

# (株)CRI・ミドルウェア

## 映像・音声を専門としたミドルウェア開発会社

- 1990年 人工知能・CD・音声・映像技術の研究開発  
FM-Towns・メガCDなど
- 1993年 サターン用CDシステムの開発
- 1995年 サターン用ADXのリリース **50タイトル**
- 1997年 Dreamcast用ADX・Sofdecリリース **450タイトル**
- 2006年 PS2・Xbox・GC・DS・PSP用ミドルウェア **1100タイトル以上**

ゲーム機に特化したミドルウェア

## CRIWARE (プロダクト紹介)

- マルチストリームサウンドシステム 『CRI ADX』
- 高画質動画システム 『CRI Sofdec』
- サウンドオーサリングシステム 『CRI Sound Factory』
- 仮想ディスクファイルシステム 『CRI ROFS』
- ロパターンデータ作成システム 『CRI Clipper』
- 着信メロディ制作ツール 『CRI CP Sound Craft』
- 高画質・高品質動画エンコード 『CRI Movie Encode』
- 統合オーディオソリューション 『CRI Audio』**

**New  
Release!**

# 目 次

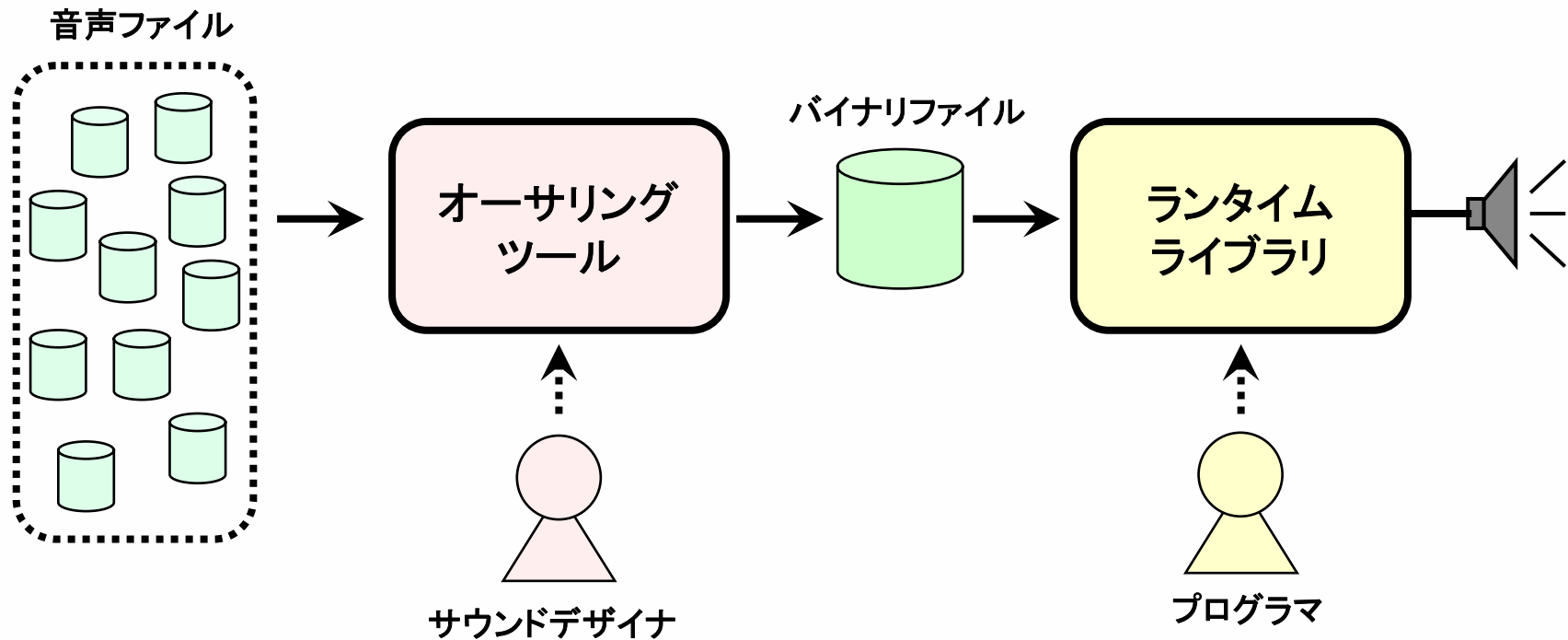
1. CRI Audioの概要
2. CRI Audio Craft の基本操作
3. プリミティブシンセサイザ
4. インタラクティブサウンドのデザイン
5. コンプレックスシンセサイザ
6. エフェクト
7. シーケンス再生
8. CRI Audioランタイムライブラリ
9. ストリーミング



# CRI Audio の概要

# CRI Audio とは

**CRI Audio** は、サウンドデザイナーが**オーサリング**したサウンドデータをプログラマが**簡単に再生**するためのミドルウェアです。



# サウンドのオーサリング

## ■ バランス調整

音量や音質のばらつきを調整したい。

## ■ 音声の加工

1つの波形データから複数の様々なサウンドを再生したい。

## ■ 音の変化のデザイン

ゲームの状況に応じてサウンドがどのように変化するかをデザインしたい。

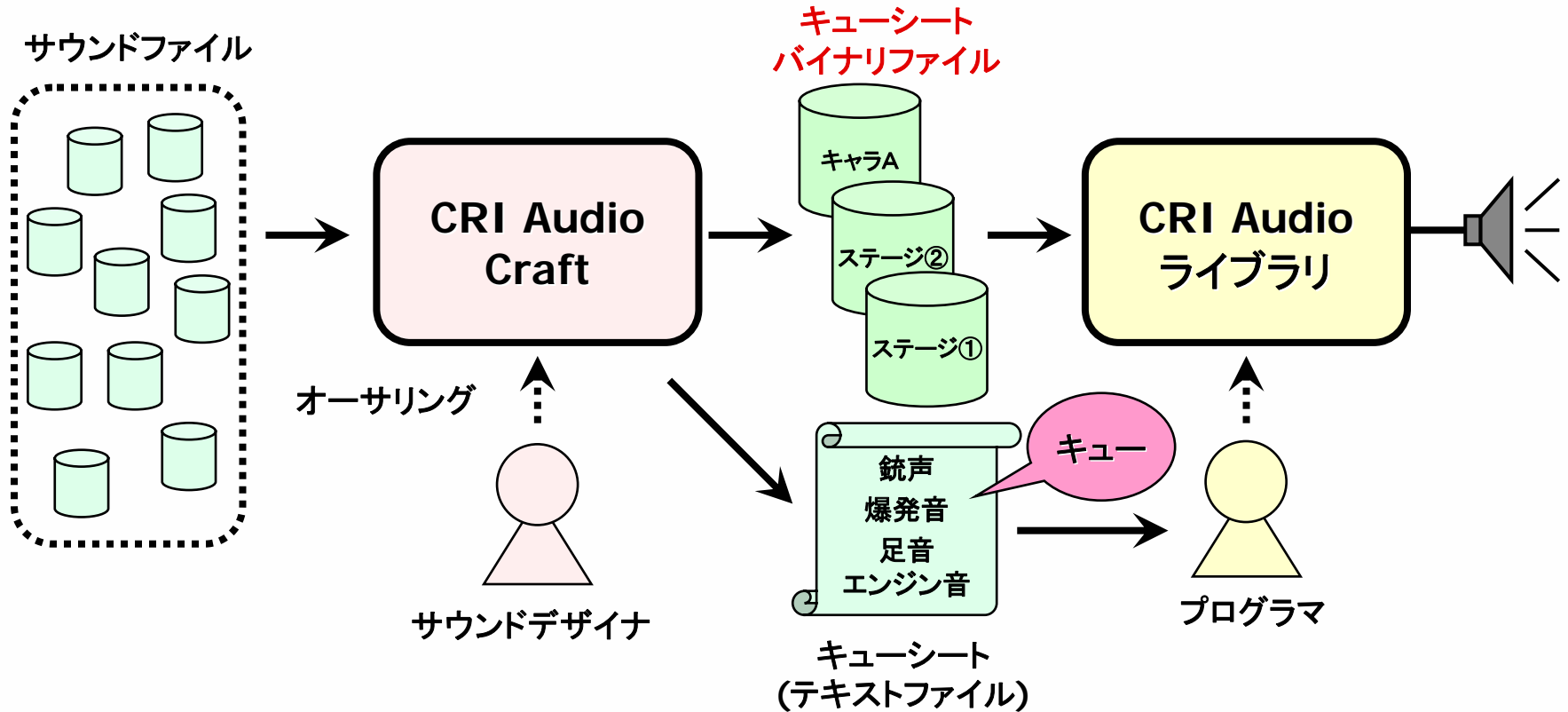
## ■ 時間軸上へのサウンドの配置

時間軸上にサウンドを配置し、シーケンス再生したい。

## ■ パッキング

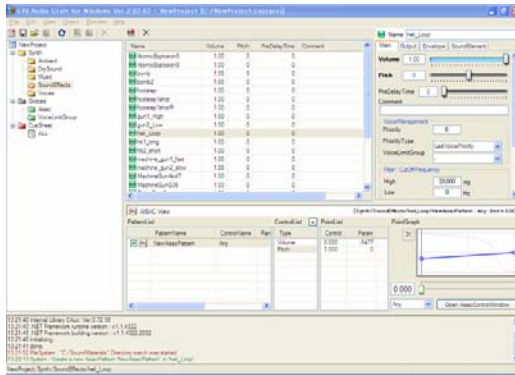
大量の音声ファイルを分類してパッキングしたい。

# ワークフロー

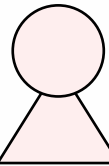


# デモ ~ ワークフロー

## CRI Audio Craft

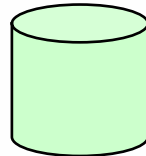


オーサリング



サウンドデザイナー

キューシート  
バイナリファイル



## ゲームプログラム

// キューシートの作成と読み込み

```
CriAuCueSheet* csht = CriAuCueSheet::Create(...);  
csht > StartLoadCueSheetBinaryFile (...);
```

// CRI Audioオブジェクトの作成

```
CriAuObj* auobj = CriAuObj::Create(..);
```

// キューシートの登録

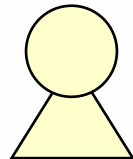
```
auobj->AttachCueSheet(csht, ...);
```

