An Introduction to Tilde

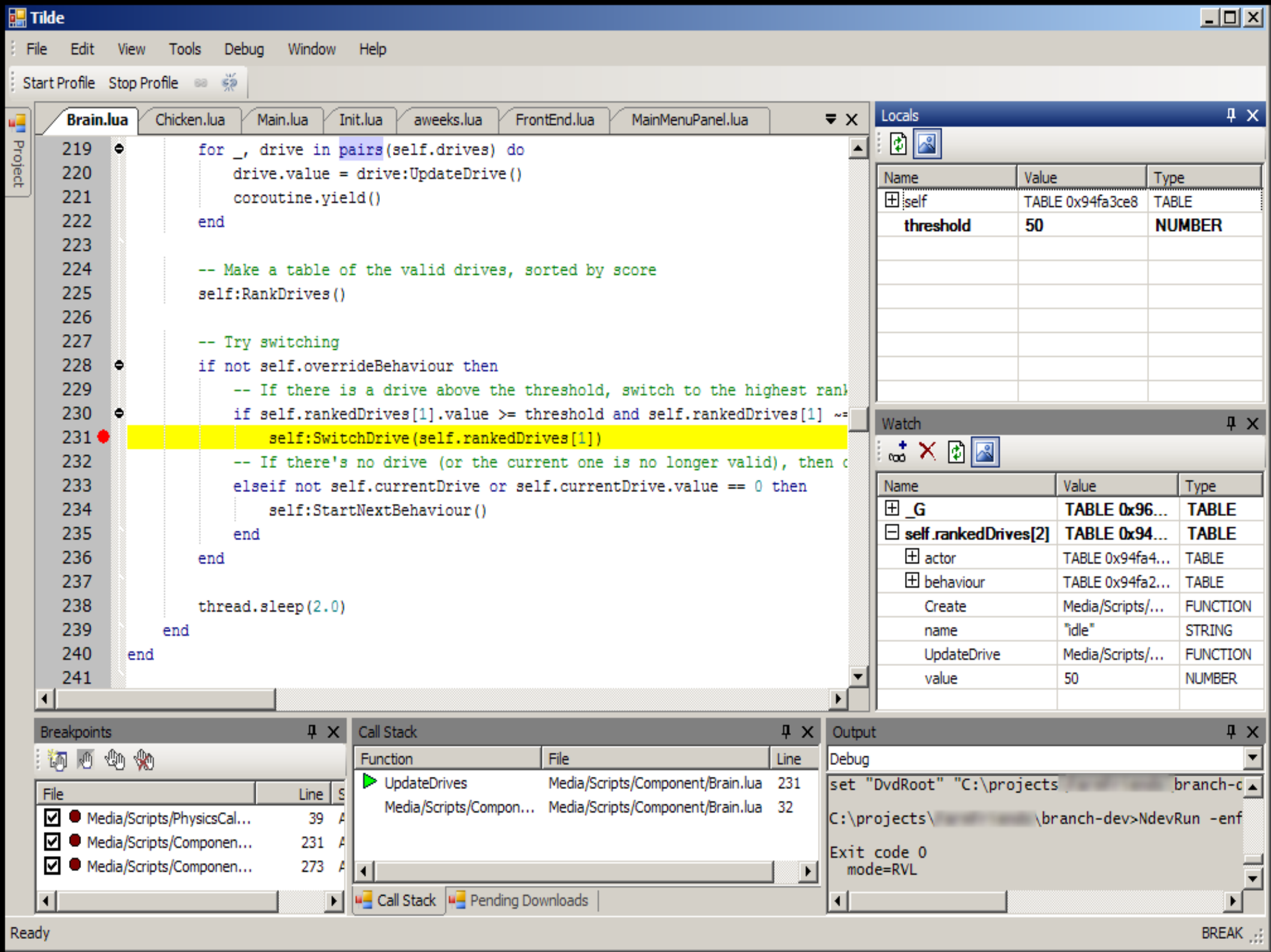
Presentation on a FOSS tool for Lua development

