

MAPS July 2003 Summary

This month Stefan introduced to me the ‘Thermos’ line of thought – ‘How do it do?’ Based on a joke about a French, English and Polish scientist, the Thermos line of thought is to think about how something knows what to do (and more subtly where it looks for information). When we ran into huge problems of not being able to save to our networked database, but were able to save locally, Stefan suggested an experiment. So we copied some pictures and sounds to the machine where the server was located, then tried to run our application. When it worked fine where before it had crashed, we knew we had found the problem.

Previously, the program had been writing NULL or nothing to the database when it should have been writing blobs for the pictures and sounds. After the copy, it was writing real information. So we decided that in order to fully solve this problem, we’ll have to learn something about Proxy servers, and we’ll be talking to Sybase soon.

Image experiment

To move forward on our research with images, we went on a hunt for published research already on the subject. We found a few useful things, and are still collecting more sources from the library. In particular, Stefan had me read a chapter from a book called Visual Perception that is a collection of articles on that topic. Stefan said the book itself wasn’t specifically what we’re looking for, but had me read a useful chapter about extracting information from journal articles that was at the end.

The first couple of weeks of next month I will be collecting and editing pictures for our image experiment. The basis of our testing (as I understand it) is to see what context (or no context) makes items in images the most recognizable. We will determine this by how fast someone can tell what they’re looking at and also correctness.

In the preliminary stage of taking pictures, we already expounded some subtle things that need to be addressed, such as how much context is provided. In the example of a pair of scissors in a drawer, if the tape is fully viewable also, it is hard to tell if they should find the scissors or the tape. In any case, the picture must make it clear what item we want them to find. This can be done by cropping the picture so that the only whole item is the one we want to find. After I’ve taken some good pictures, I’ll be putting them on a palm device and testing some of my co-workers, just to make sure it is as obvious as we want.

Debugging, gui, code stuff

This month I spent a decent amount of time working on the gui. There was a problem with the scroll bar on the Picture Select Display that I fixed by adjusting how it was calculating the max value for the scroll bar.

I also did a lot of code reduction to reduce repetition. I was able to chop a repeating piece of code in refreshPicSelectDisplay and just have it call displayNodesPSD.

Next I made it so that only pictures that are the right size (and can therefore be stored in the database) are displayed in the pic select display. I wrote a function called check file size that checks if the file is too big. Only ones that are the right size will be displayed.

I went on to add some functionalities such as beginner/advanced user preferences. Right now the only difference between beginner and advanced is that when you hold the mouse over the picture in the picture select display in advanced mode, the tool tips tell you the name of the file, whereas in beginner mode, they tell you what double and single clicking does.

I also added some debug stuff, a var called Debuggy, so when it is set to true, all the debug info will print. I have another variable called OldDebug, so that I can keep some debugging stuff that I'm not using to debug, although it can all probably be deleted.

The next big thing I went to do in the code was to implement Auto-save. There are three auto-save options in the preferences: Off, save after # of prompts, save after # of minutes. The user can choose how many prompts or minutes they want to save after. The default for prompts is 5 and the default for minutes is 2. To implement the save after number of prompts, I check how many prompts are in the script each time after a node is added or deleted. To implement saving after a certain number of minutes, I have a timer run while the script is being edited. Since the timer control in VB only goes up to about a minute, I have a counter that counts up the number of minutes before asking to save. This timer is enabled with a node is added. It is disabled when the person runs the script in the script player. It is then re-enabled. The counter is reset after the save (or non-save if they didn't want to save).

While adding options in the Preferences form applying to the user level and auto save info, I ran into a problem with the radio buttons on the properties form. When the form opened up, they would not be set. Even in design-time I was trying to set some of their values to true, and it would change itself back to false. I discovered that for some reason VB was considering all of the radio buttons that were not contained in a frame on the SStabs to be in one group. Since the Preferences form has a SStab control on it, and each tab has a few options, it was grouping them. The ones that didn't seem to be affected by this problem were ones that I had grouped into a frame on that particular tab. So I added frames to all of the tabs and the problem was solved.

So our database problems have come to a head. First of all, one July 15 (despite four power failures interrupting our work) we discovered that "noodles" is a keyword of Sybase. When a picture file started with the word 'noodles', the thing would crash when we tried to save. But everything worked fine if we started it with anything else ('foodles') or when noodles was in the middle of the filename ('bowl of noodles'). After some more testing, I discovered that the file name cannot begin with an n. It's interesting because there's a file called "not_this_bus" in the bus pictures, and it never seemed to cause this crashing problem before (but it does now). The error that is received from sybase says: "[Sybase][ODBC Driver][Adaptive Server Anywhere]Integrity constraint violation: Column 'Image' in table 'ScriptNode' cannot be NULL" This is the same error we are getting when the network database crashes. (At least until we changed the database to allow nulls, then the error for the networking occurs when you try to load a NULL). This is not just for pictures either. A prompt file that begins with an 'n' or 'noodles' also crashes. Since .wav's are allowed to be null in the db, it doesn't crash until load time with the type mismatch found. Second, it turns out that we're going to have to find out about a Proxy server because we cannot write to the NetworkMAPS database unless the files it wants it can find on it's own machine.

This is what we did to find this out. First of all, it was crashing when saving or loading because it couldn't read the file and was writing NULL to the database. Second, we tried copying a couple pictures and sounds onto the machine where the server lives. Then we tried running it again, and it worked fine. Which says to us that we need to have those files on the server's machine. Or in other words, the xp_read_file function or whatever is passing the instructions off to the server, where it is then trying to find and read the files, rather than passing the actual blobs of file info to the server. So Stef is going to help us get in contact with Sybase to figure out Proxy server stuff. For now, we're leaving this problem alone.

Unfortunately, not all the time spent on the gui this month was productive. In an attempt to fix the double blank problem, to no avail, some time was wasted.

After a lot of messing around, I thought I'd fixed the problem when I made it not display the second blank node in display nodes, the only problem is that it would still be there later.

I thought the problem was in counting that second blank node, but when you mess with the TotalScriptNodesZIndex, you end up one blank node short sometimes. I'm not sure if there is a solution without at some point having two blank nodes.

My motivation for trying to fix this problem was for the displaying of partially filled nodes in the whole script view. I think that some playing with the display of that will have to be done. In the end I decided that without re-writing a lot of code (and that may not even do it!) I don't want to attack this problem and I really don't think it is that much of a problem altogether.

Another thing I did was to change the whole script view to display nodes in the middle of the script as black, those that have only a sound as a red box, and those that have only a picture are now outlined in red in the whole script view.

Some more things should probably be deemed beginner and advanced user functions, but I don't want to take any functionality away from the beginner, so for now...