1 Introduction

Currently browsing virtual worlds requires specialized dedicated software. Powerful computers are needed to run the software at a usable level. One solution to the problem is the creation of a web based virtual world browser.

1.1 Purpose

Accessing virtual worlds requires specialized dedicated software to connect and use the virtual worlds. The trouble with this software is it requires a powerful computer to provide a pleasant user experience. Virtual world viewers have many uses including allowing the creation of content, providing a window into the virtual world to explore and collaborate with others connected to the virtual world. A web based viewer would allow the use of less powerful computers to connect and view the virtual world.

1.2 Background

In January 2007 Second Life was released by Linden Research, often referred to as Linden Labs. Sparked by the innovation provided in the Second Life platform Opensim and its derivatives have since built a mostly-Second Life compatible server in C# (a programming language similar to Java created and backed by Microsoft). Opensim is based on what was once known as LibSecondLife later renamed to LibOpenMetaverse libOMV for short. "What started with Wireshark, a disassembler, and a desire to bring non-player characters into the Second Life world turned into a five year development effort by over 30 people to create one of the largest and most active open source C# projects at the time." [http://jhurliman.org/bio]

Today LibOMV provides the low level communication infrastructure to allow users to login to Second Life or Opensim Virtual Worlds.

In July of 2007 Katharine Berry began development of a text only web based client for Second Life called Ajax Life. She maintained the AjaxLife service until 2010 when Linden Labs notified her that the service wasn't feasible. Their determination of infeasibility was based on the fact that their user management tools could not identify users based on their IP addresses because the web based viewer masked them due to its design. Before she called it quits she had many features working including local chat(location based), user to user IM, inventory view, user profiles, nearby avatars, map with teleports, contacts and search. Katharine was able to achieve this using LibOMV running on her server and providing an interface that javascript running in one's web browser can use to communicate with Second Life.

In May 2009 Lkalif began development of Radegast text only metaverse client. Based on

LibOMV and developed in C# Radegast stated goal is to provide a "lightweight client for connecting to Second Life and OpenSim based virtual worlds"[http://radegast.org/wp/]. Recently development has added a 3D rendering with minimal interaction to the client.

RealXtend http://realxtend.org/2015/02/03/the-realxtend-application-portfolio/#more-529
RealXtend WebTundra Webgl viewer https://github.com/realXtend/WebTundra

Pixie Viewer http://pixieviewer.com/
Pixie Viewer shows that rendering of objects in the browser is possible and avatars.

OpenVCE

OnLook http://metaverseink.com/blog/?p=566
This is not a web based solution but cool none the less.

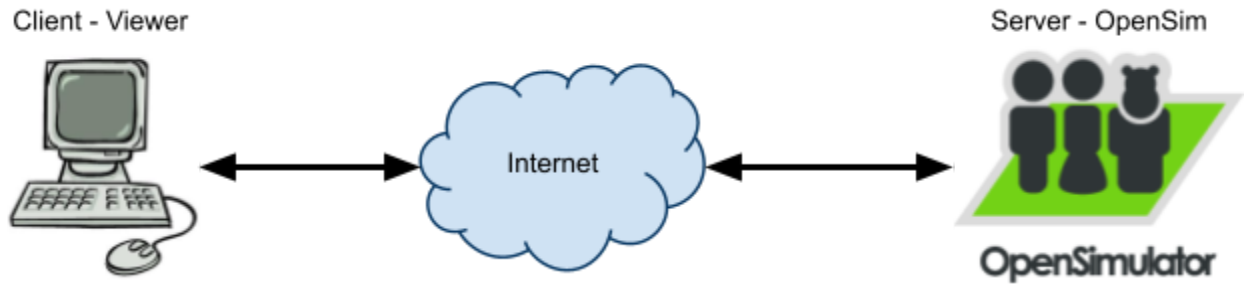
SceneVR http://www.scenevr.com/
SceneVR shows that rendering of object in the browser is possible.

Emscripten https://github.com/kripken/emscripten
Has anyone tried to compile the SL viewer with Emscripten?
https://github.com/kripken/emscripten/blob/master/tests/openjpeg/CHANGES#L416

Virtual World Framework
https://virtual.wf/

1.2.1 Current SL/OpenSim Architecture

All movement is processed by the server (OpenSim) and relayed to the client. I think the raw arrow keys are sent to the server and the movement data is sent back.



1.3 Assumptions

- Maximize use/reuse of existing code and technologies
- Limited user interface.

