サウンドゲームとプロシージャル

オープランニング
大野 功二

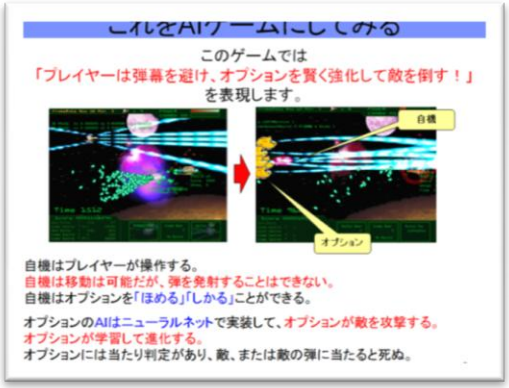
2009/09/06 v1.01

ご挨拶

オープランニングでは、
様々なゲームのお仕事をさせていただいております。

ゲーム業界で10年以上の経験と実績があり、
元プログラマーであるノウハウを活かして、
現実性の高いプランニングをモットーとしております。

さらに近年では研究開発も積極的に行っており、
IGDAのセミナーやCEDECの講演などにも
積極的に参加させていただいております。



講演の内容

- 1.プロシージャルとは？
- 2.サウンドゲームとは？
- 3.サウンド(音)について
- 4.音楽解析と映像演出
- 5.音楽(BGM)を解析しよう
- 6.音楽解析を使ったプロシージャル技術のゲームへの応用
- 7.まとめ

補足

- 講演時間は60分です。
- たくさんの情報をお伝えするために、駆け足で説明します。
- 資料は、CEDEC終了後にCEDECホームページのアーカイブ、およびオープランニングのホームページよりダウンロード可能です。
- その他、テストプログラムなどを実行したムービー資料もCEDEC終了後に公開する予定です。
- 重要なことは上記資料にてアップしますので、
  無理にメモを取らなくても大丈夫です。
  肩の力を抜いて、講演を聴いていただければと思います。

1.プロシージャルとは？

プロシージャルとは？

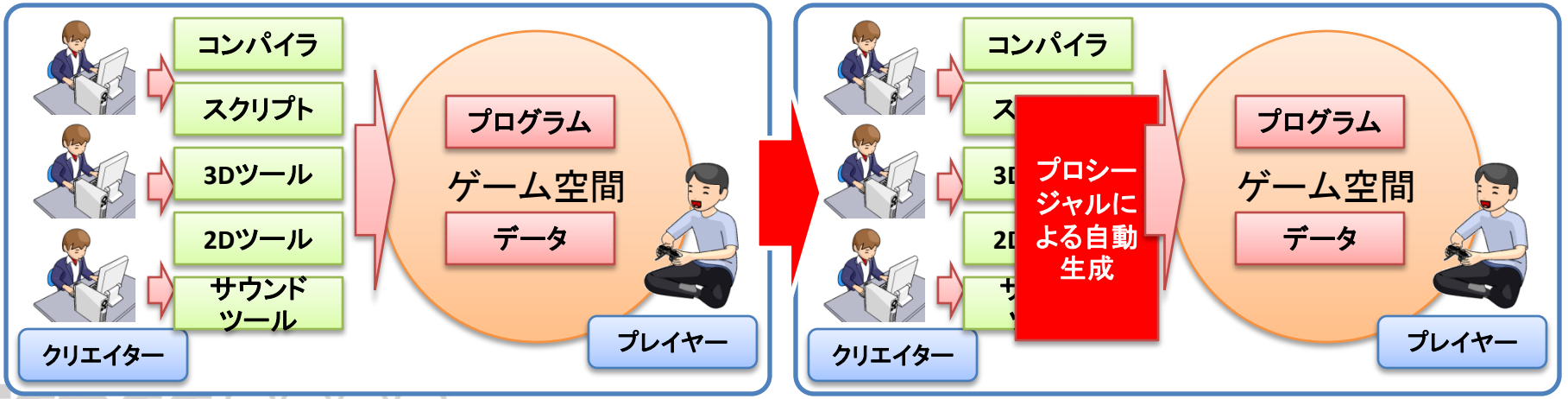
- プロシージャル(Procedural)とは？

「ゲーム空間、デジタル空間において、
自身の連続操作によって自律的な力を持つファクター」
(CEDEC 2008 IMAGIRE DAY(3) ゲーム開発のためのプロシージャル技術の応用　三宅　陽一郎氏)

「簡潔なアルゴリズムの組み合わせで、複雑な処理を実現」
(CEDEC2008 IMAGIRE DAY(4) プロシージャルグラフィックス - 理論と実践　今給黎　隆氏)

- プロシージャルの役割

人がツールなどによって作り出す「データ」や「アルゴリズム」などを、自動処理化する。



プロシージャルの使用例

Planning

- 使用例
  - ・テクスチャーなどのデータ自動生成
    景観作成ツールや3Dモデリングツールなど
    VUE(e-one software)

  - ・ゲームのマップで使う「木」などのモデル作成や配置の自動化
    FARCRY2(Ubisoft)

- 代表的なゲーム
  - ・SPORE
    世界環境、キャラ、音楽などを
    プロシージャルで自動生成

プロシージャルのメリット・デメリット

Planning

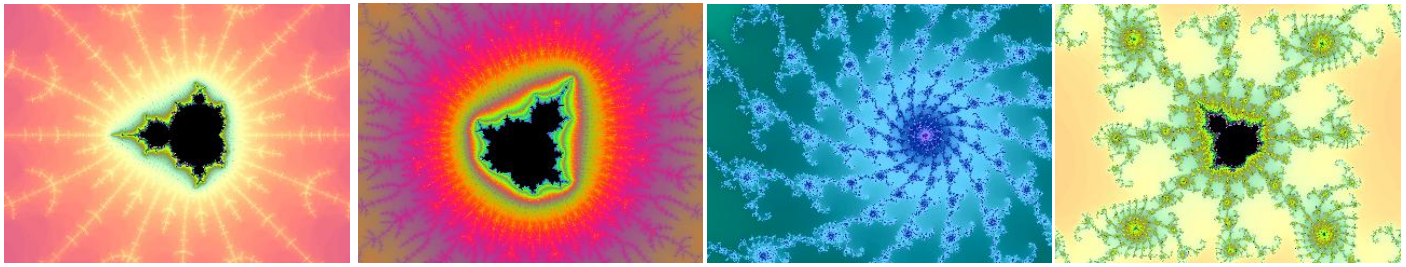
- プロシージャルのメリット

・少ないデータとアルゴリズムで、膨大な量のデータを生成できる。

・突発的に人間が意図しないデータができる場合がある。

- プロシージャルのデメリット

・複雑な演算によるプロシージャルは、計算やデータ処理など処理に時間がかかる。
(例：マンデルブロ集合など)



・人が心地よいと思う結果を出せるまでの調整に時間がかかる。
(データ作成コストが下がるワケではない)

・どんなにユニークなパラメータを入れても、その数式の限界を超えるような
人間的発想でデータが作成されることはほとんどない。
(マンデブロ集合でどんなにがんばっても、「ドラえもん」は図形に現れない)

プロシージャルの本質を掴もう！

- この講演の目的

プロシージャル技術の本質を掴み、
よりよいゲームを作るためのノウハウを持ち帰って頂くことです。

- プロシージャルの本質を掴まずにゲームを作ると・・・・・・

企画　「CEDEC 2009でプロシージャルを使えば、
　データを簡単に自動で作れるらしいぞ！」
プログラマ　「よし、じゃ早速やってみよう！！」

そして、3ヶ月が経過。
プロシージャル技術の現実が見えはじめると・・・・・・

企画　「なんだよ。全然面白いデータができないじゃん。
　これじゃ、ランダム生成と変わらないよ！！」
プログラマ　「仕様書通りに作ったよ！！　企画と仕様が悪いんだよ！！」

お約束のデスマーチへ。そして、どつきあい。

2.サウンドゲームとは？

サウンドゲームとは？

- サウンドゲームとは？

  「音や音楽などを主軸にしたデザインのゲーム」
  (一般的には「音ゲー」などと呼ばれている)

- サウンドゲームの歴史

  ・電子ゲームの初期の作品

  Simon　　(1978,ラルフベア)

  ・コンシューマーでヒットの切っ掛けとなった作品

  パラッパラッパー　　(1996,SCE,PlayStation)

- サウンドゲームの種類

  ・リズムゲーム

  ・サウンドに合わせた演出に重点を置いたゲーム

  ・音楽を使ったプロシージャルゲーム

サウンドゲームの企画的特徴

Planning

- 素材が一般的
  - ・音楽という誰でも理解できるテーマと素材を扱っている。
  - ・また、音楽素材自体がコンテンツとして非常に強力である。

- 簡単
  - ・リズムゲームなど、簡単に遊べるゲームを作ることが出来る。
  - ・「音に合わせるだけで気持ちよい」ゲームデザインも確立されている。
  - ・セッションなど多人数プレイも楽しい

- 音楽の差し替えなど、データ追加だけでさらに遊べる
  - ・ゲームシステムを構築してしまえば、
    後は音楽データの差し替えで、末永く遊べるコンテンツになる。

- 継続してヒットを狙えるゲームジャンル
  - ・ビートマニア,DDR,ギターヒーローなど、
    固定ファンが付きやすいゲームジャンルである。

リズムゲーム

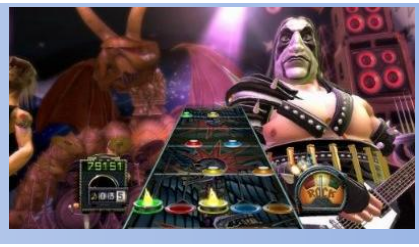
「リズムゲーム」とは、音楽に合わせて表示される操作を
ゲーム用インターフェイスデバイスで入力して遊ぶゲームです。

- パラッパラッパー(1996,SCE,PlayStation)
  画面に表示されるボタンを、
  リズムに合わせて押すゲームです。

- beatmania(KONAMI)
  画面に表示されるボタンを、
  リズムに合わせて押すゲームです。

- DanceDanceRevolution(KONAMI)
  画面に表示されるボタンを、専用ダンスマットで
  リズムに合わせて押すゲームです。

- Guitar Hero(Harmonix Music Systems)
  画面に表示されるボタンを、専用ギターコントローラで
  リズムに合わせて押すゲームです。

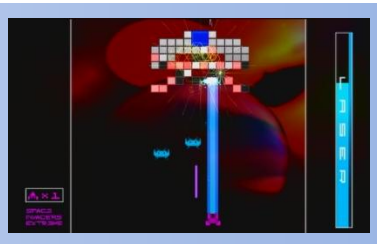


音との演出に重点を置いたゲーム

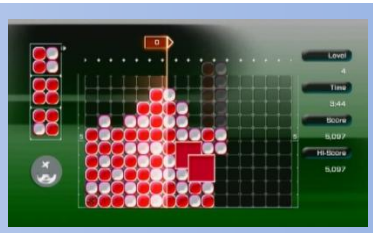
音との演出に重点を置いたゲームでは、プレイヤーの操作や、
現在のゲーム状況を、BGMに合わせて効果音を再生したり、エフェクトを実行します。

- Rez(SEGA)
  敵をロックしたり破壊したりすることで、
  BGMに合わせて効果音やループが再生されます。

- SPACE INVADER EXTREME(TAITO)
  弾を発射したり、敵を破壊することで、
  BGMに合わせて効果音が再生されます。



- LUMINES(Q ENTERTAINMENT)
  ブロックの破壊や連鎖で
  BGMに合わせて効果音やループが再生されます。
  サウンドファイルに合わせて遊ぶこともできるので、
  プロシージャル的要素もあります。

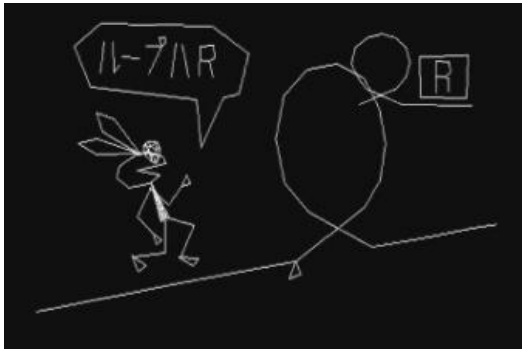


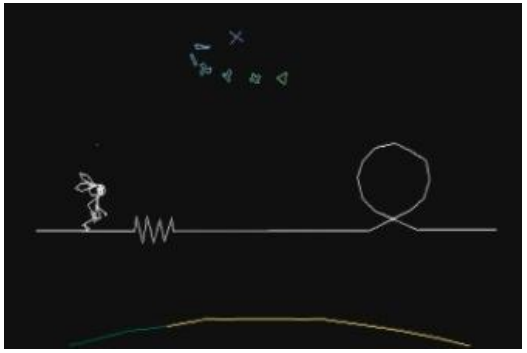
- EVERY EXTEND EXTRA EXTREME
  (Q ENTERTAINMENT)
  敵の破壊や連鎖で
  BGMに合わせて効果音やループが再生されます。
  サウンドファイルに合わせて遊ぶこともできるので、
  プロシージャル的要素もあります。

プロシージャル

CDやMP3ファイルなど、音楽データを使って
ゲームステージなどを自動生成するゲームです。

- ビブリボン(SCE)
  →CDの波形データからステージを作成

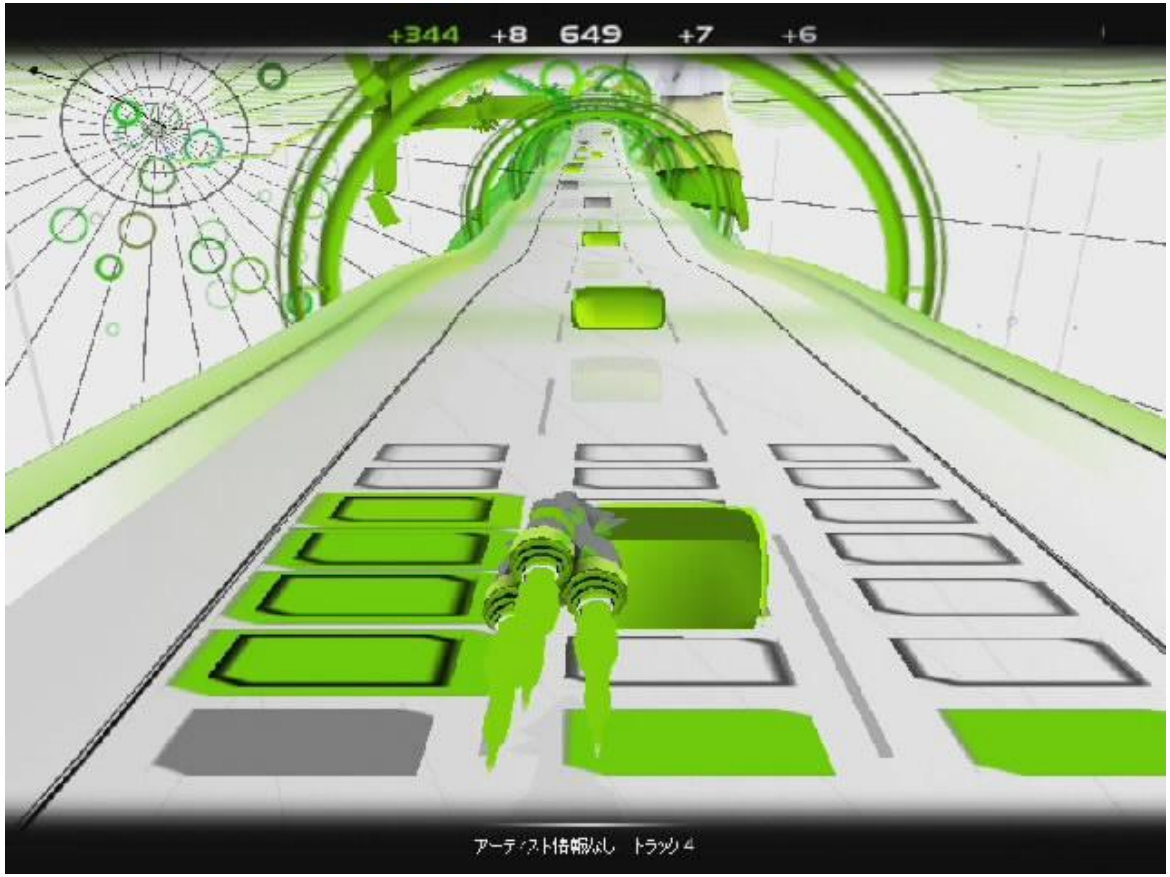




- Audio Surf(Steamで販売されているインディーズ作品)
  →MP3など音楽ファイルの波形からステージを作成

Audio Surfのゲーム説明

- Audio Surfは、MP3など音楽ファイルの波形データから
  コースを自動生成してプレイヤーが遊ぶゲームです。



ムービー説明

Audio Surfのゲームプレイまでの流れ

- Audio Surfは下記のような流れで遊びます。

1.

2.

3.

4.

5.

6.

