O Ambiente App Designer do MATLAB

Pedro Cortez Lopes Rafael Lopez Rangel Luiz Fernando Martha



CIV2801 – Fundamentos da Computação Gráfica Aplicada 2024.2

Interface Gráfica do Usuário (GUI)

- Uma interface gráfica permite a interação do usuário do computador com um programa por meio de elementos gráficos (botões, menus, etc.)
- O ambiente App Designer permite a criação manual de uma interface gráfica do utilizador (GUI Graphical User Interface) de forma rápida e interativa
- A comunicação entre as ações realizadas pelo usuário sobre os elementos gráficos e a resposta do programa se dá por meio do paradigma da Programação Orientada a Eventos
- Interfaces gráficas também podem ser criadas no MATLAB apenas com código de programação (criação programática), ou pelo ambiente GUIDE



Interface Gráfica do Usuário (GUI)

Antes de começar a montar uma interface gráfica é necessário projetá-la. Para isso, deve-se pensar em como se dará a interação do usuário com o programa:

- Quais funcionalidades estarão disponíveis e como serão executadas
- Quais componentes serão utilizados para permitir o usuário realizar essas tarefas
- Prever possíveis erros de uso do programa para cercar as ações do usuário
- Organização do layout

Para projetar interfaces com um grau de complexidade relativamente alto, recomendase esboçar o layout em uma folha de papel.

http://asktog.com/atc/principles-ofinteraction-design/



Interface Gráfica do Usuário (GUI)

Após projetar a interface, deve-se decidir qual técnica será utilizada para criá-la. No MATLAB é possível criar uma interface de três formas: Utilizando o ambiente App Designer, o ambiente GUIDE, ou fazendo programaticamente, apenas através de código de programação (pode ser feito baseando-se em manipular Figuras – ideia por trás do GUIDE – ou seguindo o padrão de código orientado a objeto do App Designer).

	Pontos positivos	Pontos negativos
App Designer	 Ambiente gráfico interativo integrado com código Formalização de código seguindo POO De manuseio fácil e intuitivo Perspectiva de manutenção e avanços no futuro 	 Ainda em desenvolvimento, não está otimizado (aplicações de moderada complexidade podem ficar lentas)
GUIDE	 Ambiente gráfico interativo para manuseio de componentes gráficos Relativamente eficiente Maior extensão de referências e fóruns online para buscar ajuda 	 Código menos formal para o contexto de um programa estruturado Menos opções de componentes gráficos Sem perspectiva de avanços
Código	 Maior controle de posição e comportamento de todos os componentes gráficos em cada diálogo Maior controle da interação entre diferentes GUIs Maior liberdade ao desenvolvedor 	 Código deve ser fornecido para descrever todos os aspectos da interface gráfica Em geral, maior investimento de tempo

Programação Orientada a Eventos

- O fluxo do código é guiado por indicações externas chamadas eventos.
- Eventos são as diferentes ações que usuários podem realizar sobre os componentes adicionados à interface.
- Cada evento está associado à uma função chamada *Callback*, disparada quando se verifica a ocorrência de tal evento, que define a reação do programa.
- No App Designer, as *Callbacks* são associadas a <u>métodos</u>, no contexto da POO, de uma classe que descreve a aplicação gráfica.



```
Button1_click_CallbackFunction
{...}
```

```
Button2_click_CallbackFunction
{...}
```

```
Button3_click_CallbackFunction {...}
```

App Designer – Quick Tutorial

• Um simples tutorial para o uso do App Designer é disponibilizado pela MathWorks:

<u>https://www.mathworks.com/help/matlab/creating_guis/create-a-simple-app-or-gui-using-app-designer.html</u>

