage', ow(i), gray+inc*m, defRect);
end
```

```

%マウスを隠す
HideCursor;

%結果ファイルを開く
fid=fopen(Filename,'a');

%教示画面
Screen('FillRect', window, gray);
Screen('DrawText', window, [' 縞模様が左に傾いていたら "F", ' ...
    ' 右に傾いていたら "J" を押す。'], 330, 300, black);
Screen('DrawText', window, 'Click to start.', 330, 330, black);
Screen('Flip', window);

%クリックを取得したら画面を消す
GetClicks; %クリックの取得
Screen('FillRect', window, gray);
Screen('Flip', window);

trCond=[randperm(NofTr)]; %trCond: ランダム変数
WaitSecs(ITI*2); %ITI*2 秒待つ

%刺激提示・反応取得
for i=1:NofTr; %NofTr 試行繰り返す
    Screen('WaitBlanking', window); %モニタとの同期
    Screen('PutImage', window, white,fixRect); %注視点の提示
    Screen('Flip', window);
    Screen('WaitBlanking', window, 50);
    keyIsDown = 0; %キー情報の消去

%刺激の提示: oriCond は 1 or 2, なので, trCond 変数によってランダムにする
    %defRect の位置に提示
    Screen('CopyWindow', ow(oriCond(trCond(i))), window, defRect, defRect);
    Screen('Flip', window);
    %StartSecs: 反応時間の計測開始
    StartSecs = GetSecs;
    Screen('WaitBlanking', window, 10); %刺激を 10 フレーム提示
    Screen('FillRect', window, gray);
    Screen('Flip', window);

    %反応の取得, 制限時間 1500ms
    while keyIsDown == 0 & GetSecs<StartSecs+1.5;
        [keyIsDown, t, keyCode] = KbCheck;
    end;

%反応時間 (ms) とするため 1000 倍にする
    ReactionTime=(GetSecs - StartSecs)*1000;

%キー入力が無かった場合, 反応は-になる
    ans='-';

%反応の正誤判断
    if KbName(keyCode) == 'f';
        if oriCond(trCond(i)) == 1;
            ans = 'x'
        else
            ans = 'o'
        end;
end;

```

```

%反応の正誤判断
if KbName(keyCode) == 'j';
    if oriCond(trCond(i)) == 2;
        ans = 'x'
    else
        ans = 'o'
    end;
end;

%反応の記録
%d: 整数, %f: 浮動小数点, %c: 文字
fprintf(fid, '%2d \t %2d \t %4.0f \t %c \n', i, oriCond(trCond(i)), ...
        ReactionTime, ans);
end;

%終了の教示
Screen('DrawText', window, 'Thank you.', 330, 300, black);
Screen('Flip', window);

%マウスクリックを取得すると実験終了
GetClicks;
fclose(fid);
Screen('CloseAll');
clear all;

```

文字提示プログラム (Letter.m)

```

ListenChar(2);
AssertOpenGL;
%OS で共通のキー配置にする
KbName('UnifyKeyNames');

%%%%%%%%文字提示プログラム
%左が出たら F, 右が出たら J を押す

FlushEvents('keyDown');
Filename='result.dat'; % 結果は result.dat というファイルに出力される

%変数
ITI=1;
NofTr=10; %試行数
oriCond = [1 2 1 2 1 2 1 2 1 2]; %1: 右, 2: 左

%提示画面の設定
whichScreen = max(Screen('Screens'));
[window, ScreenRect] = Screen('Openwindow', whichScreen);
[FontName, FontNumber] = Screen('TextFont', window, 'Takao Pゴシック')

%刺激の大きさと注視点の設定
fixRect=CenterRect(SetRect(0,0,3*2,3*2), ScreenRect);

```

```

%色, 背景色の指定
black = BlackIndex(window);
white = WhiteIndex(window);
gray = (black + white) / 2;
inc=white-gray;