1.メインメニュー

2.キャラクター選択
（キャラごとにゲームルールが変わるので、同じ曲でもゲーム性が変わります)

3.音楽ファイルの選択

4.音楽データ(波形)の解析

5.解析のレポート表示

6.ゲームスタート

Audio Surfの音楽解析過程

Planning

Audio Surfでは、5つのフェーズで音楽を解析しています。
(下記のような処理をしているのではないかと推測しています)

1.Scanning Song Buffer

波形データバッファのスキャン(解析)
ボリュームなどの最大値・最小値などを調べていると思われます。

2.Calculating Intensity Outliers

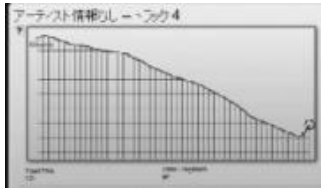
フーリエ解析(FFT)や曲の盛り上がり解析、BPM取得のための
前準備をしていると思われます。

3.Detecting Instrument Voices

曲で使われている特徴的な音の解析をしていると思われます。
(音のキックタイミングなど)

4.Smoothing Highway Curves

作成したコースをゲームでプレイできるように、
激しい変化の部分にスムージング処理を書けている
と思われます。



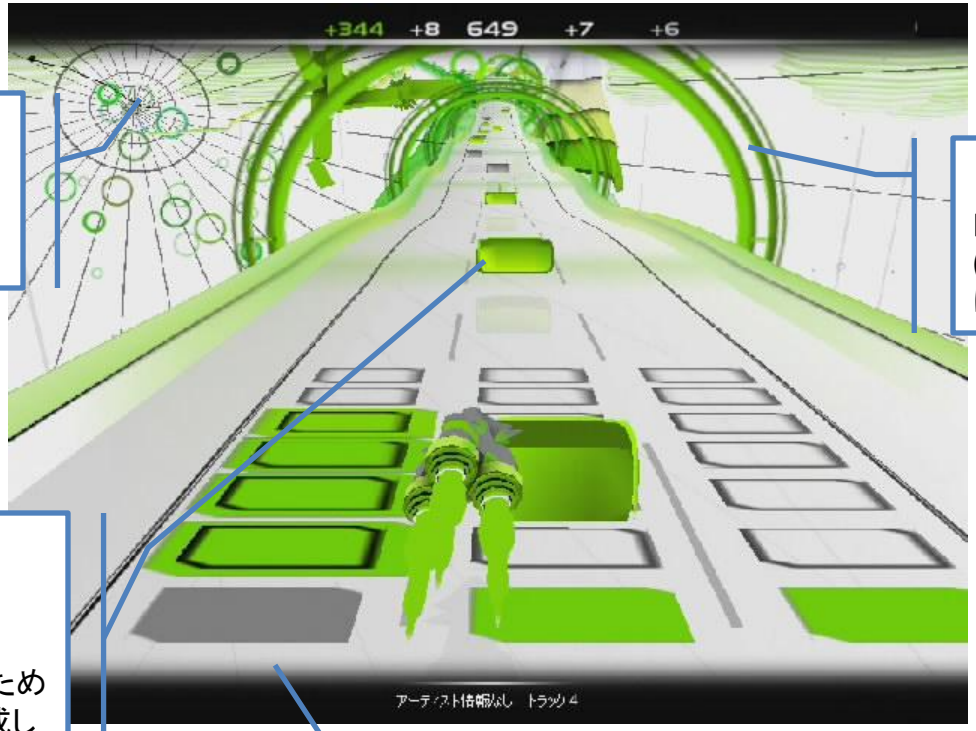
5.Finalizing Environment Tempo

BPMに合わせてコースデータを最終調整をしているのではないか
と思われます。

Audio Surfの音と演出

Planning

Audio Surfは、音に合わせて、様々な画面効果を使って演出しています。

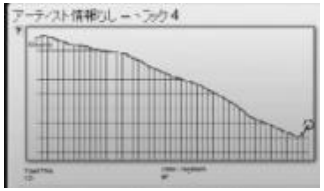


その他背景エフェクト
曲の波形データに合わせて動きます。

リング
曲で使われている特徴的な音(ハイハットやスネアドラムなど)に合わせて表示されます。

ブロック
曲の特徴的な音に合わせて、生成していると思われます。
ただし、ゲーム性を確保するために、必ずしも音データから生成しているわけではなく、難易度にあわせてブロックの位置や種類を調整していると思われます。

コース
曲の盛り上がりにあわせて、コースがこう配しています。
また、ドラムのキック音に合わせた凹凸などもあります。



3.サウンド(音)について

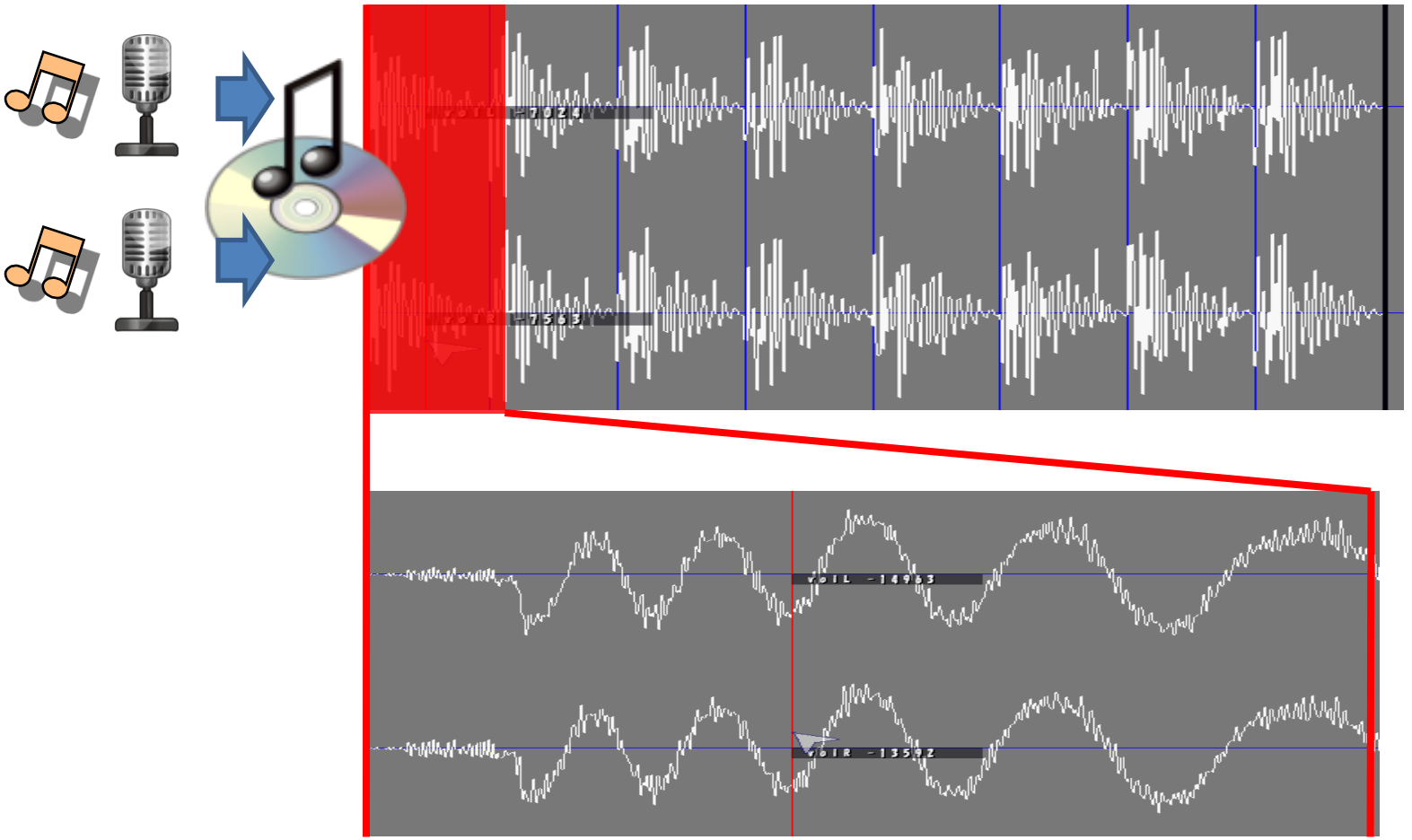
サウンド(音)の基本

- 「音」ってなんだ？
  →空気の振動

- ゲームで使われる「音」の種類
  - 人の声
  - 楽器の音
  - SEなどで使われる環境音(爆発音など)
  - ノイズ

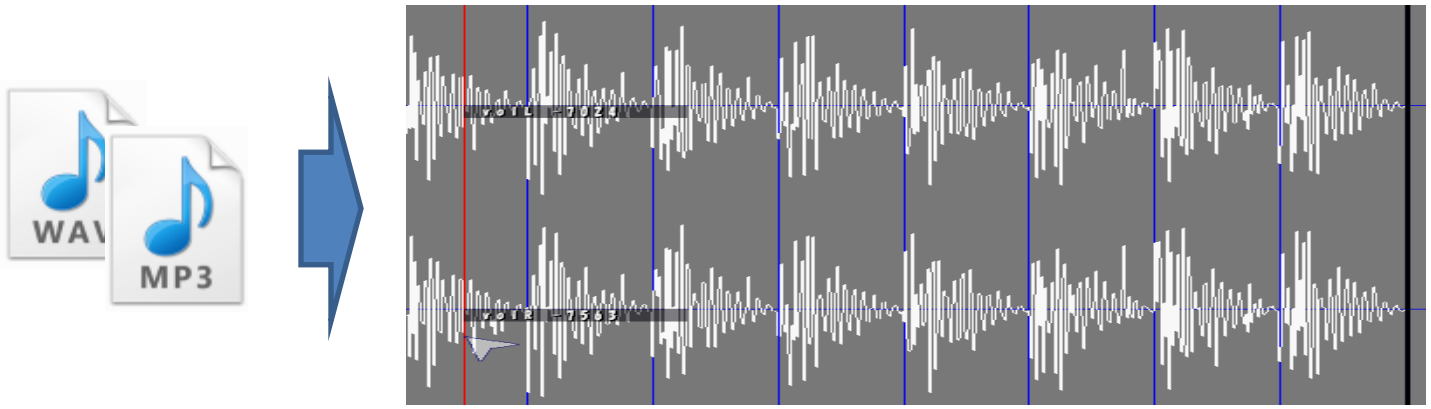
波形で見る音

- 音の波形は、「空気中で振動する音の状態」を横波で表現したものです。



音楽データに含まれる「音」のデータ

MP3やWAVEファイルなど音楽データに含まれる「音」のデータは、
下記のように表現されています。



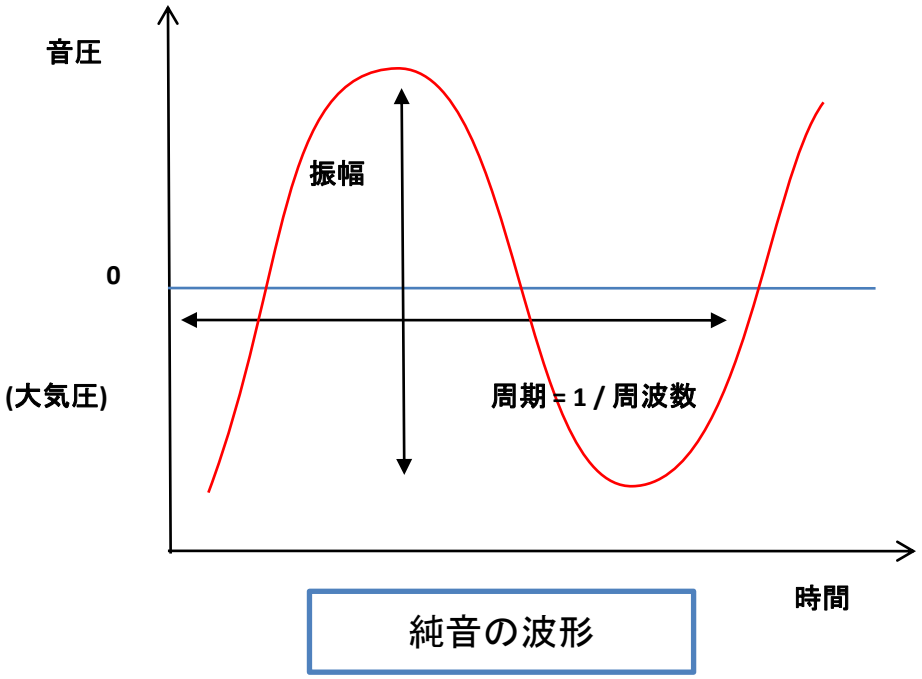
- チャンネルデータ(モノラルなら1ch,ステレオなら2ch)

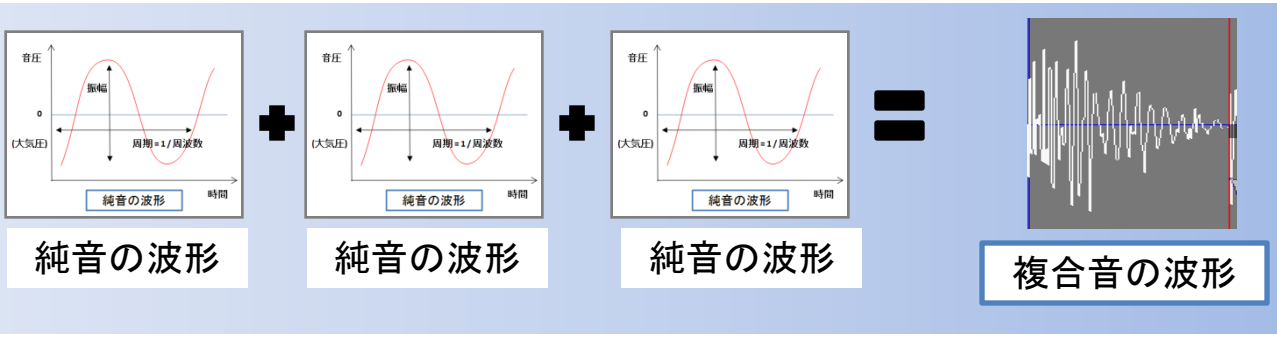
音の強さ　　ー　音圧(8Bit,16bitなど)
音圧は、数値で表現されます。
8Bitなら0-255,16ビットなら0-65535

音の成分　　ー　周波数(Hz)
周波数は、1秒間に振動する音の周期の解像度を表現しています。
44.1kHz,22.05kHz,11.025kHzなど

純音と複合音

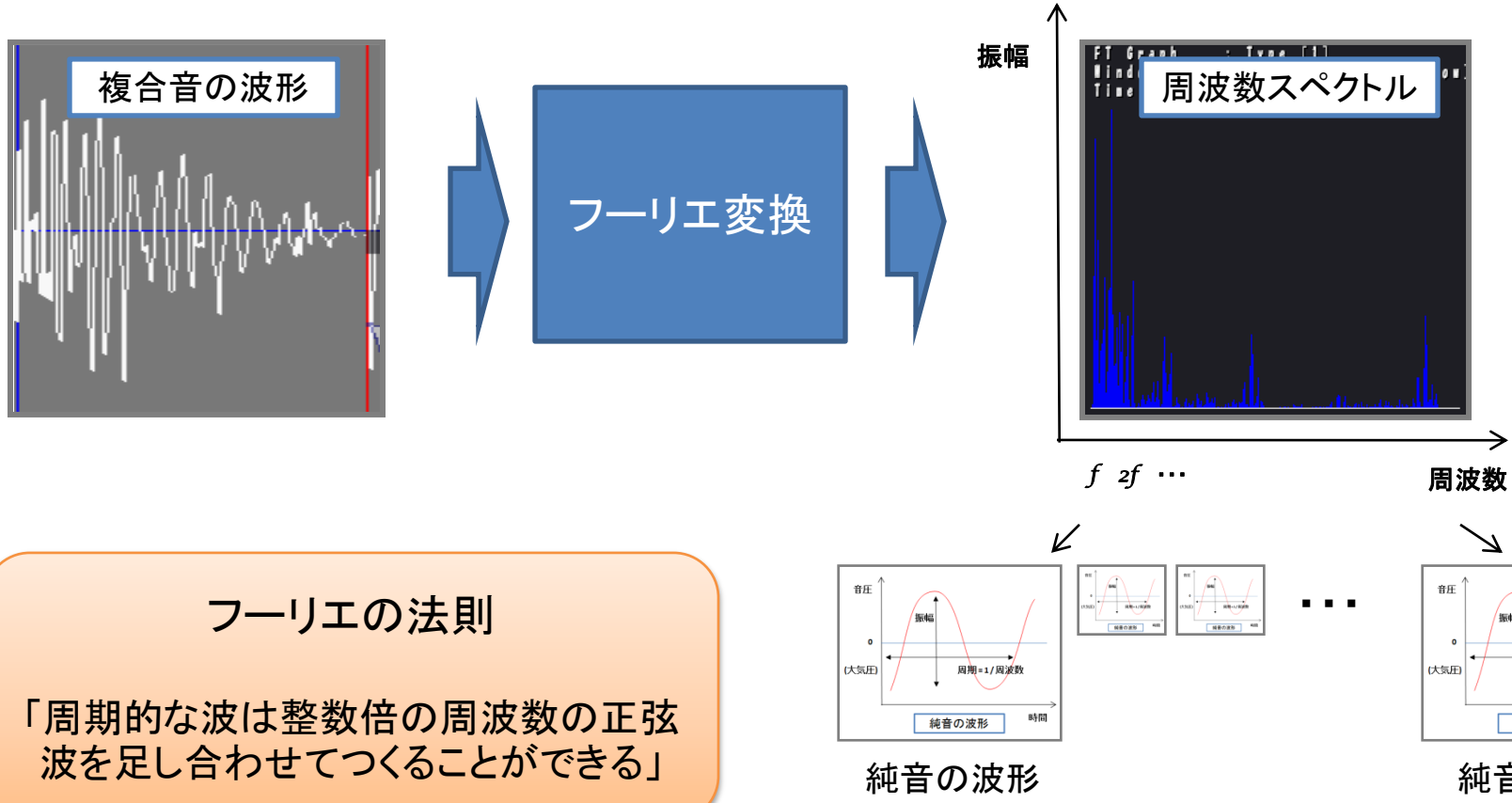
- もっとも単純な音は「純音」と言われ、正弦波(Sin)で表すことができます。
  純音(倍音になる純音)を組み合わせた音を「複合音」といいます。