// 効果音の再生

```
auobj->Play("GUN", err);
```



ランタイムライブラリで  
簡単に再生



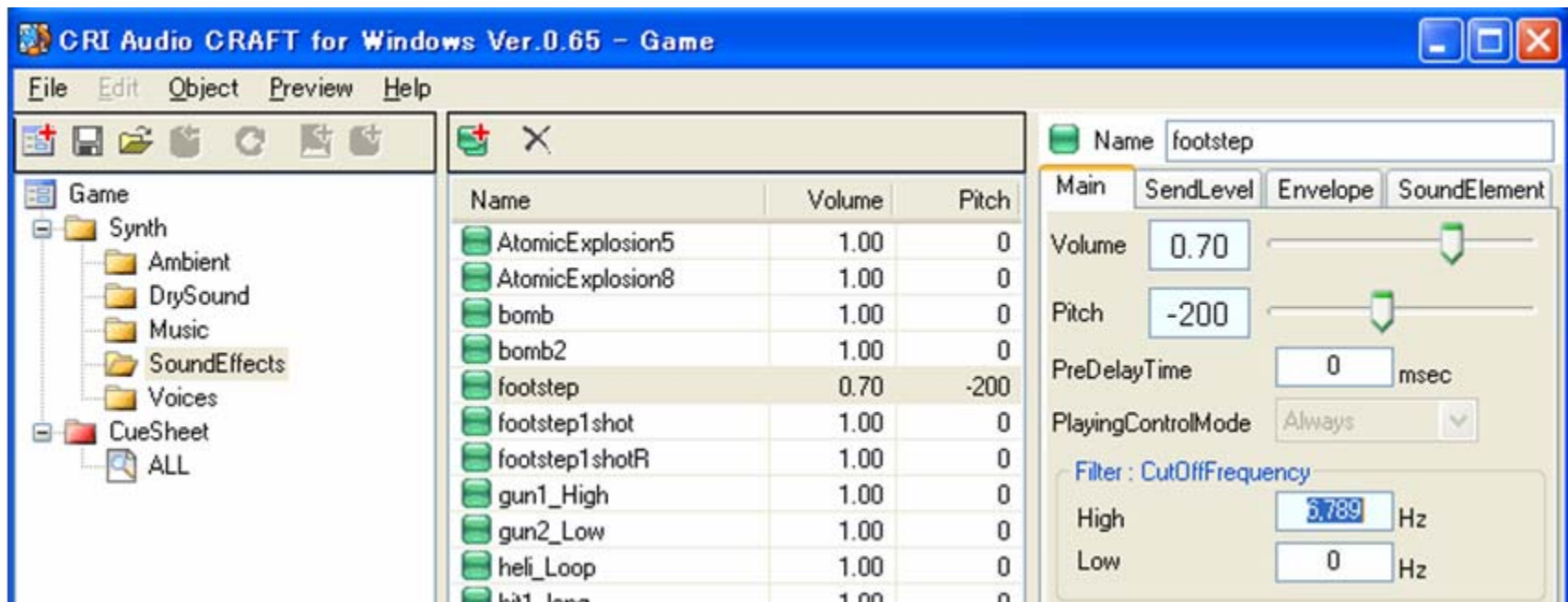
プログラマ



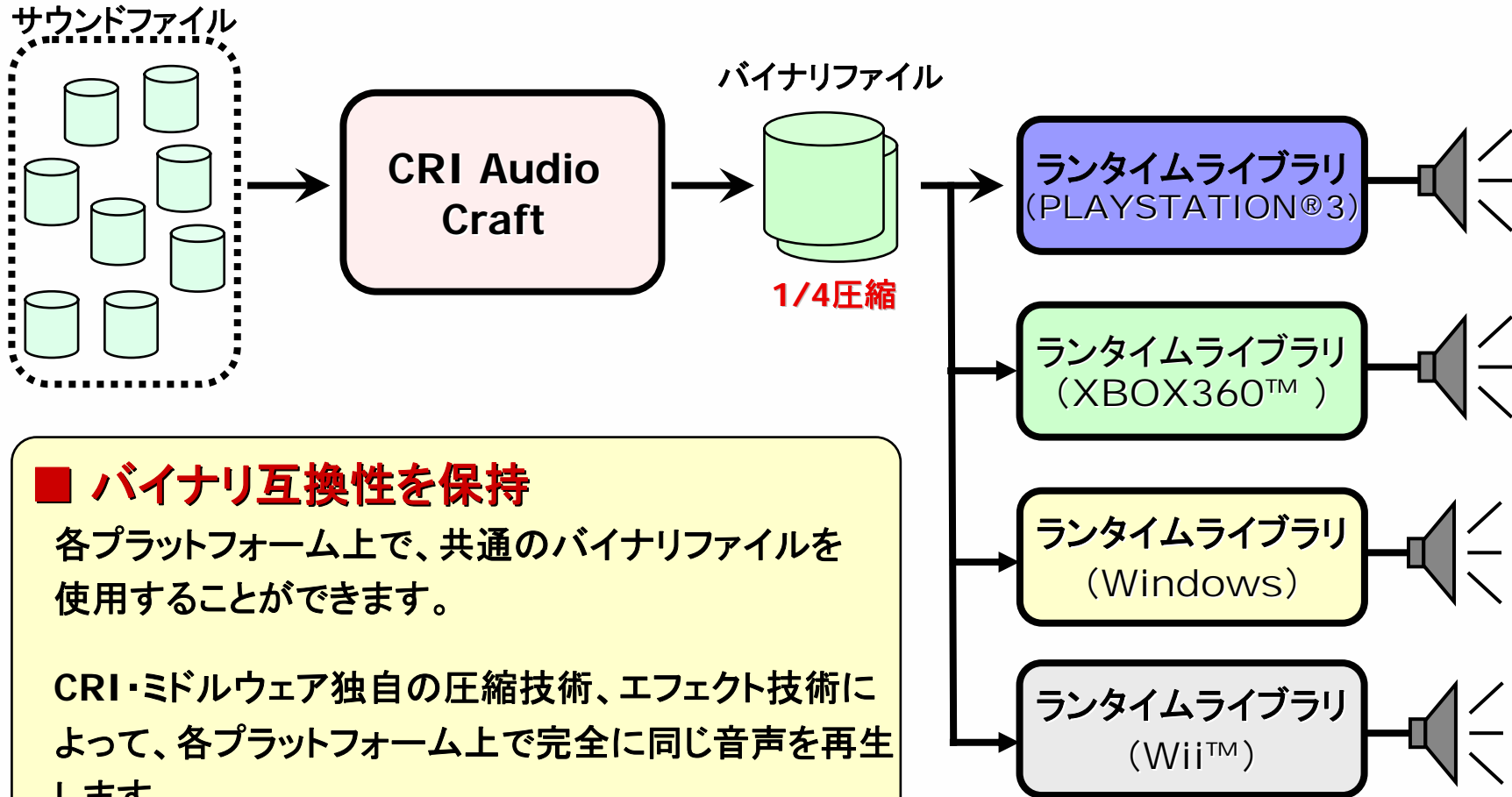
# デモ ～ 再生パラメータの調整

## ■ 膨大なサウンドデータを効率よく調整

フォルダ内にあるサウンドデータに対して、音量やピッチなどを一括設定できます。



# マルチプラットフォーム



## ■ バイナリ互換性を保持

各プラットフォーム上で、共通のバイナリファイルを使用することができます。

CRI・ミドルウェア独自の圧縮技術、エフェクト技術によって、各プラットフォーム上で完全に同じ音声を再生します。

# ADX圧縮技術

## ■ 高音質

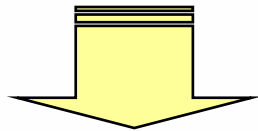
CDクオリティ、世界最高水準です。

## ■ 低負荷

1.2GHz Pentium-M で10%/64音です。

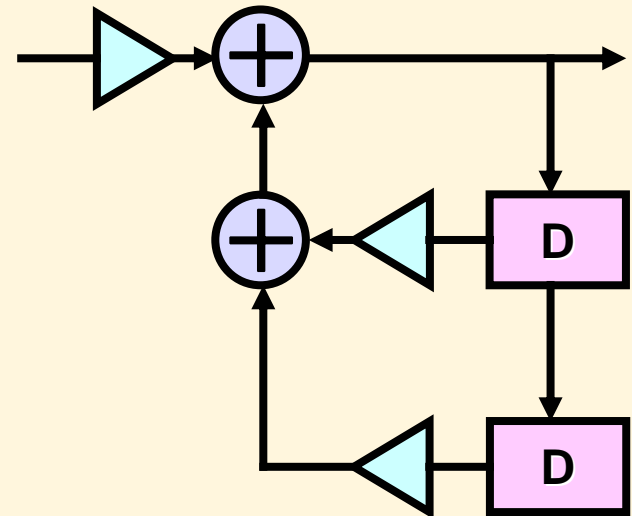
## ■ シームレス・ループ

サンプル単位でループ可能です。



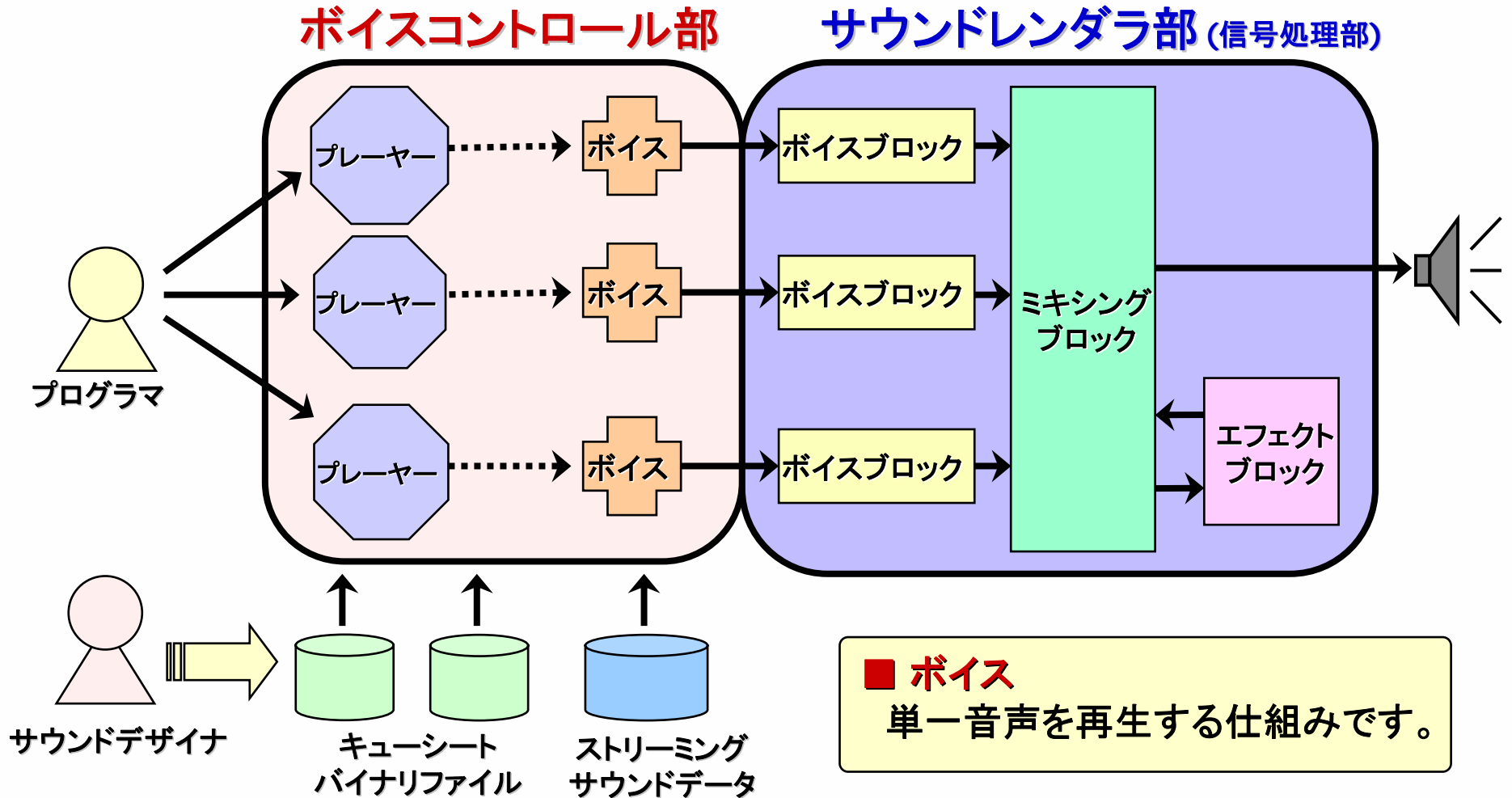
ソフトシンセサイザ  
音源

## ADXデコーダ 虎の巻



乗算3回、加算2回

# CRI Audio の仕組み



# ボイス

## ■ ボイス

単一の音声を様々な加工をしながら再生する仕組みです。

## ■ ボイスパラメータ

1. ボリューム

2. ピッチ

3. プリディレイ

4. エンベロープ

ADSR + デイレイ + ホールド

5. フィルタ

高域カットオフ周波数

低域カットオフ周波数

6. ダイレクトセンドレベル

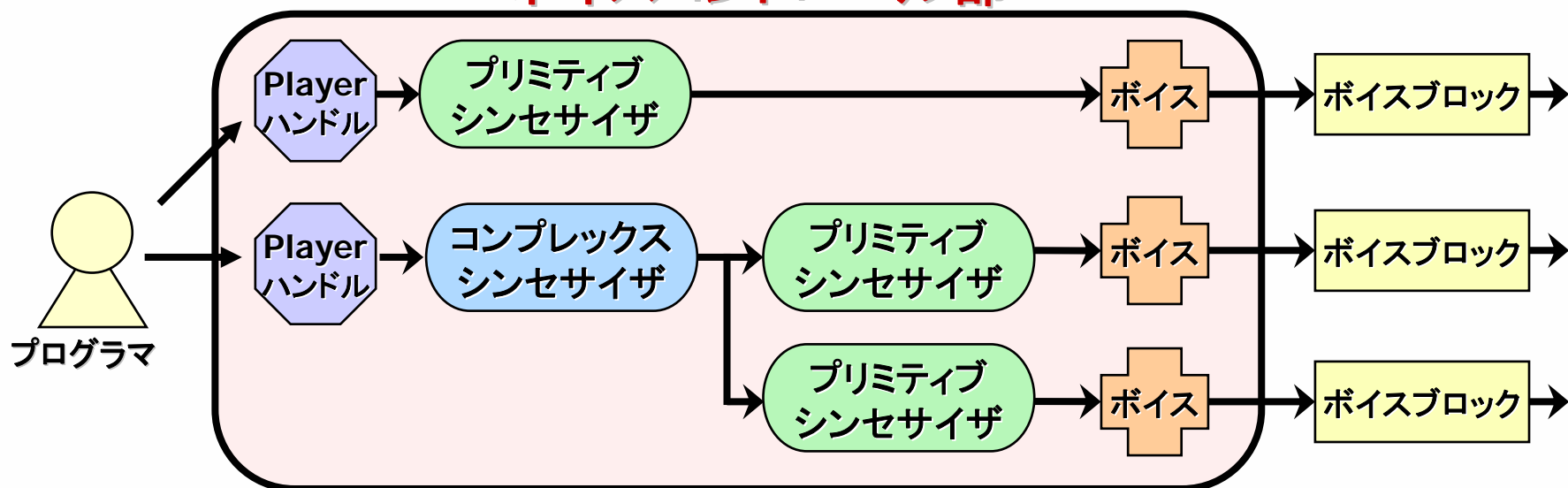
左	左サラウンド
右	右サラウンド
センタ	LFE
拡張1	拡張2

7. ウェットセンドレベル

0:リバーブ

# ボイスコントロール部

## ボイスコントロール部



### ■ ボイスによるサウンドの再生・制御

プログラマが再生関数を実行すると、各シンセサイザによってボイスパラメータが決定され、ボイスブロックから音声が出力されます。

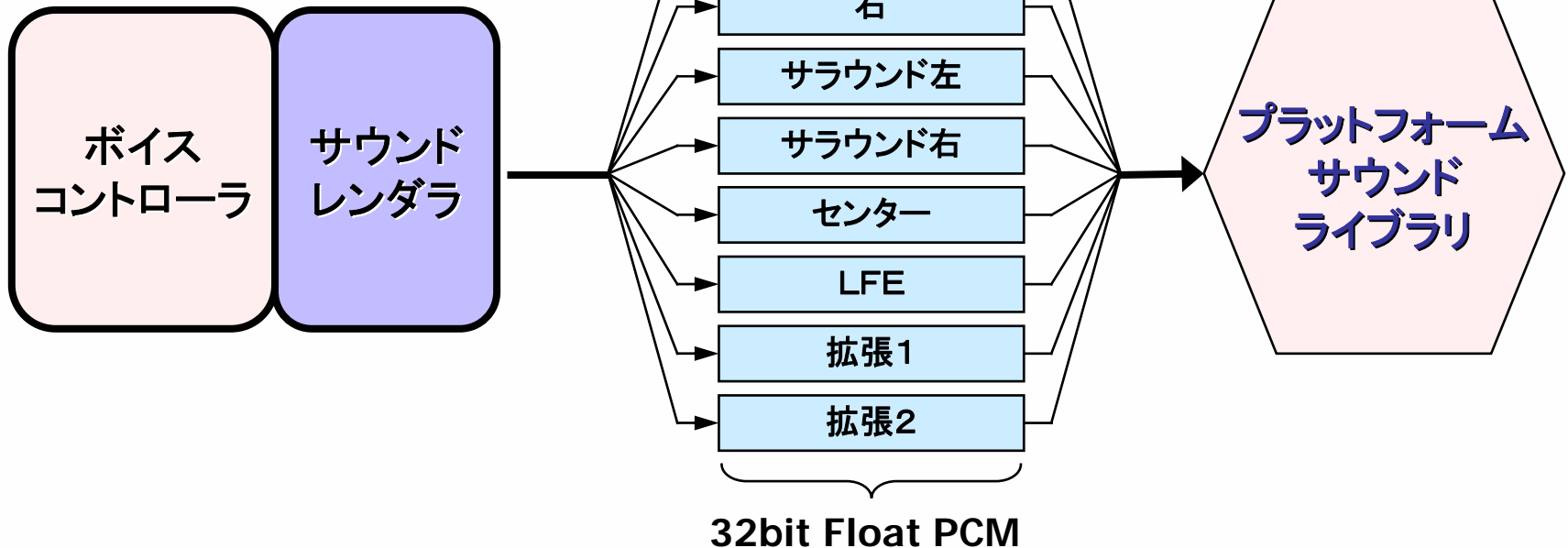
# CRI Audioからの出力

## ■ CRI Audioの出力

サウンドレンダラは単純なPCM データとして出力します。

このPCMデータを各プラットフォームのサウンドライブラリに入力します。

### CRI Audioライブラリ

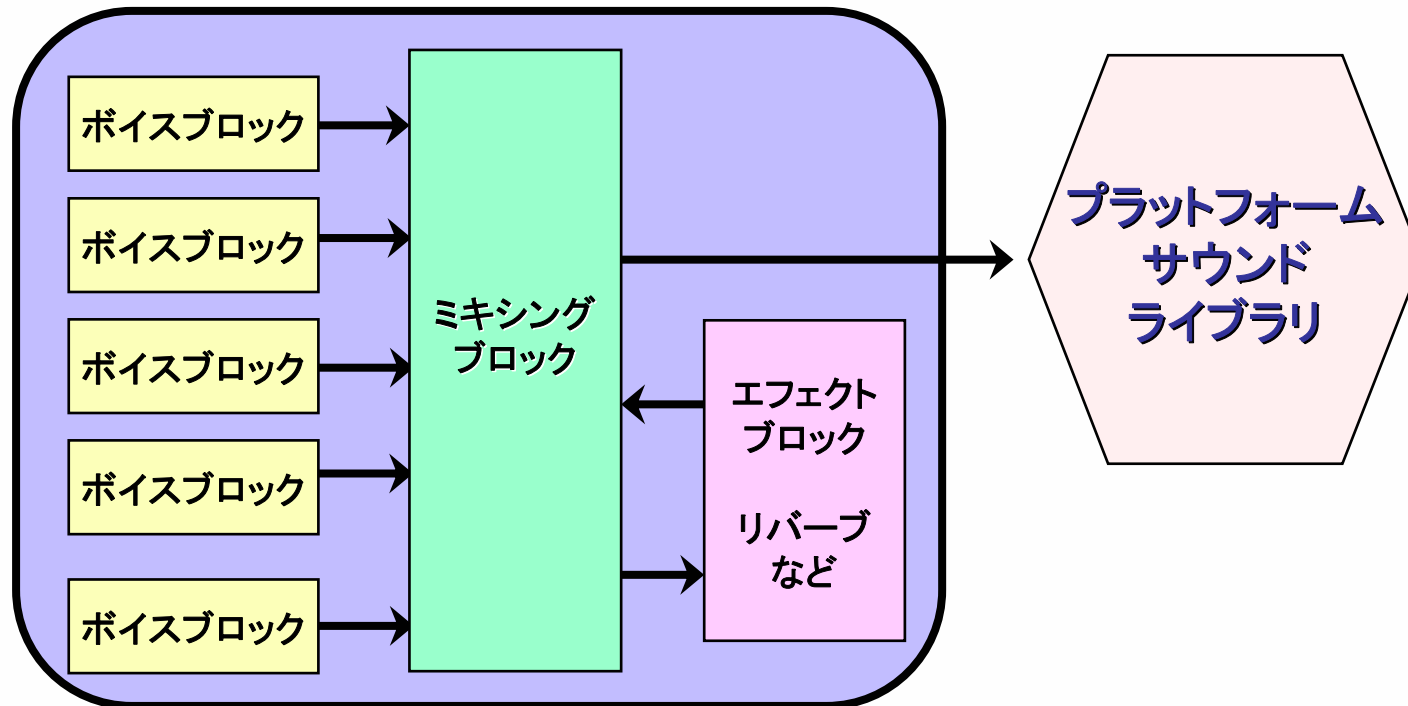


# CRI サウンドレンダラ

## ■ 標準的なミキサーコンソールと同じ構成を採用

ボイスブロックからの音声をミキシングして、エフェクタに送ることができます。

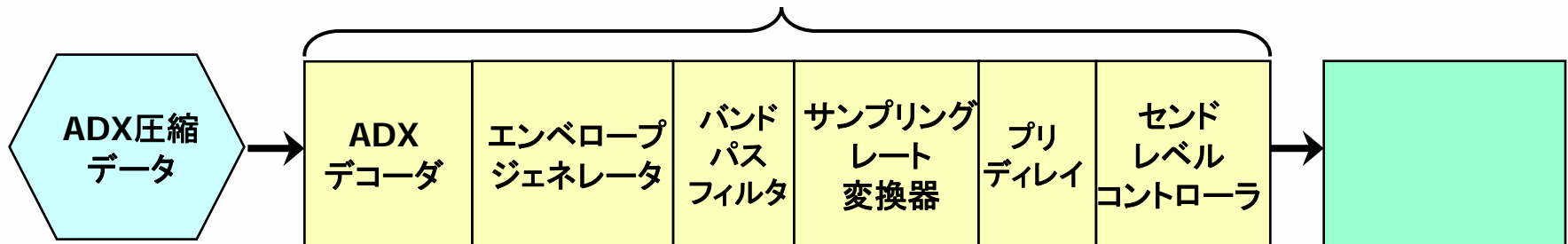
### サウンドレンダラ





# ボイスブロック

## ボイスブロック



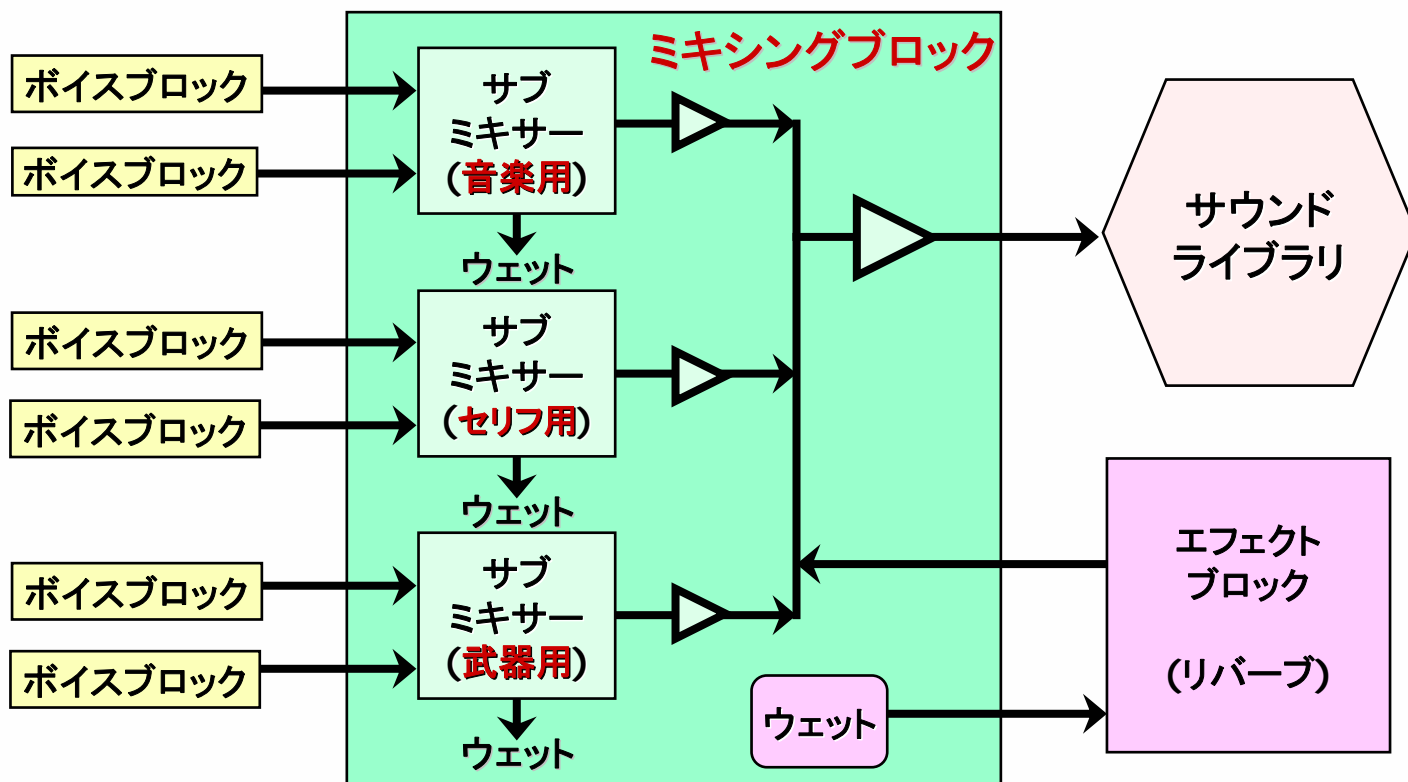
### ■ ボイスパラメータ

1. ボリューム
2. ピッチ
3. プリディレイ
4. エンベロープ  
ADSR + デイレイ + ホールド
5. フィルタ  
高域カットオフ周波数  
低域カットオフ周波数
6. ダイレクトセンドレベル  
左 左サラウンド  
右 右サラウンド  
センタ LFE  
拡張1 拡張2
7. ウェットセンドレベル  
0:リバーブ

# ミキシングブロック

## ■ サブミキサー

音量をグループごとにコントロールすることができます。

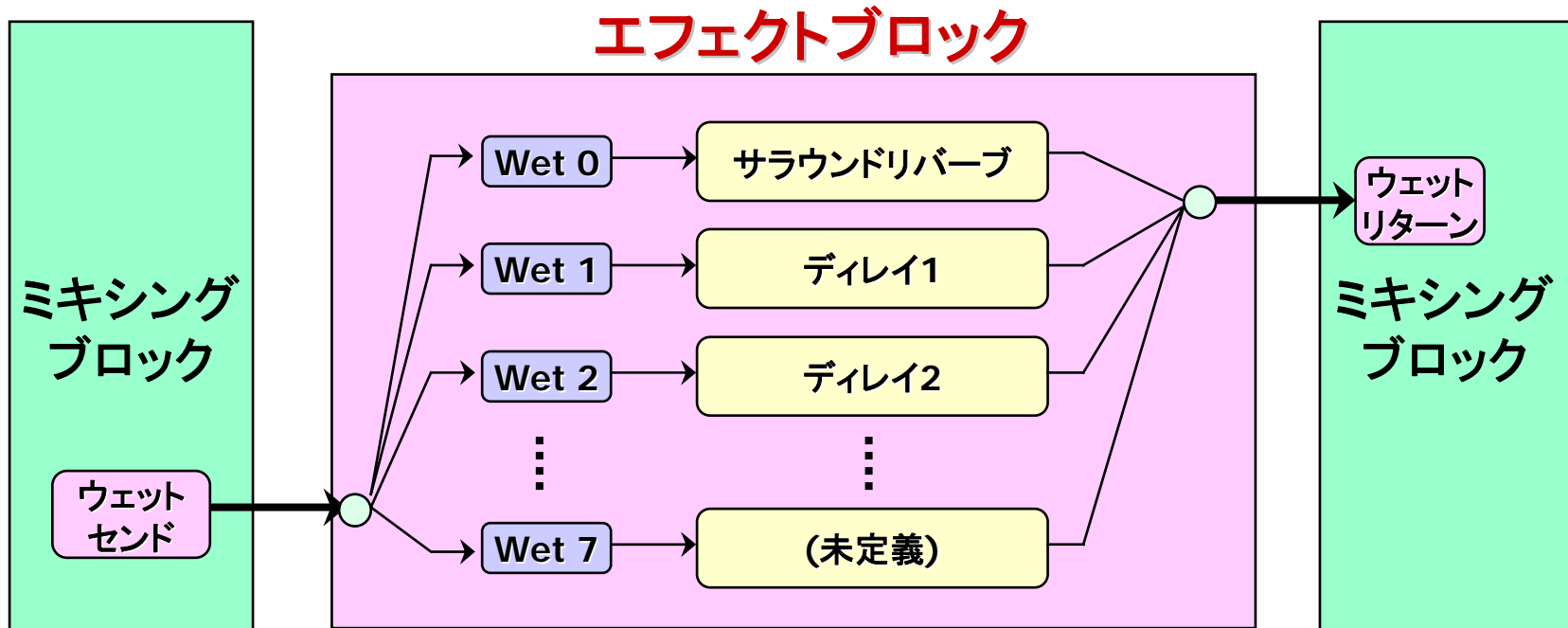


# エフェクトブロック

## ■ エフェクトブロック 内に8種類のエフェクトをサポート

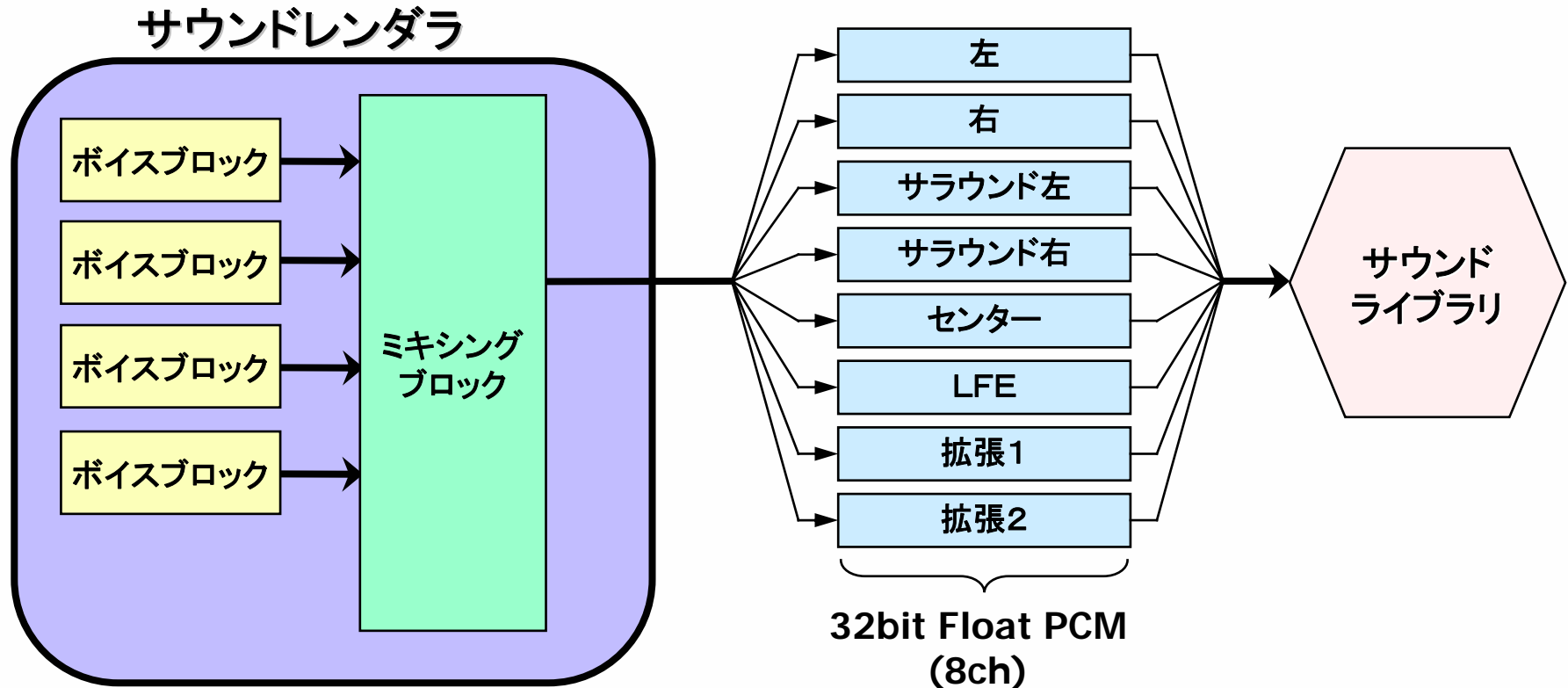
ウェット0は **4ch サラウンドリバーブ**です。

この他にも、ご要望により様々なエフェクトをサポートしていく予定です。



# サウンドレンダラからの出力

## ■ サウンドレンダラの出力 8チャンネルリニアPCM





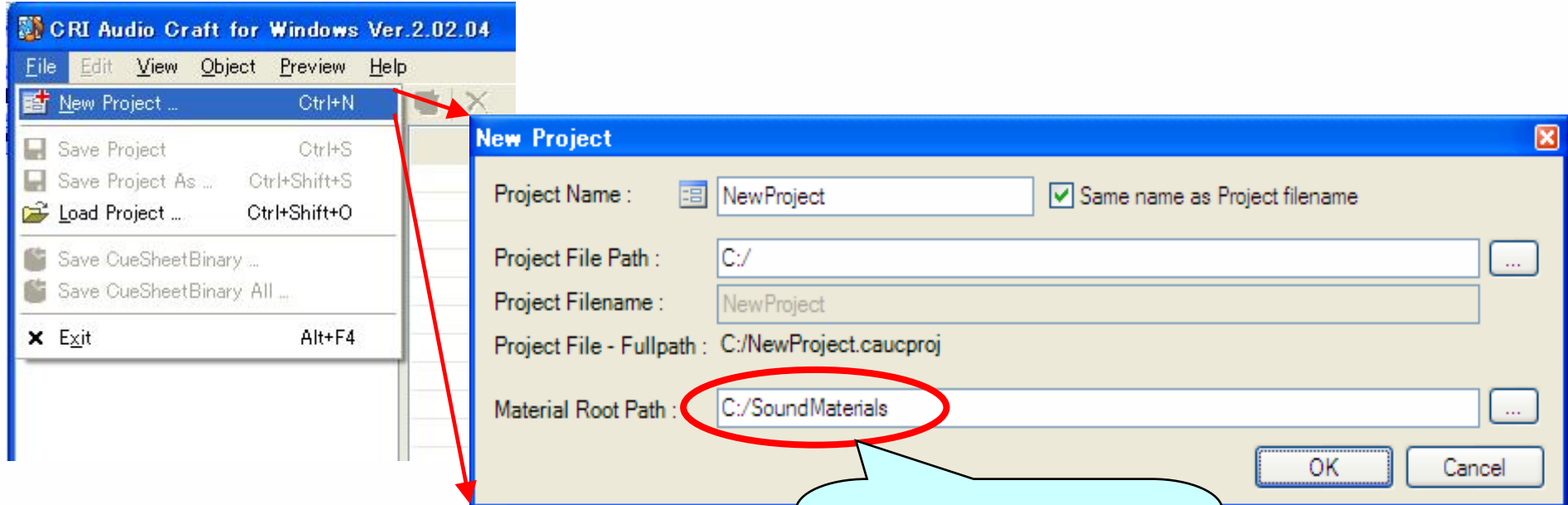
# オーサリングツール 「CRI Audio Craft」の基本操作

# 各ペインの役割

The screenshot shows the CRI Audio Craft for Windows Ver 2.02.02 interface. Four panels are highlighted with red rounded rectangles and labeled with red text:

- ツリーペイン (Tree Panel):** Located on the left, it displays a hierarchical project tree with folders like 'Synth', 'Ambient', 'DrySound', 'Music', 'SoundEffects', 'Voices', 'Globals', 'Asac', 'VoiceLimitGroup', 'CueSheet', and 'ALL'.
- リストペイン (List Panel):** Located in the top center, it shows a table of audio assets. The table has columns for 'Name', 'Volume', 'Pitch', 'PreDelayTime', and 'Comment'. Assets listed include 'AtomicExplosion5', 'AtomicExplosion8', 'bomb', 'bomb2', 'footstep', 'footstepTshot', 'footstepTshotR', 'gun1\_High', 'gun2\_Low', 'hel\_Loop', 'ht1\_long', 'ht2\_short', 'machine\_gun1\_fast', 'machine\_gun2\_slow', 'MachineGunAk47', and 'MachineGunG36'.
- プロパティペイン (Property Panel):** Located on the top right, it shows the properties for the selected 'hel\_Loop' asset, including 'Output', 'Envelope', 'SoundElement', 'Volume' (set to 1.00), 'Pitch' (set to 0), 'PreDelayTime' (set to 0), 'Comment', 'VoiceManagement', 'Priority' (set to 0), 'PriorityType' (set to LastPriority), 'VoiceLimitGroup', and 'Filter: CutOffFrequency' (High: 20,000 Hz, Low: 0 Hz).
- ディテールペイン (Detail Panel):** Located in the bottom center, it shows the 'AISAC View' for the selected asset, including a 'PatternList' table with columns for 'PatternName', 'ControlName', and 'Type'. It also includes a 'PointList' table with columns for 'Volume' and 'Pitch', and a 'PointGraph' showing a graph of these parameters over time.
- ログペイン (Log Panel):** Located at the bottom, it displays system logs, including messages like 'Internal Library CACD: ver.0.72.18', 'NET Framework runtime version: v1.1.4322', and 'File System: Create a new AsacPattern 'NewAsacPattern', in 'hel\_Loop'.

# 新規プロジェクトの作成



マテリアルフォルダ  
を指定します。

# 大量のサウンドデータの調整

## ■ 膨大なサウンドデータを効率よく調整

フォルダ内にあるサウンドデータに対して、音量やピッチなどを一括設定できます。

The screenshot displays the CRI Audio Craft for Windows Ver.2.02.06 interface. The main window title is "CRI Audio Craft for Windows Ver.2.02.06 - NewProject [C:/NewProject.caucproj]". The menu bar includes File, Edit, View, Object, Preview, and Help. The left sidebar shows a project tree with folders like Synth, Ambient, Dialog, Music, and SoundEffects. The central pane is a table of sound data with columns for Name, Volume, Pitch, PreDelayTime, and Comment. The 'bomb' sound data is selected, and its properties are shown in the right-hand panel. The 'Main' tab is active, showing Volume (0.83), Pitch (0), and PreDelayTime (0) with corresponding sliders. Below these are sections for 'VoiceManagement' (Priority: 0, PriorityType: LastVoicePriority, VoiceLimitGroup: .) and 'Filter: CutOffFrequency' (High: 20,000 Hz, Low: 0 Hz).