%文字刺激: ow1 と ow2 を刺激として用いる
ow1=' みぎ';
ow2=' ひだり';

%マウスを隠す
HideCursor;

%結果ファイルを開く
fid=fopen(Filename,'a');

%教示画面
Screen('TextSize', window, 28);
Screen('FillRect', window, gray);
Screen('DrawText', window, ['"ひだり" が出たら "F", ' ...
                              '"みぎ" が出たら "J" を押す。'], 330, 300, black);
Screen('DrawText', window, 'Click to start.', 330, 330, black);
Screen('Flip', window);

%クリックを取得したら画面を消す
GetClicks; %クリックの取得
Screen('FillRect', window, gray);
Screen('Flip', window);

trCond=randperm(NofTr); %trCond: ランダム変数
WaitSecs(ITI*2); %ITI*2 秒待つ

%刺激提示・反応取得
for i=1:NofTr; %NofTr 試行繰り返す

    Screen('WaitBlanking', window); %モニタとの同期
    Screen('PutImage', window, white,fixRect); %注視点の提示
    Screen('Flip', window);
    Screen('WaitBlanking', window, 50);
    keyIsDown = 0; %キー情報の消去

%刺激提示の準備: oriCond は 1 or 2, なので, trCond 変数によってランダムに
する

    txWidth=0;
    if oriCond(trCond(i))==1;
        targ=ow1;
    else;
        targ=ow2;
    end;

```

```

%文字幅を取得する処理
txBox=Screen('TextBounds', window, targ);
txWidth=RectWidth(txBox);

%刺激の提示 1024 x 768 pixels
Screen('DrawText',window, targ, 1024/2-txWidth/2, 768/2-10,black);
Screen('Flip', window);
%StartSecs: 反応時間の計測開始
StartSecs = GetSecs;
Screen('WaitBlanking', window, 10); %刺激を 10 フレーム提示
Screen('FillRect', window, gray);
Screen('Flip', window);

%反応の取得, 制限時間 1500ms
while keyIsDown == 0 & GetSecs<StartSecs+1.5;
    [keyIsDown, t, keyCode] = KbCheck;
end;

%反応時間 (ms) とするため 1000 倍にする
ReactionTime=(GetSecs - StartSecs)*1000;

%キー入力が無かった場合, 反応は-になる
ans='-';

%反応の正誤判断
if KbName(keyCode) == 'f';
    if oriCond(trCond(i)) == 1;
        ans = 'x'
    else
        ans = 'o'
    end;
end;

%反応の正誤判断
if KbName(keyCode) == 'j';
    if oriCond(trCond(i)) == 2;
        ans = 'x'
    else
        ans = 'o'
    end;
end;

%反応の記録
%d: 整数, f: 浮動小数点, c: 文字
fprintf(fid, '%2d \t %2d \t %4.0f \t %c \n', i,
oriCond(trCond(i)), ReactionTime, ans);
end;

```

```

%終了の教示
Screen('DrawText', window, 'Thank you.', 330, 300, black);
Screen('Flip', window);

%マウスクリックを取得すると実験終了
GetClicks;
fclose(fid);
Screen('CloseAll');
clear all;

```

画像提示プログラム (Picture.m)

```

ListenChar(2);
AssertOpenGL;
%OS で共通のキー配置にする
KbName('UnifyKeyNames');

%%%%%%%%%画像提示プログラム
%鳥が提示されたら F, 虎が提示されたら J を押す

FlushEvents('keyDown');
Filename='result.dat'; % 結果は result.dat というファイルに出力される

%変数
ITI=1;
NofTr=10; %試行数
oriCond = [1 2 1 2 1 2 1 2 1 2];

%提示画面の設定
whichScreen = max(Screen('Screens'));
[window, ScreenRect] = Screen('Openwindow', whichScreen);
[FontName, FontNumber] = Screen('TextFont', window, 'Takao Pゴシック');

