OpenSimulator:

School Quick Start Guide

First Edition: June 2011

By David W. Deeds, Ph.D.,

IT Manager/Teacher

Changchun American International School

Foreword

Your Humble Author would like to thank Changchun American International School (CAIS):

- Directors Daniel and Irene Chou, as well as Principal Mary Pazsit, for providing the opportunity to prove just how great 3D virtual worlds are for education, and without whose support this e-book wouldn't have been possible
- Colleagues and students, for intermittently managing to go 15-20 minutes at a stretch without interrupting so the work could get done

Your Humble Author would also like to thank:

- All of the organizations/individuals mentioned, but the ReactionGrid folks especially: Kyle and Robin Gomboy, and particularly Chris Hart, who served as editor
- CAIS Co-teacher and Honorary Geek Alex Makosz, who contributed and edited
- The OpenSimulator development community, for their dedicated devotion of time and effort toward getting to Version 1.0 (and beyond!), special xie-xie to Justin Clark-Casey
- All the website, ning, blog, wiki, etc., owners who are helping to distribute this e-book and otherwise spread the word about OpenSimulator (and Second Life) in education

*** This e-book is dedicated to the memory of 3D Virtual Worlds Pioneer Chris Surridge. ***

Legal Stuff

This means you are "free to copy, distribute and transmit this work, under the following conditions":

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial — You may not use this work for commercial purposes.

No Derivative Works — You may not alter, transform, or build upon this work.

E-mail it to anybody and everybody. Link to it. Post it on websites, blogs, wikis, etc. Just ensure you mention that it's being provided courtesy of Changchun American International School. You can't make any money with it. You can't change it. Simple enough.

For more information, go to **http://creativecommons.org/licenses/by-nc-nd/3.0/**.

This e-book is being provided to you free of charge via the generosity of Changchun American International School. Here's your part of the deal. As a condition of accepting and using this e-book, you agree not to hold CAIS or Your Humble Author (YHA) responsible for anything that might go wrong. Your Humble Author has personally performed every step in this e-book, and everything turned out to be a stunning success, much like YHA himself. If your computer blows up, it's because YOU [screwed] up. If you do not experience instant success with your OpenSimulator classes, for whatever reason, it's because you need to take a training seminar from YHA. More information coming soon.

Permission was requested and obtained from most of the entities mentioned herein. Obvious exceptions include Dell, Microsoft, Second Life and the International Baccalaureate Organization. All organization names, logos, etc., are trademarks or registered trademarks (or at least soon will be) of their respective entities. CAIS and YHA have nothing but the best intentions, this e-book is being provided free as a public service to education, etc., so we would really appreciate not getting sued for anything. Thanks and see you 'round the grids!

Table of Contents

Table of Contents (cont.)

Introduction/Inworld Visit

The Meaning of Life…

…is a bit beyond the scope of this e-book, but this publication's purpose and its intended audience are worth starting off with.

- **Why was this e-book written?** The existing OpenSimulator documentation is typically disjointed, with snippets of information appearing here and there. To Your Humble Author's knowledge, there's never been a "one stop shop" resource that explains to (K-12, or indeed any) schools how to get started with OpenSimulator, i.e., until now. Also, most of the existing OpenSimulator documentation is written by and for programmers, making it practically incomprehensible to all but the geekiest of geeks.
- **Who is this e-book for?** This e-book is for (primarily but not exclusively K-12) teachers and administrators (network, system, principals, coordinators, etc.) who want to get started with 3D virtual worlds in general and OpenSimulator in particular. A breakdown of which chapters are intended for different roles/responsibilities is coming up on the next page.

Regarding Time and Space

Well, regarding time anyway. This e-book was finished in June 2011. While it was being produced, the latest version of OpenSimulator progressed from 0.7.1 to 0.7.1.1. This e-book's information should be valid for a while, but there's no guarantee that everything will be exactly

the same by June 2012, when this publication's next edition is tentatively scheduled for release. So you might have to "read between the lines" a little. If you like this e-book and you'd like to see another edition in 2012, particularly an advanced version, please contact Changchun American International School (**info@caischina.org**) and say: "This Deeds dude is an absolute genius. Please, please, please have him write another 3D virtual worlds guide." It's especially important to mention that you'd be willing to pay a reasonable price for the next one. If you don't like this e-book, please write to **someone@whocares.com**.

What Are You Supposed to Do?

Just start reading. Where? Depends on who you are and what you want. If you are a:

- Teacher or administrator and you'd like some basic information on using OpenSimulator, including an inworld visit and some building/programming exercises, start here with Chapter I. If you've never used a viewer, been inworld, etc., this is the starting point for you. This might be a good place for everyone to begin, regardless. You just might learn something you don't already know!
- Teacher or administrator already familiar with the fundamental concepts of grids, viewers, etc., and/or you just want directions for setting up a client/local and/or client/server installation of OpenSimulator in your school, start with Chapter II.
- Teacher and you'd like to get some tips on how to use OpenSimulator in the classroom, or rather the computer lab, including real-life lesson plans, go to Chapter III.

Please note, one more time: Although this e-book was written primarily for K-12 schools, a lot of the information, particularly the installation instructions, can be used by any organization.

Overview: Brave New (Virtual) Worlds

In the history of education, there have been three major tool development milestones:

1. Invention of the printing press
2. Mass production of chalk and blackboards
3. Development of 3D virtual worlds

Think #3 is an exaggeration? Well, maybe a little. But according to the **2011 Horizon Report**, an annual publication by the New Media Consortium (**http://www.nmc.org**) and the EDUCAUSE Learning Initiative (**http://www.educause.edu/eli**) that predicts adoption of various educational technologies, "games-based learning," under which designation 3D virtual worlds such as Second Life and OpenSimulator are in the "collaborative" category, will be MAINSTREAM in the worlds' schools, international and otherwise, by as early as 2013. Why wait until then to get started?

> "Proponents of games-based learning in higher education point to its role in supporting collaboration, problem-solving, and communication, the 21st century competencies needed by American students outlined by Secretary of Education Arne Duncan in late 2010 in the National Education Technology Plan. Advocates also underscore the productive role of play, which allows for experimentation, the exploration of identities, and even failure. Gaming also contributes to the development of a particular disposition well-suited to an information-based culture and rapid change."

– 2011 Horizon Report, p. 21 (**http://www.nmc.org/publications/2011-horizon-report**)

If you work with and/or for people who aren't yet sure about this whole 3D virtual worlds thing, then show them this report. It should do the sales job for you!

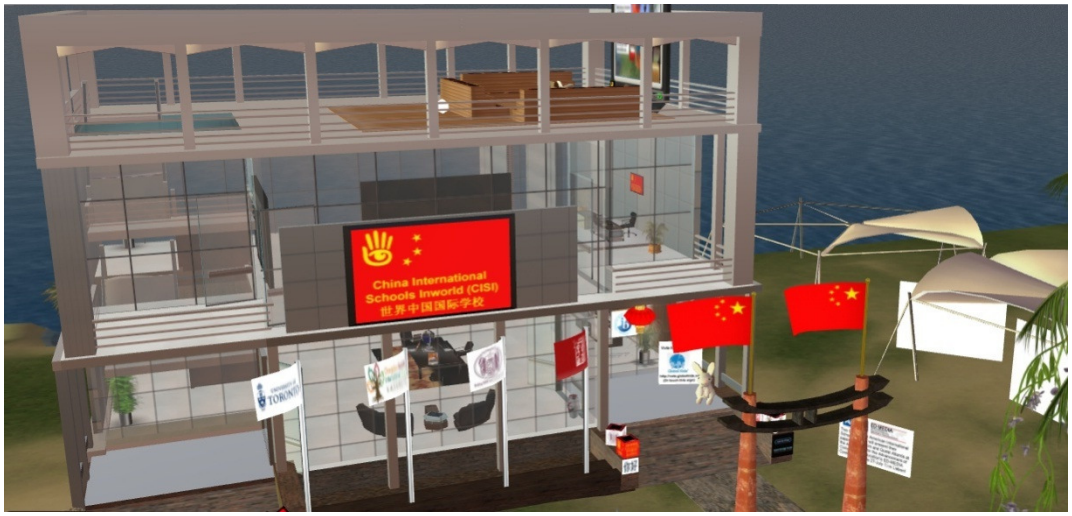
But YOU obviously don't need to be convinced of the fantastic potential 3D virtual worlds represent for education. Otherwise, why are you reading this? You downloaded this e-book because you already know WHY you want to use 3D virtual worlds. You're here to find out HOW your school can get going with them, OpenSimulator especially if not specifically, and without spending years studying toward an advanced degree in computer science/education. You're in the right place. But before you start banging on a keyboard, we need to get a bit more introductory stuff out of the way.

Relax. Take several deep breaths. Everything's going to be OK.

First, OpenSimulator's predecessor, Second Life, must be mentioned. Second Life (SL) is a commercial Massively Multiplayer Online (MMO) game, or a MMO Role Playing Game (MMORPG), or a MultiUser Virtual Environment (MUVE), depending on which Ph.D. candidate you ask. Your Humble Author recommends you don't ask any Ph.D. candidate about this, unless you have lots of time on your hands. Some more expert advice: You and your students can call a 3D virtual world a GAME, but don't tell your supervisors or their parents that! Since SL's introduction in 2003, millions of users have gone inworld, nurtured communities, created fabulous things, even bought and sold commodities via its own economy. However, SL has traditionally been for adults only, and if you teach in a K-12 school, your students are not. Adults, that is.

Since January of 2011, the minimum age for SL access has been lowered to 16, so, if you're in an International Baccalaureate school, e.g., this means Middle Years Program 5 and Diploma

Program students can participate (grades 10-12). (The graphic below is a snapshot of the China International Schools Inworld cybercampus in SL. There'll be more on this and SL in general later.) But this still leaves Pre-Kindergarten to Middle Years Program 4 kids (aged 3-15) needing somewhere else to go.



That's where OpenSimulator comes in. OpenSimulator entered the 3D scene as of 2007. It's often called "OpenSim," although this is the name of a separate product entirely. The OpenSimulator developers prefer "OpenSimulator," so that's the name this e-book uses. Its developers will tell you that it's not just a free, open source clone of Second Life, although many people will inevitably consider it so. Think of it as being like Second Life. It's like Second Life but you can have it your way, and this is just one of the reasons that make it so great for schools!

Second Life is a public grid, which means it's very much like a public chatroom, just in 3D; with OpenSimulator you can have complete control over your private 3D virtual world. No one comes in or goes out without your permission, and when you're trying to convince skeptical parents of younger munchkins on the idea of taking their little darlings into 3D virtual worlds,

this is a great selling point. You have the ability to set parameters you'd never get the chance to mess around with in Second Life: everything from object size to the number of regions (each a 256-square-meter plot, aka a sim or an island) you own. Did Your Humble Author happen to mention it's FREE?

The backend of OpenSimulator is mainly just a big database, SQLite by default. You can "upgrade" to MySQL and even SQL Server Express if you have the skills (and/or YHA's next e-book!). The database conversion is not overly difficult, but it's certainly an advanced topic and not covered here. To see what you and others are doing with the database, in other words, to interact with it, you need a (free) viewer, a special client program. A browser is to the Internet as a viewer is to OpenSimulator. Now the viewers undeniably are clones of the SL version, release 1+ in particular. The OpenSimulator software runs on a server, the viewer runs on clients. So if you have a computer lab with 20 PCs, and you want an in-house setup, you need one server installation of OpenSimulator and 20 client installations of a viewer, such as Hippo or Imprudence (OK, so viewer names can be funny!). If you choose to have a third party vendor host your OpenSimulator installation, you just need the clients with viewers. Everything works

the same, it's just that your OpenSimulator installation is "in the cloud" or accessible via the Internet. More on this later.

Changchun American International School uses both Second Life, for students 16+, and OpenSimulator, for all the rest, 3-15. We've been taking students into our various OpenSimulator installations since 2009 and the results have been nothing short of phenomenal. OpenSimulator has been used primarily for Middle Years Program (aka grades 6-10) Technology classes, but also for the Diploma Program (high school) Information Technology in a Global Society (ITGS) class and Primary Years Program (elementary school) ICT classes.

One of the many reasons 3D virtual worlds are a "natural" for international schools is that students who are less than proficient in the language of instruction can succeed despite the communications barrier. At CAIS it's not unusual to have native speakers of English together in the same classroom as kids who are just starting to learn the lingo. OpenSimulator exercises enable the ultimate differentiated learning environment: the strugglers can excel alongside the fluent, who can readily transcend the minimum requirements to whatever levels they wish, and this means approaching the level of professional software developers. All students learn

sophisticated computer-aided design, programming, graphics, and perhaps most importantly, project management skills. More on this later too. Enough of the sales pitch! You're anxious to get underway, so let's do it.

Getting Started: Assumptions/Conventions/Etc.

Assumptions: About You, Your School

This e-book assumes that you have above average computer knowledge/skills. Perhaps you're the IT/ICT or Technology teacher for your school. Or perhaps you teach another subject, but you're just handy with PCs. You don't need to be a complete geek to follow this e-book's directions. You know the difference between directions and instructions, right? Directions assume you know the vocabulary involved. One chef sharing a recipe with another chef gives directions. Instructions assume you don't understand the terms being used. This e-book gives directions. It takes for granted that you already know what an avatar is, what LAN stands for, etc. If you don't know the difference between a client and a server, e.g., you are probably going to need some assistance performing technical tasks such as setting up the client/server installation. However, there's no reason why you can't understand and do everything else yourself. There really aren't that many new terms you need to know in the beginning. You'll acquire a much nerdier vocabulary as you progress. Relax. Take several deep breaths. Everything's going to be OK. If you want to use a third party vendor, all you have to complete are client viewer installations! No worries in any case.

This e-book assumes that you, like the vast majority of computer users, run Windows as your school's primary operating system. Sorry to all of you Mac (Linux?) schools out there! To use OpenSimulator, you need: (a) the OpenSimulator software, (b) an OpenSimulator viewer, or

(c) both. The latest releases of the OpenSimulator software itself are available for Windows and Linux platforms. Viewers are available for Windows, Linux and Mac. So yes, if you're a Mac (or even Linux) school, you can certainly use OpenSimulator. The OpenSimulator software, if more than one student is to use it simultaneously, which is what you need for classes, must be run on a server. So, you must have a Windows or Linux server unless you opt for a third party vendor, then you don't need a server in-house whatsoever. The viewer must be installed on all client machines, but it doesn't matter if your students use Windows, Linux or Mac clients. This will be discussed further in the topic "Do It Yourself Second Life," which is what OpenSimulator is sometimes jokingly called, much to the dismay of the OpenSimulator folks. Oh yes, the browser used while writing these directions was Internet Explorer, specifically Version 8, so if you use a different browser or version, you might need to adapt a little. Don't worry, you can handle it!

Conventions: What Different Text Means

The conventions used in this course have been made deliberately simple.

If you see text in **bold red**, this indicates something you see on your monitor that you are supposed to interact with in some way. Example:

Click the **OK** button.

This means that you're going to see "OK," it's a button and that you should interact with (that's right, a fancy way of saying click) it.

If you see text in **bold black**, this means that you see it on your monitor but that it's just to be read, you don't interact with it.

Example:

The **Preferences** dialog box displays.

Links to Internet pages, e-mail addresses, etc., are purple and bold. Example:

http://www.educause.edu/eli

If you're to type an IP address into OpenSimulator, though, it isn't underlined:

http://10.10.7.163:9000

Code/file text appears in Courier font so you won't get letters mixed up with numbers:

```
llTargetOmega(<0,0,1>,1,1);
```

Can't get much less complicated that that! Steps to be followed are identified via topic headings but not painstakingly numbered, as this tends to drive people crazy.

Do It Yourself Second Life: Three Choices

You have three choices re: the OpenSimulator software:

- **3RD PARTY/HOST**. You pay a third party vendor to host your virtual world. This means you don't have to buy, maintain or support a server. It does cost money, but it eliminates the need for a big investment up front if you don't already have a server available. And you can save a lot of currency because you don't need personnel to keep things running. This should be your school's first choice if you can afford it, and you have reliable, speedy Internet service. Running a client/server setup requires not just a server itself, but also the personnel who are capable of not just

installing but maintaining it as well. Changchun American International School is small, but due to generous director investments in hardware, a midrange server was available for dedicated use. (Don't panic, your OpenSimulator server doesn't have to be dedicated to OpenSimulator, although this would be nice!) The server's presence saved the day, because although CAIS started with a vendor, a switch to the school server had to be made because of Internet connection problems! And Your Humble Author just happens to have 15 years of computer industry experience, so a system/network administrator wasn't required. The bottom line is that it might be just plain easier if you start with a host. You can always move your OpenSimulator installation in-house later (and remember, you might have to!).

- **CLIENT/LOCAL.** You set up client/local installations on different student PCs. This means the software only runs locally, so that only the student using the PC it's installed on can use it. Although this can be useful for getting started, ultimately your goal should be to have all your students working together. If your school has neither the money for a vendor account nor a local area network/server (and there are international schools out there in this situation), the local/client setup is better than nothing. With a client/local installation, you install the OpenSimulator software and the viewer on each PC. You'll see how this works in the "Step-by-Step School Setup" chapter. With a client/local setup, you really don't have to do anything with the OpenSimulator software except decompress or "unzip" it. It's ready to go with it basic configuration "out of the box," but you won't be content with this for long. You'll want to tweak the installation, and this means that if you have 20 PCs with OpenSimulator, you have to change all 20. Not that this is hard to do, it's just tedious. Now remember, Your Humble Author's bias is in favor of

students working together, but if you just want to try OpenSimulator on a trial basis, this would be the easiest thing to start with. Let each student work on his/her client PC. They can build, program, etc., in fact, do everything except work with other students. If you and your powers-that-be decide you like OpenSimulator, then you can take the next step, the client/server setup.

- **CLIENT/SERVER.** This means you install OpenSimulator on a server and the viewer on clients, such as PCs in a computer lab. This will be the way to go if you can't afford a host and/or don't have reliable, speedy Internet service, but you do have a LAN. Or, as is the case with some schools, your network firewalls disable some Internet access (this is usually a matter of policy, you could always "punch holes" in firewalls to allow users to get through) and as strange as it might sound, even some university professors are in this situation. OpenSimulator grids (a lot of cyberturf regions connected together via OpenSimulator) and Second Life are NOT affected by The Great Firewall of China, just in case you're worried about this. But until CAIS solved its Internet problems, which were due to internal configuration issues, having a client/server installation was the only feasible way to use OpenSimulator. The client/server scenario works very well, with the exception of having to save and load database backups for each class, but this quickly becomes routine. If you don't have a LAN, of course, this won't work, but you could create a "mini-LAN" by connecting a hub to a server and then hooking up several PCs to it. This is beyond the scope of this e-book, which assumes that you have a LAN in your school. If you don't have a server available, your boss is going to ask if one of your regular desktops can be converted into a server. Although the answer is technically yes, this isn't a good idea because it's unlikely the standard desktop/laptop can

handle more than a few concurrent users. With a client/local setup, there's only one user, and your clients can manage this. Once the number of concurrent users gets past one or two, you need a server. More on this later.

Some hosted solution choices will be discussed and compared first, then you'll go inworld.

Getting Your OpenSimulator Cyberturf: Lots of Choices

There's good news and bad news when it comes to choosing a third party vendor for hosting your OpenSimulator installation. Let's just say "host" from now on, OK? The good news is that you have lots of choices re: hosts. The bad news is that you have lots of choices re: hosts. Maria Korolov's **Hypergrid Business** ezine is a great resource for keeping up with what's happening with 3D virtual worlds and the publication always features a list of hosts that you can consult:

http://www.hypergridbusiness.com/OpenSimulator-hosting-providers

Obviously, this e-book can't cover all of these choices. Instead, three have been selected:

- ReactionGrid
- Jokaydia Grid
- Dreamland Metaverse

These three have been chosen, first of all, because they were closely analyzed for a recent educational technology conference at which Your Humble Author presented. But there's more to it than that. First, these three grids have been recommended to YHA for various reasons by other 3D virtual world professionals. These three hosts offer both private grids (grids are cyberturf regions connected together, remember?) and public grids (these are lots of different

people's grids connected together). You're wondering why one thing can have three words for it and three other things can have only one word for them. Welcome to 3D virtual world vocabulary. You'll get used to it. Just remember that a private grid means you and your students get in. No one else. And no one can get out, either! Unless you opt for hypergridding, which is the process of "jumping" from one grid to another. A public grid is one like 3rd Rock Grid (**http://www.3rdrockgrid.com)** or OSGrid **(http://www.osgrid.org**), which consists of hundreds of computers running OpenSimulator connected together. You'll be going inworld soon, so this will make sense then if it doesn't now.

But in the meantime, consider that a public grid potentially presents the same essential problem that Second Life does, namely that anyone can join. Second Life was for adults only for years for a very good reason, namely, there were some unusual (some downright weird!) residents doing/saying/showing stuff that was inappropriate for kids. Thus choosing a host isn't simply a matter of comparing costs. You should select one that is "PG," i.e., everyone using the grid knows that everything has to be "kid-friendly." If you only want a private grid, of course, this isn't a consideration, but if you only want a private grid, shame on you. Half the fun (not to mention the learning experience!) of getting your students into 3D virtual worlds is allowing them to connect, communicate and collaborate with other learners around the world! The list of choices on **Hypergrid Business** ezine seems long, but after you eliminate the ones that allow "adult content" your options are much more limited. Many of the new grids on the scene were created for a certain marketplace niche, namely adults who are looking for somewhere to go with less of the restrictions Second Life imposes. They're not interested in accommodating kids or schools. You should choose from among the grids that specialize in educational needs, institutions, etc.

When you first start talking to your managers about costs, you're going to be under pressure to choose the host with the lowest price. Your Humble Author knows this because of potent psychic powers. No, because YHA knows just every school these days is running on a tight budget, or at least managers are going to tell you this, whether or not it's actually true. But there are other factors to consider, and one of the main ones (besides being "PG") is customer service. Many of the hosts available are of the "mom and pop shop" variety. In computer terms, this means three nerds sharing a house, with some servers in the garage. Will they still be in business years from now? And in the meantime, are they able to offer prompt troubleshooting and additional services? Just a few things to think about beyond the price tag. Oh yes, look for vendors that offer discounts to educational organizations, something else to consider beyond the initial quotes.

ReactionGrid is Changchun American International School's host. Jokaydia and Dreamland were recommended to Your Humble Author by various OpenSimulator gurus when it came time to put together a conference presentation. Again, take a peek at the **Hypergrid Business** list and you'll see you have many alternatives. Use the information in the following comparisons as a guide to making your own choice. Don't be shy about asking hosts questions, they will be glad to make a sales pitch. Please note that the price comparisons use quotes that were in effect as of June 2011. They are subject to change so please don't call these vendors and tell them you saw cheaper prices here! OK, enough with the disclaimers, let's get on with it:

- **REACTIONGRID.** You can have your own private grid and/or have regions in the ReactionGrid grid. CAIS has both, a four-region private grid, plus a region in the ReactionGrid grid itself. We can hypergrid back and forth. This means if we want complete privacy, we can turn hypergridding off and work on our private grid.

When we want to visit others, we can turn hypergridding on and go to the ReactionGrid grid. Or we can just log directly into ReactionGrid. You don't need both: you can have one or the other. If you've never visited a grid before, you're wondering what the heck all this means; relax, take deep breaths, etc., you'll soon be going inworld. Check out the various plans ReactionGrid offers (if the page has changed, go to **http://www.reactiongrid.com** and follow links):

http://reactiongrid.myshopify.com/collections/virtual-worlds

If you can't get management approval right away for any of the more extensive plans, you can pay a setup fee of a couple of hundred (that's American) dollars, and then a monthly fee, usually under one hundred, for a four-region private grid. ReactionGrid's grid is guaranteed "PG," or kid friendly. ReactionGrid also offers Jibe server accounts. Jibe is an alternative platform for producing 3D virtual worlds that can be embedded in a web page (**http://jibemix.com**), which is way beyond the scope of this e-book, but hopefully will be the subject of a future one by Your Humble Author. ReactionGrid has an online technical support system, with technicians in various locations around the world so time zones aren't much of a hindrance to getting help.

For more information, go to **http://www.reactiongrid.com** and click **Virtual Worlds** or e-mail Kyle Gomboy (**create@reactiongrid.com**).

- **JOKAYDIA.** Jokaydia started as part of ReactionGrid but is now a completely separate grid, specializing in educational organizations. Because of this, naturally, Jokaydia is guaranteed "PG" or kid-friendly. Jokaydia offers both Second Life and

OpenSimulator cyberturf, so it's one of the few vendors, if not the only one, which allows you to get both from one source. If your school wants both a Second Life and OpenSimulator presence, this is a very important factor to consider. You could have just one "landlord," which should make payments simpler. All of your cyberneighbors will be schools or other educational institutions, and a lot of people especially like this aspect of renting a Jokaydia sim. No businesses or other organizations, just teachers and learners.

http://www.jokaydiagrid.com/rent-a-sim

Jokaydia's costs are lower in the beginning because the initial, one-time setup fee is only US$50. Around US$25 per region per month is the norm for the Jokaydia OpenSimulator grid. Remember that sim equals region equals island! You rent certain lots of cyberturf in SL, not entire regions. They can make your own private grid as well; contact them for prices. Jokaydia supports and encourages hypergridding between different grids for educational purposes. Jokaydia has an online technical support center.

For more information, go to **http://www.jokaydiagrid.com** and click **About** or e-mail Jo Kay (**joannamkay@gmail.com**).

- **DREAMLAND METAVERSE.** This vendor features no setup costs at all, and a slightly different pricing scheme that accounts for the fact that often with 3D virtual worlds it's not the number of regions that matter as much as the number of avatars you have on a region at any given time. The effect caused by having too many avatars on one sim at one time is called "lag" and it can bring your class to a

standstill if it's bad enough. Dreamland offers regions that are called "residential," assuming a maximum of 40 avatars, and "professional," assuming 80 avatars. If you plan to have large numbers of students inworld simultaneously, this is an important factor to consider. Ask other vendors about the maximum number of avatars their regions can handle. Dreamland can connect you with one of the biggest grids, OSGrid (**http://www.osgrid.org**), or set up your own private grid. A residential region will cost around US$45 a month, a professional around US$90. Dreamland also has partners offering a wide variety of 3D virtual world services, so if you need building, consulting, programming, etc., done, they can connect you with other vendors.

For more information, go to **http://www.dreamlandmetaverse.com** and click **About** or e-mail Snoopy Pfeffer (**snoopy.pfeffer@yahoo.com**).

Confused yet? Probably, but you should know by now: relax, take deep breaths, etc. Take your time choosing a host. Make a comparison/contrast grid for your organization and then rank your priorities. What's the most important thing to your school? Cost? You've been warned about focusing on prices, but Your Humble Author knows that's what your managers are most concerned with. Performance? Community? Other services? Again, don't be shy when it comes to asking vendors questions. They're used to dealing with 3D newbies, or "noobs" as you're known. Ask around before making a choice, and this means requesting to speak with existing customers. 3D virtual world users tend to be a very helpful bunch, because they remember being a noob like you is not easy. They're also very good about giving you frank advice, just be sure to heed it.

Going Inworld (the First Time?)

Please note that ReactionGrid is being used as the example for these exercises because this is the vendor used by Changchun American International School and thus is most familiar to Your Humble Author. As mentioned previously, there are other grids you may visit (OSGrid, 3rdRock, on and on) by following the same basic steps for creating an account/avatar, as well, of course, as following exactly the same steps for installing a viewer. If you can't use Hippo (or just don't want to!), use another.

Having a basic avatar/account in ReactionGrid is free, just as in Second Life. You can explore, you can interact and you can even build in special areas called sandboxes. More on this later. There's no premium membership as Second Life has, if you're familiar with this. You need to spend money when you want a region (or two or three) of your own. Most of the other public grids out there work much the same way.

Going inworld (the first time?) involves these steps:

1. Creating Your Account/Avatar
2. Checking Viewer System Requirements
3. Downloading and Installing Viewer

As you should know by now (and if not, wake up!), to visit an OpenSimulator grid or just plain old region, you must use a viewer. This e-book covers the Hippo OpenSimulator viewer, since it was designed specifically for OpenSimulator. Just like most others, it can do "double duty," i.e., work with both Second Life and OpenSimulator. Hippo only does Windows and Linux.

If you use Mac clients, you need to use a viewer such as Imprudence instead:

http://wiki.kokuaviewer.org/wiki/Imprudence:Downloads

There's Phoenix, too:

http://www.phoenixviewer.com/downloads.php

And Kirstens, **http://www.kirstensviewer.com/S21/**, among others.

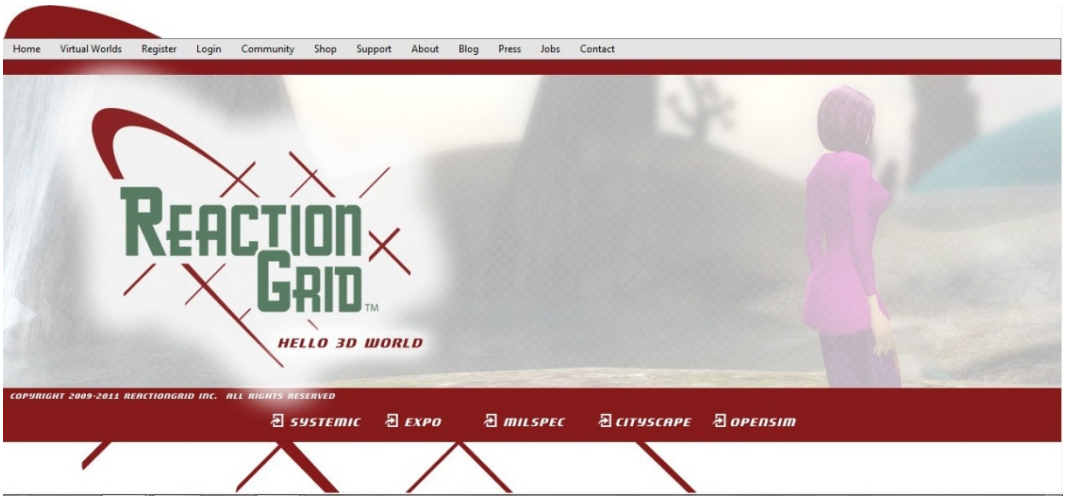
Most of these third-party viewers are 90% the same ("third-party" meaning no longer made by Second Life). They're variations of the old Second Life Viewer Version 1.23, long abandoned but still popular. Second Life made drastic changes to its viewer as of Version 2 and OpenSimulator viewers hopefully will never follow suit! (The reason many people don't like SL Viewer 2+ is that the programs are too demanding hardware-wise.) Some of the third-party viewers, like Kirsten's, are special versions made for designers or other specialists. Unless you're in a Mac shop, you'll be fine with Hippo at first, then shop around if you'd like to try a different one. Here's a tip: For whatever mysterious reasons, Hippo is available only in English. CAIS started with Hippo but switched to Imprudence so our students can use the viewer in Chinese, Korean, German, French, Spanish, etc. (and yes, at CAIS, we need all those languages, and wish the software also came in Slovakian!).

Creating Your Account/Avatar

Using your favorite browser, go to:

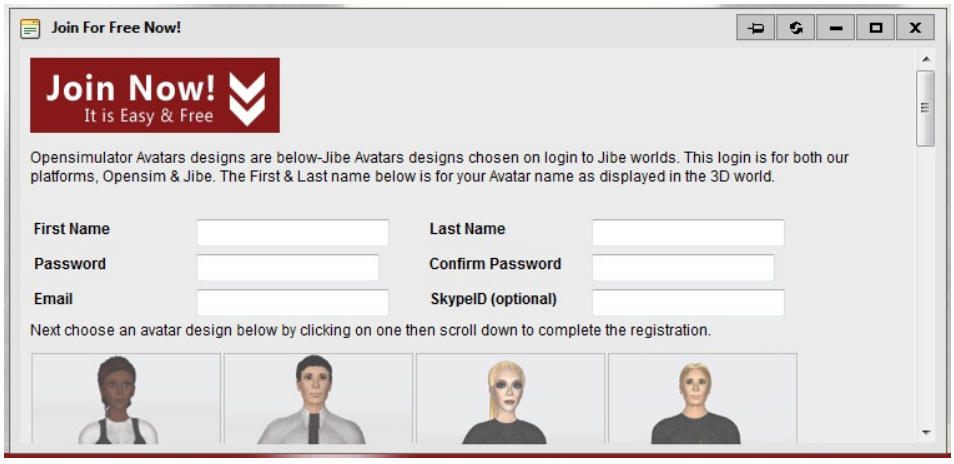
http://www.reactiongrid.com

The ReactionGrid home page displays.