Name	Volume	Pitch	PreDelayTime	Comment
AtomicExplosion5	1.00	0	0	
AtomicExplosion8	1.00	-130	0	
bomb	0.83	0	0	
bomb2	0.83	0	0	
footstep	0.83	0	0	
footstep1shotL	0.83	230	0	
footstep1shotR	0.83	0	85	
gun1_High	1.00	0	136	
gun2_Low	1.00	0	0	
heli_Loop	1.00	0	0	
hit1_long	1.00	0	0	
hit2_short	1.00	0	0	
kan	1.00	0	0	
machine_gun1_fast	1.00	0	0	
machine_gun2_slow	1.00	0	0	
MachineGunAk47	1.00	0	0	
MachineGunG36	1.00	0	0	




# プリミティブシンセサイザ

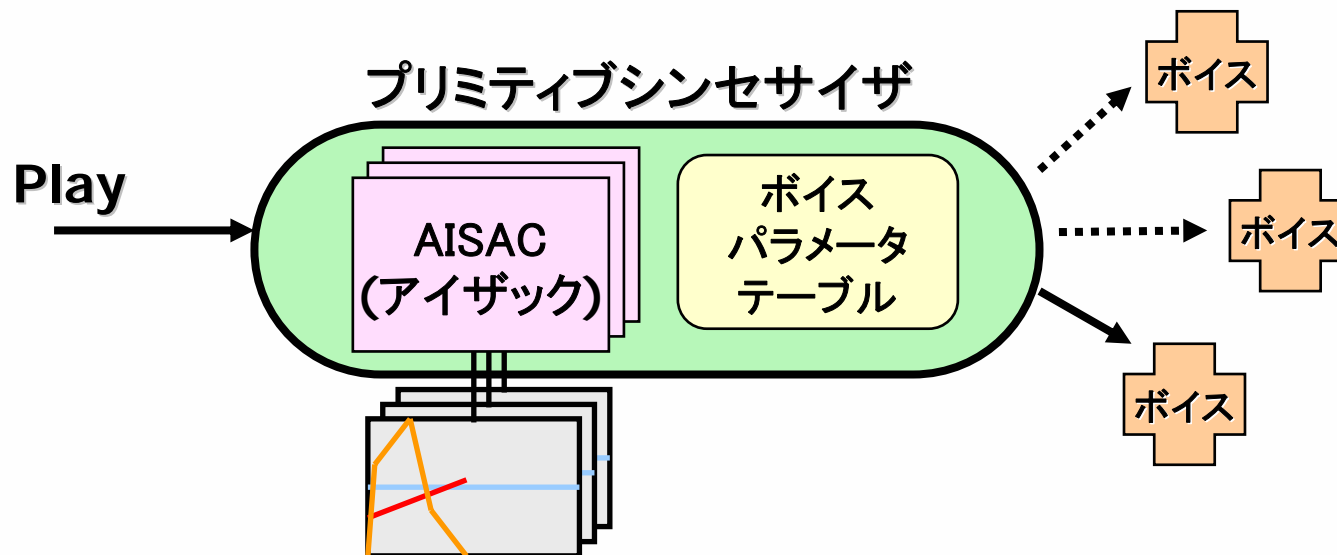
# プリミティブシンセサイザ

## ■ ボイスの生成

プリミティブシンセサイザは、ボイスパラメータテーブルとAISACを使用して、ボイスを生成します。

## ■ アイザック (Advanced Interactive Sound and Active Controller)

プログラマが指定するコントロール値からサウンドデザイナーの設計したグラフに従って、ボイスパラメータを決定する仕組みです。



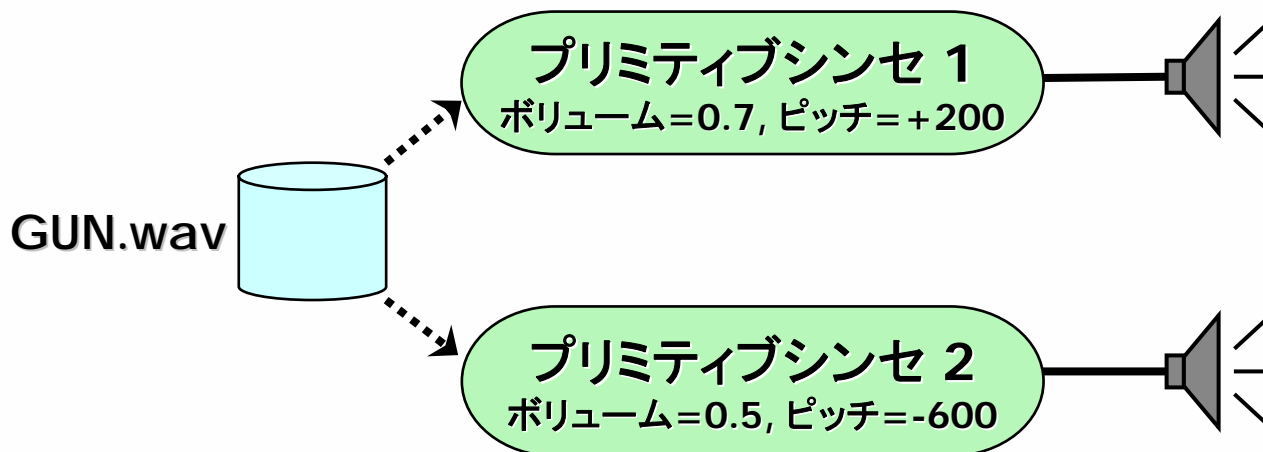
# 波形データの共有

## ■ シンセサイザ

各シンセサイザは、音声の再生パラメータのみを持ちます。

## ■ 複数の効果音に対して音声データを共有

複数のシンセサイザが1つの波形データを共有することができます。



# ボリューム

## ■ リニアスケール

- 1.0で原音のまま、0.0で無音になります。
- デシベルでは無音がマイナス無限大になります。  
→ 内部を浮動小数として表現しているため不向き。

## ■ プログラムのボリューム値と乗算

$$\text{再生時のボリューム値} = \text{サウンドデザインのボリューム値} \times \text{プログラムのボリューム値}$$



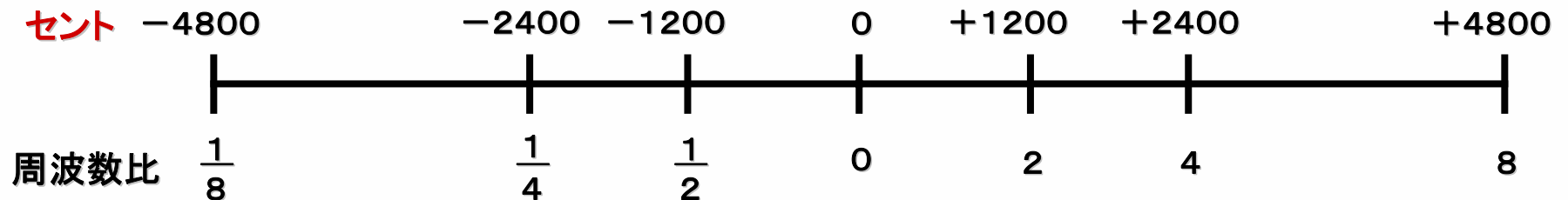
# ピッチ

## ■ 対数スケール (セント値)

- +1200セントで1オクターブ上昇します。(周波数としては2倍)
- 100セントで半音となり、感覚にマッチします。
- -4800~+4800セントまで指定可能です。

## ■ プログラムのピッチ値と加算

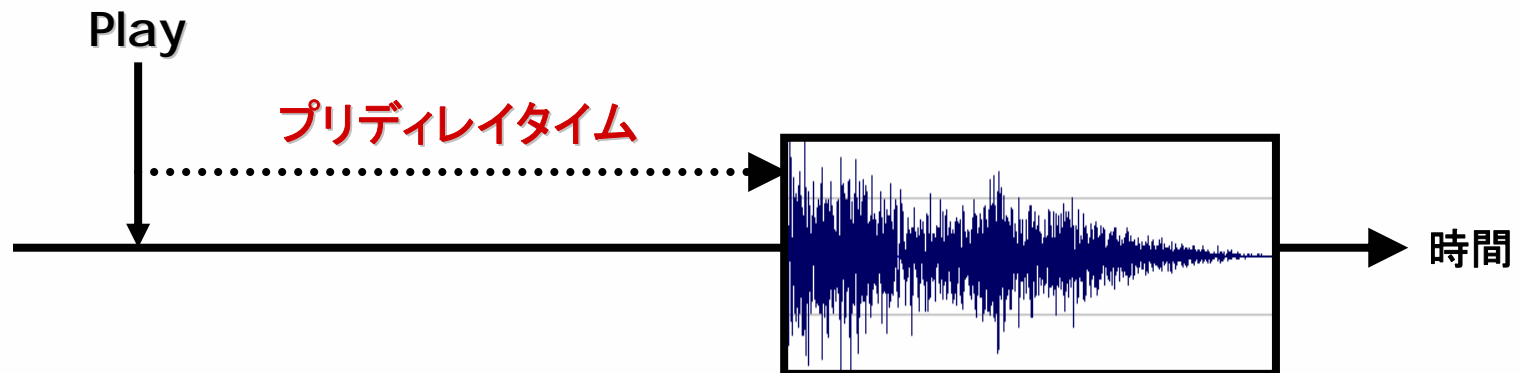
$$\text{再生時のピッチ値} = \text{サウンドデザイナーのピッチ値} + \text{プログラムのピッチ値}$$



# プリディレイ

## ■ 発音までの遅延時間

- 再生指示がされてから、実際に発音するまでの時間を設定します。(最大64秒)
- 発音が始まる前からエンベロープがかかります。
- コンプレックスシンセサイザ内で威力を発揮します。



# フィルタ

## ■ バターワース型バンドパスフィルタ

- 低域カットオフ周波数と高域カットオフ周波数を指定します。

## ■ 正規化周波数でのコントロールをサポート

- 0.0を24Hz、1.0を24000Hzとして対数スケールで正規化。
- 自然な感じでフィルタのかかり具合をコントロールできます。

## ■ プログラマによる設定方法

- カットオフ周波数を直接設定します。
- 正規化周波数によるオフセットゲインコントロール(後述)ができます。

# 正規化周波数

## ■ サウンドデザイナの設定した値をプログラマがコントロール

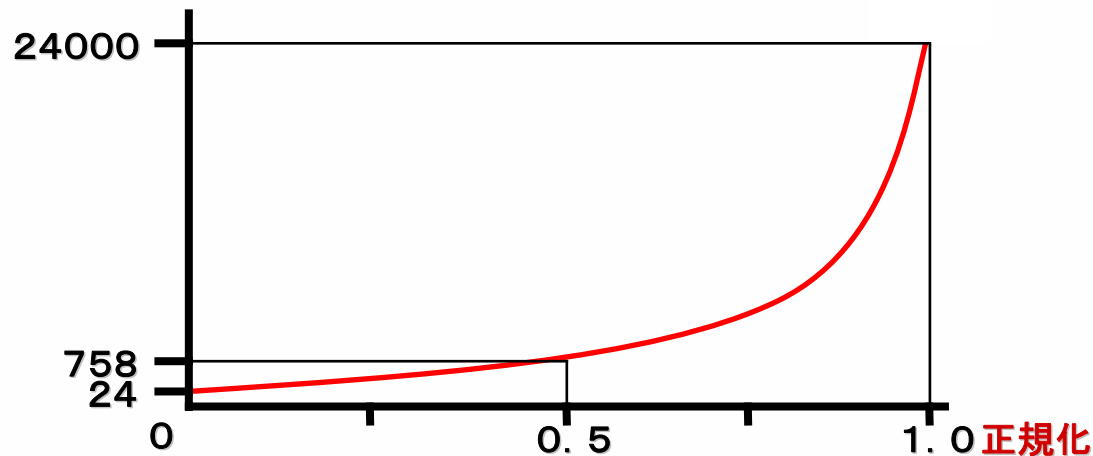
周波数でコントロールするのは困難です。

## ■ 計算式

実際のカットオフ周波数  $x$  (Hz) 正規化周波数  $y$

$$y = \text{pow}(24, (\log_{24}(24000)-1)*x+1)$$

$$x = (\log_{24}(y)-1)/(\log_{24}(24000)-1)$$



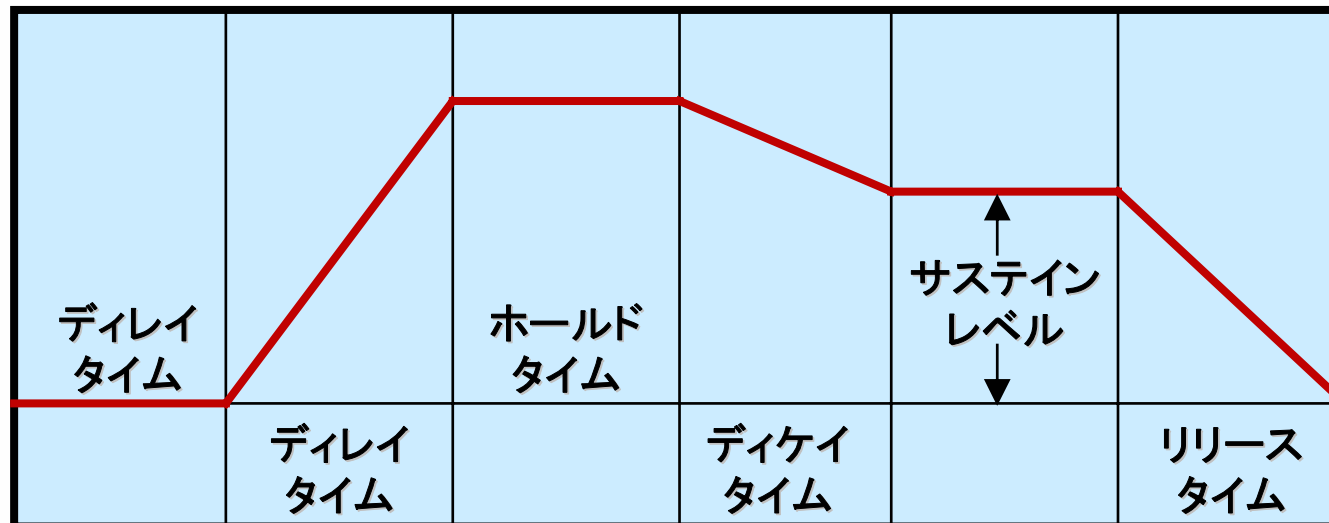
正規化	通常
0.0	24
0.2	96
0.4	380
0.6	1514
0.8	6028
1.0	24000



# エンベロープ

## ■ 伝統的なADSR+ディレイ+ホールド

- ディレイタイム、アタックタイム、ホールドタイム、ディケイタイム、サステインレベル、リリースタイムを指定できます。
- ディレイタイム中は、発音は開始しますがボリュームが0(無音)となります。

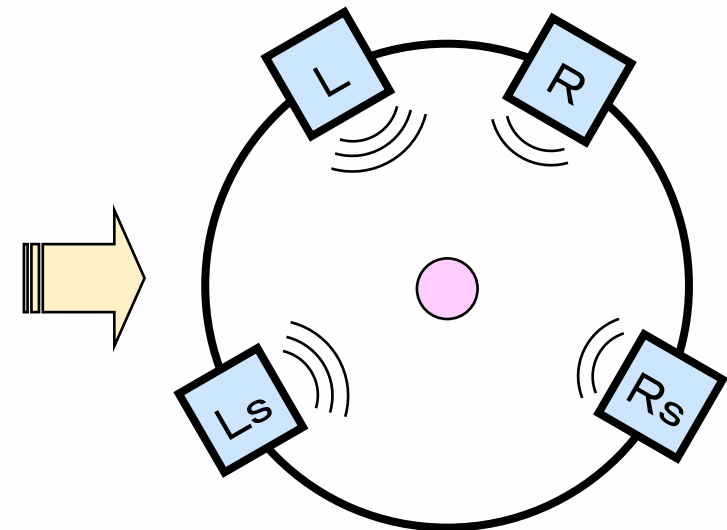
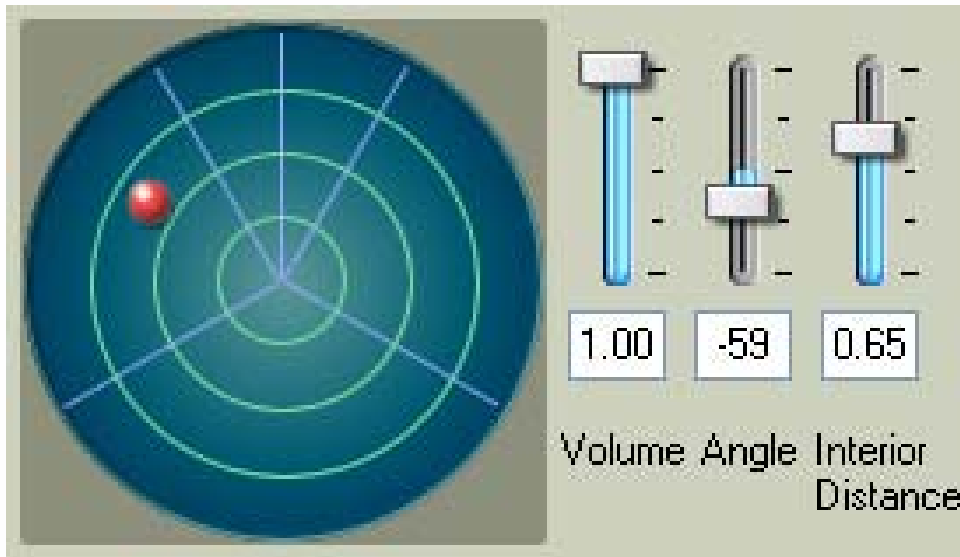


# 3Dパンニング

■ 角度とインテリア距離から各スピーカーからの出力レベルを計算

■ プログラムのコントロール

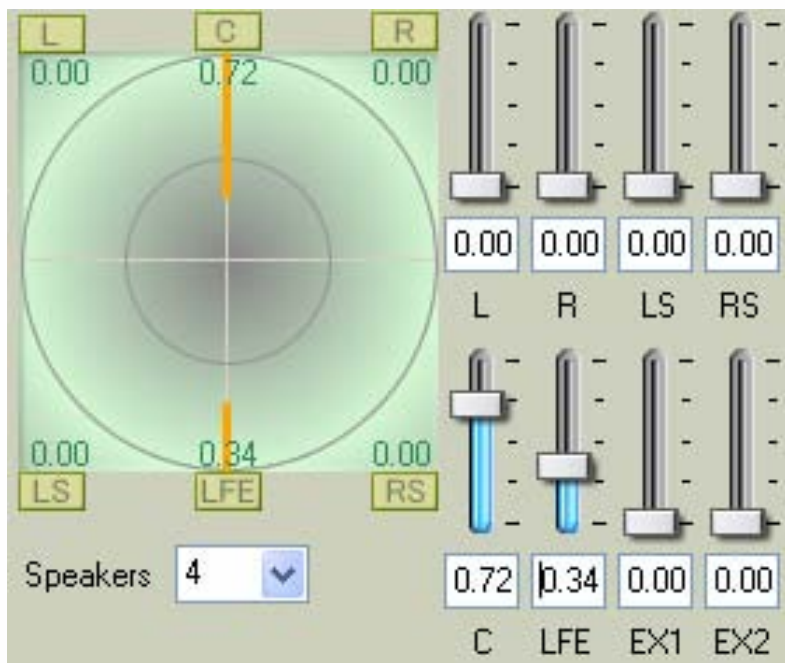
- サウンドデザイナーの設定した値から加算して回転します。 → オフセット
- サウンドデザイナーの設定した値まで0.0から1.0で回転します。 → ゲイン



## ダイレクトセンドレベル

### ■ 各スピーカーからの出力レベルを独立に設定可能

- センタースピーカー、サブウーファー (LFE) が効果的です。
- 左、右、左サラウンド、右サラウンドスピーカー → 3Dパンニング機能



**センタースピーカー**  
→ セリフをクリアに！  
**サブウーファー (LFE)**  
→ 大迫力の爆発音！

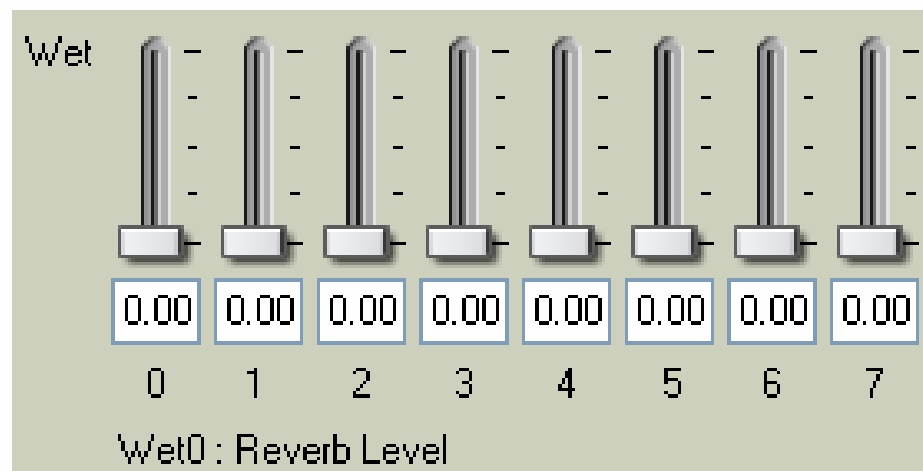
## ウェット(エフェクト) SEND

### ■ 各エフェクタへの出力レベルを設定可能

- ウェット0は4チャンネルサラウンドリバーブに接続しています。
- デイレイ、コンプレッサなど追加予定です。

### ■ 4チャンネルサラウンドリバーブ

- 高品位なリバーブを低CPU負荷で実現しました。



# オフセット・ゲインコントロール1

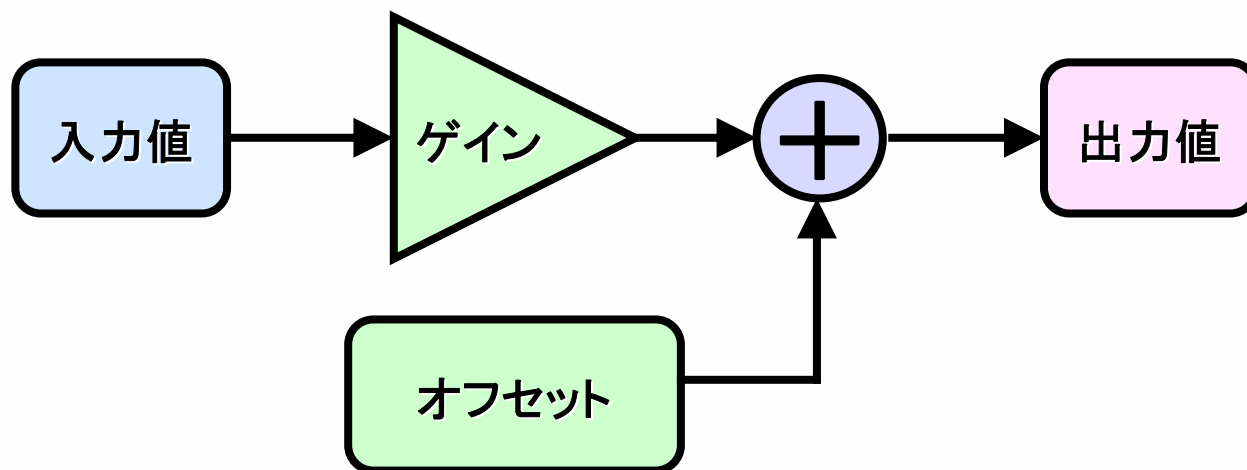
## ■ プログラマからの柔軟なコントロールを実現

- 設定値 = (サウンドデザイナーの設定値) × ゲイン + オフセット

## ■ 設定例

まったく変更しない場合 : ゲイン=1.0、オフセット=0.0

完全に乗っ取る場合 : ゲイン=0.0、オフセット=(新しい値)



## オフセット・ゲインコントロール2

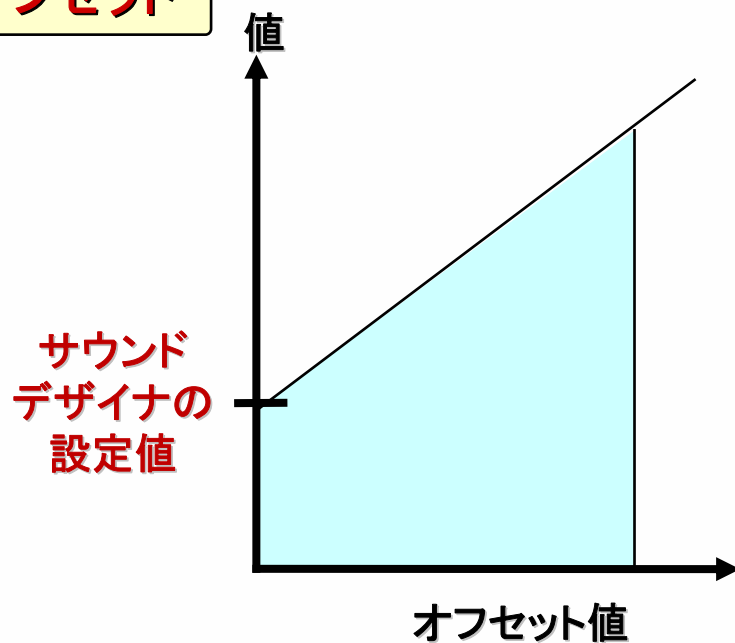
■ オフセット

サウンドデザイナーの設定値からの相対値です。

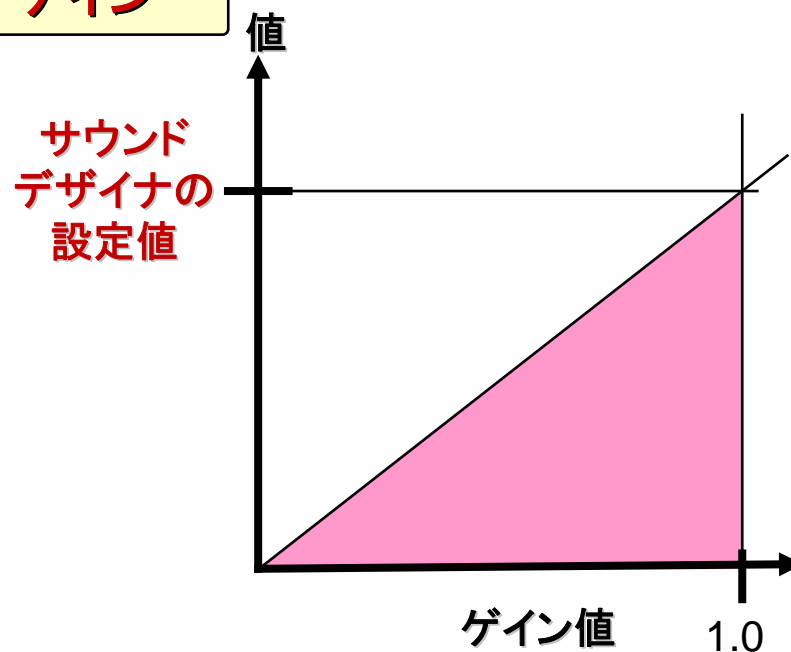
■ ゲイン

サウンドデザイナーの設定値の影響度です。

オフセット



ゲイン

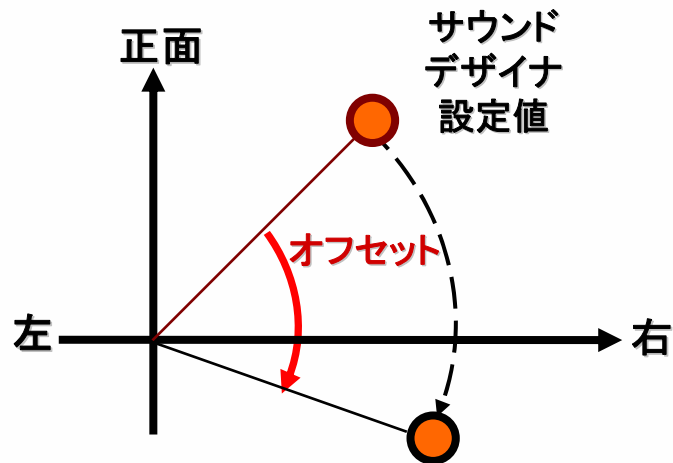


# オフセットによるコントロール

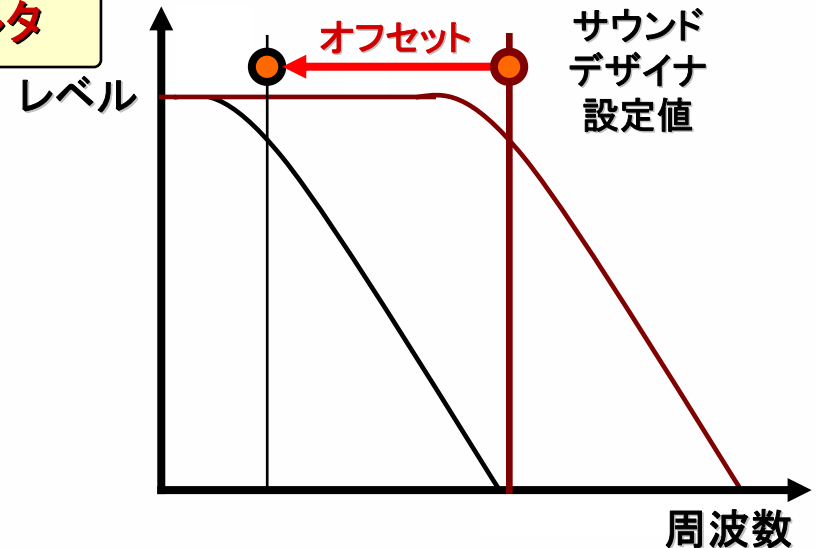
## ■ オフセットによるコントロール (ゲイン=1.0)