- Unable to create content
    - Should be easy enough to edit notecards and view textures
- Will the avatar be rendered?
- Ability to move around the virtual world.

1.4 Constraints

- **Must be web based without need for browser plugin**
- Limited lighting and shadows
- Access OpenSim based worlds without a browser-plugin
- Might not render mesh
- WebGL polygon counts in the browser could limit prim count and mesh complexity
- Code must be BSD/MIT license compatible

2 Methodology

There are many possible approaches to developing a Web Based Virtual World Viewer. The following requirements describe a system based on the AjaxLife architecture as described in depth in Appendix B.

3 Functional Requirements

This section describes the levels of the possible system. Starting from the most barebones working prototype to a workable product.

3.2 Levels

Below are the levels in which the web viewer could work. The first, Level 0, describes the features in AjaxLife and the following levels build upon that.

3.2.0 Level 0: AjaxLife -- Starting Point

- Login to OpenSim VW & Logout
- Chat (includes friend online/offline notifications)
- IM (user to user messaging)
- Group Messages
- Basic User Profiles
- Contacts (Online, Offline, Groups)
- World Map
- People Search
- Nearby Avatars

- Inventory
- Statistics
- Region Map (Mini Map)

See Appendix A for screenshots.

3.2.1 Level 1: Walkable 3D map

- Display 3D content from the perspective of a person or avatar in the VW.
- Simple avatar, single prim, solid color non-changing shape.
- Physics handled by OpenSim Server
- No client side LSL/OSSL

3.2.2 Level 2: Interactive Walkable Map

- Open and use notecards, including editing. (No embeds, pics etc)
- Customizable UI, possibly having a HUD with teleport locations
- Favorite Teleport locations
- Ability to show outside webpages (iframe?)

3.2.3 Level 3: More Usable inventory & User Interface

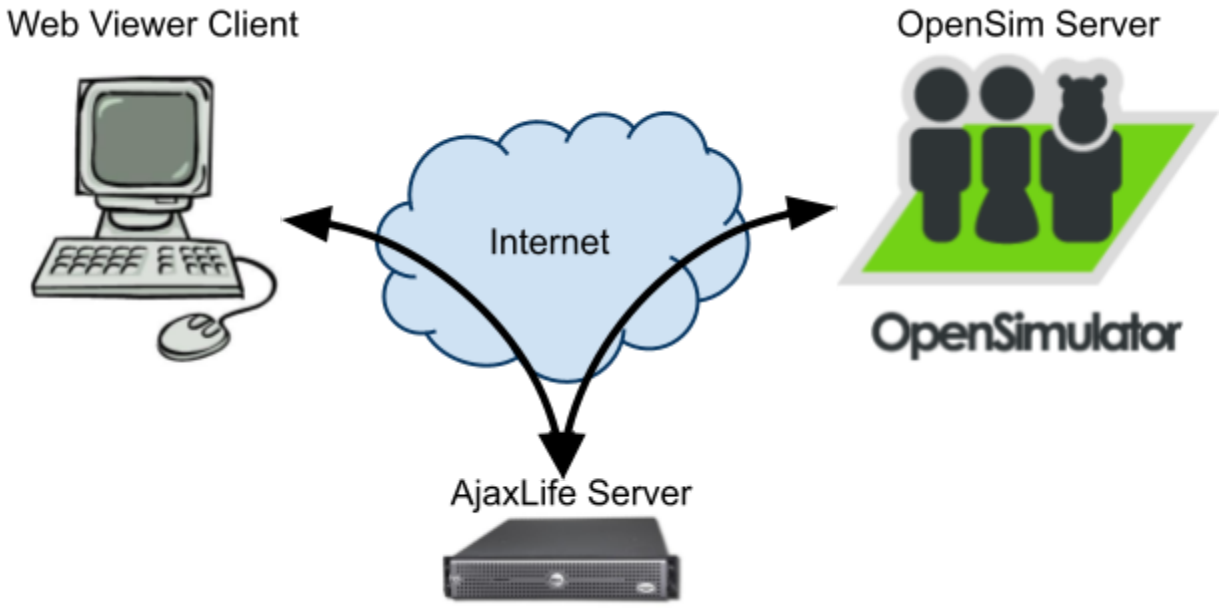
- LSL/OSSL client support
    - llSetTargetOmega -- make objects spin
    - touch_start - handle user interactions

4 Possible Solutions

The following section discusses possible solutions from a high level architectural point of view.