Click the **Register** menu and then click the **Fast and Easy Registration** command.

The **Join For Free Now!** dialog box displays.



The **First Name** and **Last Name** boxes are for your avatar's first and last name, not yours (unless you want your avatar to have the same name as yours, and why not?)! Type the first and last names you want for your avatar.

If your desired avatar name is taken, you'll be advised to choose another one. WRITE DOWN your avatar's name and password so you can use it later for your client/local and client/server installations. Encourage this practice among your students, too. If they use the same username and password for your various grids, you won't have students constantly forgetting. Or at least this will be less likely. In any case, use a notebook or special sheets so students can record their information. They WILL forget!

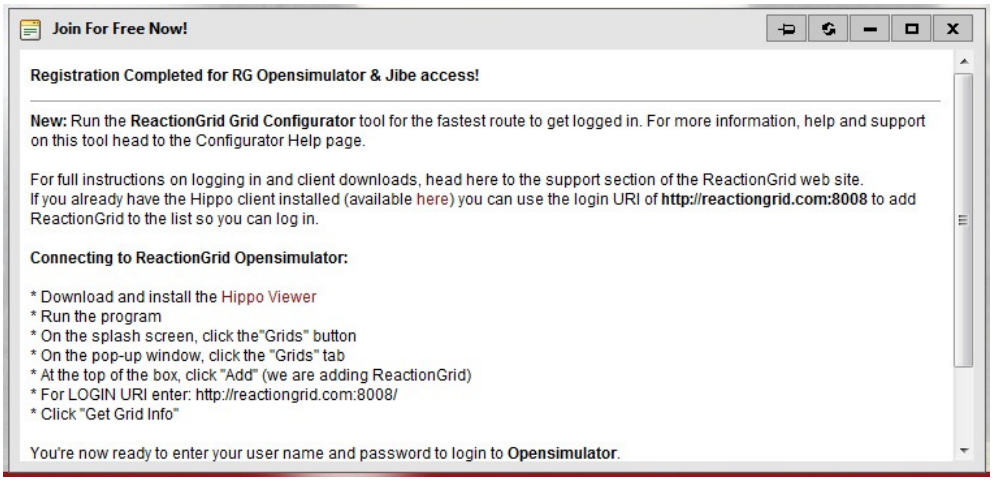
You can always change the way your avatar looks, but you can't change its name, so choose carefully!

Type in the rest of the information: **E-mail** (yours, not your avatar's!), optional **Skype ID** (yours, why would your avatar have a Skype ID?), etc.

Then choose your starting avatar look by clicking one of the pictures. Remember, you can always change the way your avatar looks later!

Ensure that you click the **I have read the Terms** check box. If you want to subscribe to their newsletter, click that check box too. Type the robot-catcher code and click the **Register Now** button.

The **Registration Completed** window displays.



You can see the directions (not instructions!) for adding the ReactionGrid grid to the Hippo viewer, but pay no attention to these now. You don't have a viewer yet! That's coming up next, but first, you need to check viewer system requirements first.

Checking Viewer System Requirements

Before you begin, you should check the viewer system requirements. Go to:

http://secondlife.com/support/system-requirements/

Don't panic! First of all, these Second Life requirements are for their new Version 2+ viewers.

You will be able to get by with less beefy hardware, in some cases far less, especially if you update your video drivers, which is sometimes necessary. If you're "in the ballpark," to use the American expression for "close to it," you should be OK. The biggest factor is video cards. Use these "requirements" as guidelines.

Example of PCs that Changchun American International School uses as client PCs for OpenSimulator:

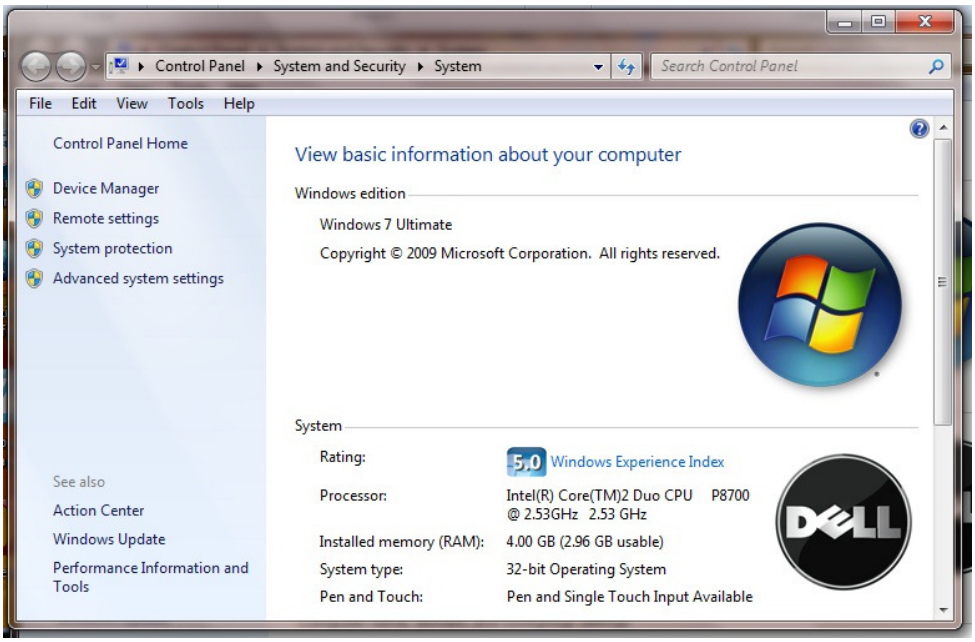
- Dell Dimension C521's running Windows Vista Business, Service Pack 1
- AMD Athlon 64 X2 Dual Core 3600+ CPU, 1.9 GHz
- ATI Radeon X1300/X1550 Series, 128MB of VRAM

These PCs were fine for connecting to OpenSimulator and Second Life after the video drivers were updated. If you start your OpenSimulator viewer and you get an error message stating that your hardware does not meet the minimum requirements, it is most likely because of your video card or drivers. Update your drivers according to the instructions for your particular card, system, etc.

If updating the video drivers doesn't fix the problem, the next step is to increase the video RAM. If that doesn't work, then it's time to investigate replacing the video cards (or PCs entirely).

Another note of caution has to do with 32- versus 64-bit operating systems. On your PC desktop, right-click the **Computer** icon and then click **Properties**.

The **System** window displays.



Look for **System type**. If you have a 64-bit system, you might have problems with your viewer recognizing your current video driver. It's not just 64-bit machines that experience this problem, it can happen whenever Windows selects/assigns one of its drivers for your card. You might have to change drivers to a more specific one, i.e., exactly the one for your particular card, or just a newer one. You can download drivers for most video cards from the Internet these days.

BUT don't worry about anything until something bad happens! Chances are the program will install and run with no problems, so give it a try first and see what happens before you start updating drivers or anything else. If your PCs are less than three years old, and COTS (Commercial Off The Shelf), versus STIG (Slapped Together In Garage), you should be OK.

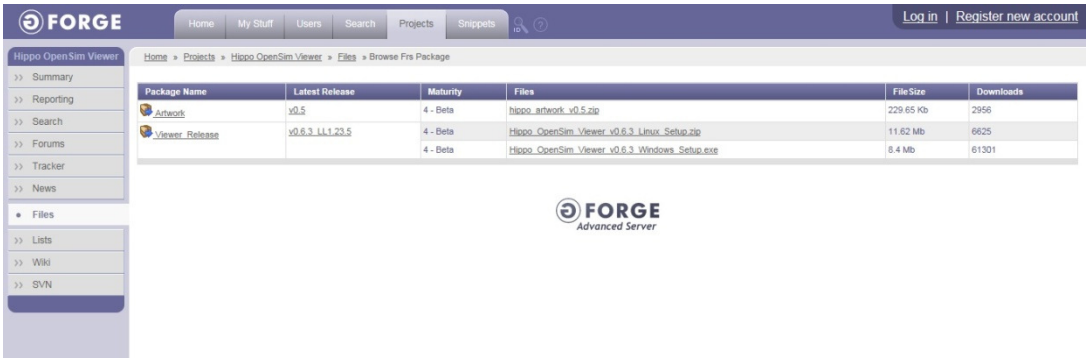
Downloading and Installing the Viewer

Using your favorite browser (YHA used IE 8 to write this e-book), go to:

http://mjm-labs.com/viewer/download.php

Or, if the file has been moved, start at the home page (**http://mjm-labs.com**) and follow the links.

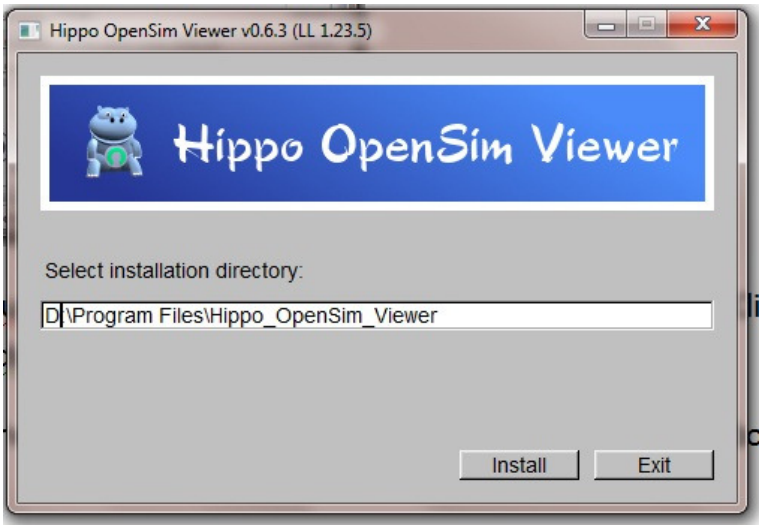
Click either the **Download from OpenSimulator Forge** or **Download from SourceForge** link. The file is the same. Assuming you use the OpenSimulator Forge link, the **Downloads** page displays:



Ensure that you click the **Windows_Setup** version. The **File Download** dialog box displays. Click the **Save** button and save the file to your downloads directory (e.g., **d:\Downloads**).

When the download is complete, click the **Run** button. The **Hippo OpenSimulator Viewer** dialog box displays. Click the **Next** button.

The **Select Installation Directory** dialog box displays.

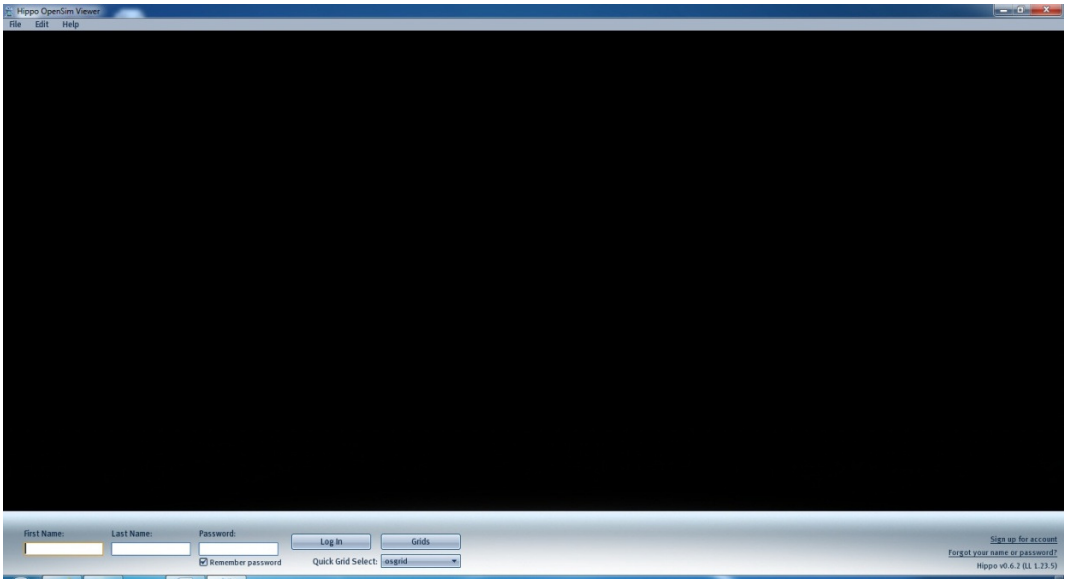


To select the default directory, just click the **Install** button. Or make a change (as in designating drive **D:** in the screenshot) and then click **Install**.

The installation program will begin downloading additional files. Go and get a cup of coffee. This might take a few minutes.

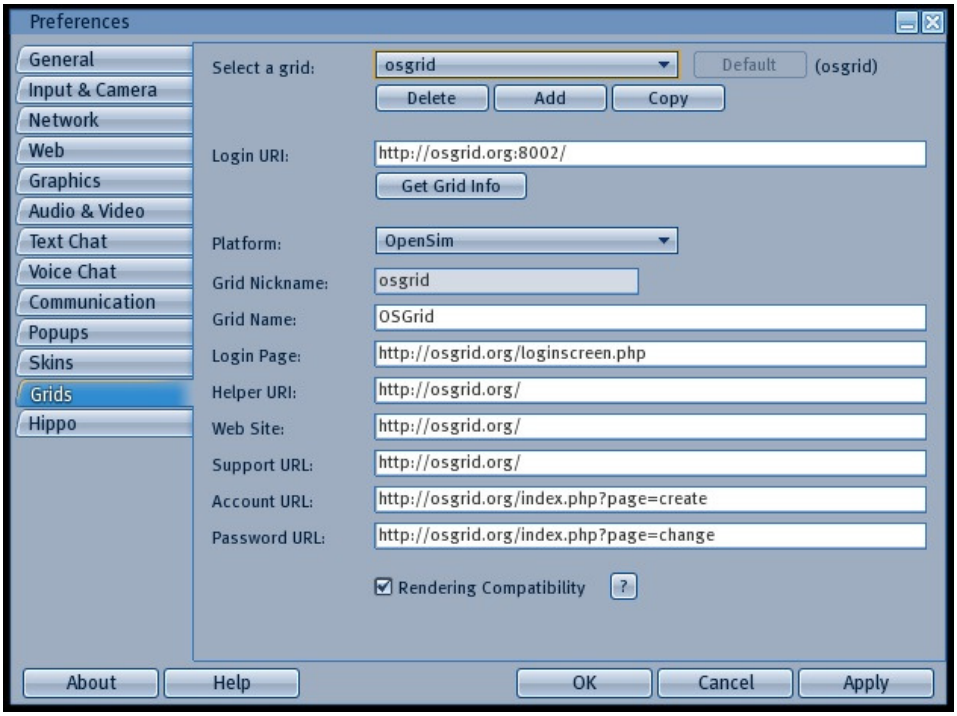
When you see **Installation Finished Successfully**, click the **Start** button. The **Hippo OpenSimulator Viewer** window displays, and you might see…not much of anything.

Depending on several factors, the grid "splash screen" for the default grid might not display and you'll see just black. No worries.



First, click the **Quick Grid Select Down Arrow** button and look for **reactiongrid**. Is it there? If yes, click it. You see **reactiongrid** display next to **Quick Grid Select**. Skip ahead to "Now you're going inworld."

If **reactiongrid** does not appear in the list, click the **Grids** button. The **Preferences** dialog box displays.



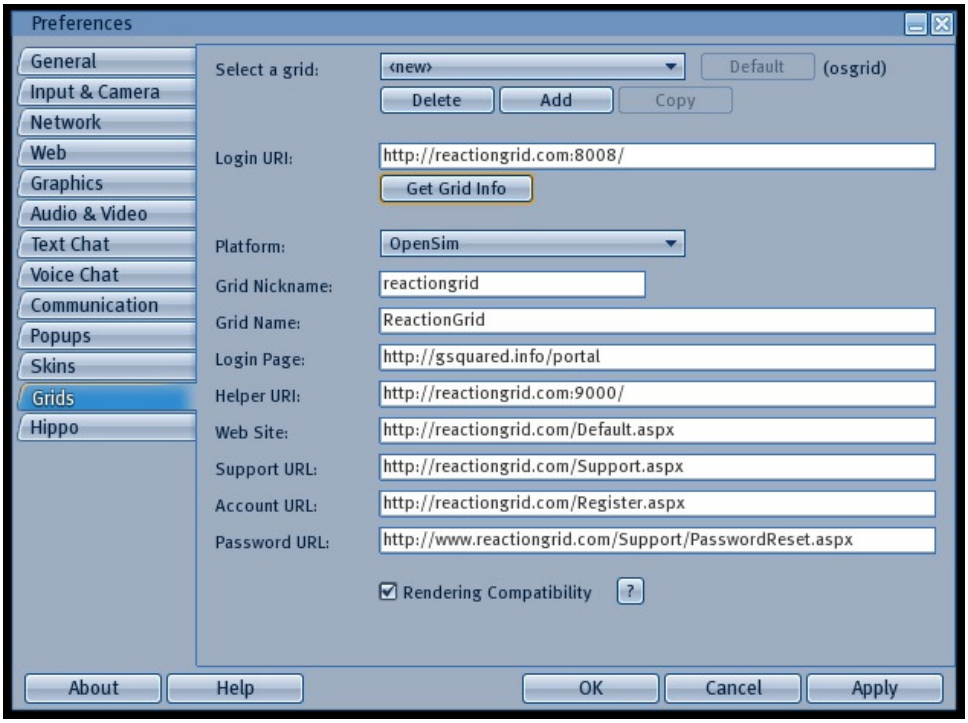
Don't pay any attention to all the tabs on the left just now. Instead, look at the top of the box and click the **Add** button.

In the **Login URI:** text box, type:

http://reactiongrid.com:8008/

Then click the **Get Grid Info** button.

If everything's working correctly (e.g., your Internet connection is fine, and you're not firewalled from using the 8008 port), then this is what you see:



Click the **OK** button. You now see **reactiongrid** next to the **Quick Grid Select** button.

If you get an error message such as **Can't get grid info from server**, then you have an Internet connection problem. Or a firewall problem. Dealing with firewalls is beyond the scope of this e-book, but the ReactionGrid folks have a free tool you can use to detect firewall blocks:

http://metaverseheroes.helpserve.com/index.php?_m=downloads&_a=viewdownload&downloaditemid=12&nav=0

Hopefully, everything's OK. The only times YHA has seen the **Can't get grid info** message is when CAIS was experiencing Internet problems.

Now you're going inworld.

Type the first and last names of your recently created avatar in the respective **First Name** and **Last Name** text boxes. Type your password in the **Password** text box. Click the **Log In** button.

You see a progress bar which shows you that you're making, well, progress, hence the name "progress bar." You might get an error message about this or that region not being available or that you've been moved to another region. This is because you don't have a "home" set yet, nor do you have a last location. This has been corrected in some viewers, but you still might see these messages. Anyway, if you get such a message, just click the **Close** button.

You're inworld!

By the way, if it's too dark to see, it's because it's nighttime! This works the same way in the real and virtual worlds! Click the **World** menu and then the **Environment Settings** command. Then **Midday**. There, that's better! Can't do that in the real world!

OK, take your first steps. Press the **Up Arrow** key to walk forward, the **Down Arrow** key to move backward. Press the **Left Arrow** key to turn left, the **Right Arrow** key to turn right.

Are you ready to fly? Press the **Page Up** key. Fly around a little but don't go too far. Try to land back in the little square around the fountain where you first arrived. Press **Page Down** to return to land, er, cyberland.

Look for the sign that says **Visit Help Island**. You could fly there, but it might be better to walk at this stage. Noobs such as yourself tend to overdo flying and get lost. You'll be doing a lot of walking inworld, so you might as well get some practice. Before you get too far, you'll realize that Help Island has no bridge to it, so you'll either have to fly or walk through water. Even inworld, you can't walk ON water, but you can walk through it! (YHA can, of course, walk on water in both the real and virtual worlds, but you had guessed this already.)

Whichever route you choose, let's assume you've made it to the end of the street with the Star Wars speeder.

This means you've made it to Help Island. Or maybe you've taken another route, are in another location, but you're nevertheless on Help Island. How do you know? Look at the top of your

viewer. See the graphic. If it says **Help Island**, that's where you are. If not, don't panic. You'll make your way there as part of the first exercise. Walk around for as long as you want. Left- and right-click on stuff, you can't break anything! When you're finished, log out by clicking the red **X** in the upper righthand corner of the viewer. You'll start your first exercise by logging in again.

Exercises

You've made it. Not just inworld, but also through the first part of this chapter. Now you have three exercises to complete. You're going to be introduced to the three most important things you and your students will be doing in 3D virtual worlds (and as an added bonus, just about everything you learn about OpenSimulator will also apply to Second Life):

- Interface/Navigation
- Building
- Programming

Interface/Navigation Exercise

OK, we'll cover the interface first, naturally, then navigation. It's assumed you've logged out (quit, in other words) so that Hippo isn't running.

Interface

Before actually going anywhere (virtually, that is), let's take a quick tour of our interface, a fancy word for our viewer. Double-click the **Hippo** (or other) icon on your desktop. The **Hippo OpenSimulator Viewer** window displays. Chances are you see nothing but black.

But, depending on which grid you have selected in the **Quick Grid Select** pulldown, whether or not the grid is up, etc., you might see the "splash screen" for the grid, such as this one for the 3rd Rock Grid.

Now remember, depending on how much time has passed between this writing and your reading, or on which viewer you're using, things might be slightly different. But unless something earth-shattering occurs, most of these third-party viewers are going to be pretty much the same. As mentioned before, they're clones of the Second Life Viewer 1.23, or one-point-something anyway. All the menus and commands aren't present when you're not connected to a grid. We've already worked with the pre-login controls at the bottom. Let's take a look at the top.

Click the **Edit** menu and then **Preferences**. What displays? That's right, the **Preferences** dialog box. For the vast majority of these, the installed defaults are just fine. We're going to look at a few that you might want to change.

Start with the **General** tab. Click the **Show Start Location on Login Screen** check box if it isn't selected already. This will allow you to make a login location choice. This will make more sense after you create a home! Look for the **Language** pulldown at the bottom. If you use Imprudence, you can select another language. You have to quit and then restart the viewer, but then your students can use the viewer in their native language.

Click the **Network** tab. Seems like you'd want **Maximum Bandwidth** to be as high as possible, right? Wrong. Not when you have, e.g., 20 PCs all trying to connect at the same time. At CAIS, we have 20 PCs in our computer labs and this setting is kept on 500 or below. If you've overcome video driver problems and your video still seems choppy, try lowering this value. Set **Disk Cache** to its maximum value of 1000 MB. Keep your disk cache off your **c:** drive, put in on **d:** or another. As students start using Hippo, they might find that logging in starts taking much more time than usual. If so, click the **Clear Cache** button. You'll have to quit the viewer and then start again, but it'll be worth it.

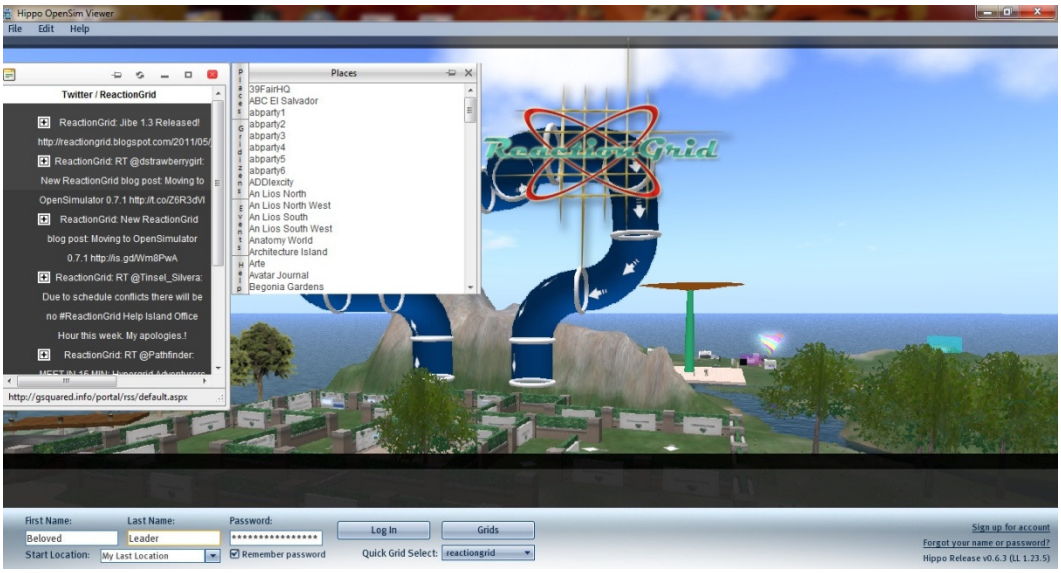
Click the **Graphics** tab. See the **Window Size** pulldown? Another way of compensating for poor video performance is making the window size smaller. **Quality and Performance** is another way of helping, yes, quality and performance. Turn it down and things won't be quite so sharp, but everything will be faster. Don't mess with the rest of the **Graphics** controls until you're a little more experienced, noob!

Same goes for the rest of the tabs and their controls, although you're unlikely to bring the system to its knees by fooling around.

Tip: If you must "tweak," just use the old geek trick of writing down what you change just in case you need to change it back.

Voice chat, streaming music, etc., will have to wait until Your Humble Author's Advanced OpenSimulator course, coming soon to a PC near you.

OK, in the **Quick Grid Select** pulldown, click **reactiongrid**. The **ReactionGrid** splash screen displays.



Log in! You should know how to do it by now.

You're back inworld, on the **Core1** region (top of the screen!). Now the entire Hippo OpenSimulator Viewer interface is visible/available. One of the hardest things for you as a noob to figure out will be what was meant to work only in Second Life and thus doesn't work in OpenSimulator. You see, the viewer was developed for Second Life originally, and in SL you have lots of things you can't do in OpenSimulator. Your students are going to struggle with this

too. Relax, take deep breaths, etc. You'll get used to it. If you try something and it doesn't work, it's probably meant only for Second Life.

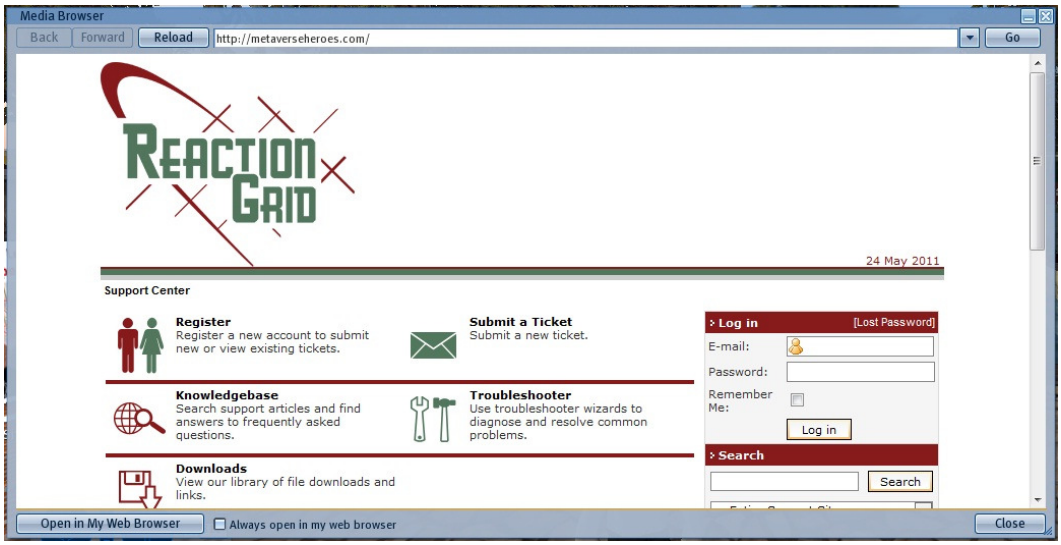
The last part of this exercise will have you experimenting with the various commands to see what they do. You're going to find that on a day-to-day basis, you're not going to need most of them.

Let's continue with the menus at the top, keeping in mind that we're only interested in the basics. Click the **File** menu first. Importing is an advanced topic. You can upload any .jpg and use it as a texture. You'll find out about textures and uploading during the building exercise. Click the **Edit** menu and you see lots of choices, some of which work and some don't!

Click the **View** menu. All that matters to us at this point is that **Toolbar** and **Camera Controls** are clicked/selected. Click the **World** menu and you'll notice that most commands have two ways of being executed. For example, to fly you can press the **Page Up** key or you can click the **Fly** command. You've already been told about the **Environment Settings** command. OpenSimulator has an internal clock, that's set to California time, just like Second Life. Don't ask why! Just remember that you can override the clock so it's daytime all the time.

Tools? We'll get to this menu during our building exercise.

Click the **Help** menu, and then the **Help** command. No one knows why you have to click **Help** twice! The **ReactionGrid Support Center** page displays.



Spend as much time as you want perusing the Knowledgebase. Lots of valuable information. Or experiment with other menu commands. Otherwise, move on to "Navigation. "

Navigation

Now let's take a look at the boxes and buttons on the bottom of the viewer (if you don't see them, it's because you didn't click **Toolbar** on the **View** menu). Sure, maybe this is technically still interface stuff but just relax, take deep breaths, etc. There are other things that have deliberately been left off until later exercises. Right now we are mainly interested in just the **Mini-Map** and the **Map** buttons.

But first, click the **Chat** button. Type **Hi, I'm inworld!** and press **Enter**. Who are you talking to? Yourself? First sign of insanity. Anyway, this is how your avatar will communicate with all others within a certain distance. If you want a private conversation, click the **IM** button. Is voice possible? Yes, but this is an advanced topic. Next e-book!

Click the **Mini-Map** button. The **Mini-Map** window displays. This shows you what's happening in the region you're on. You see the directions, E, W, N and S, which come in handy when moving around. If other avatars are on the sim with you, you see them as green dots.

Now click the **Map** button. The **World Map** window displays. This is going to be your most powerful tool when it comes to Navigation. Look at the map on the left. Each square is a region, sim or island. Yes, three words for the same thing! In the center should be an icon that represents you. The light emanating from your icon indicates which way your avatar is facing. Now that's pretty cool. Ensure your mouse cursor is over the map itself so that you see the crosshair. Scroll your middle mouse button (or do whatever it is you do on a Mac) to zoom out. Or use the **Zoom** slider bar in the lower righthand corner. See all the islands? Each square is either a generic, open-to-everybody ReactionGrid region or a region owned by a school or individual or organization.

OK, before you do anything else, click the **World** menu at the top of the viewer. Then click **Set Home to Here**. Now wherever you roam, you can always go home. Hey, that rhymes!

Look for green dots that indicate other avatars are inworld. If you don't see any, it's OK. Remember that time zones play a big part as to who is inworld at any given time. Click on another region. A red circle appears. Click the **Teleport** button. A progress bar displays, indicating you-know-what, and you're there. If it doesn't work for whatever reason, click another region and try again. If it's nighttime, you know what to do. If it seems you're under water, you probably are. Press the **Page Up** key to fly up into some breathable air. You can survive underwater in OpenSimulator, of course, but unless someone has built something beneath the waves there's little reason to be there.

You have another major way of moving around using the **World Map**. See the text box to the left of the **Search** button? Type some text in there and then click, yes, the **Search** button. Try searching for and teleporting to **Caisland**, Changchun American International School's region. Then try **Help** Island. YHA said you'd be going back and it's a great place to learn. When you have visited a couple of other regions, click the **Map** button again, and then click the **Go Home** button. Yes, that's what it's for.

Just one more thing to try and then you're on your own to explore. We're assuming you're back on **Core1**, which you were supposed to set as your home! Walk to the **Microsoft/IBM** sign and turn right. That cool-looking, glowing thing is an archway, one of the ways you can hypergrid from one region to another.

Walk through it. Wow! Starting to fully comprehend just how great this is going to be for you and your students? If not, shame on you. Imagine "virtual field trips," which will allow your students to visit other learners around the world. Oh, now you get it! It's going to be cool.