3Dパンニング	サウンドデザイナーの設定したポジションを基準に回転します。
ドライ・ウェットセンド	サウンドデザイナーの設定値にさらに加算します。
フィルタ	サウンドデザイナーの設定した正規化周波数に加算します。

### 3Dパンニング



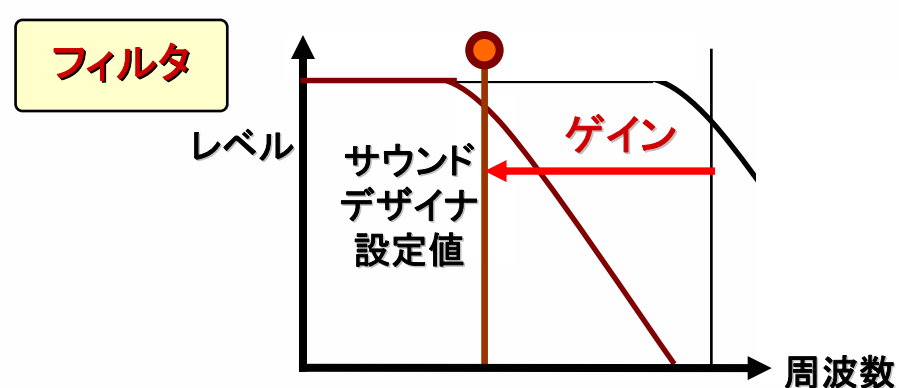
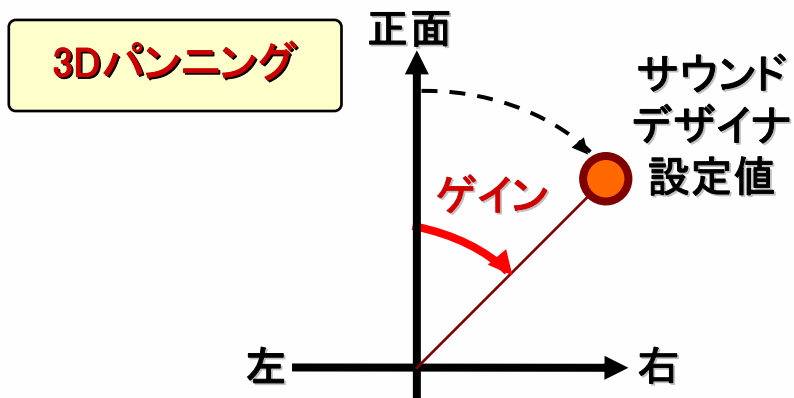
### フィルタ



# ゲインによるコントロール

## ■ ゲインによるコントロール (オフセット=0.0)

- |                  |   |
|------------------|---|
| 3Dパニング           | サウンドデザイナーの設定したポジションまで回転します。                           |
| ドライ・ウェットセンド      | ゲイン = 0.0 でミュートします。<br>ゲイン = 1.0 でサウンドデザイナーの設定値となります。 |
| フィルタ (正規化周波数で計算) |   |
| 低域カットオフ周波数       | 1.0 から サウンドデザイナーの設定値まで変化します。                          |
| 高域カットオフ周波数       | 0.0 から サウンドデザイナーの設定値まで変化します。                          |





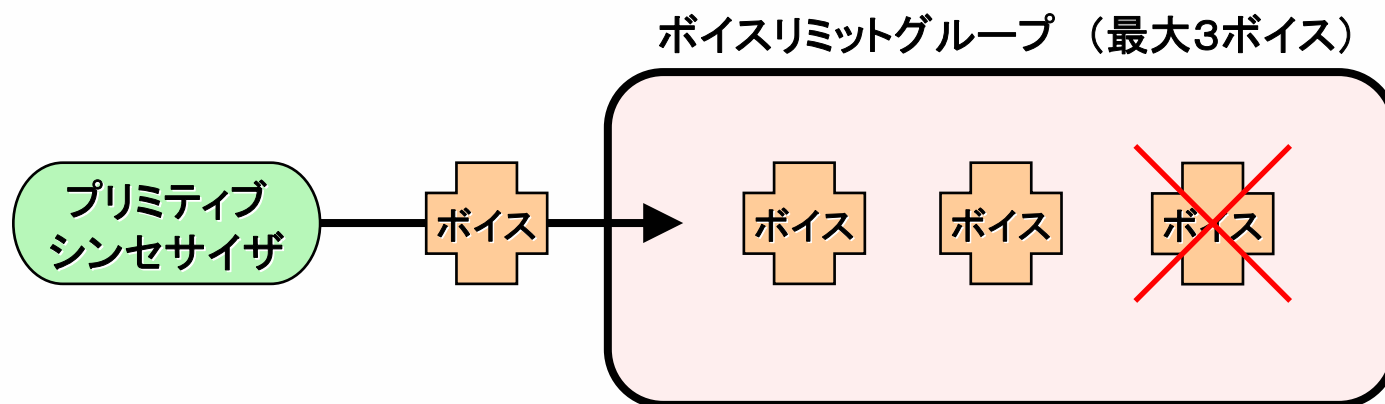
# 最大発音数制御

## ■ 同時に発音可能なボイス数を制限

- 声や環境音やなどは1つしかならしてはいけない。
- 多くの足音を同時に再生するとノイズになってしまう。

## ■ ボイスリミットグループ

- ボイスリミットグループに最大発音数を設定します。
- プリミティブシンセサイザごとに所属するボイスリミットグループを指定できます。



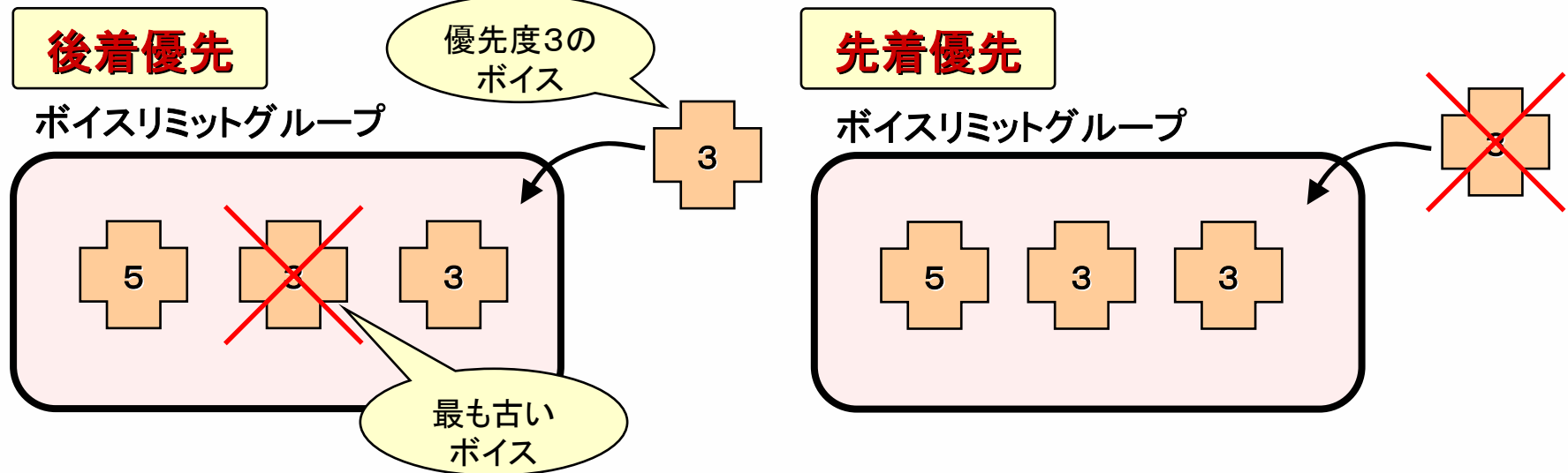
# 発音プライオリティ

- 最大発音数を超えた場合、高い優先度のボイスを再生
- 先着優先・後着優先をサポート

同じプライオリティの処理

後着優先の場合は、最も古いボイスを再生を停止します。

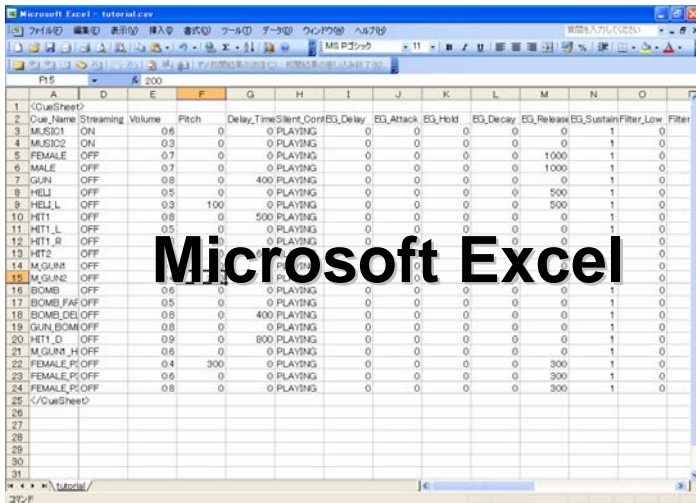
先着優先の場合は、再生されない。



# Excelとの連携

## ■ CSV形式ファイルのインポート

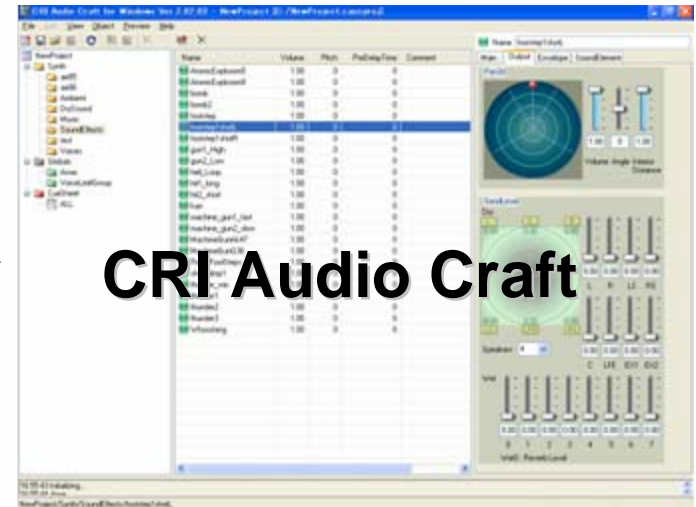
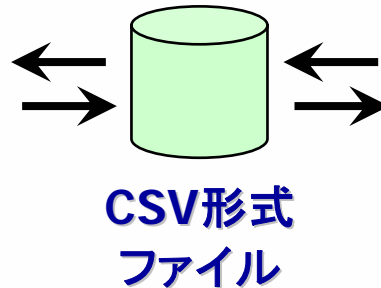
- CRI Audio Craftからボリュームやピッチなどのパラメータを**CSV形式ファイル**でエクスポートします。
- Microsoft Excelなどの表計算ソフトから**CSV形式ファイル**を編集します。
- **CSV形式ファイル**からボリュームやピッチなどのパラメータをインポートします。



Microsoft Excel - tutorial.csv

QusName	Streaming	Volume	Pitch	Delay_Time	Silent_Conf	EG_Delay	EG_Attack	EG_Hold	EG_Decay	EG_Release	EG_Sustain	Filter_Low	Filter
MUSIC1	ON	0.6	0	0	PLAYING	0	0	0	0	0	1	0	
MUSIC2	ON	0.3	0	0	PLAYING	0	0	0	0	0	1	0	
FEMALE	OFF	0.7	0	0	PLAYING	0	0	0	0	1000	1	0	
MALE	OFF	0.7	0	0	PLAYING	0	0	0	0	1000	1	0	
GUN	OFF	0.8	0	400	PLAYING	0	0	0	0	0	1	0	
HELL	OFF	0.5	0	0	PLAYING	0	0	0	0	500	1	0	
HELL_L	OFF	0.3	100	0	PLAYING	0	0	0	0	500	1	0	
HIT1	OFF	0.8	0	500	PLAYING	0	0	0	0	0	1	0	
HIT1_L	OFF	0.5	0	0	PLAYING	0	0	0	0	0	1	0	
HIT1_R	OFF	0.5	0	0	PLAYING	0	0	0	0	0	1	0	
HIT2	OFF	0.8	0	0	PLAYING	0	0	0	0	0	1	0	
M_GUN	OFF	0.8	0	0	PLAYING	0	0	0	0	0	1	0	
M_GUN2	OFF	0.8	0	0	PLAYING	0	0	0	0	0	1	0	
BOMB	OFF	0.6	0	0	PLAYING	0	0	0	0	0	1	0	
BOMB_FAF	OFF	0.5	0	0	PLAYING	0	0	0	0	0	1	0	
BOMB_DELOFF	OFF	0.8	0	400	PLAYING	0	0	0	0	0	1	0	
GUN_BOM	OFF	0.8	0	0	PLAYING	0	0	0	0	0	1	0	
HIT1_D	OFF	0.9	0	800	PLAYING	0	0	0	0	0	1	0	
M_GUN_H	OFF	0.6	0	0	PLAYING	0	0	0	0	0	1	0	
FEMALE_PS	OFF	0.4	300	0	PLAYING	0	0	0	0	300	1	0	
FEMALE_PS	OFF	0.6	0	0	PLAYING	0	0	0	0	300	1	0	
FEMALE_PS	OFF	0.8	0	0	PLAYING	0	0	0	0	300	1	0	

Microsoft Excel





# インタラクティブサウンド AISAC

# インタラクティブサウンドのデザイン

## ■ インタラクティブサウンド

ゲームの状況に応じてサウンドを変化させたい。

例えば、リスナーと音源の距離や角度によってボリュームやピッチを変化させたい。

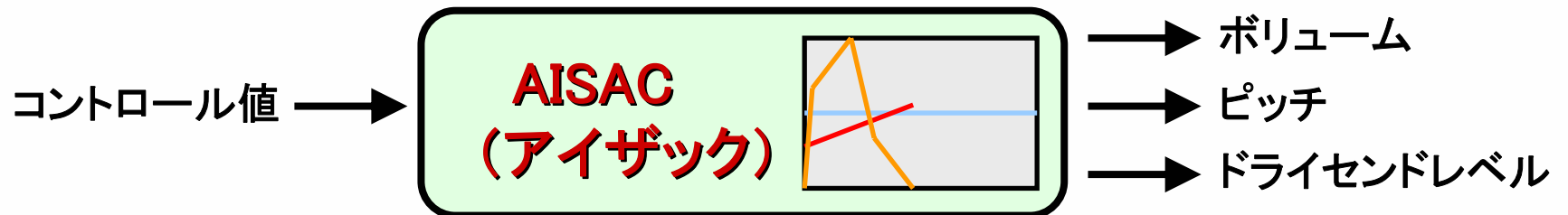
→ 物理的な計算では意図どおりに演出できない。

→ **AISAC (Advanced Interactive Sound and Active Controller)**

## ■ AISAC (アイザック)

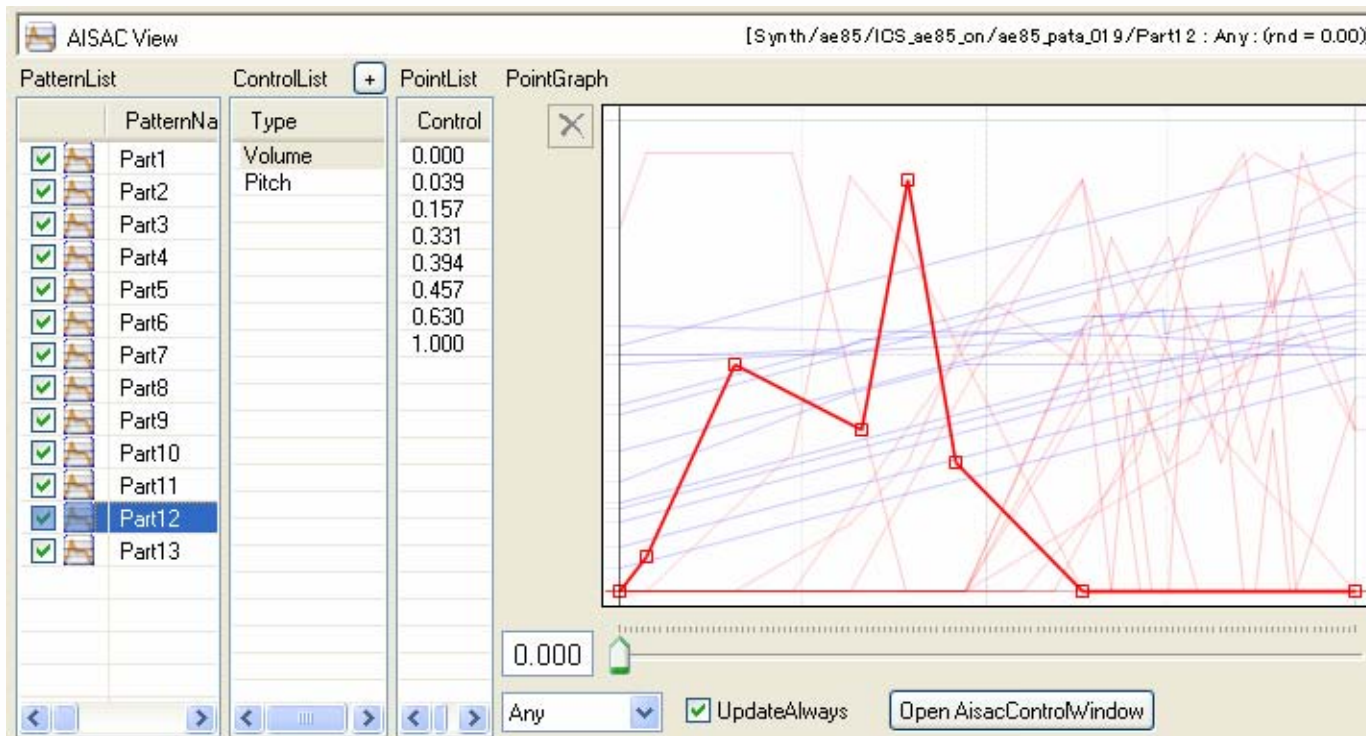
プログラマはゲームの状況に応じてコントロール値を決定します。

サウンドデザイナーは、このコントロール値に対してどのようにサウンドが変化するかを変換グラフによってデザインします。



# AISAC (アイザック)

■ **Advanced Interactive Sound and Active Controller**  
グラフによってサウンドデザイナーの感性を表現します。



# AISACのコントロールパラメータ

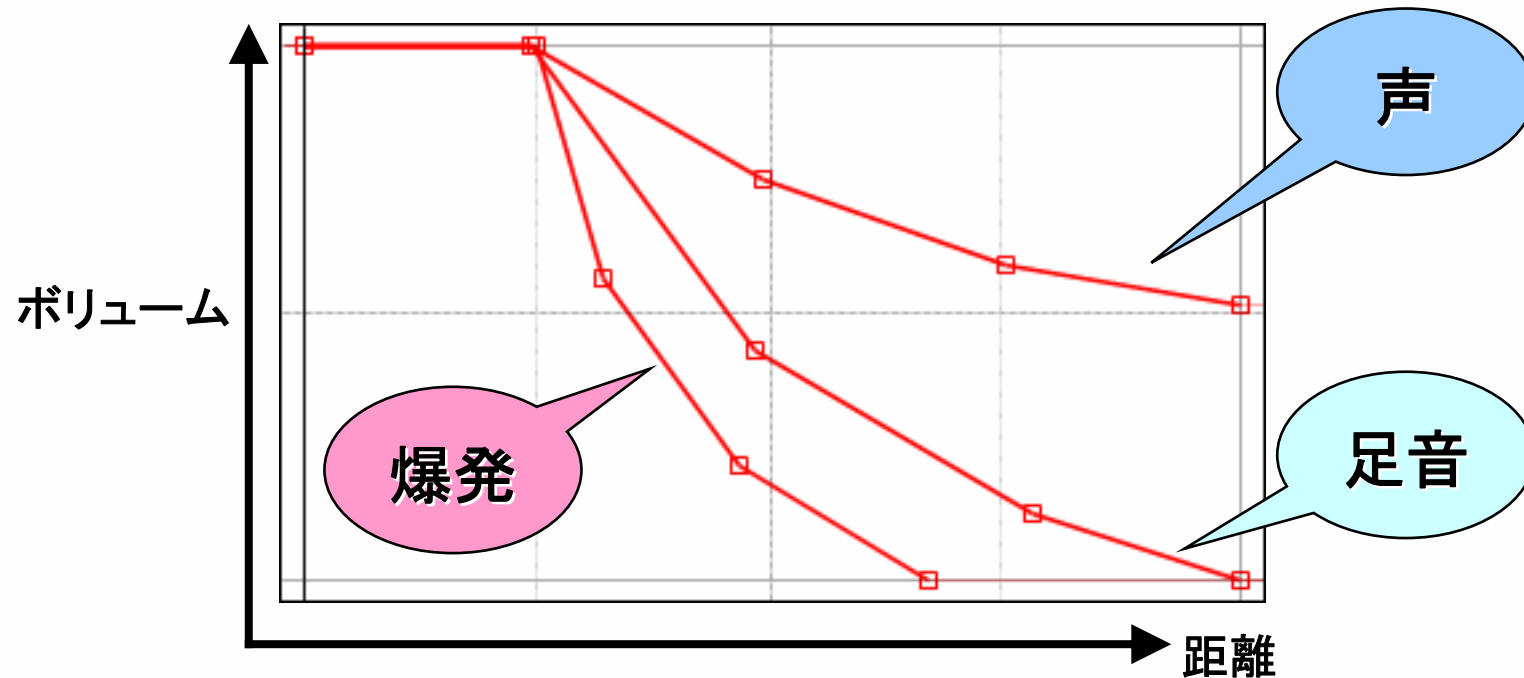
## ■ コントロールパラメータリスト

ボリューム	0.0 ~ 1.0 (リニアスケール)	ゲイン
ピッチ	- 4800 ~ 4800 (セント)	オフセット
プリディレイ	0 ~ 6000 (ミリ秒)	オフセット
フィルタ・高域カットオフ周波数	1.0 ~ 0.0 (正規化周波数)	ゲイン
フィルタ・低域カットオフ周波数	0.0 ~ 1.0 (正規化周波数)	ゲイン
3Dパンニング・角度	-180 ~ 180 (度)	オフセット
3Dパンニング・インテリア距離	0.0 (中心) ~ 1.0 (外側)	ゲイン
センターセンド量	0.0 ~ 1.0 (リニアスケール)	オフセット
サブウーファーセンド量	0.0 ~ 1.0 (リニアスケール)	オフセット
リバーブセンド量	0.0 ~ 1.0 (リニアスケール)	オフセット