4.1 Ajax Life architecture

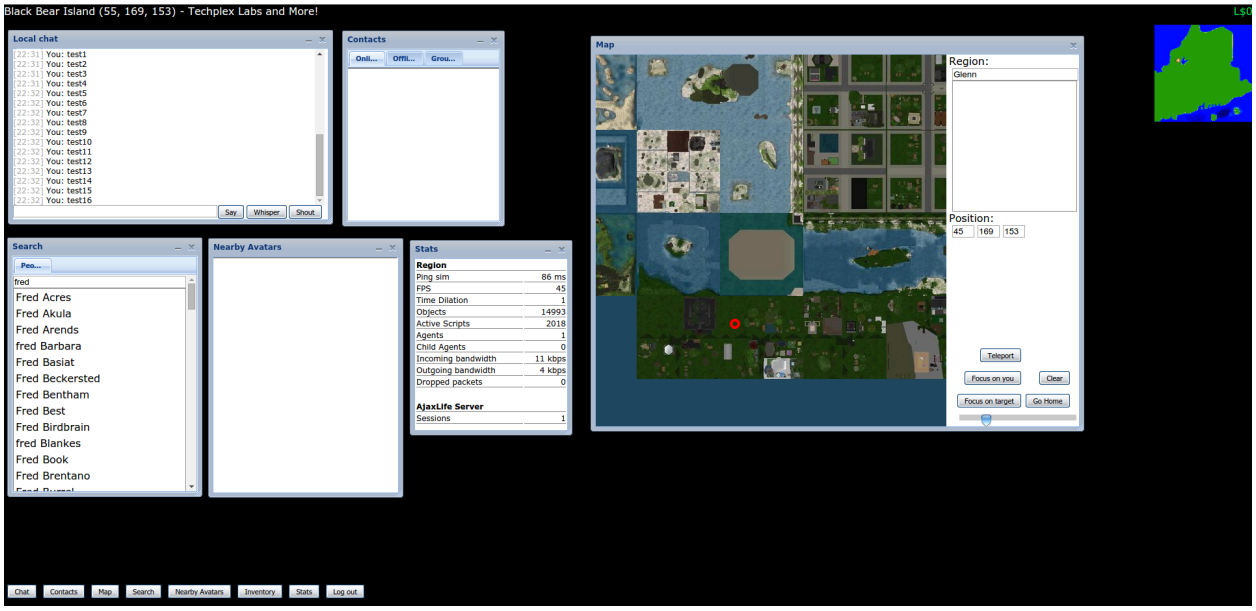
In the architecture depicted below, the client acts as a dumb terminal sending the user's actions to the AjaxLife Server. The AjaxLife Server then manages the connection to the Opensim Server by relaying messages to and from the AjaxLife Server. The AjaxLife Server will use LibOpenMetaverse to communicate with Opensim's existing client connection capability.

