

MAPS November-December 2003 Summary

These two months were spent working on the GUI and stuff related to scripts. To begin with, I manually entered into a spreadsheet a bunch of different scripts. These were taken from a packet of tasks that had been broken down already. Each step of the task would be a prompt. Essentially we created templates for scripts. All that remains to be done is to record the sounds and take the appropriate pictures for these. Next I will be inserting these templates into the Network MAPS database.

The rest of my time was spent working on the GUI. There were a number of tasks that I set out to do. 1) Fully comment the code. Since I will be graduating in May and leaving MAPS ☹, someone will need to be able to make sense of my code, and Stefan in particular was becoming concerned that I knew the code so well, and if someone else tried to work on it, they would be lost. So I went through, updated all of the functions comment stubs at the beginning (including 'where called from' info, and 'what it does'). I then went through all of the code and added comments to almost every line, at least every segment, describing what was going on. This was a bit mind-numbing. I also broke up some code into smaller functions, particularly in the DB module.

2) Add some requested functionality – a Whole Script View with 2 images per row, thus larger images than the 3 per row that it was currently. This was a fairly easy task because I could copy most of the code from the 3/row stuff. There was some minor tweaking but ultimately it was an easy functionality to add.

3) Redesign the GUI. After getting comfy with this version of the GUI, we decided that a few things could be better, and that the GUI could be sleeker. We are still in the process of this re-design, but I made some initial changes. A) I got rid of some redundancy of buttons below the nodes at the bottom, making one set of buttons that had the 'play', 'play from here', and 'delete' functionality apply to the selected node, rather than having a set of buttons for each node. B) I moved the drive and directory lists to a dialog box and added browse buttons for the picture and sounds thus reducing the amount of stuff on the form. This also allows the list of picture sound names to be larger. There is a box that appears displaying the current path to those lists on the forms, which I may move to the dialog box later. C) I add a new clock button that the user can click on and it will adjust the time step for the selected prompt. This button appear next to the buttons described in A). The time step for each node is still displayed below that node. D) I made some slight adjustments to the arrangement of the form, mostly to best use the extra space that opened up from removing the directory and drive boxes. It probably still needs to be reorganized to look better. E) I made it so all of the fonts of everything on the forms was controlled by a global variable that can be changed. There is also one for the size of fonts on the form and the size of fonts in lists (such as the directory lists).

4) DEBUG! Wahoo, lots of fun, and by now there were lots of bugs that had popped up. The following is a list of those bugs and the fixes.

- Fixed the scrolling/refreshing of the whole script view (2/row)
- Bug with the whole script view options staying checked and thus refreshing the displays properly when new pictures or sounds are added or when nodes are deleted. This was a difficult fix. I finally found the problem in the expandForm function. It was resetting both of the checked values to false. I did not discover

this until I had set up some global variables to do the same thing as the checked property does.

- Bug if choosing a picture name from the file list, then clicking on a different picture in the Picture Select Display, then clicking to add picture to prompt. It adds the picture that was last double clicked in the Picture Select Display. It does not add the one that was last single clicked, nor the one that is highlighted in the list. This was fixed so that if you double or single click on a picture in the Picture Select Display, it selects the name of that picture in the file list, and then adds that one appropriately.
- The change time step box was showing the default time step whenever you clicked on it. It seemed more logical to show the time step of the selected node if it was different than the default, however, so I made that so.
- The time step was not being associated with the correct node when inserting a blank in between two nodes. It wasn't following the node it was associated with. This was fixed, turns out some lines in display nodes had been commented to account for some of the redesigning issues.
- The selected node was not being highlighted appropriately. Fixed.
- The 'play', 'play from here', and 'delete' buttons were not working on the apparently selected node. This was a problem of a variable not being set and was fixed. Also, inserting blank nodes using the insert lines was getting messed up and was fixed when the variable was fixed.
- Upon starting the application, the Script Name box would display 'bus to kennys', the name of a script in the database that was NOT loaded. This was fixed and a global was added to contain 'Your Script Name' as an unnamed script name.
- The whole script view 2/row was suddenly not displaying. Turns out that on the design of the main form, the picture boxes were placed inside the frames that are used as background outlines that represent when a sound is missing from a prompt. The fix was just to make the picture boxes not belong to those frames.
- If the paths to the pictures and sounds were changed, it was not being saved as a preference and it would reset to the default upon restarting the application or choosing File->New Script.
- Some fonts and font sizes were still out of whack. Those were fixed.
- The auto-save preferences were not being saved fully. It would correctly save which option (off, prompts, minutes) but not the number of prompts or minutes. This was fixed and it was also set so that you could choose both number of prompts and minutes and it would save both values, even though only one option can be chosen at a time. That way the user can input their prompt and minute preferences and choose between them, without the values of the un-chosen option being reset to the default.

There was a fairly big issue that arose and it is discussed next. It concerns appending nodes in the GUI and about blank nodes.

The issue is this:

Before recent changes, the behavior of the GUI was that as you append nodes, there would always be a blank node at the end that could be clicked to switch you into append mode. Sometimes, in special cases noted in the debug notes, there would be two blank nodes at the end (right after completing a node, i.e.

making that node contain both a picture and sound). If the user were to click themselves into edit mode, then wish to return to append mode, I changed it so that append would switch to the first blank node at the end. Later I changed it to delete any other extraneous blank nodes at the end. This led me to the question and some issues.

What began to lead me to the problem was undo. Basically if you added a couple of nodes, went into edit mode and then switched back to append, some blank nodes would be deleted (this was the functionality I added). Then if you did undo, nothing would happen at first (what was happening was that it was undoing stuff with the deleted extra blank nodes).

Next I thought, So how about we just don't let the user insert extra blank nodes at the end? Well then they can't insert blank nodes as placeholders, and this doesn't seem right either.

So how about not deleting the extra blank nodes? Now if you insert a bunch of blank nodes, then go into edit mode, and switch back to append by clicking on the last node, it will highlight the first blank node at the end, rather than the one clicked, this isn't intuitive either.

My best solution keeping everything else the same, is to only select the first blank node at the end when the user manually switches to append mode (either by choosing the menu item or the shortcut (Ctrl-A)). Then if they just click on the last blank node, it will go into append mode as usual. But then there would be no way to tell if they had clicked on the last node wanting to go into append mode or edit mode, and thus this won't work either.

So it seems that we may have to return to the original method, which has a few extra blank nodes possibly slipping in. It's not a huge problem because the user can delete them (if the nodes are in the middle, or extra at the end of the script).

Another alternative would be to do away with the 'clicking on the blank node at the end to go to append' thing and get rid of extra blank nodes at the end.

Alternatively, there could be a button on the form at the end that says something to the effect of 'go into append' or 'add to the end of the script'.

Finally I came up with a few other features/functionality that would be nice. 1) Adding a checkbox to various places in the preferences and dialog boxes that would allow the user to save their input value as the default value for that option (such as time step). They could also perhaps change the default values in the preferences form itself.