• Também está disponível em vídeo:

https://www.mathworks.com/videos/app-designer-overview-1510748719083.html

App Designer – Quick Tutorial

• Para executar o tutorial do App Designer, abra App Designer Start Page

📣 MATLA	AB R2023a -	acader	mic use								
HOM	1E	PLOT	s	APPS							
		ţ	P	÷÷÷ C		۵	23	•	• •	\mathbb{A}	
Design App	Get More Apps	Install App	Package App	MATLAB Coder	Application Compiler	LESM	Class Diagram Viewer	Code Analyzer	Code Compatibilit	Data Cleaner	D
	pen App D	esigner	to create o	r edit an app					A	PPS	
	3 🔽 🖾	•	C: • Use	rs 🕨 lfmartha 🕨	Documents +	MATLAB + E	xamples • R202	3a 🕨 matlab 🕨	TutorialAppDe	esignerExample	
Current F	older		\odot	Command Wind	low						
N	ame 🔺			New to MATLA	3? See resource	s for <u>Getting St</u>	arted.				
省 Tut	torialAppDe	esignerE	xample	<i>f</i> x >>							

App Designer – Quick Tutorial

• Clique em "Show examples" in the Apps section.

Getting Started How-to Videos GUID	E Migration Strategies Display Graphics in Ap	p Designer Find and Create UI Components	Release Notes
Apps			
Blank App	2-Panel App with Auto-Reflow	3-Panel App with Auto-Reflow	
			Show examples

• Selecione "Interactive Tutorial"

Examples: General



Inicializando uma Janela de Interface Gráfica Vazia



App Designer

É possível elaborar o layout de uma GUI, atribuindo e modificando as propriedades dos componentes gráficos. É habilitado um sistema *drag and drop* para posicionamento interativo de componentes.



			CONTAINERS		
COMMON					
Axes	Button	Check Box	Grid Layout	Panel	Tab Group
			FIGURE TOOLS		
30 Date Picker	Drop Down	[123] Edit Field (Numeric)	Menu Bar		
abc	~	А	INSTRUMENTAT	ION	
Edit Field (Text)	Image	Label	90 Degree	Discrete Knob	Gauge
a b c	⊙a ⊜b	1 2	Gauge		ougo
List Box	Radio Button Group	Slider	Knob	O Lamp	Linear Gauge
0	STATE	123			
Spinner	State Button	Table	0		Ø
			Rocker Switch	Semicircular Gauge	Switch
			12271		
Text Area	Toggle Button Group	Tree	Toggle Switch		



compontentes gráficos



É possível inspecionar e modificar propriedades de componentes gráficos, na aba lateral direita.



Clique com botão direito na UIFigure abre opções, incluindo de adicionar *callbacks*

Run	F5	
Cut	Ctrl+X	
Сору	Ctrl+C	
Paste	Ctrl+V	
Show grid	Alt+G	
Snap to grid	Alt+P	
Show alignme	e <mark>nt hi</mark> nts	
Show resizing hints		
Callbacks +		
Help on Selection		
	Run Cut Copy Paste Show grid Show grid Show alignme Show resizing Callbacks Help on Select	

Inspector Callbacks		
StartupFcn	Function name 🔹	
CloseRequestFcn	Function name 👻	
ButtonDownFcn	Function name 🔹	
▼ KEYBOARD		
KeyPressFcn	Function name 🔹	Ø
KeyReleaseFcn	Function name -	
- WINDOW		
SizeChangedFcn	Function name	
WindowButtonDownFcn	Function name -	
WindowButtonMotionFcn	Function name 🔹	N
WindowButtonUpFcn	Function name	K
WindowKeyPressFcn	Function name 🔹	
WindowKeyReleaseFcn	Function name	
Mandary Case III Mana III an	-	Test

Add StartupFcn callback	
Add CloseRequestFcn callback	
Add ButtonDownFcn callback	
Keyboard Callbacks	•
Window Callbacks	•
Select existing callback	

A medida que a interface gráfica é modificada interativamente, o código é atulaizado automaticamente. É possível visualizar e modificar partes do código neste mesmo ambiente, selecionando a aba *code view*.