周波数スペクトルで見る音

- 一定の周期を持った音は、「フーリエ変換」を使って、
  調波関係(倍音になる純音)の正弦波に分解することができます。
  これを視覚化したものを「周波数スペクトル」といいます。



フーリエの法則

「周期的な波は整数倍の周波数の正弦波を足し合わせてつくることができる」

「音」の構成

- 音を表現するための3つの要素　(正式な音楽用語ではありません)

・音色　— 人の声や楽器など、「音の個性」を表したもの。
倍音の含まれ方、倍音の時間的変化

・音高　— ドレミなど「音階」を表すもの。
純音、倍音、ピッチ、周波数、音階の時間的変化

・音強　— 音の強さ「音圧」を表すもの。
音の大きさ、音の長さ、音量の時間的変化

「音色」その1

- 「音色」を数値化することは、人間の心理的な側面に影響されるため、難しいとされます。そこで、「音色」を示す3つの音色因子を例として説明します。

・「迫力因子」

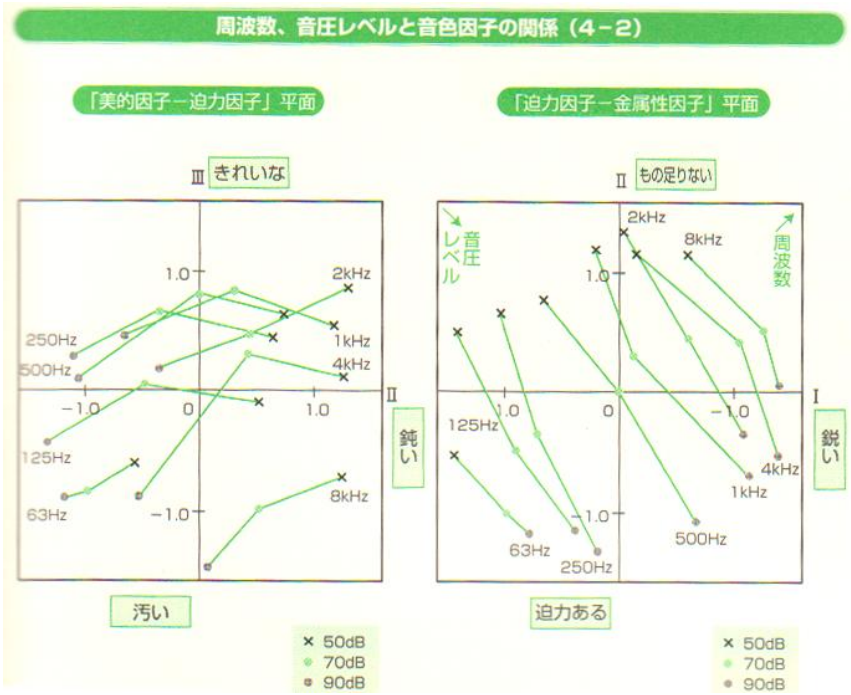
音の「迫力」を表す因子です。
主に音圧レベルとの関係しますが、
周波数にも影響されます。
音圧レベルが高いほど「迫力がある」、
低いほど「もの足りない」と感じます。

・「金属製因子」

音の「鋭さ」を表す因子です。
主に周波数と関係しますが、音圧レベルにも影響されます。
周波数が高いほど「鋭い」、低いほど「鈍い」と感じます。

・「美的因子」

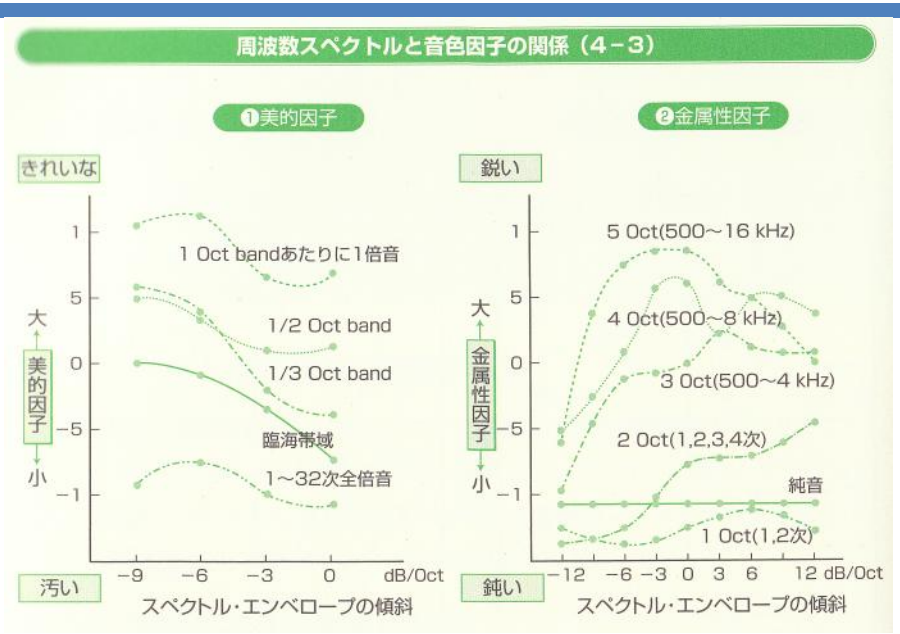
音の「美しさ」を表す因子です。
周波数の関数として、山形のカーブを描くと
「美しい」と感じます。

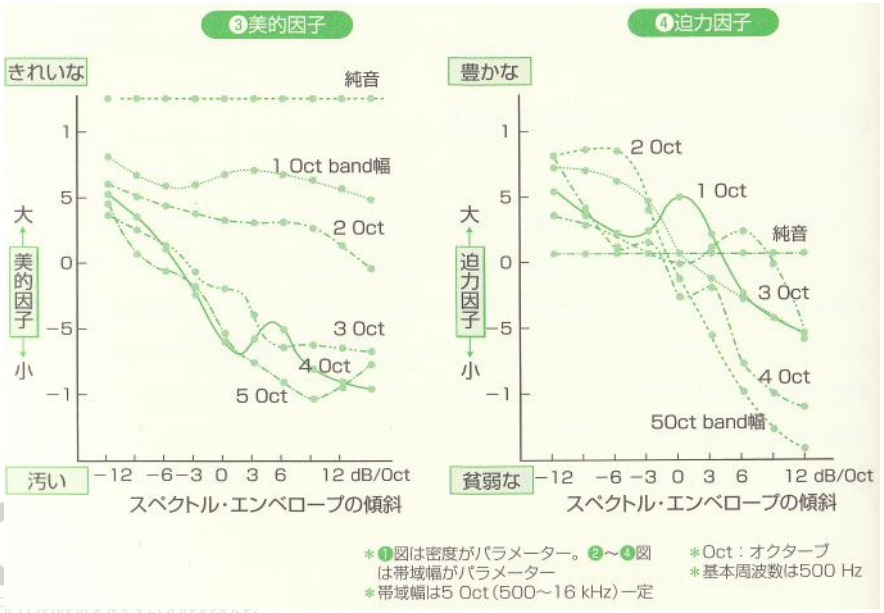


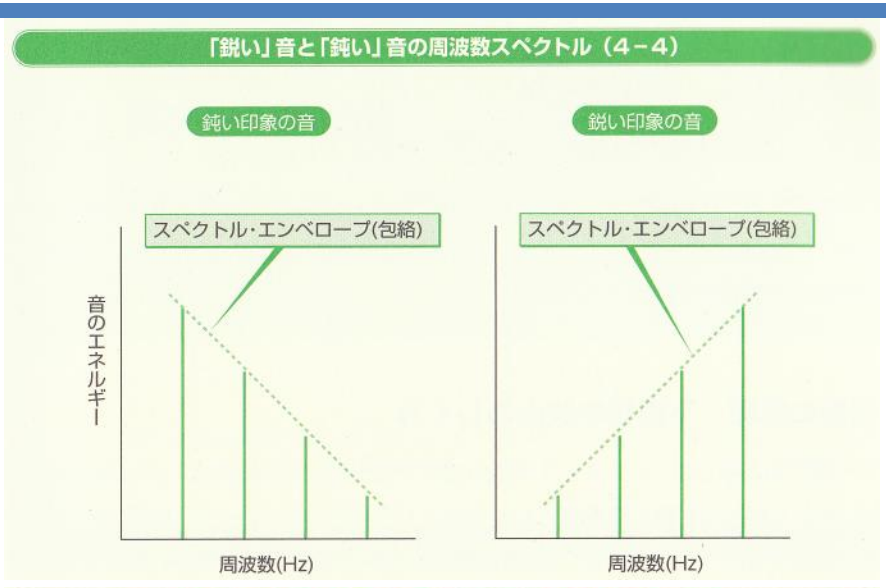
27

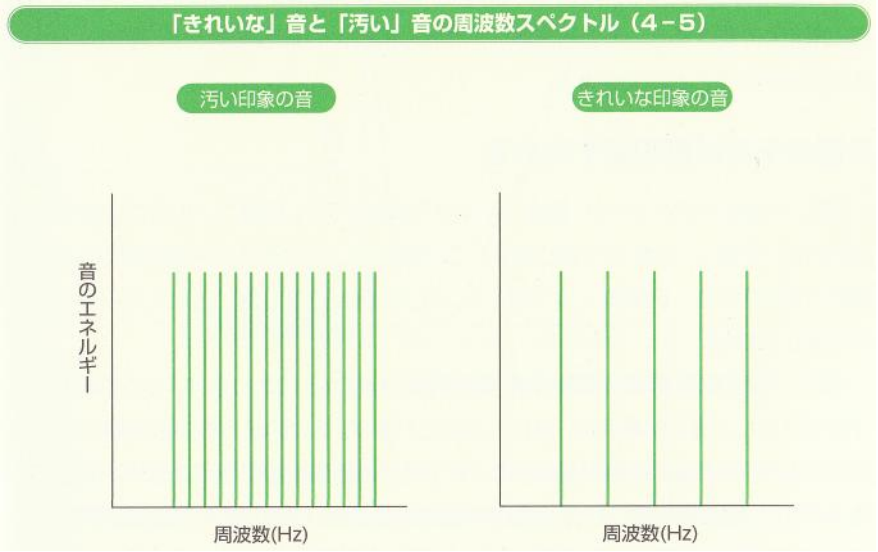
よくわかる最新音響の基本と仕組み　岩宮 眞一郎[著] 秀和システム

「音色」その2









よくわかる最新音響の基本と仕組み　岩宮 眞一郎[著] 秀和システム

「音階」

- 「音階」である音の高さは、音の周波数と対応します。

「純音」では、その音の周波数が音の高さになります。
「複合音」では、その音の「基本音」が音の高さになります。

人間が聞こえる音は、20Hzから20kHzです。
しかし楽器の音は、4kHzを超える音はあまりありません。
楽器では、ピアノがもっとも広い音域を扱うとされ、
27Hzから4.186kHzの音が出ます。
これはオーケストラのほとんどの楽器の音域を
カバーしています。

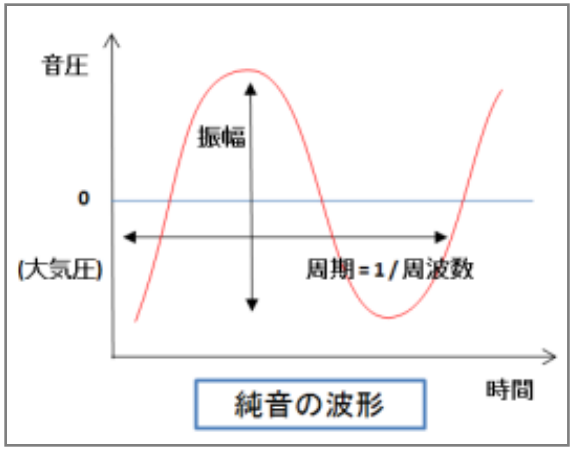


図1-6 音名と周波数の関係

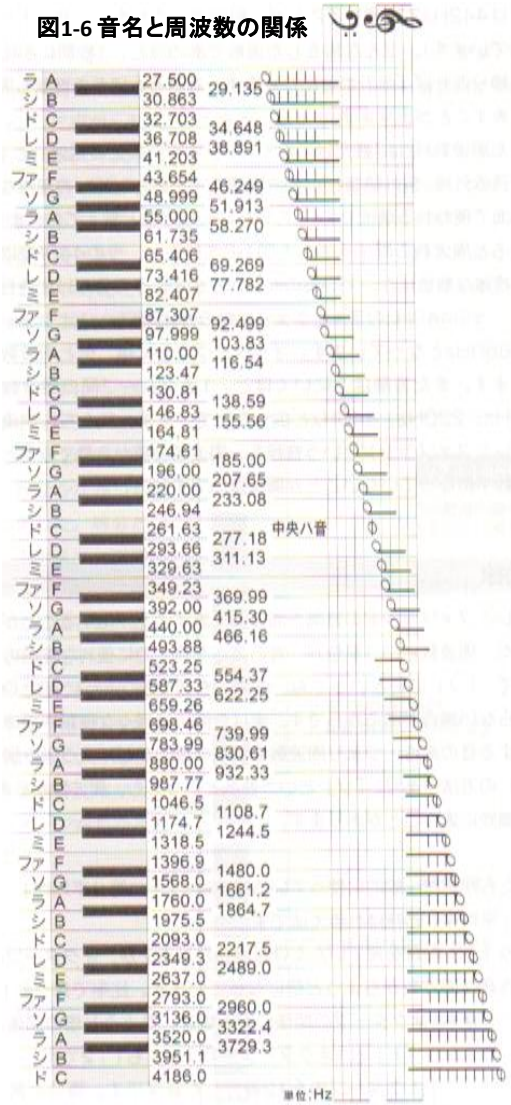


図1-7 平均律の周波数比を鍵盤の幅で表現

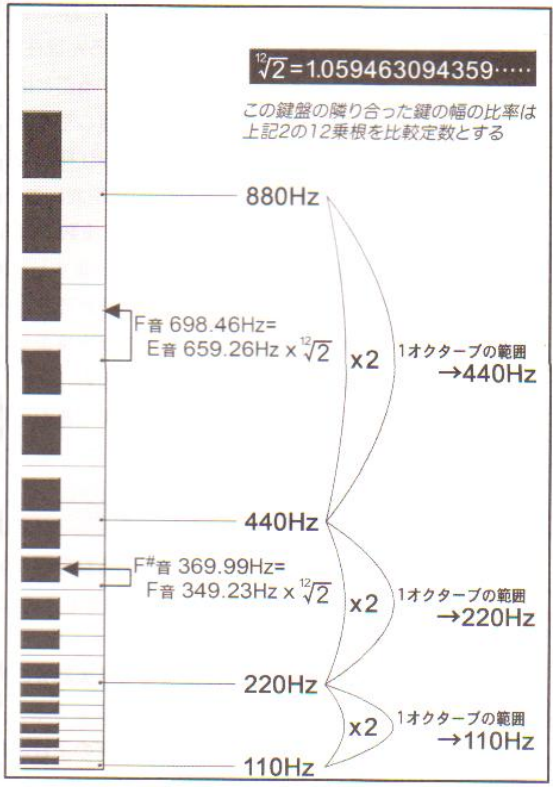
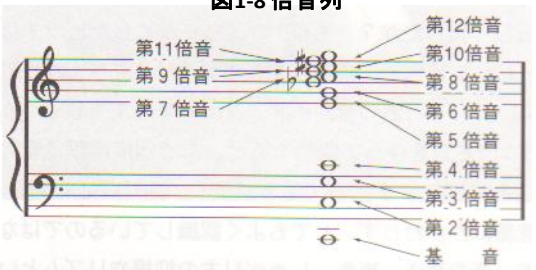


