

Today Is Our Last Meeting!

- Exam Solutions and Discussion
- Excel

Exam 2

- Overall, great improvement in the class. Congrats!
- Two sections and two types of questions:
 - understand vs. write code
 - apply vs. extend knowledge

Q1:

Sequence of events in the experiment:

- 1.- Fixation screen for fixtime: .5 seconds.
- 2.- Search Screen for 1 refresh "c times" = ontime x 11.7 ms \approx 100ms.

```
c=0;
```

```
while ~KbCheck & (c < ontime),  
    Screen('DrawTexture',window,search,[],[]);  
    Screen('Flip',window);  
    c=c+1;  
    s=s+1;  
end;
```

.

Q1:

Sequence of events in the experiment:

- 1.- Fixation screen for fixtime: .5 seconds.
- 2.- Search Screen for 1 refresh "c times" =
ontime x 11.7 ms \approx 100ms.
- 3.- Fixation Screen for 1 refresh "c times" =
offtime x 11.7 ms \approx 900ms.
- 2&3 REPEAT until a response is made OR
s=totaltime number of refreshes have
elapsed =10s.
4. Blank screen for ITI (~1.5 seconds).

WHO DID NOT RUN THE PROGRAM?

Q2:

2.1 cx cy: global to be used in both script and grid.m

Alternative?

Declare grid with an output:

```
[cx cy]=grid(i);
```

2.2 Cx and cy represent coordinates of upper left corner of each cell (where to draw stimuli).

Q3: colors

Everyone got this right.

-> Just remember to declare variables BEFORE you use them.

Q4: Set Size

4.1 Change setsize
variable:

setsize=[16,32]; to

setsize=[4,8];

Q4: Set Size

4.2 Change setsize variable
AND number of trials:

setsize=[16,32]; to

setsize=[8,16,32];

```
for i=1:T_TRIALS
    DUMMY(i).setsize=setsize( fix((i-1)/(T_TRIALS/numel(setsize)))+1);
    DUMMY(i).tid=mod(i,2)+1;      %target id codes target color
end;
```

How many noticed the dynamic
allocation of setsize in the
design?

Q4: Set Size

1: T_Trials (imagine T_Trials =9)

[1 2 3 4 5 6 7 8 9]

so: (i-1) is:

[0 1 2 3 4 5 6 7 8]

numel(setsize) = number of levels of variable setsize, 3, (8, 16 & 32)

so $T_Trials / \text{numel}(\text{setsize}) = 9 / 3 = 3.$

so **(i-1)/(T_TRIALS/numel(setsize))**

[0 0.33 0.66 1 1.33 1.66 2 2.33 2.66]

so that rounded towards zero (fix) +1:

[1 1 1 2 2 2 3 3 3]

Q4: Set Size

So: if `T_trials` is a multiple of the number of levels of setsize, `n`, this little manipulation divides the vector `[1:T_TRIALS]` into `n` groups of same size and increasing from 1 to `n`:

Ex: `T_TRIALS = 60, setsize=[8, 16]`

--> 1-30 trials: level 1

--> 31-60 trials: level 2

`T_TRIALS = 60, setsize=[8, 16, 32]`

--> 1-20 trials: level 1

--> 21-40 trials: level 2

--> 41-60 trials: level 3

BUT: `T_TRIALS = 30, setsize=[4,8,16,32]`

--> 1-8 trials: level 1

--> 9-16 trials: level 2

--> 17-24 trials: level 3

--> 25-30 trials: level 4 ---> Unbalanced design.

Q5: Grid

Spatially GRID is arranged according to the commented description in the grid.m program:

```
%  1  2  3          10 11 12
%  4  5  6          13 14 15
%  7  8  9          16 17 18
%
%              +
% 19 20 21          28 29 30
% 22 23 24          31 32 33
% 25 26 27          34 35 36
```

In addition, we add an offset (from the upper left corner) and a random amount of jitter to each item (to avoid collinearities).

```
tempy=cy+(round(rand(1)*jitter))+
offset;
```

Q5: Grid

PLUS: we randomly select locations:

Create an array of zeros:

[0 0 0 0 ... 0] one for each location

turn the first SS minus 1 into Ones (for Distractors)
and SStth item into a 2 (for target)

[1 1 1 ... 1 1 2 0 0 0 0 0 0]

And randomly permuted:

[0 2 0 1 1 0 0 .. 1 0]

Q6: Timing

6.1: Exam booklet is better at RT: Checking the KB is the "only" thing we do...

```
while (t3-t2 < ontime) & (breakflag==0),  
    [touch, secs, keyCode] = KbCheck;  
    t3=secs;  
    if touch  
        breakflag=1;  
        stoptime=secs;  
    end;  
end;  
• while ~KbCheck & (c < ontime),  
•     Screen('DrawTexture',window,search,[],[]);  
•     Screen('Flip',window);  
•     c=c+1;  
•     s=s+1;  
• end;
```

WHO GOT THIS RIGHT?