# AISACによる距離減衰

## ■ 距離減衰カーブ

- 音の重要度によって減衰カーブは異なります。
- 例えば、声や足音はよく聞こえるようにし、爆発音は強弱のメリハリをつけます。

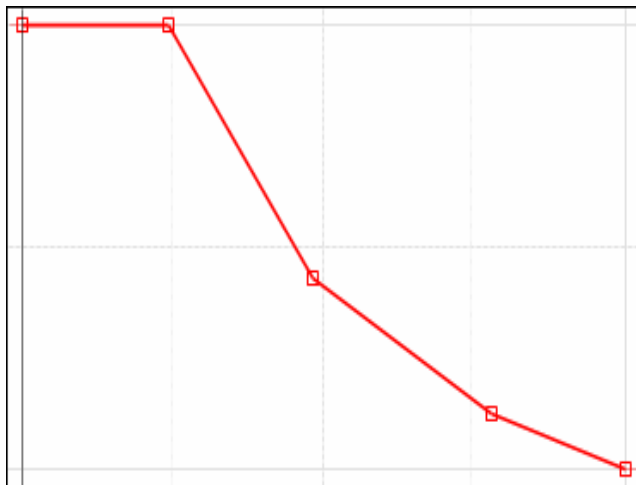




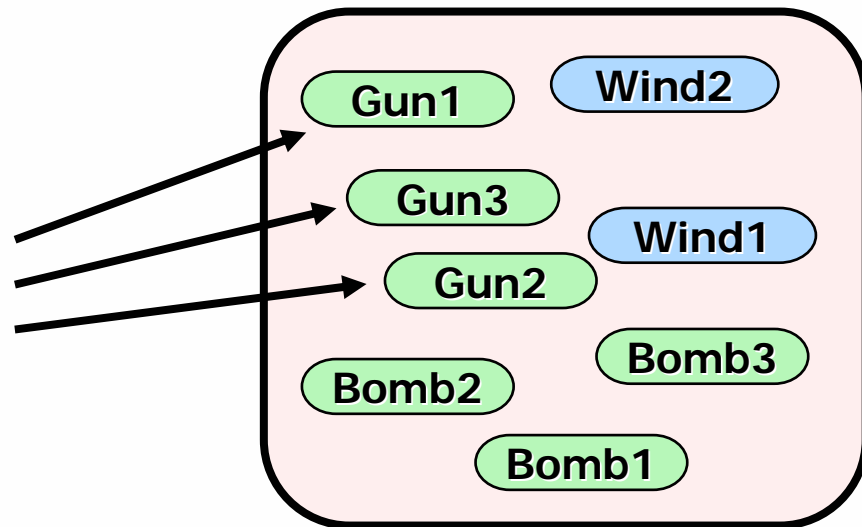
# グローバルAISAC

## ■ AISACのテンプレート機能

- デザインしたAISACのグラフを、多くの効果音に適用できます。
- テンプレートのグラフを修正すると、参照しているすべて効果音に適用されます。



銃タイプのAISACグラフ

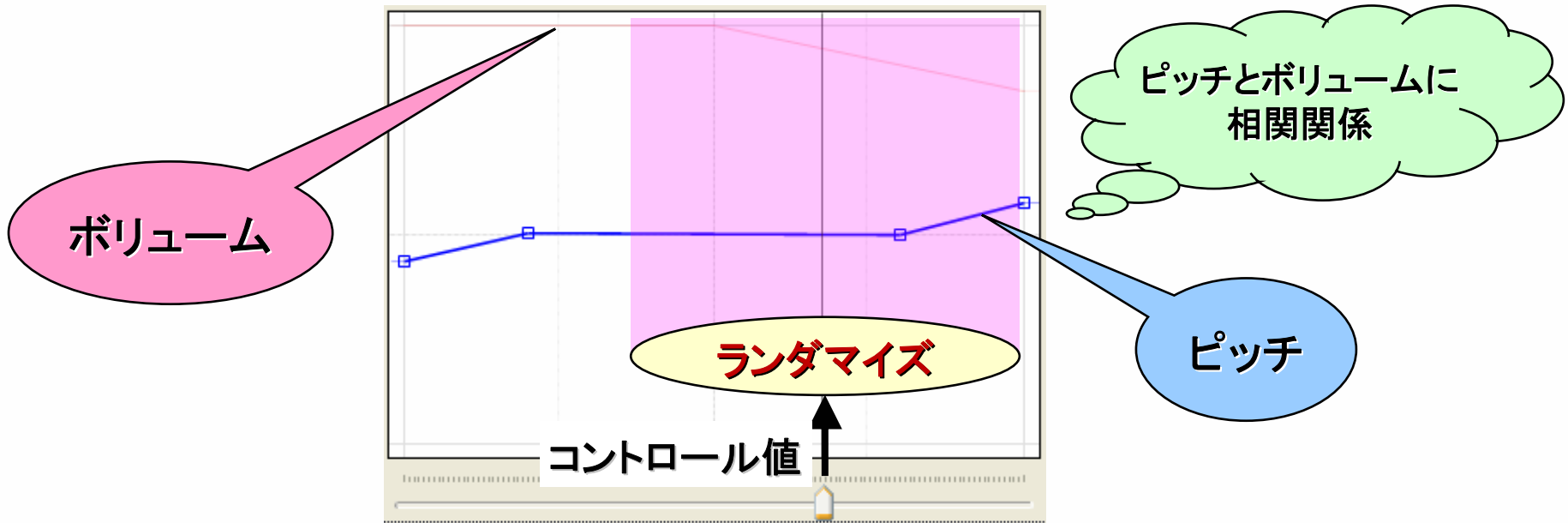


シンセサイザ

# ランダムAISAC

## ■ 確率密度をAISACグラフとしてデザイン

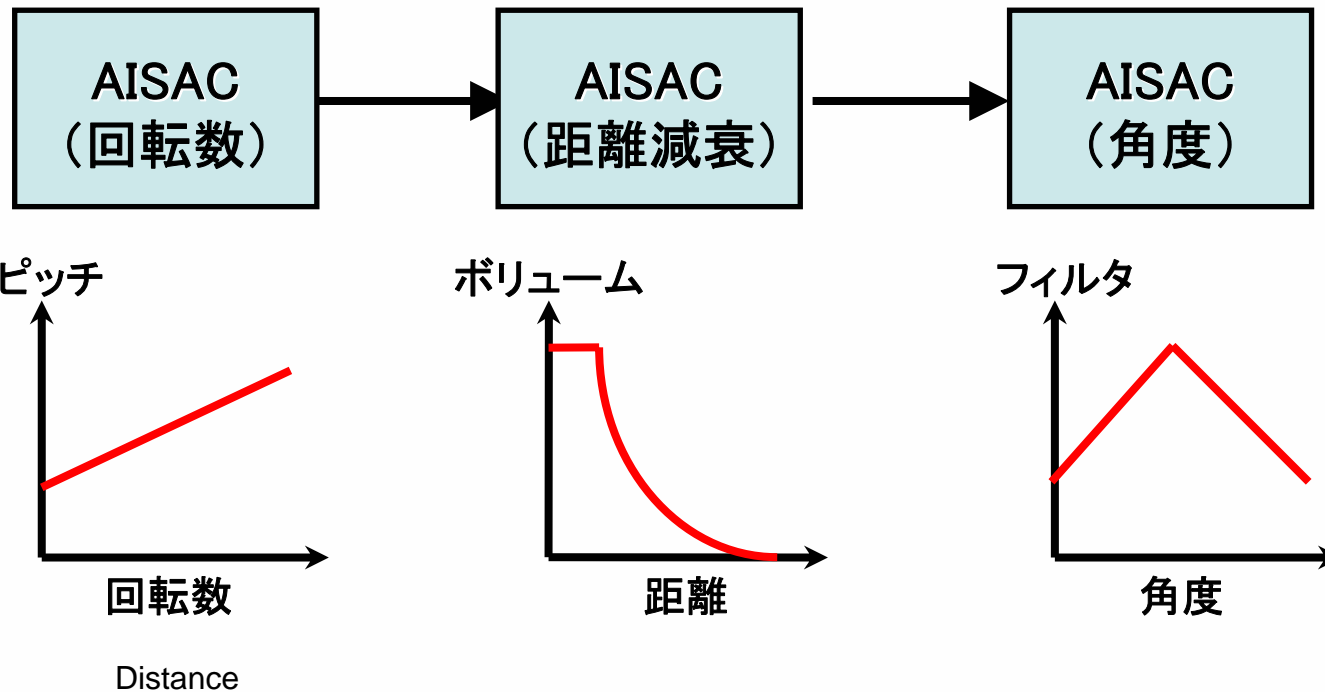
- コントロール値を指定した範囲内(ランダムレンジ)でランダムマイズします。
- ランダムレンジに1.0を指定すると、常にランダムマイズできます。
- ボリュームやピッチなど多くのパラメータをランダムマイズできます。



# 直列 AISAC

## ■ AISACの直列接続

AISACの直列接続を使用すると、ゲームの状況にあわせて、複数のパラメータをコントロールすることができます。




# コンプレックスシンセサイザ

# コンプレックスシンセサイザとは

## ■ コンプレックスシンセサイザとは

複数のサウンドを組み合わせて再生する仕組みです。

## ■ コンプレックスシンセサイザの役割

### (1) バリエーション

波形データを共有し、少ないメモリで様々な効果音を作成できます。

### (2) 一括加工

所持するシンセサイザの音をまとめて加工できます。

### (3) 時間軸への配置

プリディレイにより時間軸上に複数の効果音を配置できます。

### (4) サラウンド

モノラル効果音からサラウンド効果音を作成します。

### (5) モーフィング

クロスフェードにより音をモーフィングします。

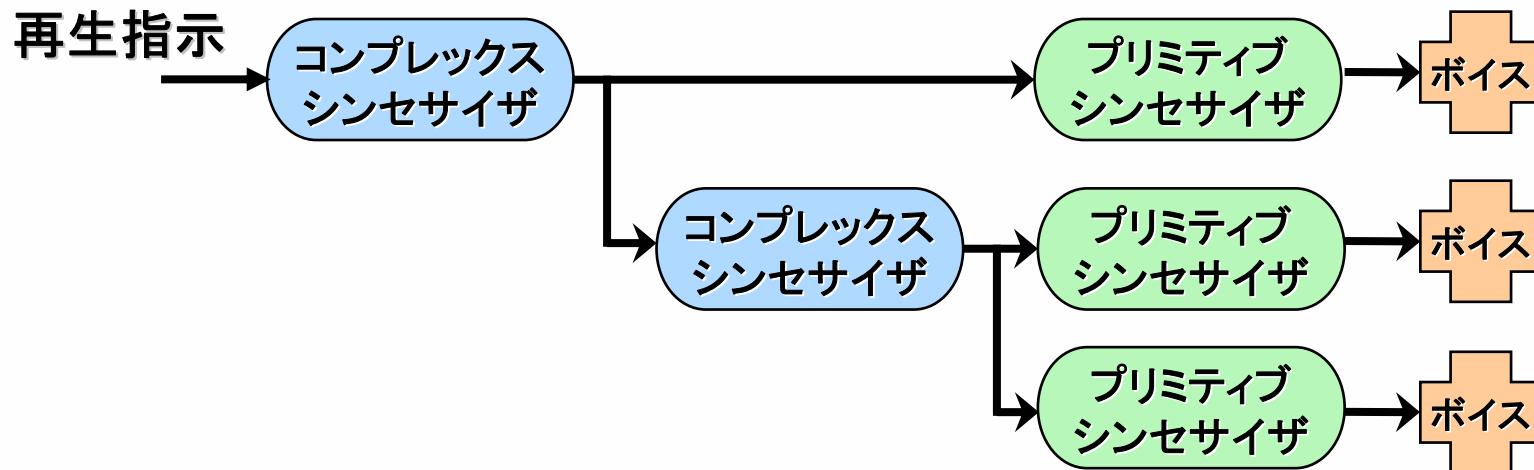
# コンプレックスシンセサイザの仕組み

## ■ 複数のボイスを同時に生成

コンプレックスシンセサイザは、「子シンセサイザ」によって複数のボイスを同時に生成します。

## ■ ボイスパラメータの継承

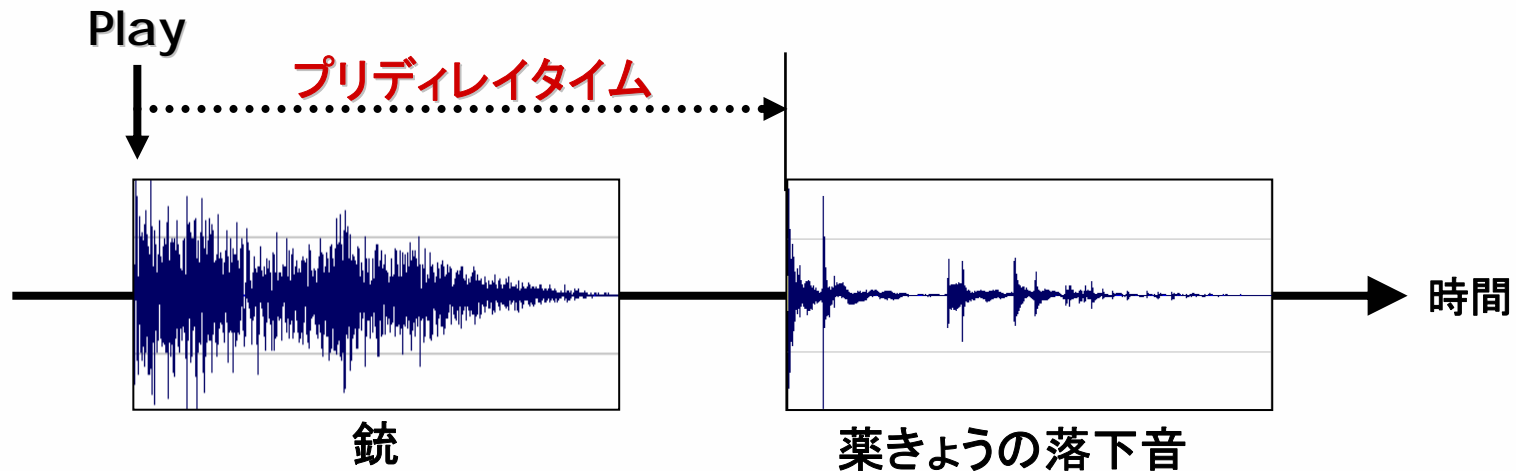
コンプレックスシンセサイザにボイスパラメータを設定すると、「子シンセサイザ」にも影響します。



## コンプレックスシンセ(時間軸への配置)

### ■ プリディレイによる簡易シーケンス

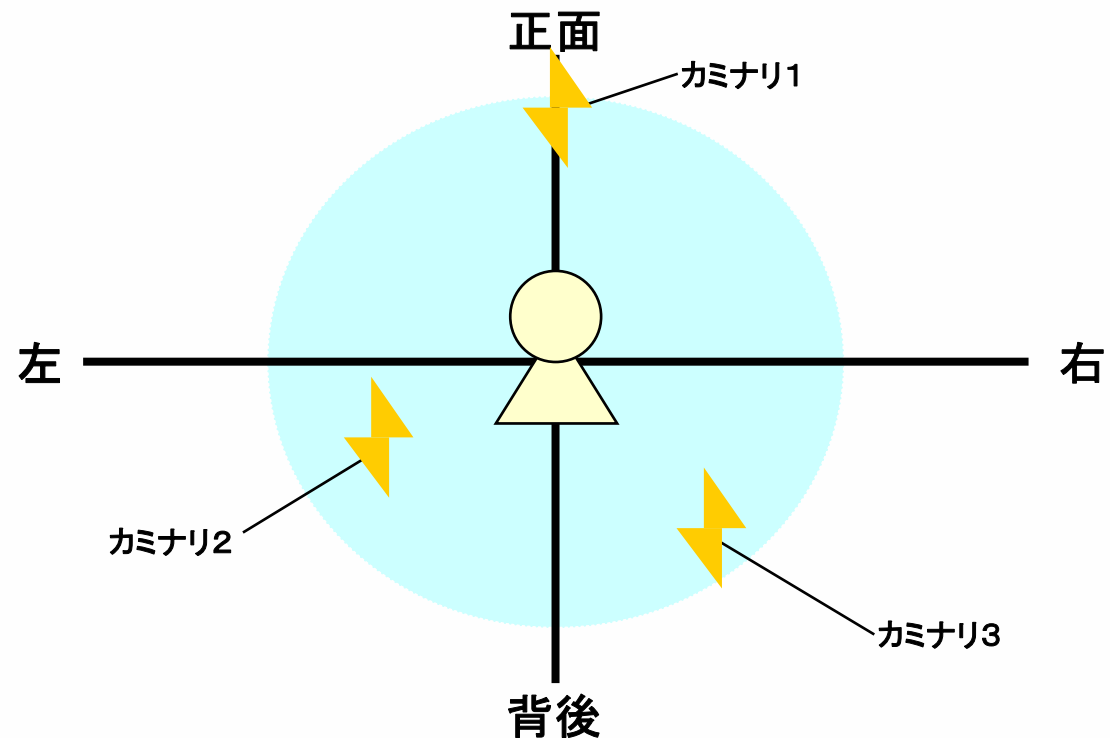
- 異なるプリディレイタイムを設定し、簡易シーケンスを作ることができます。
- AISACのプリディレイタイムにより、プログラマがコントロールできます。
- AISACのプリディレイタイムをランダムイズすることができます。



# コンプレックスシンセによるサラウンド

## ■ 3Dパンニングによる5.1チャンネル効果音

モノラルの効果音から、簡単に5.1チャンネルの効果音を作成できます。

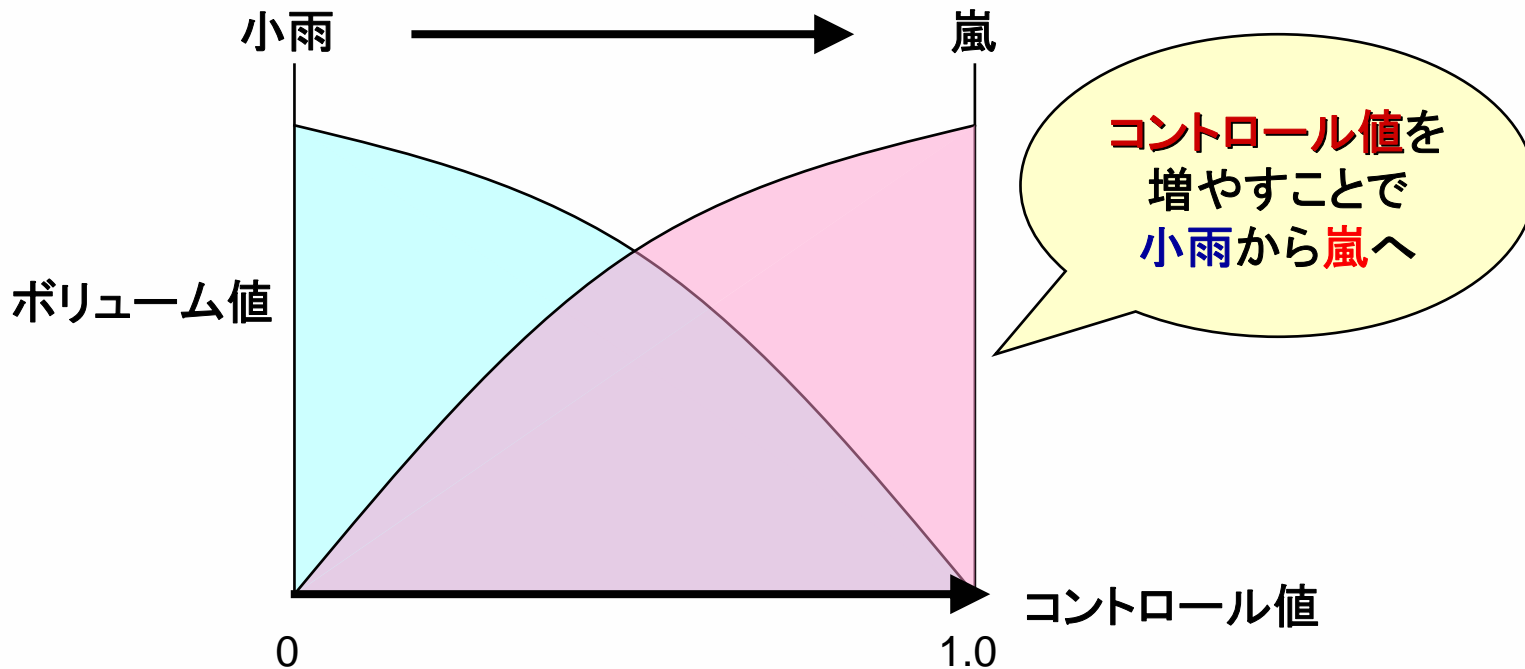




# コンプレックスシンセによるモーフィング

## ■ クロスフェードによる効果音のモーフィング

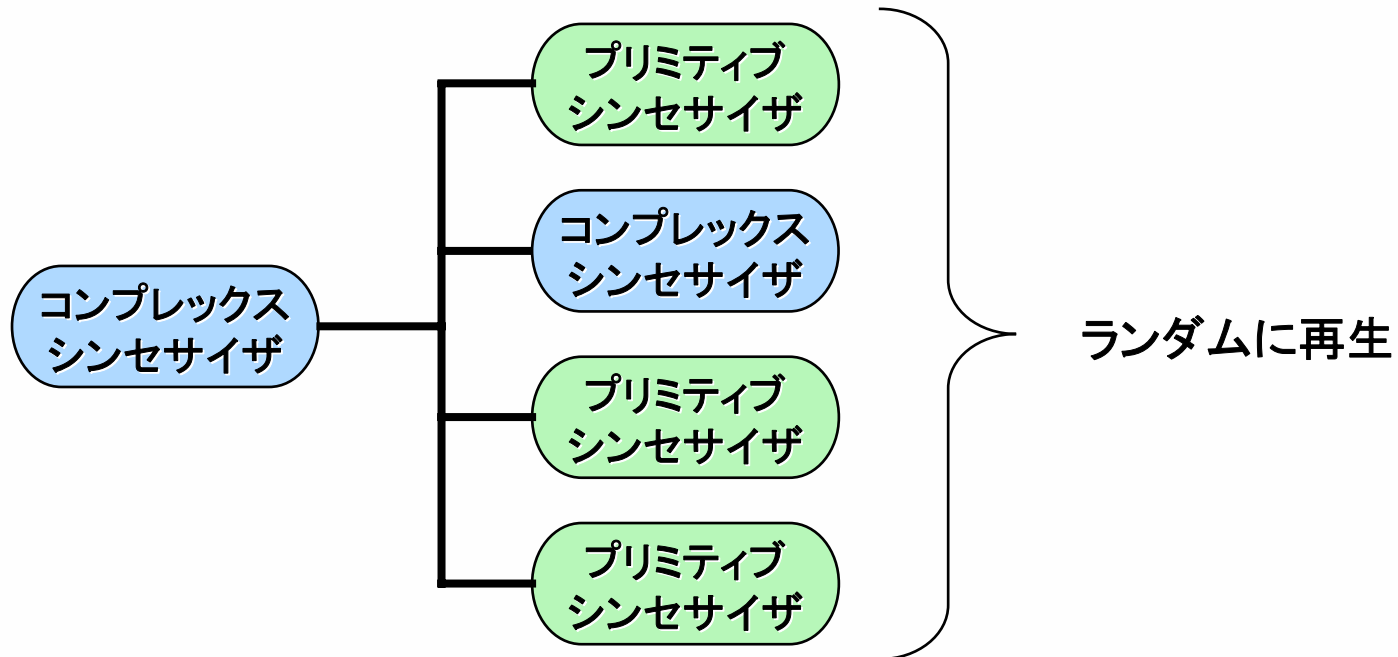
AISACによるクロスフェードを使用して、ある音から別の音へモーフィングできます。



