Best Practices for Serious LSL Development

Michael Thome (aka Vex Streeter)

The Abstract:
The Scripter's Nightmare list is endless

You've finally worked out that last feature in your latest scripted product and wham! your computer crashes and you've lost hours of work...

Or the new version of your widget has a baffling new bug and you can't figure out what changed...

… So what can you *do* to avoid some of these pitfalls? Well, here are some ideas:

The Introductions: Who are you?

- You already know how to script
- You do complicated projects
- You want to improve your efficiency
- You (probably) don't come from a software development background

The Introductions:
Who is this Vex guy, anyway?

- Co-Author of *Scripting Your World: the Official Guide to Second Life Scripting*
- 20+ years as a Computer Scientist for BBN Technologies

The Canonical Outline

- General Programming Best Practices
- Scripting in-world
- Scripting out-world
- Workflow
- Support and bugfixing
- Wishlist
- Q&A

General Programming Best Practices

- Know your requirements
- Design it before you code
- Reviews
- Version control
- Test for function and performance
- Support plan

General Programming Best Practices (big jobs)

- Coding standards
- Bug tracking
- Portability
- Operating System bugs
- Documentation

Programming In-World: Organize!

- Avoid losing track of object and script versions:
  - sort order just isn't good enough
  - name or describe objects in development with versions
  - take copies and/or take into inventory and re-rez
- Save scripts early and often
  - It doesn't have to compile to save!

Programming In-World: Organize, Organize, Organize!

- Build on land you own
- Extend Inventory management techniques
  - Use what works for you
  - Major scripts should know their version
  - on_rez reporting of version numbers

Programming in-world: Plan for the future!

- Be modular (see Best Practices slide)
- Don't optimize prematurely
- Use multi-state initialization
- Build-in script and inventory updaters
- Include "lockdown" script deleters
- Nagware if customers ignore product instructions

Programming In-World: Safe Testing …

- Use “dead man's” switches for things that move or proliferate
  - timeout (and die)
  - travel too far (and die)
  - detect sim crossings (and die)
  - listen for kill signal (and die)

Programming In-World: … and Debugging

- Beta grid - learn it. know it. love it.
  - expensive uploads
  - sim killers
- Test products in a variety of situations
  - Does it stop working properly?
  - Does it fail gracefully?
  - ... Might it steal money?