Q6: Timing

6.2 Script version is 'slightly' better at timing events because we are synchronized to the monitor's behavior, natural progression of monitor events.

Better still without KbCheck.

- while ~**KbCheck** & (c < ontime),
- Screen('DrawTexture',window,search,[],[]);
- Screen('Flip',window);
- c=c+1;
- s=s+1;
- end;

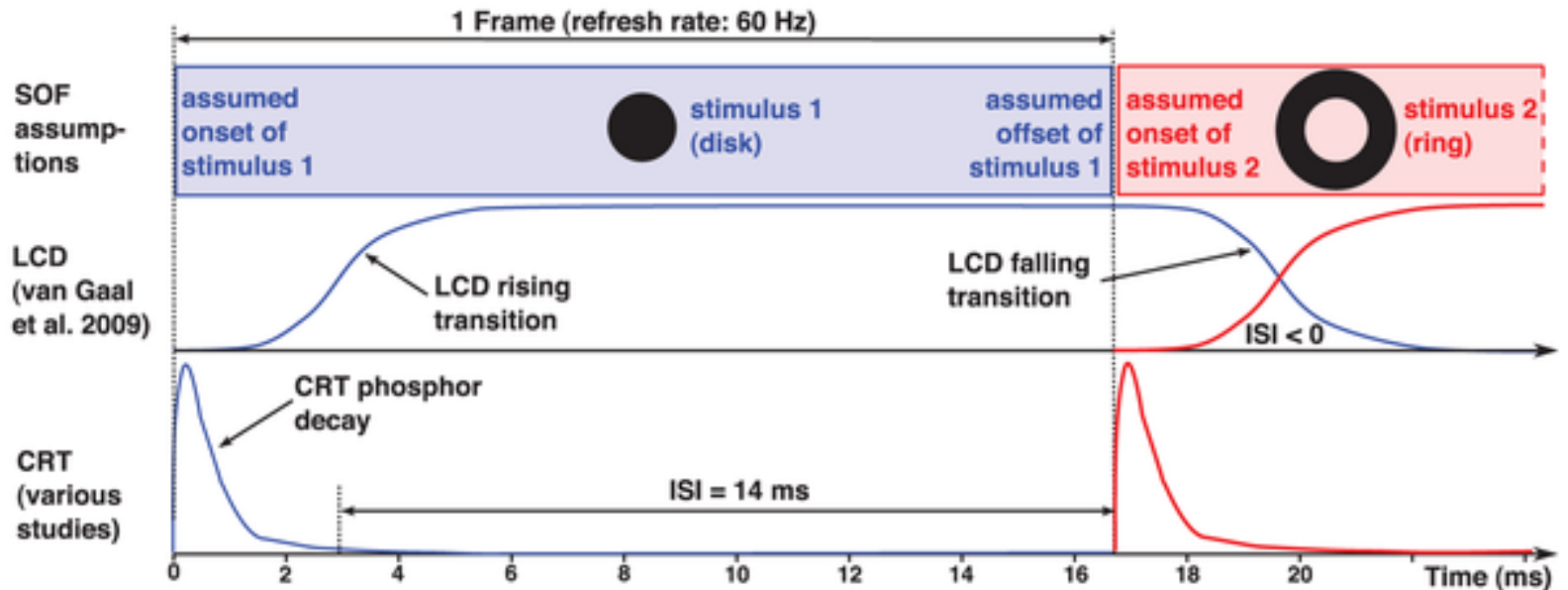
We dynamically adapt to the behavior of the monitor making sure the same number of "flashes" of information appear to the participants...

(The RT based is Ok, as long as your system does NOT fluctuate much and you have a good slack).

Parenthesis:

This is the REAL WORLD

Parenthesis:
This is the REAL WORLD



What's more important to you?
Reality or Perceived Illusions...

Writing code:

Solution: Copy-and-paste the tee.m function and delete the conditionals...

```
function mask(wp,x,y);
    %wp=window pointer:which image to draw 'T' on
    %x and y are coordinates for top-left corner of letter

    lgth=20;
    xpos=x;
    ypos=y;
    width=3;
    height=7; %not used
    colvalue = [0 0 0];

    screen(wp,'DrawLine',colvalue,xpos+(lgth/2),ypos,xpos+(lgth/2),ypos
        +lgth,width,height);
    screen(wp,'DrawLine',colvalue,xpos,ypos+lgth,xpos+lgth,ypos+lgth,width,height);
    screen(wp,'DrawLine',colvalue,xpos,ypos,xpos,ypos+lgth,width,height);
    screen(wp,'DrawLine',colvalue,xpos,ypos+(lgth/2),xpos+lgth,ypos+(lgth/
        2),width,height);
screen(wp,'DrawLine',colvalue,xpos+(lgth/2),ypos,xpos+(lgth/2),ypos
    +lgth,width,height);
    screen(wp,'DrawLine',colvalue,xpos,ypos,xpos+lgth,ypos,width,height);
    screen(wp,'DrawLine',colvalue,xpos+lgth,ypos,xpos+lgth,ypos+lgth,width,height);
screen(wp,'DrawLine',colvalue,xpos,ypos+(lgth/2),xpos+lgth,ypos+(lgth/
    2),width,height);
```

(bold lines are repeated.. could be taken off).