%刺激の大きさと注視点の設定
defRect=CenterRect(SetRect(0,0,800,600), ScreenRect);
fixRect=CenterRect(SetRect(0,0,3*2,3*2), ScreenRect);

%色, 背景色の指定
black = BlackIndex(window);
white = WhiteIndex(window);
gray = (black + white) / 2;
inc=white-gray;

```

```

%画像の読み込み ow1: 虎 (pict1.jpg), ow2: 鳥 (pict2.jpg)
%pict1: http://upload.wikimedia.org/wikipedia/commons/thumb/4/41/
% Siberischer_tiger_de_edit02.jpg/800px-Siberischer_tiger_de_edit02.jpg
%pict2: http://upload.wikimedia.org/wikipedia/commons/thumb/7/7d/
% Eagle_Owl_IMG_9203.JPG/120px-Eagle_Owl_IMG_9203.JPG

for i =1:2;
    stmname=['pict' num2str(i) '.jpg'];
    yy = imread(stmname, 'jpg');
    ow(i)=Screen('OpenOffscreenWindow', window, gray);
    Screen('PutImage', ow(i), yy, defRect);
end

%マウスを隠す
HideCursor;

%結果ファイルを開く
fid=fopen(Filename,'a');

%教示画面
Screen('FillRect', window, gray);
Screen('DrawText', window, [' 鳥が現れたら "F", ' ...
                             ' 虎が現れたら "J" を押す。'], 330, 300, black);
Screen('DrawText', window, 'Click to start.', 330, 330, black);
Screen('Flip', window);

%クリックを取得したら画面を消す
GetClicks; %クリックの取得
Screen('FillRect', window, gray);
Screen('Flip', window);

trCond=randperm(NofTr); %trCond: ランダム変数
WaitSecs(ITI*2); %ITI*2 秒待つ

%刺激提示・反応取得
for i=1:NofTr; %NofTr 試行繰り返す
    Screen('WaitBlanking', window); %モニタとの同期
    Screen('PutImage', window, white, fixRect); %注視点の提示
    Screen('Flip', window);
    Screen('WaitBlanking', window, 50);
    keyIsDown = 0; %キー情報の消去

%刺激の提示: oriCond は 1 or 2, なので, trCond 変数によってランダムにする
    %defRect の位置に提示
    Screen('CopyWindow', ow(oriCond(trCond(i))), window, defRect, defRect);
    Screen('Flip', window);
    %StartSecs: 反応時間の計測開始
    StartSecs = GetSecs;
    Screen('WaitBlanking', window, 10); %刺激を 10 フレーム提示
    Screen('FillRect', window, gray);
    Screen('Flip', window);

```

```

%反応の取得, 制限時間 1500ms
while keyIsDown == 0 & GetSecs<StartSecs+1.5;
    [keyIsDown, t, keyCode] = KbCheck;
end;

%反応時間 (ms) とするため 1000 倍にする
ReactionTime=(GetSecs - StartSecs)*1000;

%キー入力が無かった場合, 反応は-になる

ans='-';

%反応の正誤判断
if KbName(keyCode) == 'f';
    if oriCond(trCond(i)) == 1;
        ans = 'x'
    else
        ans = 'o'
    end;
end;

%反応の正誤判断
if KbName(keyCode) == 'j';
    if oriCond(trCond(i)) == 2;
        ans = 'x'
    else
        ans = 'o'
    end;
end;

%反応の記録
%d: 整数, f: 浮動小数点, c: 文字
fprintf(fid, '%2d \t %2d \t %4.0f \t %c \n', i, oriCond(trCond(i)), ...
        ReactionTime, ans);
end;

%終了の教示
Screen('DrawText', window, 'Thank you.', 330, 300, black);
Screen('Flip', window);

%マウスクリックを取得すると実験終了
GetClicks;
fclose(fid);
Screen('CloseAll');
clear all;

```