OK, now explore as much as you want to. Remember to **Go Home** when you need/want to. Proceed to the next exercise when you're ready.

Building Exercise

We're assuming you're home, on **Core1**. Before we start talking about the mechanics of building, etiquette has to be addressed. In this case, OpenSimulator and Second Life are the same. You're either on your cyberturf or someone else's. In many if not most cases, you will be technically unable to put stuff on other's people's virtual property. In other words, the owners or renters/leasers will have set land permissions that prohibit you from doing this. You should be able to figure out why without too much thought. When you've got your own chunk of ground, you won't want others littering it either. It's important to remember that unless you're on your own private grid, you're a guest on somebody else's.

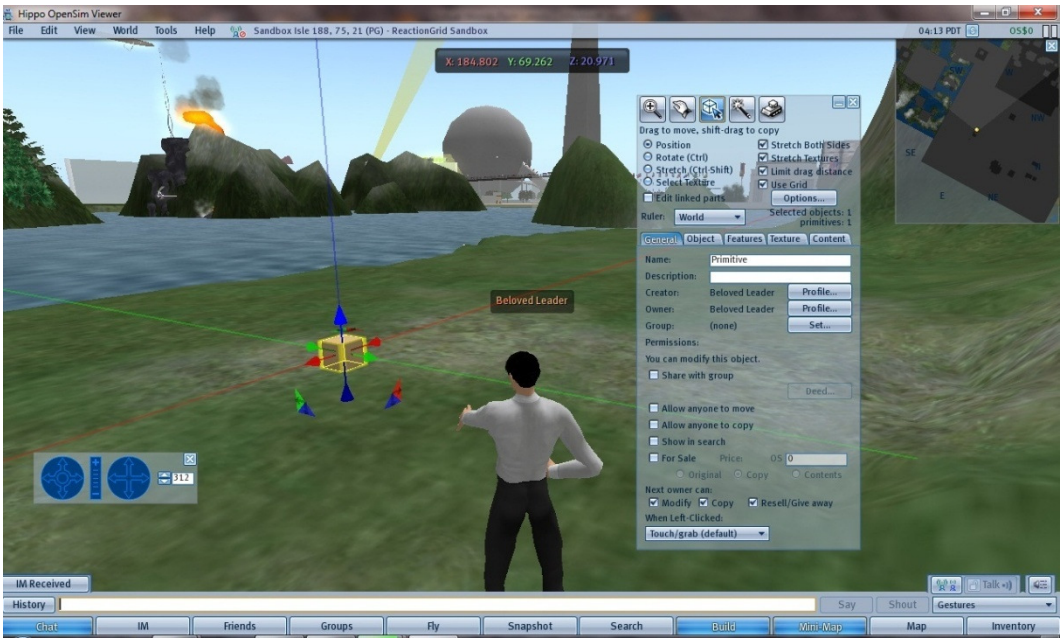
But enough of that. Time to visit a sandbox. A sandbox is a region (or part of one) that someone has generally designated as being a place where everybody (or just about everybody) can build stuff. The sandbox is cleared periodically, which means that even if you don't clean up after yourself, someone else will. Don't worry. You won't lose your valuable work. You'll be saving your things in your Inventory.

Click the **Map** button and in the **Search** text box, type **sandbox**. What then? Yes, click the **Search** button! The folks at ReactionGrid have provided an entire region as a sandbox. Click the **Teleport** button to get there. Fly around a bit and find an open patch of cyberturf. You don't need a lot of room.

Pause for a moment and think of all the great things you've seen so far: buildings, cars, you name it. ALL of those things were created using the same tools you are about to learn how to

use. It's just a matter of practice. As you and your students start off, your creations are going to seem simple and perhaps even crude, but keep trying. With the right amount of knowledge and skill, you or one of your kids could be a professional relatively soon! Here you go. Your first primitive.

Click the **Tools** menu. Then **Select Tool**, then **Create**. The **Click inworld to build** dialog box displays. The default shape is a cube, that's fine. You can see that your mouse cursor has become a magic wand! Click the ground in front of you. A cube appears, or rezzes, as we 3Ders say. And you'll notice that your dialog box has changed.



It's your first primitive! And you know this because you can see that its name is **Primitive**! Your YHA tends to use the terms "primitive" and "object" interchangeably. Yes, two more words for the same thing! Actually, in Second Life things are objects and in OpenSimulator things are primitives. The default option button selection at the top is **Position**, and you can see three arrows on your primitive. Why do you think they call it 3D? Because everything has THREE

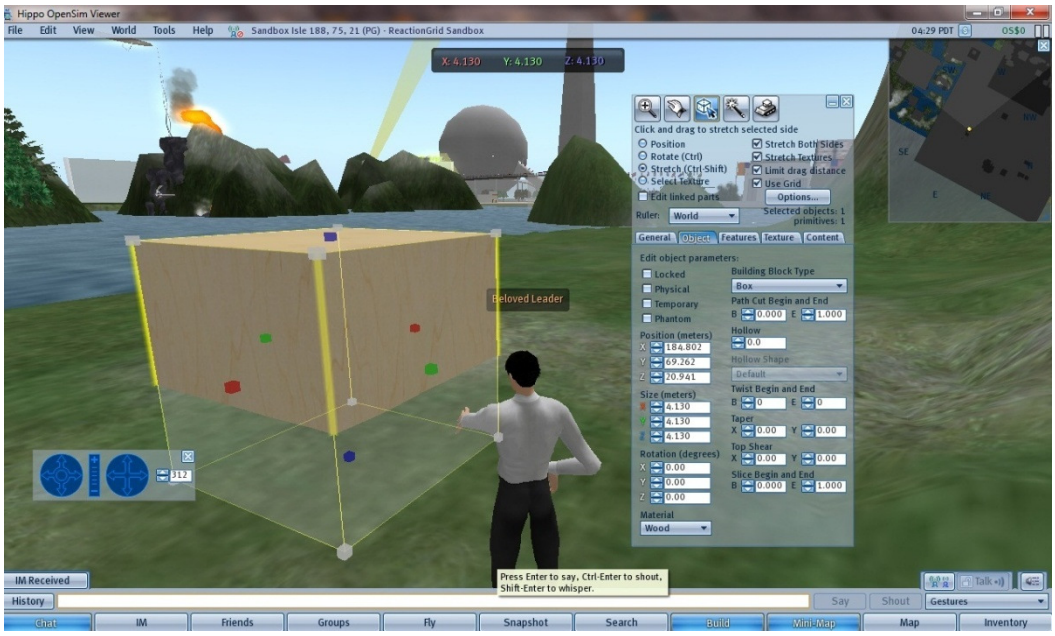
DIMENSIONS. Everything, and that includes your avatar, has a three-dimensional location. Primitives have a three-dimensional size, and so on.

Click and drag the **Blue Arrow** to move your primitive up and down. Click and drag the **Red** and **Green Arrows** to move it back and forth. Now click the **Object** tab and look at the **Position** coordinates. Get it? It's easy. You can also change a primitive's location by typing numbers in the respective boxes. **X** is red, **Y** is green and Z is blue. Give it a try. Move your primitive or object (thing!) around by typing in numbers.

Click the **Rotate** option button, beneath **Position**. You see three colors of bands. This is how you can change the rotation in degrees of your primitive (or object!). Look at the bottom under the **Object** tab and you can see how you can type in the numbers for **X**, **Y** and **Z**. Each one is called an axis, which will be important later in, yes, you guessed it, the Advanced OpenSimulator e-book. Try rotating your object both ways, by clicking and dragging on the colored bands, and then by typing in numbers.

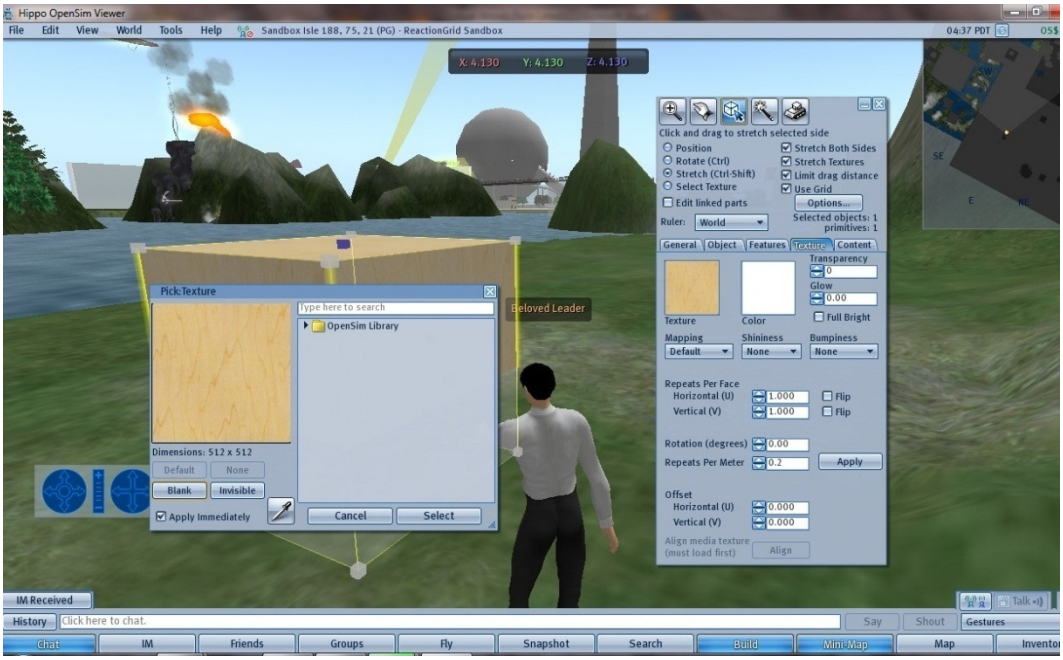
Click the **Stretch** option button, beneath **Rotate**. You see cubes. **Red Cube** for **X**, **Green Cube** for **Y**, **Blue Cube** for **Z**. Click on any one of them and drag to change the appropriate dimension's size. Now look at the **Gray Cubes**. Click and drag one of these and your primitive changes proportionately, i.e., all dimensions change the same amount in size. Look under the **Object** tab and yes, you have boxes you can type the values into for **X**, **Y** and **Z**. But there's no equivalent of the Gray Cube.

Again, try working with your object both ways, via the cubes and the boxes.



Click the **Texture** tab. "Textures" are graphics, or images as 3D virtual worlds call them once they're stored in your Inventory. Told you about 3D vocabulary! You can upload any graphic, .jpg or .png, to OpenSimulator and put it on a primitive. First, we'll just use the built-in examples of textures. All primitives in OpenSimulator (and Second Life) start off as plain old wood. No fun. Click the **Texture** box.

The **Pick Texture** dialog box displays.



Click the **OpenSim Library Down Arrow**, then the **Texture Library Down Arrow**. A list of available textures displays. Click **bricks** and then the **Select** button. Your primitive is now made out of bricks! Let go, or deselect, your primitive by clicking the **X** in the upper righthand corner of the dialog box. If you want to use a .jpg or .png on your PC as a texture, click the **File** menu and then **Upload**. Select the file and keep clicking **OK**. If the file you uploaded is named **XXX** then click the **Inventory** button and search for **XXX**. There it is. Click the file and drag it to your primitive. Wow! Cool, huh?

Press the **Left** or **Right Arrow** key and spin around to take another look at everything around you, to remind yourself that all of it was created using the same dialog boxes you've just been using. There's no way an e-book such as this can even begin to cover all the possibilities of OpenSimulator (or SL!) building. Entire books have been written on the subject, and a few will be suggested to you in later discussion. You will find, however, that many of your students will

not require or even desire a textbook, that they'll learn the most about building just by trying the different controls. This is your assignment for the remainder of the time you want to spend on building right now. But first, a quick introduction to making copies of things you create and putting them into your Inventory.

Right-click your primitive (it's the thing that looks like bricks, remember?) and then click **Edit** on the pie menu. The **No Apparent Name Dialog Box** displays, guess we'll call it the **Edit** dialog box. Click the **General** tab if it's not already selected. In the **Name** text box, type something you can remember and find later: **Brick Box** or something. Right-click the primitive again, or rather, **Brick Box** again. Then click **Take** on the pie menu. It's gone! Now click the **Inventory** button in the bottom righthand corner of your viewer. You have so little stuff in your collection now, it appears so that you can see it. Click it and drag it to the ground. Now take it back again. That's cleaning up after yourself! If you had clicked **Take Copy** on the pie menu, you would have left the primitive where it is and taken a copy. If you had clicked **Delete**, your primitive would be in your **Inventory Trash** folder.

Take your time experimenting with the various options for building. With the primitive selected for editing, click the **Object** tab and notice what happens when you manipulate **Path Cut** or **Hollow**. Try **Twist Begin and End**, **Taper**, **Top Shear**, **Slice Begin and End**, etc. The best (not to mention the most fun!) way to learn all these functions is just to try them and see what happens. Be sure to take your stuff with you when you're finished.

Programming Exercise

How does everything (and everyone, if you're now thinking of avatars as humans, which you might soon!) move around in Second Life and OpenSimulator? It all happens courtesy of a physics engine. Details about this will definitely have to wait for the Advanced OpenSimulator e-book, but just think of it as the programming that tries to simulate how the real world works. You step off a building, you fall to the ground. In Real Life and Second Life, because of gravity, genuine and virtual. But the physics engine can't anticipate and accommodate everything we humans might want to happen in 3D virtual worlds. We want control over our environment, real or virtual! That's where Linden Scripting Language (LSL) comes in, a built-in programming lingo that's used by both Second Life and OpenSimulator. "Built-in" just means that the viewer interface plus the database backend serves as the Integrated Development Environment, in other words, you don't need anything else. LSL allows you to make your primitives interactive, to create rich, rewarding inworld experiences involving motion, simulations, etc.

Some might argue that LSL programming shouldn't be addressed in an introduction like this at all, but Your Humble Author contends that since it's so easy to do simple stuff, what the heck. LSL is a powerful language and those who work hard enough to master it can do some amazing things. But if you just want to achieve simple functionality, it's extremely easy to use, particularly because you can beg, borrow and steal existing scripts from a variety of sources. Using your favorite browser, Google **second life scripts** and you'll get an idea of the thousands that are available for free.

A note of caution: OpenSimulator is always a little behind Second Life in development. Plus the physics engines don't match. There are advantages and disadvantages to this. Not all SL vehicle scripts work right in OpenSimulator, but OpenSimulator has the ability to handle nifty joints

and hinges. All this means to you at this point is that you might find a script that works in Second Life but doesn't work in OpenSimulator. This is a small price to pay for the advantage of starting with an existing script. And a lot of them do work! Students learn a lot by "tweaking" code that's already written.

You work in a K-12 school and so you're teaching youngsters who've probably never programmed before. Your Humble Author has learned from experience that jumping right into banging out LSL code isn't an effective technique, hence the introduction via existing scripts. Another way to gently ease your little darlings into programming is to start from scratch, or rather start with Scratch:

http://info.scratch.mit.edu/Scratch_1.4_Download

Scratch is a programming language, and IDE, all free, designed by the Massachusetts Institute of Technology (MIT) specifically to help kids with no programming experience learn how to code. It's available for Windows and Mac! When inexperienced students have to type code, as seasoned IT teachers know, a semicolon or whatever in the wrong place making an entire program inoperable can be extremely frustrating, not to mention counterproductive regarding the learning experience. With Scratch, learners can focus on program design and snap together blocks of code, much like (well, exactly like) putting together puzzle pieces. After your students are familiar with the basics of variables, loops, and so on, they can more easily move on to other languages. The best idea is to use Scratch for Second Life (S4SL) as an intermediate step. S4SL is also free from MIT:

http://web.mit.edu/~eric_r/Public/S4SL/

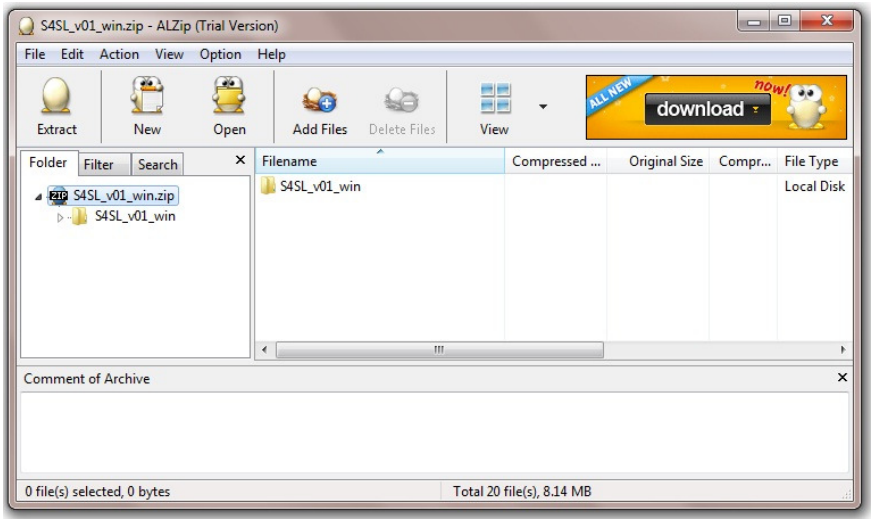
It's also available for Windows and Mac. Click the **Windows** or **Mac** link. The **File Download** dialog box displays. Click the **Save** button and select a directory to download to (e.g., **d:\Downloads**).

When the .zip file has finished downloading, click the **Open** button. Your default compression utility will start. If you don't have a compression utility installed, shame on you. Close the **File Download** dialog box and get a program such as WinZip or ALZip:

WinZip: **http://www.winzip.com/win/en/index.htm**

ALZip: **http://www.altools.com/ALTools/ALZip.aspx**

Both companies offer free evaluation releases. Download/install your choice, then return to the folder where you downloaded Scratch for Second Life and double-click the file. Follow the WinZip or ALZip program's directions to decompress or unzip the file.



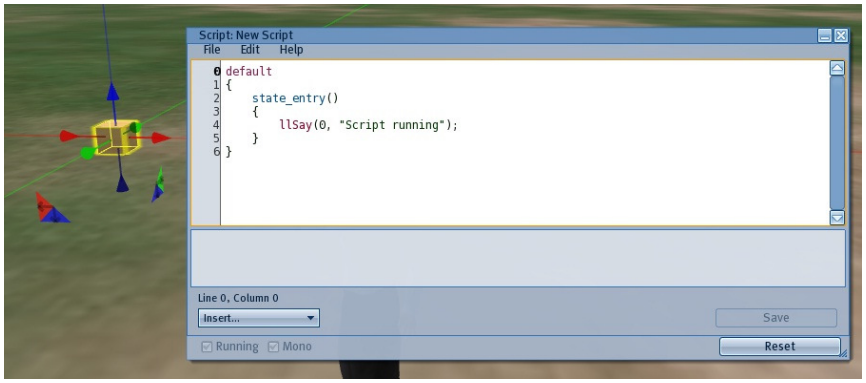
Create and then designate an extraction (unzipping) location (e.g., **d:\S4SL**) and deselect the option of creating a filename folder, if offered. When the .zip file unzips, it'll create a folder of its own.

So you'll have, e.g.:

d:\S4SL\S4SL_v01_win

OK, we're ready to go with Scratch for Second Life, but let's start with the LSL basics. If you're not already logged into ReactionGrid, do so now. Find your way back to the sandbox you were in before (remember how to do that?). Find the same spot you had before, assuming it's available, or locate another place, and create another cube (and how about that?). Or open your Inventory and drag your **Brick Box** (or whatever you called it) to the ground. Right-click the object and then click the **Content** tab. Then click the **New Script** button under **Contents**. **New Script** and an icon display. Double-click **New Script**.

The **Script: New Script** dialog box displays.



LSL is a "state-based" language, meaning that every object has a default state. When you first create an object, its default state is to essentially do nothing. Well, OK, to actually do nothing. If you want it to be doing something else, all the time, you have to change its default state. If you want it to be interactive, you have to define a trigger of sorts and then what the object is supposed to do. For example, just a bit later you're going to make a door that opens when an avatar touches it. Right now, we just want to make our object or primitive spin.

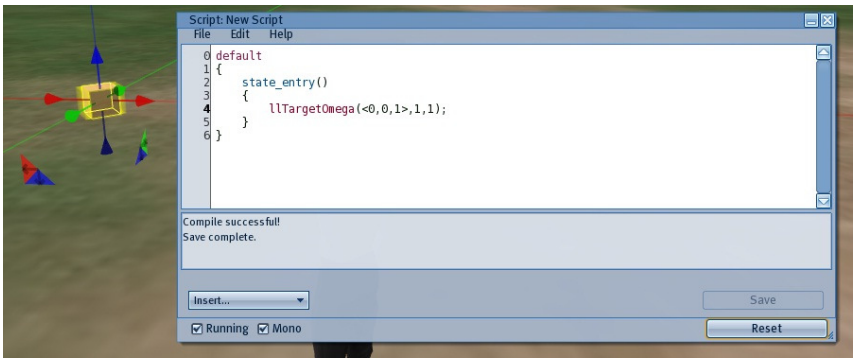
Change the text:

```
llSay(0,"Script running");
```

To this:

```
llTargetOmega(<0,0,1>,1,1);
```

Click the **Save** button and you see: **Compile successful. Save complete**.



Now "let go" of the object. That is, close the **Script: New Script** and the **Edit** dialog boxes. It's spinning! Not very fast, but it's spinning.

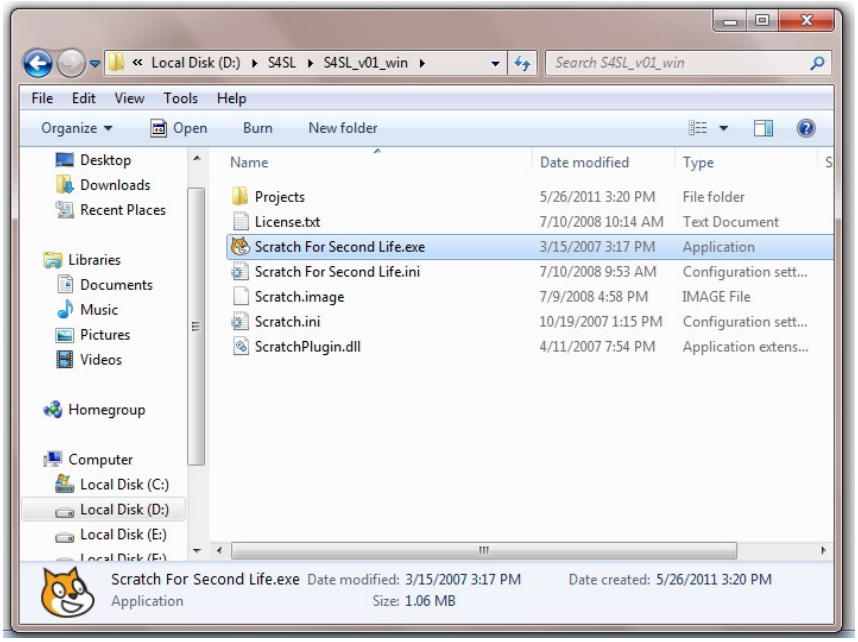
What do you think **<0,0,1>** means? Yes, good, the X, Y and Z axes. The 0's or 1's are vector components, so, e.g., 1 would mean a positive rotation, -1 a negative rotation. Try changing the values, save again and see what happens. Do you want to make it move faster? Change the middle number, the spin rate, from 1 to another value, like this:

```
llTargetOmega(<0,0,1>,5,1);
```

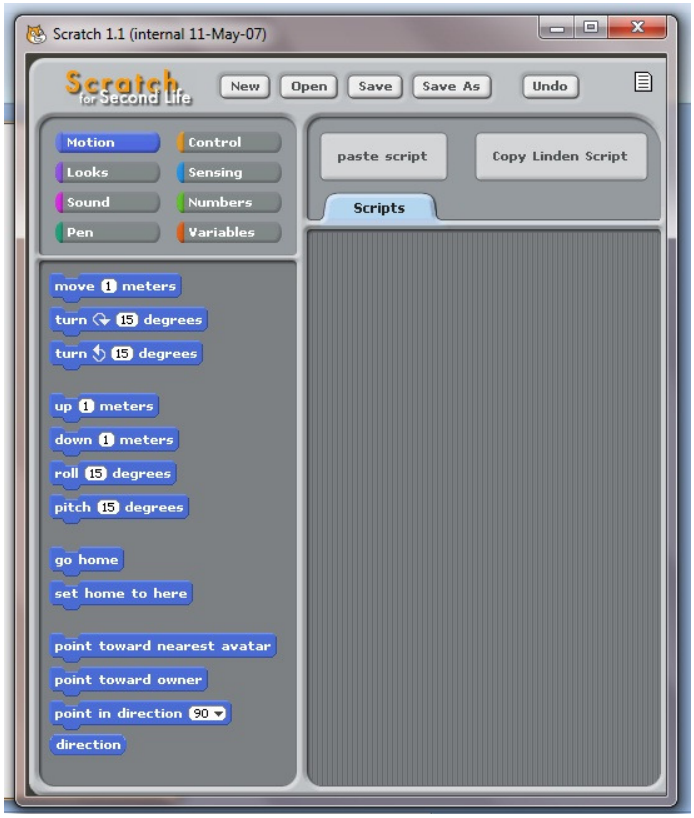
How high up can you go?

OK, you get the idea. From these simple beginnings, people have progressed to becoming professional programmers. You and your students can too. Now let's try Scratch for Second Life. Open your folder (wasn't it **d:\S4SL\S4SL_v01_win?).**

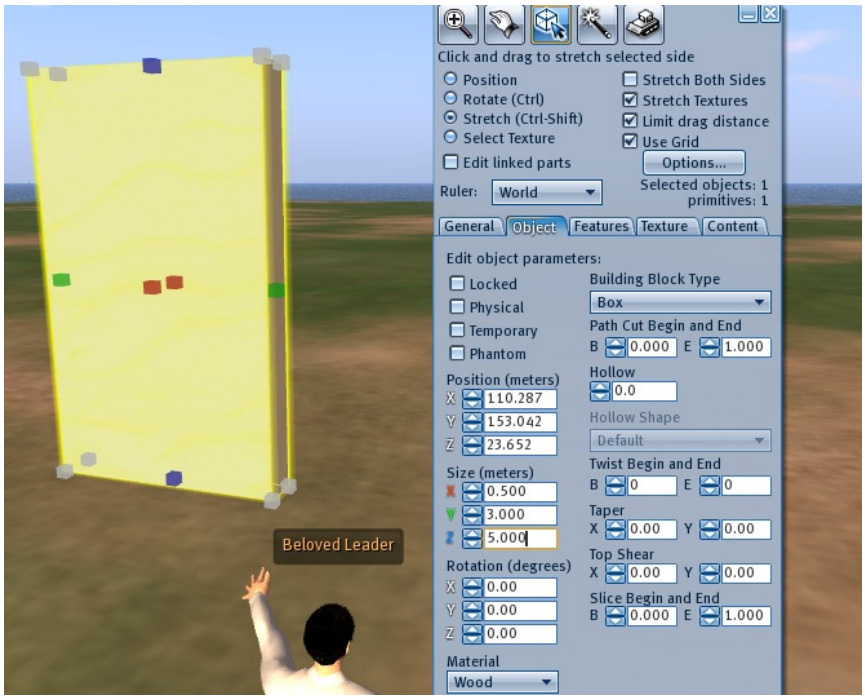
Create a desktop icon by right-clicking **Scratch for Second Life.exe** and then clicking **Send to Desktop**. Then double-click the cat icon.



The **Scratch 1.1** (for Second Life IDE) window displays.

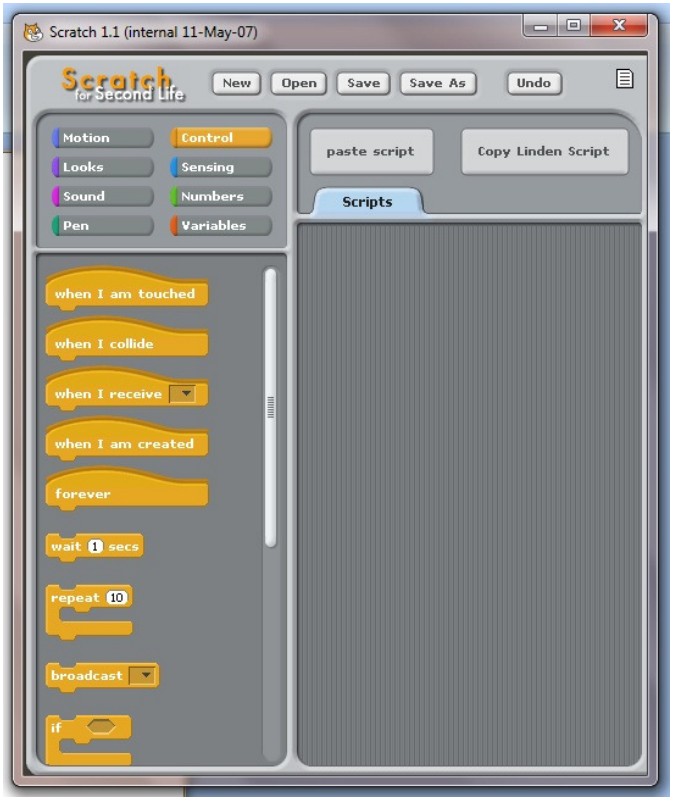


OK, now let's say you've asked your students to build a house. You want them to create a door that will open when an avatar touches it. Create a cube, and then stretch it so it's 0.5 meters thick, 3.0 meters wide and 5.0 meters high.



Then change the texture to something else, maybe a different color of wood, such as mahogany. Remember how to change a primitive's texture? If not, shame on you. Go back a few pages and refresh your memory.

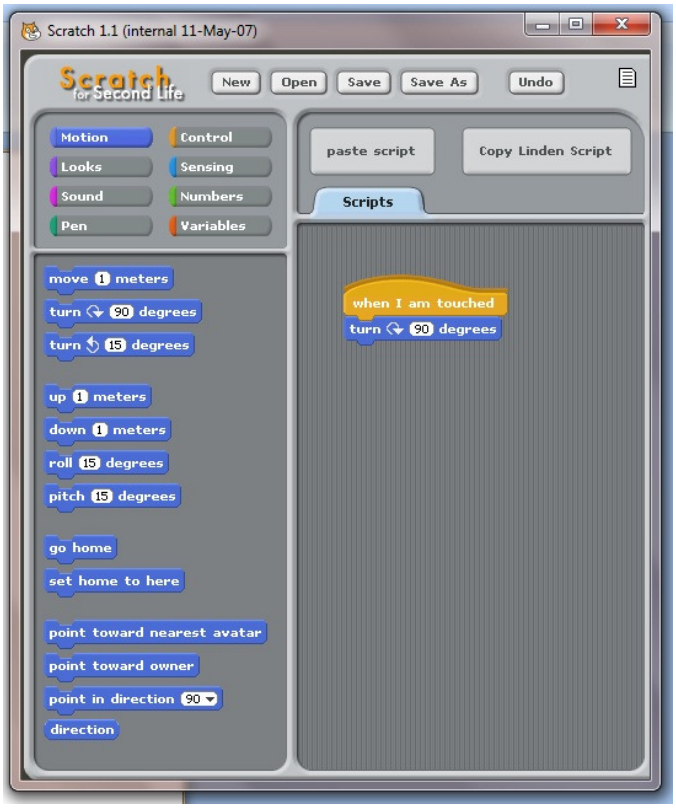
Switch back to the Scratch window. Click the **Control** button. The **Control** commands display.



Click the **When I am touched** control, or puzzle piece, and drag it to the right under the **Script** tab.

Then click the **Motion** button. The **Motion commands** display. The second command from the top is the **turn clockwise X degrees** command. Click the **15** to highlight the number, then type **90**. Click and drag the command to the right under the **Script** tab.

Ensure that the two puzzle pieces snap together.



Did you ever think programming could be so easy? You have your program in Scratch, now it's time to convert to LSL. Click the **Copy Linden Script** button. Go back to OpenSimulator and you will find that your avatar is waiting patiently for you. Right-click the door, click **Edit**, click the **Contents** tab, click the **New Script** button, double-click the **New Script** icon. The **Script: New Script** dialog box displays.