IF YOU DID NOT UNDERSTAND THE QUESTION, WHY DID YOU NOT ASK???

Writing code:

Stimuli:

1. Create a mask window:

- `maskw=Screen('MakeTexture',window,white_background);`

2. Draw on it:

```
Screen('DrawLine', maskw,black,Xcentre-9,Ycentre,Xcentre+9,Ycentre);
```

```
Screen('DrawLine', maskw,black,Xcentre,Ycentre-9,Xcentre,Ycentre+9);
```

And: add masks to it:

Inside of the "populating function", to match exact x,
y coordinates of items:

Right after

```
ell(search,tempcol,tempor,tempx,tempy);
```

```
mask(maskw,tempx,tempy);
```

and after tee:

```
tee(search,DATA(itrial).tid,tempor,tempx,tempy);
```

```
mask(maskw,tempx,tempy);
```

Writing code:

Trial events: copy-and-paste search display code and replace with masking window:

```
c=0;
• while ~KbCheck & (c < ontime),
•     Screen('DrawTexture',window,search,[],[]);
•     Screen('Flip',window);
    c=c+1;
    s=s+1;    %don't forget, we're counting total time
end;
```

Writing code:

AND DON'T FORGET TO CLOSE YOUR MASK SCREEN on each trial! (or you will get nasty memory effects).

```
Screen(maskw, 'Close');
```

Writing code:

7.3 DESIGN: DON'T CONFOUND VARIABLES!

```
for i=1:T_TRIALS
```

```
    DUMMY(i).setsize=setsize(fix((i-1)/(T_TRIALS/numel(setsize)))+1);
```

```
    DUMMY(i).tid=mod(i,2)+1;
```

```
    DUMMY(i).mask=mod(i,2)+1;    %CONFOUNDS TID AND MASK.
```

```
end;
```

Solution:

```
for m=1:2
```

```
    for t=1:2
```

```
        for ss=1:numel(setsize)
```

```
            for rep=1:whatever
```

```
%such that T_Trials=whatever x numel(ss) x 2 x 2
```

```
                dummy(i).setsize= ss;
```

```
                dummy(i).tid=t;
```

```
                dummy(i).mask=m;
```

```
            end;end;end;end;
```

OR: come-up with another "shortcut" formula.

Writing code:

7.4 An if or switch statement would do, just try to
always do those OUTSIDE time sensitive pieces of code...
so, if a masked trial,
 run this entire sequence of events,
if not
 run this other sequence of events.

Download data:

<https://netfiles.uiuc.edu/alleras/www/>

launch Matlab.

Analyzing Data:

1. Matlab:

Done: You can easily export/import the vector with RTs and operate on it in any possible mathematical way.

I do concatenate my data on Matlab. Remember Concat.m?

2. Today, Excel:

Why: Quick, reliable, graphics...

Analyzing Data:

1. On Matlab:

Run `Concat_nozero.m` to Concatenate the
6 data files (all in same directory).

2. Open overall file with Excel.

2.1 Click on column label A

2.2 Data -> Text_to_Columns

2.3 Space as delimiter.

3. Now you are ready!

Navigating cells:

Select the top cell with data.

- HOLDING Ctrl key down, push the down arrow key, right arrow key, etc...

- > takes you to the end of CONTIGUOUS cells.

- HOLDING Shift key down: selects cells

- Push both together to select the WHOLE set of DATA!!!

Sorting DATA

-Data-> Sort: select the order by which to organize your data.

Examples:

- Eliminate practice trials (block =-1).
- Eliminate first three trials of every block (warm-up trials).

Pivot Tables

Let's calculate accuracy in this task,
per condition, per subject.

- Go to overall data.

- Select it all (Ctrl+Shift: down and left).

- Insert -> Pivot Table

 - Select first option.