A App Lesigner - app I.mitapp	
B A A Creatly Constraints Show region of Show region bits Show region bits Show region bits	
Save Tr IP in Same Size Grouping	
FILE ALIGN ARRANGE SPACE VIEW RUN	Ā
app1 mlapp ×	
COMPONENT LIBRARY	Design View Code View Code View
	Search p
	app.UlFigure
	Inspector Callbacks
Axes Button Check Box	Search P 📰 😰
	WINDOW APPEARANCE
30 4 - 123	POSITION
Date Picker Drop Down Edit Field	
(valienc)	IDENTIFIERS
	▶ INTERACTIVITY
b 0.b 1.2 List Boy Ratio Button Silder	
Group	
Spinner State Button Labie	
Text Area Toggle Button Tree Group	
CONTAINERS	
Grid Layout Panel Tab Group	
FIGURE TOOLS	
PEL	
P1	► [

📣 App Designer - app1.mlapp			- 0 ×
DESIGNER EDITOR			
Callback Function Property AppInput Q Fin FiLE INSERT NAVI app1.mlapp × X	To Comment % Sec Enable app coding alerts Show Tips Run ATE EDIT VIEW RESOURCES RUN		×.
▼ CODE BROWSER			COMPONENT BROWSER
Callbacks Functions Properties		Design view Code view	Search
Search 👂 🛟	1 Classdef app1 < matlab.apps.AppBase	Î.	app.UIFigure
Search Add a caliback function to make your apprespond to user interactions such as button clicks. APP LAYOUT	<pre>% Properties that correspond to app components properties (Access = public) UlFigure matlab.ui.Figure end % Component initialization methods (Access = private) % Create UIFigure and components function createComponents(app) % Create UIFigure and hide until all components are created app.UIFigure = uifigure('Visible', 'off'); app.UIFigure.Position = [100 100 640 480]; app.UIFigure.Name = 'UI Figure'; % Show the figure after all components are created app.UIFigure.Visible = 'on'; end % Construct app function app = app1 % Create UIFigure and components createComponents(app) % Construct app % Create UIFigure and components createComponents(app) % Register the app with App Designer register@and.and.uIFigure)</pre>		app.UIFigure Inspector Callbacks Search P The second sec
	<pre>35 36 if nargout == 0 37 clear app 38 end 39 end 40 40 41 % Code that executes before app deletion 42 function delete(app) 43</pre>		
14	44 % Delete UIFigure when app is deleted	*	▶1

app1.mlapp* ×	app1.mlapp* ×	app1.mlapp* ×	
- CODE BROWSER	CODE BROWSER	▼ CODE BROWSER	
Callbacks Functions Properties	Callbacks Functions Properties	Callbacks Functions Properties	
Search 🔎	Search D	Search D 🔂 🗸	
Add a callback function to make your app respond to user interactions such as button clicks.	Add a function to organize your code into helper or utility functions. Call the function with app as the first input argument: func(app,);	Add a property to create a variable to store and share data between callbacks and functions. Specify the property name with the prefix app. to access the property value: app.Property = someData;	
Na possí	visualização do código, na aba vel adicionar <i>callbacks</i> , funçõe	esquerda, é es (métodos) e	
propr	riedades, públicos e privados, r	no contexto de	

POO. Estas modificações são automaticamente implementadas no código.



App Designer



App Designer – Ferramentas Auxiliares



App Designer – Arquivos

- Um arquivo .mlapp é gerado ao salvar projetos no App Designer.
- Este arquivo pode ser rodado diretamente pelo App Designer, ou chamado por código para inicializar a aplicação gráfica.
- É possível ainda copiar o código, na aba Code View do App Designer, e salvá-lo em um arquivo .m. Desta forma, pode-se trabalhar de forma puramente programática com a interface.