~~Programming In-world: Performance~~

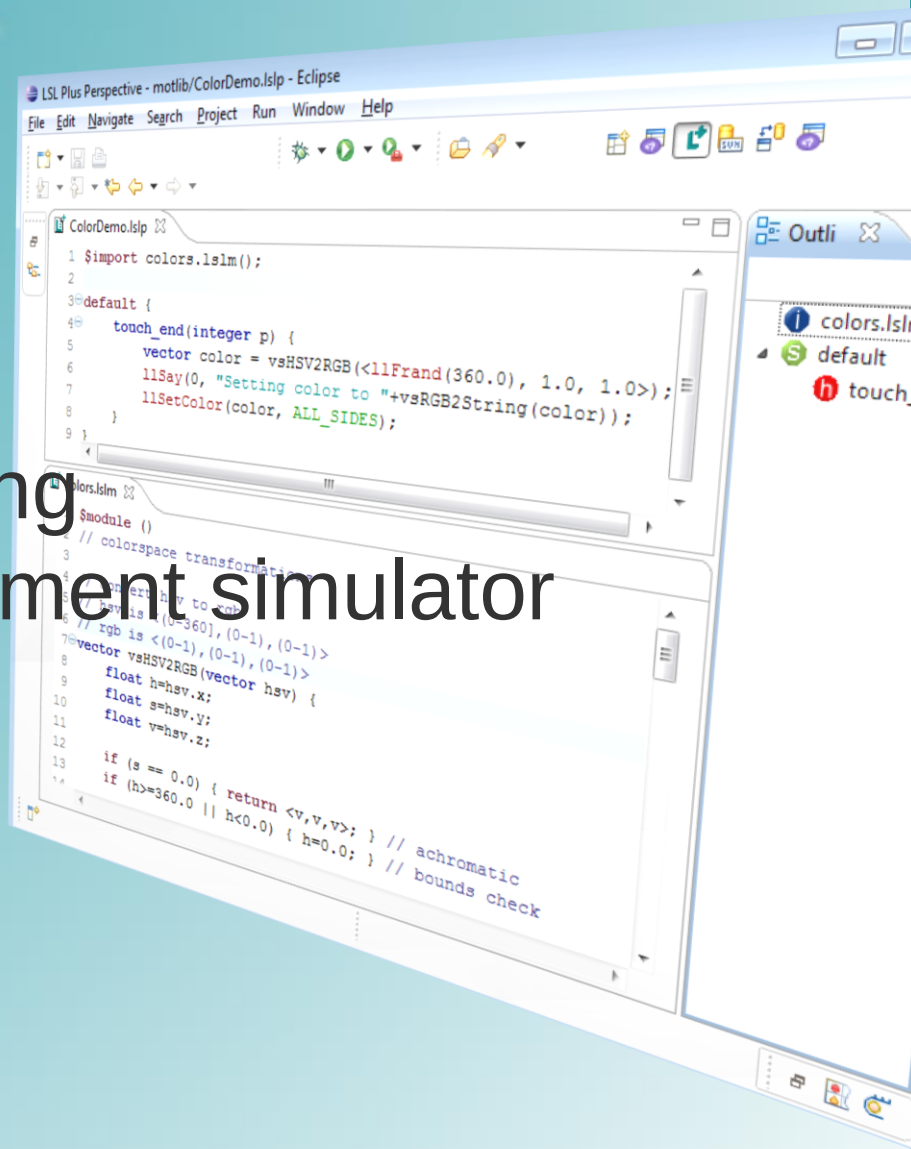
See yesterday's talks:

Slaying the Lag Monster: Best Practices in SL Scripting for High Performance, JB Hancroft