 - Next

ExerciseRT.xls [Compatibility Mode] - Microsoft Excel

PivotTable Tools

File Home Insert Page Layout Formulas Data Review View Options Design

PivotTable Name: PivotTable1 Active Field: Field Settings

Options PivotTable Active Field Group Sort & Filter Refresh Change Data Source Data Clear Select Move PivotTable Actions Summarize Values By Show Values As Fields, Items, & Sets Calculations PivotChart OLAP Tools What-If Analysis Tools Field List +/- Buttons Field Headers Show

A3

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	error	(All)												
2														
3														
4	Subject													
5	1													
6	2													
7	4													
8	5													
9	6													
10	7													
11	8													
12	9													
13	10													
14	11													
15	12													
16	13													
17	14													
18	15													
19	16													
20	17													
21	18													
22	19													
23	20													
24	Grand Total													
25														
26														
27														
28														
29														
30														
31														
32														
33														
34														
35														
36														
37														
38														
39														
40														
41														
42														
43														
44														
45														
46														
47														
48														

Drop Column Fields Here

Drop Value Fields Here

PivotTable Field List

Choose fields to add to report:

- ☒ Subject
- ☐ block
- ☐ trial
- ☐ targ loc
- ☐ targ ori
- ☐ disp_con
- ☐ targ col
- ☐ resp
- ☐ RT
- ☒ error
- ☐ trial code

Drag fields between areas below:

Report Filter: error

Column Labels:

Row Labels: Subject

Values:

Defer Layout Update Update

Oops! Forgot a factor!

Drag and drop it from the menu into the
Pivot Table!

Or Hide it

By selecting one in the table and right click and pick Hide.

FORMULAS in Excel

Cell Address

relative: B2

Absolute Column: \$B2

Absolute Row: B\$2

Absolute BOTH: \$B\$2

FORMULAS in Excel

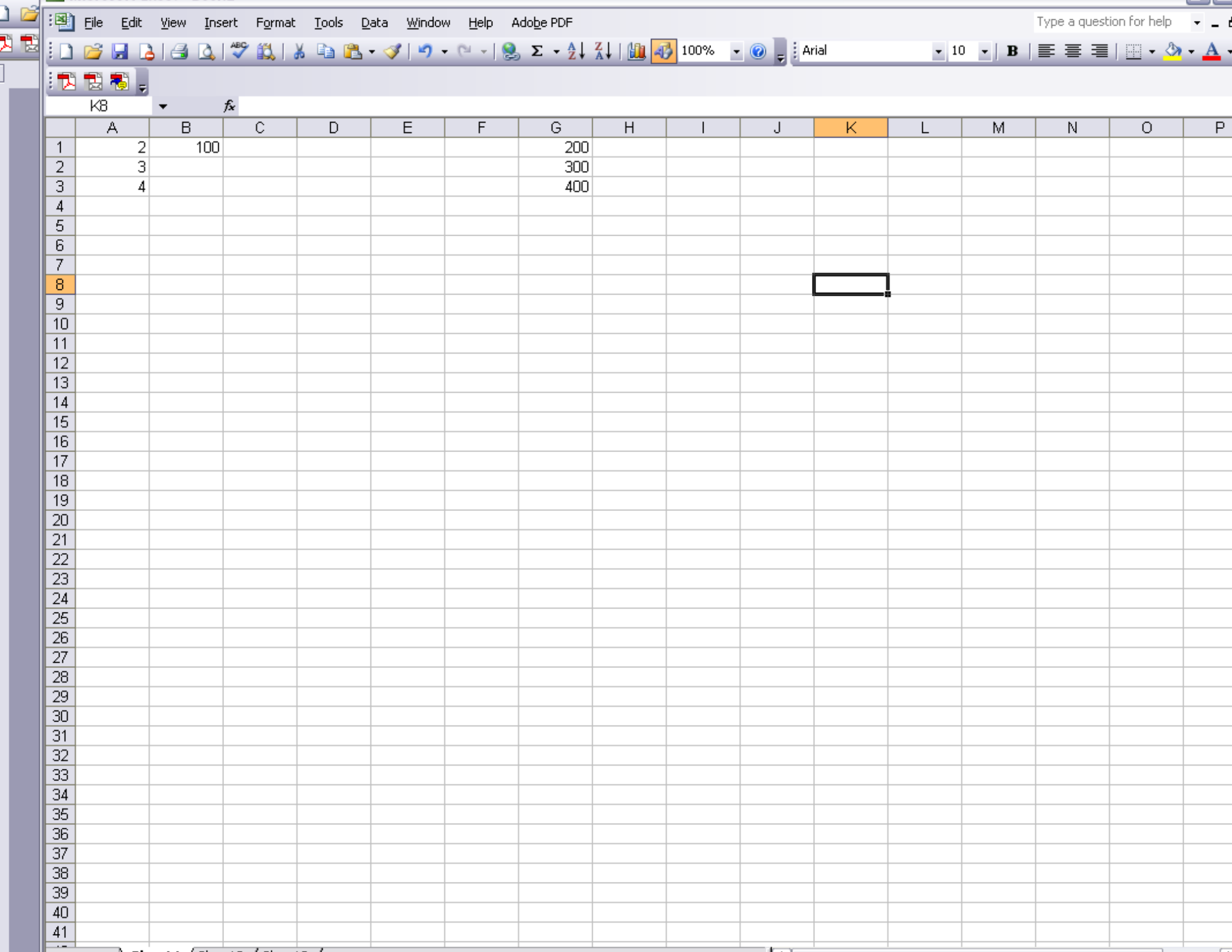
If Cell A1 had the formula:

=G1 / (B\$1)

Then copying this formula to cell A2
would result in:

=G2 / B\$1

when filling-down, index gets updated
UNLESS absolute \$.