Select the entire script in the **Script: New Script** dialog box and press the **Delete** key. Click in the now blank window, ensuring that your insertion point (that blinking line) is in the window. Then press the **Ctrl** and **V** keys (paste, in other words). The code from S4SL displays, all hundreds of lines of it. S4SL creates essentially the same program each time, just with unused

options commented out. This is great, because it will help your students learn. But never mind this now. Click the **Save** button. You should see: **Compile successful. Save complete**.

Let go of the object by closing both dialog boxes. Right-click the door and click **Touch** on the pie menu. It turns clockwise 90 degrees. Cool, huh?

Continue experimenting with programming as long as you like. Just remember if it's nighttime to go to bed at a reasonable hour. Sometimes when people start doing this, it's hard to stop. Google **second life scripts** and try some more complex code out to see if it works. Or go get some code at **http://lslwiki.net**.

Congratulations. You have finished (and survived) the Introduction chapter. Now, if you've decided to not go the host route, it's time you set up your school OpenSimulator installation. That's coming up next.

Step-by-Step School Setup

What Are You Supposed to Do?

For this chapter, it's assumed that you're following the directions as you read them. If you do everything in this chapter, i.e., install a client/local or client/server setup of OpenSimulator, plus install and configure Hippo on your computer lab PCs, your school will be ready to go. This is it. The moment you've been waiting for. Can you meet the OpenSimulator challenge? Do you have what it takes? Yes. Relax. Take several deep breaths. Everything's going to be OK.

Hold it right there! Before you get started. This e-book covers only "regular OpenSimulator." You do have installation alternatives, beyond our scope but definitely worth mentioning:

- **Diva Distro:** Hypergrid-preconfigured package, get it via OpenSimulator website
- **Kitely:** Virtual worlds on demand via Facebook, **http://www.kitely.com/#!home**
- **Sim-on-a-Stick:** Run OpenSimulator on a USB drive, **http://www.simonastick.com/**

Maybe you'll find one of these alternatives to be easier and thus better for you.

Client/Local, Then Client/Server

First you're going to set up a client/local installation, then you're (optionally, of course) going to set up a client/server installation. If you just want a client/local setup for your school, then stop following these directions at the appropriate point and just copy the client/local setup on all your PCs. The reason it's a good idea to follow these steps in this order for the client/server

setup as well is that you can get the work done on your client/local PC before moving your OpenSimulator installation to the server. This is especially important if you're not allowed to touch a server. Your Humble Author doesn't let non-IT teachers get anywhere near CAIS servers! Plus you get to use both installations after you're done!

The Client/Local setup consists of these steps:

1. Downloading and Unzipping the Binaries
2. Starting the Executable and Creating the Region
3. Logging Into Your Own Private Grid

The Client/Server setup consists of these steps:

1. Consulting Your Administrator and/or Checking Your Server
2. Copying Your Directory and Tweaking the .INI
3. Starting the Executable and Adding Your Grid

Creating Your Client/Local Installation

Downloading and Unzipping the Binaries

OK, remember that you're starting with a client PC, your laptop or desktop, not a server. The older versions of OpenSimulator came with installation programs, but not anymore. Installers really aren't needed, because all you need to do is decompress ("unzip") the downloaded file and you're (almost) ready to go.

The latest version of OpenSimulator is available via the OpenSimulator website, which is a wiki. Ensure that you download OpenSimulator only from the OpenSimulator website/wiki. If you get OpenSimulator from another site, you can't be guaranteed to get the latest or even an official version.

Sometimes the latest release of OpenSimulator is downloadable from the home page:

http://opensimulator.org/wiki/Main_Page

If the download is available on the home page, you see something like this:

Ensure you click the button for the **Windows Binaries** Zip version (at the top in the graphic). Windows is the flag, Linux is the penguin! Don't download the source option, this is only if you

want to compile the code, and you don't want to do that! At the bottom there's the **Diva Distro** option, as mentioned previously.

If the download is not available on the home page, don't despair. Sometimes it's on a special Downloads page, which you get to by clicking the **Download** icon on the home page (it looks like a text file with a green arrow pointing, appropriately enough, down).

Be sure it says **Download**, though, because another identical icon takes you to **Grid List**.

http://opensimulator.org/wiki/Download

Under **Binary Packages**, click the file link, which will be something like:

http://dist.opensimulator.org/opensim-0.7.1.1-bin.zip

"Something like" meaning that the version might be different, i.e., not 0.7.1.1. Remember to look for the .zip extension, that's the Windows version!

Whether you click the button or link, the **File Download** dialog box displays. Click the **Save** button and select a directory to download to (e.g., **d:\Downloads**).

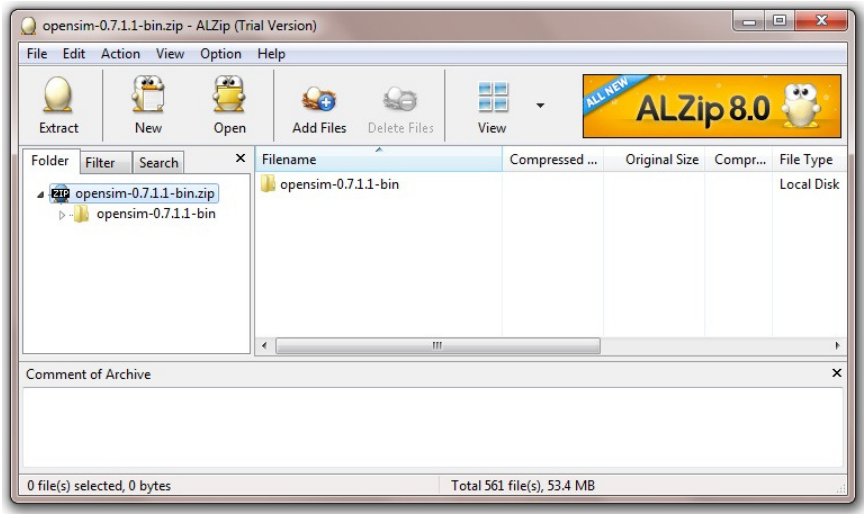
When the .zip file has finished downloading, click the **Open** button. Your default compression utility will start. If you don't have a compression utility installed, shame on you. Close the **File Download** dialog box and get a program such as WinZip or ALZip:

WinZip: **http://www.winzip.com/win/en/index.htm**

ALZip: **http://www.altools.com/ALTools/ALZip.aspx**

Both companies offer free evaluation releases. Download/install your choice, then return to the folder where you downloaded OpenSimulator and double-click the file.

Follow the WinZip or ALZip program's directions to decompress or unzip the file.



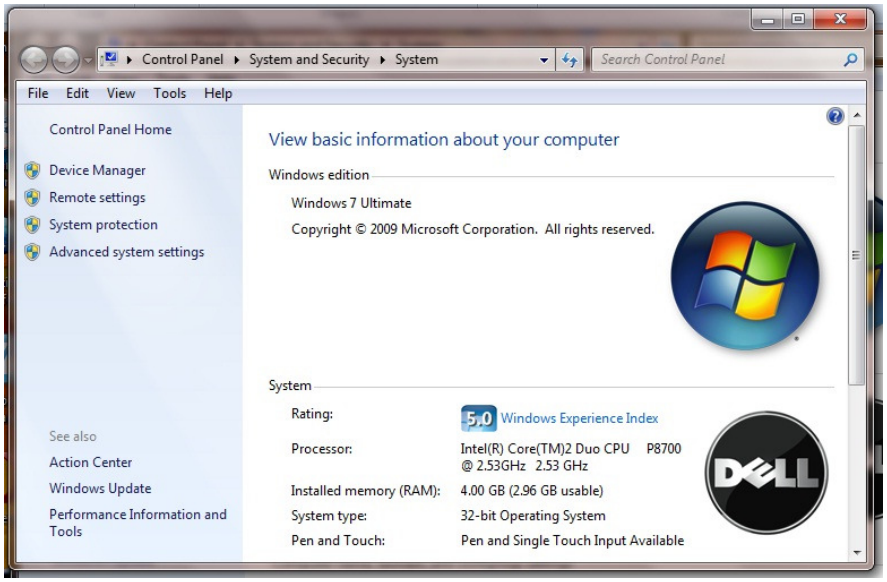
Create and designate an extraction (unzipping) location (e.g., **d:\OpenSimulator**) and deselect the option of creating a filename folder. When the .zip file unzips, it'll create a folder of its own. So you'll have, e.g.:

d:\OpenSimulator\opensim-0.7.1.1-bin

Starting the Executable and Creating the Region

OK, now pause for a moment. Does your computer have a 32-bit or 64-bit operating system? If you don't know, slap yourself, but not too hard. Then right-click the desktop **Computer** icon and then click **Properties**.

The **System** window displays:

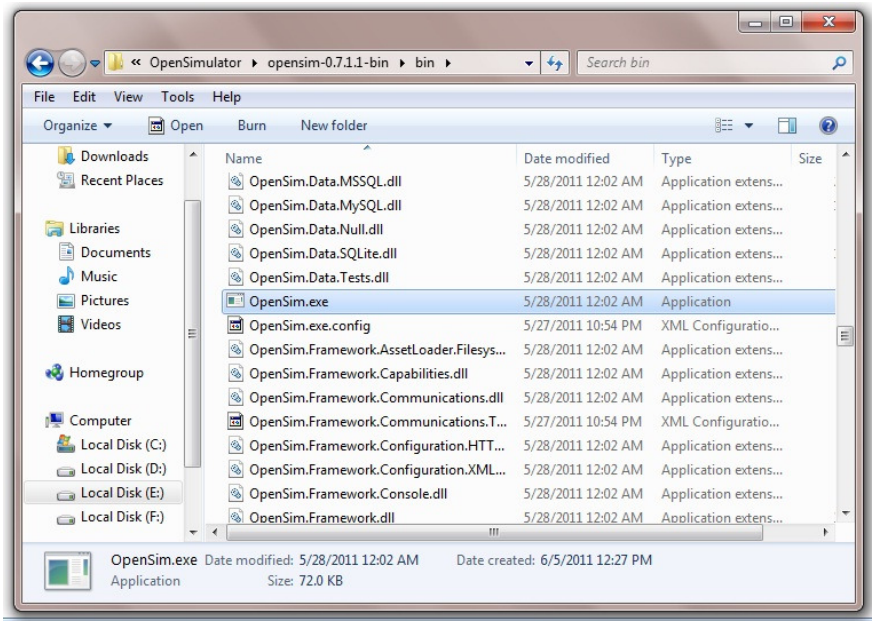


Look for the **System type** entry…now you know. The assumption is that you have a 32-bit system.

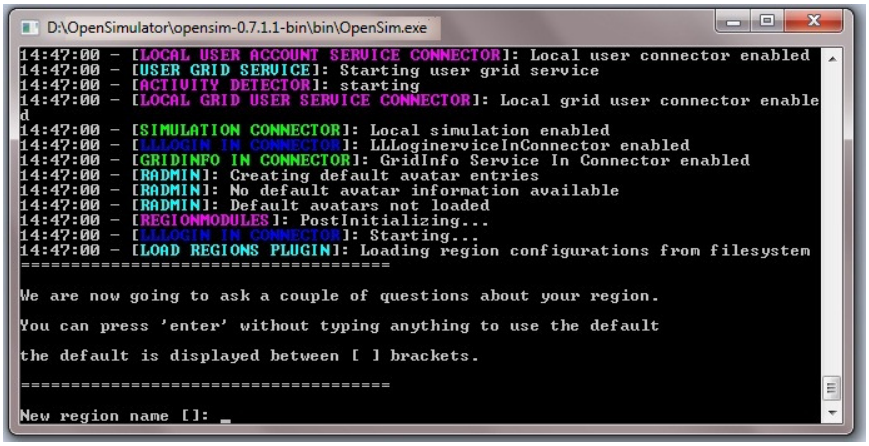
Go back to the directory where you unzipped OpenSimulator and then go one directory deeper, to **bin**. So you're in, e.g., **d:\OpenSimulator\opensim-0.7.1.1-bin\bin**.

If you have a 32-bit system, look for **OpenSim.exe**– of **Type Application** – and double-click the file. Ensure the **Type** says **Application**. Be careful: Lots of files are called "OpenSim-Something." Do you have your **Folder Options** set to show, or rather not hide, file extensions?

Again, look for **Application** under **Type**.



If you have a 64-bit system, look for **OpenSim.32BitLaunch.exe** – again, of **Type Application** – and double-click the file. A **DOS** aka a **Command Prompt** window displays. This is what all computers looked like before graphical user interfaces like Windows. Scary, that is. Don't be afraid. 99% of what you will see you either won't care about, can't do anything about or both! The scrolling text first stops at the **New region name []:** prompt.



Type the name you want for your region (e.g., **Schoolland**) and press the **Enter** key.

Up next is the **Region UUID** prompt (UUID stands for Universally Unique IDentifier, just in case you were wondering). Press **Enter**. The defaults are fine for this and the next few entries.

Next is the **Region location** prompt. Press **Enter**.

Next is the **Internal IP address** prompt. Press **Enter**.

Next is the **Internal port** prompt. Again, defaults are best at first, so accept 9000 as the port. Press **Enter**.

(If you run into a problem with conflicting ports, then you're going to need to call in some heavy-duty technical help, like from a system/network administrator, in order to determine an available port. Keep your fingers crossed that this won't happen. Most of the time it won't.)

Assuming port 9000 is fine, next is the **Allow alternate ports** prompt. Press **Enter**.

Next is the **External host name** prompt. Press **Enter**.

Next is the **Estate name** prompt. The distinction between region and estate is an advanced topic and of no concern at this point. In fact, for educators running their own virtual worlds it seldom comes up at any point! Just type the same name you did for region (e.g., **Schoolland**) and press **Enter**.

Next is the **Estate owner first name** prompt. This will designate the name of your "god" avatar, the one with full administrative permissions. Type a first name (e.g., **Beloved**) and press **Enter**.

For **Estate owner last name**, type the last name you want for your god avatar (e.g., **Leader**) and press **Enter.**

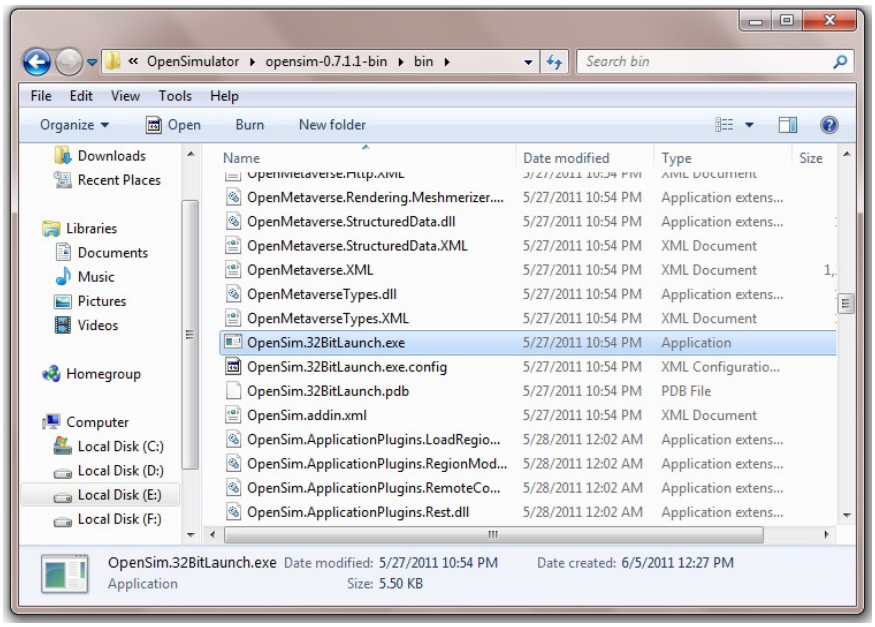
Then enter the password for your god avatar (e.g., **iamnumberone**).

Is this the same avatar information you entered for your ReactionGrid avatar? Should be. Just makes life easier. It might be a good idea to have your students use the same usernames and passwords too. Otherwise, requiring them to remember different usernames and passwords will be a challenge.

Next is the **E-mail** prompt. Press **Enter**.

Some more text scrolls by and stops at, e.g., the **Region (Schoolland)** prompt. Congratulations! You now are the proud owner of an installed OpenSimulator region.

Starting your OpenSimulator instance each time will be a matter of simply double-clicking the **OpenSim.exe** (or its 64-bit equivalent, **OpenSim.32BitLaunch.exe**) each time. It's a good idea to create a desktop shortcut icon so you won't have to go digging in the **bin** directory each time.



Go back to your OpenSimulator bin directory (e.g., **d:\OpenSimulator\opensim-0.7.1.1-bin\bin**) and right-click the **OpenSim.exe** file (or you know, the 64-bit one!). Click **Send to** and then **Desktop**. You now have a desktop shortcut to your OpenSimulator executable. You won't

have to go through the installation routine again, the program will just start running, with the settings you've just entered.

Logging Into Your Own Private Grid

If you're continuing from the first chapter, you've already installed the Hippo viewer. If not, you can't put it off any longer. Go back to "Downloading and Installing the Viewer" and follow the directions. When you have the Hippo or another viewer ready, continue with these steps.

Double-click the **Hippo** (or whatever!) desktop icon to start it.

Click the **Quick Grid Select Down Arrow** button and then click **local**.

If there's no **local** choice in the list, then shame on your viewer. See the instructions in the first chapter for adding a grid to the viewer and change the **Login URI:** information by typing:

http://127.0.0.1:9000

Whether **local** was on the list or you added it, it's now listed next to the **Quick Grid Select** button. Next:

In the **First Name:** box, type:

Beloved

In the **Last Name:** box, type:

Leader

And in the **Password:** box, type:

iamnumberone

Wait! Is OpenSimulator still running? If you didn't stop it after the previous step (look for that black box), then it is. Otherwise, double-click your desktop icon to start it.

Then click the viewer **Log In** button. You see a progress bar (you know what that means!) showing that things are happening! Again, if you get an error message stating **Your last location is not currently available**, **You have been moved into a nearby region**, or something like this, just click the **Close** button. You just created your OpenSimulator instance, so naturally you don't have a home or last location.

You're in and you're Ruth!

Ruth is the default avatar everybody starts with. Who did her hair? Why doesn't she have shoes? Or hips?? These are mysteries that will probably never be solved. Relax, take deep breaths, etc., you won't be Ruth for much longer.

Press the **Left Arrow** or **Right Arrow** key and take a quick look around. This won't take long, because you don't have much to see! All you have is a tiny bump of cyberturf.

To get a nice, flat, 256-square-meter region, you can do things the hard or easy way. The easy way is to use a file called **reactiongrid.oar**, which you should have downloaded along with this e-book. If you didn't, well, now's the time to go get it:

http://grid.reactiongrid.com/oars/reactiongrid.oar

An .oar is an OpenSimulator ARchive file, or a database backup. You can save an .oar to create a backup of a region, and you can load an .oar to restore a backup of a region. More on this later.

ReactionGrid has generously donated this file to get you started.

You have two choices. You can load the .oar from the Internet via the command:

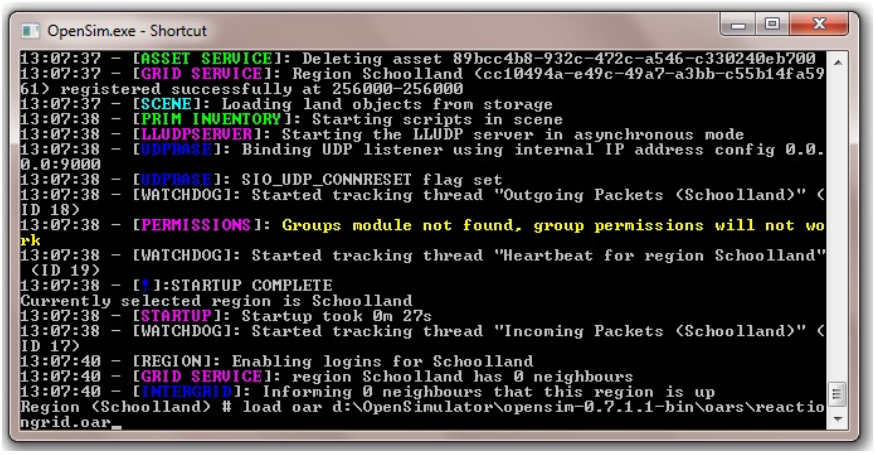
load oar http://grid.reactiongrid.com/oars/reactiongrid.oar

Or you can download it to your hard drive and then load it. YHA humbly suggests this. Download **reactiongrid.oar** and save it under your OpenSimulator directory, ideally in a subdirectory named **oars**. You'll be saving and loading .oar's later, so you might as well have it ready. So let's say you saved reactiongrid.oar as:

d:\OpenSimulator\opensim-0.7.1.1-bin\oars\reactiongrid.oar

At the OpenSimulator command prompt, type the load command and then press **Enter**:

load oar d:\OpenSimulator\opensim-0.7.1.1-bin\oars\reactiongrid.oar



Wait a few minutes. When it's finished, you'll see a message like **Successfully loaded archive**. Look around in the viewer again. If you don't see any difference, type **shutdown** at the OpenSimulator command prompt and then start OpenSimulator again. Then load the .oar file again. It's unlikely that your .oar will have difficulty loading at this point, but possible. Later, when your .oars get really big, like over 30 MB, load time might be a big factor. But don't worry about this now. Chances are everything worked the first time.

Now you have a 256-square-meter square! Walk or fly to the middle of the region (click the **Map** button and teleport there if you want) so that you can see the "shopping board." Lots of nifty outfits, skins, etc., to choose from. Your sad existence as Ruth is almost over!

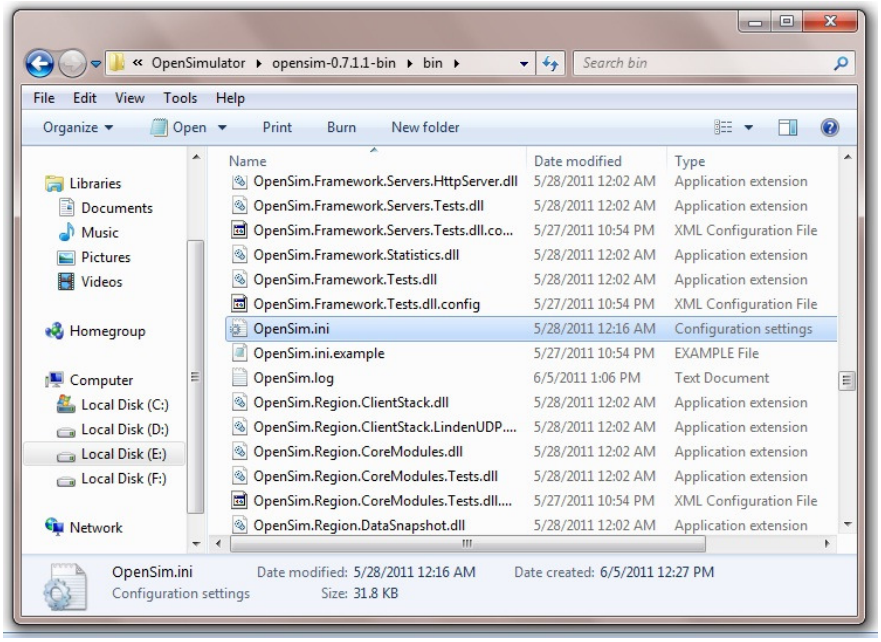
Move your mouse cursor over one of the pictures. You'll see text such as **Male Smart** or **Female Smart Casual**. Right-click the picture and the pie menu displays. Click **More** and then **Buy**. The **Buy Contents** dialog box displays.

Click the **Buy** button. What happens? Nothing, except that you get an error message stating that **Buying is not implemented in this version**. Click the **OK** button.

It's time for you to get your first taste of editing the **OpenSim .ini** file (named, appropriately enough, **OpenSim.ini**). This is an initialization file, a text file that controls different parameters when a program starts. You need to "flip a switch," as it's commonly called, so that selling things is enabled.

Please note that this does not mean that your OpenSimulator environment has an economy like Second Life, with a currency like Linden dollars, just in case you have the slightest idea what YHA is talking about. It just means that in order to "vend," or to allow avatars to give/accept stuff, the selling function has to be enabled. You are buying things, just for $0. Great bargain, huh? Don't worry about this too much right now. Relax, take deep breaths, etc.

Go to your bin directory, e.g., **d:\OpenSimulator\opensim-0.7.1.1-bin\bin**, and find **OpenSim.ini. E**nsure it's of **Type Configuration settings**.



You see the page-and-cog icon next to the left of the filename in the screenshot above? This means that .ini files have been associated with NotePad, the default text editor, which is what you need to do if the icon is blank on your screen.

But hold it right there! OpenSimulator's .ini files are formatted to be Linux-friendly, but in Windows they're a jumble. Instead of Notepad, you could use Notepad-Plus-Plus, a free utility which will make working with OpenSimulator text files much easier:

http://notepad-plus-plus.org/

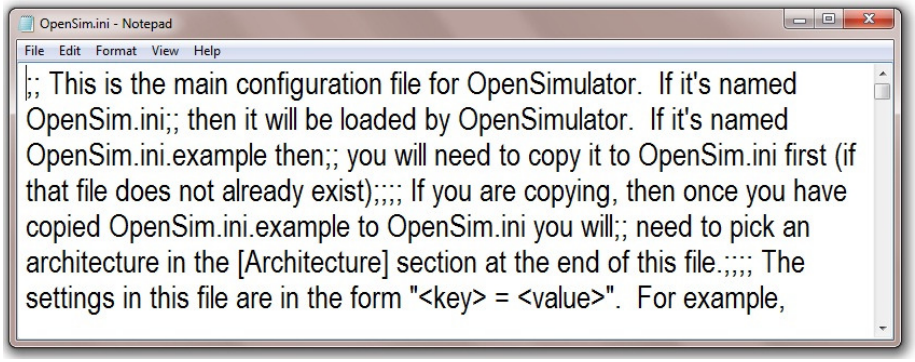
If you want to use Notepad-Plus-Plus, download and install it.

For this e-book's directions, we'll continue assuming you're using Notepad.

If you see the little page-and-cog icon, double-click the file.

If you see a blank, right-click the file and click **Open with**. **NotePad** might already be an option, if so, click it. If not, click **Choose default program**. If **NotePad** is listed, click it. Otherwise, click the **Browse** button and find the file **notepad.exe**. Once you have .ini files associated with Notepage (page-and-cog icon!), then double-click the file.

The **Notepad** window displays.

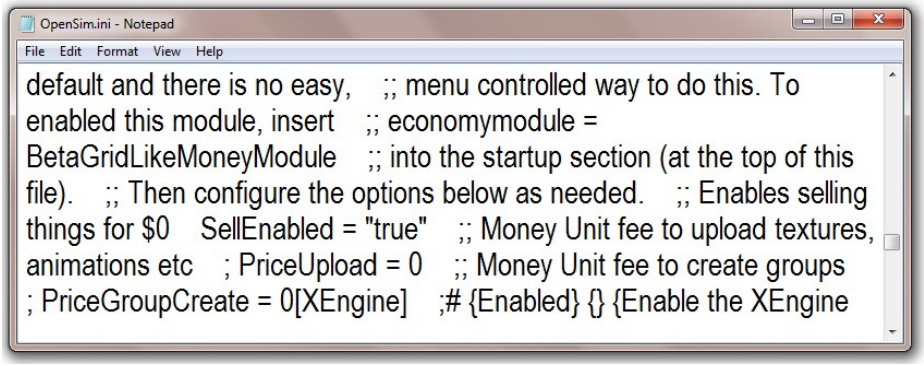


If you don't see text at the top such as **;; This is the main configuration file for OpenSimulator**, then you have the wrong file. Yes, the .ini file is a mess, which is one of the reasons why OpenSimulator is considered hard to set up using Windows. Do yourself a favor and get

Notepad-Plus-Plus soon if you haven't already. Click NotePad's **Edit** menu, and then click the **Find** command. The **Find** dialog box displays. Type **selling** in the text box and click the **Find Next** button. Close the **Find** dialog box. Go to the next line in the .ini file, where you see:

```
; SellEnabled = "false"
```

Delete the semicolon. The semicolon means the parameter is a comment and thus to be ignored. Deleting the semicolon means the parameter will take effect. Then select the word **false** and type **true** in its place. Look at this graphic to ensure you've got it right:



Click the **File** menu and then click the **Save** command. You've just "tweaked" your first .ini file. Not so hard, huh?

Now, the change won't take place until you restart OpenSimulator. Go back to the OpenSimulator window (it's black, remember?) and at the command prompt, type **shutdown** and press **Enter**. You'll be logged out of your Hippo viewer session, so click the **X** button to stop the viewer.

Now double-click the desktop icon to restart OpenSimulator.

And restart Hippo. All of your login information should still be there if you remembered to check the **Remember password** check box. Click the **Quick Grid Select Down Arrow** and then

click **local**. If you need to, type **Beloved**, **Leader** and **iamnumberone** again in the **First Name**, **Last Name** and **Password** text boxes, respectively. Click the **Log In** button.

Now go on a shopping spree! Right-click, e.g., the **Male Smart** picture. The pie menu displays. Click **More**, then **Buy**. The **Buy Contents** dialog box displays. Click the **Buy** button. Where's all your stuff? In your Inventory. This is you BEFORE:

Click the **Inventory** button. If you bought, e.g., Male Smart, then you have a **Male Smart** folder. Click the **Male Smart Folder Down Arrow**. Now right-click **Male Skin Fair** and then click **Wear**. Follow with **Shape**, **Underhair**, **Hair**, **Shirt**, **Tie**, etc. It's up to you.

You're not Ruth anymore! This (could be) you AFTER:

Now you have a client/local installation. But remember, only one avatar at a time can be in it! You can create other avatars, but each one will be alone using a client/local installation.

OK, if you just want client/local installations on your PCs, say in your computer lab, for example, then just copy your entire OpenSimulator directory (like **d:\OpenSimulator**) onto all your PCs. Each instance or installation will have the .oar-provided region and features, Beloved Leader as its god avatar, and buying will be enabled. But you will have to go through the installation procedure for the viewer on each PC.

But let's assume that you want to keep going. Next we'll create your client/server installation. You're going to use the directory you just created, so all that work was not for nothing! You're going to copy your OpenSimulator directory from your client machine to your server, with one change, and that is the IP address. Relax, take deep breaths, you know.

Creating Your Client/Server Installation

Server: Consulting and/or Checking

Now this is the step that is primarily responsible for giving OpenSimulator its undeserved reputation for being difficult to set up. Admittedly, if you don't have a LAN, then your job is going to be downright tough. You're going to have to set up a server, then connect to a hub, etc. Beyond the scope of this e-book. Hopefully you've got a LAN already. If you have a LAN server, with a static IP address, the rest is just a matter of editing another initialization file, which is easy enough once you get past the fact that the file itself is not formatted for easy reading.

Before proceeding, it's necessary to point out again that although you could take a regular desktop PC and make it into an OpenSimulator server, the performance is going to be poor, bad enough to make the experience unenjoyable if not completely impossible for your students. You need a machine powerful enough to handle concurrent users, namely the maximum number of students you will have in your classes.

Changchun American International School uses a server with these specifications:

- Dell PowerEdge 2900
- Intel Xeon CPU, 3.16GHz
- 8 GB of RAM
- ATI ES1000 Video Card
- Windows 2008 Server Service Pack 2

Please note that the video card does not meet the minimum standard requirements and so we can't run Hippo (or any other viewer) on this server. But this is OK. Later you could change the

video card in this situation, but until then you could run the viewer exclusively on your client PCs, because this is what you normally will be doing anyway. Typically, all you'll be looking at on your OpenSimulator server is your **DOS** or **Command Prompt** window. If you're not actually in the same room with the server, most likely you'll be using a Remote Desktop Connection to do this.

This server is great and not that expensive these days. YHA did some online research and found that you could get a server like this for around US$2,000 as of June 2011. Please point out to your manager, the one suffering a heart attack due to sticker shock, that the server could be used for other things besides OpenSimulator! A server with these specifications easily handles OpenSimulator classes of 20-25 students. Perhaps it could handle more than 25, but this has never been tested, at least not at CAIS! Even better news is that you could get started with a server that's a bit less powerful, one you could get for around US$1,000. Remember, getting started with a hosted solution could cost around US$300 the first month and up to $100 every month, for as long as you subscribe to the service. So within eight months you could've paid for a server, and once you pay for a server you don't keep paying! All a matter of what's most important to your school. It's very likely your school already has a server you can put OpenSimulator on and no purchase whatsoever will be necessary.

