Konzept Soundprogramm
fhat: Fachhochschul Audio Toolkit

C. Blumer, P. Brägger, J. Fischer, K. Gabathuler,
L. Gröner, P. Hodel, D. Oriet,
A. Reeder, A. Sacchi, S. Tallowitz

14. April 2005

Inhaltsverzeichnis

1 Gesamtübersicht

1.1 Plugin-Kette

Das Programm basiert auf der Idee, dass man eine Plugin-Kette aufbaut. Ausgangspunkt ist Portaudio, das einen Aufruf auf das letzte Plugin in der Kette macht. Dieses Plugin holt sich seine Samples beim vorletzten Plugin, usw. Vorgesehen ist auch Multiplexing, was in der Abbildung 1 aber nicht dargestellt ist.

1.2 Klassendiagramm

Siehe Abbildung 2

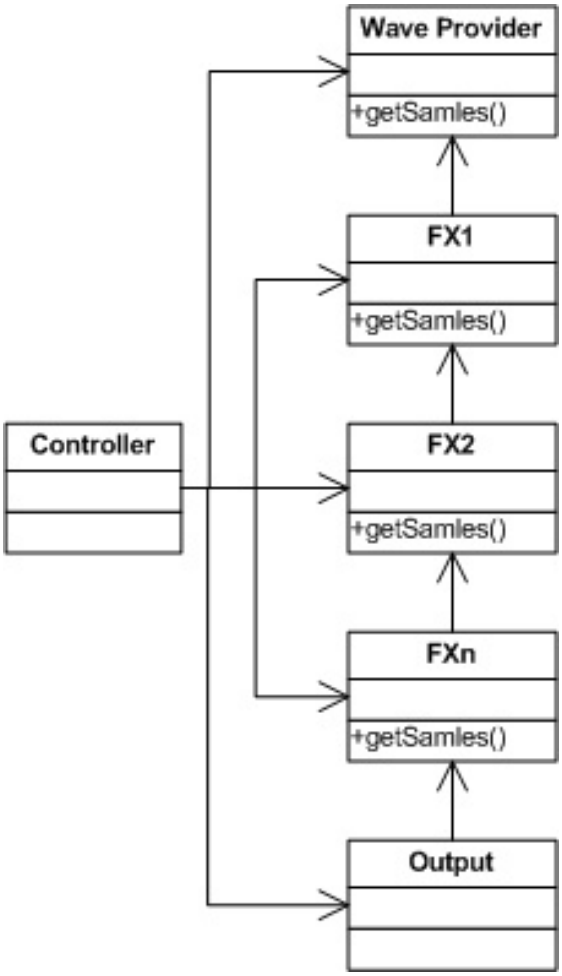

Abbildung 1: Aufrufe gehen durch die ganze Plugin-Kette bis zur Quelle

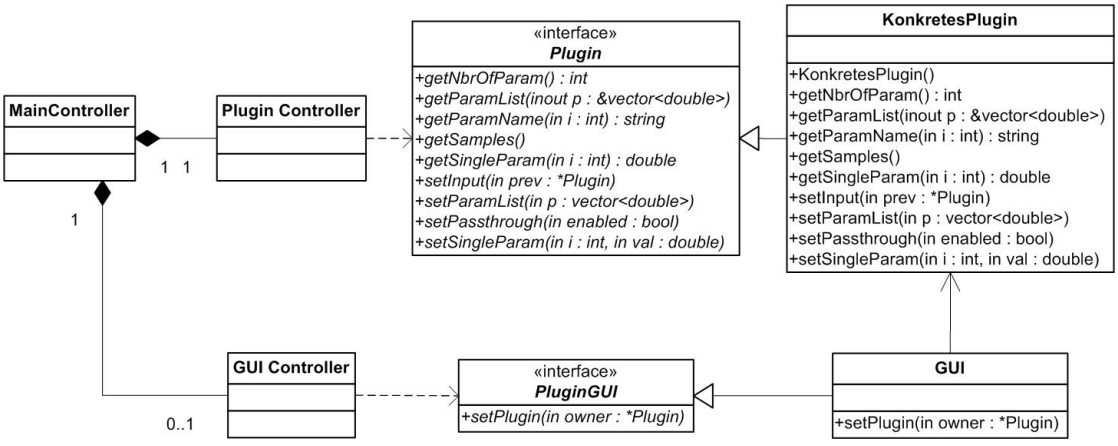

Abbildung 2: Das Klassendiagramm des Programms

1.3 Objektdiagramm

Siehe Abbildung 3

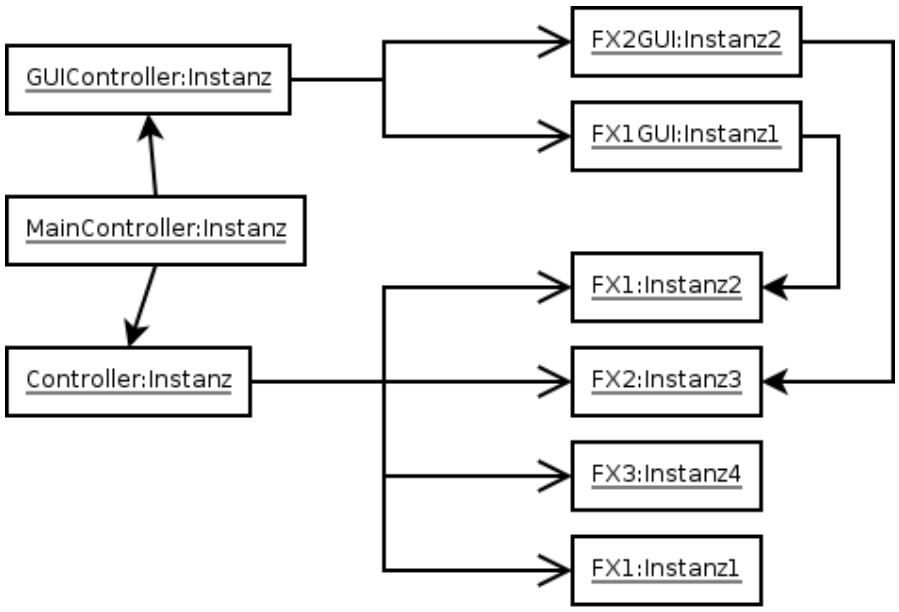

Abbildung 3: Ein Beispiel der Objekte zur Laufzeit. GUI-Objekte müssen nicht zwingend instanziert werden. In diesem Beispiel laufen zwei Geräte ohne GUI.

2 Die Plugin- und GUI-Klassen

2.1 Plugin

Die Basisklasse aller Effekte stellt die Klasse Plugin dar. Sound Generatoren (Synthi, waveReader) werden auch als Plugins betrachtet. Alle von dieser Schnittestelle abgeleiteten Klassen müssen folgende Methoden implementieren:

getSamples(int n) Mit getSamples(int n) kann jedes Plugin neue Samples beim vorhergehenden Plugin anfordern. Der integere Parameter n definiert die Anzahl der benötigten Samples.

setInput(Plugin * p) Mit setInput(Plugin * p) wird dem Plugin seinen Vorgänger mitgegeben.

setPasstrough() Mit setPasstrough() kann der Effekt umgangen werden.

2.2 GUI

Jedes GUI wird von einem MainWidget instanziert. Bei der Instanzierung muss dem GUI das zu steuernde Plugin mitgeteilt werden. Die Basisklasse GUI erzwingt folgende

Methode:

GUI(Plugin * plugin) Beim instanzieren eines GUI's muss demselben ein Zeiger auf das Plugin übergeben werden.

setPlugin(Plugin *p) Die setPlugin Funktion ermöglicht das austauschen von Plugins.

3 Die Controller

3.1 Funktionalität des PluginControllers

Der PluginController kennt alle Plugins. Er führt eine Liste mit den nötigen Informationen. Er ist auch wesentlich für das Abspielen, da er weiss welches Plugin welchem folgt.

Das ansprechen der Plugins kann über den Controller erfolgen. Der PluginController kümmert sich aber nur auf der logischen Ebene um die Plugins. Auf der GUI-Ebene hat er sein Gegenstück namens GUIController.

