

## Gigs' Corner

# Funding Second Life Open Source Development

by Gigs Taggart, Correspondent



Gigs Taggart (Jason Giglio) is an active member of the Second Life community as a scripter and contributor to the OpenSL project, land developer, and proprietor of GT HQ Store and Arcade.

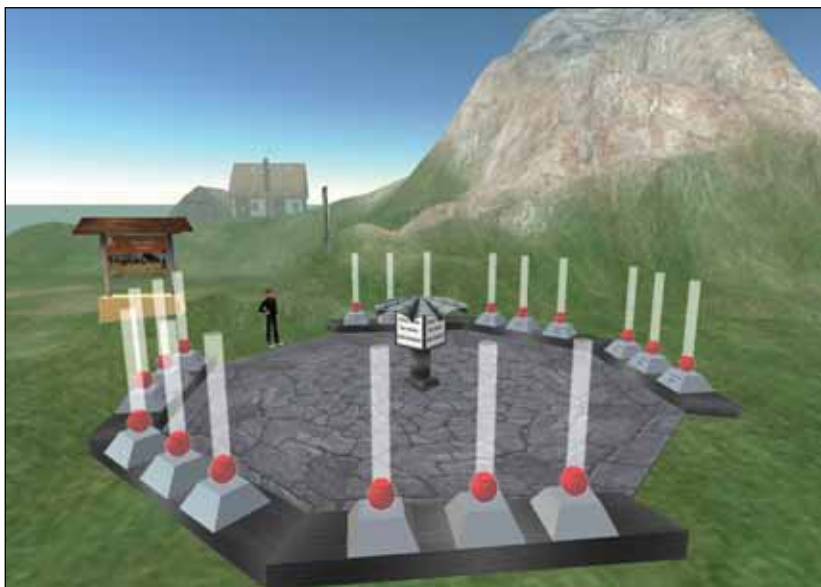
**One of my pet projects is to develop the patronage business model for the Second Life (SL) open source project.**

When people think of Open Source Software (OSS) business models, they normally think of a company that releases a product for free and offers software support contracts to make money. A user requests features and bug fixes from the managing company, and the company then pays their internal developers to write the fixes.

While this traditional business model does work—and is very popular—it isn't the only possible business model when using open source software. Because an open source software repository can be used to accept qualified fixes and enhancements from any developer or development team, a company that uses an OSS product can use their own developers to write a patch to submit to the managing company for the open source project. This model is common. A good example is SGI, a leader in high-performance computing, who used their internal developers to merge the SGI XFS file system with Linux.

The model of building on an existing open source code base is not limited strictly to companies. Individual end users or groups of end users can get together to pool their resources and hire third-party developers to provide code to the managing company. This is called the “patronage business model” of open source development. It is similar to a company having their internal developers submit code to a project, except it is more loosely coupled.

One of my pet projects is to develop this patronage business model for the



Open Source Funding Kiosk on Hippotropolis  
(131, 134, 29)

Second Life (SL) open source project.

With Second Life, there is a diverse group of end users, many with a vested interest in SL working correctly. For many users, SL represents part of their income. Fixing bugs and adding features can directly impact many users' bottom line. It makes sense for these users to pay third-party developers to fix things that Linden Lab might not be motivated to fix in a timely manner.

The key to this strategy is to develop “safe” patches that Linden Lab is able to accept into the main client source repository without breaking the build. Maintaining a custom viewer fork (off the main repository) with the fixes in it is a lot of work. Projects like the Nicholaz edition of the SL viewer (Avatar Nicholaz Beresford is known to the open source community as “the mad patcher”) help a lot to get open source patches out to users more quickly, but in the end the goal should be Linden Lab acceptance of the patches into the main SL viewer code base.

To make this happen, I am creating a kiosk on my SL land on the open source island Hippotropolis. I will nominate bugs or features that I think can be fixed