OK, back to installing OpenSimulator.

This e-book assumes two possible scenarios and provides instructions for both:

A. You have system/network administrator permissions in your school
B. You have a system/network administrator who will do the work for you

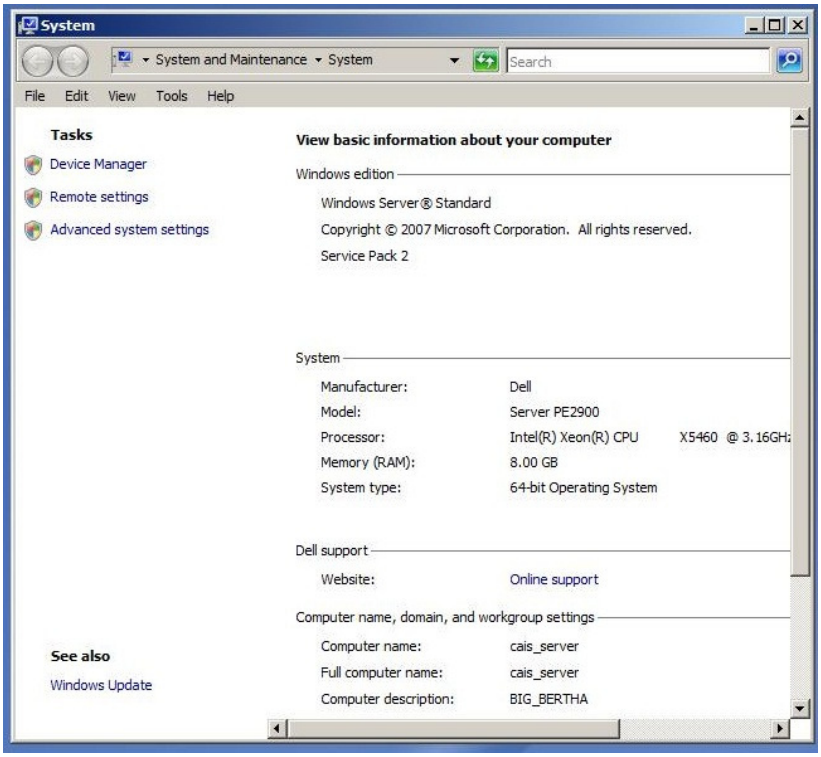
First, do you have a server with a static IP address on your Local Area Network?

If **Scenario A** is true, you already know the answer to this question. Let's say your server's address is 10.10.7.163, as the OpenSimulator server ("Big Bertha") is at CAIS.

If **Scenario B** is true, then you can ask your system/network administrator. Let's say s/he says there is indeed, and its address is 10.10.7.163.

You also need to know if your server is a 32-bit or 64-bit machine. If your server's specifications come close to the ones listed above, it's most likely a 64-bit. If you don't know, then maybe it's time to hand this e-book to somebody else! No, YHA is just joking. Just in case you and your administrator don't know, you or your administrator can go to the server desktop and right-click the **Computer** icon.

The **System** dialog box displays.



See **System type**? We'll assume your server is 64-bit like Big Bertha at CAIS.

Copying Your Directory and Tweaking the .INI

Scenario A: Take your OpenSimulator directory – yes, your client/local installation on your desktop or laptop – and copy it to your server. Yes, the directory you've been using to transform Ruth into another avatar, **d:\OpenSimulator**. Just copy it. Let's say you copy it to the **e:** drive of your server. Now you have:

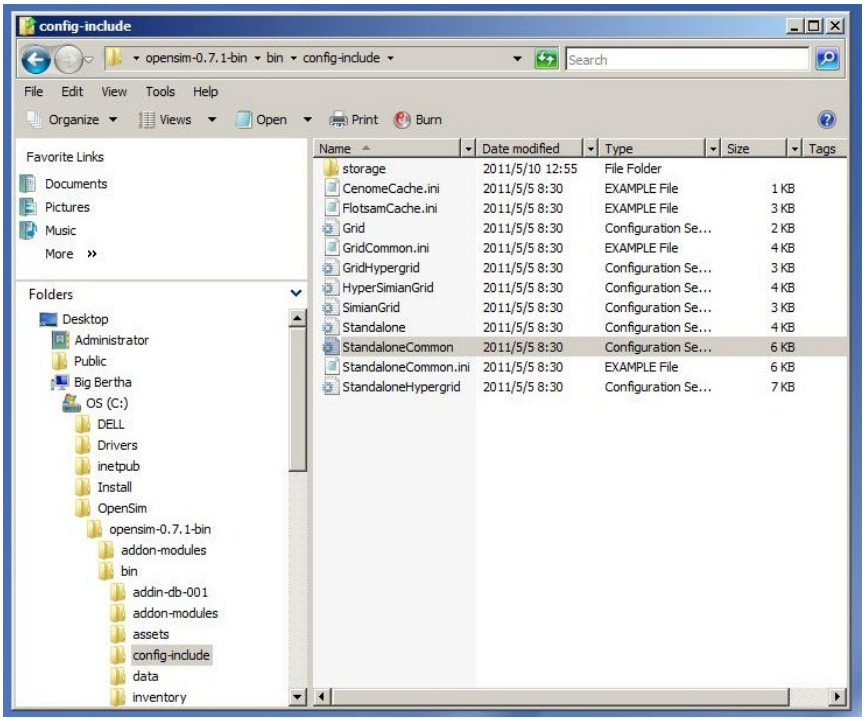
e:\OpenSimulator\opensim-0.7.1.1-bin

If you're **Scenario B**, go ahead and read this section, but relax, etc. We'll get back to you later.

Still Scenario A: Open the **bin** folder and then go one directory deeper, to **config-include**:

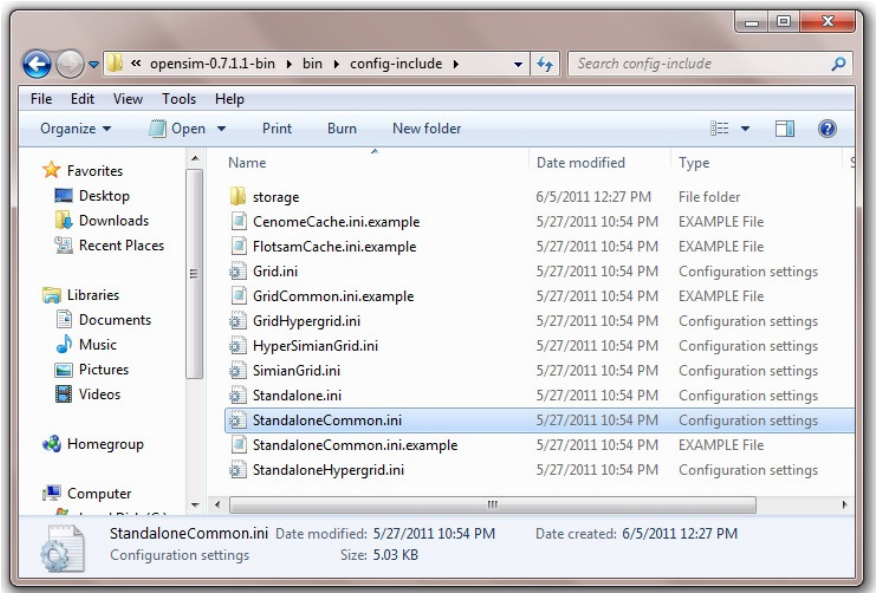
e:\OpenSimulator\opensim-0.7.1.1-bin\bin\config-include

Find the **StandaloneCommon.ini** file. Its **Type** is **Configuration Settings**. Now in the case of this file you have to be particularly careful. You're going to be looking at it on a server, and the settings might not be the same as on your client. Take a look at this screen capture from Big Bertha, showing the **config-include** directory from an earlier OpenSimulator 0.7.1 installation. The show file extensions switch had not been flipped on Big Bertha, as might be the case with the server in your school if you don't have control over it, or if you just never seem to be able to remember to do it like YHA.



The highlighted file is indeed **StandaloneCommon.ini**, but you only see **StandaloneCommon**.

If you were looking at, e.g., YHA's laptop, on which he did bother to flip the extensions switch, you'd see:



The key is to look for **StandaloneCommon** of the **Type Configuration Settings**. Now, why couldn't they name these files something like **EDIT_THIS_ONE** and **NOT_THIS_ONE**? It's a mystery. But you can see why someone who's not very geeky might have problems with this.

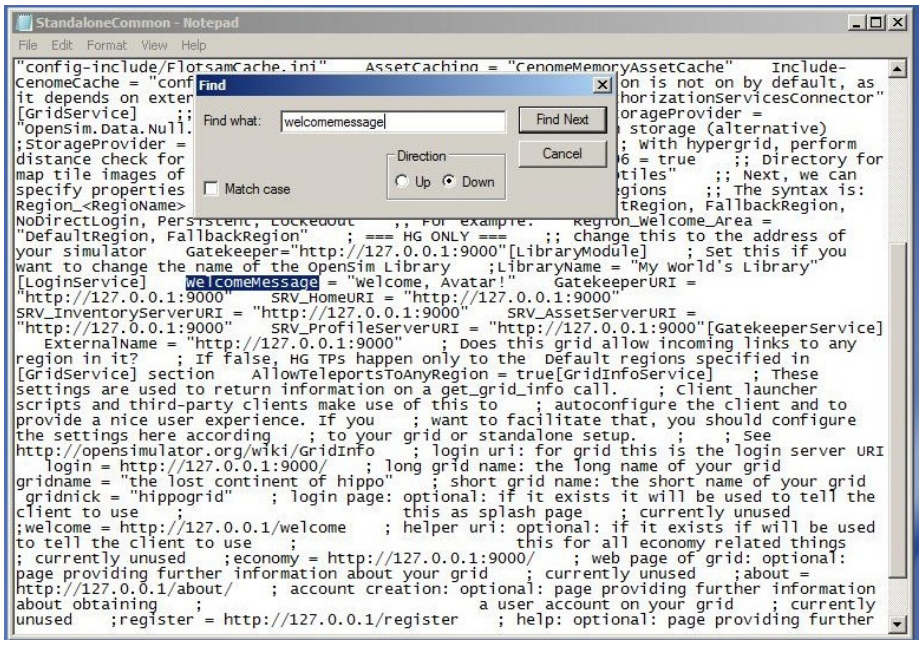
You see the page-and-cog icon next to the left of the filename in the screenshot above? This means that .ini files have been associated with NotePad, unless you've installed Notecard-Plus-Plus (http://notepad-plus-plus.org). As before, we'll assume the default of NotePad. You'll need to associate with file with some text editor if the little icon is blank. If you see the little page-and-cog icon, double-click the file.

If you see a blank, right-click the file and click **Open with**. **NotePad** might already be an option, if so, click it. If not, click **Choose default program**. If **NotePad** is listed, click it. Otherwise, click the **Browse** button and find the file **notepad.exe**. Then double-click the file.

The **Notepad** window displays. Click the **Edit** menu and then the **Find** command. The **Find** dialog box displays. In the **Find what** text box, type:

welcomemessage

Then click the **Find Next** button. The text is found:



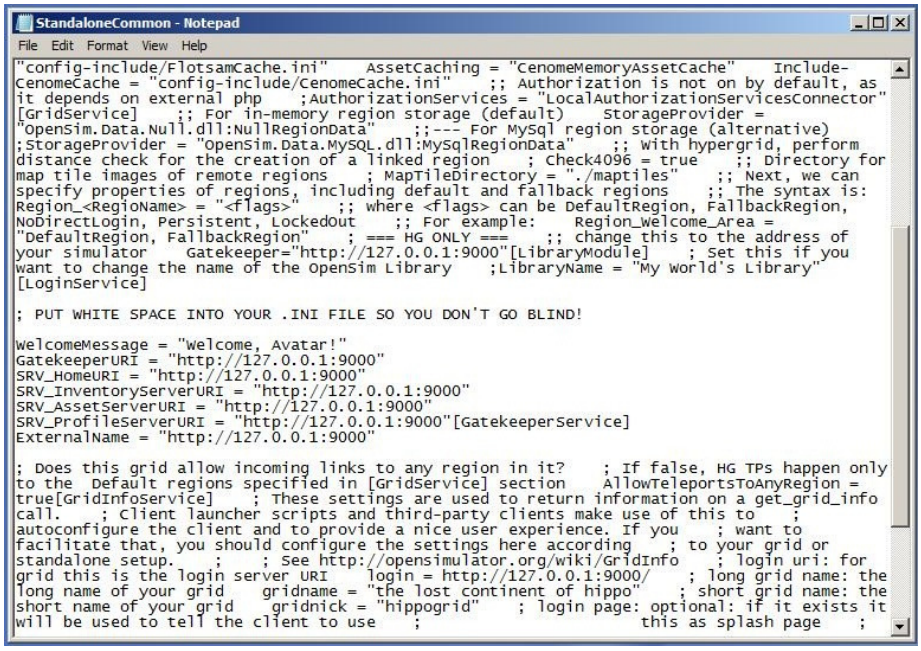
Close the **Find** dialog box.

Remember, if you're using NotePad, these .ini files can be hard to read, but you can help yourself by inserting some white space, which has no effect on the file's functionality. As you continue to tweak this and other initialization files, put in carriage returns and otherwise invest some time/effort in making them readable. It will pay off in the long run.

Place your insertion point just before **WelcomeMessage** and press the **Enter** key several times. If you want to type a comment such as **Start editing here**, or, as in the example below, **PUT SOME WHITE SPACE INTO YOUR .INI FILE SO YOU DON'T GO BLIND!**, ensure that you

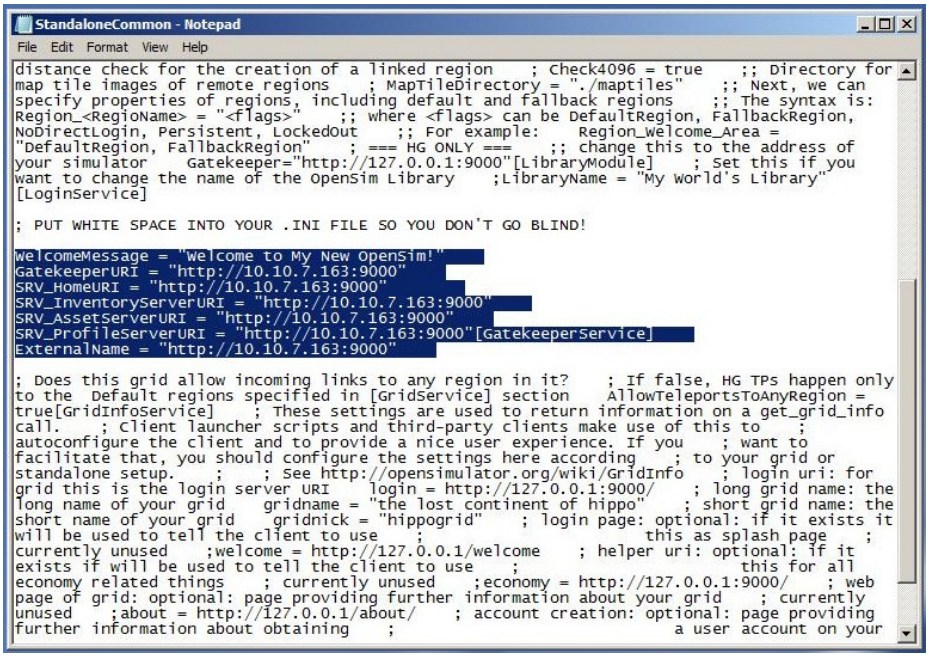
start the line with a semicolon, so that it's considered a comment. After the IP address for **ExternalName**, press **Enter** a few times too.

Now you see something like this:



Now replace **127.0.0.1** with your IP address, e.g, **10.10.7.163**. Keep the port number as 9000, unless you had to change this default value. If you wish, change **Welcome, Avatar!** to something like **Welcome to My New OpenSimulator!** Just be sure to put your message inside quotation marks.

Now your **StandaloneCommon** file should read something like this:



Click the **File** menu and then the **Save** command. There. That's it. Was that so hard? You think so? Relax, etc. As the product itself matures, so will everything else. What most teachers tend to want is an installation/configuration program. You know, an installer you can start and then click option buttons to choose this and that feature. Would be nice. Maybe one day.

Hello, **Scenario B** people? Wake up!

You haven't been forgotten! Think about what was just done. Everything in your OpenSimulator directory is the same for a client/local and client/server installation, except for the different IP address in the **StandaloneCommon.ini** file. So, if you have no access to or permissions for the server, first make a copy of your **OpenSimulator** directory on your client PC. That's right, if you set up OpenSimulator in:

d:\OpenSimulator\opensim-0.7.1.1-bin

Then copy it to another drive like **c:**, e.g.:

c:\OpenSimulator\opensim-0.7.1.1-bin

Then go to the **config-include** directory and edit the **StandaloneCommon.ini** file, replacing the local IP address, **127.0.0.1**, with the server IP address, e.g., **10.10.7.163**. Just like it was done for Scenario A. Then give this OpenSimulator (on **c:**, remember) directory to your system/network administrator so s/he can copy it onto the server for you. Then delete the copied directory (the one on **c:** drive) on your client PC. It will just confuse you later! That's it, you're done! Your administrator will copy the directory onto your server and you'll be (almost) ready to go.

Starting the Executable and Adding Your Grid

Starting the Executable

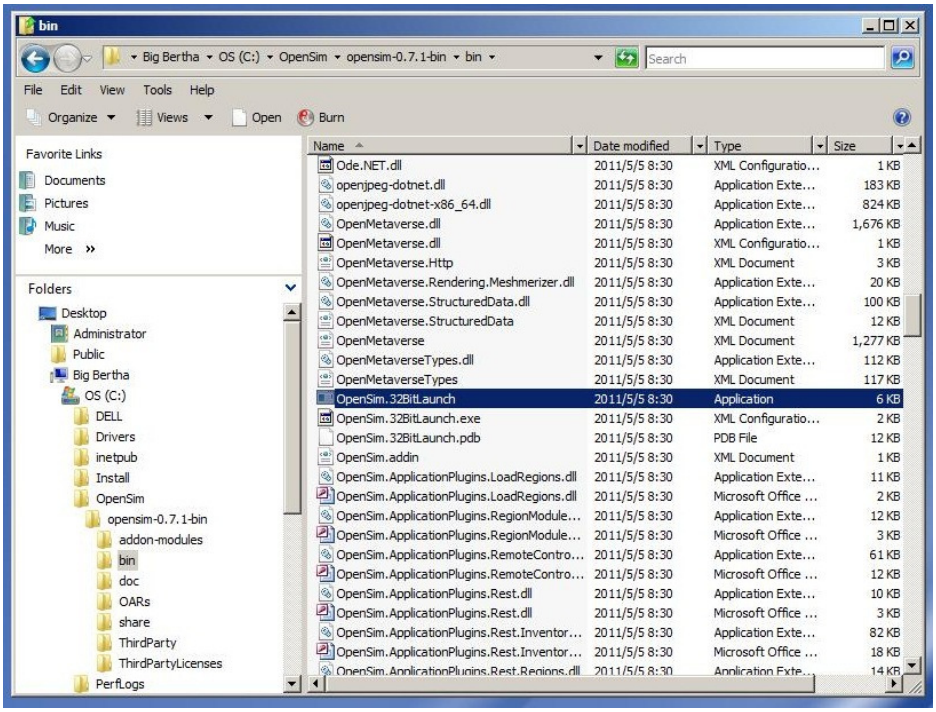
Now it's time to start OpenSimulator again.

If you're a **Scenario A** person, go to:

e:\OpenSimulator\opensim-0.7.1.1-bin\bin

And find the **OpenSimulator.32BitLaunch.exe** file, of **Type Application**. Of course, should you by chance have a 32-bit server, you will want to use **OpenSim.exe** as you did in the client/local installation. Double-click the file. Or have you created your desktop shortcut?

This screen capture again shows an earlier installation on Big Bertha. You don't see the **.exe** extension, but you do see **Type Application**:



Why don't you create that desktop shortcut now if you haven't already so you won't have to go digging around in directories (and possibly double-clicking the wrong file)?

The text scrolls by in the command prompt window and stops at the **Region (Schoolland) #** prompt.

And back to **Scenario B!**

Of course, if you're a **Scenario B** person, you will have to ask your system/network administrator to start it for you. Inconvenient for you? Yes, but also for your administrator! It won't be long before your administrator gives you permission for a Remote Desktop Connection to the server (sorry, setting this up is beyond the scope of this course!) so that you can start and stop OpenSimulator on your own. A Scenario A person will want this too if the server is, e.g., off in some server room somewhere. Remember that desktop shortcut so you won't have to rummage through directories each time! Later you'll be saving and loading .oar's too, but never mind this now.

You've done it! All the other configuration work was done during the client/local setup. Now you can call all your geeky friends and brag that you set up an OpenSimulator client/server installation!

Adding Your Grid

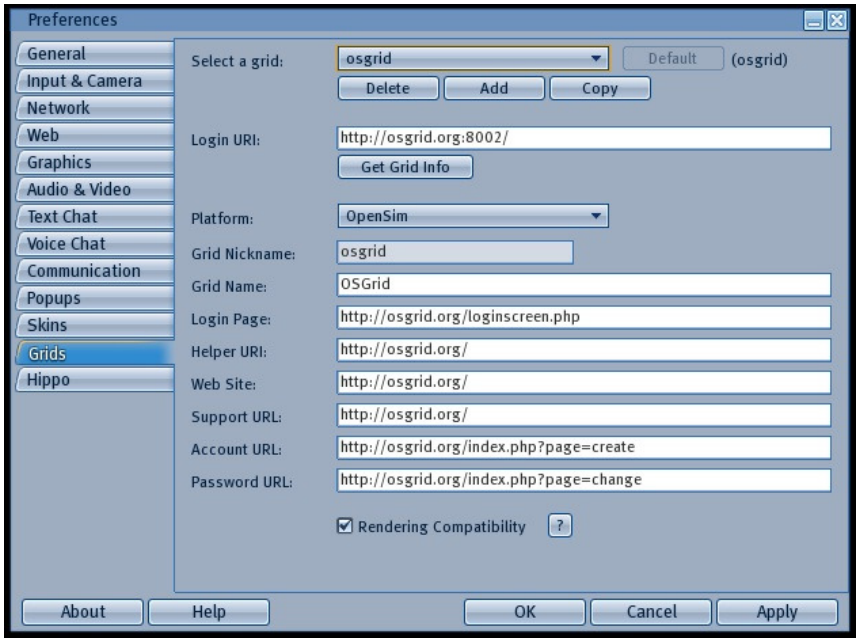
Whoops! Have you installed Hippo or your viewer of choice on all yours client PCs yet? If not, time to do this. Or at least to install some viewer. Allow for 5-10 minutes per PC if you choose Hippo, although if you can get the downloaded files to download faster, you might shave off some of this time. As far as YHA knows, an individual installation has to be done for each PC, regardless of viewer chosen. If you know of a way to batch this, please share.

Pause for a moment. Think about whether or not you want your students to use the viewer in their native language. For example, a Chinese kid could use the Chinese version. If this sounds good to you, then install Imprudence instead:

http://wiki.kokuaviewer.org/wiki/Imprudence:Downloads

Remember, you have other choices! And you can always change later. Go back to the first chapter's "Downloading and Installing the Viewer" topic if you need directions again for getting Hippo going. After you have Hippo installed on all your client PCs, the next step is to add your new grid to all the Hippo viewers. Or you can do the installation on each PC and then add the grid, it doesn't matter, up to you. The grid adding process only takes a minute but must be done manually for each machine. No, there's no way to automate or batch it, at least as far as Your Humble Author is aware. Relax, take deep breaths, etc.

Start the Hippo viewer on each PC and click the **Grids** button. The **Preferences** dialog box displays.



Click the **Add** button.

In the **Login URI:** text box, type your server's IP address. Our example is:

http://10.10.7.163:9000/

9000 was the default port number, remember? If you had to change this port number when you set up OpenSimulator, change it here too.

In the **Grid Nickname** text box, type something like:

My New OpenSimulator

Or whatever you like, although keeping this somewhat brief is a good idea. Click the **OK** button. You now see **my_new_opensimulator** next to the **Quick Grid Select** button.

Now you're going inworld.

Type the first and last names of your recently created avatar in the respective **First Name** and **Last Name** text boxes. Type your password in the **Password** text box. Click the **Log In** button.

You see a progress bar which shows you that you're making, well, you know, right? Yes, progress. You might get an error message about this or that region not being available or that you've been moved to another region. This has been corrected in some viewers. Anyway, if you get such a message, just click the **Close** button.

You're inworld, in your new client/server installation! This is the one your students will be able to log into, just as soon as you add your grid to all the Hippo viewers on all their PCs!

You did it! Congratulations. Wear your plastic pocket protector, highwater pants and mismatched socks with pride, for today you are truly a GEEK!

Exercises

Just kidding. If you've followed this section's directions, installing a client/local and client/server setup of OpenSimulator, plus installed and configured Hippo on 20 or so computer lab PCs, you've worked hard enough already! Didn't seem like that much effort, right? Time flies when you're having nerdy fun! Now the techie part is over, time for the teacher track, via which you'll get some tips and tricks for getting started with OpenSimulator in your classes.

Lesson Plans/Next Steps

What Are You Supposed to Do?

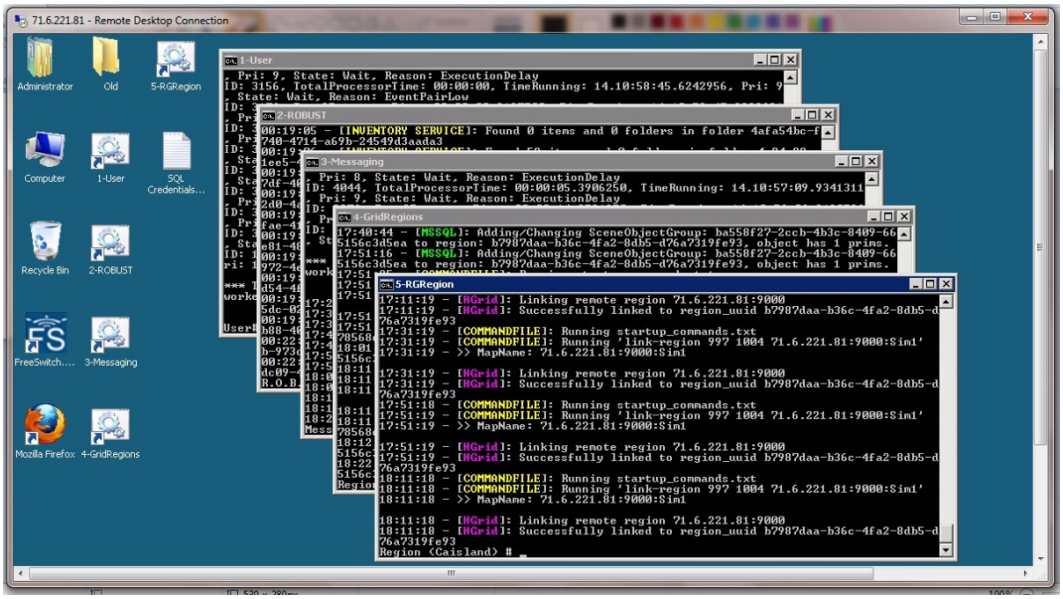
OK, now you have an OpenSimulator installation of some kind up and running. Unless you opted for the client/local strategy, all your students can connect, communicate and collaborate with each other. We're assuming you chose external host or in-house client/server. First, we need to cover some more geeky stuff, then administrative and other matters, then we'll get into best practices, actual lesson plans, etc. Just start reading. Begin with the introductory information and then peruse the lesson plans or whatever as desired/required.

YHA needs to manage expectations. This is not going to be a master's degree thesis on pedagogy, or indeed anything else. Some of the theories, methodologies, etc., will be mentioned, but not in detail. YHA will be writing all those books later, you'll just have to be patient! This chapter includes just enough information to get your OpenSimulator classes started. The "More Resources/Further Study" section lists lots of websites, books, etc., that will provide other handy suggestions and background information.

Some More Geeky Stuff

One of two scenarios is going to be true (or maybe both, like at CAIS, but never mind this now). Your OpenSimulator installation is either running on a host server or in-house server. If it's on a host server, then just about everything server-related is going to be someone else's, not your, responsibility, except for class maintenance activities such as creating users, saving/loading .oars, etc. You'll be responsible for everything client-related, of course, and remember, the viewer is what you'll be spending the most time/effort on because this is what your students will be using.

Chances are you will be establishing a Remote Desktop Connection and working with your host server via an interface that looks something like this:



Don’t panic! You’ll be taught what to do by your host. Or you can wait until YHA’s next e-book! The point is that all this is beyond the scope of what we’re trying to do here. We’re going to try to briefly cover “A Day in the (Nerdy) Life” of a teacher, assuming you’re managing your own OpenSimulator server.

This assumption includes that you’re running a “standalone” OpenSimulator installation, not a grid, and so you use only one DOS or Command Prompt window, not more, as you see above!

Consider Big Bertha, the OpenSimulator server at CAIS. Every morning YHA boots her up, every afternoon YHA shuts her down. Big Bertha is only accessible via our LAN, i.e., inside the school itself. It is possible to make Big Bertha accessible via the Internet, but school policy forbids this (and rightly so, in YHA’s humble opinion). As of this writing, OpenSimulator is being used daily by all MYP classes, regularly by one PYP grade and intermittently by two others. We use just one region per class, although it’s possible to have more than one while keeping the standalone (versus grid) configuration. It’s just easier this way.

You will need to create user accounts for your students, via the (imagine this!) **create user** command, again in the "black window" aka the DOS or Command Prompt:

create user Suzy Qiu

You'll need to enter a password, then just press **Enter** for everything else. The defaults are fine. Has YHA mentioned getting your kids to write down their usernames and passwords?

Before each class starts, the appropriate .oar is loaded. If it's June 8th and MYP 5 had its previous class June 6th, the command is something like:

load oar e:\opensimulator\oars\myp5-06jun.oar

"Something like" meaning that your path, e.g., could be different. You get it. You're going to need naming conventions so files will sort properly, you can find them, etc.

At the end of the class, the .oar is saved via a command something like:

save oar e:\opensimulator\oars\myp5-08jun.oar

And, if your little darlings don't write down their passwords, and sometimes even if they do, you will be regularly using the **reset user password** command:

reset user password Suzy Qiu newpassword

And that's pretty much it, Monday through Friday. Most of your focus is going to be on using your viewer on clients, particularly during classes.

When YHA gets the time, the goal is to switch from the default SQLite database to MySQL or SQLServer Express. We suspect the main reason that .oar's start taking a long time to load above 30 MB in size is that we're still using SQLite. But the interim solution is to simply shut down (using the, you guessed it, **shutdown** command) and then restart OpenSimulator for each class. This only takes a few minutes and is much quicker than the load waiting otherwise.

Again, if your viewers start taking a long time to connect, trying clearing the cache. On the **Preferences** dialog box, click the **Network** tab, then the **Clear Cache** button. You'll have to stop and then restart the viewer. It'll be worth it to do this on a regular, e.g., weekly basis.

Just a bit more about hardware before we move on. YHA tends to use the term "classroom" interchangeably with "computer lab," because most of the time there's no difference for IT teachers. There'll be a note on the distinction later. Every CAIS classroom, lab or not, comes equipped with a SmartBoard. Nice, huh? You don't need one of these per se in your lab, but you should have at least a projector and screen. Last year, when CAIS had only one computer lab, we kept Big Bertha there. Now that we have two labs, we keep Big Bertha in YHA's office.

So much for the techie stuff, hardware-wise anyway, now it's time for some teacher stuff, but it'll still be necessary to discuss software! Since it's impossible to cover everything possible, YHA is going to continue to describe how we do it at CAIS and you can decide what's right and not right for your OpenSimulator classes.

