

Software Infrastructure Part 2

CS 422: Intelligent Avatars Lab

Spring 2010

Software to Install

Install the following on Windows:

(Mac/Linux users should either dual boot into Windows or use a virtual copy)

1. Second Life Viewer
 2. Visual Studio 2008 with C# Support (comes with .NET 3.5 framework)
 - Do not use other versions of Visual Studio or .NET
 3. Lib Open Metaverse
 4. ~~Cygwin with OpenSSH~~ **Putty** No longer need SSH tunneling
 5. MySQL Connector/ODBC driver
- Note: everything can be downloaded for free, see the Software Resources page on the class website.

Database

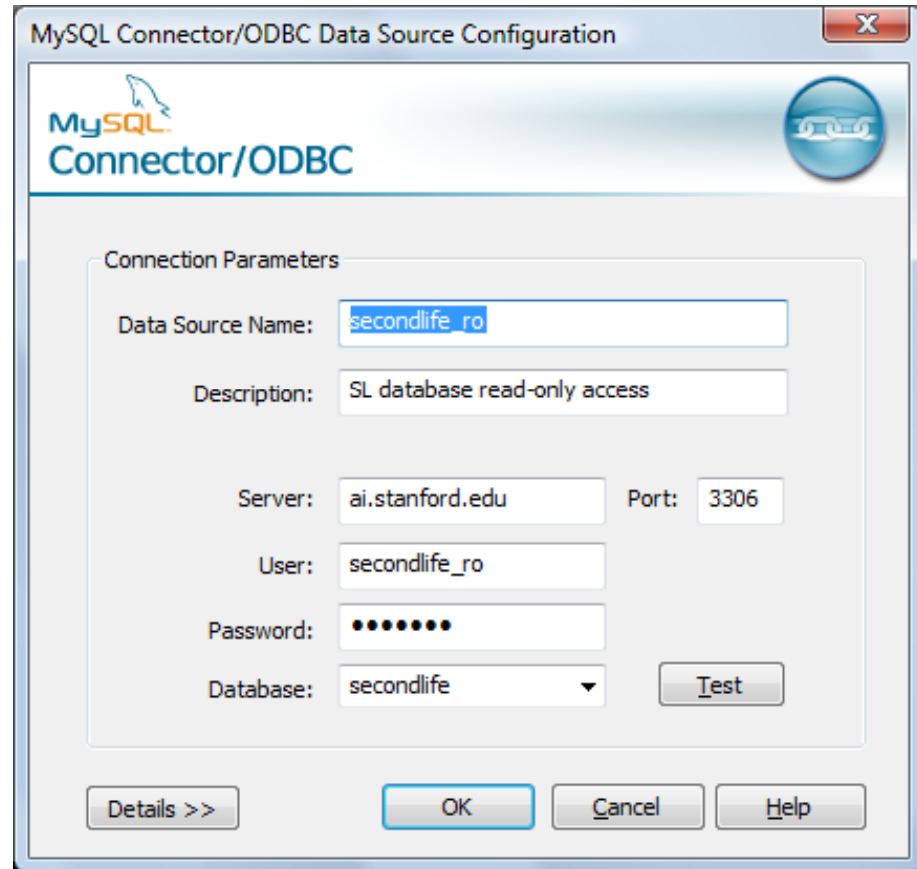
- MySQL database on ai.stanford.edu
- Collecting the following information when the Avatar Shadow is attached:
 - Each IA's location every 10 seconds
 - All local chats heard/spoken by IA
 - Any time the IA teleports
 - Information on parcels visited
 - Meetings of IA with other avatars
 - “Meeting” is any two avatars within local chat distance of each other

Database – SSH Tunnel

- No longer needed. Wooahoo!

Database – ODBC Connection

- Control Panel > Administrative Tools > ODBC Data Sources
- System DSN > Add... > MySQL ODBC 5.1 Driver
- Use following configuration:
 - Data Source Name = secondlife_ro
 - Server = ai.stanford.edu
 - Port = 3306
 - User = secondlife_ro
 - Password = (see TA)
 - Database = secondlife
- Test your connection
- Demo!