# コンプレックスシンセ(ランダムタイプ)

## ■ ラインダムタイプの動作

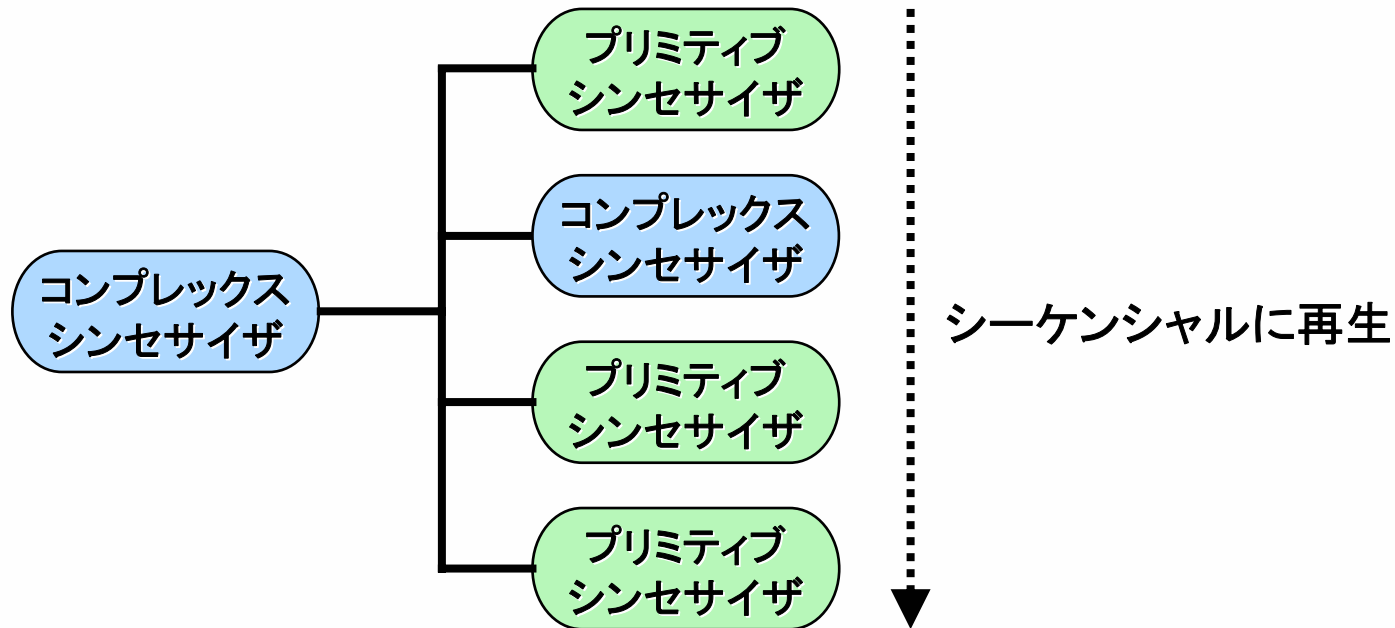
- コンプレックスシンセの所持するシンセの内の1つをランダムに再生します。



# コンプレックスシンセ(シーケンシャルタイプ)

## ■ シーケンシャルタイプの動作

- コンプレックスシンセの所持するシンセを順番に再生します。



# コンプレックスシンセとランダムAISAC

## ■ ランダムAISACによる多彩なバリエーションの実現

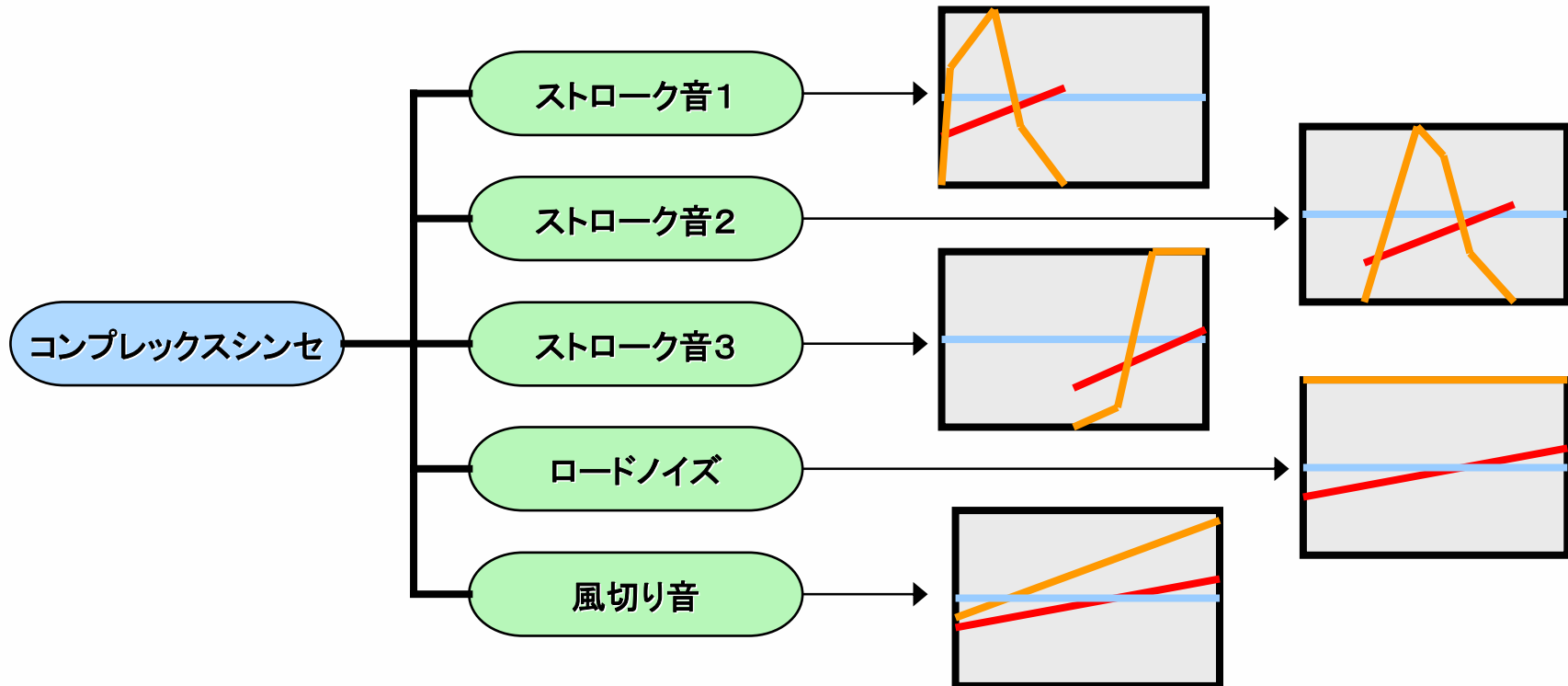
- コンプレックスシンセとランダムAISACを組み合わせることによって、非常にバリエーション豊かな効果音を作成できます。



# 並列 AISAC

## ■ AISACの並列接続

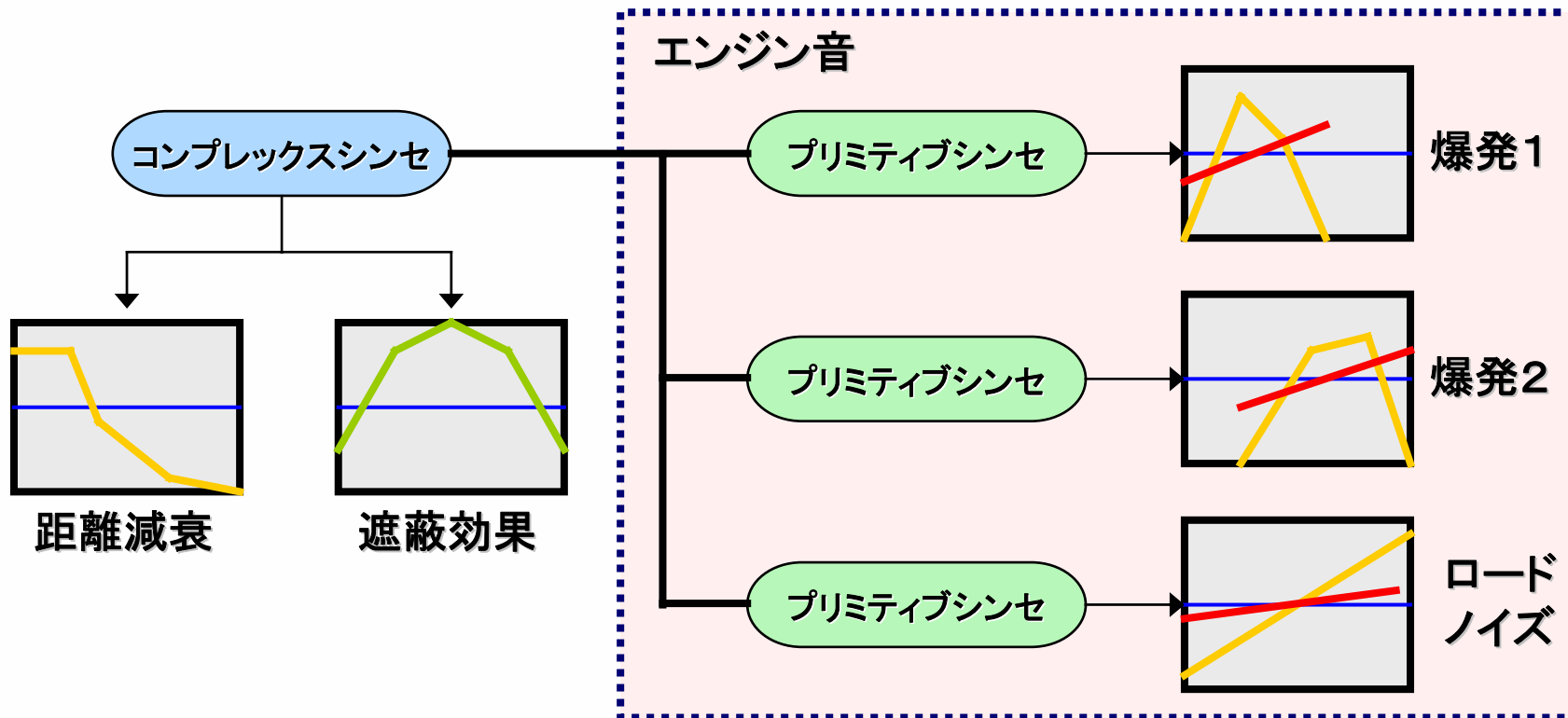
AISACの並列接続を使用すると、ひとつのコントロール値で複数のシンセサイズを制御することができます。(例:エンジン音、インタラクティブミュージック)



# 直列＋並列 AISAC

## ■ 直列AISACによるコンプレックスシンセのコントロール

並列AISACを持つコンプレックスシンセを直列AISACで制御することができます。



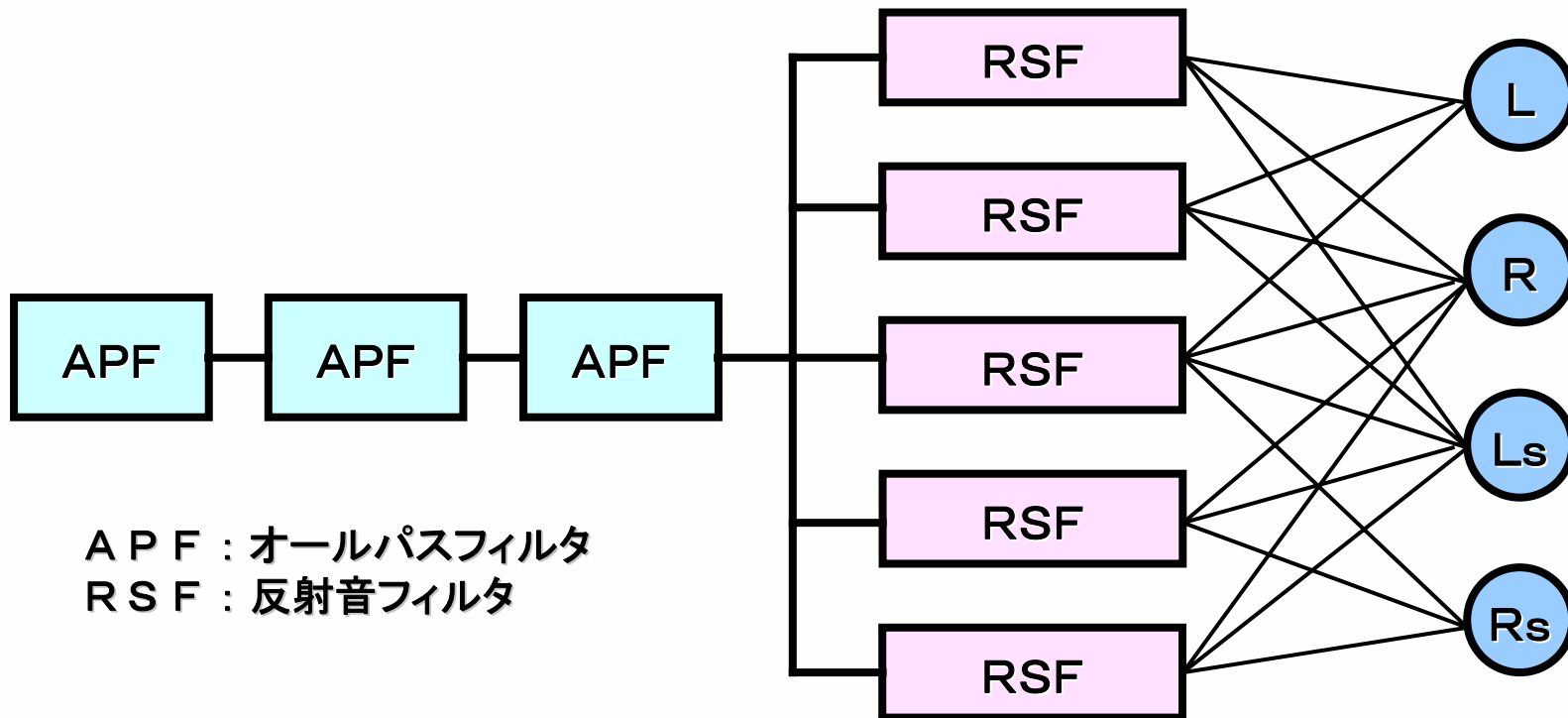


# エフェクト

# 4チャンネルサラウンドリバーブ

## ■ 高品位な4チャンネルリバーブを低CPU負荷で実現

9本の反射音フィルタによって自然なサラウンドリバーブをかけることができます。



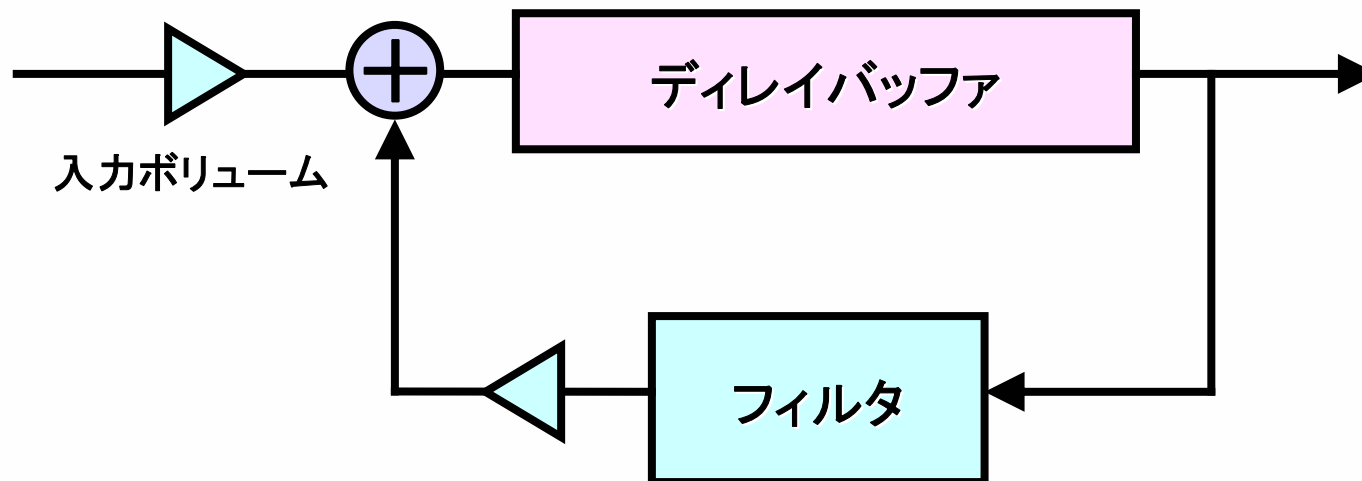
APF : オールパスフィルタ  
RSF : 反射音フィルタ



# ディレイ

## ■ フィードバックディレイ

オーソドックスなエフェクトであるフィードバックディレイをサポートします。



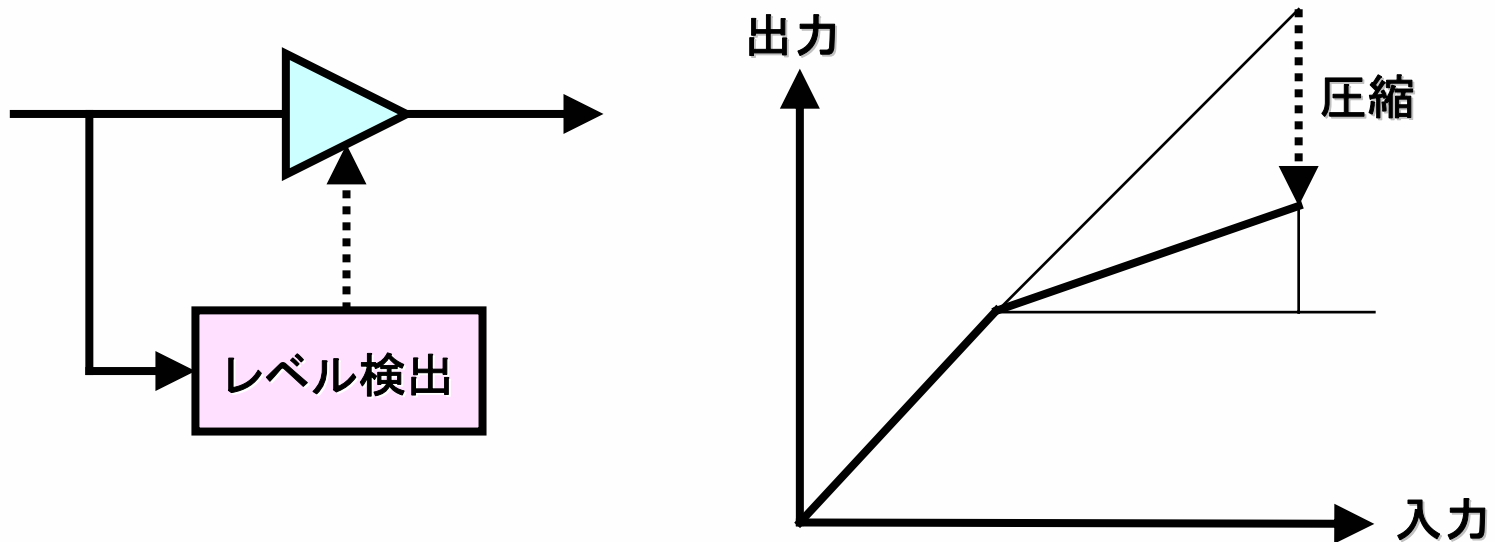
# コンプレッサ

## ■ セリフなどの特定音声のコンプレッサ処理

セリフなど特定の音声に対してのコンプレッサ処理をサポートする予定です。

## ■ 最終出力段でのコンプレッサ

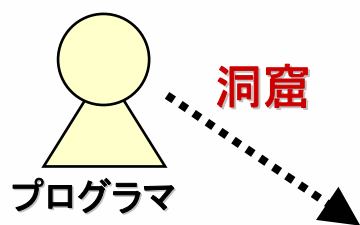
ミックスした音声信号がクリップされないようにコンプレッサ処理を施します。



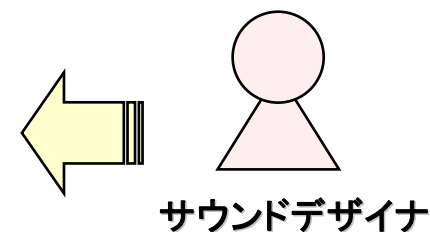
# エフェクトセット管理機能



## ■ エフェクトパラメータの管理

- 残響時間やプリディレイタイムなどパラメータをエフェクトセットとして管理できます。
- プログラマはエフェクトセット名を指定するだけでパラメータを設定できます。



エフェクト名	プリディレイタイム	残響時間	ローパスフィルタ
ガレージ	30	1500	5000
砂漠	0	100	8000
洞窟	50	4000	3000
森	60	2000	6000



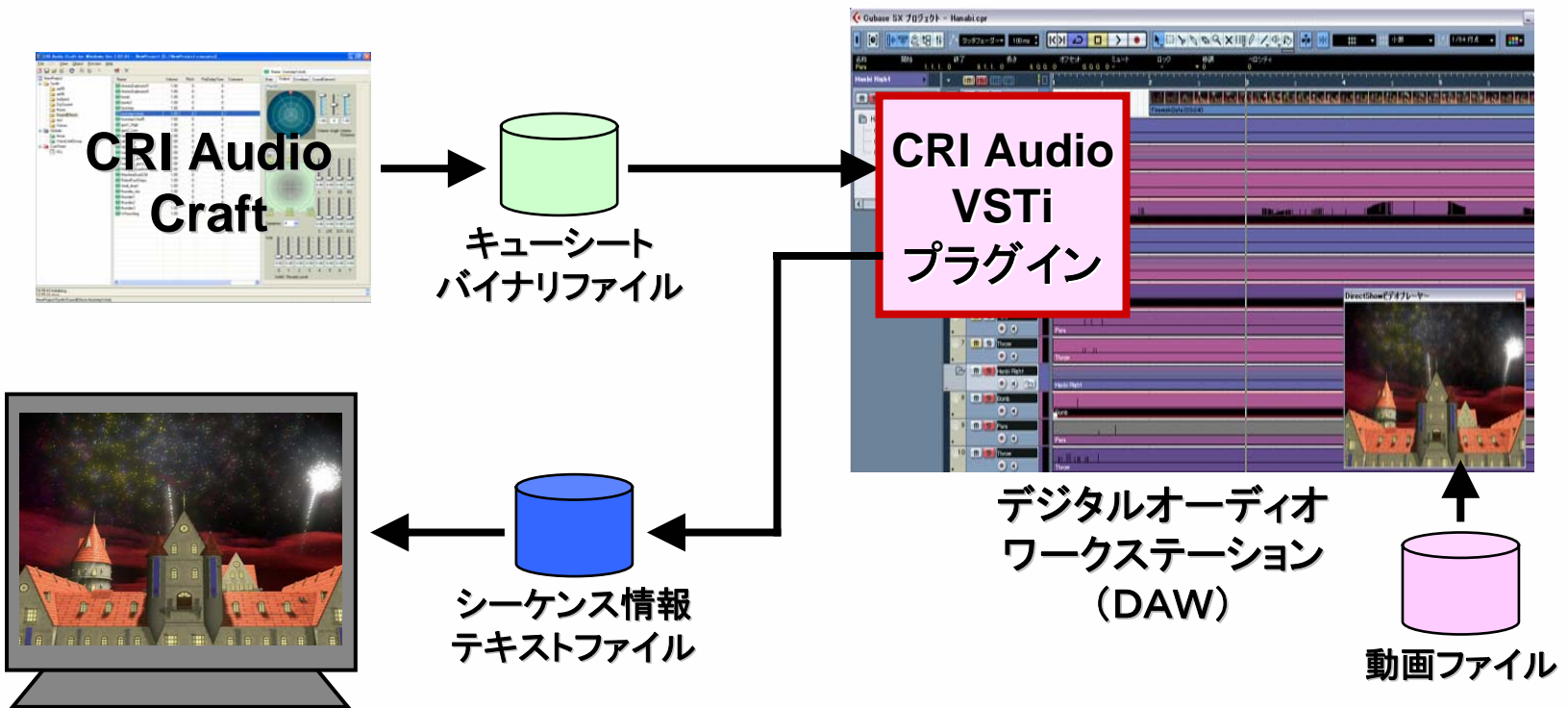


# シーケンス再生

# VSTプラグインによるシーケンス再生

## ■ 市販のDAWでシーケンスを作成可能

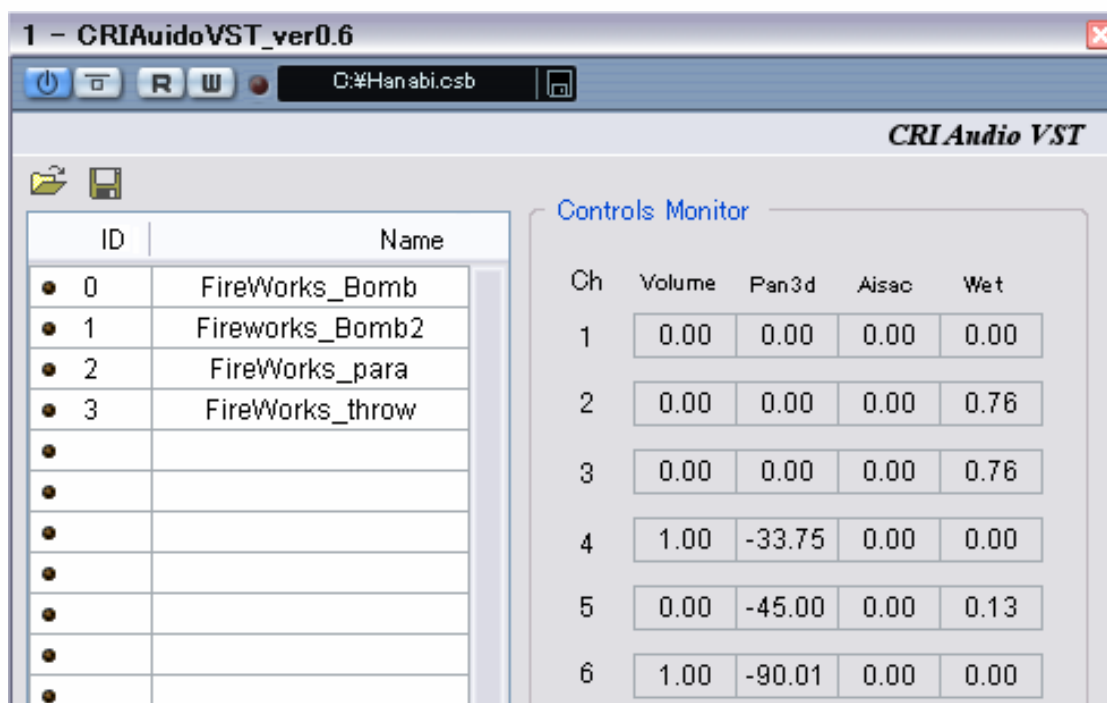
- CRI Audio VSTインストルメント・プラグインを提供します。
- CubaseやSOLなどのDAWからCRI Audioで作成した効果音を再生できます。