Administrative Matters

"Administrative matters" is usually just a fancy way to say "paperwork." CAIS is an International Baccalaureate Organization (IBO, **http://www.ibo.org**) World School, and when discussing bureaucratic issues, sometimes it's difficult to separate IBO requirements from CAIS procedures. Please note that YHA does not represent, purport to speak for, etc., the IBO! Also, we're talking about MYP Technology classes here, not PYP ICT or DP ITGS. If you're in an IBO school, you can "read between the lines;" if you're not, the IBO aspects of the plans don't matter to you, so the goal here is to be as generic as possible. In general, YHA keeps paperwork to a minimum for several reasons, mainly for the sake of principle, but also the fact that many of his students can't read or write English! Now that's a good reason, huh?

Speaking of not killing trees, in our CAIS Technology classes we generally do not use hard copy textbooks. E-books are preferred, and for each major project subject, at least one e-textbook (one that's as easy to understand as possible) is featured on each PC desktop. Sometimes hard copy reference books will be kept in the lab, e.g., the **Joomla Bible**, as a supplement to the e-books. Students are encouraged to use the Internet for solving problems, because, among other reasons, they can search in their respective native languages.

CAIS runs on a six-day cycle, with approximately 30 cycles per year, or 7-8 per quarter, 15 per semester, give or take a cycle. Report cards are issued each quarter. During each cycle, three Technology classes are held, and each class lasts 50 minutes. So there are 2.5 hours in what passes for a week, very close to the three hours per week YHA was accustomed to teaching in universities, therefore YHA tends to equate a cycle with a college class. All activities are timed accordingly, with many if not most assignments featuring cycle-by-cycle goals.

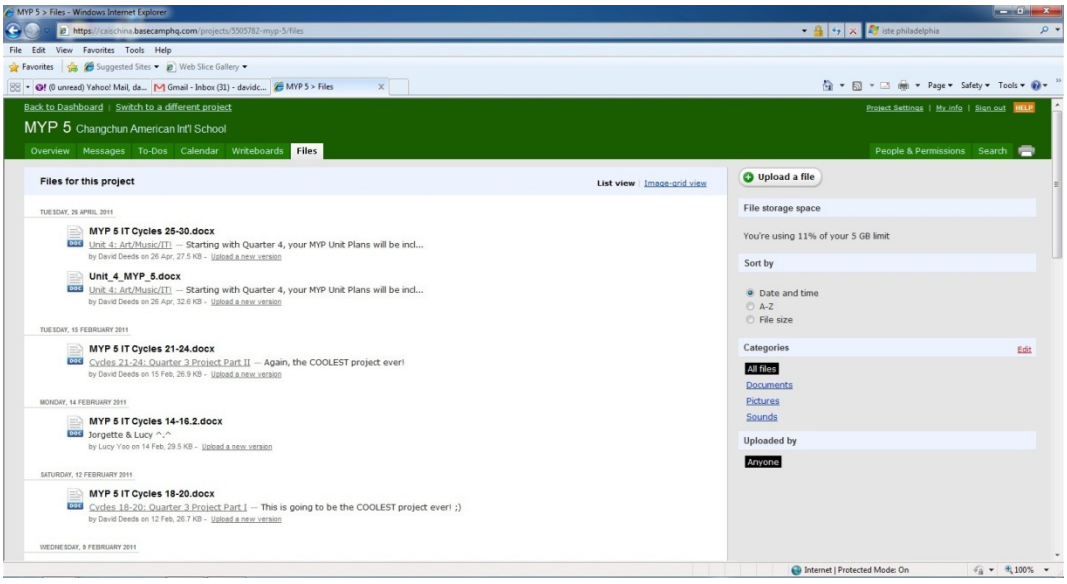
YHA has found for OpenSimulator classes that it's best to make each quarter a discrete unit, with a project to be completed for each one. Projects feature different "strands," so that it's not just OpenSimulator every time. But the OpenSimulator strand is always present because this is the one we know all our students can do. Remember, YHA started using 3D virtual worlds in the first place because students who are not fluent in English can succeed despite the language barrier. Each project equates to one or more lesson plans. Less or more time usually doesn't work: with less, students rush and skip steps; with more, students lose focus. Projects, not tests. There are NO TESTS in CAIS OpenSimulator (or indeed any Technology) classes, because tests are silly (not to mention outdated, ineffective, etc.).

Instead, students are assessed formatively (see, YHA already wrote his master's thesis!) via observation sheets, and also summatively by comparing results (namely, deliverables according to specifications, completed by deadline) with the assignments. IBO Technology classes are conducted according to what's called the Design Cycle: Investigate, Design, Plan, Create and Evaluate. It's brilliant for a wide variety of reasons, just one of which is that it's darn near the Software Development Life Cycle (SDLC) that YHA and many fellow geeks know so well. Anyway, each student is observed in class, cycle by cycle, and notes are recorded. Is Suzy Qiu, e.g., doing a good job of investigating, designing, planning? Informal tips for improvement and other feedback are provided throughout. The Design Cycle is what students do every day, it's how they're assessed formatively every day, and it's how they're assessed summatively every unit/project. YHA loves The IBO Way.

As mentioned, each project involves one or more lesson plans, which closely if not exactly follow the Design Cycle. For example, if a project lasts six cycles, the first three could cover Investigate, Design and Plan, the second three Plan, Create and Evaluate. Or you could make each "mini-cycle" (half the quarter, e.g.) into a cycle by itself. You get the idea. Each lesson plan

specifies checklists, addressing deliverables and deadlines. Before each end-of-quarter effort starts, students are given a rubric to indicate different point assessments for different performance levels. Students get a grade for each step of the Design Cycle. YHA tried giving students project grade sheets, but these documents were considered redundant, since the end of each quarter/unit corresponded with a report card being issued anyway. Each report card includes suggestions for improving performance for the following unit.

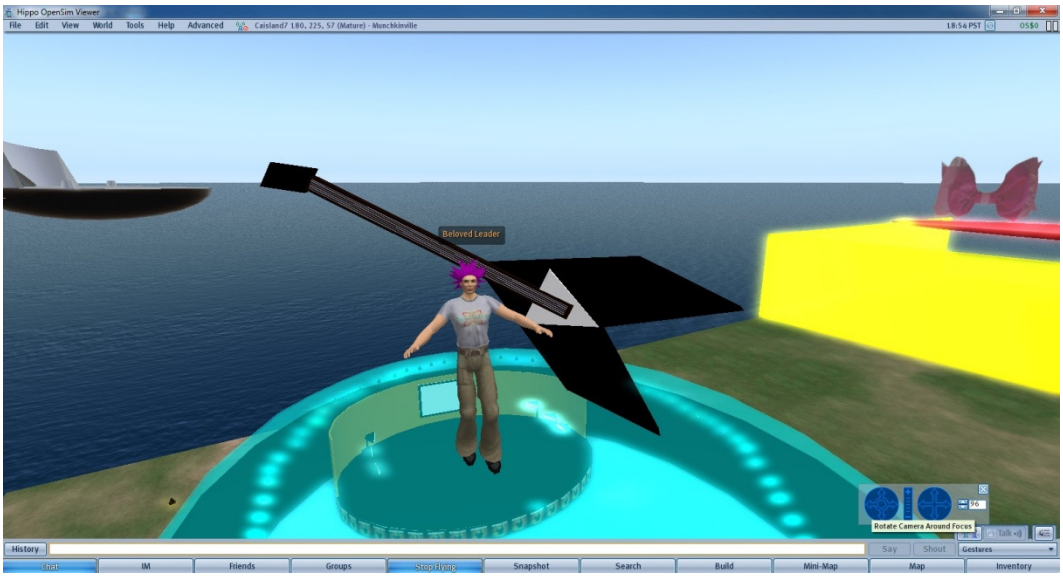
At CAIS, we use Basecamp (**http://www.basecamphq.com**) to manage our Technology classes. Students get their lesson plans via Basecamp after receiving an e-mail notification. When they finish completing the typed portion of their lesson plans (Word documents), they upload the lesson plans back to Basecamp. The lesson plan usually is also the project plan, as well as other deliverables. As much as possible is done electronically; sometimes students prefer to sketch, e.g., a design with paper and pencil. If something produced can be uploaded, then it is: a screen capture, program code or whatever. Remember, for OpenSimulator classes, the .oar's stored on your server will be your most valuable record of progress. Oh, yes, if you want a free, excellent project management program, use OpenProj (**http://www.openproj.org**). Just keep in mind that it'll be hard for your non-native-English-speaking students to use.



Enough with the paperwork! On to those first classes.

Those First Classes

OK, guess this technically counts as paperwork too, but make sure you have your student accounts created before your first class starts. Have your students write down what they want for usernames and passwords, and if it's not obscene or otherwise objectionable, go ahead and take care of it. Otherwise, your first class will be a waste. Your students will be driving you and each other nuts because some will be inworld while others are waiting.



In some ways, your first class should be like most of your others. YHA will typically enforce an "all-monitors-off" policy for the first 5-10 minutes. This is the only time during which YHA could be even vaguely considered to be lecturing. The time is used to issue class instructions and/or demonstrate something new. If you're an IT/ICT/Computer Teacher, you know that once a student is within arm's length of a PC, you no longer have his/her full attention. During YHA's university days, a routine was established of starting each class in a regular classroom for the "briefing." Then and only then did YHA allow learners into the lab to work on their assignments. Room shortages have prevented doing this at CAIS, until our new floor was

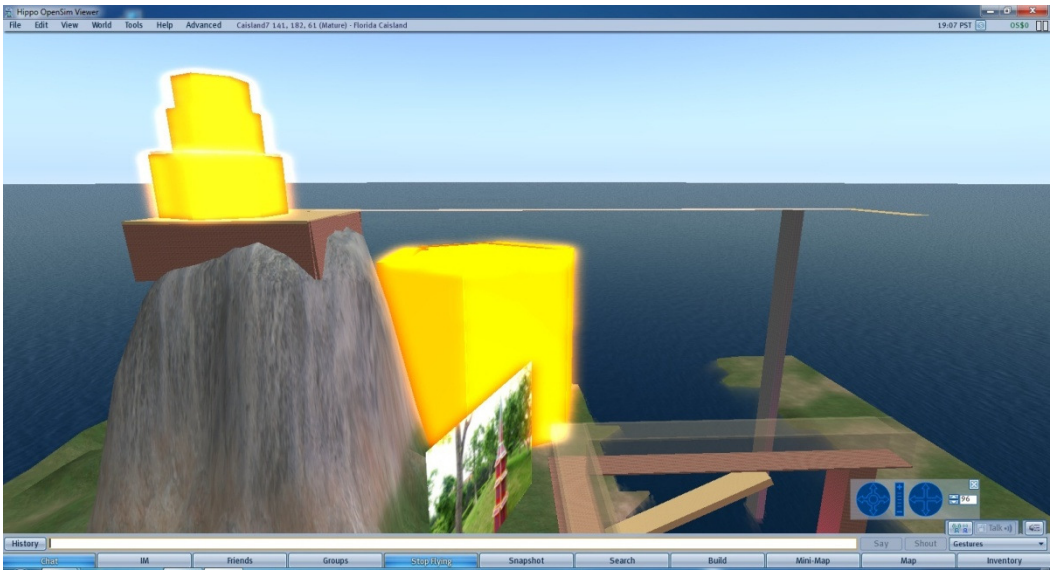
opened. Now YHA is considering this strategy for all his classes again. But YHA digresses. He's prone to that, as you've noticed. If students are in the middle of a project and no start-of-class instructions are needed, YHA will simply tell them to "Sit down and get to work!"

Now where were we? Oh, yes. First class. Show them how to start your viewer. This includes changing the language, if applicable. Show them how to log in. Then sit back and watch the craziness. Don't count on anything useful getting done for at least the first cycle, maybe two. Whether your students are three or 23 years old, they will GO WILD. Picture the younger munchkins running around the lab, squealing with delight. It's pretty comical. The thing to keep in mind is that although students will, regardless of age, eventually calm down, their level of enthusiasm for your OpenSimulator classes will be consistently higher than for anything else you've ever taught during your career. At CAIS, we literally have to drag kids away from their computers when classes are over!

YHA recommends not citing any rules regarding appearance, because naturally breaking them will be the first thing students want to do, although a ban on nudity should be enforced for simple propriety's sake. Imagine the reaction our principal had when she first walked by the lab

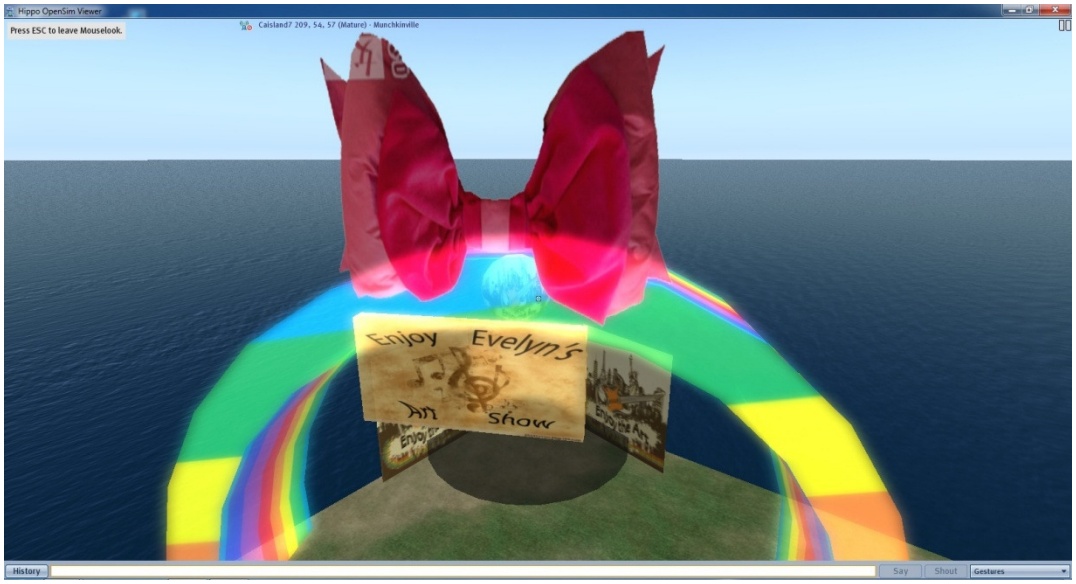
and heard YHA exclaim: "Hey! Put some clothes on over there! And not just underwear, I want to see PANTS!" The guys will experiment with making themselves look grotesque, but most of them will settle on a somewhat normal appearance soon enough. Not that this really matters that much. Gals will want to look like normal humans but always want to get more clothes, so teaching them how to create their own makes for a topic of endless fascination later.

After Suzy Qiu has changed her hairstyle 25 times, you will probably need to call a general moratorium on appearance editing. It takes students a while to settle on a "look," so tell them the halt is temporary and they can get back to their appearances later. Students will quickly discover that they can push each other, and weirdly enough, it's the older learners you'll probably have to tell repeatedly to cut it out. Then there's the territoriality you have to contend with. 256 square meters but they just can't keep off each other's patch! You might need to draw some boundary lines. Remember that you can always expand your property by building platforms above the default land. This will settle many cyberturf battles.



Speaking of land, be sure to disable "terraforming," or the ability to change the terrain. Using your god avatar, right-click the land (anywhere), click the **Options** tab, and then clear the **Edit Terrain** check box. There are other advanced and admin functions, but yes, you guessed it: next e-book. The reason you can't allow students to change the land without supervision is that as soon as they discover they CAN make mountains and lakes, this is literally ALL each kid is going to do until you pry the mouse from their fingers. Within minutes, your entire region will be useless to try to build anything on. Keep your god avatar logged in on one lab PC, and allow a team leader to use "god" to edit the terrain under your watchful eye, only after s/he has proven they have some kind of design or plan.

Kids will also go nuts when they realize they can make trees. You'll just have to insist they cease and desist at a certain point. Even "god" can't stop them from doing this. They'll get over it.



During these first few classes, it's important that you establish your role, namely, as more of a coach than a traditional teacher. Stay on the move, pausing to answer questions, explaining how to do this or that, etc. Keep in mind that if you have 15+ students and it's a 50-minute class, you need to not get bogged down. You should try to divide your attention evenly among the

students, especially at first. If you can't figure something out in a few minutes, tell the student you'll get back to him/her later. This will become less important as the students grow more accustomed to being self-reliant regarding project work. Most of them will be doing their own thing and you can spend greater amounts of time with the few who need extensive help with various matters.

Now Keep Going

You'll no doubt need to do some "just-in-time learning" yourself during your first months of OpenSimulator classes, unless you get a lot of advance time to prepare, and if your schedule is anything like YHA's, you seldom if ever have this luxury. During YHA's corporate days, it was not uncommon to have a conversation like this:

Boss: "What do you know about X technology?"
YHA: "Absolutely nothing."
Boss: "We need you to teach a class on it next week. Don't worry, [they] won't know either."

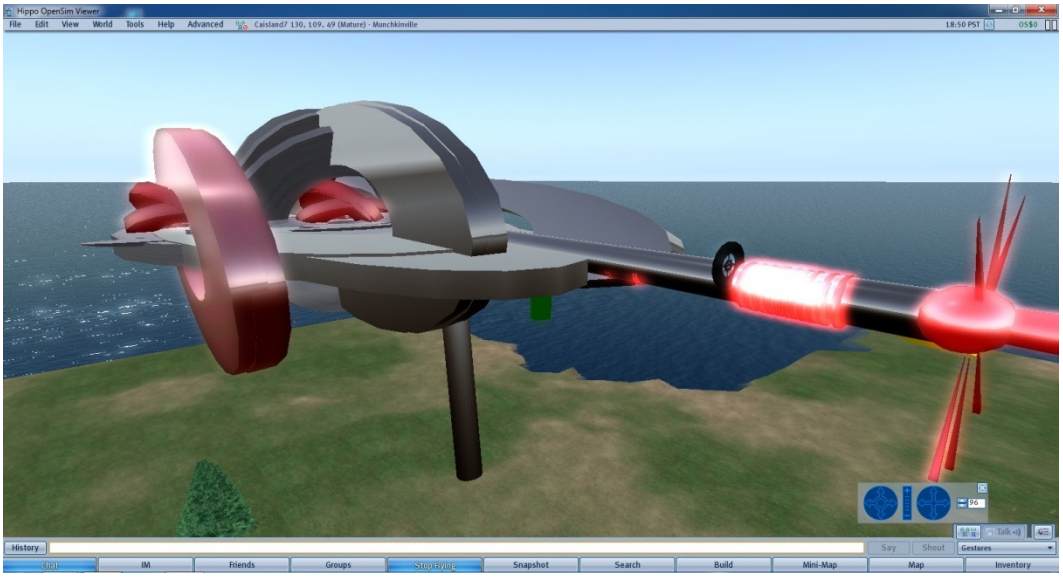
You get the idea. What YHA did for his first 3D virtual world classes was give assignments that corresponded with what he needed to learn how to do too. Tell your students that OpenSimulator is new to you as well, and that you'll be learning together. (The Ph.D. candidates are thinking: "Aha! Reciprocal guidance!") Start off with an overview: interface/navigation exercises, then building and programming exercises, as in the first chapter.

Then move on to more complex building assignments. How do you build a house? OK, you rez a cube, you stretch it, you hollow it out. Now, how does one create a door? Cut the cube? Well, you don't get a straight edge that way. How about if you create separate pieces and then want

just one? Oh, you can link them! (The Ph.D. candidates are thinking: "Yes, that's problem-solving, aka task-oriented instruction. Definitely this is learner-centric, active learning. It's the ultimate social constructivism, for students are literally – well, virtually – creating their own reality. This Deeds dude is a GENIUS!" It's true, so why deny it?)

Stick with building for a while, then get back into programming. By the time your students are getting really good at building things, they're going to want to make their creations do cool stuff and that'll be the point you know they're ready to get into some serious coding. Even if you've used Scratch and Scratch 4 Second Life to introduce them to programming, use S4SL again and have them analyze the code it generates. Then have them tweak existing scripts. Soon they'll be cranking out their own LSL programs.

So just, you know, relax, take several deep breaths. Everything's going to be OK.



When your students ask you a question you don't know the answer to, just tell them you don't know. Then encourage them to help you investigate, or better yet, to do it themselves. You'll be amazed at what puzzles your students can solve on their own once you are willing to LET GO.

YHA will often profess to be ignorant even when he's actually not (OK, rare occasions!) just to see if the little darlings can figure it out. Some of the more impressive "how-to" discoveries are going to be made by your students, so be prepared to shift your ego into neutral.

Chances are you're an experienced IT/ICT/Etc. instructor and you've been teaching the following for years:

- Computer-Aided Design
- Multimedia
- Programming
- Project Management

Everything that worked for you and your learners before will work now, and it'll be even better/easier, because with OpenSimulator all of your student activities will include a HOLISTIC aspect impossible under any other circumstances. Nothing will be an end in itself. With a CAD program, you can design, e.g., a car. That's it. With OpenSimulator, you can design a car, then write a script to make it move, then hop in it and take a ride! With a sound recorder and a camcorder, you can make digital audio and video. With OpenSimulator, you can produce 3D cartoons, with students learning CAD, programming, etc., simultaneously ! You can teach a programming language, and students can wait months for the "payoff," i.e., the chance to see their code do something. With OpenSimulator, the payoff is immediate, and you can't take a submarine ride with, e.g., a Visual Basic program after you get it to work!

The effectiveness and convenience of teaching project management will probably be the biggest advantage you notice OpenSimulator offers. Everything with OpenSimulator has a SOCIAL component that you simply can't duplicate with any other tool, in any other environment. Soon

your students will be working together to a degree you never dared to imagine before. And then there are the virtual field trips, taking kids off to visit students in other schools, other cultures, other countries! Conducting language exchanges so that your students teach others their native languages and vice versa! OpenSimulator makes interdisciplinary units, or cross-curricular instruction, easy and fun. Are students learning about Earth Day in their Science class? Then have them build windmills that spin and greenhouses that steam in OpenSimulator. See the art and music lesson plan coming up for more inspiration.

But YHA is getting back into "sales pitch" mode again. Time to move on to the best practices.

Best Practices

OK, to prevent this chapter from becoming a book, these CAIS Technology Teachers' Best Practices will be summarized as bullet points. The tips have been sorted into alphabetical order so you won't get the impression one is necessarily more important than the others:

- Allow for student interpretation of assignments. As you'll see next, it's a good idea to keep your lesson plans as mere skeletons for your students to "flesh out." The overall assignment could be, e.g., to build a green (eco-friendly) city. What exactly does this mean? Well, have discussions with your students, then let team members discuss the issues among themselves. If students seem like they're deviating from the overall goals, then gently nudge them back in the right direction. Otherwise, give them some slack and see what develops. You will have to occasionally tell students to STOP and move on, because they will get stuck at certain points in the Design Cycle.

- Combine individual work and teamwork, but emphasize the latter. Nothing beats OpenSimulator for teaching project management. You really can't help but design your assignments so that students must communicate, collaborate and cooperate in order to succeed. Project management teaching tends to seem "forced" sometimes, but OpenSimulator enables it to happen naturally and almost effortlessly. Let them divide themselves into teams, as long as cliquishness doesn't dominate. You might have to assign teams at first to get the proper skill mix. Swap roles around so everyone gets a chance to be the team leader, designer, etc.
- Don't be afraid to separate students from the technology for as much as a period or a little more. With younger children (or just those lacking self-control), following and paying attention to the process of the Design Cycle sometimes requires a little more explicit intervention, particularly when it comes to planning, designing and evaluating. This relatively short time away from the virtual environment can lead to much more focused and productive work once they are back inworld. Just remember to only provide the coaching necessary for them to get it done by themselves and not to separate them for too long or too frequently – these are still lessons on learning via technology and that should remain the focus.
- Get and stay out of the way, because it's not about you. OpenSimulator classes are by definition learner-centric. Encourage students to be as independent as possible. This is where not just OpenSimulator is perfect, but also the Design Cycle, particularly the Investigate step, comes in. After showing students how to use help systems, do searches, etc., try not to answer a question unless a student explains what s/he has done to try to solve a problem. Obviously, this applies more to older students, the younger ones will need more help. The books listed in the "Resources" section will confirm that this

approach will disorient kids at first, particularly in Asia where students have become accustomed to having a passive role, but ultimately they will adapt to (and love!) being an active participant in their learning experience.

- Give students a chance to start off with a new .oar when they are beginning a new project. Unless you plan on making use of their previous constructions, clearing out the space for them both gets rid of clutter that would be in their way and also allows them to use their new skills to build a more interesting environment that matches their new talents and abilities. A blank .oar to an OpenSimulator student is just like a blank canvas to an art student.
- It doesn't hurt to give students some inspirational examples to get their creative juices flowing and give them a target to aspire to. Showing students what professionals have built in places like ReactionGrid or (even better!) taking students in for a tour of what another class has built can lead to the formation of all kinds of new ideas. Don't think that students can only learn from those above them either. Some of the most engaging inspiration witnessed this year at CAIS was what our MYP 1 students got out of the relative chaos and mayhem of a PYP 2 .oar.
- Keep all assignments open-ended, and ensure that performance is assessed according to individual abilities, contributions, etc. OpenSimulator assignments enable the ultimate differentiated classroom. Even special needs students can succeed on their own terms. And gifted students are free to reach whatever goals they wish. Remember, at CAIS we have native speakers of English and kids who are absolute English beginners in the same Technology classes! As you read the lesson plans, note that a general direction is given, but micromanagement is counterproductive.

- Start with simple building assignments, then progress to increasingly complex ones, but always try for a cohesive theme. For example, you could ask students to build a small town, with some working on the streets, others on the buildings, etc. Don't be too disappointed, though, if your students aren't immediately able/willing to work with all the others on a particular assignment. It's best not to force it, at least at first. Let each team work together. Eventually they'll progress to class-wide collaboration, cooperation, etc. As we found with an "interdisciplinary unit," a fancy term for combining work across different subjects, it was when a class had one common art project to complete that they were able to really cooperate as a team adapting (improving!) their masterpiece in OpenSimulator.
- Take advantage of opportunities to connect what students are building in virtual environments with the real world. You certainly don't want to force students to merely copy the real world in virtual space (where is the fun in that?), but encouraging them to observe or investigate real world scenarios helps add another rich layer of creativity to their virtual spaces. One example from this year has been MYP girls carefully watching each other perform a dance in real life so that they could learn to accurately model it while creating inworld animations. Another has been students investigating historical, cultural and environmental conditions in working on a whole-class project to design a "Mesa Verde meets New Mexico" representation of an ancient adobe structure. These connections provide great opportunities for extending learning in holistic ways.

That's not all the Best Practices CAIS has noted so far, but that's all you're going to get in this e-book. You can see that YHA has several more tomes to write! Now for some actual lesson plans. The first one will feature an introduction, then the last two will just be presented.

Lesson Plan Example 1

Three actual CAIS lesson plans follow. The first explanatory note you need is that in CAIS Technology classes, YHA is known as Beloved Leader. Co-teacher and Honorary Geek Alex Makosz is called Grand Poobah. You want to know why? You can't handle the truth! Please accept YHA's humble apology that the lesson plan formatting didn't always survive the transition to this e-book. This e-book was finished in a hurry to correspond with Beloved Leader and Grand Poobah's presentation at the 2011 AACE'S ED-MEDIA Conference, so at a certain point YHA just stopped tweaking styles. And please note that in many cases, these plans have been "tightened up" in the interest of saving space. Students normally have a lot more room to write! OK, enough excuses, let's get on with it.

Note the skill/knowledge objectives are separated from the language/knowledge objectives. This is because the lesson plans were developed following the guidelines of Sheltered Instruction Observation Protocol (SIOP), a patented process that is: for teaching content to students who aren't fluent in English; way beyond the scope of this e-book. Other aspects of the plans, such as listing vocabulary words up front, are also in accordance with SIOP suggestions.

Note that there are individual tasks and team tasks, as well as individual and team deliverables. Some things each student is responsible for doing, other things the team is responsible for accomplishing. Note that there are also at least two "strands" per project, e.g., Scratch AND OpenSimulator. In a later lesson plan, students create machinima ("machine cinema") using Audacity, CamStudio and VideoStudio, so there are three strands in addition to the OpenSimulator component. You see how OpenSimulator ties everything together? Your students won't just be recording a sound file, e.g., they'll be producing one for a 3D cartoon or a

podcast to be streamed to their virtual world. Nothing is an end in itself, everything has a further purpose. Almost Zenlike, huh?

Each lesson plan has a breakdown of the schedule cycle by cycle. At first you will have to ensure that students are keeping track of progress and time, but very soon they'll be able to do this themselves. Checklists are provided for all the deliverables. There's never any doubt as to what is due and when. Each lesson plan is essentially a project plan in that individuals/teams must divide the work among themselves cycle by cycle.

Please note, and this is one of the most important aspects, that an assignment might be something like "Write a Scratch program," but there is never any stipulation as to what the program will do, or anything else. This is up to the student or group of students. There are always choices to be made, i.e., the project establishes certain parameters and then learners are able (encouraged!) to interpret, adapt, etc. Planning is always emphasized, as is designing. It was attempted to make students use a graphics program to create their designs, but this became a lesson plan all its own and wasn't worth the distractions. Students typically sketch out their designs with low-tech pencil and paper.

Some of you cynics are thinking: "Wait a minute! If some of your students aren't fluent in English, how can they possibly follow such plans?" Well, we maintain different standards of accomplishment, that's all. We have many bilingual students who help explain things, and not everyone is expected to write polished responses to the questions. Remember, the plans are guidelines for what's happening, just frameworks. "Build something using OpenSimulator" is an example of what a lesson plan will say. Actually building whatever that something turns out to be is the primary goal of the student, as well as the lesson itself! It's the technical skills and knowledge attainment that really matters, not the paperwork!

MYP 4 IT LESSON PLAN: CYCLES 10-12 (Nov. 24-Dec. 17)

Quarter 2 Project Part I: Scratch/OpenSimulator

Objectives

Skill/Knowledge Objectives	• As an individual, learn the basics of Scratch programming. Continue improving your OpenSimulator skills, including using/writing/editing scripts. • Then, again as an individual, plan how after the Christmas break your team will create a Scratch program and an OpenSim design together. • Complete and upload this deliverable, as an individual, by the deadline. • Demonstrate that you can handle the Design Cycle: Investigate, Plan, Design, Create, Evaluate…and Attitudes!
Language/Knowledge Objectives	• Follow verbal/written instructions in English. • Use the e-book textbook and online help to use software. • Learn and use English vocabulary words.

Vocabulary (Computer-specific! Record additional words at the end of this document.)

Code	
Control	
Coordinates	
Costume	
Dynamic	
Execution	
Interpreted	
Language	
Mode	
Operator	
Script	
Sprite	
Stage	

Schedule

Cycle 10	• As an individual, go through the Scratch exercises with your Beloved Leader's or Grand Poobah's guidance. • Work on your Scratch and OpenSimulator checklists.
Cycle 11	• Continue working on your Scratch and OpenSimulator checklists. Start on your original Scratch program and individual OpenSimulator construction. • Choose a team for your Scratch/OpenSim projects after Christmas.
Cycle 12	• Finish your Scratch and OpenSimulator checklists. Talk with your future team. • Complete this document and submit it (as an individual) before/by the deadline.

Deliverables: Each INDIVIDUAL must complete this document and upload it to Basecamp

NLT 0800 December 17th. Print, complete and submit one design page per person.

Scratch Checklist

Note: When a task/skill says "evidence," your Beloved Leader or Grand Poobah will need to say "Yes."

Task/Skill	Done/Demo
Investigation: At least half independence (evidence)	
Project Plan: Created, followed, ideally evaluated (this document)	
Design: Created, followed, ideally evaluated (this document)	
Evaluation: Evidence, explanation of process	
Mr. Wiggly Program (or other Scratch book example) done	
Original Scratch program done	
Identify strengths and weaknesses: Creating sprites, backgrounds or designing programs, etc.	
Help others according to S & W	
EXTRA CREDIT: Upload program to Scratch site	

What will your original Scratch program do?

Do you have enough time? Or do you need to change your design/plan?

How will you divide the work re: time?

Cycle 10

Cycle 11

Cycle 12

MYP ___ Name _____

Scratch

Sketch two designs for your Scratch stage(s)/program(s). Choose one and be prepared to explain why it's better.

OpenSimulator

Sketch two designs for Caisland 7/ReactionGrid. Choose one and be prepared to explain why it's better.

OpenSimulator Checklist

Note: When a task/skill says "evidence," your Beloved Leader or Grand Poobah will need to say "Yes."