Alterando as Propriedades da Janela

Algumas propriedades básicas da janela de interface gráfica (UIFigure):

- Name: Define o nome que aparecerá na barra superior da janela.
- Tag: Propriedade presente em todos os componentes que define o nome pelo qual estes são identificados no código. É importante sempre modificar essa propriedade para cada componente adicionado para que o código não fique confuso, já que o nome padrão é a numeração na qual os componentes são adicionados.
- **Resize**: Indica se é permitido o redimensionamento da janela.

Descrição de todas as propriedades da janela de interface: <u>https://www.mathworks.com/help/matlab/ref/matlab.ui.figureappd-properties.html</u>

Componentes Gráficos

Algumas propriedades comuns à diversos tipos diferentes de componentes gráficos:

- Enable: Permite tornar um componente desabilitado ou inativo para o usuário. No primeiro caso, o componente fica com uma aparência sombreada, já no segundo continua com a mesma aparência de como se estivesse habilitado porém usuários não são capazes de utiliza-lo.
- Estilos de Fonte: Grupo de propriedades presente em qualquer componente que exiba algum texto e que permitem definir o tipo, tamanho, cor e orientação da fonte.
- Position: Permite definir as coordenadas X e Y do canto inferior direito do componente em relação à origem da janela de interface ou do painel no qual esteja inserido. Também inclui a possibilidade de alterar a largura e altura do componente. [X Y Largura Altura]

Componentes Gráficos

- **String**: Propriedade também presente em componentes que exibam algum texto e que permite definir o texto exibido.
- Tag: Como já mencionado, essa propriedade é presente em todos os componentes e define o nome pelo qual estes são identificados no código. É importante sempre modificar essa propriedade para cada componente adicionado.
- **Visible**: Propriedade presente em todos os componentes que, se desativada, torna o componente invisível.

Descrição de todas as propriedades dos componentes gráficos: <u>https://www.mathworks.com/help/matlab/ref/matlab.ui.figureappd-properties.html</u>

Componentes Gráficos: Label

Caixa de texto para exibir strings que não pode ser modificada pelo usuário.

Este componente não está associado a nenhuma função *callback*, sendo utilizado apenas para rotular outros elementos e fornecer informações ao usuário.

Algumas vezes a informação exibida deve ser modificada, para isso a propriedade "Text" deve ser acessada em um função *callback* e modificada.

Ex.: *Label* que indica o nome de uma barra de escala (componente do tipo *Slider*)



Componentes Gráficos: Edit Field (Numeric/Text)

Caixa de texto que pode ser alterada pelo usuário para inserir ou editar valores numéricos/texto (Numeric/Text).

Associado à *callback ValueChangedFcn*. Utilizado comumente para permitir a entrada de dados em programas.

Esses valores fornecidos pelos usuários nas caixas de texto podem ser obtidos acessando a propriedade "Value" do respectivo componente.

A propriedade "Value" só é alterada quando se clica fora da caixa de texto ou a tecla "Enter" é pressionada ("Enter" + Ctrl para caixas multilinha).



Componentes Gráficos: Push Button

Botão simples para realizar alguma ação quando ativado por um clique do botão esquerdo do mouse.

Esta ação (resposta do programa ao evento de clique do mouse sobre o botão) deve ser programada em uma função callback que é disparada no instante em que se libera o clique no interior do botão.



Componentes Gráficos: State Button

Semelhante ao *Push Button*, porém possui dois estados: ativado (pressionado) e desativado (não pressionado).

Quando ativado, a propriedade "Value" adquire o valor lógico *true = 1*, e quando desativado adquire o valor *false = 0*.

Para verificar se um *State Button* está ativado ou desativado, basta checar sua propriedade "Value".



Componentes Gráficos: Check Box

Assim como *State Buttons*, possui o estado ativado e desativado, associado a valores lógicos *true* e *false*.

Este componente é utilizado quando se deseja prover ao usuário opções independentes, que podem ter seleção múltipla.



