

Graphical User Interfaces I



The central station is the bubble gum planet with 280 pounds of clay and three head phone lines that come out and orbit the mass. The clay is not only on the top of the table but under it and under the chairs. It is inspired by the gum under tables rebelliously put there by those imprisoned in boring pedantics. It is a monument to the struggle of repressed physicality. Its form is never finalized as people freely sculpt it.

Outline

- Announcements
 - HW III due Wed. 5PM
- Homework II
- Summary of Friday
- The New GUIDE
- Programming converttemp.m
- Advanced GUI features

Homework II

- Very impressed with HW II--lots of fun to grade
- Some cool ideas
 - STOP sign: use polar coordinates to position vertices
 - mypcolor--put NaN's in last column and row of "fake" matrix
 - Ensures that data determines color limits

When we last saw our heros...

- GUI's are made of UIcontrols of various ilks (slider, push buttons, radio buttons etc.)
- Position the controls and set properties with GUIDE
- 'callback' field is a function call that gets executed when the control is activated

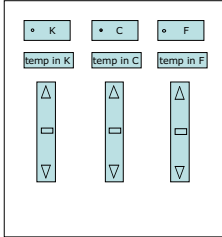
The New GUIDE

- Position your objects, set their options, and save as guiname. Matlab will create two files:
 - guiname.fig--binary file that contains all of the objects you created and their exact properties (including data fields)
 - guiname.m--function that creates your GUI & has "stubs" for the callbacks
 - 1. loads the .fig file and renders it
 - 2. gets handles to the objects
 - 3. callback functions are subfunctions of this file

The New GUIDE--callbacks

- Callbacks are automatically set to
 - Tag_Callback(h,eventdata, handles, varargin)
 - H=this object (gcbo)
 - eventdata is blank
 - Handles=handles to all objects as a struct (handles.tag gets handle to object with tag=tag)
 - Varargin--allows you to pass more info
 - Function stub is placed in guiname.m as a subfunction
- You can manually set callbacks with the property editor
 - No stub is created

Example converttemp



- We want to create a simple GUI to convert between Kelvin, Celsius, and Fahrenheit temperatures
 - Users will specify temperatures by typing or moving sliders

Example: converttemp.m

- We've set the positions, lets set some data:
 - Set sliders' min & max values +/- 100°C
 - Set sliders' values=0°C
 - Set edits' strings=num2str(value)
 - Set Celsius button to on (value=1) and others to off (0)
 - Set Kelvin and Fahrenheit sliders' visibility to off
 - Set figures userdata to handle matrix
- Let's figure out the callbacks!

$h = \begin{bmatrix} \text{radio1}, \text{radio2}, \text{radio3} \\ \text{edit1}, \text{edit2}, \text{edit3} \\ \text{slide1}, \text{slide2}, \text{slide3} \end{bmatrix}$

Functionality of converttemp

- Top-down design is critical to GUI development
 - Figure out the big picture first, then the details:
 1. Sketch layout of objects
 2. Create layout with GUIDE
 3. Describe what should happen when buttons are pressed, sliders slid in English or pseudocode
 4. Add functionality gradually--simple things first

Functionality of converttemp

- 3. Describe what should happen when buttons pressed, sliders slid
 - Radio Buttons: turn corresponding slider on, others off
 - Edits:
 - 1. convert string to number & adjust corresponding slider's value
 - 2. convert value to other units and update their edits & sliders
 - Sliders:
 - 1. convert value to string & adjust corresponding edit's string
 - 2. convert value to other units and update their edits & sliders

Miscellaneous & Advanced GUI features

- File Open/Save dialogs- `uigetfile` & `uisavefile`
 - can place these calls in any function
 - especially common inside callback of GUI button
- Common GUI features:
 - There are 18 fields that all Matlab objects have (things like position, type, and tag)
 - `ButtonDownFcn` allows objects to respond to user

Button Down Functions

- `ButtonDownFcn` acts like a callback--it contains a function call that is executed when user clicks on object
 - Could make stop sign turn green when clicked
 - Get the $z=f(x,y)$ of a patch where the user clicked (x,y) :
 - function `zlookup`
 - `ax=get(gcbo,'parent');` %axes of object
 - `xy=get(ax,'CurrentPoint');`
 - `<get x, y, Z, from patch's userdata>`
 - `z=interpZ(x,y,Z,xy(1),xy(2));`
 - `<send results to appropriate place>`