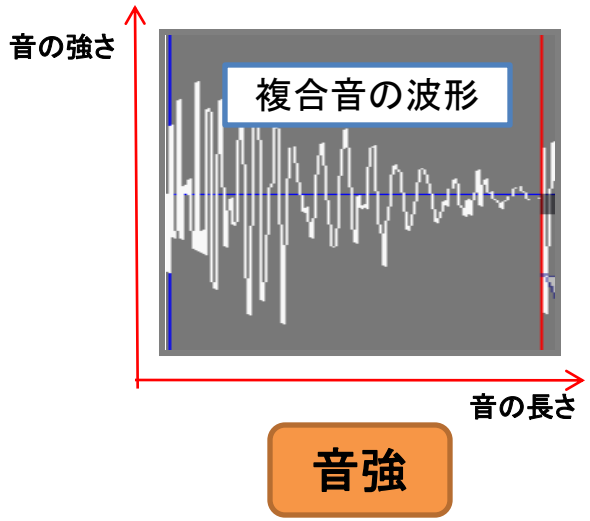
図1-8 倍音列



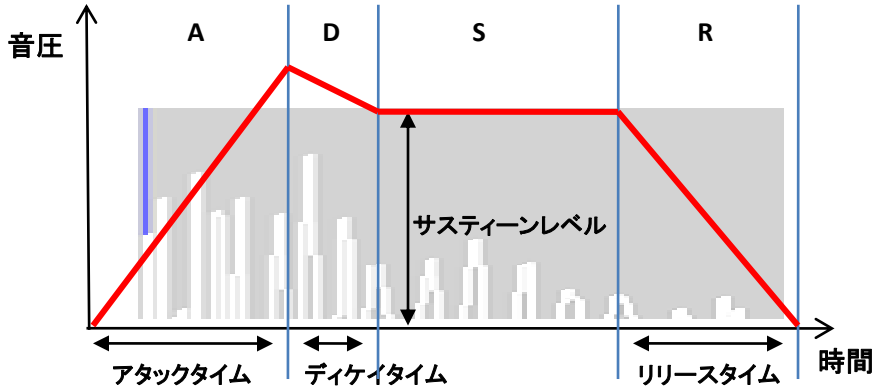
楽しい電気楽器　自作のススメ　　米本 実[著] オーム社

「音強」

- 「音強」は、「音の強さ」と「音の長さ」などを表します。
  「ADSR方式のエンベロープ」を適用することで、
  音の発生から消えるまでの音圧を表現することもできます。



音強



ADSR方式のエンベロープ

[**ADSR方式のエンベロープ**]

・アタックタイム(A)	音が立ちあがって最高の音量に達するまでの時間
・ディケイタイム(D)	最高の音量から減少して、ある一定の持続する音量になるまでの時間
・サスティーンレベル(S)	ディケイタイム後の持続するときの音量
・リリースタイム(R)	鍵盤を放してから、音が消えるまでの時間 (音が減衰して消えるまでの時間)

音の基礎知識を手に入れると・・・

Planning

- 企画　「音がズギャーン！ってときに、
  敵ババーンと登場するようにして」

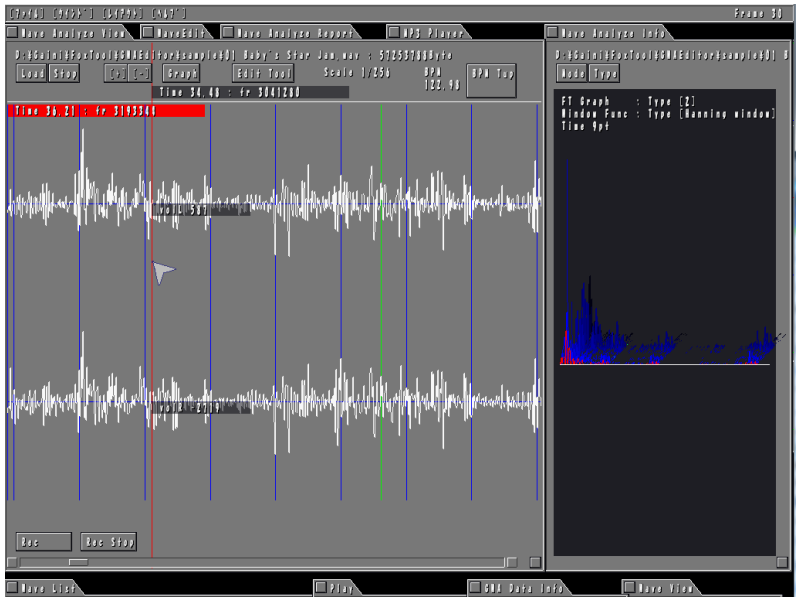
  プログラマ　「え！？」

- 企画　「鋭くボリュームが大きい音のときに、
  敵を音のアタックタイムと同時に
  出現させてほしいんだ」

  プログラマ　「鋭い音(金属因子的に)で、
  ボリュームが大きいタイミングだね。
  アタックタイムのタイミングなら、
  なんとか音の解析で検出できるかも。
  検討してみるよ。」

オリジナル波形解析ツール

- 波形解析をリアルタイムに視覚化して確認できるツールを制作しました。



- 再生中の音楽データのFFT情報などをリアルタイム表示

- 音楽データの解析結果はXMLで出力可能
  - ・周波数基本情報
  - ・BPM解析情報
  - ・特徴的な音のFFT分析データとアタックタイミング情報など

実演

4.音楽解析と映像演出

Mirage Engine

- Mirage Engineは使ったビジュアライザエンジンです。
  サウンドに合わせて映像を作ったり操作することが可能です。

ムービー説明

Mirage Engine

- 音の解析は、リアルタイムに行っている。

- 音を解析した音圧値を使って画面効果に反映している。
  ・画面の明滅
  ・ムービーの再生スピード
  ・画面の拡大縮小
  ・その他エフェクト

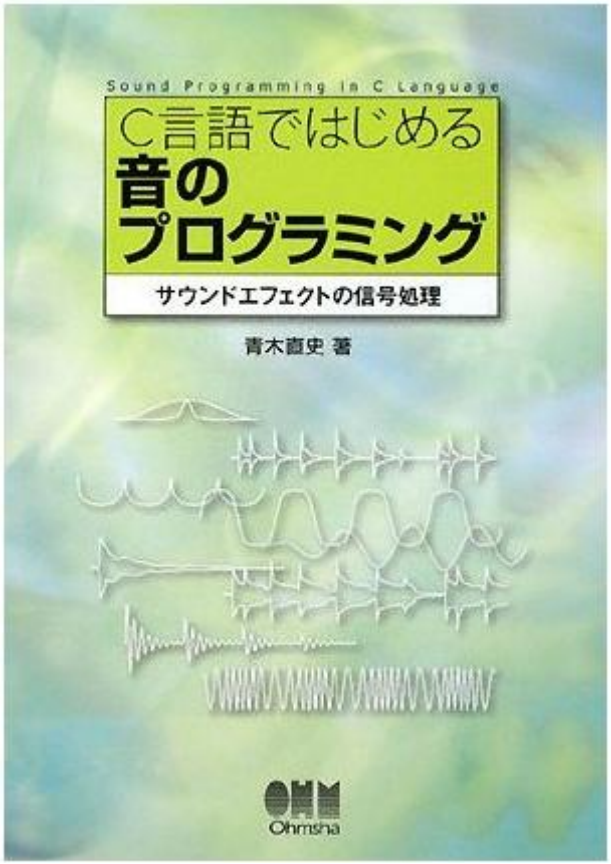
- 再生中の波形データを使ってグラフィックを作成
  ・波形リンググラフィック
  ・波形データから、パーティクル生成

- 自動再生にした場合、AIがテンポ(BPM)に合わせて
  エフェクトの操作や、スキンの変更を行っている。

5.音楽(BGM)を解析しよう

お勧めの本

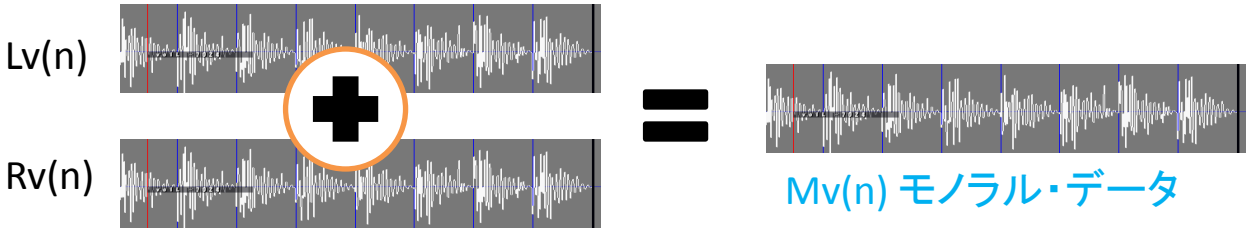
- C言語ではじめる音のプログラミング　青木 直史[著] オーム社



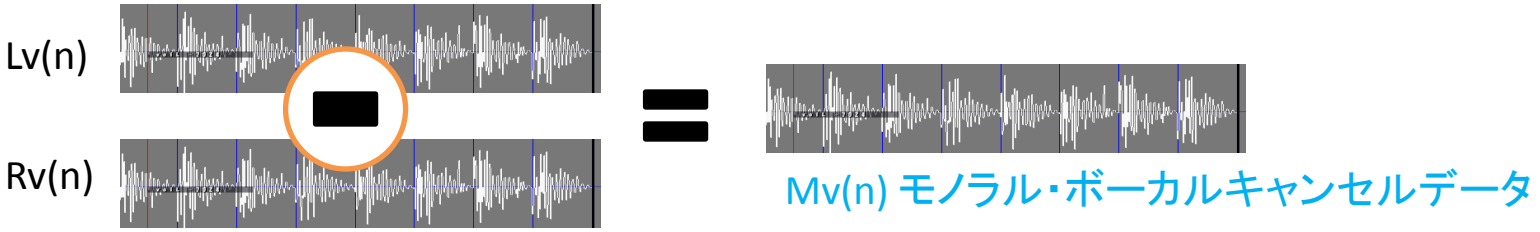
- 音の解析と加工について、丁寧に解説されています。
- 高速フーリエ変換など、基本的に使う音の解析アルゴリズムを網羅しています。
- すべての項目がサンプルプログラムで説明されています。また、ソースコードもホームページからダウンロードできます。
- 音楽解析プログラムを始めるための
最初の一冊としてお勧めです。

波形プログラムの基本

- モノラルにしてみる　(簡単！)　Mv(n) = (Lv(n) + Rv(n)) / 2



- カラオケにしてみる　(簡単！)　Mv(n) = Lv(n) – Rv(n)



- ボーカルのみにしてみる (できない！？)
  モノラルデータ - カラオケデータ = ボーカルデータ？？？

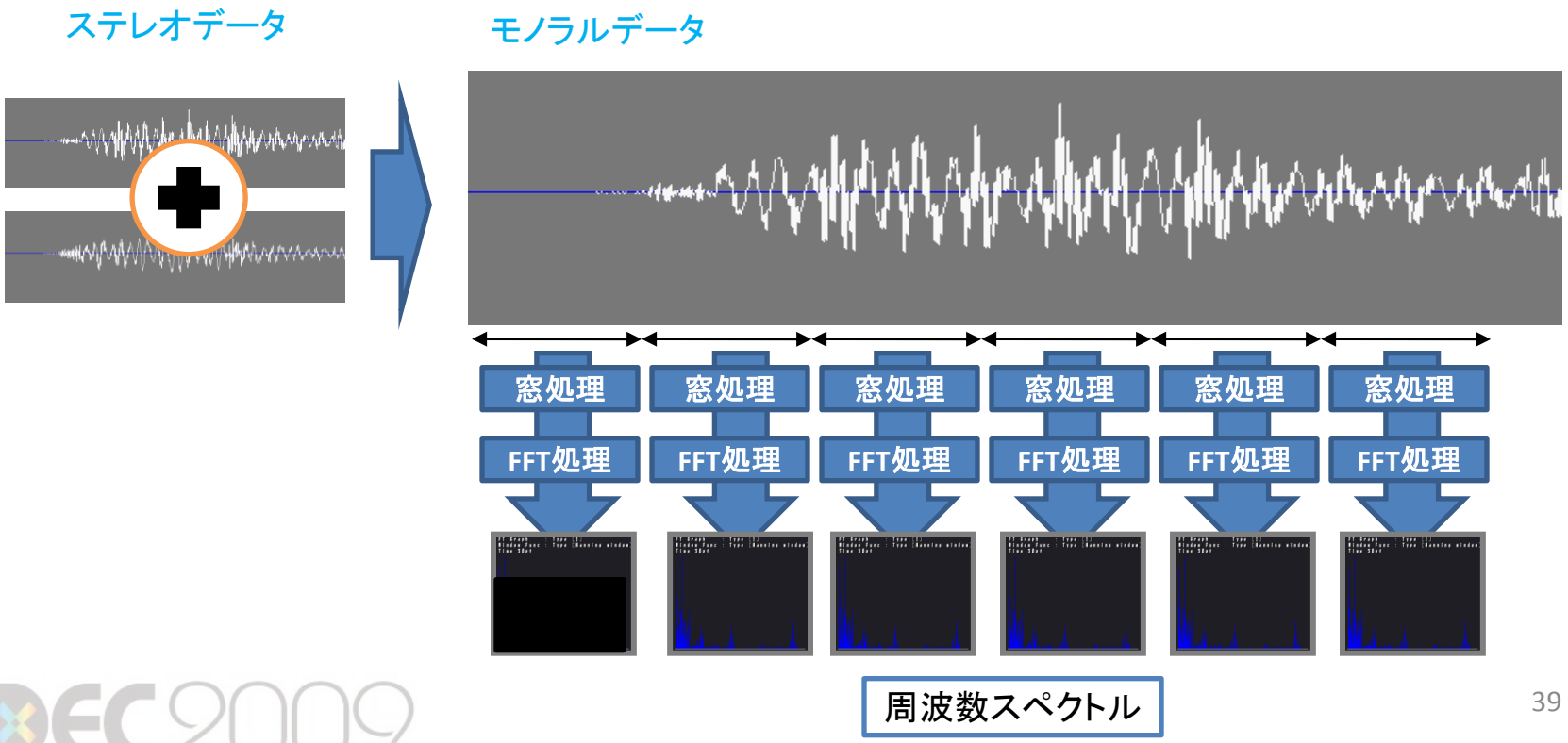
実演

・Mv (n)　モノラルボリューム値(音圧)
・Lv(n)　左チャンネルのボリューム値(音圧)
・Rv(n)　右チャンネルのボリューム値(音圧)

高速フーリエ変換(FFT)で波形データを解析する

- 高速フーリエ変換 (FFT)とは？
  フーリエ変換を高速に処理するアルゴリズムです。

  1.ゲームで使う場合は、波形データをモノラルデータへ変換。
  2.解析する波形バッファを、任意の大きさでカットして、窓処理を行う。
  3.高速フーリエ変換で周波数スペクトルに変換します。



窓処理とは？

- **窓関数**

連続する波形データから音成分を解析する場合に、つなぎ目の解析精度を安定させるために使う

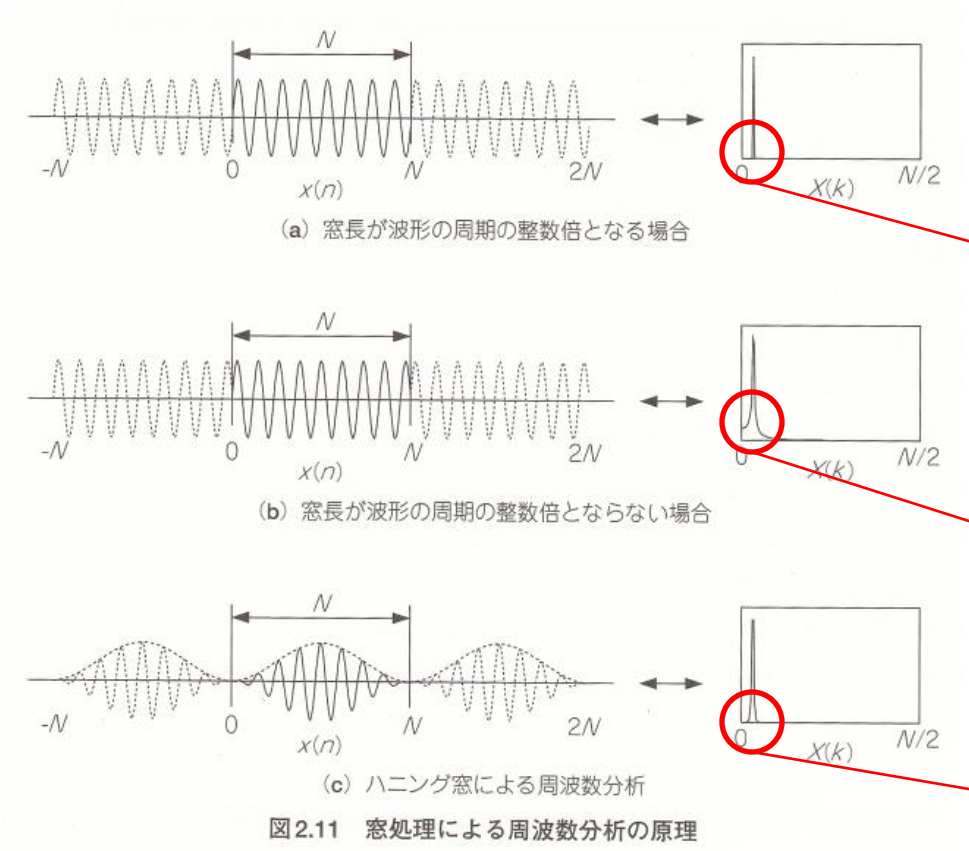


図2.11　窓処理による周波数分析の原理

(a)　窓長が波形の周期の整数倍となる場合、フーリエ変換の誤差はない

(b)　窓長が波形の周期の整数倍となる場合、フーリエ変換により誤差が生じる。

(c)　窓長が波形の周期の整数倍となる場合、フーリエ変換による誤差を軽減できる。

ディジタル・サウンド処理入門(ディジタル信号処理シリーズ)
青木 直史[著] CQ出版社

周波数スペクトルを利用する上で、誤差が許容範囲内であれば、処理速度の軽減のために窓処理をしないという方法もあります。

その他：音の解析に役立つプログラム

Planning

・FIRフィルタ,IIRフィルタ
　信号解析用のデジタルフィルタを作るためのアルゴリズム
・ノイズゲート,ノイズサプレッサ
　ノイズ除去のためのアルゴリズム
・イコライザ
　音の周波数の任意の帯域をブーストしたりカットできるアルゴリズム

