Darkstar usability: Week of 9/28/09

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

Rick

I've also been mostly working on python scripts last week.
I have no issues on Darkstar, too.

Ya-Ching

This week I worked on zoing. Because I'm familiar with Darkstar Channel, I didn't meet too much problems. However, I feel I need more control of Channel. For example, the message will be broadcasted to all the clients who listen to the channel. If I can lock certain client then this client won't able to receive the message from this channel until I unlock this client, it would be very helpful for me to implement the zoning.

Lynne

I've been working on python code last week, so I didn't use Darkstar's
functions nor meet any problems.

John

I have not really had any problems with Darkstar this past week. However, I did spend all of the past week writing generic MMOG code that virtually every MMOG will need: client managed and controlled channels, and zoning (node based). Both of which went well and did not have any complications with the server.

Akash

For the past 2 weeks I looked at the learning process of Darkstar. I went thru the manuals, website links and documentation to try and find out the details of the learning curve involved while using Darkstar. This was aimed at:

1. Getting a feel of what all a user has to go thru to learn Darkstar from scratch.
2. Finding a few things which might help make it better.

Following are the things that I noticed

A. Website:

What it does good:

The website is pretty detailed with the information provided which is a good thing because it is always easy for a user to just find it in one place. IF a programmer views the website he will understand exactly what Darkstar is.

What it can do better:

The website assumes that all the people who are viewing the site are programmers or have a sufficient enough programming knowledge. Also the website does not specifically mention any prerequisites that one has to know to start using Darkstar. I feel that if a Designer/Producer (or any NON-programmer) is looking at the website he can get an idea about Darkstar but will need a programmer's opinion to drive the point home. If the website can offer some way of making this category of people understand by using terms that they are familiar with then it might reach out to more people. (like how it can affect schedule, profits, hardware costs, etc.)

B. Darkstar Server and Client Tutorial Manuals

What it does good:

The server manual is pretty descriptive and in-depth as per the functioning of PDS. The information is backed up with diagrams so that it is easy to understand. The server manual covers all the basic details that a programmer needs to know to use the PDS. The best practices section shows exactly the things one should or should avoid, which is a handy.

What it can do better:

Again there are no prerequisites listed at the start of the manual, so the user has no idea if he possesses sufficient enough knowledge to be able to understand it. Providing this will enable the users to judge themselves what all they need to know to be able to use PDS easily. The language used is pretty much programmer centric and is perfect for advanced programmers, but might prove a bit hard for users with basic/limited programming knowledge who want to learn Darkstar.

C. Hands on Lab

What it does good:

This was a pretty good tutorial exercise. It mentions all the necessary prerequisites, requirements and notations used. It also has a list of additional resources.

What it can do better:

While doing the exercise alt tabbing between programs and the tutorial window is a bit troublesome. To avoid this, is it possible to have the lab exercise available as a PDF so that it is easier to just print it all at once and have it like a handbook on the side while doing the exercise.

D. Documentation (Darkstar Java Doc)

What it does good:

All the classes used are listed with their method summary and method detail. If a programmer refers to this most of his problems will be solved.

What it can do better:

There is no search function for searching specifics.

The easiest and user friendly Programming Documentation that I have come across is the Unity 3D scripting reference. It has the most easy to understand language and proper linking structure to show the relationships between different functions, variables etc. It is not as in depth as the Darkstar Java doc, but then sometimes a user just wants to clear a small specific doubt or find out why a certain error has occurred. And it is nice to not have to cycle through a heavily detailed explanation. In such cases the simple approach that Unity has is very effective time wise.

Time span of the basic darkstar learning process:

- Learning what Darkstar is - 1-2 hrs
- Going through the Manuals - 4-5 days

  Server Tutorial (67 pages)

  Client Tutorial (21 pages)

- Hands on Lab (Snowman World) - 1-2 days

  (It originally is a 2 hour workshop but in the absence of a tutor takes more time)

- Learning the Client API - Differs for different Client types

  Less for users using Java Client (based on community posts)

  Moderate for users using C/C++ (based on community posts)

- 5 days to a week for users using Python (Panda3D) (From team’s experience)

Conclusions:

- If a person wants to learn darkstar he has all the necessary resources and links available at one point. (the website)
- The learning curve is somewhat steep considering the available tutorials and time spent.
- The learning process takes about 1 or 2 weeks depending on what the client technology is.
- The steep learning curve makes it a bit difficult for novice users to just get darkstar and try it out for short term rapid prototyping.
- If there can be a way of reducing the learning time then more people will use it for short term projects and prototyping.
- But for a long term project the time spent learning is very efficient in comparison.