By Andrew Bailey, CTO, Tantalus

& Allen Weeks, Lead Programmer, Tantalus.

mailto:andrew@tantalus.com.au
mailto:aweeks@tantalus.com.au

TANTALUS


Tilde
File Edit View Tools Debug Window Help
Start Profile Stop Profile
Brain.lua Chicken.lua Main.lua Init.lua aweeks.lua FrontEnd.lua MainMenuPanel.lua
Project
219 for _, drive in pairs(self.drives) do
220 drive.value = drive:UpdateDrive()
221 coroutine.yield()
222 end
223
224 -- Make a table of the valid drives, sorted by score
225 self:RankDrives()
226
227 -- Try switching
228 if not self.overrideBehaviour then
229 -- If there is a drive above the threshold, switch to the highest rank
230 if self.rankedDrives[1].value >= threshold and self.rankedDrives[1] ~=
231 self:SwitchDrive(self.rankedDrives[1])
232 -- If there's no drive (or the current one is no longer valid), then c
233 elseif not self.currentDrive or self.currentDrive.value == 0 then
234 self:StartNextBehaviour()
235 end
236 end
237
238 thread.sleep(2.0)
239 end
240 end
241
Locals
Name Value Type
self TABLE 0x94fa3ce8 TABLE
threshold 50 NUMBER
Watch
Name Value Type
_G TABLE 0x96... TABLE
self.rankedDrives[2] TABLE 0x94... TABLE
actor TABLE 0x94fa4... TABLE
behaviour TABLE 0x94fa2... TABLE
Create Media/Scripts/... FUNCTION
name "idle" STRING
UpdateDrive Media/Scripts/... FUNCTION
value 50 NUMBER
Breakpoints
File Line
Media/Scripts/PhysicsCal... 39
Media/Scripts/Componen... 231
Media/Scripts/Componen... 273
Call Stack
Function File Line
UpdateDrives Media/Scripts/Component/Brain.lua 231
Media/Scripts/Compon... Media/Scripts/Component/Brain.lua 32
Call Stack Pending Downloads
Output
Debug
set "DvdRoot" "C:\projects branch-c
C:\projects\ \branch-dev>NdevRun -enf
Exit code 0
mode=RVL
Ready
BREAK

Overview

- Why Lua?
- Quick Lua Introduction.
- Why Tilde?
- Quick Feature set overview.
- Solutions using Lua and Tilde.
- How to install and implement.
- The Future.
- Where to get it.

Why Lua?

- Game development problems
  - Slow turnaround on code changes
  - Consistent frame rates
  - Complicated and inefficient game logic
    - C++ state machines
      - Polling 'ticks' inefficient
    - C++ fixed (run time) class hierarchies
      - Code refactoring downtime
  - Unrecoverable crashes/asserts when anything goes wrong
    - C++ Pointers! (not worth recovering as any memory can be trashed).

Why Lua? (concepts)

- Lua is a programming language, not a scripting language.
- Use it for Game logic, not services.
- Garbage collector as memory manager (no leaks).

Quick Lua Introduciton

- What is Lua?