……などなど

難しい！？

アルゴリズムは書籍・ネットで公開されており、
プログラムのサンプルソースも豊富にあるので、
必要なときに調べて実装するくらいのつもりで問題ナシ

音楽(BGM)を解析する

Planning

- 音楽から取得したい情報

  A.楽器のアタックタイミング

  B.使われている楽器や音階の判別

  C.音楽のテンポ(BPM)

  D.音楽の盛り上がり

  E.フレーズの解析・繰り返しなど

A.楽器のアタックタイミング

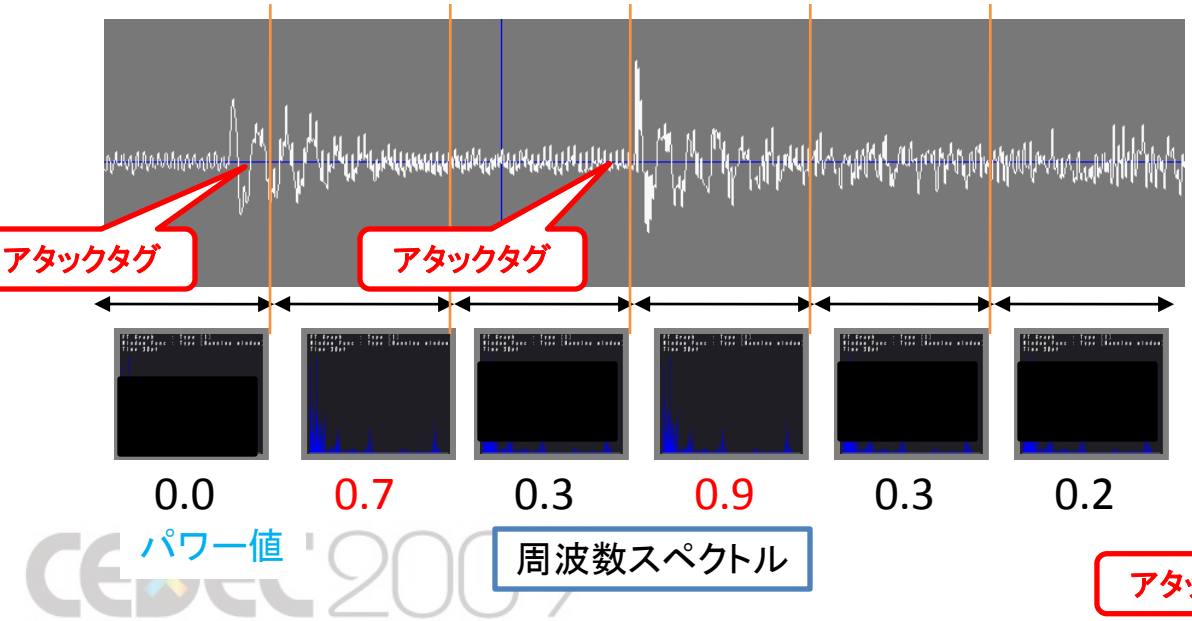
- 楽器のアタックタイミング

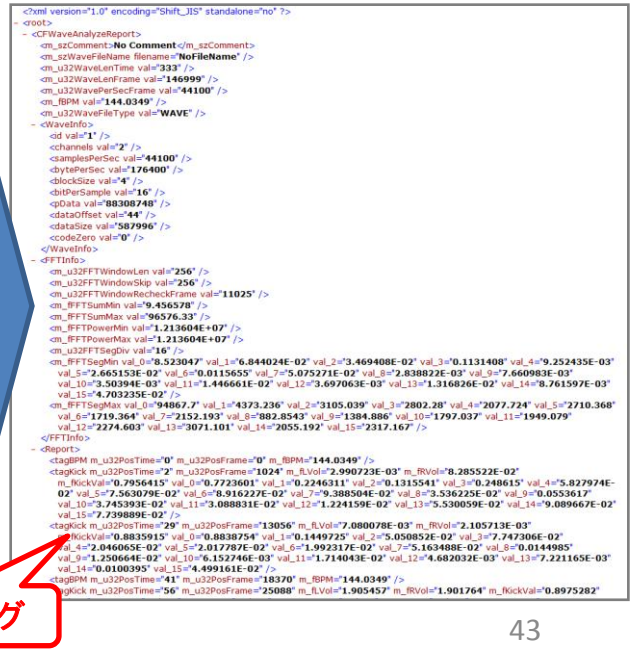
高速フーリエ変換(FFT)で解析した値の合計を「パワー値」とし、
前後のパワー値の変化が大きいか、一定値以上であれば、音のアタックタイミングと判断する。

1.音楽データすべてを速フーリエ変換(FFT)で256の帯域に分解しながら解析する。
さらにその256段階の帯域の値の合計値(パワー値)の最大値、最小値を求める。

2.パワー値を最大値で割って、数値を0.0～1.0にする。

3.音楽データすべてのパワー値を比較して、パワー値の変化が大きいか一定値以上であれば、
「アタックタグ」として登録する。



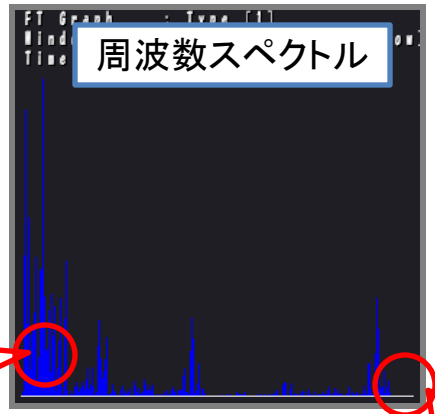


B.使われている楽器や音階の判別

- 音楽に使われている音声や楽器の判別する方法は？
  - ・厳密に、波形データからどんな楽器が使われているか詳細に判別するのは非常に難しい。
  - ・音階も、複合音の場合は、判別が非常に難しい。

- 判別可能な情報はあるのか？
  - ・ドラムや特定の電子楽器など、
    周波数帯に特徴のある楽器などは判別しやすい。

    例：　極端に周波数の低い音は、ドラム系。
    　　極端に周波数の高い音は、電子楽器の高音など。



ダンス系ミュージックであれば、
周波数の低い音はドラム系

極端に高い周波数で規則的な
音は高音の電子楽器系

  - ・MP3タグなどより、曲のジャンル情報などを頼りに
    楽器を特定していくことは可能になるかもしれない。

  - ・音声は、モノラルデータとカラオケデータの
    解析結果の差を見ればある程度判別が可能

楽器の判別や正確な音階の判別は難しいが、
音の特徴を数値化するくらいなら現時点でも可能

C.音楽のテンポ(BPM)

- 音楽のテンポはBPM(Beats Per Minute)という単位で表現
  120BPMの場合、1分間に120回のビート(四分音符)を刻んでいる。

- BPM解析の方法
  ・人間の耳で聞き手でタップして解析。

  ・MP3ファイルなどの場合、
  タグ情報にBPM値が設定されているのでその値を取得。

  ・ドラムなど特定のリズムを刻む楽器の周波数を調べて、
  その周期からBPMを自動計算する。

- BPM解析の難しさ
  ・ドラムのキックが強いダンス系ミュージックは解析しやすい

  ・ボーカルのみや、頻繁に楽器が入れ替わるクラシックなどは、
  ビート自体が刻まれないため解析しにくい。

BPM Detectの概要

- BPM Detect (The SoundTouch Library Copyright © Olli Parviainen 2001-2009)
  http://www.surina.net/soundtouch/
  「SoundTouch」にBPM Detectのソースがあります。

  BPMを検出するためのライブラリです。
  下記のような特徴があります。

  ・ドラムなど特定のリズムを刻む楽器の周波数を調べて、その周期からBPMを自動計算する。
  ・波形データを入力するだけで解析を行いBPMの値を取得できる。
  ・検出できるBPMの値は、80～190まで。(調整することで45～230までは解析可能)

  ・アルゴリズムが単純で高速。
  ・FIFOバッファを使った波形データ解析なので、リアルタイム処理が可能。
  ・調整を行えば、BPMが変化する曲でも解説可能。
  ・ソースコードが公開されている。

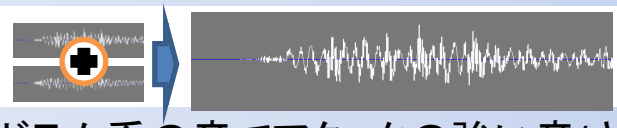
  ・LGPL(**GNU Lesser General Public License**)ライセンスである。
  商用利用でプログラム配布を行う場合は、LGPLライセンスの理解が必要。

BPM Detect解説-1

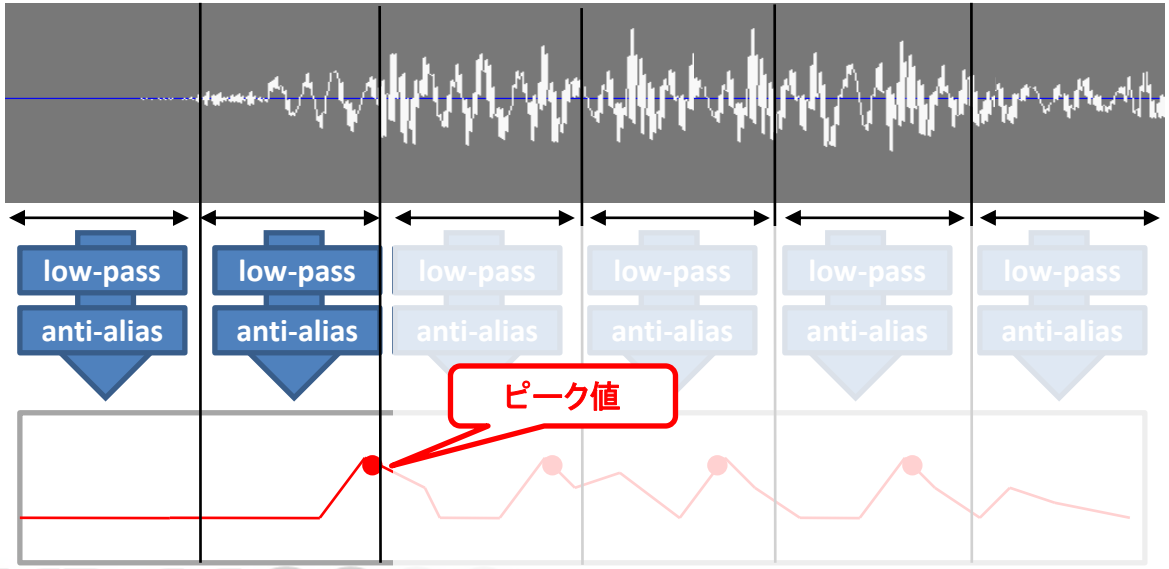
- BPM Detectの解析アルゴリズムの流れ

1. BPM解析用FIFOバッファに再生中の音楽波形データをコピーする。

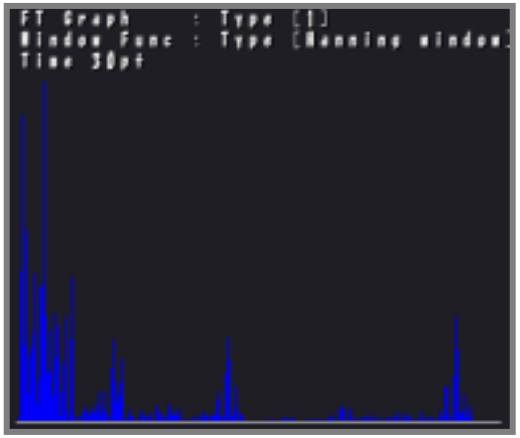
2. 波形データをモノラルデータに変換する。



3. ドラム系の音でアタックの強い音(キック音)などに注目した、解析用データを作成する。
周波数500Hzの音を、low-pass filterやanti-alias filterを使って抽出し、
エンベロープを作成する。また、そのピーク値を保存。



500Hzの音のエンベロープから作成した解析用データ

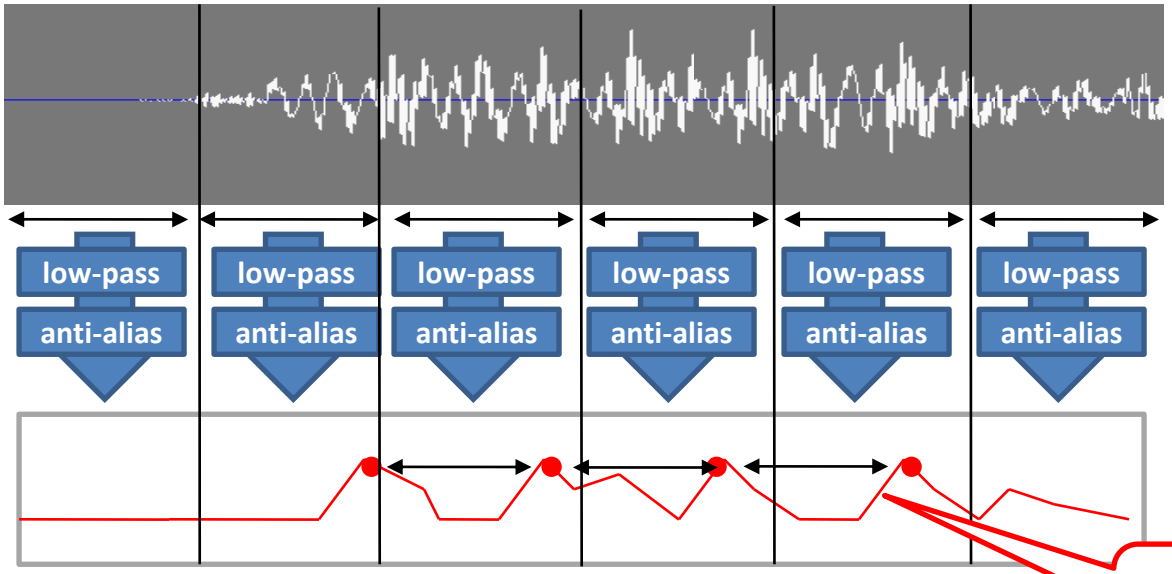


500Hzの
音を抽出

BPM Detect解説-2

- 前のページの続き

4. 3.で計算したピーク値が、FIFOバッファに存在するか検索。
   近い値のピーク値を見つけたらリズム音と考えて、
   ピーク値からピーク値の幅をBPM値として計算する。



500Hzの音のエンベロープから作成した解析用データ

ピーク値とピーク値の幅から
BPMを計算する。

5. 1に戻ってBPM解析を継続する。

その他のB PM解析アルゴリズム資料

Planning

- その他のB PM解析アルゴリズム資料

  - BPM Detect　(LGPLライセンス)

    The SoundTouch Library Copyright © Olli Parviainen 2001-2009

    http://www.surina.net/soundtouch/

    (「SoundTouch」にBPM Detectのソースがあります)

  - **Beat Detection Algorithms** (By Frédéric Patin aka YOV408)

    http://www.gamedev.net/reference/articles/article1952.asp

    (Thursday, May 15, 2008)

    http://www.gamedev.net/reference/programming/features/beatdetection/

    (Monday, October 01, 2007)

  - Beat This ：**A Beat Synchronization Project**

    http://www.owlnet.rice.edu/~elec301/Projects01/beat_sync/beatalgo.html

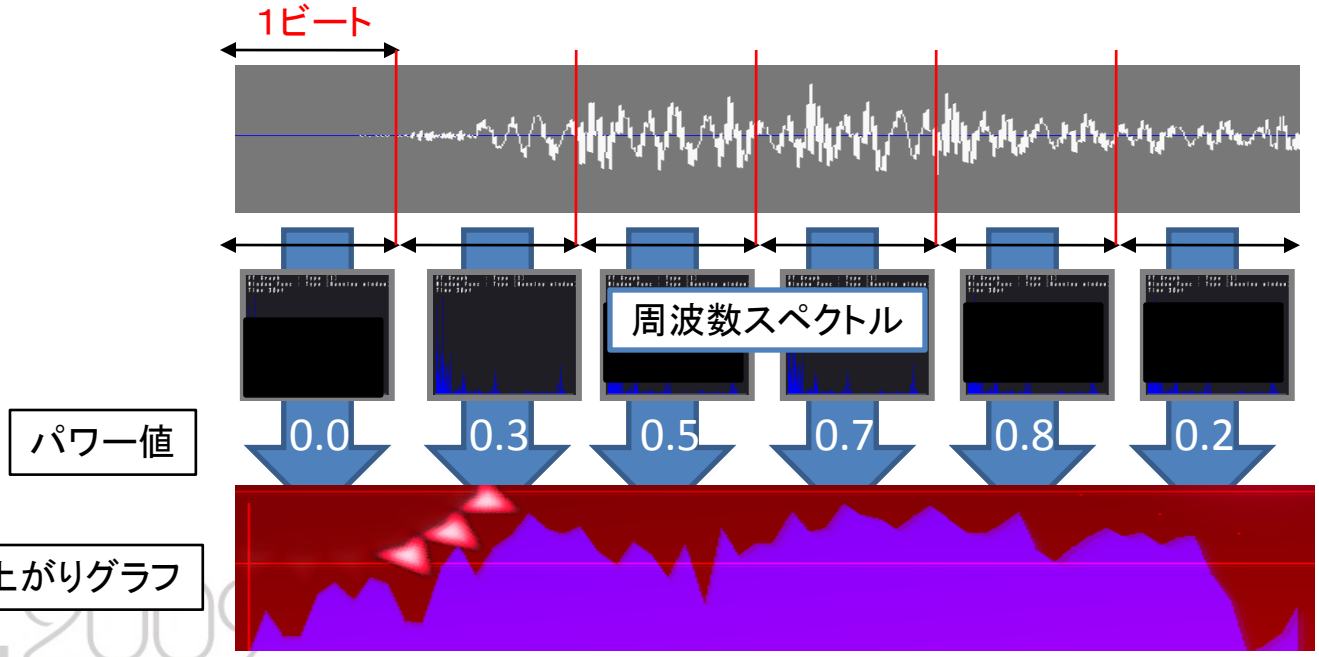
-

D.音楽の盛り上がり

- **音楽の盛り上がりとは？**
  一般的な音楽の場合、音圧が高く純音が多くなっている部分が盛り上がっている。

- **盛り上がり解析の手順**
  1.曲のBPM(テンポ)解析を行う。
  2.曲データの全般にわたってFFT解析をしてデータ化を行う。
  3.FFT解析の値を合計して、パワー値として曲のタイムラインにあわせて保存。
  4.1で解析したBPMの値から、1ビート分の時間を計算する。
  5.ビート分のパワー値の合計をさらに曲のタイムラインにあわせて記録。
  　これが曲の抑揚を示すグラフデータとなります。