Task/Skill	Done/Demo
Investigation: At least half independence (evidence)	
Project Plan: Created, followed, ideally evaluated (this document)	
Design: Created, followed, ideally evaluated (this document)	
Evaluation: Evidence, explanation of process	
OpenSim construction done	
Identify strengths and weaknesses: Creating textures, building techniques, etc.	
Help others according to S & W	
EXTRA CREDIT: Upload to Florida Caisland	

What will your OpenSim construction look like (and what will it do)?

Do you have enough time? Or do you need to change your design/plan?

How will you divide the work re: time?

Cycle 10

Cycle 11

Cycle 12

MYP 4 IT LESSON PLAN: CYCLES 14-16 (Jan. 18-Feb. 17)

Quarter 2 Project Part II: Scratch/OpenSimulator

Objectives

Skill/Knowledge Objectives	• FIRST: Ensure your Cycle 13 document is completed and uploaded. If you want to change partners/teammates, speak up as of Cycle 14. • Pair up into teams of 2. We'll have teams of 3 only if necessary. You may work by yourself as long as you understand this will mean extra work! • Complete and upload a Scratch program. Best: Use S4SL to change Scratch code to Linden Scripting Language and upload the code. More on this later. Complete this document. Upload the Word part, hand in the hard copy. • Demonstrate that you can handle the Design Cycle: Investigate, Plan, Design, Create, Evaluate…and Attitudes!
Language/Knowledge Objectives	• Follow verbal/written instructions in English. • Use the e-book textbook and online help to use software. • Learn and use English vocabulary words.

Vocabulary (Computer-specific! Record additional words at the end of this document.)

Event	
Float	
Function	
Integer	
Key	
Level	
List	
Local	
Parameters	
Rotation	
Scope	
State	
String	
Value	

Variable	
Vector	

Schedule

Cycle 14	• Start on the Scratch/S4SL program you've designed or start again as of Cycle 14. PLAN! Figure out what your code and/or objects will do.
Cycle 15	• Continue working on your Scratch/S4SL program. Work with objects as desired/required.
Cycle 16	• Finish your Scratch checklist. • Upload this document and your code (if S4SL), hand in your designs.

Deliverables: Each TEAM must complete this document and upload it to Basecamp NLT 0800

February 17th. Print, complete and submit one design page per team.

Scratch/S4SL Checklist

Note: When a task/skill says "evidence," your Beloved Leader or Grand Poobah will need to say "Yes."

Task/Skill	**Done/Demo**
Investigation: At least half independence (evidence)	
Project Plan: Created (this document)	
Design: Created (this document)	
Evaluation: Some evidence of process	
Original Scratch/S4SL Program & Objects/Primitives	
EXTRA CREDIT: Helping others	

Will you work on a Scratch program exclusively or will you use S4SL and OpenSim?

What will your original Scratch/S4SL program do? (What will it make prims do?)

What are the sprites or objects, sounds, etc., involved? Will you create or get them?

How will you divide the work re: people and/or time?

Cycle 14

Cycle 15

Cycle 16

MYP ___ Name(s)_____

Scratch (only if not doing S4SL)

Sketch two designs for your Scratch stage(s)/program(s). Choose one and be prepared to explain why it's better.

OpenSimulator (only if not doing Scratch)

Sketch two designs for the objects involved with your S4SL program, and provide a general idea of what your code will make them do. Choose one and be prepared to explain why it's better.

Reflection

Team

How did you and your teammate(s) work together on this project? What would have improved your performance? What will you do differently for the next project?

Did you and your teammate(s) follow the Design Cycle? If not, why not?

Individual

1. How much do you think your Design Cycle knowledge/skills have improved over the past semester? How about your ability to perform as a team member?

2 How do you think you can improve your Design Cycle knowledge/skills next semester? How about your ability to perform as a team member?

Lesson Plan Example 2

MYP 4 IT LESSON PLAN: CYCLES 18-20 (February 24-March 21)

Quarter 3 Project Part I:

OpenSimulator/Audacity/CamStudio/VideoStudio

Objectives

Skill/Knowledge Objectives	• This quarter we'll be expanding our skills/knowledge re: OpenSimulator building and programming, as well as adding something new: MACHINIMA! • We're also going to be (finally) visiting other schools and helping other students worldwide via ReactionGrid and the IB Virtual Community. • Demonstrate that you can handle the Design Cycle: Investigate, Design, Plan, Create, Evaluate…and Attitudes!
Language/Knowledge Objectives	• Follow verbal/written instructions in English. • Use the e-book textbooks and online help to use software. • Learn and use English vocabulary words.

Vocabulary (Computer-specific, please! Don't define, e.g., "string" as a small rope!)

Cast/actors	
Dolly	I don't mean Barbie!!
Flow	
Framing	
Pan	
Point of view	
Scripting	
Set	
Shoot	
Storyboard	
Zoom	
Programming	See previous lesson plans for programming vocabulary terms.

Schedule

Cycle 18	• Divide into teams of 2 or 3. Get together with EVERYBODY in the class and try to decide on a THEME for your entire region (256 square meters). Remember, this is your SET for your machinima (movie)! Otherwise, divide your ReactionGrid sim (Sim3) into parts as needed. • Before you turn on a computer, you must complete a sketch of what you will build, along with a project plan for how you will build, program, shoot and edit. Keep your machinima project reasonable… 4 minutes or less…using whatever you can make or scavenge from ReactionGrid. • Go to the IBVC and look at the different groups. Work with MYP 5 to create an MYP 5 Virtual Worlds Group.
Cycle 19	• Start building your set. Decide what's going to move or otherwise be programmed. Revise your storyboard if needed. Gather stuff from ReactionGrid. Experiment with CamStudio and camera controls.
Cycle 20	• Finish your checklist. • Complete this document and submit it before/by the deadline.

Deliverables (Deadline is March 22):

Each TEAM must complete this document and upload it to Basecamp.

Each TEAM must complete a design sheet and give it to Grand Poobah/Beloved Leader.

Checklist

Note: When a task/skill says "evidence," your Beloved Leader or Grand Poobah will need to say "Yes."

Task/Skill	Done/Demo
Investigation: At least half independence (evidence)	
Experiment: Hypergridding	
Experiment: CamStudio/Camera Controls	
Project Plan: Created (this document)	
Design: Created (this document)	

Storyboard Done	
Script Done	
Set Started	
Plan to write/record music	
EXTRA CREDIT: Help others	

How will you divide the work re: time?

Put these steps in the project plan (next page), in the order you will follow:

Consider writing/recording different background music.

Create your project plan.

Decide on a class theme or divide land into team plots.

Decide on what prims will be programmed and what they'll do.

Design your set on paper.

Draw your storyboard.

Ensure ReactionGrid and Florida Caisland avatars tested.

Experiment with CamStudio and OpenSim camera controls.

Test hypergridding: ReactionGrid <-> Florida Caisland.

Write your script.

Anything else?

Project Plan

Cycle 18

Cycle 19

Cycle 20

How will you divide the work re: people? Who will do what?

Set Design

MYP ___ Team Members _____

Storyboard

MYP ___ Team Members _____

MYP 4 IT LESSON PLAN: CYCLES 21-24 (March 22-April 21)

Quarter 3 Project Part II:

OpenSimulator/Audacity/CamStudio/VideoStudio

Objectives

Skill/Knowledge Objectives	• This quarter we'll continue expanding our skills/knowledge re: OpenSimulator building and programming, as well as adding something new: MACHINIMA! • We're also going to be (finally) visiting other schools and helping other students worldwide via ReactionGrid and the IB Virtual Community. • Demonstrate that you can handle the Design Cycle: Investigate, Design, Plan, Create, Evaluate…and Attitudes!
Language/Knowledge Objectives	• Follow verbal/written instructions in English. • Use the e-book textbooks and online help to use software. • Learn and use English vocabulary words.

Vocabulary (Add words you don't understand and we'll discuss them in class.)

Word	Definition

Schedule

Cycle 21	• Finish whatever you need to from Cycles 18-20. Finish your sets and programming. Shoot your video. Record your dialog and/or music using Audacity. Edit your video using VideoStudio.

	• Keep your machinima project reasonable…up to 4 minutes …using whatever you can make or scavenge from ReactionGrid. • Sketch your final set design. Explain on your design sheet why you made changes. Complete your project plan and do the reflection exercise. • Do some marketing. Find other students on the IBVC and invite them to join your group. What else can you do? Discuss with GB or BL. Extra: Ask other schools. • Visit ReactionGrid and contact owners of other grids. Make arrangements to visit at least one other school's grid. Visit at least one other school.
Cycle 22	• Continue with your checklist tasks.
Cycle 23	• Finish your checklist. • Complete this document and submit it before/by the deadline.

Deliverables (Deadline is April 21):

Each TEAM must complete this document and upload it to Basecamp.

Each TEAM must complete a design sheet and give it to Grand Poobah/Beloved Leader.

Checklist

Note: When a task/skill says "evidence," your Beloved Leader or Grand Poobah will need to say "Yes."

Task/Skill	Done/Demo
Investigation: At least half independence (evidence)	
Project Plan: Created (this document)	
Design: Created (this document)	
Complete set and related programming.	
Shoot and edit video. Complete your video.	
Visit at least one other ReactionGrid grid	
Write/record or find music and use it in production	

EXTRA CREDIT: Help others	
EXTRA CREDIT: Make a gesture with Avimator	

How will you divide the work re: time?

This time, use your own steps. Copy the ones in the previous lesson plans if you must, but try doing it yourselves first.

Project Plan

Cycle 21

Cycle 22

Cycle 23

How will you divide the work re: people? Who will do what?

(**E-book Note:** We won't repeat the Set Design and Storyboard sheets here. In the computer lab, many of these sheets are provided so that students feel free to do more than one. It's almost inevitable that they'll go through at least three designs and storyboards before settling on their final versions.)

Lesson Plan Example 3

MYP 4 IT LESSON PLAN: CYCLES 25-30 (May 4-June 28)

Quarter 4 Project: Art/Music/IT!

Objectives

Skill/Knowledge Objectives	• During this unit, you will combine Art and IT…and ideally Music and IT too. You will create art in Art class, then recreate or display it via OpenSimulator! If time permits, you will create music in Music class, then perform/stream it via OpenSimulator! • Demonstrate that you can handle the Design Cycle: Investigate, Design, Plan, Create, Evaluate…and Attitudes…in more than one class simultaneously. This will be your first true InterDisciplinary Unit (IDU).
Language/Knowledge Objectives	• Follow verbal/written instructions in English. • Use the e-book textbooks and online help to use software. • Learn and use English vocabulary words.

Schedule

Cycle 24	• Finish your machinima from the last unit if you haven't already done so.
Cycle 25	• Discuss with your Art/Music teacher what you might do. Discuss with your classmates how you would divide up Florida Caisland and ReactionGrid Caisland for your art/music shows. Draw a diagram.
Cycle 26	• Start creating your art in Art class. Make a decision: Will you try to recreate it or just display it? If you're creating sculptures, you should try to recreate it. Draw at least one design diagram with your Art Teacher. If you're doing Music, talk with your Music and IT Teachers.
Cycle 27	• Create your project plan. You should be able to create a basic project plan based on your experiences this year. Continue creating your art/music! Start recreating in Caisland if applicable. Start preparing Caisland for your shows. If the Internet is cooperating, we'll work in Florida Caisland. If not, we'll use Caisland 7 on Big Bertha.
Cycle 28	• Continue creating your art/music. Continue recreating in OpenSimulator if applicable. Continue preparing Caisland for your shows.
Cycle 29	• Finalize the creation of your art/music and OpenSimulator recreations, as well as the preparation of OpenSimulator for your shows. Start your reflection and portfolio gathering.

Cycle 30	• Help other MYP classes advertise your show to friends, family and people worldwide! Help with the creation of an IBVC group. Be a tour guide!

Deliverables (Deadline is June 24):

Each TEAM must complete this document and upload it to Basecamp.

You are putting on an art and/or music show via OpenSimulator: Florida Caisland and ReactionGrid Caisland. In Florida Caisland, your class has an entire region to design into an art gallery, concert stage, etc. The idea is that you invite visitors to ReactionGrid Caisland for a "teaser" and then they'll hypergrid to Florida Caisland for the bigger venue. Or we load the Florida .oar to ReactionGrid. You want a COHESIVE THEME for at least your class' area. The art and music are up to you! ;)

Checklist

Note: When a task/skill says "evidence," your Beloved Leader or Grand Poobah will need to say "Yes."

Task/Skill	Done/Demo
Investigation: Almost complete independence (evidence)	
Project Plan: Created (this document) and mostly followed	
Turf Designs: Created (this document) and mostly followed	
Creation Designs: Created (Art Teacher) and mostly followed	
Music Designs: Created (Music Teacher) and mostly followed	
Creations: Reproduced and/or displayed/performed	
Advertising: IBVC or otherwise	
EXTRA CREDIT: Help others	

Project Plan

(For each task, indicate WHO will do it. If all of you will, say ALL.)

Cycle 25

Cycle 26

Cycle 27

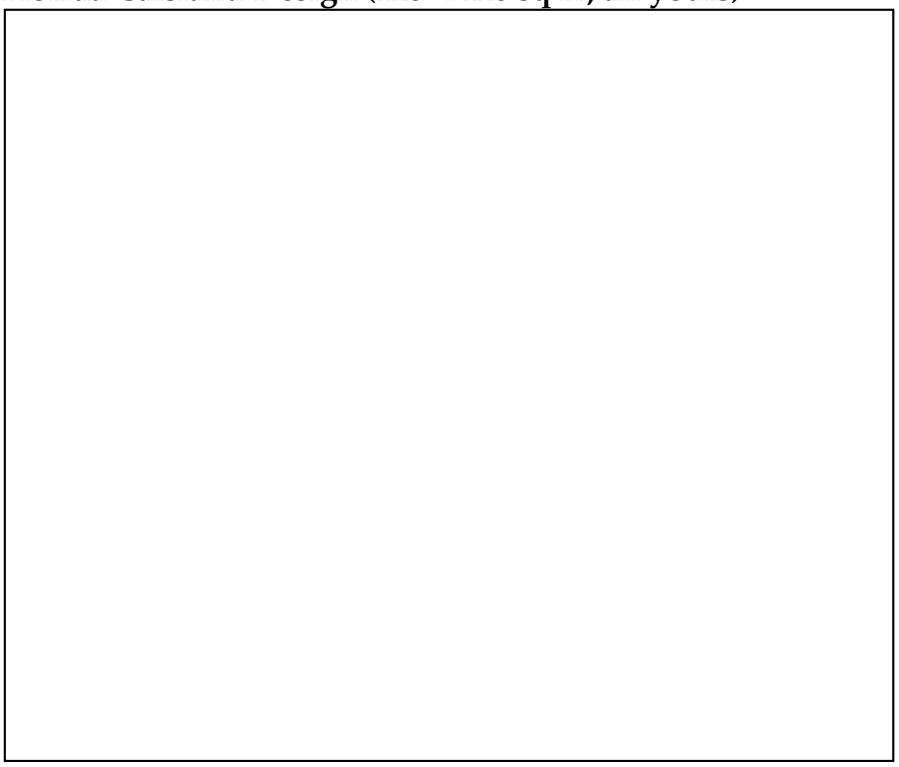
Cycle 28

Cycle 29

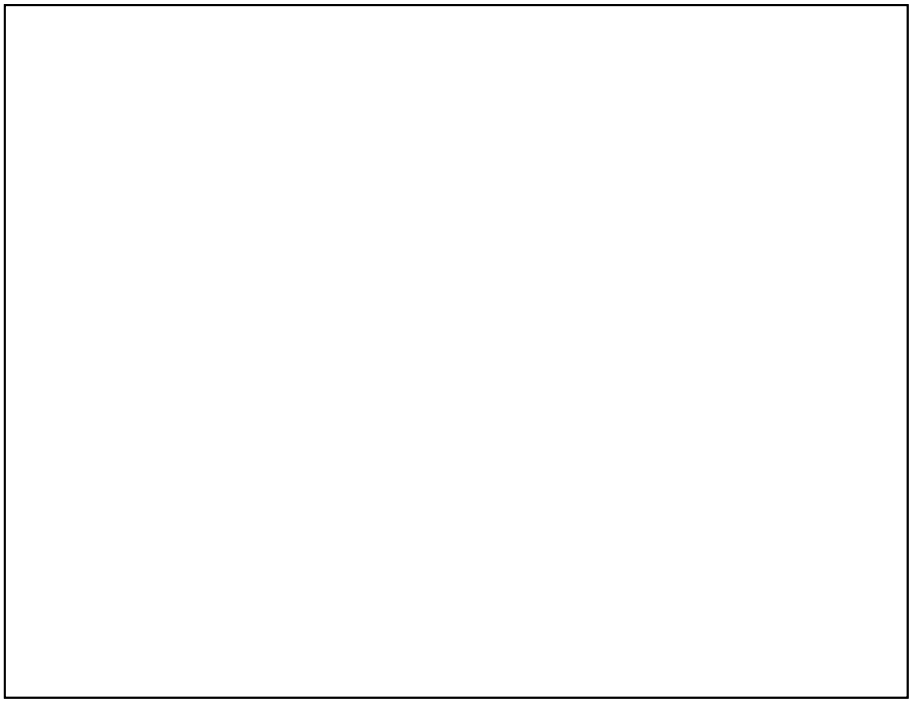
Cycle 30

MYP 4 Team Members _____

Florida Caisland Design (128 X 128 sq m, all yours)



ReactionGrid Caisland Design (256 X 256 sq m, shared)



Reflection

How has your team improved re: the Design Cycle over the past year?

Does your team believe you deserve the highest grade (6) re: any particular part of the Design Cycle? Why or why not?

How do you think you can improve re: the different steps?

Investigate

Design

Plan

Create

Evaluate

Attitudes

Exercises

So what do you think? Are you ready to put together an OpenSimulator lesson plan of your own? Sure you are. You have a lesson plan (or whatever you call 'em) format that your department or school uses. Don't reinvent the wheel. First, try fitting the examples you've just read into your format. Then come up with your own ideas. You're going to LOVE teaching OpenSimulator in your school, as part of your IT/ICT/Technology classes or indeed any classes! It'll require an adjustment period, not just for you, but for your fellow teachers and especially your administrators as well. Relax. Take deep breaths. Everything's going to be OK.

Getting a (Second) Life

Imagine tomorrow the IBO (or other international or K-12 schools') OpenSimulator Grid makes its debut (we're going to make it, be sure to read "No Subtlety Whatsoever" at the end of this e-book). We will have attained Nerdvana, right? Well, yes and no. Until the OpenSimulator grid achieves critical mass and even afterward, there's going to be a need to connect, communicate and collaborate with all the 3D virtual pioneers who are going to visit our grid either infrequently or, well, never. This means university/college teachers/students and all the other interesting people (avatars!) in Second Life. CAIS, e.g., is in a very isolated location and many international schools are in a similar situation. YHA regularly attends inworld meetings held by professional teacher organizations, get-togethers impossible without Second Life.

So, although the OpenSimulator grid is the most critical thing for K-12 students/teachers, the second most important is the establishment of an IBO/K-12 Second Life island, at least for teachers if not for students too. Or two, or three, or a hundred K-12 SL islands. If, for whatever reasons, the OpenSimulator grid is not immediately feasible, then we pioneers should quickly proceed with starting the first K-12 SL Island. Eight schools could go together and finance one-eighth of a region each for around US$600 year. The cybercampuses could be put up in days. We're talking one-week-from-commitment turnaround! What are we waiting for??

But whether or not, e.g., the IBO or another K-12 organization has one or one hundred Second Life islands anytime soon, Second Life is going to quickly become a vital part of your school's classes, at least for students 16 and older. You can easily have some SL cyberturf of your own. A brief introduction to the Information Technology in a Global Society (ITGS) course is coming up

in a few pages. And this class is just one example of how SL can be used to teach high school students. Remember, CAIS is using SL for middle school kids too!

To use Second Life, you need an account/avatar (if you're surprised, then slap yourself, but not too hard). The basic account is still free. You also need a viewer (hopefully no surprise!), and you should get a special one for Second Life later. Right now you can stick with Hippo or whichever one you've been using for OpenSimulator.

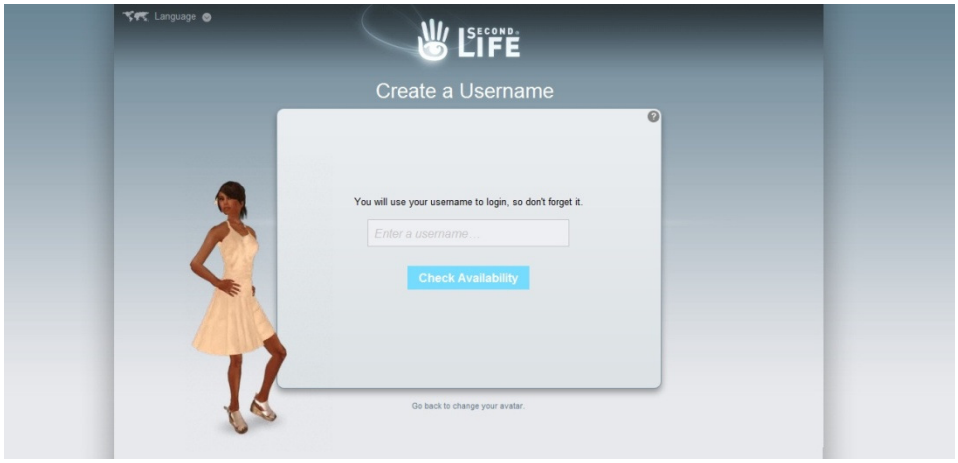
Creating Your Account/Avatar

Go to **http://secondlife.com** and click the **Join Now** button, or go directly to **https://join.secondlife.com/?lang=en-US**.

Either way, the **Choose an Avatar** window displays.

This isn't OpenSimulator, so you don't have to start off as Ruth! The avatars will rotate, or you can click the **Left Arrow** or **Right Arrow** button. When the avatar you like is up front, click the **Choose This Avatar** button. Remember, once you're inworld you can change the way you look. You don't even have to be human if you prefer being an animal or something else!

The **Create a Username** window displays.



Pause and give this some thought, because you can't change your username. Sometime around the start of 2011, SecondLife stopped allowing new users to choose both first and last names. Now your last name is going to be "Resident." You choose just your first name. Don't be disappointed if you type in a name, click the **Check Availability** button and discover your choice is already taken. Just try again. Your Humble Author's students typically add the current year, e.g., if "MinCho" is unavailable, use "MinCho2011" instead. "YourHumbleAuthorRocks" was available, so Your Humble Author took it! Sorry, you can't have it. Yes, you're jealous, and rightly so. When you see **It's available!**, click the **Next Step** button. The **Just a few more questions…** window displays. Type your e-mail address and select/type the other information. Then click the **Create Account** button.

Now, why did Your Humble Author not insist that you use the same username that you used for your OpenSimulator account(s)? Well, mainly because it's impossible, because your last name is already designated. But you could still make it easy to remember, e.g., "Min Cho" in OpenSimulator could try for "MinCho Resident" in Second Life. You have to admit that "YourHumbleAuthorRocks" is not only true, but it's a geeky-cool name too.

The **Select an Account** window displays. Be sure to click the **Select** button under **Free Account**! If you want to have a cybercampus for your school, someone will need to have a premium account. As the teacher, that's most likely going to be you. But you can always upgrade later. You're the only one who needs the premium account. All your students and even coworkers can stick with the free one.

The **Welcome to Second Life** window displays.

But DON'T click the **Download and Install Second Life** button right now. Remember, you can use your Hippo (or just about any OpenSimulator) viewer to connect.

Later, after you finish reading this e-book, i.e., you should choose the Second Life viewer that's right for you. Try the latest Version 2+ viewer first. When this sentence was written, this was Version 2.6.9. It'll probably have changed ten times by the time you read this. Maybe it'll be up to Version 3! Anyway, go to **http://secondlife.com/support/downloads/?lang=en-US** to get the latest viewer.

Now, if the latest viewer doesn't work, don't despair. That's right: Relax, take deep breaths, etc. The new SL viewers are pretty demanding hardware-wise, so don't be too shocked if your old

or lower-end PCs can't run it. You have an alternative, and that's the abandoned but reliable Viewer 1.23, the one all the OpenSimulator clones are based on. Scroll to the bottom of the page listed above so you can see **More Viewers**. Click the **Second Life Wiki** link.

The **Linden Lab Official: Alternate Viewers** page displays. Scroll to the bottom and you see **Viewer 1.23**. Click the **Windows**, **Macintosh** or **Linux** button. But don't do this now!

Still here? Good. You can deal with the Second Life viewers later. Or not. You can always stick with, e.g., Hippo. The new 2+ viewers have some impressive functionality, but if your clients can't handle them, it doesn't really matter how wonderful they are. **Another tip:** Try to ensure that everyone in your class is using the same viewer at any given time. This is especially true for inworld classes. If you're using, e.g., 1.23 and your students have 2.63, it's going to be a bit difficult for you to help them with their problems. OK, enough. Let's go inworld.

Start your Hippo viewer. Click the **Quick Grid Select Down Arrow** and click **secondlife**. It's hard to imagine that Second Life would not be on the list, but just in case it isn't, click the **Grids** button, then the **Add** button. In the **Login URI:** text box, type:

https://login.agni.lindenlab.com/cgi-bin/login.cgi

In the **Grid Nickname** box, type **Second Life**.

OK, whether or not **secondlife** was in the list, now you're ready to go inworld. Type your first name, last name and password. Then click the **Log In** button.

The **Terms of Service** dialog box displays. If you have lots of time on your hands, read it. Otherwise, let's keep going. Click the **I Agree to the Terms of Service** check box and then click the **Continue** button. A progress bar indicates…progress, that's right…and you're inworld.

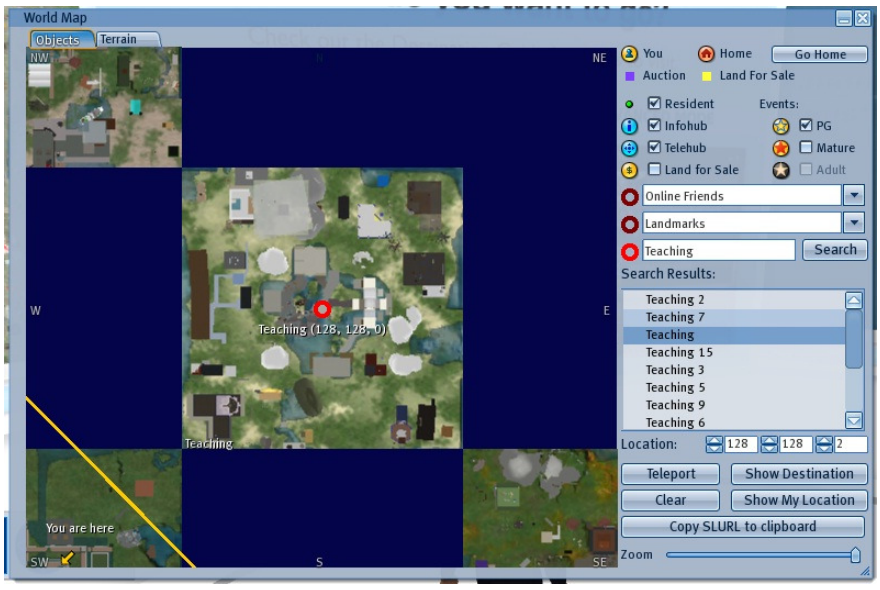
You're on **Welcome Island**. One of them, anyway. There are a lot.

Chances are another newbie ("noob," that's you, remember?) or two will have arrived at the same time, so take a few minutes to say "Hi." Now, if you did the exercises in the first chapter, you already have a good idea of what to do. But Your Humble Author suggests that if being in 3D virtual worlds is relatively new to you, don't rush through the Second Life tutorial. You can learn plenty. Many of YHA's students dash through the orientation and then weeks later have to ask how to do simple things like sitting down! But if there is a hurry, you can have a veteran avatar give you a landmark, which will provide you with a shortcut.

Let's assume you make it to the **Where do you want to go?** board. You're going to visit Your Humble Author's cybercampus, China International Schools Inworld (CISI) on the Teaching sim! CAIS, Beijing International School and Western Academy of Beijing currently comprise the CISI Group. More to come soon!

The instructions on the board are for the 2+ viewers, so ignore them.

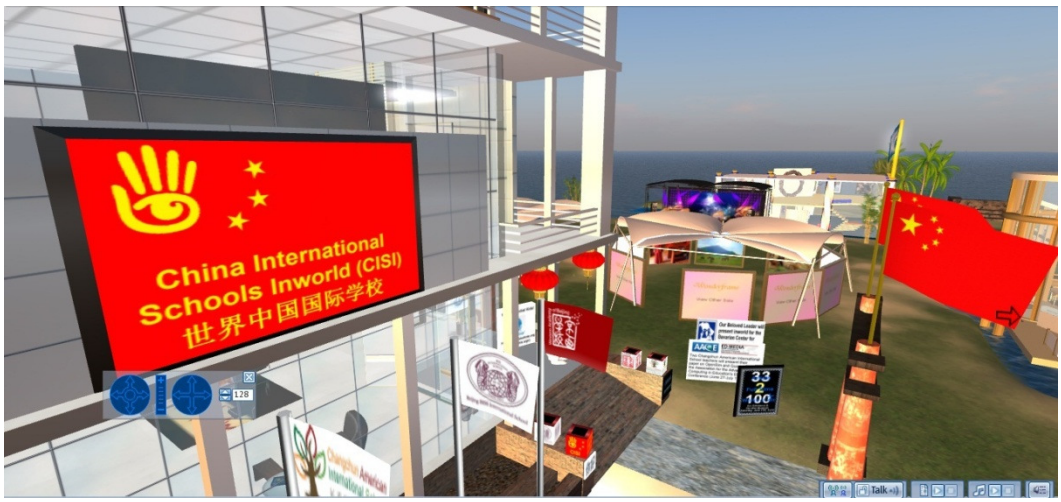
Again, if you did the exercises in the first chapter, you already know what to do. If you don't, shame on you. But it's OK, get over it, let's keep going. Click the **Map** button at the bottom of your viewer. The **World Map** dialog box displays. In the **Search** text box, type **Teaching** and click the **Search** button.



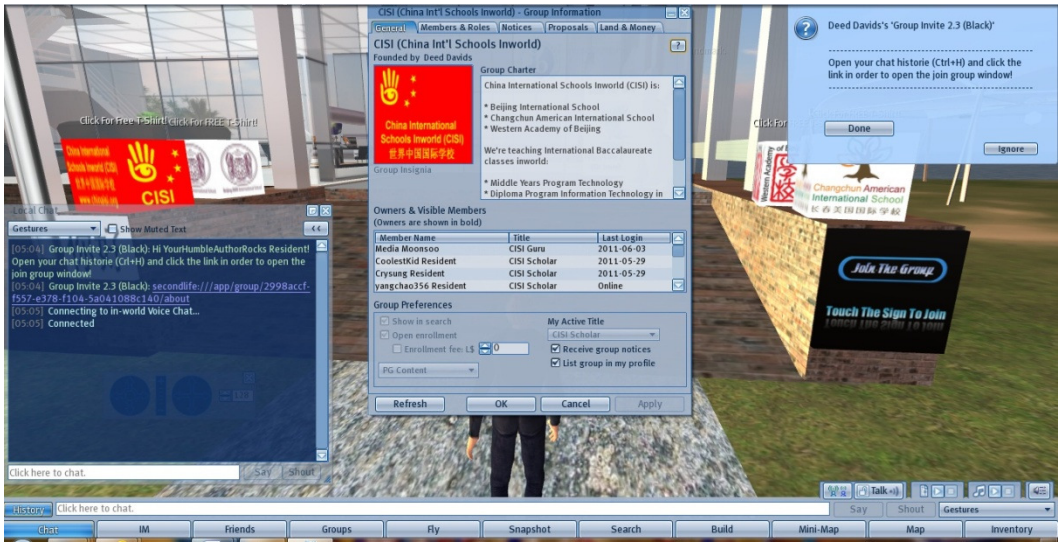
There's your destination! Click the **Teleport** button. You might see a message such as "this region doesn't allow such-and-such teleports." Just click the **OK** button.