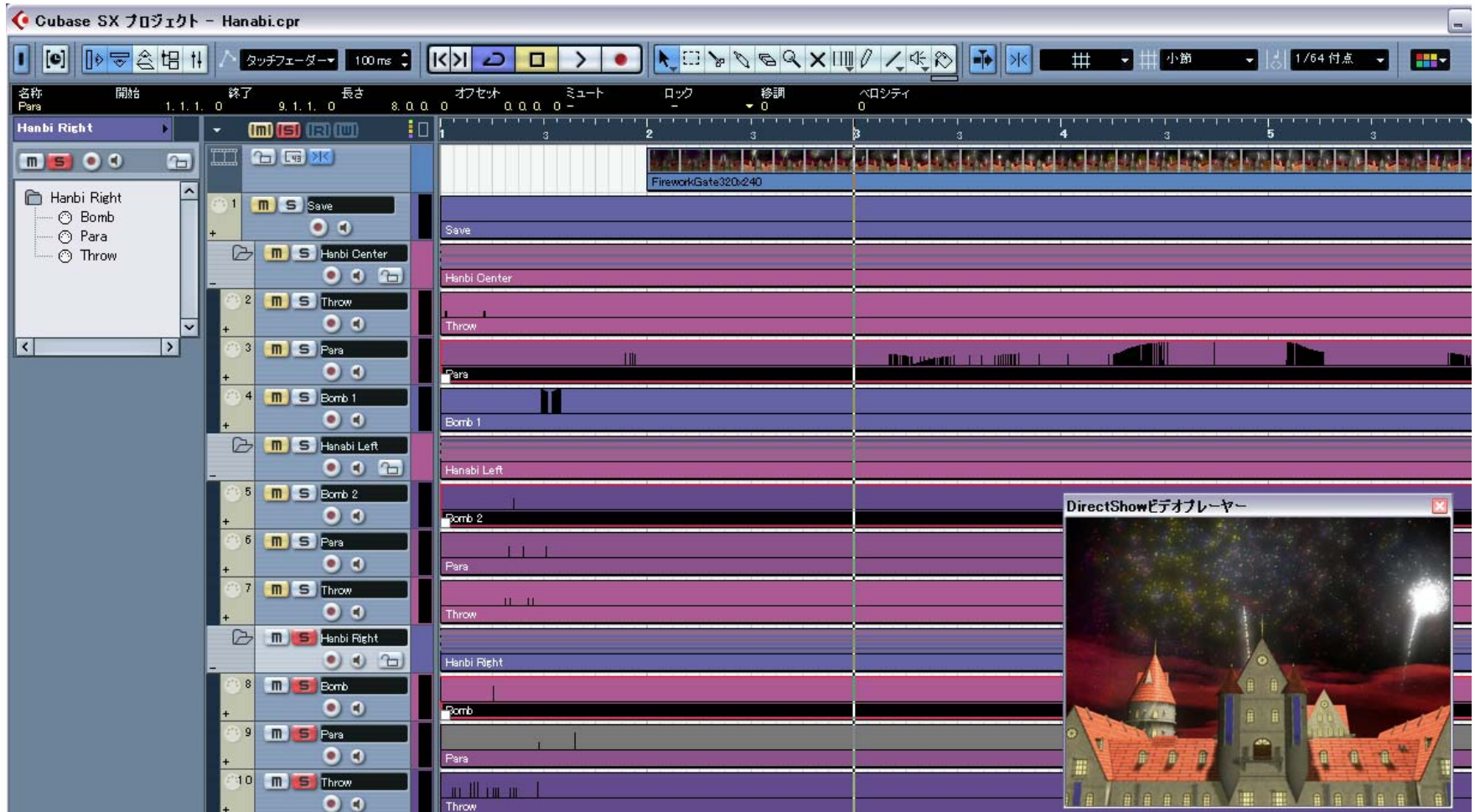
# CRI Audio VSTインストルメント プラグイン

## ■ キューシートバイナリをVSTインストルメントとして再生

- チャンネルにキューを割り当て、ノートONメッセージで再生できます。
- コントロールチェンジによりAISACをコントロールできます。



# Cubase SXによるシーケンスの作成



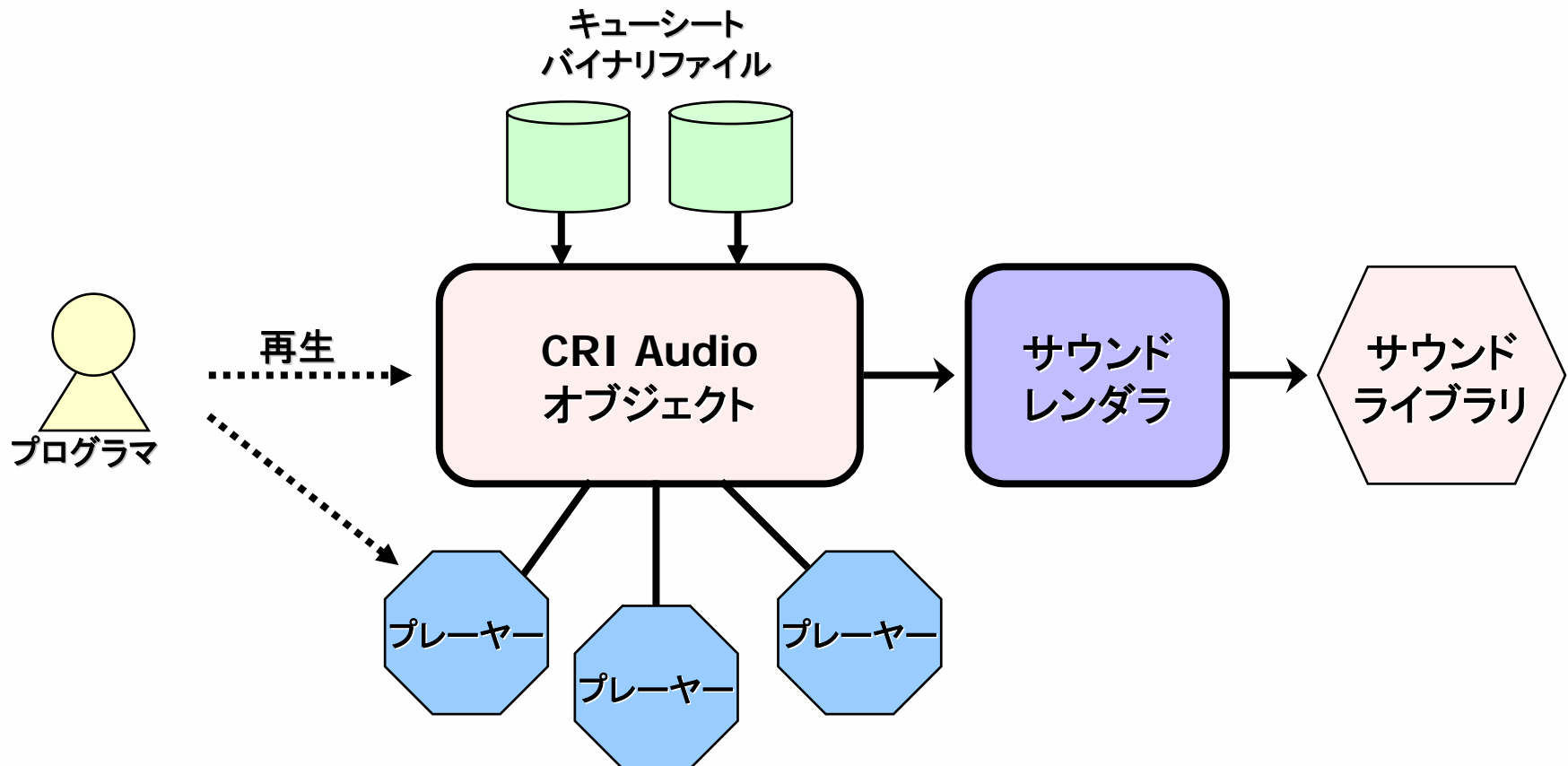


# CRI Audio ランタイムライブラリ



# ライブラリの構造

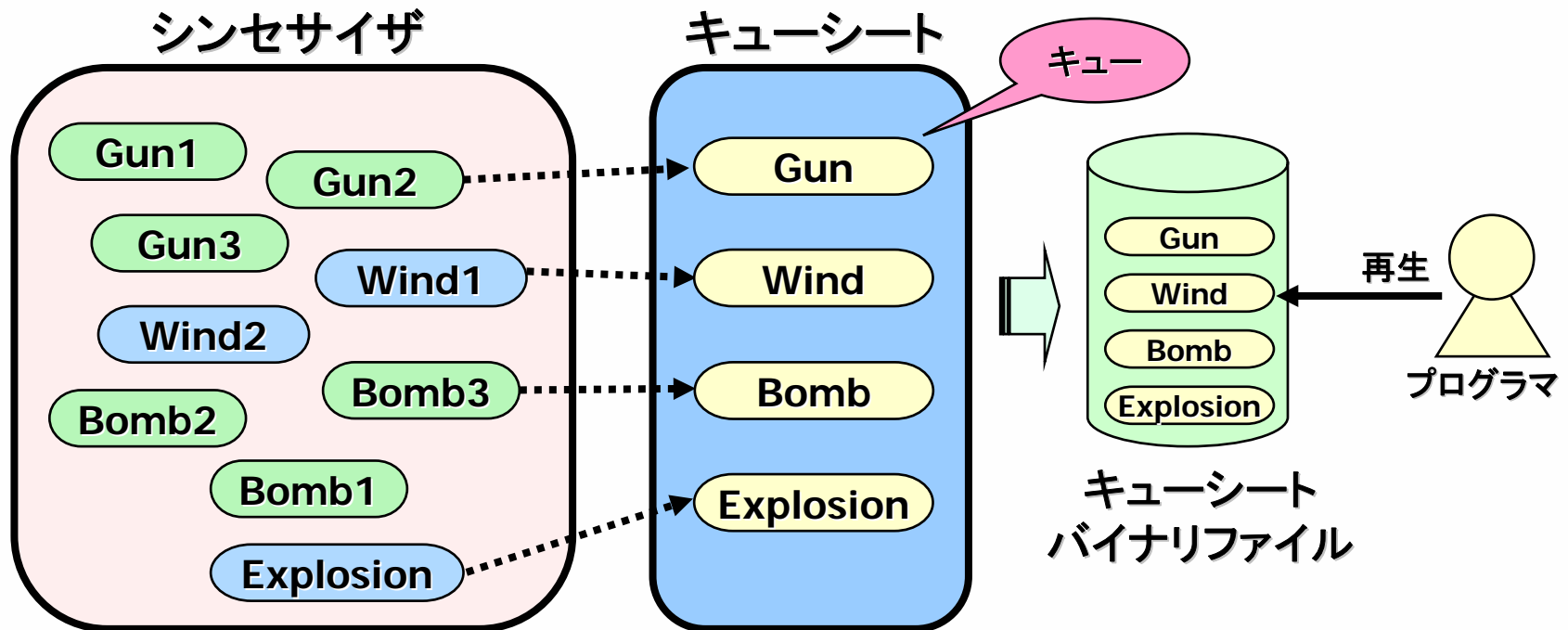
## ■ CRI Audio ライブラリの構造



# キュー

## ■ キュー

プログラマは、**キュー**を指定してサウンドを再生します。  
サウンドデザイナーは作成したシンセサイザを**キュー**として**キューシート**に登録します。  
プログラマは、シンセサイザを意識せずに、簡単にサウンド再生できます。



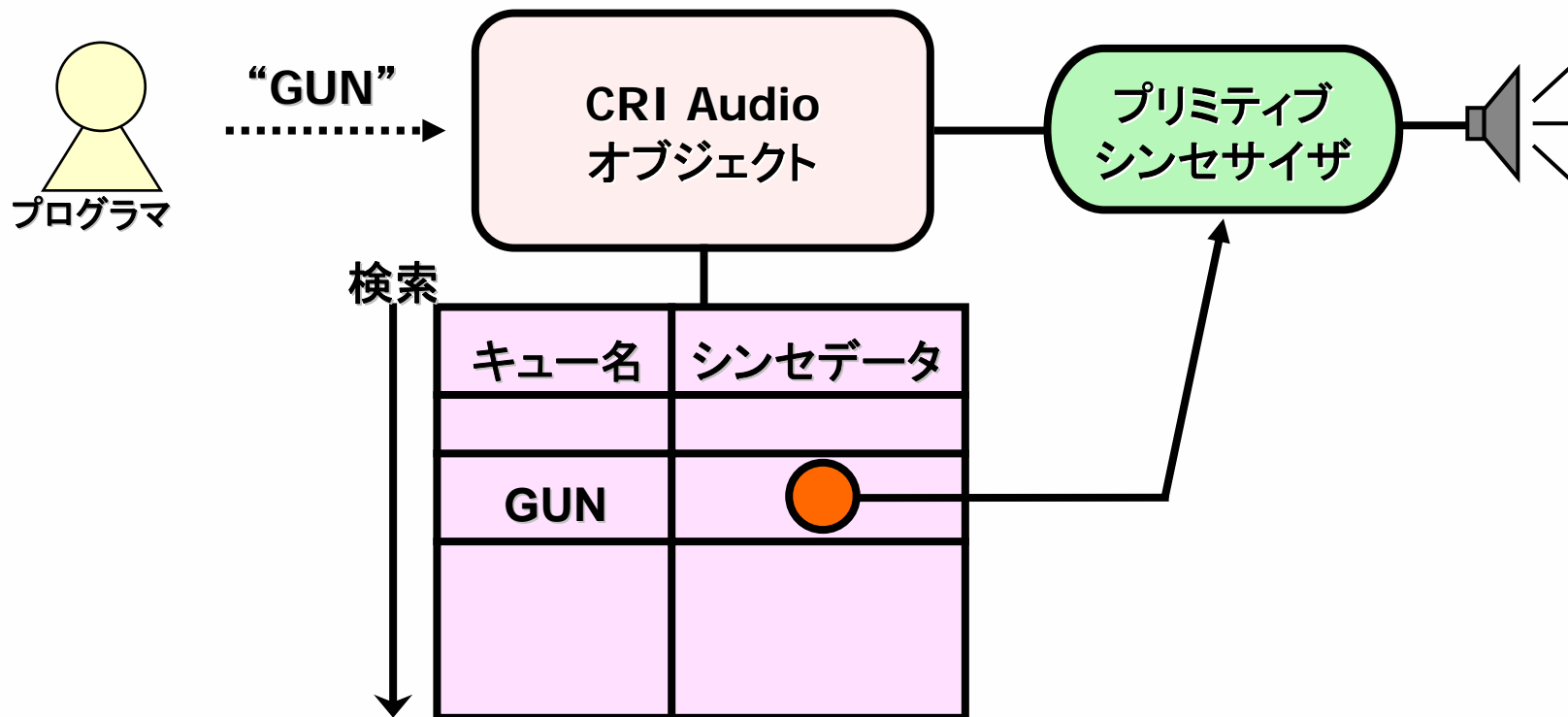
# サンプルコード

- // キューシートの作成
- CriAuCueSheet\* cuesheet = **CriAuCueSheet::Create**(heap, err);
  
- // キューシートバイナリの読み込み
- cuesheet->**StartLoadCueSheetBinaryFile** ("sample.csb", err);
  
- // CRI Audioオブジェクトの作成およびキューシートの登録
- CriAuObj\* auobj = **CriAuObj::Create**(heap, sndrhdr, "Sample", err);
- auobj->**AttachCueSheet**(cuesheet, err);
  
- // 効果音の再生
- auobj->**Play**("GUN", err);
  
- // 再生状態の監視
- for (;;) {  
    CriAuObj::**ExecuteMain**(err); // 内部状態の更新  
    if ( auobj->**GetPlaybackStatus**(err) == auobj->PLAYBACK\_STATUS\_STOP )
- break;
- }

# CRI Audioオブジェクトによる再生

## ■ CRI Audioオブジェクト

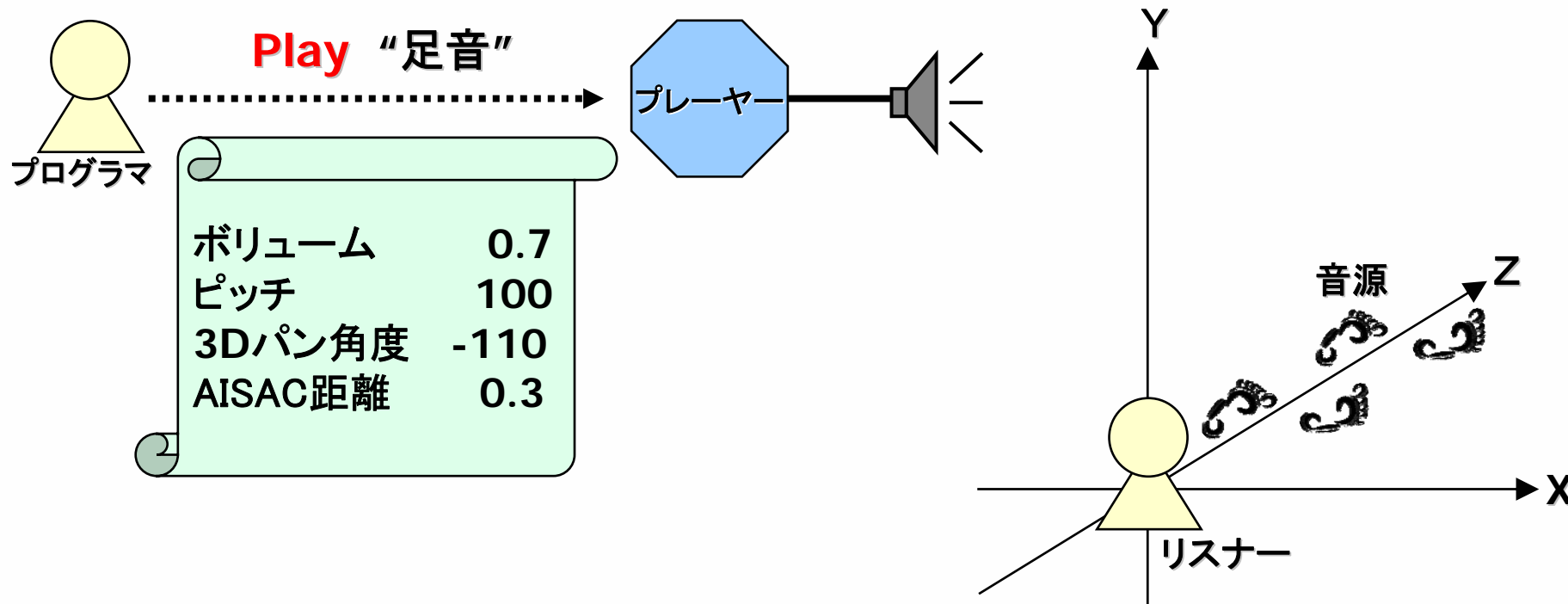
CRI Audioオブジェクトは再生指示を受けると、キューシートを検索し、キューにリンクしているシンセサイザを再生します。



# CRI Audioプレイヤーによる再生

## ■ CRI Audioプレイヤーによるパラメータ指定

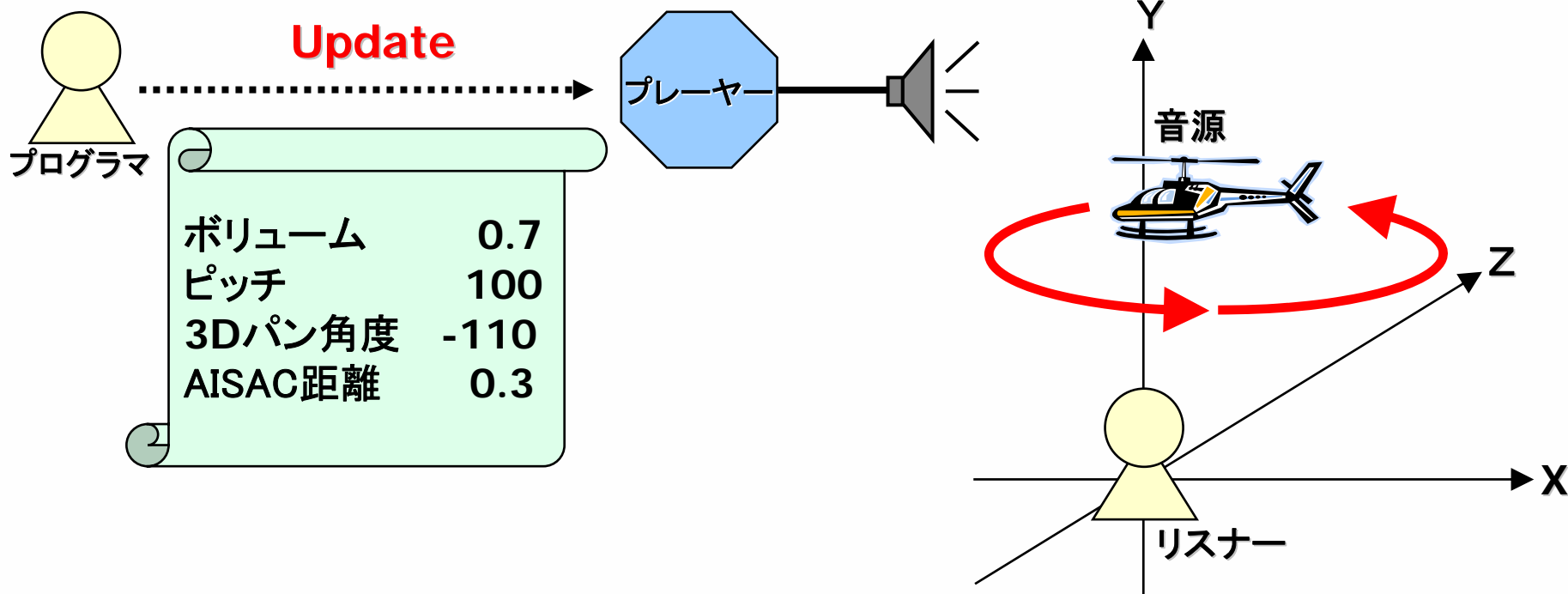
プレイヤーによって、音量やピッチ、AISACのコントロール値を設定できます。



# リアルタイムコントロール

## ■ サウンドのリアルタイムコントロール

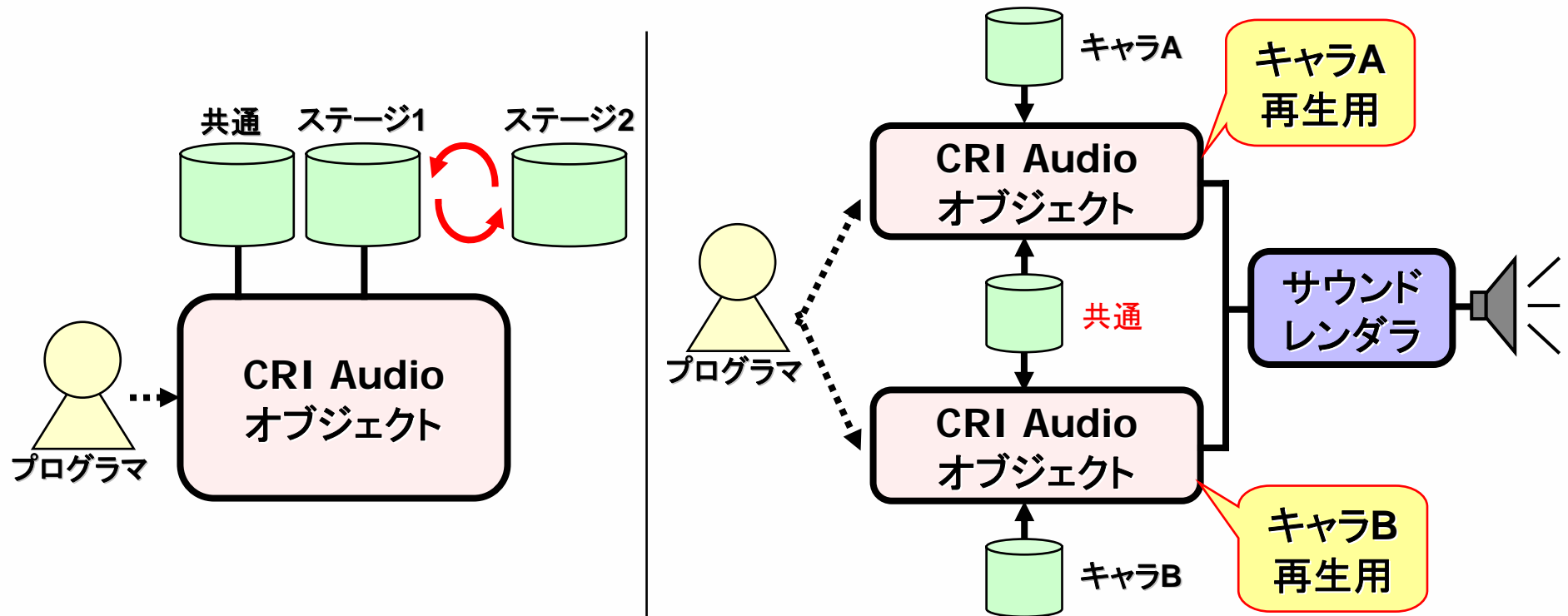
再生中のプレイヤーに対して、音量やピッチなどパラメータを設定できます。



# 複数キューシートのロード

## ■ 複数キューシートの検索

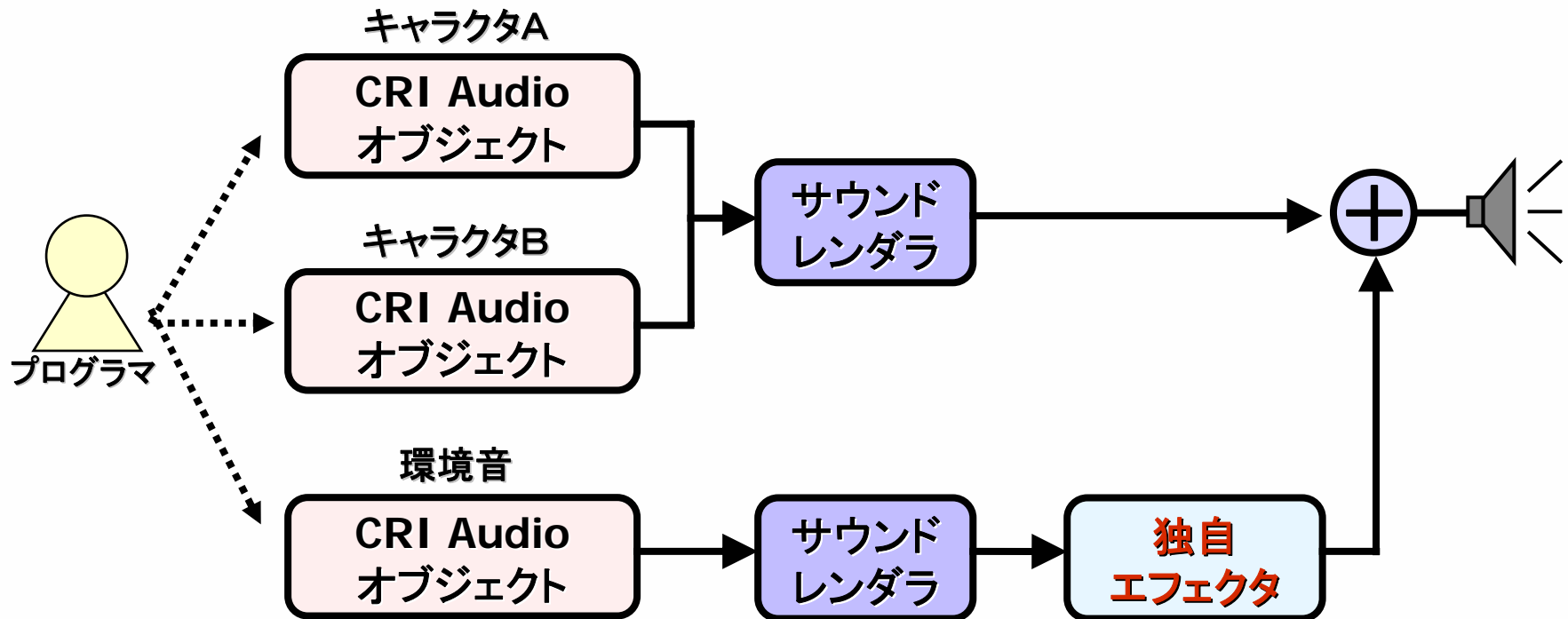
- CRI Audioオブジェクトには複数のキューシートを登録できます。
- ステージ固有のキューシートだけを変更することができます。