FORMULAS in Excel

If Cell A1 had the formula:

=A5/(\$A8)

Then copying this formula to cell
B1 would result in:

=B5/\$A8

when moving rightwards, LETTER
gets updated UNLESS absolute \$.



B1 fx =B5/\$A8

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	2	3	4												
2															
3															
4															
5	200	300	400												
6															
7															
8	100														
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															

Another useful shortcut

Command-D:

Fills-down a selection of cells.

Multi-tasking!

Open file Exercise RT.

-Divide data into three worksheets:
full set, correct RT, incorrect RTs and
delete trial code =3.

Now you are ready to multitask:

- Go to worksheet with correct RTs
- While holding Ctrl key down, select the worksheet with incorrect RTs.
- > until you unselect the second worksheet, everything you do on one sheet will be repeated on the second (third, fourth...) worksheet!

Rushing code

So, we not always have instant access to the processor (other tasks in the waiting cue).

- > Close any windows programs while running your experiments
- > You can also "rush" or "prioritize" the thread of your code.

Rushing your code

Different operations have different **priorities** in your computer.

- > checking status of Kb.

- > make sure to be on time for display refresh...

“Priority” allows us to alter the priority of a piece of code.

Rushing your code

Windows process priorities:

0: like any other task...

1: high priority level:

2: real time priority level.

process priority + Thread priority
= real priority.

"Round Robin" fashion: low
priorities get the crumbs...
(OSX has 9 priority levels)

Rushing your code

About high priority:

- Blocks Time Manager: Microseconds function (used in GetSecs) advances at coarser steps 0.3ms (rather than 0.02ms) and might overflow...

Though my personal tests have failed to show that: in XP, as many as 2000000 GetSecs failed to overflow.

Rushing your code

About real time priority:

- blocks keyboard input (no Ctr-Alt-Delete!)
- but keystrokes (and mouse clicks) are sent to the Kb queue (and event queue), so can be read by KbCheck, GetClicks IF THESE functions ARE also RUN AT PRIORITY 2!

Priority

```
call:Priority(newpriority)
```

```
[window,rect]=Screen('OpenWindow',2,0);
```

```
refresh=Screen('GetFlipInterval',window);
```

```
load durer;
```

```
durer=Screen('MakeTexture',window,X);
```

```
Priority(2);
```

```
vbl=Screen('Flip',window); %synchronizing to retrace.
```

```
for i=1:36
```

```
    Screen('DrawTexture',window,durer,[],[],10*i);
```

```
    vbl=Screen('Flip',window      , vbl +  
    (3-0.5)*refresh,[],1);
```

```
end;
```

```
Priority(0);
```

```
KbCheck;
```

```
Screen('CloseAll');
```

MaxPriority

Allows you to determine the
Maximum priority level that will
allow your functions to execute
properly!

Call:

```
priorityLevel = MaxPriority(windowPtr, ['Flip'], ['MakeTexture'], ...  
    ['BlankingInterrupt'], ['SetClut'], ['ClutMovie'], ...  
    ['SND'], ['sound'], ['speak'], ['DrawTexture'], ...  
    ['GetSecs'], ['WaitSecs'], ['cputime'], ...  
    ['KbCheck'], ['KbWait'], ['CharAvail'], ['GetChar'], ...  
    ['EventAvail'], ['GetClicks'], ['GetMouse'], ['GetTicks']));
```

MaxPriority

```
load durer;
durer=Screen('MakeTexture',window,X);
Priority(MaxPriority('DrawTexture','Flip'));
vbl=Screen('Flip',window); %synchronizing to retrace.
for i=1:36
    Screen('DrawTexture',window,durer,[],[],10*i);
    vbl=Screen('Flip',window, vbl + (3-0.5)*refresh,[],1);
end;
Priority(0);
KbCheck;
Screen('CloseAll');
```

GUIs

What are GUIs?

Graphical User Interfaces.

We already know one:

questdlg

quesdlg

ButtonName=QUESTDLG (Question)

- automatically resizes window to fit question.
- ButtonName is the name of the button that was pressed (a string) .
- Stops execution of Matlab until user responds.

questdlg

```
ButtonName=questdlg(Question,Title,Btn1,Btn2,  
    DEFAULT);
```

-up to three buttons.

-Default is optional.

```
ButtonName=questdlg('What is your wish?', ...  
    'Genie Question', ...  
    'Food','Clothing','Money','Money');
```

```
switch ButtonName,  
    case 'Food',  
        disp('Food is delivered');  
    case 'Clothing',  
        disp('The Emperor''s new clothes have arrived.')  
    case 'Money',  
        disp('A ton of money falls out the sky.');
```

end % switch

INPUTDLG

ANSWER = INPUTDLG(PROMPT) creates a input dialog box where users can enter text, saved in the cell array ANSWER. PROMPT is a cell array containing the PROMPT strings.