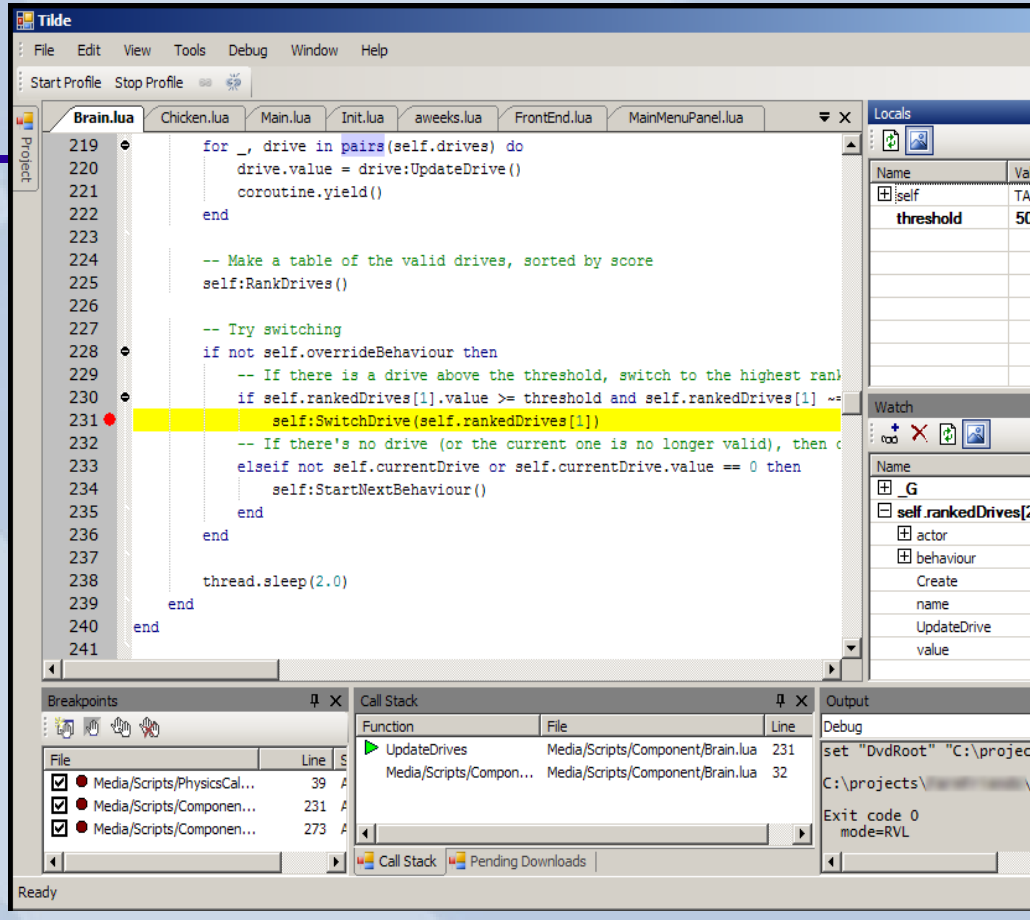
  From http://www.lua.org/about.html :-
  Lua is a powerful, fast, light-weight, embeddable scripting language.
  Lua combines simple procedural syntax with powerful data description constructs based on associative arrays and extensible semantics. Lua is dynamically typed, runs by interpreting bytecode for a register-based virtual machine, and has automatic memory management with incremental garbage collection, making it ideal for configuration, scripting, and rapid prototyping.

Quick Lua Introduciton

- Why choose Lua? (again from the site)
  - Lua is a proven, robust language
  - Lua is fast
  - Lua is portable
  - Lua is embeddable
  - Lua is powerful (but simple)
  - Lua is small
  - Lua is free

Why Tilde?