# フレキシビリティ

## ■ 構成のフレキシビリティ

- 用途に応じて様々な構成を作ることができます。
- サウンドレンダラの出力に独自のエフェクタを追加することもできます。





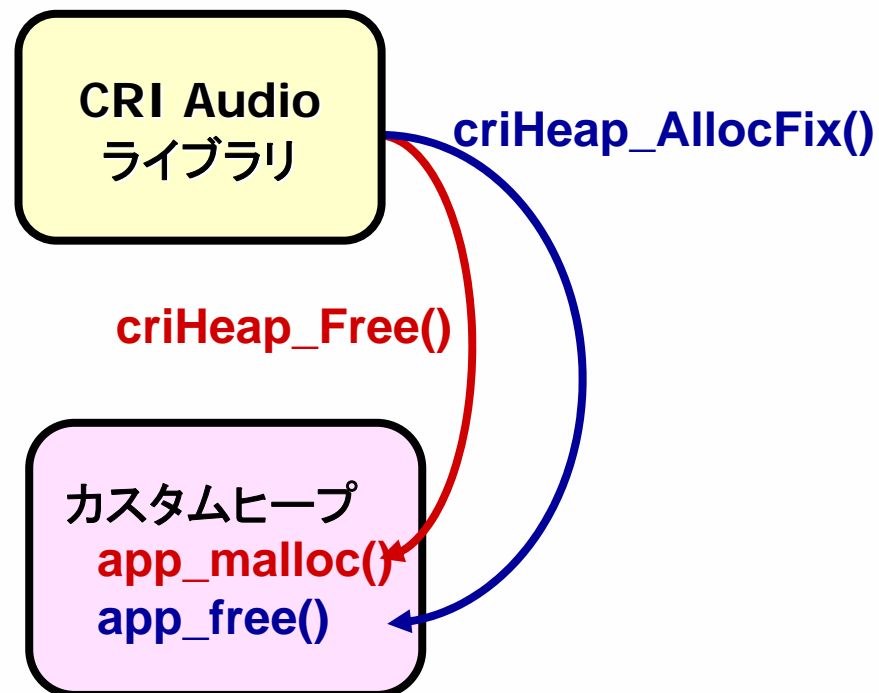
# メモリ管理システム(ヒープ)のカスタマイズ

## ■ ヒープ関数群の置き換え

CRI Audioライブラリの使用するヒープ関数群を置き換えることができます。

### <カスタムヒープの指定>

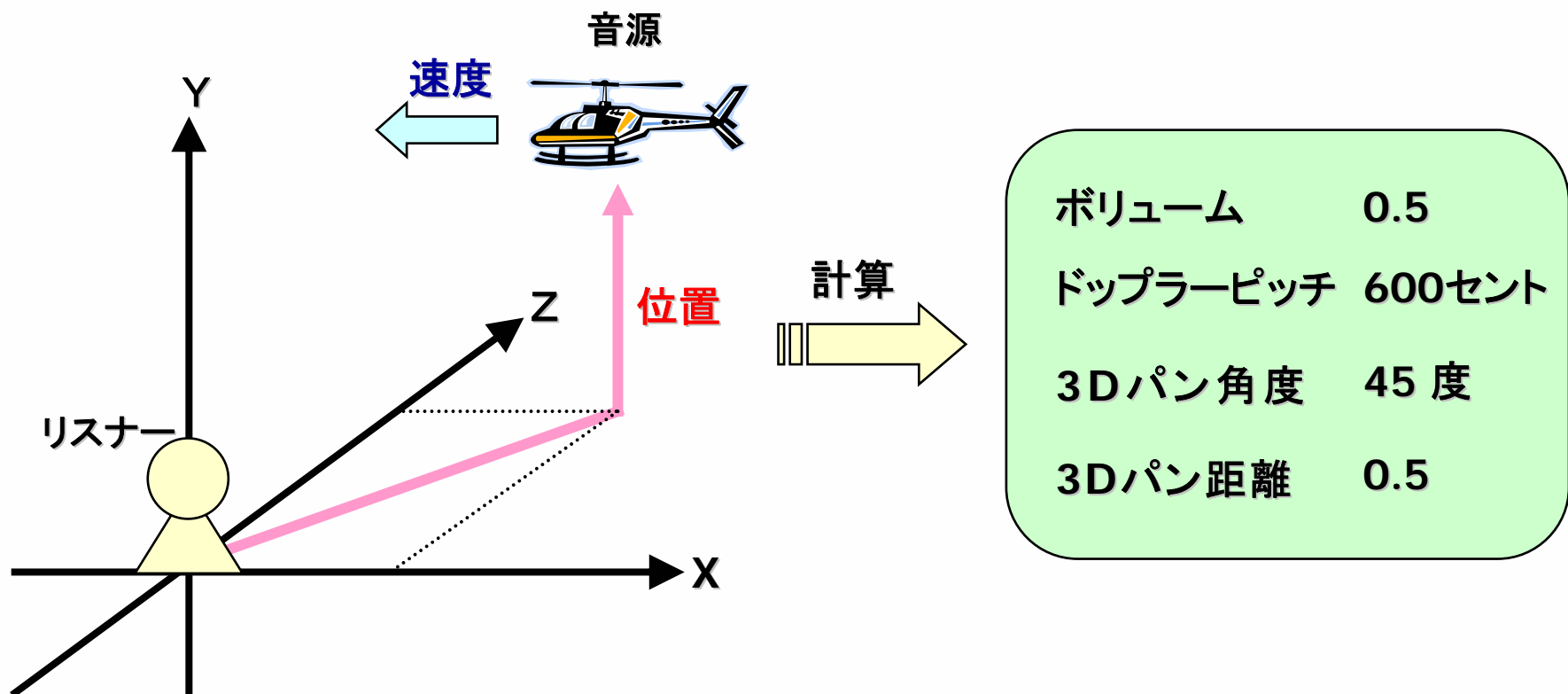
```
CriHeap heap;  
  
heap = CustumHeap_Create(ptr, size);  
  
CriAuObj::Create(heap, sr, name);  
:  
:
```



# 計算による3Dポジショニング

## ■ リスナーとサウンドソースからの再生パラメータの計算

位置や速度などからドップラーピッチや距離減衰を計算します。



# ユーザーデータの埋め込み

## ■ サウンドデザイナーからプログラマへ付加情報を伝達

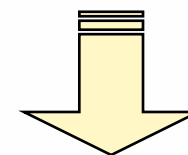
- サウンドデザイナーからセリフのテキストなどの情報をプログラマへ渡すことができます。
- キュー単位でテキスト情報として設定することができます。

### <ユーザーデータの設定>

Name	UserData
DialogA01	おはようございます
DialogA02	おはよう
DialogA03	今日は何をなさいますか
DialogA04	そうだな...

### <ユーザーデータの取得>

```
Char8 *text;  
  
text = csht->GetUserData("DialogA02");
```

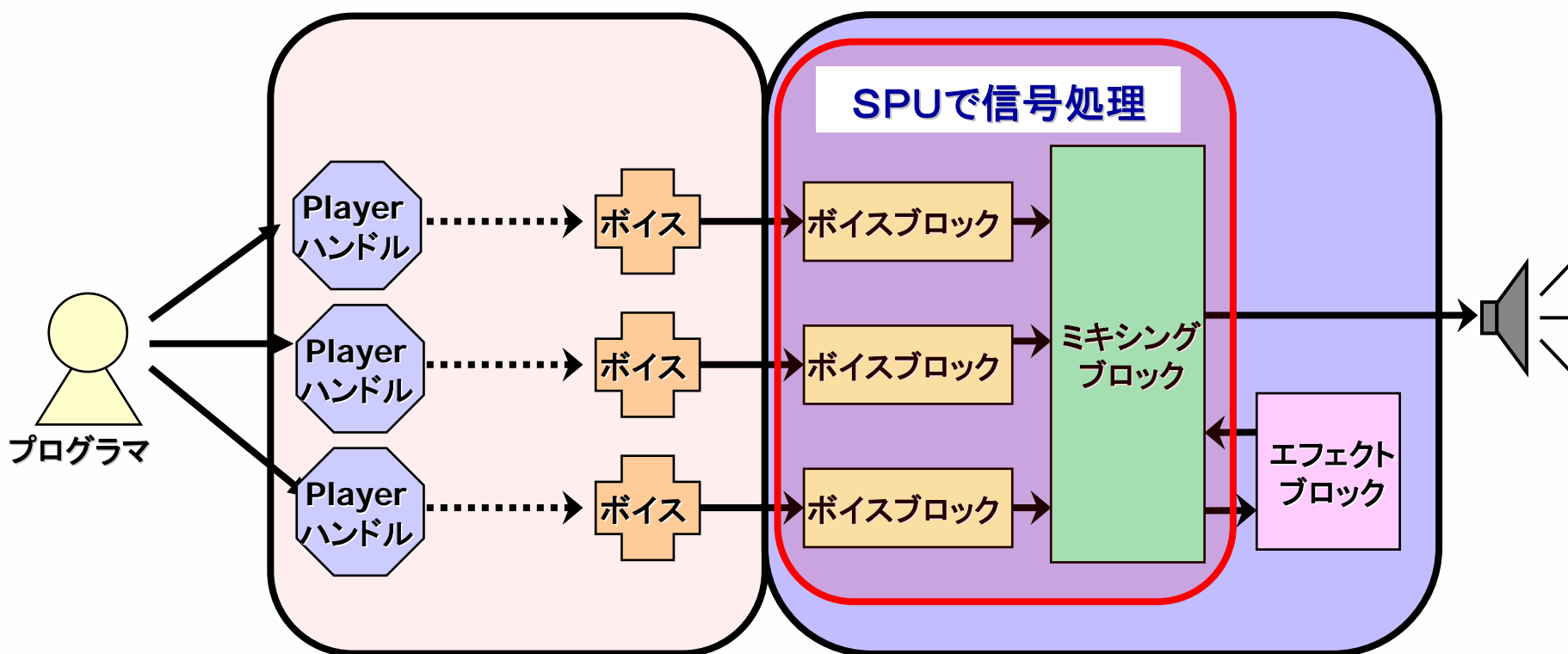


**text**には”おはよう”が代入されます。

# PlayStation®3版CRI Audio

## ■ SPUによるシンセサイズ処理

デコードやフィルタ・ミキシング処理などをSPUで行うため、PPUに負担をかけません。

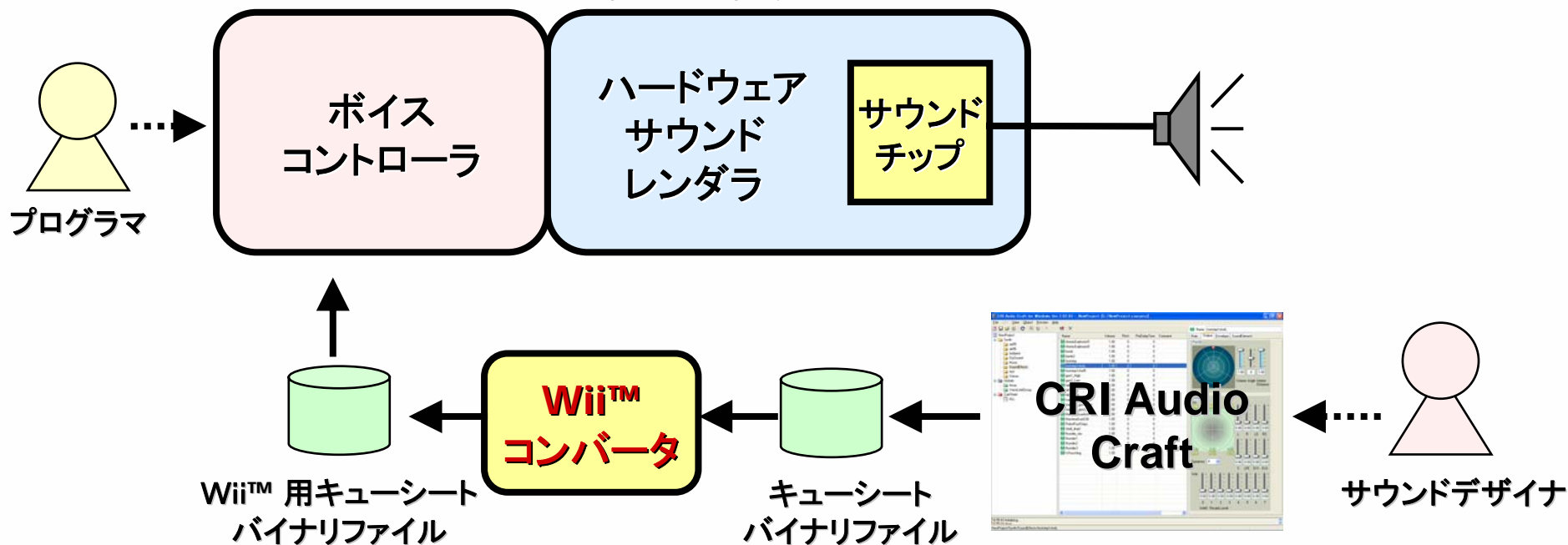


# Wii™版CRI Audio

## ■ ハードウェア版サウンドレンダラ

Wii™の持つサウンドハードウェアを利用するサウンドレンダラを検討中です。

### CRI Audioライブラリ



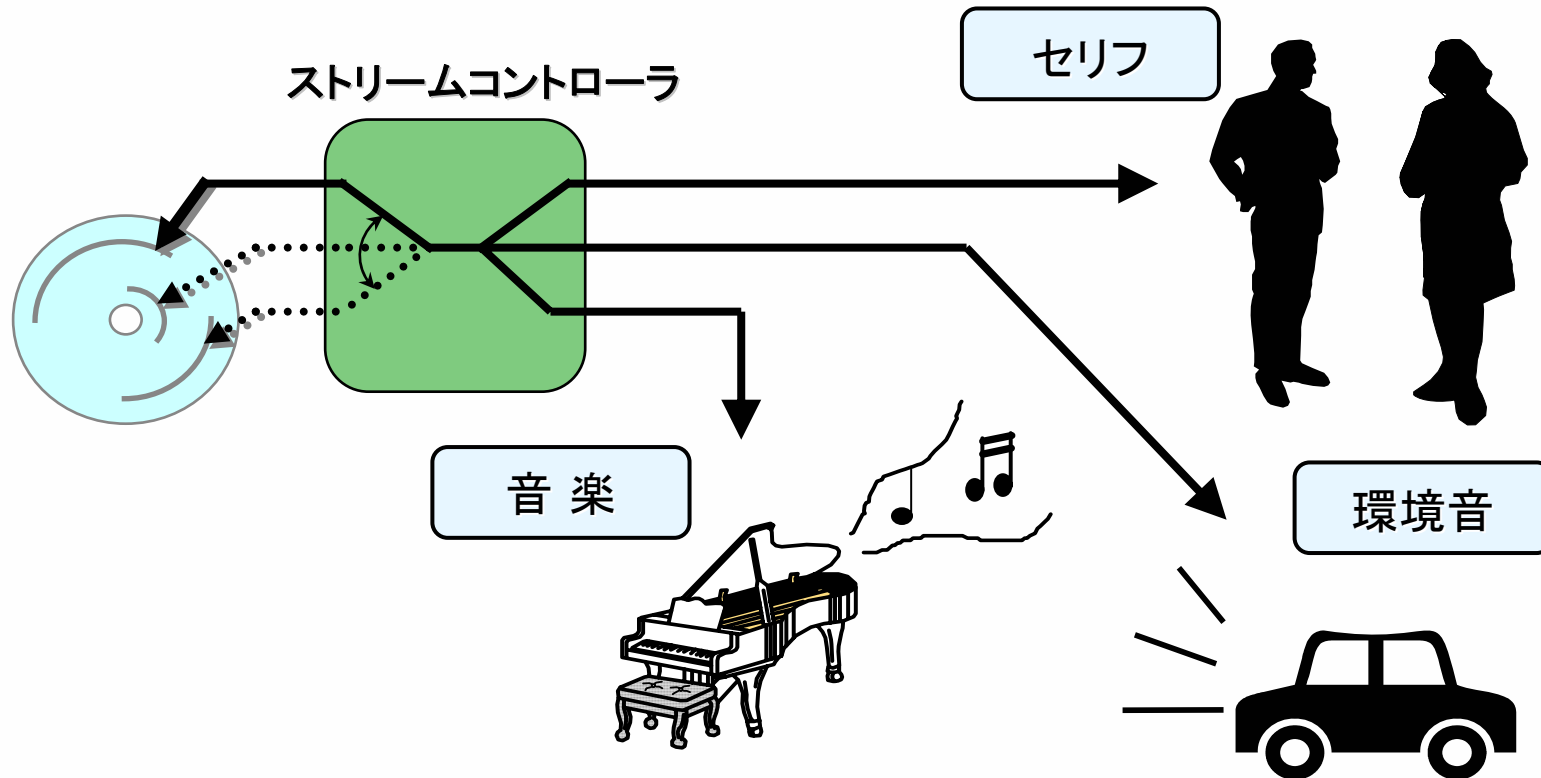


# ストリーミング

# マルチストリーミング再生

## ■ CRI ADXを継承する安定したマルチストリーミング

複数の音声ファイルを途切れることなくストリーム再生します。



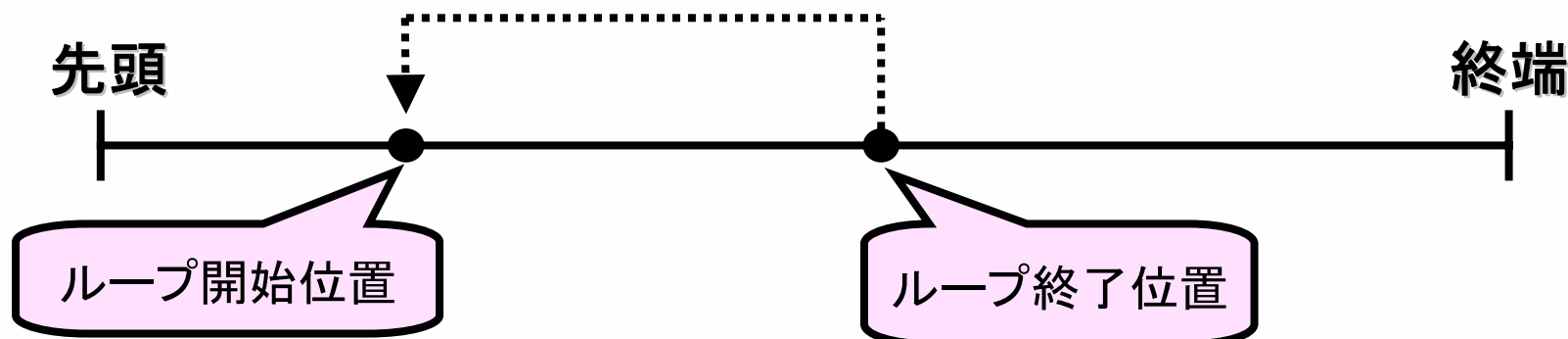
# シームレスループ再生

## ■ 圧縮音声ファイルのシームレスループ再生

圧縮音声ファイルでありながら、完全なシームレスループ再生ができます。

## ■ サンプル単位でのループ再生

サンプル単位でループポイントを指定できます。





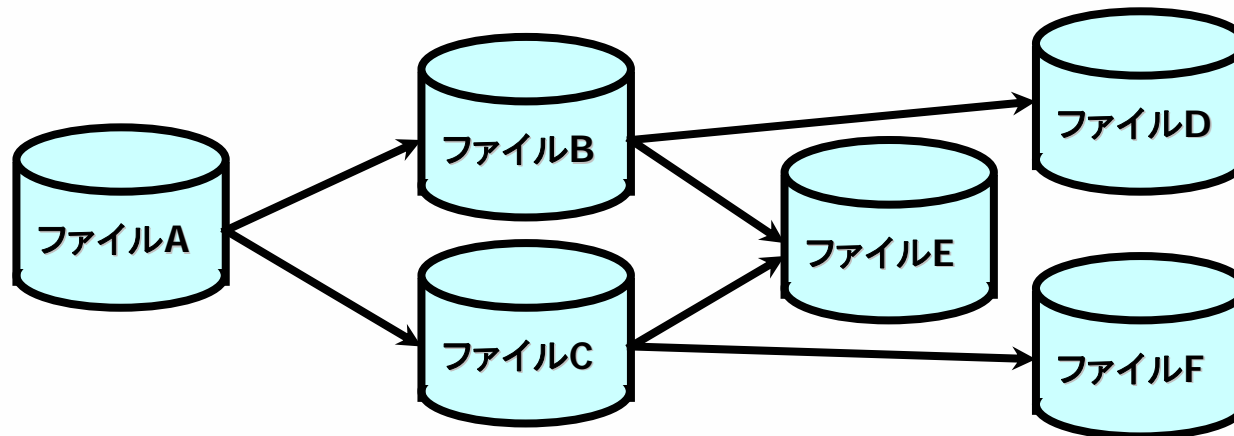
# シームレス連結再生

## ■ 圧縮音声ファイルのシームレス連結再生

音声ファイルをシームレスに接続し再生します。

## ■ ダイナミックエントリー機能（分岐再生）

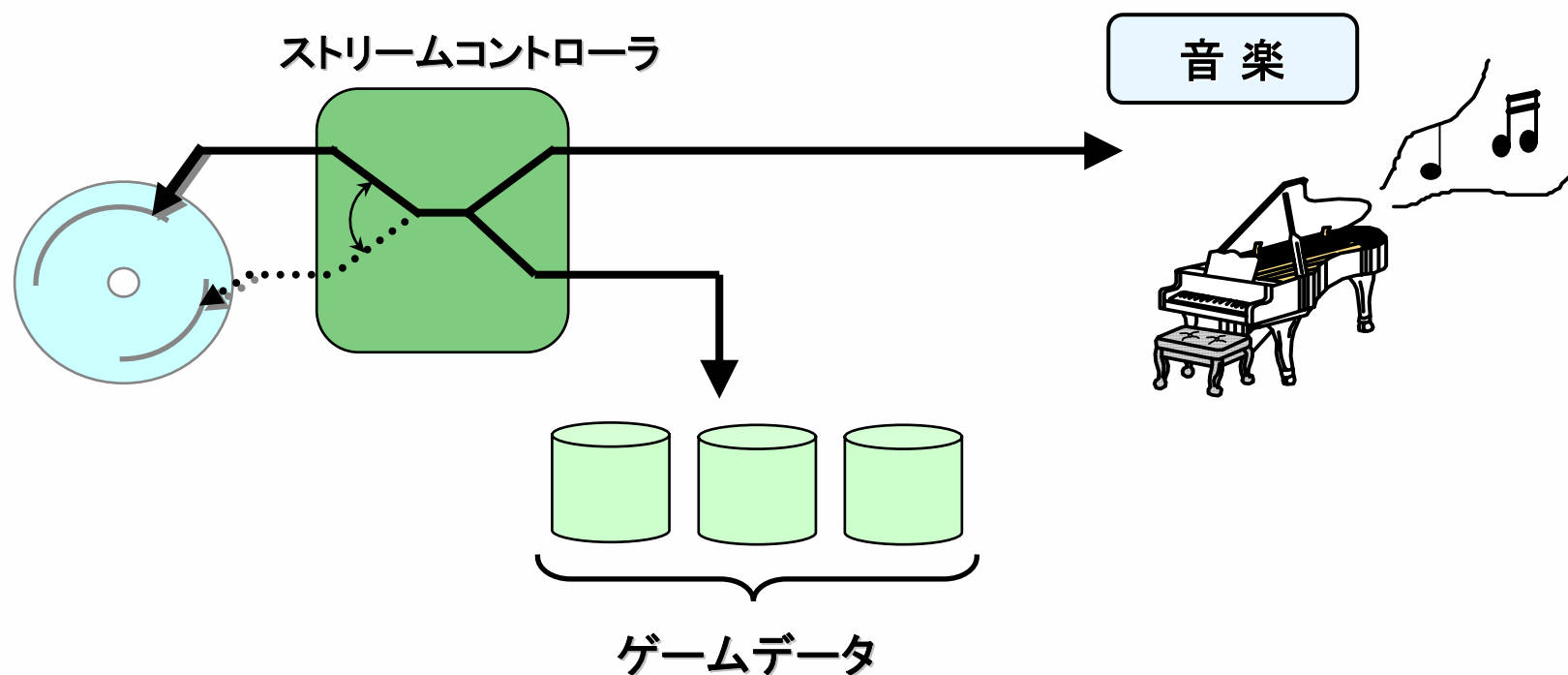
ゲームの状況に応じて、再生中に次の再生するファイルを登録できます。  
スポーツゲームの実況中継などに利用できます。



# データの裏読み

## ■ バックグラウンドでのファイル読み込み

音声ファイルを再生中に、バックグラウンドでデータファイルを読み込みめます。  
バッファリングを最適化し、高速にデータロードすることが可能です。



# CRI File System

## ■ ストリーミング指向ファイルシステム

- オープン・クローズの負荷がありません。
- 読み込みプライオリティを設定できます。

## ■ 大量ファイルのパッキング

- 大量のデータファイルをパッキングできます。

## ■ データ圧縮

- データを圧縮してパッキングでき、ディスクスペースを節約できます。
- ロード時に高速展開し、読み込み時間を短縮できます。

## ■ ファイル配置

- 高速にデータを読み込むために、ファイルの配置をコントロールできます。

# まとめ

- プリミティブシンセサイザ
- コンプレックスシンセサイザ
- AISAC (アイザック)





# お問い合わせ先

**CriAudio**@cri-mw.co.jp