Ex:

```
prompt={'Enter the ethnicity of Participant','Enter the gender  
of the participant'};  
name='Participants demographics';  
numlines=1;  
defaultanswer={'hispanic','male'};  
answer=inputdlg(prompt,name,numlines,defaultanswer);  
options.Resize='on'; %makes window  
resizable  
options.WindowStyle='normal'; %normal or modal...  
options.Interpreter='tex'; %rendering text using LaTeX  
answer=inputdlg(prompt,name,numlines,defaultanswer,options);
```

INPUTDLG

Limitations:

user can enter whatever he/she wants...

Better to have control over options.

GUIDE

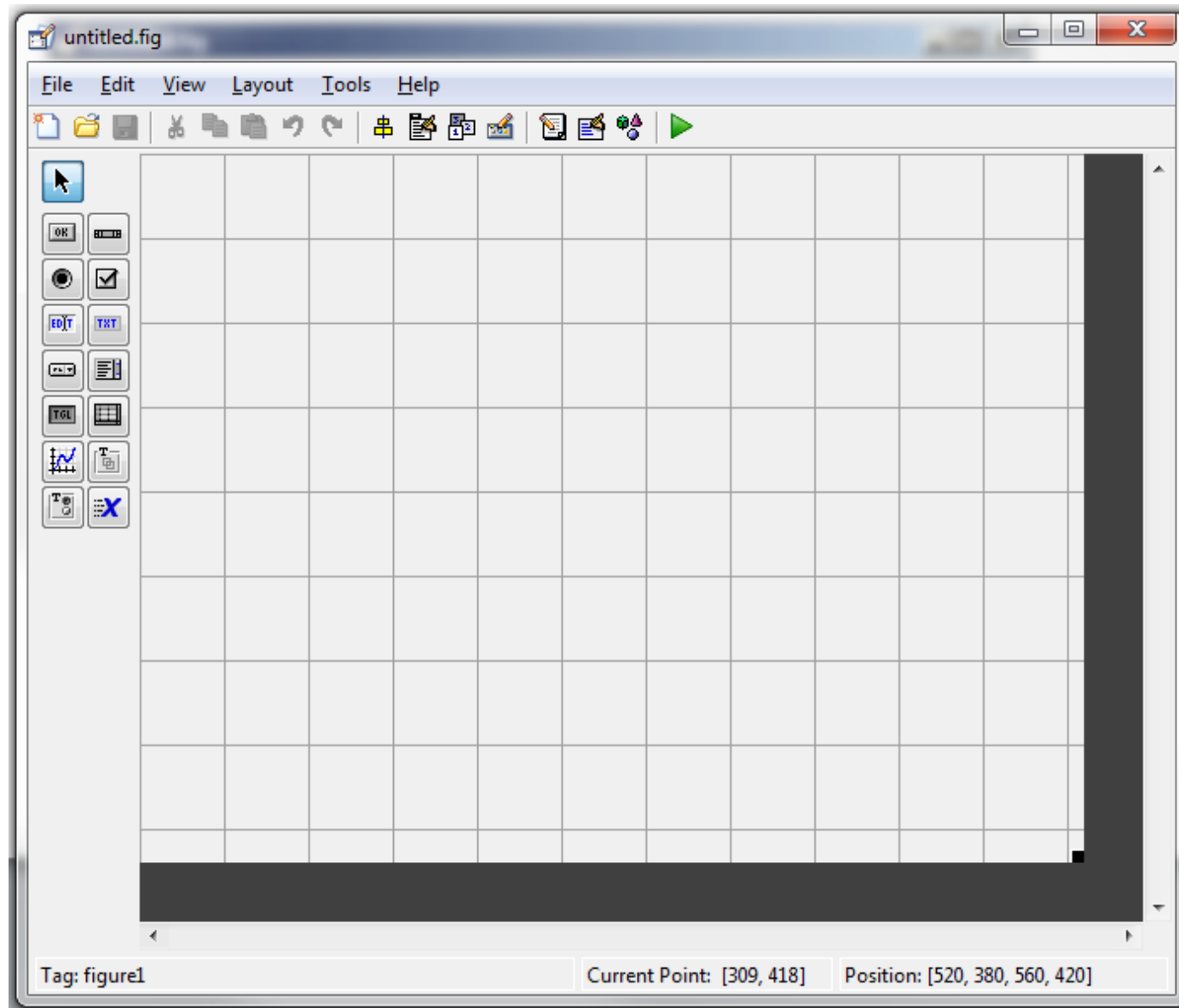
Powerful MATLAB interface to create GUIs.

Complicated.

Type guide.