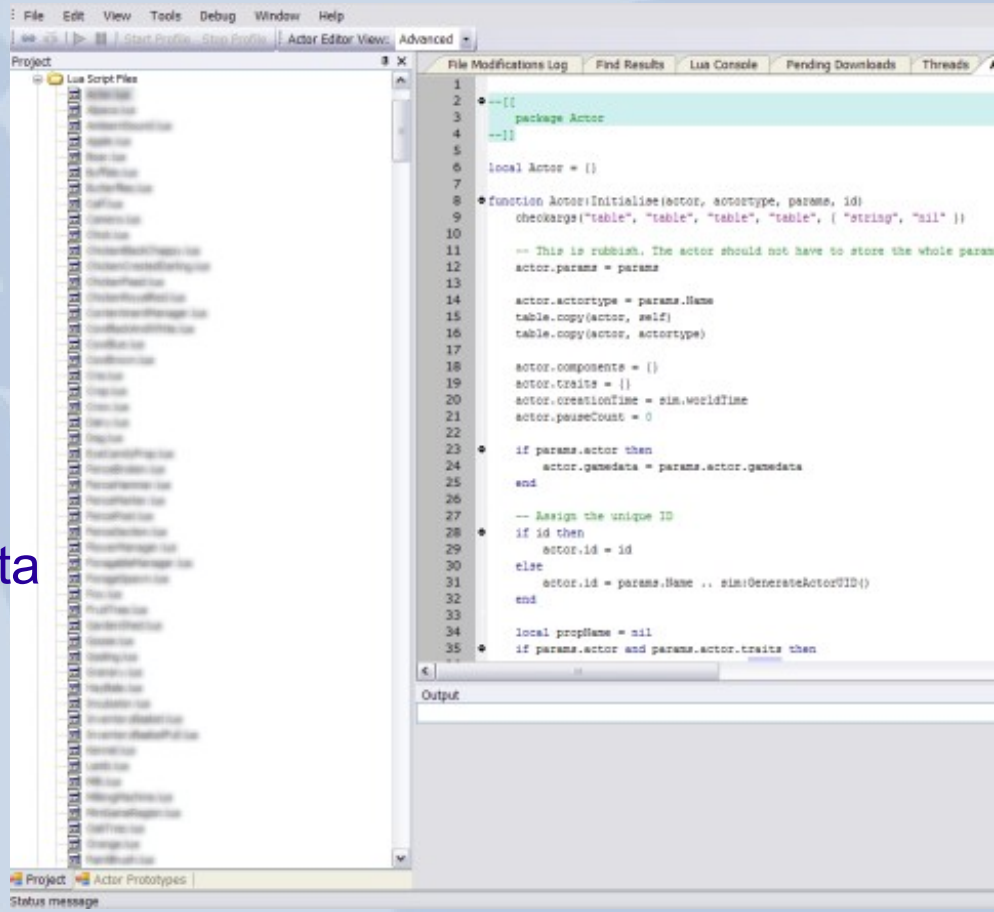
- Decided to use Lua
  - That only took a decade.
- It's a programming language, it needs debugging.
  - You are kidding yourself if you think not.
- Commercial and Free Debugger availability
  - Poor
- Write our own.
  - Took one week to do POC version.



TANTALUS

Quick Feature Set Overview

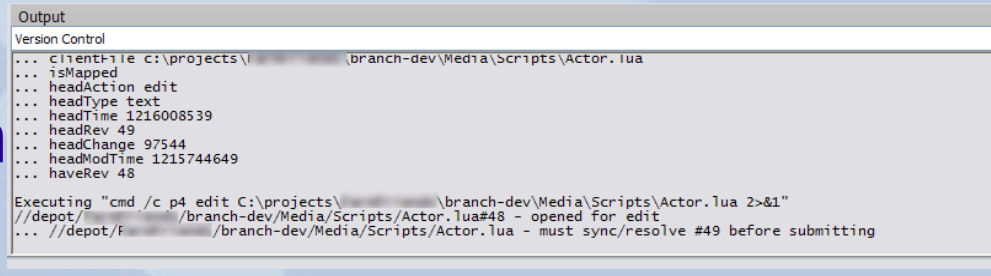
- Lua debugger features
  - Breakpoints
  - Stepping
  - Stack trace
  - Local variables and upvalues
  - Watches
  - Expanding tables and userdata
  - Viewing table and userdata metadata
  - Threads
  - Filtering in variable windows
  - Catching Lua errors
  - Script downloading and execution
  - Lua console
  - Console snippets



