

# The Developers

by Surfdaddy Orca



Surfdaddy Orca, editor-in-chief of The Seventh Sun, is a real world surfer with a love for science and technology writing. A former Intel employee, he is a veteran of several Internet startups and specializes in software quality infrastructure.

This is the second in a series of articles on the recent Dr. Dobb's Life 2.0 Summit held inside Second Life (SL) April 28th through May 4th, 2007.

This article covers developer presentations from the Life 2.0 Summit (see box).

Several of these developers (Timeless, Hiro, and otakup0pe) are part of the Second Life Developer Archipelago (SLDA), an effort to organize "the talented developers of Second Life into one locality as a hub of business development and community."

**Dr. Dobb's Life 2.0 Developer Presentations**

- "Physics of Second Life"*  
Bradford Russell
- "Physics of Weapons"*  
Timeless Prototype
- "Givers, Vendors, Rezzers, Builders"*  
Hiro Pendragon
- "Building Tools and Their Technology"*  
Skidz Tweak
- "Developer Tools and Resources"*  
otakup0pe Neumann
- "SL-Web Integration"*  
Felix Wakmann

The intent is to foster communication and collaboration between developers. Scripters, builders, designers, and content creators are welcome to house their homes, stores, and work areas at the SLDA to show off their wares and talents, to work, and to collaborate on projects.

Other developers at the Summit included Vyrnox Ming, Cubey Terra, and Aimee Weber. Vyrnox Ming's presentation on Chisel was discussed in the May issue of the Sun. Cubey Terra provided a hands-on demo of flight in Second Life, but did not go into the technology behind it. Aimee Weber was involved in a panel discussion of the 3D UI and did not do a separate presentation on her highly respected, well-known work.



Panel discussion at Dr. Dobb's conference - (from left) Babbage Linden, Jeffronius Batra, Alem Theas, Dobbs Fredriksson, Eddy Stryker, John Zhaoying

## BRADFORD RUSSELL

### Physics of Second Life



Bradford Russell

Bradford Russell is one of the Dr. Dobb's Journal crew, one of the hosts for the Summit. He gave an overview of the physics of motion and energy in Second Life, which uses the proprietary Havok Physics™ engine. According to the Havok web site, Havok Physics provides comprehensive SDK support for the following areas:

- Collision Detection—including Continuous Physics™
- MOPP™ Technology - for compact representation of large collision meshes
- Dynamics and Constraint Solving
- Vehicle Dynamics

- Data Serialization and Art Tool Support
- Visual Debugger for in-game diagnostic feedback

MOPP (Memory Optimized Partial Polytope) is a way of compressing bounding volume data for arbitrary polygon meshes. It has a dramatically reduced memory footprint compared with traditional bounding volume approaches without sacrificing any runtime performance.

According to Bradford, underneath the Havoc layer there is a polygon database provided by Linden Lab. The physics model (including center of mass, velocity, torque, and energy) is built on top of Havoc and the database server. It currently delivers 45 frames per second. This model controls avatar motion, bumping, and script behavior by calculating the next frame of position and the number of polygons. The results are rendered on the user's computer screen.

The simulation elements for motion include:

- Client-side motion (tasking for flexible prims, for example)
- Non-physical motion (doesn't have to be realistic)
- Impulse motion (push once, what happens afterwards is controlled by physics)

## ► Developers

CONTINUED FROM 14

- Continuous force
- Targeted  
(SL decides how to get from point A to B)

“Particles” in SL are sprites rendered flat using functions related to the LSL built-in function `llParticleSystem()`. Non-physical motion involves `llSetPrimitiveParams()` and `llSetPos()` which centers the object, sets a non-primitive flag, and tells the simulator to be “not interested in the physics model.” Because of Havoc, this works only 10 meters in any direction, although `llSetPrimitiveParams()` can chain 10 meter segments.

Energy involves a motion scaling factor that controls how objects gain energy over time. There is a scaling factor for any motion requested: “if you run out of energy, you stop.” You gain energy over time, and the trick is to move more slowly than the energy being replenished.