Enter: Blank GUI (default)

GUIDE



GUIDE

Today:

Simple example to get user information.

1. Drag a button group

SimpleGui.fig

File Edit View Layout Tools Help



how to get user input

Ethnicity

- ☒ White
- ☐ Native American
- ☐ African American
- ☐ hispanic

(Ethnicity)



Gender

Male

female

Button Group

Enter

Cancel

GUIDE

Today:

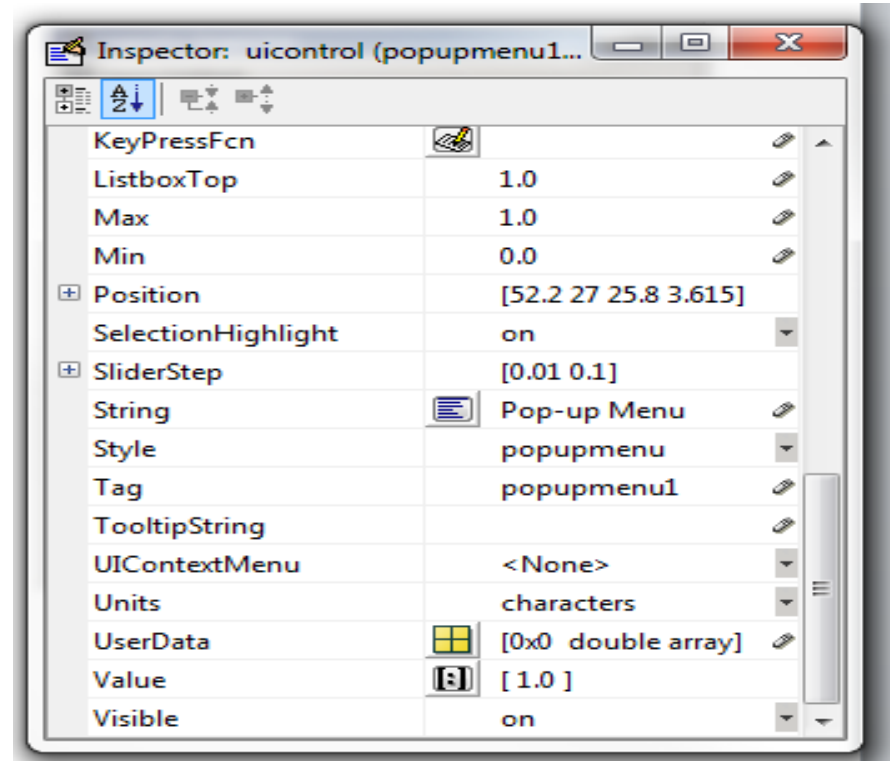
Simple example to get user information.

1. Drag a button group
2. Add buttons to it
3. Add a pop-up menu
4. Add a push-button

GUIDE

Property Inspector:
Select a component.
View Property Inspector.

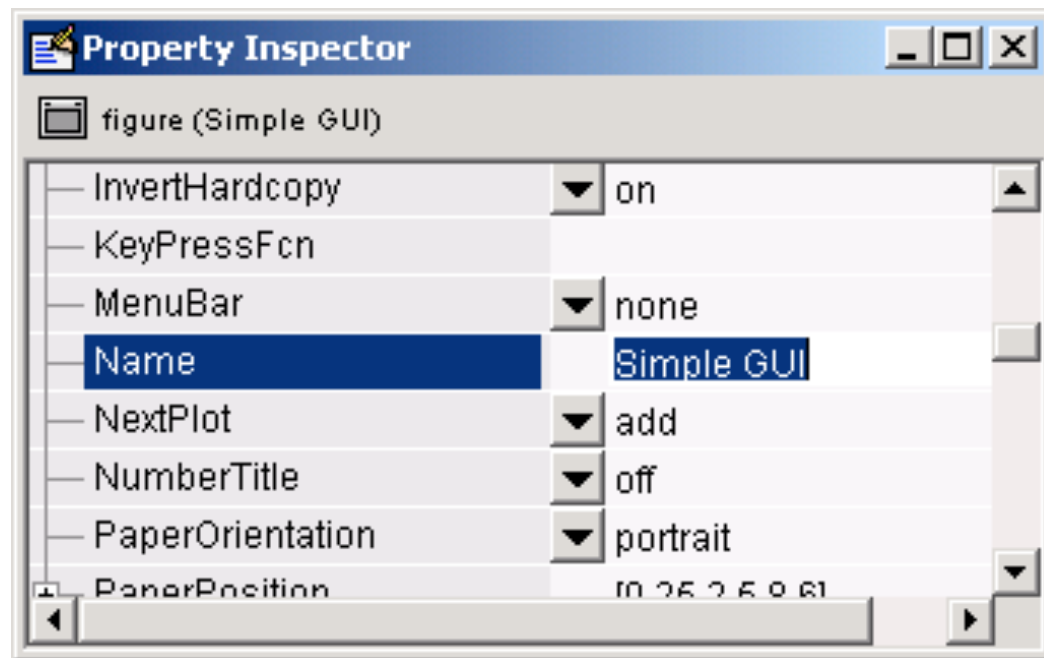
Use the property Tag to uniquely
Refer to each GUI component