Scripting Under The Hood, Babbage Linden

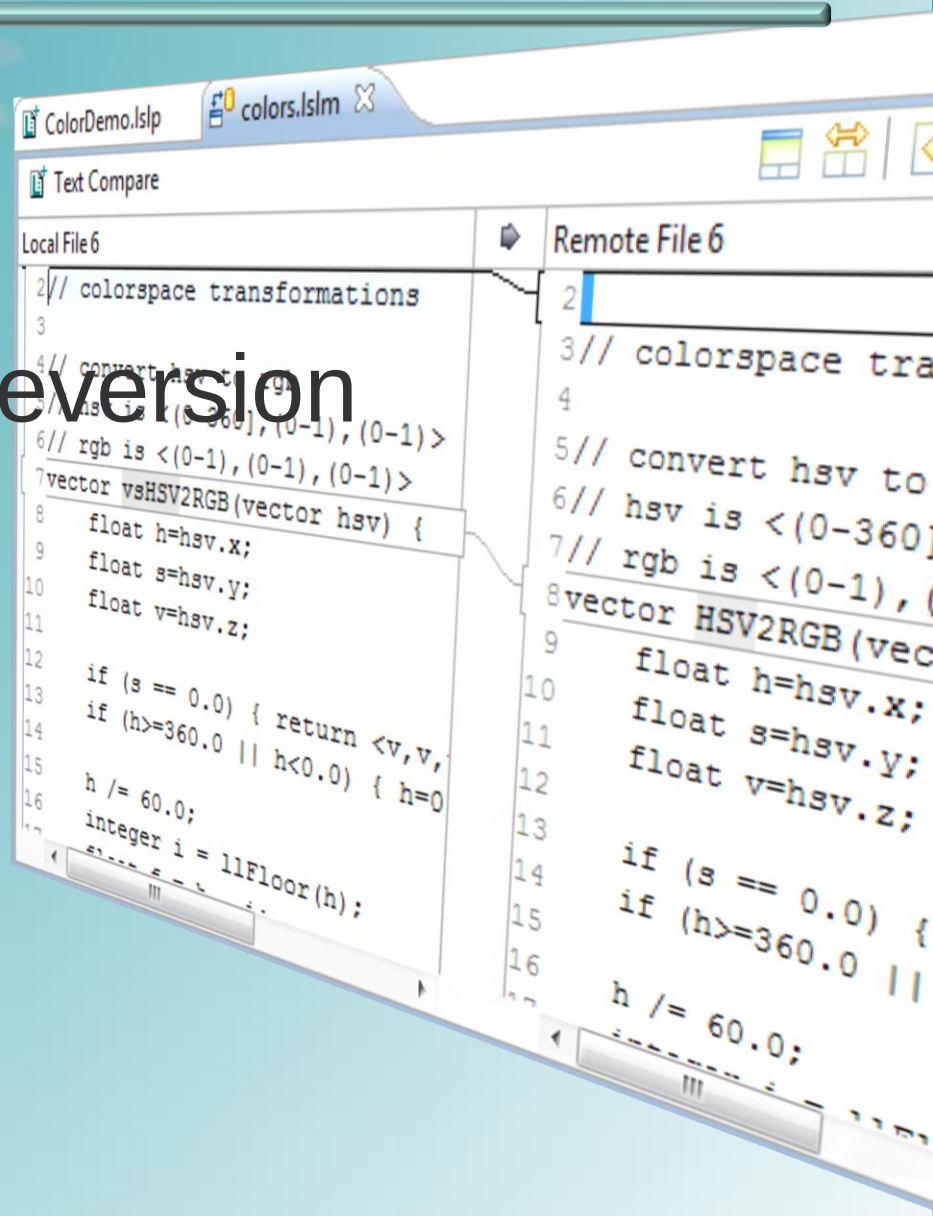
Programming in the Out-world: Integrated Development Environments

- LSL Plus w/Eclipse
  - Eclipse is a serious IDE
  - support for libraries
  - optimization
  - improved syntax checking
  - off-line scripting environment simulator
  - other language support
- Honorable mentions:
  - LSL-Editor
  - ByronStar SL
  - LSL-aware Editors



Programming in the Out World: Version Control? Why?

- Backup
- Change tracking and reversion
- Check-in notes
- Supportability
- Collaboration



Programming in the Out World: Version Control Options

- Subversion
  - Command line: svn
  - Eclipse plugins: Subversive, Subclipse
  - TortoiseSVN (windows)
- Alternatives
  - git
  - cvs
- Picking your repository location

Programming in the Out World: Other Tools

- LSL frontends
  - LSL Plus
  - Emerald LSL Preprocessor
  - lslint
  - historical: ESL, qLab lslcc
- Software process tools
  - testing: LSL Plus
  - bug tracking: bugzilla
  - documentation: doxygen?
  - design
  - requirements

Workflow: the basics

In World
- edit
- save
- update version
- test
- repeat

Out World
- edit
- save
- version
- export to SL
- test
- repeat

Workflow 2: Script updaters

- Take copies of updated objects as backups
  - Version them - never edit your master copies!
- Use viewer export/import functions:
  1. export a basic updater device to disk
  2. run a little script that copies your lsl scripts into the right places in your exported copy
  3. re-import the edited updater
  4. run and update

Wishlist

- IDE to in-world bridge:
  - delivery of scripts
  - some analysis/debugging
- Full object backup (with proper perms) including scripts
- Better front-ends
- Better portability across different virtual worlds

Wrap up

Contest

Drop me a notecard with something I should have included in this presentation but didn't. A lucky responder will get a signed copy of *Scripting Your World*

Copies of the slides and other related information:

http://syw.fabulo.us/wiki/SLPro2010