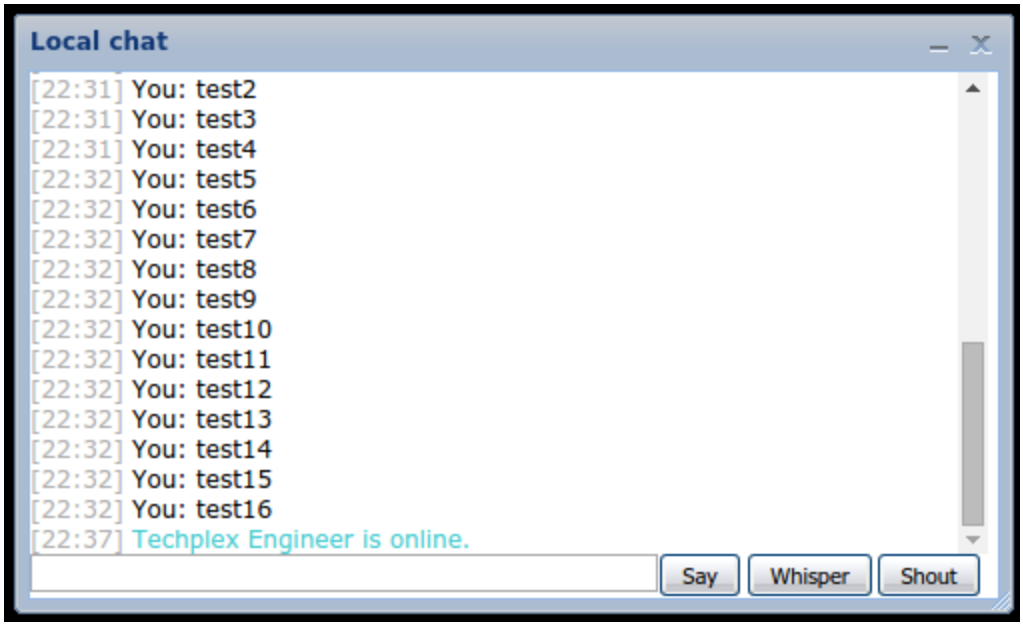


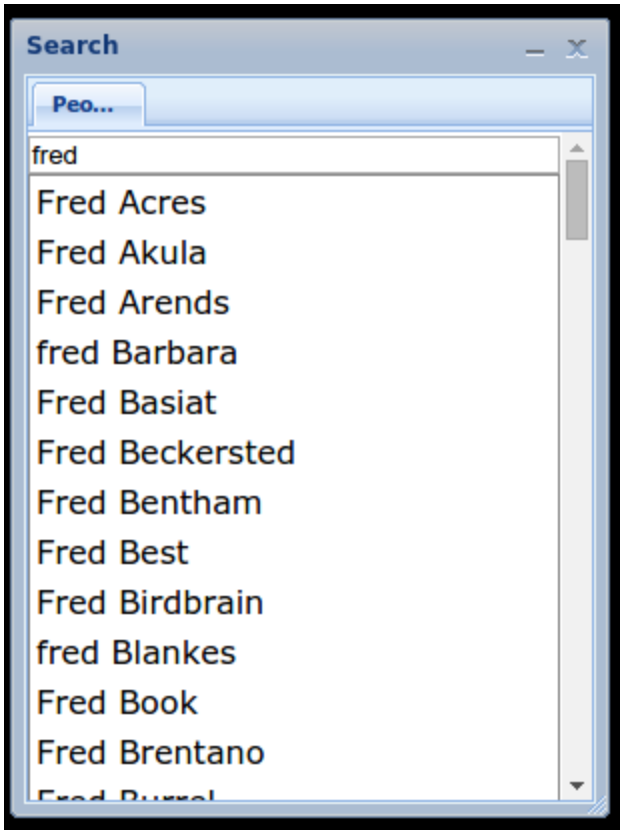
4.2 Websockets direct to Opensim

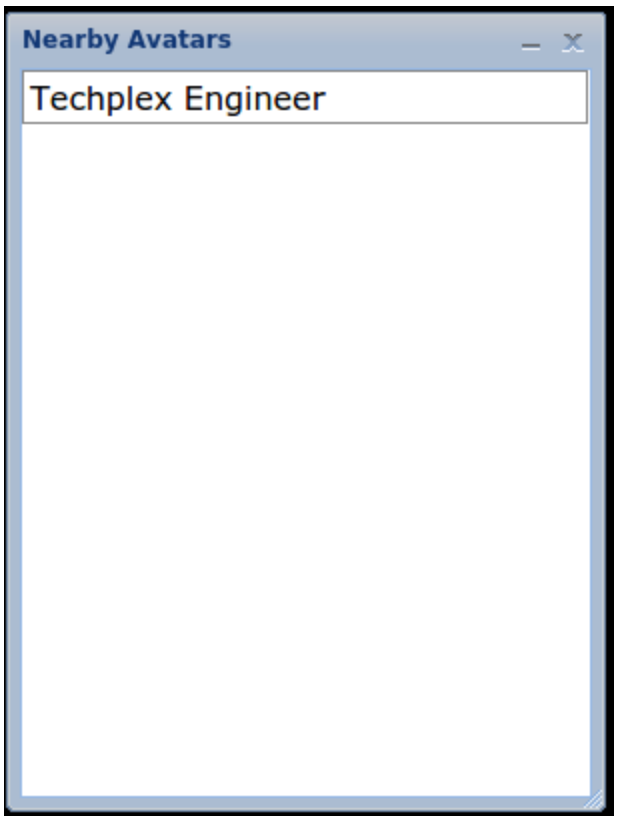
This approach has the benefit of not requiring an extra service in between the client and opensim. Ideally the Web Viewer Client could directly connect to opensim using Websockets and provide a viewer interface.