Impulse motion involves `llApplyImpulse()` and `llPushObject()` according to the following formula:

$$E(\text{lost}) = \text{llVecMag}(\text{force}) / 20,000$$

`llVecMag` returns a float that is the magnitude of the vector (the distance from vec to  $\langle 0.0, 0.0, 0.0 \rangle$ ). Hence the energy lost is equal to the magnitude of the force vector divided by a constant of 20,000.

A continuous force is set using `llSetForce()`, and targeted motion involves the following built-in functions:

- `llMoveToTarget()`
- `llTarget()`
- `llSetHOverHeight()`
- `llGroundRepel()`

External energy sources (which are not under script control like the other classes of motion) are gravity, wind, collisions, and friction.

Bradford concluded his presentation with the comment that the physics model “can change with next upgrade” of SL.



Both motion and energy are required to shoot a pinball in “Meteor Storm” and “Stellar Dogfight” by Aindreas McGee. Both can be found at Aindres Arcade & Games.

```
{
  state_entry()
  {
    llSetStatus(STATUS_PHYSICS, TRUE);
    // Apply realworld physics.
  }
}
```

A simple example of Linden Scripting Language (LSL).

### TIMELESS PROTOTYPE

#### Multi Gadget and Weapons



“Physics of Weapons”  
Timeless Prototype

Timeless Prototype is a veteran SL programmer and is responsible for the infinitely useful Multi Gadget tool. Here are a just few of the capabilities of Multi Gadget:

- Gives the owner a demo copy of the Timeless EVA Suit (vehicle).
- Gives the owner the Reuters HUD.
- Gives the owner a Multi Gadget Rad Droid.
- Enables the Multi Gadget Builder’s Camera.
- Gives the owner the DSD (Data Storage Device).
- Gives the owner the Multi Gadget Grapple Hook.
- Gives the owner a Multi Chair Coffee Table.
- Turns on/off the channel listener.
- Sets the tuner to listen to channel 1.
- Enables the thought bubbles for AFK, scripting, teleporting, or busy.
- Turns on the online checker to be notified of online status changes for the UUID provided.
- Rezzes a flare, tear gas canister, light, light unit, sky station, temporary catcher, bridge, swing gate, permanent platform, hectic light storm, wooden bridge, or scripiter’s Zen Mat.
- Adjusts avatar flight assistance.
- Turns off/on agent sensing.

#### What a tool!

Timeless provided a high-level presentation on the physics of weapons: “Essentially the weapon is just a bullet rezzing script, however to help with immersion the weapon can be designed to almost any level of realism you’ve got time for.” He said that it “does not matter technically what the weapon

# ► Developers

CONTINUED FROM 15

looks like, how it's rotated or where it is attached. You could have sunglasses that fire bullets for example."In addition to weapon design, Timeles said that weapons systems include health, damage, weapon control, avatar animation, effects, energy, bullets (or other projectiles like the flying monkeys that were shot at Timeless during his presentation), and targets.



Monkeys, as well as bullets, can be shot from weapons

The LSL built-in functions used in weapon construction include:

**Health:** Set on a per-parcel basis, World > About Land > Options tab > uncheck Safe. A heart appears at the top of your screen. No specific built-in functions.

**Damage:** Reduce an avatar's health when a prim and avatar collide with each other at high speed. Involves functions related to:

```
llSetDamage(integer percent)
```

**Weapon Design:** Can look like anything, attaches to an avatar, and is essentially a projectile rezzing script. No specific built-in functions.

**Weapon Control:** Check for mouse click in mouselook mode. Involves functions related to:

```
llRequestPermissions() llTakeControls()
llGetRootRotation()
```

**Avatar Animation:** Make an avatar appear to aim a weapon. Poser® is a good product for creating BVH animation files. After obtaining permissions to animate an avatar, involves functions related to:

```
llStartAnimation()
llGetAgentInfo()
```

to detect mouselook mode.