Caixa Desativada



Caixa Ativada

Componentes Gráficos: Radio Button Group

Componente similar à *Check Box*, porém a diferença é que este pode ser utilizado para dar opções mutualmente exclusivas ao usuário, ou seja, apenas um *Radio Button* de um grupo pode ser selecionado.

O nome de um Button Group pode ser alterado através da propriedade "Title".

Ex.: Seleção de idioma:



Componentes Gráficos: Drop Down

Componente que exibe uma lista de opções quando aberto, e quando fechado indica a opção corrente.

Geralmente é utilizado quando se deseja criar uma lista de opções mutualmente exclusivas, porém não se dispõe de espaço suficiente para apresentar tais opções em um grupo de *Radio Buttons* ou *State Buttons*.

Para editar as opções deve-se modificar a propriedade "Items", em que cada linha corresponde a uma opção diferente.

Para determinar qual a opção que está selecionada, deve-se acessar a propriedade "Value".

Uma função *callback* pode ser disparada sempre que a opção for trocada.

Opção 1	-
Opção 1	
Opção 2	
Opção 3	
Opção 4	
Opção 5	

Componentes Gráficos: Slider

Sliders são componentes geralmente utilizados para variar um valor numérico entre um valor mínimo e um máximo.

É atribuída à propriedade "Value" um valor que corresponde a posição do objeto deslizante que é proporcional aos valores mínimo e máximo atribuídos à propriedade "Limits" ([min,max]). Quando o objeto está no extremo esquerdo do *Slider*, a propriedade "Value" assume o valor da propriedade "Limits(1)", e quando se encontra no extremo direito assume o valor da propriedade "Limits(2)".

Uma função *callback* é disparada somente quando o objeto deslizante é solto, *ValueChangedFcn*, enquanto outra é chamada durante o movimento do *Slider*, *ValueChangingFcn*.



Componentes Gráficos: Table

Componente para armazenar dados textuais ou numéricos em forma de tabela.

O número e formatação de linhas e colunas pode ser definido editando a propriedade "ColumnFormat". Os dados podem ser acessados e editados através da propriedade "Data", na qual é possível fornecer um vetor ou matriz de dados que serão exibidos na tabela.

Componentes Gráficos: Panel

Tem apenas o objetivo de organizar a interface do programa em painéis que agrupam diferentes tipos de componentes.

O nome de um painel pode ser alterado através da propriedade "Title".

Componentes Gráficos: UIAxes

Eixos são onde os objetos gráficos são plotados e geralmente são utilizados como o canvas da interface gráfica.

Quando um comando de plotagem é executado, este utiliza o eixo ativo que é o mais recentemente criado ou o último utilizado. Em interfaces com mais de um eixo, devese especificar qual eixo é o ativo antes de executar qualquer comando de plotagem para evitar que se utilize o eixo errado.

Os eixos são naturalmente tridimensionais e possuem diversas propriedades e comandos que permitem controlar a visualização dos objetos gráficos.

https://www.mathworks.com/help/matlab/ref/matlab.ui.control.uiaxes-properties.html



Funções Úteis

Caixas de Diálogo:

Diferentes tipos de caixas de diálogo predefinidas podem ser exibidas com comandos simples, disponíveis em:

https://www.mathworks.com/help/matlab/develop-apps-using-the-figurefunction.html

Manipulação dos Eixos

Especificar o estilo do eixo: *axis(UmHandleParaUIAxes,'equal');* Limpar todos os objetos gráficos: *cla (UmHandleParaUIAxes);* Objetos gráficos não sejam substituídos na próxima plotagem: *hold(UmHandleParaUIAxes,'on');*

Especificar os limites de um eixo: UmHandleParaUIAxes.Xlim = [a, b];

Ativar a grade: UmHandleParaUIAxes.XGrid = true;

Definir as coordenadas do centro da câmera: *campos(UmHandleParaUIAxes,x, y, z);*

Obter as coordenadas do centro da câmera: var = campos;

