Media Management

Michael LeGore

TJHSST Computer Systems Lab

Abstract

The goal of this project is to develop an application to control several types of media during a live performance. Also a goal is to create a Graphical User Interface (GUI) that will allow a user to create his or her own performances easily and with the ability to customize the application.

Background

In a live performance there are many inputs and outputs to keep track of. Musical scores, lighting, cameras, and other media. For humans to coordinate all of these elements in a live performance, it often can involve tens to hundreds of people. If there could be a way of keeping track of all of these medias using a computer, it would allow the common man to create his own professional level performance. The goal of this project is to start work on such a program that would allow this to be possible. And while completion of this project would be impossible given that I am a single person and I have only a year to work on it, I still have been able to demonstrate several key priniciples that allow for this program to be possible.

Development

For any performance there are often many forms of media that can be involved, but those can often be condensed down to several base types of media. The base types of media can be linked together by the user to either form the behaviors needed for a performance or to create derived form of media.

Sound

The playback of sound is one of the main building blocks of a multimedia presentation, being able to play music or sound clips in 2D and 3D space allows the user to coordinate sound with other media, either by beat tracking or other means. In a musical presentation it is also useful to be able to apply effects to music or to take inputs from microphones or audio devices.

Video

Video playback and visualization helps make up the repertoire of a engaging sensory performance. Video can also be extended into the playback of prerecorded video files or to inputs live from cameras.

Hardware

Hardware is the most vague of the types of media, it can be anything that would involve the computer sending or receiving commands to electronic devices outside of the computer. For example, this would include controlling the luminosity and direction of a spotlight, or telling a robot to move forward. As you might imagine, this could have applications for programs outside of live presentations. To achieve this I have devised a simple API for “talking” with hardware devices using a textual interface. The API allows the hardware device to identify its methods, and to identify what type of device it is, allowing for a hardware device to mask itself as another class.

Joining them all together

Using all of the basic forms of media we can combine several to make a new type or behavior. We can achieve this by making each type of media have access to the other forms of media through some sort of interface. If we then use a scripting language to access the methods given by each form of media. This would not only allow for flexibility for the programmer, but also for the end user in that they would be able to customize the application to their needs.

Implementation

To implement this system, I decided to create a resource manager in C++ that keeps track of all of the references to the resources. This class is the gateway for all classes to interact with other resources and to facilitate the communication between the same.

The implementation also requires that each resource give access to its methods using a lookup table. This is needed so the a resource can call the methods of an object after compile time, without needing the compiler to resolve pointers. Instead the resource manager keeps track of function pointers in a map that corresponds string names of the methods to their function pointers.

Current State

Currently, I have a GUI that can play back sound clips that are loaded by the user. These clips can be paused, stopped, rewound, or skipped. I am working on allowing the gui to send commands to the hardware, but I have not finished that branch. I also have a Resource Manager, as described above, that handles the creation and interfacing with each of the media resources. This resource manager handles the memory allocation and factory methods of the Resources.

Conclusions

My project needs to be defined better before I go on. I have coded the logical equivalents of Lego blocks, and I now need to construct the actual Lego sculpture. This previous quarter I have started upon this goal, and the third quarter will be focused around that goal.

References

Live Performance Tools, Stefan Müller Arisona, ETH Zurich

Real Time Beat-Synchronous Audio Effects

Live Performance Visuals– Pascal Mueller, ETH Zurich

Corebounce Software - http://www.corebounce.org/wiki/Software/

Max5 MSP - http://www.cycling74.com/story/2007/10/5/91222/9559