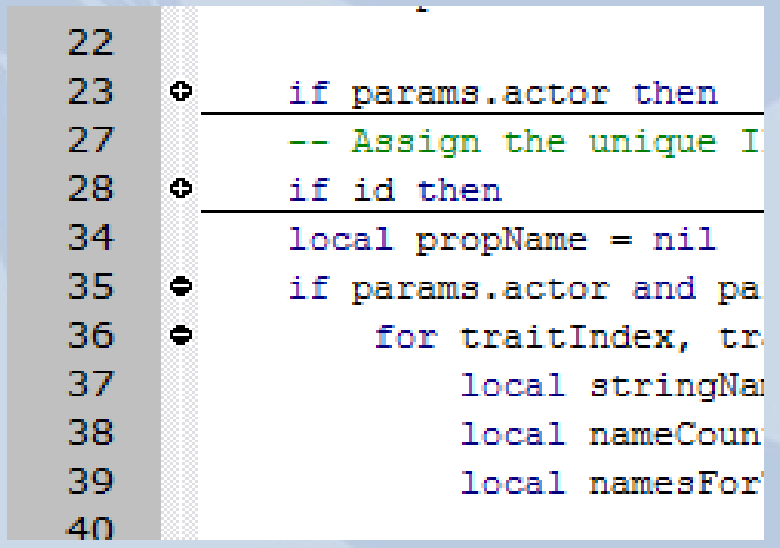
TANTALUS

Quick Feature Set Overview

- Editor features
  - Source control integration



  - 'Folding' editor

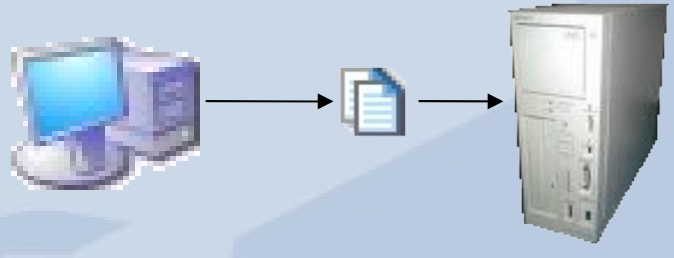


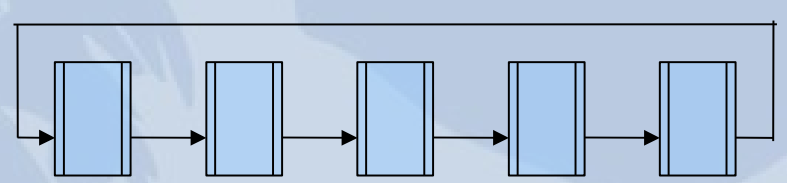
  - Find and replace
  - Find file in project
  - Customizable project file formats

TANTALUS

Solutions using Lua and Tilde

- Uploading of new code to running game
- Time budgeting via round robin scheduling
- Writing game logic using co-routines and blocking conditions

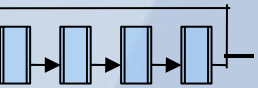




```
-- hide all panels after 3 sec : KEN
if not self.killTutorialThread then
        self.killTutorialThread = thread.create(
                function()
                        thread.sleep(3)
                        self:HideControllerTutorialPanel()
                end, "HUD:AddControllerTutorialPanel")
end
thread.detach(self.killTutorialThread)
```

Solutions using Lua and Tilde

- When Lua code causes an error it just stops running, so



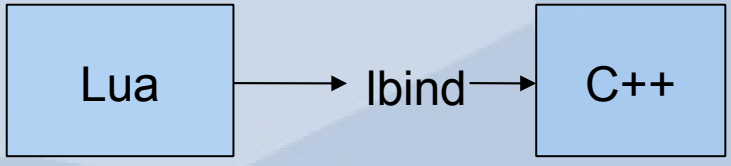
  - a bug in an unrelated part of the game to that being worked on doesn't cause the game to halt

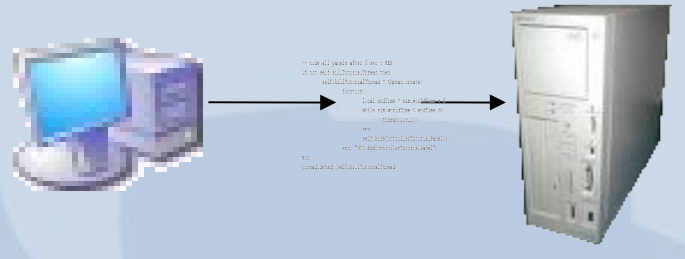
  - errors in code under development can be recovered from, allowing the programmer to try a new version without resetting the game

  - Lua will eventually garbage-collect objects when code is stopped unexpectedly

How using Lua and Tilde has changed the way we make games

- We write game code in two layers: native and script with custom Lua binding glue
- Tilde lets us execute code snippets we enter on the PC, allowing us to experiment or recover from errors (edit and continue that works)





TANTALUS

How using Lua and Tilde has changed the way we make games

- Development builds were much more stable than with other projects, so the game was always playable to some extent.
- Runtime Duck typing frees us from narrow inheritance hierarchies.

