Coordination of Sound and Commands

Matthew Frohliger

Project Description: Design a file format that will coordinate commands with a section of sound

General Requirements:

- Develop a set of commands for at least two applications
- The system sychronizes the sound with the commands
- Do you need a command language?
- Write an editor that will create, play, save, open, close, fast F, reverse, search, other...

Solutions

Created an application (SoundSync) providing an easy to use interface to sychronize an audio track with a series of commands.

- Allows users to use any mp3 file (with a constant bitrate)
- Integrates with any windows applications, using standard Windows messages
- Windows message numbers need to be known to send commands to the applications (Can be easily found with any message sniffing application such as Spy++ or Winspector Spy)
- Users can seek, play, pause and zoom to allow for easy and accurate editing
- The interface is based on popular music editing software such as Cakewalk and ProTools

Methodology

- Decided to use standard Windows messages for "commands"
  - Allows for easy and reliable communication
  - Has no added restrictions on what can be manipulated (if you can program it, it can be manipulated by SoundSync)

- Created a test console application (including a class) that can open and send messages to any windows application

- Created a test console application (including a class) that uses the Microsoft Mulitmedia Control Interface to play, pause and seek in an audio track

Methodology

1. Started development on the main SoundSync application taking special care to allow for easy extensions later on by using constants instead of numbers, and being sure to use classes and functions properly (modular)

2. Began adding features to SoundSync to make it user friendly and functional

3. Created GadgetLights, an application that allows the user to control the Gadget Master board using the ntport library

4. Created the outlet box, allowing the gadget board to control electrical flow to each outlet individually

Demonstrations

Strategies

- Start Early
- Design early and thoroughly
- Start small and make sure everything will work before starting anything big
- Grow the application (or even better, let it grow itself)
- Allow for growth
- Think about the user

Knowledge

- Events!!! - Skills developed during CS350 and on my own
  - Interface building, SDI and message handling within an application
  - Sending messages from one application to another
- Operating Systems
  - A firm understanding on how event based operating systems communicate with applications

Extensions

- Improved UI
  - Allow Copy-Paste
  - Reduce Flicker
  - Improve seaches and sorts
  - Change multiselect to be more natural
  - Add message sniffer to allow users to be able to add commands without any knowledge of Windows messages
  - Visual representation of audio
- More support

Advice

- Start Early
- Design...Design...Design...
- Start small, and make sure your designs work (by thinking them through on all levels as well as writing small "proof of concept" applications) before putting any significant time into any coding
- Make all designs and code modular (think objects)

Q & A

More Demonstrations...