Darkstar usability: Week of 8/31/09

Akash was at GDC last week, so he was not working with Project Darkstar.

Rick

These two weeks I have worked on scripting a mini game on Panda3D with the Server APIs of Darkstar.

I am pretty happy about the robust functions and debugging messages Darkstar provide.

But Sometimes I feel helpless when I am coding because there are no online documentations like a dictionary that I can consult it to the APIs very quickly, especially the explanations or examples of some specific API.
For instance, I was not very sure when I should implement ManageObject with Serializable as when I should not. And finally I got the answer from Seth's email.

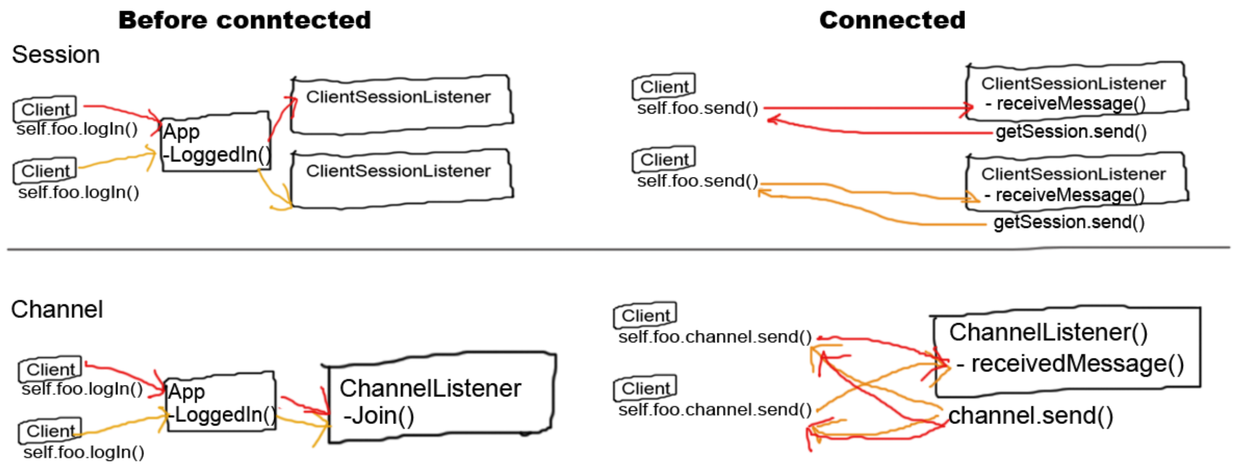
In addition to the Server API Programming Guide we have now, Still I think I need other resources to learn Darkstar further more if there is any.

Ya-Ching

This week I worked on ChannelListener. In the beginning, I felt confused with the receiveMessage() function of SessionListener and ChannelListener. Finally I understood how it works. If I had the client send message through channel, then the receiveMessage() in ChannelListener will be invoked. On the contractly, the client send message from the session, then SessionListener will get the message.

It took me one day to figure out it. I guess it is because the same function name (receiveMessage)and similar class name (both of them are Listener.)cause me confused. (And ClientSession actually is different from ClientSessionListener, but it's easily confused.)
Now I understand the structure, everything is easy.

It would be great if there is a simple but clear diagram showing how Channel and Session works. (The attachment is the picture about my understanding of Channel and Session, just as an example.)



Lynne

I also worked on implementing channel listener this week. With the channel listener, the server doesn't need to broadcast to the client through session but through channel instead. I found that channel listener is really convenient to use for dealing with specific object managing between server and the client. However, it should be always kept in mind that there is always a reference to the channel and the pointer of the reference is not null. I was stuck a few times because of these problems.

As Ya-ching mentioned, if there is a simple diagram of how the channel and session work in Darkstar's tutorial would be really helpful.

John

Most of my problems this week stemmed more from the Java language than Project Darkstar API. To help with network code during prototyping, I wrote the inverse of the PandaCommander class so that the clients were able to tell the server to run commands without having to write code to specifically handle the packet. I am well aware that this is terribly unsafe but it allows the team to focus on getting the game working before having to tackle how to optimize the networking code.

Java issues aside, the only real issue I had with Project Darkstar this week was that of authenticating clients with the server. I browsed the forums and took a look at the Bunny Hunters code for it, then tried to get that code to work in the context of my feature testing app (Roaming World). After a couple hours of not getting it working I decided to scrap it and write my own authentication.

However, since I expected writing proper authentication (as a service/plugin) in the Project Darkstar may of taken longer than desired, I wrote an authentication scheme that lets the client connect with a name, then authenticate or register than name once connected.

Lastly, managing multiple games seems slightly more difficult with Project Darkstar than without. However I have not delved very deeply into this and this coming week I will be hitting on this more.