How using Lua and Tilde has changed the way we make games

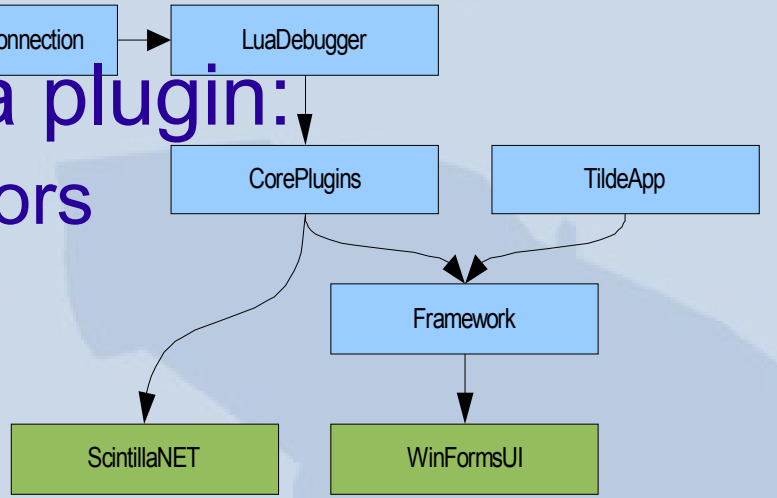
- Coding standards and patterns needed because of less strict compiler checking. This sounds negative but is a good thing.
- Tilde lets us manage the usage of threads in Lua.
- Tilde is a minimal extensible framework for us to embed our own tools; the Lua debugger is itself a plugin
- We could develop the entire game within Tilde, including C++ coding (IDE replacement)

Dealing with Lua issues

- Garbage collector performance
  - We run the GC incrementally under a time budget so it always takes a fixed portion of the frame; it completes a cycle every 5 to 10 seconds.
  - Destructors don't execute immediately, so we need to dispose resources when we have finished with them (change our low-level resource handling a bit).
- Multi-core performance
  - Non issue, Lua shouldn't be using more than one core. LVM is not really re-entrant.
  - Via the co-routines makes it easier to schedule multi-core C++ service code.
  - C++ Thread scheduler needs OS thread mutexing.

Tilde Architecture

- Open source libraries are used extensively:
  - Scintilla - fully featured syntax-highlighting text editor control
  - DockPanel Suite - Visual Studio-style window docking
- Architecture is pluggable and is based on a model/view framework
- Almost everything in Tilde is a plugin:
  - Documents and document editors
  - Project file
  - Source control
  - Lua debugger
  - Debugger transport layer



TANTALUS

How to install and implement

- Integrating the lua debugger
  - Add the tilde source files to your project build system.
  - Modify configuration files as required, and patch Lua source.
  - Extend virtual classes so the host code works in your game engine environment.
  - Modify the main loop of your application to instance the host class, and tick each frame.
  - There is an example for a PC implementation.

The Future

- New features
  - 'Intellisense'
  - Breakpoints conditional on Lua expressions
  - Support for more source-control providers (only Perforce atm)
  - Support for more project file formats (only .vcproj atm)
  - Customizable shortcut keys

The Future

- Documentation
  - Full user documentation
  - Full developer documentation
  - Basic maintainer documentation (for people maintaining or modifying Tilde)

Where to get it

luaforge.net/projects/tilde

License

- MIT
- Contribution license not yet

Questions?

- URL's
  - http://www.tantalus.com.au/tilde
  - http://luaforge.net/projects/tilde
  - http://www.lua.org
  - http://en.wikipedia.org/wiki/Duck_typing