**Effects:** Enhance weapon realism with smoke, muzzle flash, and so forth. Involves functions related to:

```
llCollisionSound() llSetStatus(STATUS_PHYSICS, TRUE)
llParticleSystem(), llSetAlpha()
llSetPrimitiveParams() llSetLinkPrimitiveParams()
```

**Energy:** Reduced when an object is rezzed and measured using llGetEnergy(). Larger objects drain more energy. Linden Lab provides a "grey goo" mechanism to prevent server resource depletion when rezzing too rapidly (for example, a Tommy gun that shoots too many bullets too fast).

**Bullets:** Have a script to initialize a "bullet" prim so that it will be affected by physics (essentially gravity) and not return to the owner if it goes off world.

**Targets:** Can use a "collision" event to detect when an object collides with it. Involves the functions:

```
llVecMag(llDetectedVel(0))
llGetObjectMass()
llSetRot()
llSetPos()
```

Player versus player collisions involve:

```
llDetectedName(0)
llKey2Name(llGetOwnerKey(llDetectedKey(0)))
```

Timeless provided a sample pistol with a complete weapons script and comprehensive notes that are available at Dr. Dobb's Island.

## HIRO PENDRAGON



"Givers, Vendors, Rezzers, Builders"  
Hiro Pendragon

"These sort of tools should be free" says Hiro Pendragon of givers, vendors, and rezzers. Hiro, dressed as a cyberpunk samurai with vibrant red hair, is another Second Life veteran programmer. His name is on almost every free vendor that is available to newbies.

Givers, Hiro explains, control what piece of clothing, prefab, or other object to give, whom to give it to, when, and how to give it. The LSL built-in functions to do this are:

```
llGiveInventory()
llGiveInventoryList()
```

Vendors, on the other hand, "give away more than one thing out of one thing." Holovendors provide a 3D representation rather than a picture, while a basic 3-prim vendor with display arrows is used with images.

The built-in functions involved in vendors include:

```
llLinkedMessage()
// to object - forward, backward
llSetTexture()
llGiveMoney()
```

(See the example on the next page.)

```
link_message(integer link, integer num, string mes, key id)
{
    if (llGetPermissions() & !PERMISSION_DEBIT) state broken;
    if (mes == "NEXT") // Link forward
    {
        display += 1;
        if (display >= llGetListLength(items)) display = 0;
        llSetObjectName(name + llList2String(items,display));
        llMessageLinked(LINK_SET, DISPLAY_STRING, moneyText(llList2String(prices,display)), "");
        llMessageLinked(LINK_SET,1,"PIC",llGetInventoryKey((key)llList2String(pics,display)));
        llMessageLinked(LINK_SET,1,"LOADNEXT",llGetInventoryKey((key)llList2String(pics,display + 1)));
        llMessageLinked(LINK_SET,1,"LOADPREV",llGetInventoryKey((key)llList2String(pics,display - 1)));
    }
    else if (mes == "PREV") // Link backward
    {
        display -= 1;
        if (display < 0) display = llGetListLength(items) - 1;
        llSetObjectName(name + llList2String(items,display));
        llMessageLinked(LINK_SET, DISPLAY_STRING, moneyText(llList2String(prices,display)), "");
        llMessageLinked(LINK_SET,1,"PIC",llGetInventoryKey((key)llList2String(pics,display)));
        llMessageLinked(LINK_SET,1,"LOADNEXT",llGetInventoryKey((key)llList2String(pics,display + 1)));
        llMessageLinked(LINK_SET,1,"LOADPREV",llGetInventoryKey((key)llList2String(pics,display - 1)));
    }
}

money(key id, integer amount)
{
    if (amount == (integer)llList2String(prices,display)) // Accept money if amount is correct
    {
        incSold(display);
        llGiveInventory(id, llList2String(items,display));
        llWhisper(0,"Thank you for your purchase!");
    }
    else
    {
        llWhisper(0,"Sorry that is the wrong amount for this item.");
        llGiveMoney(id, amount);
    }
}
```

## ► Developers

CONTINUED FROM 17

Hiro also talked briefly about network vendors that communicate to a centralized server or object. These are used quite frequently as a service within SL.