Mover a câmera para uma visão 2D ou 3D: view(UmHandleParaUIAxes,2); ou view(UmHandleParaUIAxes,3);

Funções Úteis

> Manipulação de Arquivos:

Criar um arquivo txt para escrita: txt = fopen(fullname, 'wt');Criar um arquivo txt para leitura: txt = fopen(fullname, 'rt');Abrir um arquivo na tela: winopen('NomeDoArquivo.extenção');Imprimir dados: fprintf(txt, 'texto ou número %f', valor);Ler dados: fscanf(txt, '%f', NúmeroDeDados);

Support for interactive and programmatic app building

Support for programmatic app building only

Feature	GUIDE	App Designer
Component Library	0	0
Property Inspector	0	0
Component Browser	ø	0
Callback Sharing	0	0
App Templates	0	0
Single-File Management for Code and User Interface		0
Modern Look & Feel		0
Toolstrip-Based Desktop		0
Rich Canvas Interactions		0
Embedded Code Editor		0
Rich Property Editors		0
App Metadata Management		0
Robust Programming Editor and Workflow		0
Quick Start Interactive Tutorial		0
App Testing Framework Support		0

Support for interactive and programmatic app building

Support for programmatic app building only

Code Editor

Feature	GUIDE	App Designer
Code Debugger	0	0
Code Folding	0	0
Code Analyzer	0	٢
App Input Arguments	0	0
Rich Programming Aids		0
Quick Code Navigation		0
Global Component Rename		0
Utility Functions via Custom App Methods		0

Graphics Support

Feature	GUIDE	App Designer
2D & 3D Plotting	0	0
Axes Support (axes, polaraxes, geoaxes)	0	0
Axes Interactions (pan, zoom, rotate)	0	0
Figure Interactions (keyboard and mouse events)	0	0
Printing Support	0	

Support for interactive and programmatic app building

Support for programmatic app building only

- 1. Requires MATLAB Compiler.
- 2. MATLAB Online is available with select licenses.

App Sharing & Access

Feature	GUIDE	App Designer
Share as MATLAB App Install File	0	0
Share as MATLAB Standalone Desktop App '	0	0
Share as MATLAB Web App 1		٢
Create, Edit, and Run Apps in MATLAB Online *		٢
Export as Standalone MATLAB Class File		0

Containers & Layout Managers

Feature	GUIDE	App Designer
Panel	0	0
Menu Bar	0	0
Tab Group	0	0
Scrollable Containers		0
Grid Layout Manager		٢
Auto-Reflow		0
Context Menu	0	
Toolbar	0	

Support for interactive and programmatic app building

Support for programmatic app building only

Common Components

Feature	GUIDE	App Designer
Button	0	0
Drop Down	0	0
Edit Field (Text)	0	0
Label	0	0
List Box	0	0
Radio Button Group	0	0
Slider	0	٢
Table	٥	0
Text Area	•	0
Toggle Button Group	0	0
Date Picker		0
Edit Field (Numeric)		0
Image		0
Spinner		0
State Button		0
Tree		0



Support for interactive and programmatic app building

3. Requires Aerospace Toolbox.

Support for programmatic app building only

Instrumentation

Feature	GUIDE	App Designer
Gauge		0
90 Degree Gauge		0
Linear Gauge		0
Semicircular Gauge		0
Knob		٥
Discrete Knob		٥
Lamp		٢
Switch		0
Rocker Switch		0
Toggle Switch		0
Airspeed Indicator ^a		0
Altimeter ³		0
Climb Rate Indicator ^a		0
EGT Indicator *		0
Heading Indicator ^a		0
Artificial Horizon ^a		0
RPM Indicator ³		0
Turn Indicator ⁵		0

Support for interactive and programmatic app building

Support for programmatic app building only

Dialogs

Feature	GUIDE	App Designer
Alert Dialog	0	0
Confirmation Dialog	0	0

Component Extensibility

Feature	GUIDE	App Designer
ActiveX Control	0	