You'll be in the center of the sim unless you clicked on a certain location. Spin around until you see the **Need some help in SL?** sign. If there is no such sign, don't panic. The idea is to walk north. Click the **Mini-Map** button at the bottom of your viewer and move toward **N(orth)**.

Who knows what will be happening on the cybercampus when you arrive. When this e-book was being finished, we were preparing for an international art/music show (Lesson Plan #3).



Go to the "front steps." Right-click the **Join the Group** sign. Then click **Touch** in the pie menu.



A message appears in the upper righthand corner. Click the **Done** button. The link to the **Group Information** dialog box is in your chat history. Click the **History** button above the **Chat** button in the lower lefthand corner. The **Chat History** window displays, and you see the group link. See it? It says **Group Invite**. Click that link and the **Group Information** dialog box displays. Click the **Join (L$0)** button. And then the **Join** button. Click the **OK** buttons and otherwise close the windows and boxes. You're now a member of China International Schools Inworld Group and so we can keep you informed of what we're doing in Second Life.

Getting Your SL Cyberturf: Lots of Choices

Yes, just like with OpenSimulator but even more so, you have an abundance of choices for a home of your own in Second Life. But, as a teacher representing a school, you really only have a handful of good SL possibilities. There's no "Do It Yourself" option with SL. You will be paying some organization, nonprofit or otherwise, for your cyberturf. Once you're an SL resident, and

especially after you start joining groups, hardly a week will go by without getting some offer for land at a discount. But when choosing the location for your school's cybercampus, you should consider several factors other than just price. The first question is: Should you buy or should you rent/lease? You don't really "buy" anything since you pay by the year regardless. What this really means is that you either have, e.g., an entire island or you occupy a lot or parcel on that island. Even if you have an island of your own, and ultimately CAIS would like to see the International Baccalaureate or other collection of international schools have their own region, you need to make the choice of going it alone or having some assistance from a "landlord" or "landlady" ("landperson"?). You also have to think about what type of organization you're renting/leasing from and who your neighbors are (and are going to be!). Some business owner might give you a great deal, but the type of customers wandering around might not exactly be conducive to maintaining a productive educational environment. Plus you never know who's likely to rent the lot next to yours. Imagine explaining to parents why your cyberneighbors include a bar, nightclub or worse. No, you want to rent/lease from an educationally-oriented individual or organization. You want other teachers, students and schools around you. That's right, we're talking about the "PG" or kid-friendly rating of the place. Ideally, you'd also have mutual or other facilities provided free of charge. And, of course, you have to think about customer service. When you have a problem, will you have someone to contact and will that person respond promptly/effectively?

Believe it or not, Your Humble Author does intend to get to the point. CAIS leases its cyberturf from the New Media Consortium (**http://www.nmc.org**) and Your Humble Author has been doing so since 2006. The New Media Consortium's (NMC) offerings will be discussed, along with three other alternative vendors. As before, these few organizations have been chosen from

among the many because either Your Humble Author recommends them, other people do, or both. As always, you must keep in mind that just because a price is quoted here, it is not necessarily going to be correct tomorrow (or even today!). SL prices are subject to change, perhaps suddenly and/or drastically, because Linden Labs' discounts for educational organizations might not last much longer, if they're not already gone completely. Anyway, here we go with four vendors or hosts:

- **NEW MEDIA CONSORTIUM.** You just visited China International Schools Inworld's cybercampus. It's on the Teaching sim. We occupy two 4,096-square-meter lots, for a total of 8,192 square meters. Since 2006, the NMC has held the cost to approximately US$100 per 1,000 square meters. Five years without a price hike! So yes, go ahead and get out your calculators. Good thing you're computer and not math teachers. That's US$800 a year that we currently pay. You could get an entire sim (256 square meters, remember?) for just US$5,000 a year. Again, as of June 2011. C'mon, international schools, let's chip in on an island of our own! Not so long ago SL wanted to increase rent prices for schools and the NMC resisted successfully. The NMC is a large, international nonprofit organization with a substantial SL presence, so it's understandable that it has some bargaining clout. Despite the NMC's size, however, Your Humble Author has always enjoyed personal, friendly, quick customer service whenever a question or problem arose. When you're an NMC lessee (tenant), your neighbors are all schools, and this nurtures a sense of cybercommunity. Each NMC island has at its center the "Quad," with mutual facilities everyone can share. These often include amphitheaters, meeting rooms and other such structures. This is great because you don't have to recreate classrooms on your cybercampus. And the NMC features entire islands, such as the

NMC Orientation region, at your disposal. You'll be invited to seminars, conferences and shows. Services available include designing and building, up to entire islands! For more information, go to **http://virtualworlds.nmc.org/services/** or e-mail Nancy Reeves (**nancy@nmc.org**).

- **JOKAYDIA**. Does the name seem familiar? Hope so. Yes, this is the same Jokaydia that offers OpenSim cyberturf as well. As mentioned before, you could have both an OpenSim region and a Second Life lot with just one combined bill to pay. Your school's accountants would no doubt appreciate this. Jokaydia's SL lots are on islands reserved for schools and other educational organizations. Each island has mutual facilities and is designed to be a cybercommunity. You almost get the feeling that you're on a real life college campus when you're walking around. As of this e-book's publication, you can get 1,500 square meters (350 prims) for US$450 a year. Lots of different sizes are available. Shop at the website or ask for assistance. Customer service has been rated excellent by tenants. Jokaydia sponsors seminars, events, conferences (even an UNconference, you should check this out if you get the chance) and offers designing, building and other services. For more information, go to **http://jokaydia.com/rentals/index.php** or e-mail Jo Kay (**joannamkay@gmail.com**).
- **INTERNATIONAL SCHOOLS ISLANDS**. If you've never been to Chris Smith's "Shambles" website, **http://www.shambles.net**, pay it a visit soon. It's a veritable treasure trove of educational technology resources for schools, international or not. As Shamblesguru Voom in Second Life, Chris has for years maintained some of the most popular and active destinations for teachers and students, the International Schools Islands. Next time you're in SL, search for International Schools, pop by and take advantage of all the free stuff, as Your Humble Author regularly does. Again, at last

report, in addition to regular lot offerings, you could get an individual mini-island of 512 square meters (130 prims) for US$200 a year. You'd have buildings and other (some quite interesting!) structures at your disposal. Your neighbors would be other schools, and you'd be able to attend meetings, classes on building and programming, etc. Colleges and universities conduct exhibits regularly. Your landlord? An experienced SL teacher who regularly presents at seminars and other events. For more information, go to **http://www.shambles.net/secondlife/isi_renting/index.htm** or e-mail Chris Smith (**shambles.guru@gmail.com**).

- **TIMBUCKS**. Timbucks is an island that's been divided up into plots by two dedicated educators who want to give fellow professionals a home of their own at a discounted price. As of this e-book's publication, you can get 4,096 square meters for just US$300 a year. The region is kid-friendly and your neighbors would be other schools or educational organizations. The next time you're inworld, search for Timbucks and visit. For more information, contact Tammy DeCoste (**Knotawriter@gmail.com**) in Real Life. In Second Life, search for and contact Knota Writer.

ITGS Outline and PYP Intentions

ITGS Outline

As mentioned previously, Second Life has lowered its minimum age to 16, so now IB Diploma Program (high school) students can participate. The IBO has a computer course that's in the Humanities category: Information Technology in a Global Society (ITGS). CAIS plans to take its ITGS students into Second Life next year for the class. Other China International Schools Inworld (CISI) participants, namely Beijing International School and the Western Academy of Beijing, have already been experimenting with ITGS in Second Life as a pilot course during the last quarter of the 2010-11 school year. CAIS will join them as of 2011-12.

The CAIS course outline for this course follows. No lesson plans exist as of now. This is one of the things YHA has to do after this e-book is finished! Several scholars (including a Fulbright!) have been and will continue to do research on how 3D virtual worlds revolutionize courses like this. So far, the course has been a big hit with students and teachers alike. As you probably expected, YHA will be writing about this later too.

As you read the course description, try to imagine that the traditional way of teaching it was with blackboards and books! Well, with computers too, of course, but YHA and others just can't believe that teaching ITGS was even POSSIBLE before 3D virtual worlds! How does technology affect our global society? You're not likely to find that out with your nose stuck in hard copy! Why not get out there and interact with some members of this global society? ITGS involves making a product, which usually in the past meant an attempt at creating software, even though it's not a programming course. Via the NEW ITGS, students will be running their own international businesses in Second Life, ideally offering training products and services to prepare other IBO schools to work with 3D virtual worlds. More on all this in the next e-book!

PYP Intentions

CAIS has introduced OpenSimulator to PYP, or elementary school, classes. Nothing formal has been accomplished yet as to integrating 3D virtual worlds into the PYP curricula. But this is something else we at CAIS will be doing as of the 2011-12 school year (again, just as soon as YHA completes this e-book!). PYP classes, as you might already know, are based on "units of inquiry," not much different from the MYP "inquiry-based learning," so 3D should be an easy fit. CAIS wants to establish a definitive set of computer knowledge/skills our students will have by the time they finish the PYP program, so that they're ready for MYP. 3D activities will be a vital component of these goals, as virtual world knowledge/skills will be important for their future education and indeed entire lives.

As mentioned before, OpenSimulator enables cross-curricular instruction like nothing else ever has, or probably ever will. How do simple machines work? Well, create the prims and the programs and give everybody a demonstration! You saw a lesson plan for integrating art and music classes into OpenSimulator. Imagine how much fun little kids will get out of a project like that! PYP classes have been getting into blogging lately and we'd like to tie the real and virtual worlds together, e.g., having munchkins go on 3D field trips and then writing about it, including the exchange of letters, photos, etc., with the kids they visit. More later!

Changchun American International School
Information Technology in a Global Society (ITGS) 2011-2013
Course Outline

ITGS Course:

> "The study and evaluation of the impacts of information technology on individuals and society. Explores the advantages and disadvantages of the access and use of digitized information at the local and global level. Provides a framework for you to make informed judgements and decisions about the use of IT in social contexts."
>
> -- **ITGS Handbook**

Contact Information:

Our Beloved Leader: Use Basecamp or e-mail: **davidcaisorg@gmail.com**

Timelines:

- 2 years overall, 150 hours total or 75 per year
- 30 cycles X 3 classes per year, assuming 7-9 cycles per quarter
- 30-hour project will be split 50-50 over the 2 years

E-Textbooks:

- "Discovering Computers 2011" by Shelly and Vermaat, ISBN: 978-1439079263
- "A Gift of Fire: Social, Legal and Ethical Issues for Computing and the Internet" by Baase, ISBN: 978-0136008484
- We might also use a product/software development life cycle book, others

E-textbooks will be downloaded and/or accessed online via **www.ecampus.com** and/or **www.coursesmart.com**.

You Will Need:

- Ideally, a laptop (but not necessary)
- Flash drive
- Paper notebook if needed (best to be as paperless as possible)

Hardware/Software:

The CAIS Computer Lab will be hardware-sufficient re: PCs, Internet, etc. Most collaboration software will be free or inexpensive: nings, wikis, blogs, etc. 3D, project management, etc., software should be all free: SQL Lite/MySQL, Hippo and other viewers, OpenProj, etc. Second Life and ReactionGrid/other accounts will be free. E-textbooks will cost $115 per set/student.

Real-Life Scenarios:

You will be assisting IBO schools around the world in establishing a presence in Second Life and/or ReactionGrid (or perhaps another grid, ideally/ultimately the IBO Grid!). It is possible, but not guaranteed, that the client will be the IBO itself. If not, a grid company and/or a subset of IBO teachers/administrators/students probably will be. The IBO Schools Group has already been created in Second Life and one in at least ReactionGrid will be set up soon (depending on whether or not this can be done…matter of OpenSim versions).

Products:

You will be creating and/or contributing to nings, wikis, blogs, etc., in order to help IBO teachers/administrators/students. You will be creating tutorials and other training materials. Ultimately you will create your own products and sell them inworld (maybe in Real Life too!) via your own online businesses (potentially an actual revenue generator). Project management instruction will feature both technical and business tracks.

Research/Management:

You will be doing a lot of research, both independent and BL-supervised. You will be managing this course as a software project…because it IS a software project…and a big one. You will naturally be assisted every step of the way but the overall goal is to make you INDEPENDENT …for you to take charge of your learning process. You will be creating and managing the yearly, quarterly, etc., schedules…including each cycle's lesson plans after the first month or so.

Triangle Strands and Substrands:

- **Strand 1: Social and Ethical Significance**

1.1 Reliability and Integrity
1.2 Security
1.3 Privacy and Anonymity
1.4 Intellectual Property
1.5 Authenticity
1.6 Digital Divide and Access Equality
1.7 Surveillance
1.8 Globalization and Cultural Diversity
1.9 Policies
1.10 Standards and Protocols
1.11 People and Machines
1.12 Digital Citizenship

- **Strand 2: Application to Specified Scenarios**

2.1 Business and Employment
2.2 Education and Training
2.3 Environment
2.4 Health
2.5 Home and Leisure
2.6 Policies and Government

- **Strand 3: IT Systems**

3.1 Hardware
3.2 Software
3.3 Networks
3.4 Internet
3.5 Personal and Public Communications
3.6 Multimedia/Digital Media
3.7 Databases
3.8 Spreadsheets, Modelling and Simulations
3.9 Introduction to Project Management

- **Discovering Computers**

Chapter 1 Introduction
Chapter 2 Internet and the WWW
Chapter 3 Application Software
Chapter 4 Components of the System Unit
Chapter 5 Input
Chapter 6 Output
Chapter 7 Storage
Chapter 8 Operating Systems and Utility Programs
Chapter 9 Communications and Networks
Chapter 10 Database Management
Chapter 11 Computer Security and Safety/Ethics/Privacy
Chapter 12 Information System Development
Chapter 13 Programming Languages and Program Development
Chapter 14 Enterprise Computing
Chapter 15 Computer Careers and Certification

- **Gift of Fire**

Chapter 1 Unwrapping the Gift
Chapter 2 Privacy
Chapter 3 Freedom of Speech
Chapter 4 Intellectual Property
Chapter 5 Crime
Chapter 6 Work
Chapter 7 Evaluating and Controlling Technology
Chapter 8 Errors, Failure and Risk
Chapter 9 Professional Ethics and Responsibilities

Proposed Schedule:

Year 1	**Outline**	**Comments**
Quarter 1	Introduction to course Strand 1: Intro to social and	We will start with course planning, including roles/responsibilities for project management. Research will result in understanding of

	ethical issues via scenarios Strand 2: Potential for IBO and other schools worldwide Strand 3: Overview of hardware/software involved Discovering: Appropriate hardware/software info Gift: Appropriate privacy and similar info	different 3D systems, including a comparison grid of features. Research will result in understanding of potential for IBO and other schools (teachers and students). Research will determine best collaboration/communication tools (nings, wikis, blogs, etc.) via a comparison grid of features. Tools will be established, with accompanying procedures.
Quarter 2	Project planning continues, training begins Strand 1: Attitudes, overcoming obstacles Strand 2: Emphasis on careers, extrapolation Strand 3: Under the hood: databases and other components Discovering: Appropriate hardware/software details Gift: Appropriate training and similar info	You will continue with project planning, ideally with implementation. Training programs should begin at least in-house and ideally worldwide via collaboration/communication. Career opportunities will be explored with different paths determined. Still overview re: hardware/software but studies will start getting more detail-oriented. Roles for different 3D systems will be established: Quest Atlantis, e.g.

Quarter 3	Project underway: Second Life, ReactionGrid split or	Depending on arena progress, emphasis will be on Second Life or ReactionGrid (maybe

	combination Strand 1: Globalization and cultural differences Strand 2: Education and training, also policies and government Strand 3: Computer-aided design (graphics) and programming (scripting) Discovering: Graphics hardware/software, programming Gift: Intellectual property, ethics and responsibilities	both!). Tutorials start/continue via chosen collaboration/communication tool(s). Marketing campaign implemented to spread word to IBO schools. Communications protocols and other considerations will be emphasized. Policies and procedures for visitations and other issues will need to be considered, tested and documented. Hardware/software limitations will need to be studied and documented.
Quarter 4	Project stopping point and plan for next year established. Strands 1 & 3: "Drill downs" re: all technical and behavioral knowledge/skills Strand 2: Environmental issues not already addressed Discovering: Details; Gift: Behavior	Reflection re: where we've been and where we're going will be necessary. Previously established checklist of knowledge and skills will be evaluated via exercises and other activities. Year-wrapping drills re: exams and other course-end preparation.
Year 2	**Outline**	**Comments**
Quarter 1	Project planning for Year 2 (very dependent on progress during Year 1)	Official IBO Island. If not possible, leased lot(s), perhaps on a school-by-school or –region basis. Primary importance: Used by IBO teachers/administrators (and DP students

	Emphasis during Year 2 will be student self-sufficiency re: project management. All strands, all book content will focus on details that need addressing or readdressing and student ownership of project(s).	over 16!) worldwide. Official IBO Grid. Either standalone or piggybacked with ReactionGrid or other grid. Used by IBO teachers/administrators (and MYP students under 16!) worldwide.
Quarter 2	Any/all activities needed to meet goals.	If ultimate goals can't be reached, project scope will need to be readdressed.
Quarter 3	Any/all activities needed to meet goals.	Reasonable goals firmly in place, focus is one deliverables and deadlines. Perhaps some exam/paper preparation can be accomplished.
Quarter 4	Exam/Paper Preparation Project Completion Preparation	If we accomplish HALF of what's outlined here, CAIS will be world famous! ;)

Assessment Objectives:

- **Assessment Objective 1: Knowledge and Understanding of Specified Content**

Objective	**Course**
(a) Demonstrate an awareness of IT applications and developments in specified scenarios.	You will show you can install, configure, use, etc., application viewers, databases, etc., for the purpose of creating and managing 3D virtual worlds.
(b) Demonstrate an awareness of the social and ethical significance of specified IT applications and developments.	You will examine privacy, intellectual property and other ethical issues related to global participation in virtual worlds. By necessity, you will be dealing with the various social

	aspects of lots of different people interacting online together.
(c) Demonstrate technical knowledge of ITGS terminology, concepts and tools.	Project provides "the best of all possible (virtual!) worlds" for meeting the overall purpose of ITGS: to study and evaluate the impact of IT on individuals and society.
(d) Demonstrate technical knowledge of IT systems.	You will be configuring servers and clients, managing databases, writing scripts, etc.

- **Assessment Objective 2: Application and Analysis**

Criteria	**Course**
(a) Explain the impacts of IT applications and developments in specified scenarios.	An example of a specified scenario would be establishing IBO Island in Second Life. You will be responsible for evaluating professional and other developments of users.
(b) Analyze the social and ethical significance of specified IT applications and developments.	You will by necessity be analyzing the pros and cons of 3D virtual worlds from both social and ethical perspectives (social implying avatar interactions, ethical implying business and other transactions).
(c) Transfer IT knowledge and make connections between specific scenarios.	You will have ample opportunities to transfer knowledge/skills back and forth between the Second Life and OpenSim environments.

- **Assessment Objective 3: Synthesis and Evaluation**

Criteria	**Course**
(a) Evaluate local and global impacts of specified IT developments through individually researched studies.	You will be given ample opportunities to prove you can do individual research. All impacts of 3D virtual worlds will be covered, from a local (CAIS) to a global (IBO) scale.

(b) Evaluate a solution involving IT to a specified problem using knowledge of IT systems.	You will be creating hardware/software feature comparison grids in order to make project management decisions.
(c) Discuss the social and ethical implications of specified IT policies and developments.	You will be addressing potential user concerns in both written and verbal form. Ideally, you will conduct seminars inworld with teachers/students from around the world.

- **Assessment Objective 4: Use of ITGS Skills**

Criteria	**Course**
(a) Demonstrate evidence of project management in the development of a well-organized product to resolve a specific issue.	You will be managing your own project(s) through the Product Development Life Cycle. You will be taught, among other skills, new product development flowcharts to assist you.
(b) Use IT tools and the product development life cycle (PDLC) to create an original product in consultation with a client.	You will be creating both real and virtual world products (e.g., documentation in the real world, just about anything you can imagine in the virtual world).

Grading:

Assessment Component	**Weighting**
External Assessment (3 hours) Paper 1: Five structured questions that assess in an integrated way the three strands of the syllabus. Paper 2: Response to an unseen article. **Don't worry about all this now.**	70% 40% 30%

Internal Assessment (30 hours) The Project: You will develop an original IT product for a specified client: • Cover page • Original IT product • Supporting documentation (up to 2,000 words) PLUS: You will get your quarterly/yearly grades according to our internally-created checklists, rubrics, etc. (YOUR JOB!)	30%

More Resources/Further Study

Of course, you must get an account (if you haven't already!) on the **OpenSimulator** wiki:

http://opensimulator.org/wiki/Main_Page

The Second Life wiki is a must for your Favorites list:

http://wiki.secondlife.com/wiki/Main_Page

Mary Miner's wiki features lots of valuable information, including lesson plans:

http://virtualworldspresentation.pbworks.com/w/page/27991248/FrontPage

Reading **Hypergrid Business** ezine has already been mentioned as a good way to keep track of what's happening with OpenSimulator:

http://www.hypergridbusiness.com/

CAIS Co-teacher and Honorary Geek Alex Makosz maintains a good list of resources on his site:

http://www.makosz.org/index.php/vwblogs/

RezEd is a good hub for communicating about 3D virtual worlds:

http://rezedhub.ning.com/

The **OpenSim Users Digest** is great for technical information, but fair warning, it's for the ubergeeky:

https://lists.berlios.de/mailman/listinfo/opensim-users

Once you're in Second Life, search for groups related to OpenSimulator, such as…OpenSim! There are a lot of them. Some of the SL groups YHA belongs to include: International Society for Technology in Education (ISTE) Virtual Environments Special Interest Group (SIGVE), K-12 in SL Now, Virtual Worlds Education Roundtable (VWER), Virtual Worlds Best Practices in Education(VWBPE), Virginia Society for Technology in Education (VSTE), Teen Educators in SL, etc. Some of the OpenSimulator groups in SL seem to be very specialized and "spread out," with just handfuls of members in most. Maybe YHA and you devoted fans should do something to correct this, consolidating folks so that we can all connect, communicate and collaborate.

So you want some tomes on OpenSimulator? Well, this might be the only one! No, there is a very interesting book by a fashion designer:

- "Shengri La Spirit: A Designer's Perspective of the Making of OpenSim" by Shenlei Winkler, Fashion Research Foundation, ISBN: 978-0984117116

Go to Google and do some searches. Your Humble Author found some free PDFs on OpenSimulator architecture and other subjects. Again, they seem to be mainly for a technical audience and are certainly not aimed at teachers. But the available documentation will increase as OpenSimulator becomes more popular in education.

If you know of any (other!) books about OpenSimulator and education, please let Your Humble Author know, OK?

There are plenty of books on Second Life, though, so get some and "read between the lines." Many if not most of them are "over the heads" of K-12 students, however. You'll have to do some interpretation.

Three exceptions are the great Jeff Heaton books, which are featured on the desktop of every CAIS computer lab PC as e-textbooks. Here they are, all published by Heaton Research, Inc.:

- "Introduction to Linden Scripting Language for Second Life," ISBN: 978-1604390049
- "Scripting Recipes for Second Life," ISBN: 978-1604390001
- "Introduction to Textures, Animation Audio and Sculpting in Second Life," ISBN: 978-1604390025

YHA's native-speaker MYP students can use (and love!) these books. As for our kids who can barely utter a sentence in English, these aren't much different from all the other books they can't understand. But 100% of our munchkins enjoy the included scripts! In fact, it was a learner who discovered the slideshow script that's been used to display student art in OpenSimulator!

You can get the e-book versions, with no DRM or other considerations, directly from his website for US$10 each, a darn good bargain:

http://www.heatonresearch.com/book/cat/3

On to the other Second Life books, not all of which are necessarily written for teachers.

Three books (e- or otherwise) every 3D Virtual Worlds Pioneer such as yourself should have in your professional and/or personal library are the "official guides," all from Sybex:

- "Second Life: The Official Guide" by Michael Rymaszewski, et al., ISBN: 978-0470227756
- "Creating Your World: The Official Guide to Advanced Content Creation for Second Life" by Aimee Weber, et al., ISBN: 978-0470171141
- "Scripting Your World: The Official Guide to Second Life Scripting" by Dana Moore, ISBN: 978-0470339831

Lately YHA has been getting a lot of useful information from these two books:

- "A Practical Guide to Using Second Life in Higher Education" by Maggi Savin-Baden, Open University Press, ISBN: 978-0335242146
- "The Second Life Grid: The Official Guide to Communication, Collaboration and Community Engagement" by Kimberly Rufer-Bach, Wiley Publishing, ISBN: 978-0-470-41291-6

Two books that came in handy when YHA was teaching business classes, and will be useful again next year as e-commerce again becomes a topic of instruction:

- "The Entrepreneur's Guide to Second Life: Making Money in the Metaverse" by Daniel Terdiman, Wiley Press, ISBN: 978-0470179147
- "How to Make Real Money in Second Life: Boost Your Business, Market Your Services, and Sell Your Products in the World's Hottest Virtual Community" by Robert Freedman, McGraw-Hill, ISBN: 978-0071508254

Other interesting books (in alphabetical order):

- "Cutting-Edge Social Media Approaches to Business Education" by Charles Wankel, Information Age, ISBN: 978-1617351167
- "Empowering Students with Technology" by Alan November, Corwin Press, ISBN: 978-1412974257
- "Higher Education in Virtual Worlds" by Charles Wankel, Emerald Group Publishing, ISBN: 978-1849506090
- "Learning and Teaching in the Virtual World of Second Life" (edited) by Judith Molka-Danielsen, Tapir Academic Press, ISBN: 978-82-519-2353-8

- "Learning in 3D" by Karl Kapp, Pfeiffer Publishing, ISBN: 978-0470504734
- "Learning Online with Games, Simulations and Virtual Worlds" by Clark Aldrich, Jossey-Bass, ISBN: 978-0-470-43834-3
- "Making Content Comprehensible for English Learners: The SIOP Model" by Jane J. Echevarria, et al., Allyn & Bacon, ISBN: 978-0205518869
- "Rethinking Education in the Age of Technology" by Allan Collins, Teachers College Press, ISBN: 978-0807750025
- "The Complete Guide to Simulations and Serious Games" by Clark Aldrich, Pfeiffer Publishing, ISBN: 978-0470462737
- "The Virtual Worlds Handbook: How to Use Second Life and Other 3D Virtual Environments" by Elizabeth Hodge, Jones and Bartlett Publishers, ISBN: 978-0763777470
- "The World Is Open: How Web Technology Is Revolutionizing Education" by Curtis J. Bonk, Jossey-Bass, ISBN: 978-0470461303
- "Training and Collaboration with Virtual Worlds" by Alex Heiphetz, McGraw-Hill, ISBN: 978-0071628020

YHA is currently interested in how 3D virtual worlds can be used to better teach both gifted/talented and special needs students by applying the concepts of UNschooling:

- "The Unschooling Unmanual" by Nanda Van Gestal, et al., The Natural Child Project, ISBN: 978-0968575451
- "Unschooling Rules: 55 Ways to Unlearn What We Know About Schools and Rediscover Education" by Clark Aldrich, Greenleaf Book Group, ISBN: 978-1608321162

But this will have to wait for another e-book!

About Changchun American International School

Changchun American International School (http://www.caischina.org) is the first and only international school for 3-19-year-old students established in Jilin Province, China. It is an International Baccalaureate Organization World School. Final approval for the opening of this exciting school was granted by the Ministry of Education in June 2007.

> "The International Baccalaureate® (IB) is a non-profit educational foundation, motivated by its mission, focused on the student. Our three programs for students aged 3 to 19 help develop the intellectual, personal, emotional and social skills to live, learn and work in a rapidly globalizing world. Founded in 1968, we currently work with 3,070 schools in 139 countries to develop and offer three challenging programs to over 872,000 students."

- IBO Website, http://www.ibo.org/general/who.cfm

The programs of the International Baccalaureate Organization (Primary Years, Middle Years and Diploma), recognized worldwide, are housed in our new, modern and state-of-the-art facilities offering our students access to over 15,000 square meters of green areas and 5,600 square meters of purpose built teaching and learning facilities.

We offer both expatriate and local students access to an advanced and dynamic curriculum using the latest technology and teaching methodologies in an environment designed to challenge and stimulate them to become worthy citizens of the future as well as lifelong learners.

Changchun American International School responds to the dynamic environment that is the China of today and especially that of Northeast China. It aims to play a key role in providing the growing expatriate community with modern, holistic and internationally recognized schooling, thereby attracting greater numbers of foreign professionals and their families to the area. In a technology- and knowledge-based economy now and in the future, students will have to be equipped with this kind of skills-based curriculum to apply and use their knowledge and compete effectively. The IBO also promotes the development of ethics, character and international awareness, all essential in an ever-changing and fast-paced world.

Changchun American International School (CAIS) is a community school committed to working in partnership with its staff, foreign families, the local community, its financial investors and the government to provide a safe, holistic and successful school firmly focused on the future needs and successes of our students.

The primary goal of CAIS is to continue developing into The 21st Century Educational Technology School, leading not just the IBO but all international schools inworld to OpenSimulator and Second Life. During the 2011-2012 year, CAIS will present a proposal to the IBO for the creation of an OpenSimulator IBO Grid, which will ultimately include all 3,000 schools. CAIS will also propose the establishment of an IBO Island in Second Life, so that teachers and older students can connect, communicate and collaborate with people and organizations not associated with K-12 schools, international or not. If not the IBO, we'd use

another grouping. International or just plain all K-12 schools? Would you and/or your organization like to help with research, proposals, grants, etc.? CAIS would welcome having the creation of the the K-12 Grid and/or Island included as part of, e.g., a Ph.D. candidate's dissertation studies and/or an institution's research project. Please contact CAIS (**info@caischina.org**) if you're interested.

About Your Humble Author

Regarded as the sexiest educational technologist ever, in both the real and virtual worlds, YHA devoted more than 15 years to the computer business (serving in just about every imaginable capacity, from programmer/analyst to operations director) before switching to teaching. Almost immediately after YHA and a third of his fellow employees were laid off in 2001, he got two university gigs: teaching an online, graduate-school-level software engineering course, and designing classes for a technical writing curriculum. He never looked back.

In 2002, YHA decided to add travel to his career change, moving to a university in South Korea. After streamlining his first coordinator position out of existence, he became the school's first Technology Specialist, designing, developing and delivering the Computer Assisted Language Learning program. YHA moved on to the school's International Business Department, where he taught computer and (yes) business classes to students from Korea, China, Russia, Vietnam and other countries. It was 2006 when he started taking students inworld to Second Life, because he

needed a way to teach, e.g., computer programming and business management to students who were having trouble reading English textbooks. He spent another year in Korea, as a professor in a college's computer science department, where he incorporated Second Life into almost all his IT classes, as well as other teachers' English courses, taking over 500 learners inworld.

In 2009, he decided to again change venues, to K-12 international schools. His first job didn't last long. YHA was run out of town for advocating 3D virtual worlds, because the natives think avatars steal souls…really! He escaped to Changchun in the summer of that year and that's when the 3D miracle making began in earnest. In addition to creating Virtual CAIS in Second Life, YHA established Caisland via an OpenSimulator private grid and a hypergridded grid in ReactionGrid. Hopefully all this grid talk makes some sense now that you've finished this e-book! When this publication was completed, YHA had been invited to present on 3D virtual worlds at no less than five international educational technology conferences during 2011. Contact YHA (**ddeeds@caischina.org**) if you'd like a presentation or just some help getting started with 3D.

No Subtlety Whatsoever

There's no point in being bashful about this. K-12 students (IBO, international, otherwise) NEED an OpenSimulator grid! K-12 students and teachers NEED a Second Life Island! Our planned proposal to the IBO has already been mentioned. Perhaps your organization, or one that you know of, would be interested in addition to (or instead of!) the IBO? We wouldn't need much money to get started, especially if each participating school pays its own way, but having an academic and/or corporate sponsor or two sure would be nice. The main thing is to get other people involved. YHA could use some assistance! Please contact CAIS (**info@caischina.org**) if you and/or your organization would like to get involved. Thanks and see you inworld soon!