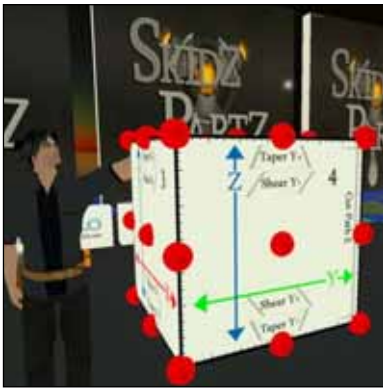
Rezzers are set up for rezzing multiples of something using velocity and energy. They involve the following built-in functions:

```
llRezObject()  
RezObject() or RezAtRoot()
```

In addition to his presentation on givers, vendors, and rezzers, Hiro participated (along with Aimee Weber) in the panel discussion on the 3D UI.

### SKIDZ TWEAK

#### Building Tools



If you've ever tried to align prims and then discovered exposed edges, then you'll appreciate Skidz Primz. Sling on your toolbelt and attach your HUD, and you can align your prims perfectly (in most circumstances).

According to the product documentation, Skidz Primz has tools built into them that allow you to quickly copy attributes from one prim to another. In addition, these tools enable you "to align edges perfectly, resize to fill gaps between prims, and snap corners together flawlessly."

Skidz Tweak founded Skidz Partz in March of 2006. He said that his first idea for a company "was going to be a warehouse of supplies where you could get anything... from jet engines and chairs to trees." He said the he found building in SL "a little difficult" and decided to design a tool that aligned prims much faster "than I could actually align them by hand."

Skidz figured a way using scripts to set the positions of prims "seven places past the decimal." This became the origin of Skidz Primz, "my most successful product."

Skidz Primz includes the following modes:

**Snap:** Enables you to snap corners, edges, and centers together.

**Scale:** Gives you the ability to copy the scale of surrounding objects to the active prim.

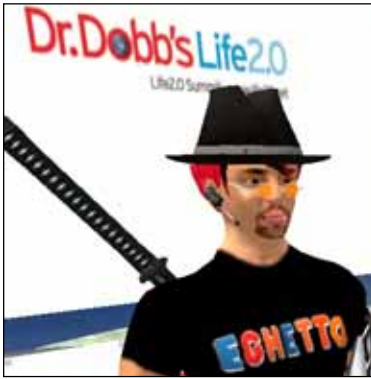
**Rotation:** Allows you match the rotation of surrounding objects.

**Gap:** Is great for filling in gaps between two Primz.

It is hard to beat Skidz Primz for building large structures rapidly and accurately in SL. Skidz also talked about other useful SL tools including "The Prim Finder," a catalog "of practically every shape you could possibly make in SL... 216 of them," and Virtual Builder Studio Pack. Mastering prim cuts, twists, dimples, and effects is an arcane art that requires considerable building experience.

### OTAKUPOPE NEUMANN

#### libsecondlife

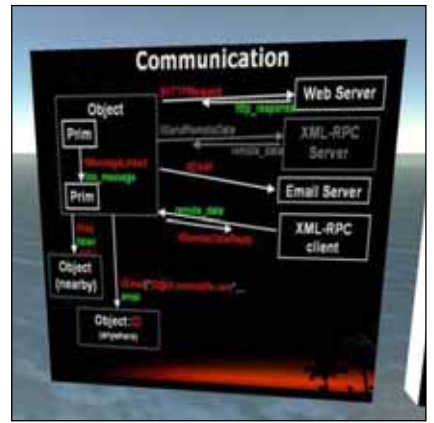


"Developer Tools and Resources"  
otakupOpe Neumann

otakupOpe Neumann is project manager for the libsecondlife project. According to the main page of the libsecondlife wiki, libsecondlife "is an effort directed at understanding how



The copybot debugging tool caused massive protests in SL because of its ability to duplicate objects (see Wikipedia)



SL to web communication occurs using HTTP, XML-RPC, e-mail, and web servers

Second Life works from a technical perspective, and extending and integrating the metaverse with the rest of the web." This includes both client and server technology, as well as third party tools.