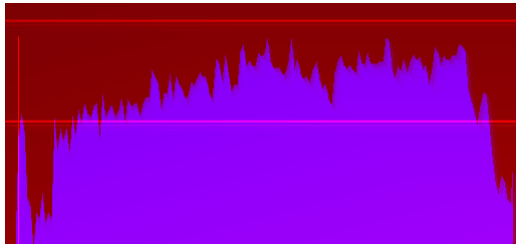
実演

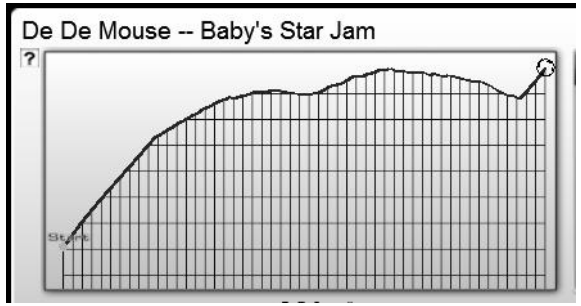


D.音楽の盛り上がり(サンプル)

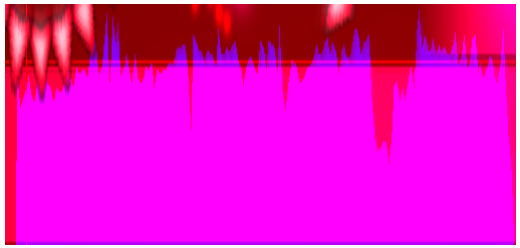
- Audio Surfのコーストの比較
  ゲームデザインの違いから、作成したグラフの結果も異なるようです。

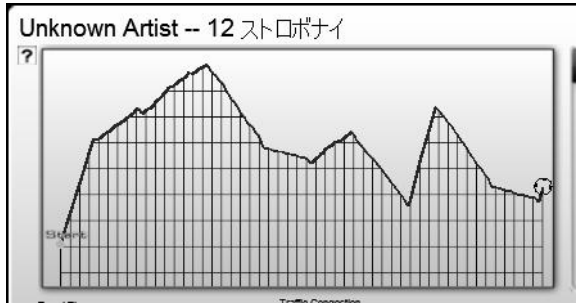
Baby's Star Jam
De De Mouse



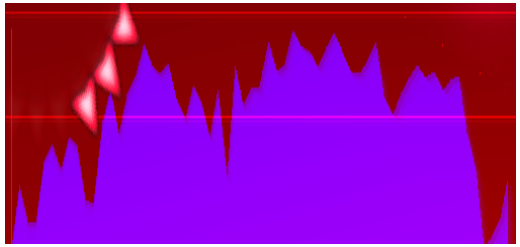


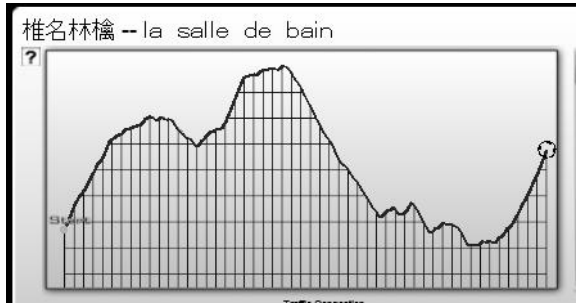
ストロボナイツ
初音ミク
作曲：kz 作詞：yae





la salle de bain
椎名林檎

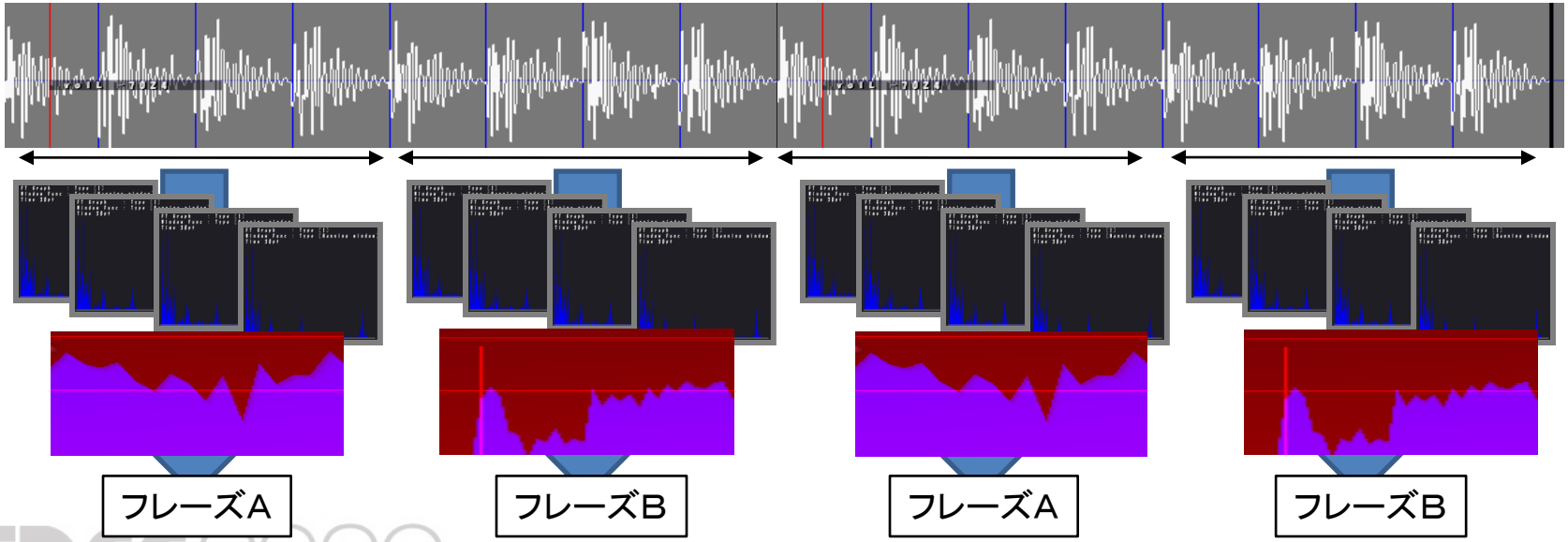




E.フレーズの解析・繰り返しなど

- フレーズの解析・繰り返しは、ある程度の精度であれば、
  これまでの音楽解析の方法を使って、フレーズの辞書化をすることで可能です。

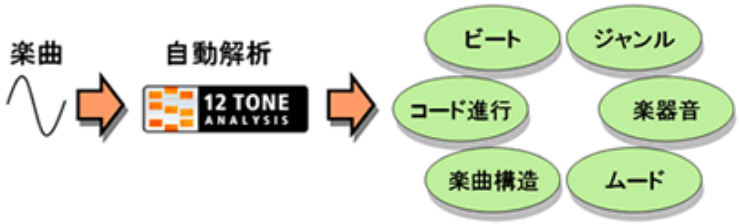
1.波形データをモノラル・ボーカルキャンセルデータにする。
2.BPMを解析する。
3.BPMから、1ビートの長さを計算する。
4.1ビートx4回(1小節分)のFFT解析データ、アタックデータ、パワーグラフなどを
　「フレーズデータ」として「辞書化」する。
5.辞書化したフレーズデータを元に、繰り返しフレーズを検索していく。



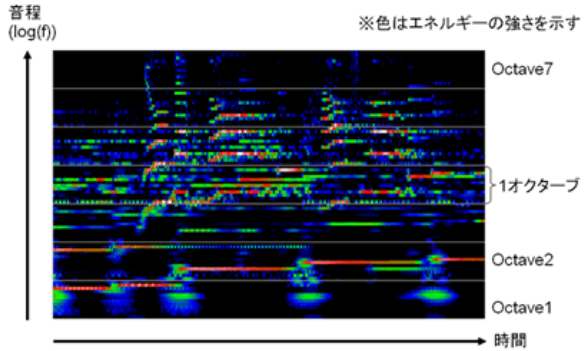
その他の音楽解析

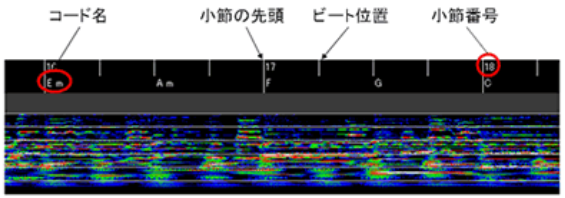
- SONY 12音解析 音楽解析技術 12 TONE ANALYSIS

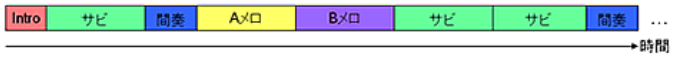
実装例：
ウォークマン(SonicStage),Rolly,VAIO付属ソフト



楽曲の波形を解析し、様々なメタデータを自動抽出する技術







12音解析のしくみ

http://www.sony.co.jp/SonyInfo/technology/technology/theme/12tonealalysis_01.html より

1・時間−音程解析

音程は1オクターブあたり12の音程（ドレミに相当）に解析する。

2・音楽理論に即した解析

解析された二次元画像を元にさまざまな信号処理、検出処理などを行い、
テンポ、拍子、小節線などのビート構造、コード進行、キー、
楽曲構造など音楽理論に即したものの検出処理を行う。