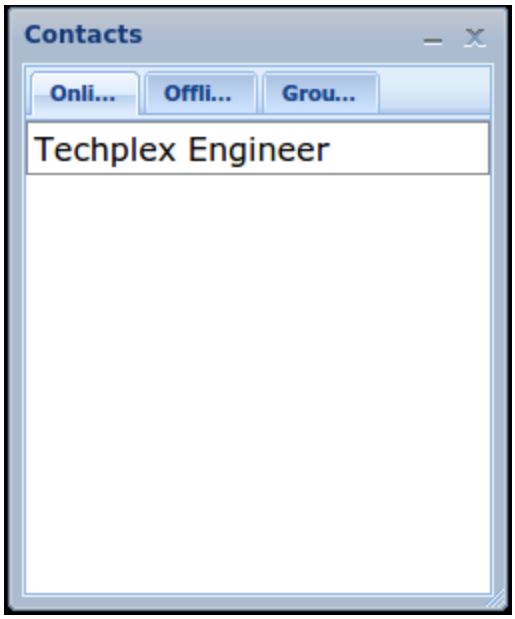
Appendix A - Ajax Life Screenshots

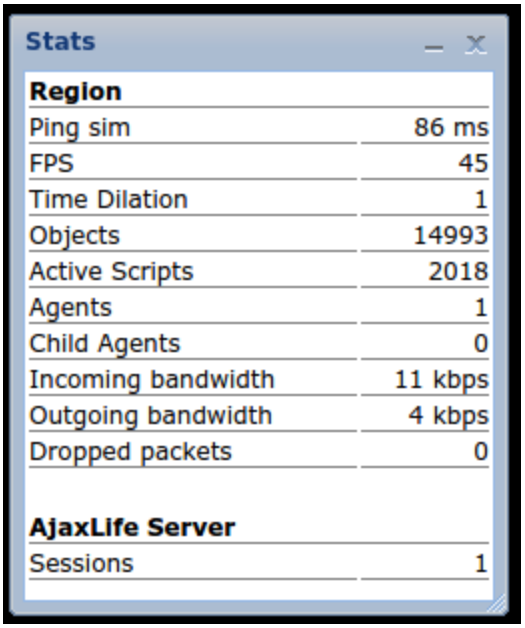

Black Bear Island (55, 169, 153) - Techplex Labs and More!
L$0
Local chat
[22:31] You: test1
[22:31] You: test2
[22:31] You: test3
[22:31] You: test4
[22:32] You: test5
[22:32] You: test6
[22:32] You: test7
[22:32] You: test8
[22:32] You: test9
[22:32] You: test10
[22:32] You: test11
[22:32] You: test12
[22:32] You: test13
[22:32] You: test14
[22:32] You: test15
[22:32] You: test16
Say
Whisper
Shout
Contacts
Onli...
Offli...
Grou...
Map
Region:
Glenn
Position:
45
169
153
Teleport
Focus on you
Clear
Focus on target
Go Home
Search
Peo...
fred
Fred Acres
Fred Akula
Fred Arends
fred Barbara
Fred Basiat
Fred Beckersted
Fred Bentham
Fred Best
Fred Birdbrain
fred Blankes
Fred Book
Fred Brentano
Nearby Avatars
Stats
Region
Ping sim 86 ms
FPS 45
Time Dilation 1
Objects 14993
Active Scripts 2018
Agents 1
Child Agents 0
Incoming bandwidth 11 kbps
Outgoing bandwidth 4 kbps
Dropped packets 0
AjaxLife Server
Sessions 1
Chat
Contacts
Map
Search
Nearby Avatars
Inventory
Stats
Log out