At the time of his Dr. Dobb's presentation, libsecondlife was not yet released. Since then, message templates and a developers' portal are now available at:

[http://www.libsecondlife.org/wiki/Main\\_Page](http://www.libsecondlife.org/wiki/Main_Page)

otakupOpe talked about the open source initiative and the classes used for messaging and avatars (chat, teleport, incoming/outgoing messages). libsecondlife includes classes that manage the following:

- Login / logoff
- Send / respond to instant messages
- Send / respond to chat
- Move an avatar
- Focus an avatar on an object
- Teleport to another location

## ► Developers

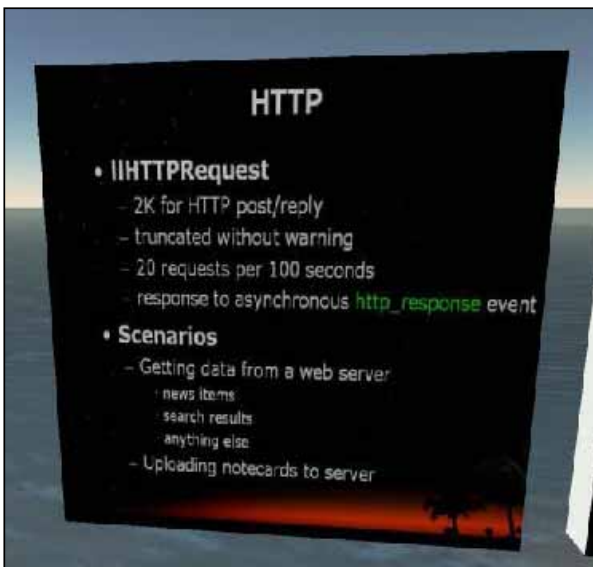
CONTINUED FROM 18

For example:

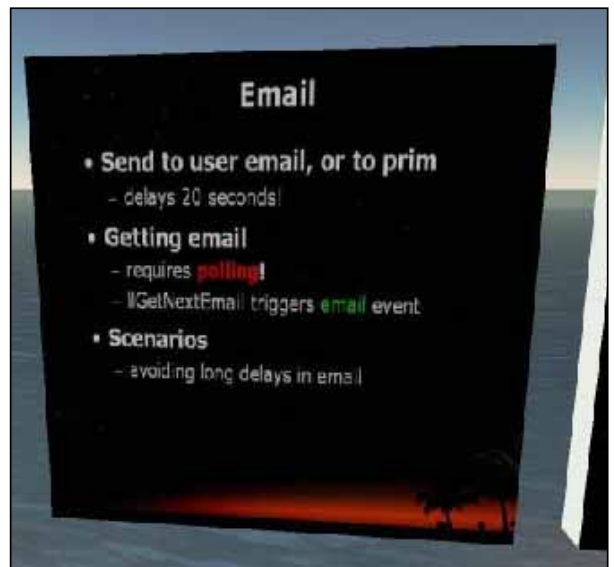
```
client.Self.OnTeleport += new MainAvatar.TeleportCallback(Self_OnTeleport);
// Definition of Self_OnTeleport:
static void Self_OnTeleport(string message, MainAvatar.TeleportStatus status, MainAvatar.TeleportFlags flags)
{
// message contains any messages regarding the teleport
// status is an enum of the current teleport status
// flags is various flags regarding the teleport
}
```