3・特徴量の抽出

時間−音程解析結果、音楽理論に即した解析結果を元に、
楽曲の分類に有効と思われる特徴量の抽出を行う。

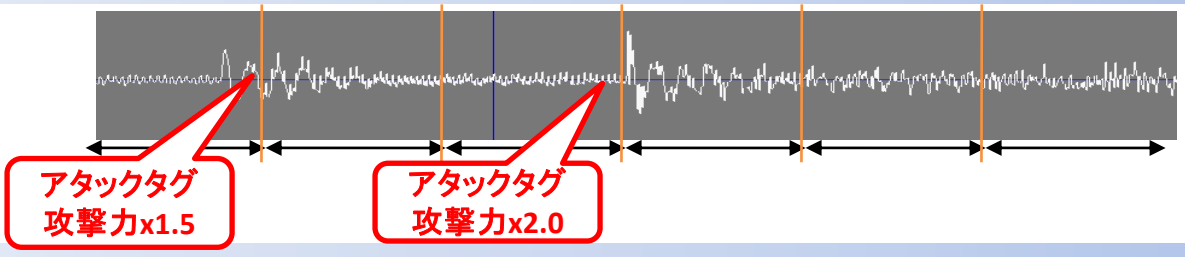
4・メタデータの推定

これまでの音楽解析で求められた特徴量を元にメタデータを推定する。
ここで推定したメタデータは、例えば音楽検索のためのキーとして利用することができる。

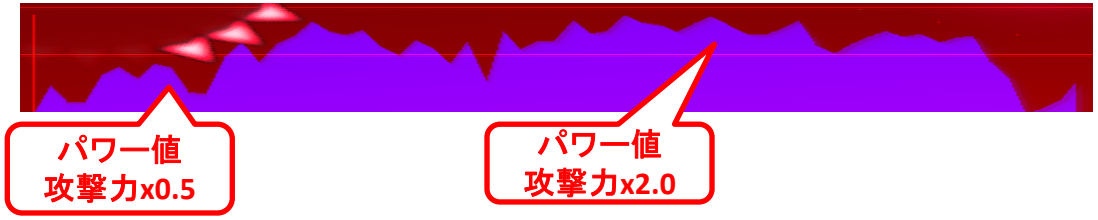
メタデータ例 ： スピード感,エネルギー感,楽器音,ムード

音楽解析を使ってできる企画

- 楽器のアタックタイムに合わせた攻撃で、攻撃力がアップするバトルゲーム

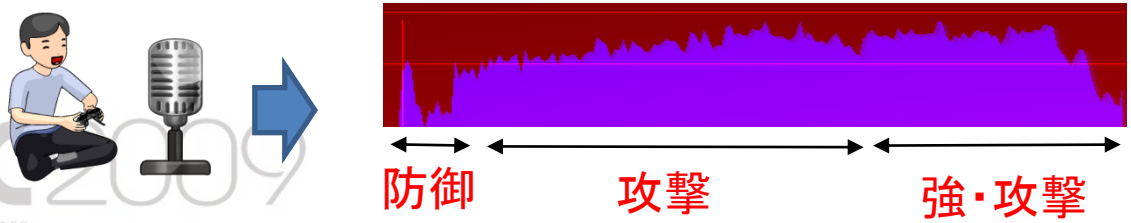


- 音楽の盛り上がりに合わせて、攻撃力が変化するバトルゲーム



- プレイヤーがフレーズを選択して、そのフレーズのパワー値で、戦うバトルゲーム。

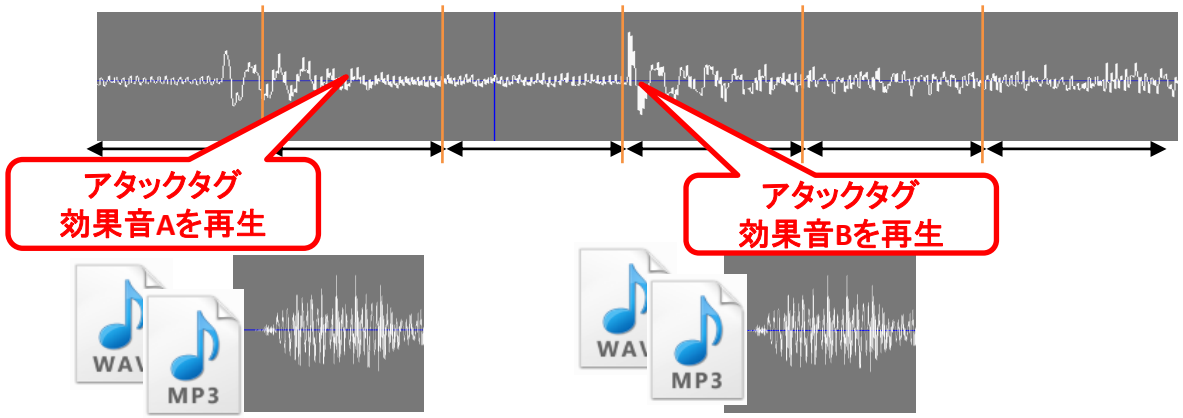
- マイクで歌って音が、そのまま攻撃になるバトルゲーム。



自動化できる演出

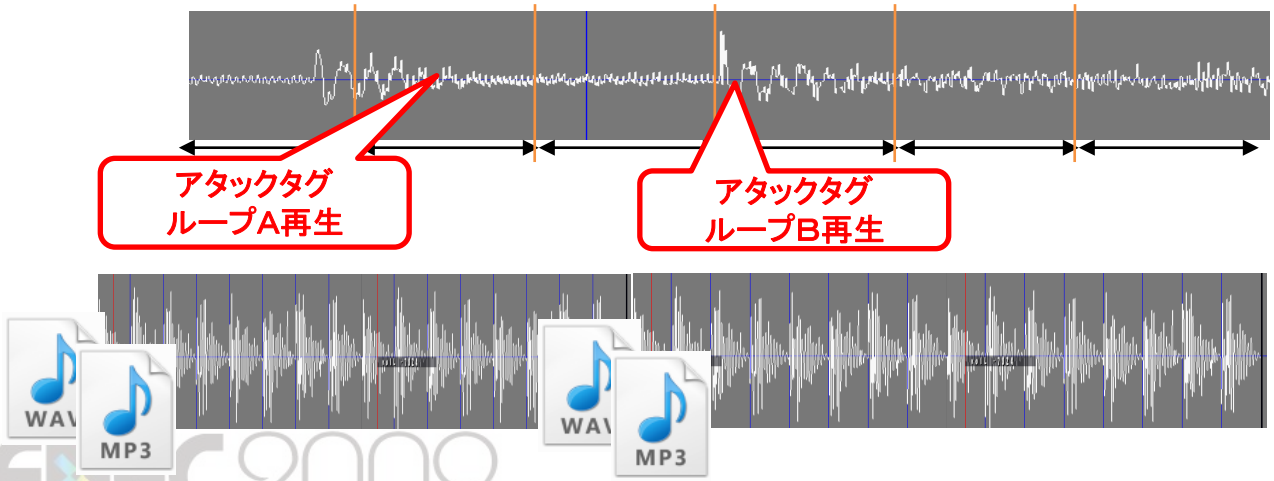
Planning

- 効果音をアタックタイミングやBPMタップタイミングで再生する。



Rezのような演出が可能

- ループサウンドをアタックタイミングやBPMタップタイミングで再生する。



DJのような演出が可能

自動化できるかもしれない処理

Planning

- アクションやシューティングゲームで、
  音楽に合わせて敵を出現させるときなどのイベントに、
  自動化処理として音楽解析を使う。

- リズム系ゲームの音楽にあわせたゲームスクリプトの自動作成

- RPGなどで、イベントシーンで使われるエフェクトと音楽の同期処理

- 格闘ゲームの入場などで、プレイヤーが選んだ音楽ファイルから、
  カメラワーク、ライティング、モーションを自動的に選択して
  音楽のフレーズに合わせる。

  また、キーフレームやモーションの長さの自動調整を行う。

6.音楽解析を使った プロシージャル技術の ゲームへの応用

シューティングゲームに応用してみる

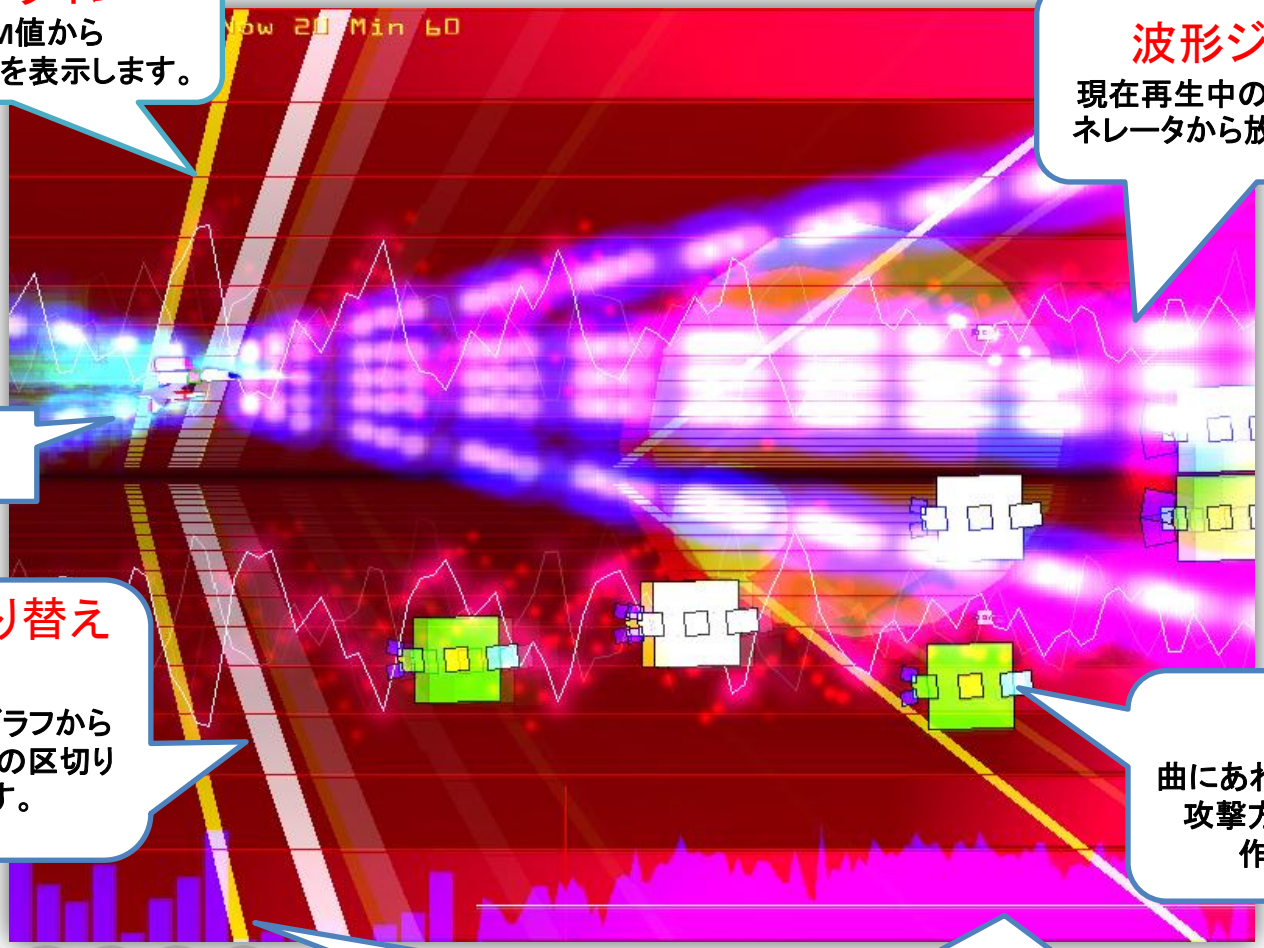
Planning

- 音楽を解析してステージを自動作成するシューティングゲームを作ってみました。
  (A.実演)
  (B.実演)

実演

解説

ゲーム中のグラフィックは、サウンドの解析データから作成しています。



BPMタップライン
計測したBPM値から
1タップごとにラインを表示します。

波形ジェネレータ
現在再生中のサウンド波形をジェネレータから放出して表示します。

自機

フレーズ切り替えライン
曲の盛り上がりグラフから解析したフレーズの区切りを表示します。

敵
曲にあわせて敵の出現位置、攻撃方法などを自動的に作成しています。

アタック音のFFTグラフ

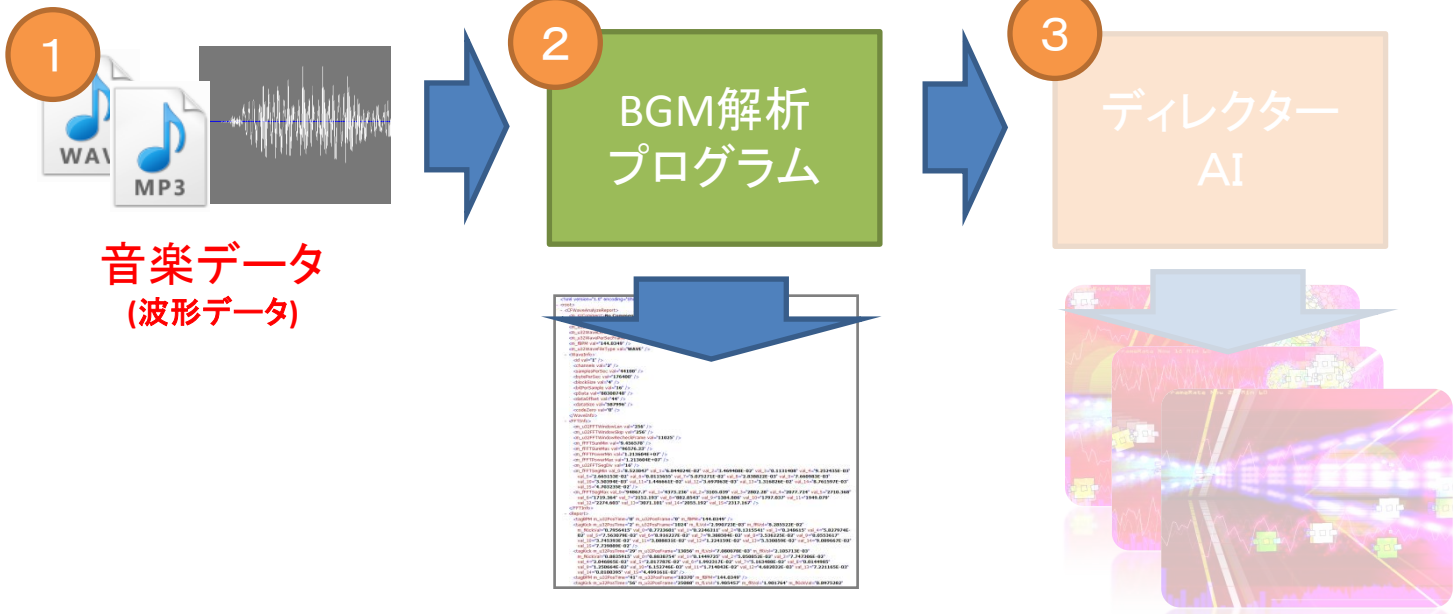
曲の盛り上がりグラフ

ステージ作成のアルゴリズム-1

- ステージ作成のアルゴリズム手順

1.音楽データのロード

2.音楽解析をゲームプレイ前に実行。解析データをC++のクラスで生成(XMLデータとしても保存)



ステージ作成のアルゴリズム-1

Planning

3.ステージを作成する「ディレクターAI」が、
解析データから、敵や敵の出現方法を生成してステージデータとして作成。
ゲームバランス調整の補正を入れる。

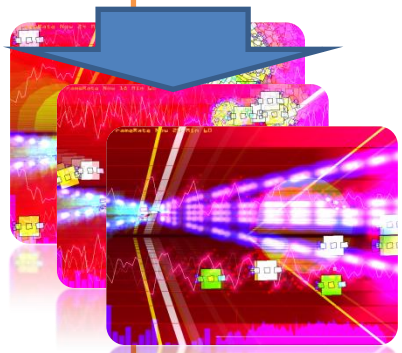
4.ゲームプレイ実行

解析レポート
ラス,XML)

結果
月データ

3

ディレクター
AI



ステージデータ
・敵の発生タイミング
・敵のフォーメーション
・敵の種類

ステージ作成の流れ

1.「音楽の盛り上がりグラフ」から、
おおまかなフレーズと思われる変化を解析。

2.フレーズごとに盛り上がりのランクを付ける。

3.フレーズのランクから、
敵の攻撃フォーメーション(戦術)を選択。
フレーズ内は、選択したフォーメーションで
攻撃する。

4.音楽解析データから、音のアタックタイミング
を取得。これにあわせて、敵の発生を行う。
また、波形データから発生位置を計算する。

5.アタック音のFFTデータから、
敵の移動・攻撃のアルゴリズムを選択する。

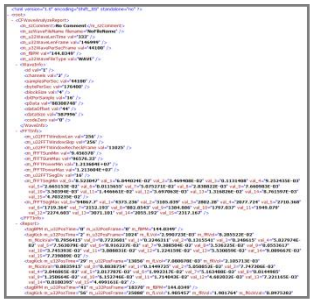
4

ゲームプレイ実行

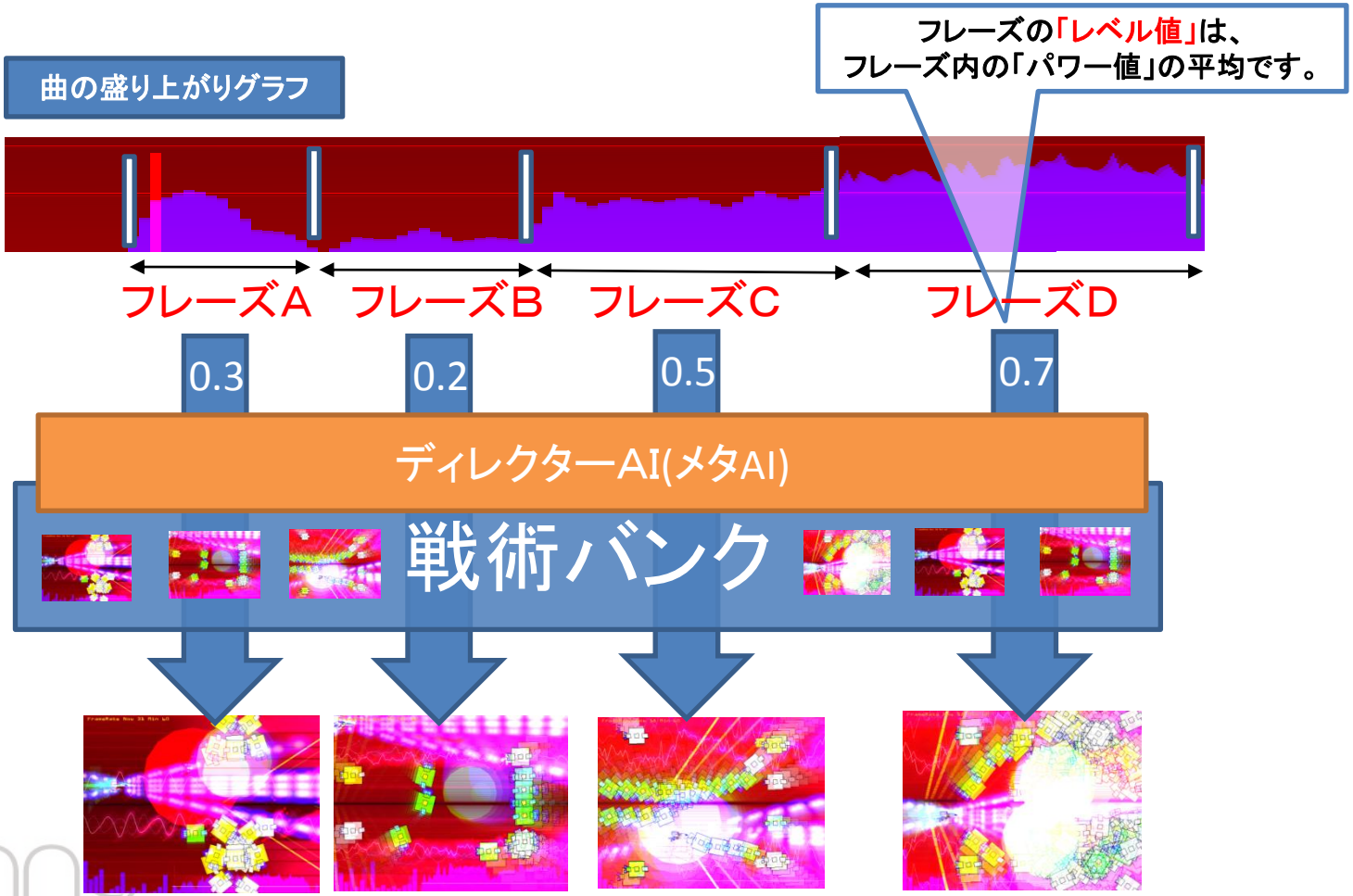
CEDEC 2007
CESA Developers Conference

「音楽の盛り上がり」から戦術を作成-1

- 「音楽の盛り上がりグラフ」から、おおまかな「フレーズ」の区切りを解析します。
  解析したフレーズに「レベル値」を設定します。「ディレクターAI(メタAI)」が、
  このレベル値から「戦術バンク」の「フォーメーション(戦術)」を選択します。



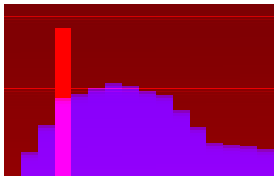
音楽データ解析レポート
(C++のクラス,XML)



「音楽の盛り上がり」から戦術を作成-2

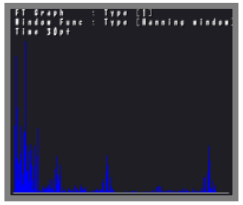
Planning

- 「戦術バンク」は階層構造になっており、音楽の盛り上がりに合わせて
  ディレクターAIが「フォーメーション(戦術)」を選択できるようにしています。



フレーズA

・レベル値　0.3
・パターン値　2

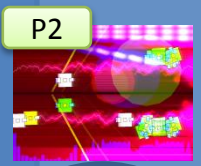


フレーズの「パターン値」は、
フレーズの最初の音のFFT値を使って、
パターン値に変換しています。

ディレクターAI
(メタAI)

戦術バンク

レベルA
(0.00-0.25)

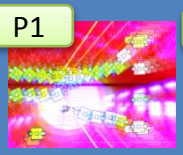


P3 P4 P5

レベルB
(0.25-0.50)

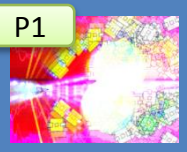
P3 P4 P5

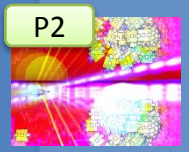
レベルC
(0.50-0.75)



P3 P4 P5

レベルD
(0-75.1.00)