Database

- Once your ODBC Connection is working correctly, two ways of querying the database:
 1. Write your own SQL queries in C#
 2. Use our API (preferred)
- To use our DB API, download the 'CS422API' Solution from the class website, which contains:
 - Code for DB API
 - Demo code how to use the API
 - Basic stalker bot code (you will be extending this for the homework)

DB API

- public static `UUID[]` `getAvatars`(`Vector3` location, `DateTime` start, `DateTime` end)
 - Returns all IAs that were within 20m of the location
- public static `ChatEvent[]` `getChats`(`UUID` avatar, `DateTime` start, `DateTime` end)
 - Returns all chats heard or said by the IA
- public static `Vector3[]` `getPlaces`(`UUID` avatar, `DateTime` start, `DateTime` end, `Ordering` order)
 - Returns all unique locations of the IA
- public static `Meeting[]` `getMeetings`(`UUID[]` avatars, `DateTime` start, `DateTime` end)
 - Returns meetings involving all of the specified avatars (avatars[0] must be an IA)
- See Lib Open Metaverse API doc for `UUID` and `Vector3`

DB API

```
public struct ChatEvent
{
    public UUID speaker;
    public UUID listener;
    public DateTime timestamp;
    public Vector3 location;
    public string message;
}
```

```
public struct Meeting
{
    public UUID[] avatars;
    public DateTime start;
    public DateTime end;
    public Vector3 location;
}
```


Database Schema

Avatars		
a_id	char	Avatar UUID
a_name	varchar	Avatar full name
a_shadow	int	1 if has avatar shadow object

Moments		
m_id	int	Moment ID (DB auto-generated)
a_id	char	Avatar UUID
p_id	char	Parcel UUID
m_timestamp	datetime	Timestamp of recorded location
m_pos_x	double	Global X coordinate
m_pos_y	double	Global Y coordinate
m_pos_z	double	Global Z coordinate
m_teleport	int	1 if was a teleport

Database Schema

Chat		
c_id	int	Chat ID (DB generated)
a_id	char	Avatar UUID of who heard the chat
p_id	char	Parcel UUID
c_timestamp	datetime	Timestamp of chat
c_pos_x	double	Global X coordinate
c_pos_y	double	Global Y coordinate
c_pos_z	double	Global Z coordinate
c_speaker	char	Avatar UUID of the speaker
c_message	text	Chat message
c_channel	int	Chat channel, default is 0

Database Schema

Meetings		
me_id	int	Meeting ID (DB auto-generated)
m_id	int	Moment ID
a_id	char	Avatar UUID of participant

Parcels		
p_id	char	Parcel UUID
p_name	varchar	Parcel name
p_region	varchar	Region name
p_description	varchar	Parcel description
p_owner	char	Owner UUID
p_group	char	Group owner UUID
p_area	int	Size of the parcel

Visual Studio

- IDE for .NET similar to Eclipse
 - Has all the nice IDE features like completion suggestions, refactoring, and a debugger
- Highest level are Solutions > Project > Files
- To compile CS422API, may need to add reference to Lib Open Metaverse dlls
- May want to install AnkhSVN plugin to make it easier to do source control and share code

C#

- Microsoft's answer to Java
 - Basically everything that Java has, C# has as well*
- MSDN is your friend, contains documentation for the C# API: <http://msdn.microsoft.com/en-us/library/default.aspx>
- Syntactically very similar to C++ and Java
- Some key differences with Java:
 - Uses namespaces to organize classes, they do not have to correspond with the actual directory structure
 - File names do not have to be the same as class names, can have multiple top-level classes in the same file
 - Has both Classes (pass by reference) and Structs (pass by value)
 - Event handling is done using delegates (function pointers)
 - To write to stdout use `Console.WriteLine()`

*Well, except for support on non-Windows platforms

Open Lib Metaverse

- Open source API in C# for controlling avatars
- API doc:
<http://lib.openmetaverse.org/docs/0.7/>
 - Browse through this API before doing any coding, a lot of basic functions have already been written for you
- Helpful tutorials (some are out of date):
http://lib.openmetaverse.org/wiki/Developer_Portal

CS422API Solution

- Program.cs
 - Source code for the DB API
- StructsAndClass.cs
 - Structs and classes used by the DB API
- Demo.cs
 - Demo and examples for the DB API
- Follow.cs
 - Code for the stalker bot
- Demo!

Summary

- Tell us your project team and SUNetid before leaving
- Homework due in a week (5PM on 4/7/10)
 - Not hard, but start early! Lots of software to install and potential hiccups along the way...
 - Wait for access to ai.stanford.edu before doing parts of homework that involve the database
- Recitation (Office Hours) in Gates 128 on Monday during class time (no lecture)
- Next class meeting: Guest lecture by Bing Gordon, co-founder of Electronic Arts, on Wednesday 4/7/10