The message templates describe the external interface to SL simulation servers. For example:

```
// Simulator Assignment - Tells a simulator where it is and who its
// neighbors are
{
  SimulatorAssign Low Trusted Zerocoded
  {
    RegionInfo          Single
    {   GridsPerEdge     S32   }
    {   MetersPerGrid   F32   }
    {   Handle           U64   }
    {   UsecSinceStart  U64   }
    {   SecPerDay        U32   }
    {   SecPerYear       U32   }
    {   SunDirection    LLVector3 }
    {   SunAngVelocity  LLVector3 }
    {   IP               IPADDR  }
    {   Port             IPPORT   }
  }
  {
    NeighborBlock      Multiple      4
    {   IP              IPADDR      }
    {   Port            IPPORT       }
    {   PublicIP        IPADDR       }
    {   PublicPort      IPPORT       }
    {   Name            Variable    1   } // string
    {   SimAccess       U8          }
  }
}
```



HTTP requests are used to send data from SL to a web site



Email can be sent to residents or to a prim

## ► Developers

CONTINUED FROM 19

otakupOpe was questioned closely about the copybot “fiasco” in November 2006 during which there was the perception that the “for sale” intellectual property of SL residents became vulnerable to free copying. He said that copybot was designed as a debugging tool for decoding objects, had built-in safeguards, and was compliant with the Linden Lab Terms of Service (TOS).

He said that he sees many educational opportunities for libsecondlife, and that it can be used as the basis of a more sophisticated search engine by a third party.

### FELIX WAKMANN

#### SL – Web Integration

Felix Wakmann is the creator of the in-world RSS feed reader, SecondView. Felix talked about connecting SL (the emerging 3D web) to the existing 2D web. Felix is the author of Smark, a search engine based on the idea of letting people publish anything without having to pay and, as reported by Lukas Chadbourne of SLNN, “search for items not possible using the SL’s built-in search tool” Felix hopes to integrate Google-like capabilities into Smark.

Felix discussed the difficulties in doing component-based development using prims in SL. He compared LSL with its 16 KB script memory constraint (bytecode, heap, and stack) to FORTRAN. He mentioned the difficulties both with in-memory storage (flat lists in LSL) and with persistent storage inside SL (parsing notecards with links that are unreadable).

He introduced the primary mechanism for communication: llHTTPRequest(), with 2 KB for message posting and 2K for data. SecondView uses this LSL built-in function extensively along with the XML-RPC built-in llOpenRemoteDataChannel(). Email communication is done using llGetNextEmail(), which requires polling.

Felix discussed the Smark Search system, which includes a searchable landmark and the user interface — the Smark Search HUD. The interface works partly like a Web browser and is capable of delivering written information since it also indexes full text, such as notecards. The HUD reads pages from the server using notecards to hold the data. Media URL can be used for JPEGs (one per parcel - images from outside SL).

It is worth mentioning that Robbie Kiama of Meta Mart has a similar search HUD called The Meta Mart. Robbie says it’s a step towards “making shopping in SL easier and more fun.” Robbie demonstrated it to Pollywog Gardenvale and me on an individual basis during the Summit.



“SL-Web Integration”  
Felix Wakmann



Felix Wakmann demonstrated the Smark Search HUD and SecondView RSS Reader

Meta Mart provides the following features:

- Search and purchase items using the Meta Mart HUD marketplace
- Create a Meta Mart Account or withdraw funds from it
- Search and purchase items using Meta Mart HUD marketplace

Felix concluded his presentation by pointing out that SL is currently like programming a network of diskless TRS-80 computers from 1975. However, he predicts that “something like Second Life will replace/merge with the [2D] web in 10 years.”

#### Epilog

“In reality, Hiro Protagonist delivers pizza for Uncle Enzo’s CosaNostra Pizza Inc., but in the Metaverse he’s a warrior prince.” Unlike Neal Stephenson’s hero (Hiro) in Snow Crash, a warrior prince in Second Life mostly likely does not deliver pizza in Real Life. He probably sits behind a computer screen, chews bubblegum, and slings a lot of code. And blows big bubbles. ☼



Rissa Maidstone (Life 2.0 event coordinator) and her dragon stand watch over the developers at Dr. Dobb’s Summit