GUIDE

Saving Values:

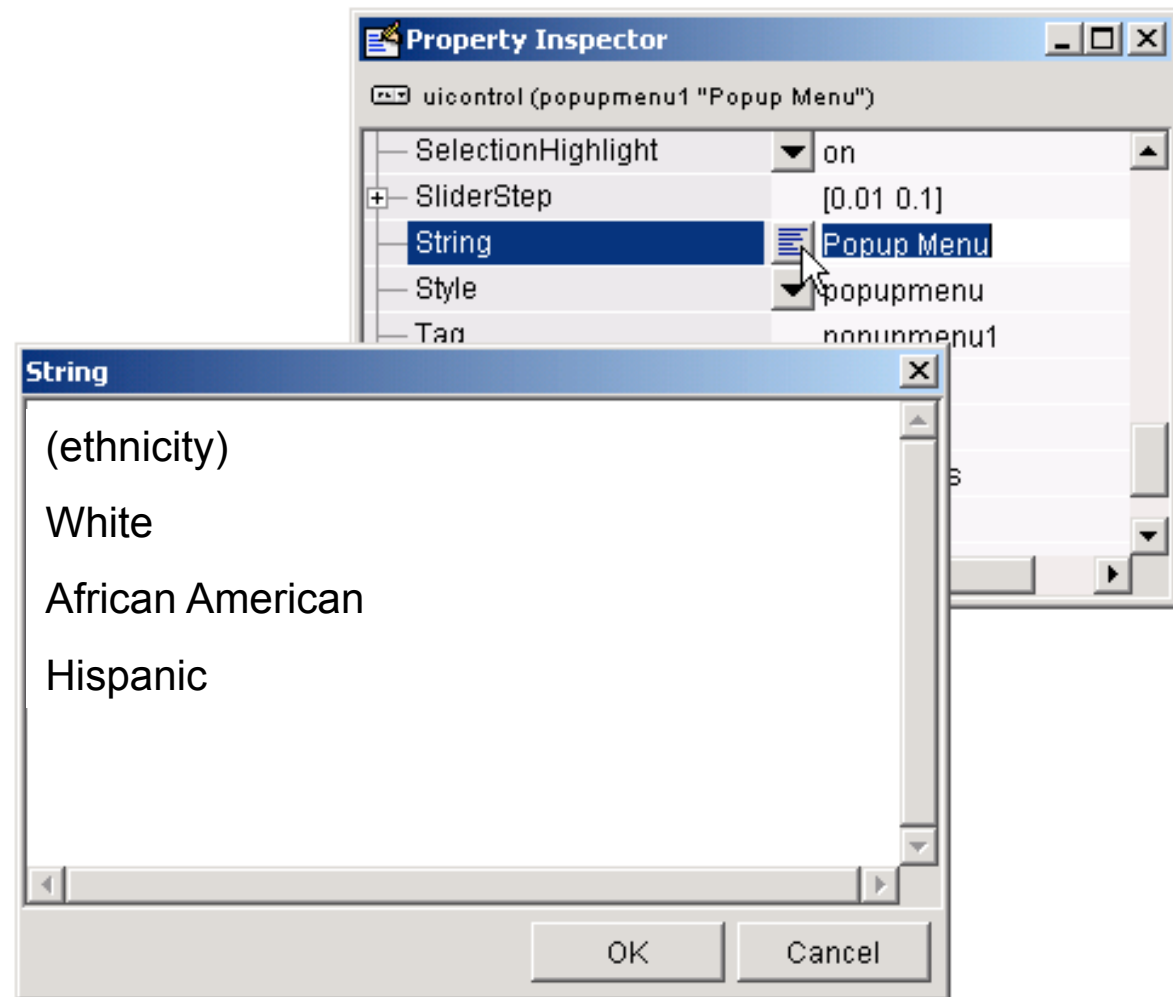
- Change Name/Tag Property of objects for ease of use.
- Save Gui.



GUIDE

Pop_up_menu:

Enter values:



GUIDE

Looking at the automatic code: Editor

```
% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject      handle to popupmenu1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell
        array
%          contents{get(hObject,'Value')} returns selected item from popupmenu1
ethnicity_entered = get(hObject,'Value');
save ethnicity ethnicity_entered;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
close SimpleGui
```

Advice

- Keep using it... Even if only to try out things on yourself (no need for a full experiment).
- Email me, Yusuke, or other class members about Matlab issues.
- Course lectures will remain online
- And, get on the habit of double checking design and event durations.
- Remember, the first time you do anything in programming it might take a long time, but not so every time afterwards!

Happy Break!