Large-Scale Development in MATLAB

David Sampson
MathWorks Consulting
Challenge

How can you keep control of a large codebase of MATLAB code?

Measures of success:

- Easy to add new features
- Easy to qualify against new releases
- Easy for new staff to ramp up
- Fewer grey hairs
Themes

- Technologies
- Patterns
- Attention to detail
- Version control and testing
Demonstration

2500 lines of code
11 classes
6 helper functions
Themes

- Technologies
- Patterns
- Attention to detail
- Version control and testing
Handles and values

- Built-in types
- Mutability
- Life cycle
Events and listeners

- Built-in events
- Listener callbacks
- Custom events
- addlistener or event.listener?
Themes

- Technologies
- Patterns
- Attention to detail
- Version control and testing
Example component types

- Data
- Importer/exporter
- Chart
- Meter
- Model-view-controller
- Application launcher
- Algorithm
- Session
Custom chart

- Use object composition
- Hide low-level APIs behind dependent properties
- Manage object life cycle
- Expose layout properties (Parent, Units, Position)

Benefits:
- Presents a convenient API
- Enables use both interactively and within applications
Model-view-controller pattern
Model-view-controller pattern

- Model is a handle class
- Model exposes events
- View/controller is constructed with and stores listeners to a model

Benefits:
- Enables separation of logic and presentation
- Enables separation of parts of presentation
Application launcher

- Create model
- Create graphical container(s)
- Create and lay out views
- Create menus and toolbars

Benefits:
- Enables application content and layout to be changed easily
Layout management

- GUI Layout Toolbox

8: A complete example

The following example application uses many of the layout features discussed above in order to create a good-looking user interface that scales well when resized. It is not designed to showcase all the layout functionality, but shows how callbacks are added to provide a user interface. It also exemplifies separating the data from the GUI, a fundamental part of creating modular and maintainable applications.

The full application is available here:
- View demoBrowser.m
- Edit demoBrowser.m
- Run demoBrowser.m

Section contents:
- 8.1. Application structure
- 8.2. createInterface
- 8.3. updateInterface
- 8.4. onInitSelection
- 8.5. Running it
- 8.6. Scalability

This toolbox was developed by Ben Tordoff and David Sampson from the Consulting group at MathWorks.
Themes

- Technologies
- Patterns
- Attention to detail
- Version control and testing
Common pitfalls

- Combining components
- Not paying attention to shapes and sizes
- Obstructing object destruction
- Overusing public APIs privately
- Overusing events
Themes

- Technologies
- Patterns
- Attention to detail
- Version control and testing
More resources

- Training
- Workshop
- Application design and development support
Thank you!

Questions?