3.2 Funktionalität des GUIControllers

Der GUIController ist für die Verbindung der Logik mit GUI verantwortlich. Er instanziert bei Bedarf neue GUI Oberflächen oder ordnet bestehende den passenden Effekten zu.

4 MainGUI

Die Hauptaufgaben des MainGUI's bestehen darin, die modularen Audio-Plugins und deren Verbindungen untereinander in ansprechender Art darzustellen, dem Benutzer zu ermöglichen die Struktur dieses Netzwerkes einfach zu verändern und auch grundlegende Aufgaben wie On/Off des Processings oder Speichern und Laden von Projekten können von dieser Klasse ausgehen. Währenddem das MainGUI zur Konstruktion von Projekten praktisch unerlässlich ist, muss es zur Wiedergabe des Konstrukts nicht unbedingt eingesetzt werden.

Die grafischen Representationen der einzelnen Module unterscheiden sich lediglich in Grösse, Farbe und Beschriftung, welche abhängig vom Plugin sind. Module und deren Verbindungen können durch den User hinzugefügt und entfernt werden. Die Anzahl möglicher Outputs und Inputs eines Moduls müssen dazu ersichtlich sein. Der User kann Module auf der Konstruktionsfläche anwählen und frei verschieben und zudem zB per Doppelklick das Einstellungs-Fenster des Plugins öffnen.

Das MainGUI aggregiert die beiden Controller-Klassen names GUIController und FX-Controller. Möchte der User ein PluginFenster öffnen, sendet MainGUI einen Request an GUIController, welcher für die Zuordnung von GUI des Plugins und dessen Logik verantwortlich ist. Der Controller sendet dann einen Pointer auf das zu öffnende Widget

zurück. Werden Module auf dem grafischen Bauplan erstellt, entfernt oder Verbindungen verändert, sendet MainGUI diese Informationen an den FXController, welcher diese Veränderungen ausführt.

Das MainGUI besitzt zur Verwaltung der grafischen Objekte Zugriff auf einen Szenengraph. Dieser Szenengraph speichert grafische Eigenschaften und Position der Objekte auf der Konstruktionsfläche, wie auch einen Wert, welcher das representierte Modul eindeutig identifiziert (zB Pointer auf das Objekt). Diese ID wird für die Kommunikation mit den ControllerObjekten verwendet. Der Szenengraph ermöglicht das einfache und effiziente Zeichnen der Szene, wie auch die schnelle Identifikation von angeklickten Objekten. Wie das geschieht ist nebensächlich, jedoch ist die Verwendung eines Quadtrees wahrscheinlich vorteilhaft.