Make things in game like Surrender and such be TRANSMITTED only!
Then have the server send it as a mass message to everyone and it executes it within the game… taking care of your ill-fated units

LogNotes:

12/3/97 12:21:19 PM
I'm currently getting errors when I run the object editor and access the Sprite property of the test object…
That's because there is no sprite definition that the test object is trying to reference.
I must build a sprite editor, and then have it edit something like "Sprites.Dat",
So I can pack it into the main Ion Pack file. Then it'll not have that problem.
I also need to add some error handler in the Object editor to handle that damned thing.
Options:
I can set it to restore the Sprite property to "Null" and have nothing there,
But that'll upset the user cuz he/she won't be able to know their sprite link is gone…
Ummm
Other option…
I can have it bring up a list of available sprites (plus a "Null" option so if he doesn't want any sprites from the list he can choose the null option and go restore the sprite and doesn't have to bother with messy sprites.. blah blah..

12/4/97 7:52:23 PM
I need to change the JPI format inside JPI so that it reads the picture definition like this:
GraphicsLibrary:
PicName:
X:
Y:

12/4/97 9:17:36 PM
Started on the sprite/picture editor, and I bet I'll be done by tonight.
Hrmm..
Added Nathan Fisher to the team and I think the team will consist of him and me.
(his screenname is Phoenix54)

I gotta change the story so it continues from War3.
That's all folks.

12/7/97 11:33:25 AM
Got the sprite/picture editor done, completely done.
Incorporated Music into the game!
I got INTERNET FUNCTIONS ALL COMPLETED AND WORKING FLAWLESSLYYYYY!!!
WAHOOOOO!
As in all the previous logs.. I'm dead tired.
Yes.
AYE.
And I miss Brigitte too.
Hmph
I made the JPI server version 1.0
Nothing really there except the ability to log on and log off.

Added already-in-use nick rejection to the server and the respective handling code to the client

File Types:

- *.Dat

Game data files. To be later compiled with the IonPacker

- *.Ion

The game file. Each episode points to a *.Ion definition file. It loads graphics and units specific to the level.
Default is "Original.Ion"

- *.Jps

(Jump Point Script)
Is a game script that dictates how the game will run, what episodes to load and suchlike.

-
Lines marked with
'Optimize
Can be commented out for the final compilation
because they are either currently unneeded or there for purely logic purposes.

Lines marked with
'ADDTHIS:
are followed by descriptions of possible improvements of the game.

Remember that the LengthX, LengthY, and LengthZ are all multiplied by Two within the game.
BECAUSE…
It takes that and also takes the negative of it, to save CPU time. Like this:

Lets say the LengthY = 1. Then it would be like this:

BorderY
|
Actual ObjectY
|
BorderY