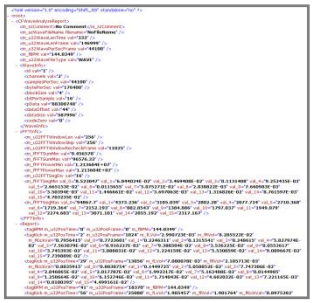




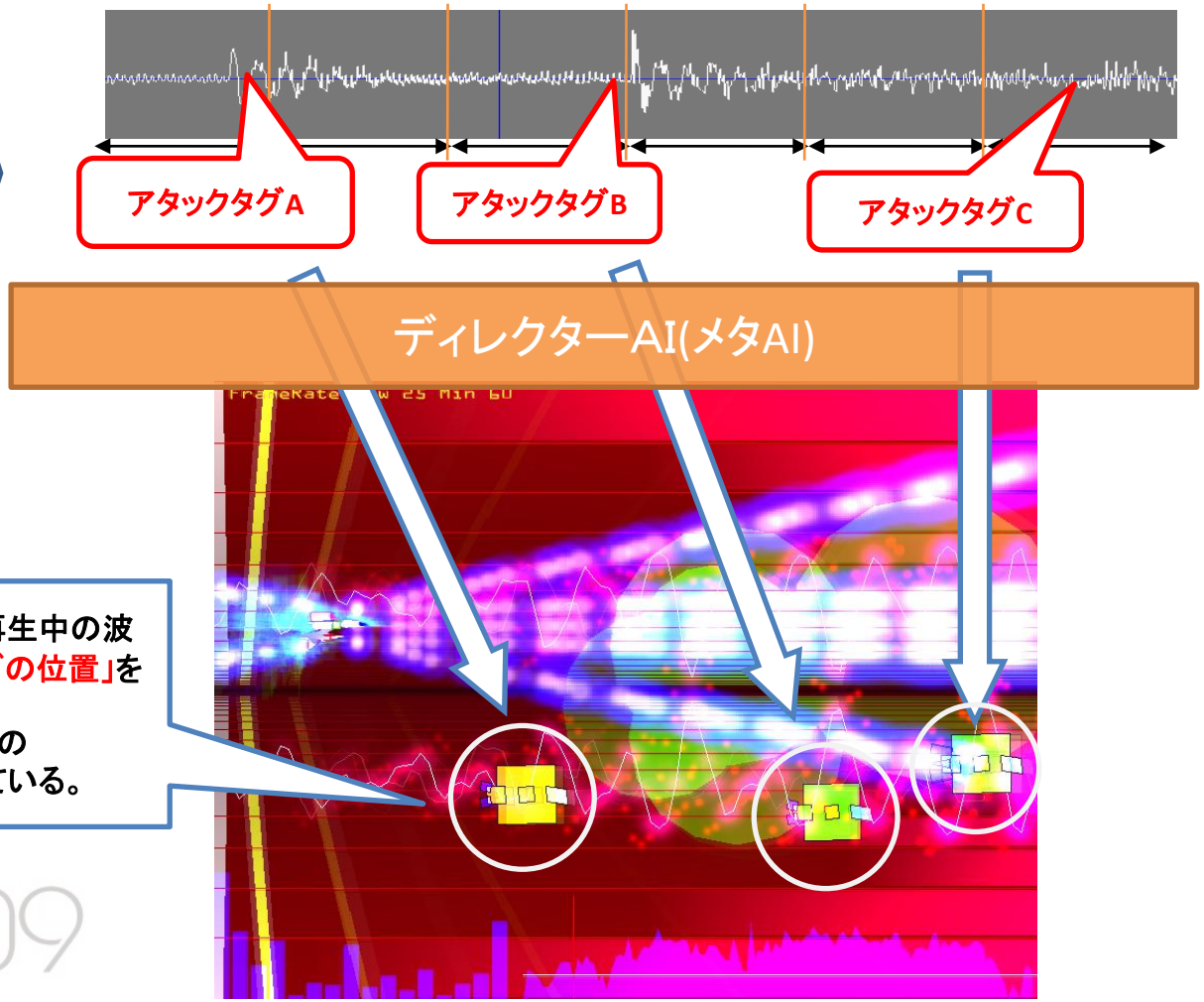
P3 P4 P5

敵の発生タイミング

- ディレクターAIが波形データのアタックタグを元に、
  現在再生中の波形データの値を利用して、
  敵の発生タイミングと表示座標を計算しています。



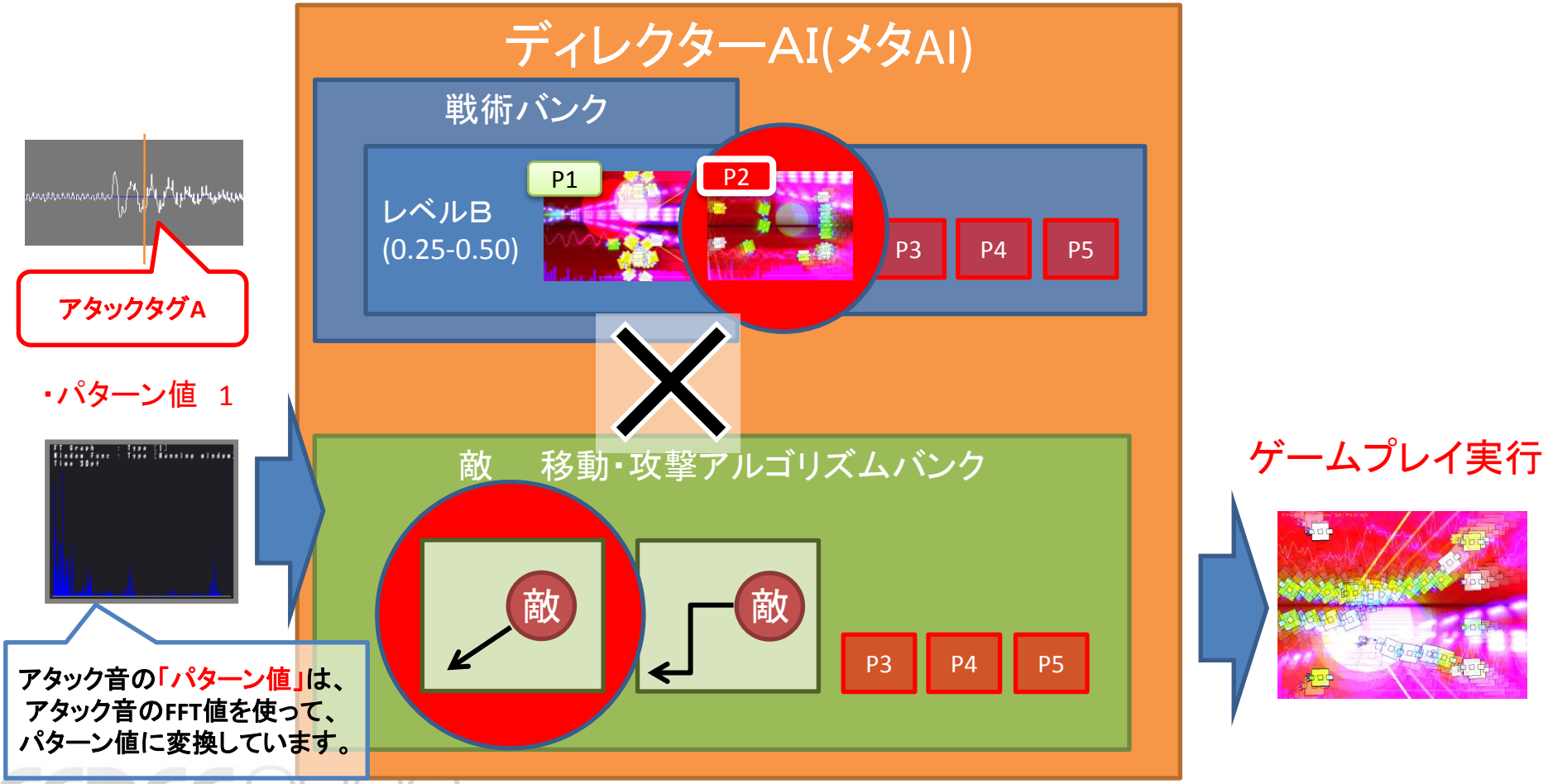
音楽データ
解析レポート
(C++のクラス,XML)



敵の「X座標」は、現在再生中の波形データの「アタックタグの位置」を基準にしている。
「Y座標」は、波形データの「音の強さ」を基準にしている。

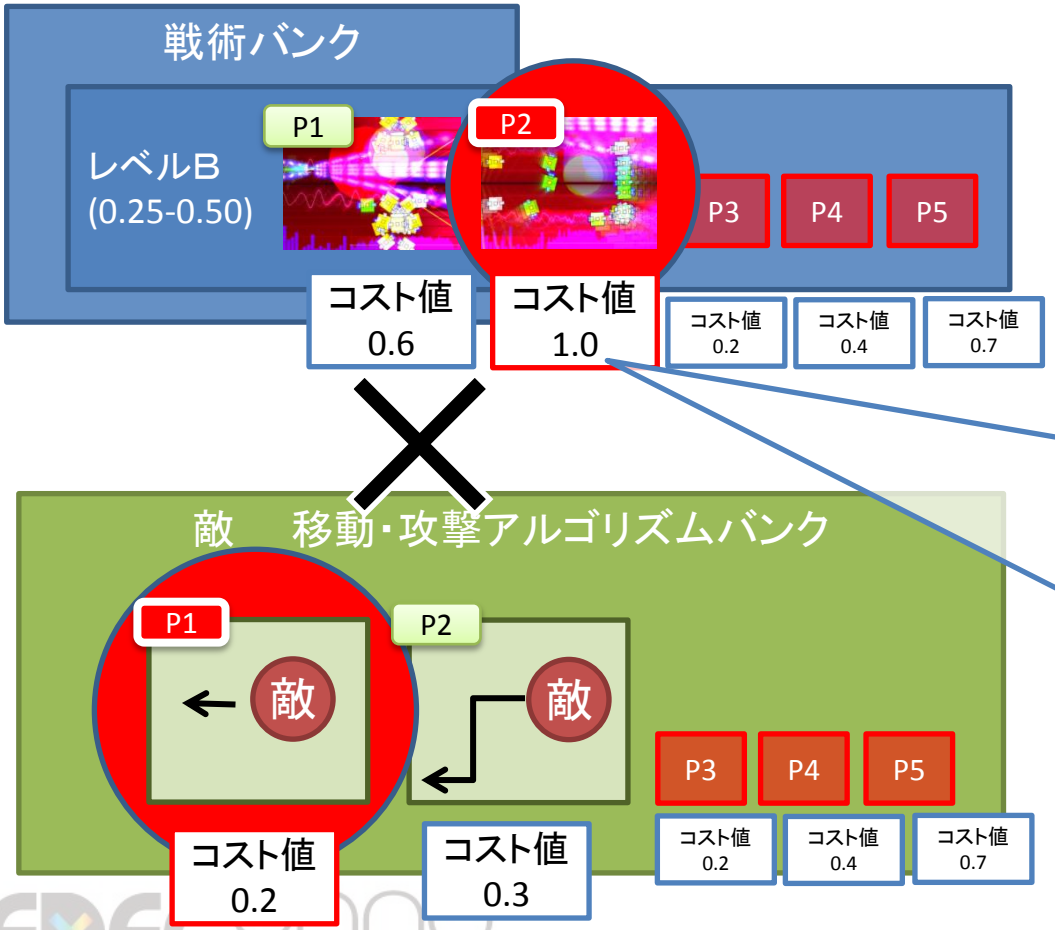
敵のアルゴリズム選択-1

- ディレクターAIは、「フォーメーション(戦術)」×「移動・攻撃アルゴリズム」を組み合わせて、敵のアルゴリズムを作成しています。



敵のアルゴリズム選択-2

- ディレクターAIは、戦術バンクからフォーメーションを選択したあと、
フォーメーションで設定されている「コスト値」以内で、敵AIを作成するように調整します。
これは、「処理落ち」や「ゲームの難易度」を調整するために使われます。



戦術バンクから選択した「フォーメーション」のコスト値が1.0で、「敵」のコストが0.2の場合、
下記の計算で、フォーメーションの中で使える敵の数などが決まります。

[敵の数の計算]
1.0(フォーメーションコスト) / 0.2(敵コスト) = 5機

[その他]
音楽が盛り上がった場合は、コスト値に補正計算を行い、敵の出現数を増やすなどの使い方もしています。

作ってみての感想 - 1

Planning

・制作当初、適当に音のアタックに合わせて、
敵をだせばそれっぽく見えるだろうと思ったら、
ランダムにしか見えなかった。

・曲ごとに、バリエーションの違いが分かるくらいの
見た目の違いを加えようとすると膨大なプリセットの数を用意して、
見た目の変化が分かりやすい敵や弾幕のバリエーションを作る必要がある。

プロシージャルだからといって、作業コストが下がるわけではない。

作ってみての感想 - 2

Planning

・Audio Surfは、これらの問題を解決するために、
優れた企画(デザイン)がされており、
音楽に合わせて出てくるプロシージャル要素は「ブロック」と「コース」だけで、
それ以外はゲーム性にあまり係わらないようにしている。
(演出的に対応させているものは、たくさんあります)

その代わりにプレイヤーキャラのバリエーションを増やすことで、
最小のコストでゲームデザインの深みを出すことに成功している。

7.まとめ

「企画」から見たプロシージャルを使う場合の注意

Planning

- プロシージャルで作成したい要素と
  自動生成したいデータ数の目標最大数を必ず想定する。

- 目標最大数に対して、用意した要素が少ない場合、
  プロシージャルで生成しても、バリエーションが少ないと思われるか、
  ランダムにしか見えない可能性がある。

- 調整するための時間が非常にかかる。

- プロシージャル技術を主軸にするゲームであれば、UGC(UCC)をお勧めする。膨大な要素の作成をユーザーにも負担してもらえる。
  さらにプレイヤー自身もデザインの楽しさを得られる。
  なのでMODは不可欠！！

- 失敗しないためにも、Audio SurfやSPOREなど
  既存のプロシージャルゲームを遊んでおくことをお勧めします。

「プログラマ」から見たプロシージャルを使う場合の注意

Planning

- プロトタイプ(テストプログラム)は必ず作成する。
  ぶっつけ本番の一発で完成品を作るのは不可能。
- 研究・実験しながらの開発となるため、
  企画が仕様を決定できない場合がある。
  (何がバグか決められない)
- 生成する要素をどこまでコントロールするか、
  事前に仕様を決定しないと、想定外のバグが多発する。
- デバッグに非常に時間がかかる。
- 音楽のリアルタイム解析は、処理速度との戦いなので、
  制作に技術力と多くのコストが発生する。
  また、精度を重視するなら、リアルタイム解析よりも、
  ゲームプレイ前に曲全体を解析する方法の方が良い。
- 失敗しないためにも、Audio SurfやSPOREなど
  既存のプロシージャルゲームを遊んでおくことをお勧めします。

まとめ

- サウンドゲームは一般向けのゲーマーにも有効な題材である。
- サウンドゲームは、楽曲を増やすことで、ゲームをさらに楽しめる題材であり、そのためにはプロシージャル技術は不可欠である。
- プロシージャル技術は、サウンドゲーム以外の面でも、
  音楽解析の技術は作業の効率化などに役立つ
- プロシージャル技術は、MODなどを提供してUCC(UGC)展開を行うことで、さらなるプレイヤーの遊び方を広げることができまる。
- なによりも開発者が作っていても、ゲームプレイが楽しい。
  想定外のデータができたり動作などをすると感動！！

最後に

プロシージャル技術の可能性は無限大！！

新しいこと、誰もやっていないことへの挑戦は大変ですが、
サウンドゲームとプロシージャル技術の未来には、
それだけの価値があると信じています！！

挑戦してみませんか？　新しいゲームデザインに！！

新しいゲームデザインに挑戦するために、
AI、プロシージャル、音楽解析などを研究しています。

興味のある方は、ご連絡をお待ちしております。

参考資料

参考資料

- 楽しい電気楽器　自作のススメ
  米本 実[著] オーム社　ISBN978-274-06732-7
- よくわかる最新音響の基本と仕組み
  岩宮 眞一郎[著] 秀和システム　ISBN978-4-7980-1656-6
- C言語ではじめる音のプログラミング
  青木 直史[著] オーム社　ISBN978-4-20650-4
- ディジタル・サウンド処理入門(ディジタル信号処理シリーズ)
  青木 直史[著] CQ出版社　ISBN978-4-7898-3090-4
- フリーソフトで作る音声認識システム
  荒木 雅弘[著]　ISBN978-4-627-84711-8

- CEDEC 2008 IMAGIRE DAY(3) ゲーム開発のためのプロシージャル技術の応用　今給黎　隆氏
- CEDEC 2008 IMAGIRE DAY(4) プロシージャルグラフィックス - 理論と実践)　三宅　陽一郎氏

(上記資料は、CEDECのアーカイバにて閲覧できます)

プロシージャルツール

Planning

- Processing
  http://processing.org/
  Casey Reas と Benjamin Fry によるオープンソースプロジェクト。
  電子アートとビジュアルデザインのためのプログラミング言語で、
  リアルタイムなビジュアライザ制作などに向いている。

- ChucK
  http://chuck.cs.princeton.edu/
  オーディオプログラミング言語。
  簡単スクリプトでループサウンドなどを作成することができる。
  また、サウンド再生中にもスクリプトの変更をリアルタイムに反映することができる。

- Quartz Composer(Mac OS X 標準ソフト)
  MacのスクリーンセーバーやiiTunesのビジュアライザに使用されている。
  GUIツールで、視覚的にスクリプトを組み立てることが出来る。
  そのため、プログラム知識がなくてもビジュアライザの作成が可能。

最後に

ご静聴ありがとうございました。

お問い合わせ

オープランニング・大野 功二
E-Mail skyfox@edit.ne.jp
skyfox@mail4.alpha-net.ne.jp
TEL 072-224-0773

この度の講演内容、および、お仕事の相談などありましたら、お問い合わせください。

- ゲームの企画立案・仕様書作成などをしております。
- ゲームに使える新技術の仕様提案、
  およびプログラムの研究開発・アドバイズなども行っております。
- ゲームAI・サウンド解析・プロシージャル技術に関する基礎研究を行い、
  ゲームおよび映像イベント用システムなどの開発も行っております。
- その他、ゲーム開発におけるご相談などありましたら、よろしくお願いします。

ホームページ http://www.edit.ne.jp/~skyfox/