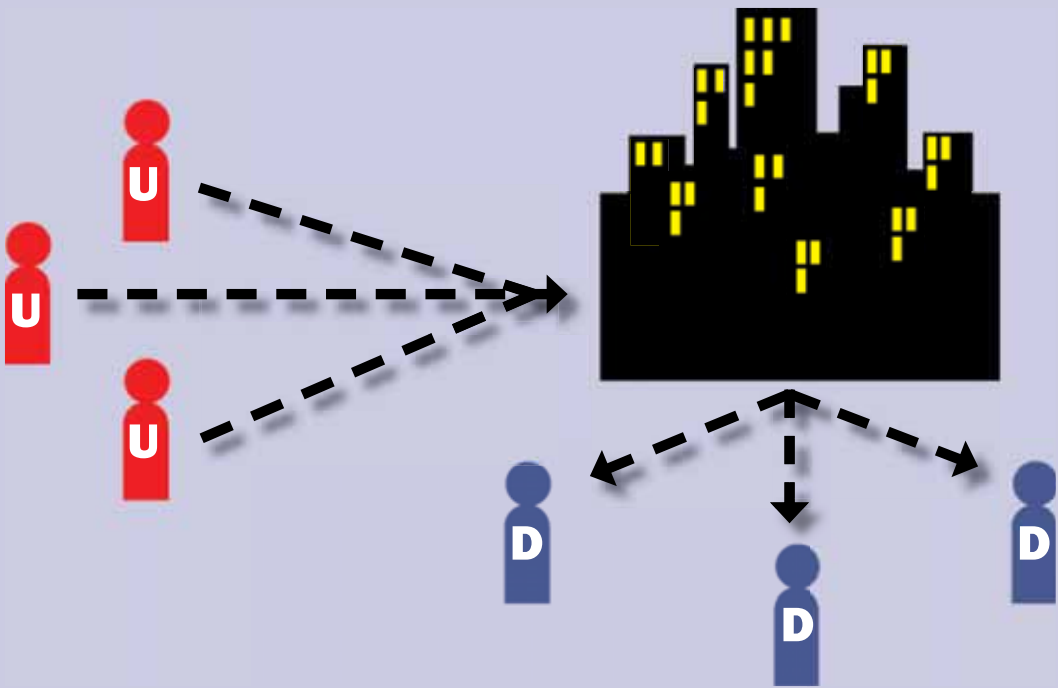
by open source developers, and that Linden Lab will be willing to merge into the main code base.

I will then place donation machines on my land to track the amount of money donated toward each bug or feature. For the first phase of this, I will be the only developer participating in this program. Once all the kinks are worked out, I will invite other major reliable open source contributors to participate.

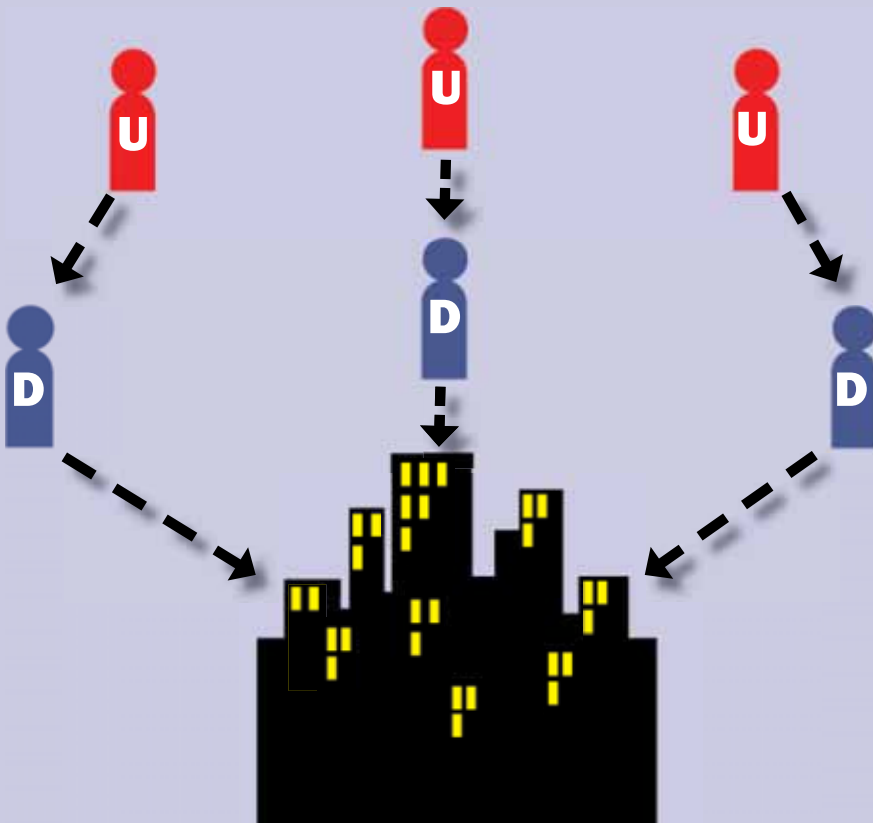
This does raise some questions. What if Linden Lab fixes the bug before the open source developer does? What if Linden Lab refuses to accept the fix? I considered refunding the people who donated money in cases like these, but decided against it. The overhead of tracking every donation and the expectations created by such a refund system are too complex. Instead, if for some reason the bug is resolved in an adverse way, I will donate the money to an open source related charity, such as the Open Source Initiative or the Electronic Frontier Foundation.

For more information, visit the Open Source Funding Kiosk at Hippotropolis (131, 134, 29). ☼

# Open Source Software Business Models



In this model, a user requests features and bug fixes from the managing company, and the company then pays their internal developers to write the fixes.



In this model, individual end users or groups of end users can get together to pool their resources and hire third-party developers to provide code to the managing company. This is called the “patronage business model” of open source development.