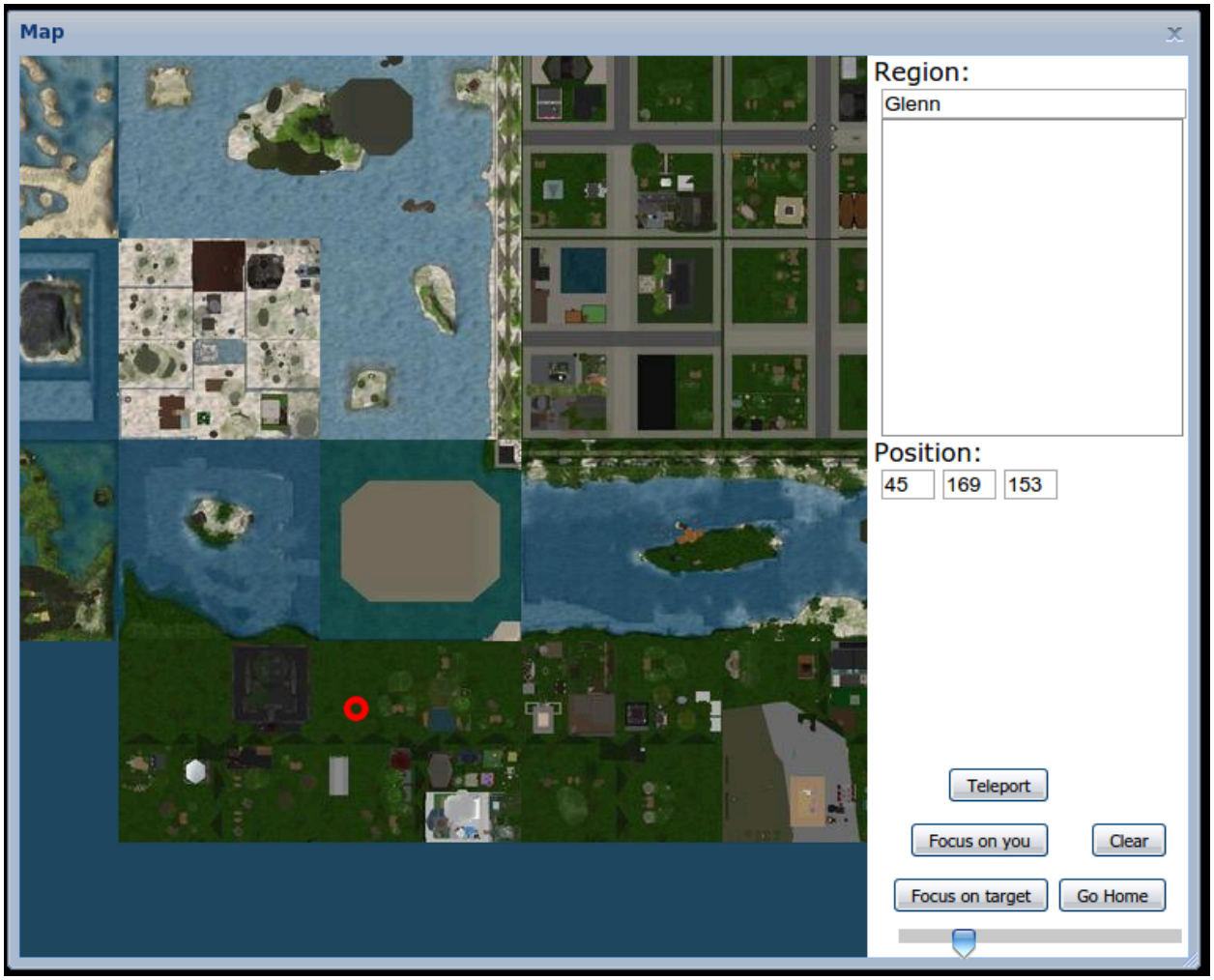

Local chat
[22:31] You: test2
[22:31] You: test3
[22:31] You: test4
[22:32] You: test5
[22:32] You: test6
[22:32] You: test7
[22:32] You: test8
[22:32] You: test9
[22:32] You: test10
[22:32] You: test11
[22:32] You: test12
[22:32] You: test13
[22:32] You: test14
[22:32] You: test15
[22:32] You: test16
[22:37] Techplex Engineer is online.
Say
Whisper
Shout


Search
Peo...
fred
Fred Acres
Fred Akula
Fred Arends
fred Barbara
Fred Basiat
Fred Beckersted
Fred Bentham
Fred Best
Fred Birdbrain
fred Blankes
Fred Book
Fred Brentano


Nearby Avatars
Techplex Engineer


Contacts
Onli...
Offli...
Grou...
Techplex Engineer


Stats
Region
Ping sim 86 ms
FPS 45
Time Dilation 1
Objects 14993
Active Scripts 2018
Agents 1
Child Agents 0
Incoming bandwidth 11 kbps
Outgoing bandwidth 4 kbps
Dropped packets 0
AjaxLife Server
Sessions 1



Minimap and Linden Balance

Appendix B - Ajax life Architecture

Katharine would be a better person to write this section, but I'll document her system as best I can.

As a preface there are two main parts to her system. The first is the C# application which provides the non-RESTfull API and handles users sessions. The second part is the client side javascript which runs in the user's web browser. From here forward let's refer to the C# application as the server and the client side javascript as the client.

Appendix Z - Resources

FRD:
http://www.jiludwig.com/templates/FRDTemplate.doc

http://opensimulator.org/index.php?title=User:Fim