Objective & Agenda

- Present what we are doing
- Give a picture where we want to go
- Open the game to get your opinion
Pharo 9
Community Impact

• Since Mid December 2019
• Issues Closed: 1600, 220 since Pharo 9
• Issues Open: 529, 191 since Pharo 9
• Contributors: 60 in Pharo 9
Pharo 9
Improvements

• Improvements in the compiler
• Improvements in speed of source files - Better usage of buffers
• Clean up of World / Hand handling - Clean up of OSWindows
• Fixing of flaky tests
• Improving Documentation / Categorization of Methods and Classes
• Improvements in code completion (see large images support)
• Improvements in spotter (see large images support)
Pharo 9
Improvements (cont…)

• Improving Exception handling and hierarchy
• Fixes in the minimal image generation
• Fixes in Iceberg & Tonel: better handling of class side traits
• Improving the API of Collections
• Improving Epicea
• Improving Settings saving, loading, and defaults.
Pharo 9

A lot of Improvements, available to download, test and have fun

pharo.org  pharo-project/pharo  Pharo Launcher
Pharo Launcher
New Release

• Fully rewritten using Spec2 and Commander
• Improved documentation website!
• Image configurations (configure each image, arguments, vm, …)
• Initialisation Scripts
• More metadata per image, and everything stored side by side with the image
Pharo Launcher
New Release

• Better Error handling
• Launch Configurations
• Faster in installations with a lot of images
• Templates Improvements:
  • Custom templates
  • Adding private Jenkins
• Support for Jenkins Pipelines
Pharo Launcher

pharo.org/download  pharo-project/pharo

pharo-project.github.io/pharo-launcher/
Large Images

- Images with a lot of Objects
  - Code
  - Data
- Pharo was slow… we improved that
Large Images

- Improving the startup and shutdown
- Improving responsiveness of Calypso and other tools
- A new tab-based morph that only draws the visible tab
- Improving the Syntax highlighter speed.
- Handling methods with big literals (Highlighting / QA Rules)
Large Images

• Adding more settings to enable/disable long time consuming features.
• Removing the non-essential calls to #allInstances
• Remove the use of pragmas in menus and commands
• A new code completion framework (Stream & Context based)
• A new Spotter backend (Stream & No-Pragmas)
• Improving loading of code, changes and source files
Large Images

External Tools

• An index manager for all the cross-reference activities

• An efficient implementation of Optimized Tries

• Integration of the Indexes

• Configuring the GC from the image to minimize the GC trashing during the generation of the indexes.
Large Images
External Tools

pharo-project/largeImages
Open Projects
Where we are going...
NewTools using Spec2

- Iteration on the playground
- Iteration on the inspector
- New Object centric Debugger
- Improvements in the DebuggerSession API and backends.
- Tab based playground
- Improving API of Code Widget for Syntax Highlighting
- Improving API of Commands in the editor
Lowcode

- Ronie Salgado is working on it
- Some issues still to fix
- Some design problems are not yet completely explored (Debugging, Stack handling, ...).
- First iteration on Memory access primitives: Benchmarks
- Documenting, adding tests
- **Objective**: pass from a prototype stage to a production-ready stage, replace in image FFI marshallings
IDLE VM

- Headless took away events from the VM
- Need for an interruptible event poll
- Reducing CPU usage while idle
- Requires Graphic Backends in the image side
- Reducing dependencies for Server installations
GTK Backend

- Using Cairo for rendering
- Needed for development of GTK applications using live programming
- Improvements in HDPI screens
- Reduction of platform-dependent code
- Better UI resources: menu handling from the image, OS UI notifications, better look and feel
GTK Backend

- Separation of the Morphic UI thread from the event processing.
- Removing Poll of events. Events are pushed to the image using callbacks.
- All events are handled in independent Callbacks.
- The requirement for good support for debugging callbacks: default return values, the correct order of return in case of multiple concurrent debuggers, management of the C call stack, extending the Debug capabilities to allow extensions.
SDL Backend

- Using Cairo for rendering
- Porting the SDL binding to UFFI
- Implementing two strategies: the main thread is the VM thread, the main thread is not the VM thread.
- Lighter backend: fewer dependencies.
- Removing redundant event polling
- More limited than GTK backend
Documentation
A never ending task...

- Improving Pharo documentation
- New open books & booklets in the production
- A new blog to document even more
- Adding tests as documentation in the VM and in the image
Thanks!!!

pharo.org

discord.gg/QewZMZa

pharo-project/pharo

thepharo.dev