演奏に対応して変化するビジュアライズの作成

5 年 C 組　國見 幸加
指導教員　藤野 智美

１．要約

本研究では、聴くものとして捉えられている音を、「見るもの」として捉えることで、新たな音の楽しみ方を見つけられるのではないかと考え、MIDI キーボードとプログラムを組み合わせ、音を視覚的に表現するツールの作成を目指した。物理エンジンを使用した、演奏に対して偶発的な反応を示す映像の製作手法を記述する。

キーワード　MIDI、物理エンジン、ofxbox2d

２．研究の背景と目的

私は習い事や部活で楽器を弾く機会があり、もともと音楽に興味があった。また、観客を巻き込むことで表現を成立させるインタラクティブアートにも興味があった。そこで､MIDI キーボードとプログラムを組み合わせ、観客と作品によって完成される対話型アートの作成を行うと同時に、音を見て楽しむという新たな音楽の楽しみ方を試みた。

使用したツール

- openFrameworks（ クリエイティブコーディング用の C++ ツールキット)
- MIDI(Musical Instrument Digital Interface)キーボード

図 1：
MIDI キーボード

３．研究内容

３.１　機能の拡張

addon を使用

addon とは openFrameworks の機能を拡張するためのライブラリ。今回は ofXBox2D と ofxMidi を使用する。

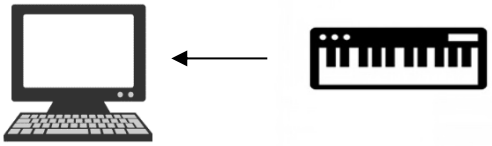
- ofxMidi　：MIDI の生成に関するライブラリ。
- ofxBox2D : 重力や反発力，摩擦，衝突判定といった物理計算を高速に行う物理エンジンのライブラリ。　※ここまで OK

３.２　MIDI に関するプログラムの作成

MIDI とは

電子楽器の演奏情報を機器間で効率良く伝達するための共通規格であり、音の波形そのものを伝達するのに比べ約 100 分の 1 に圧縮された効率のよい演奏情報である。

MIDI 搭載の電子楽器を演奏すると､その MIDI OUT 端子からは常に演奏情報が出力される。流れる信号はデジタルデータのため、コンピュータに記録させることが可能である。



MIDI IN　　MIDI OUT

図 2: MIDI による演奏情報の伝達

３.２.１　MIDI の入力に合わせてパソコンから音を出力

＜方法＞

(1) 3 つのインスタンスを用意し、インスタンス化を行う。

インスタンス化とは

クラスを基にした実際の値としてのデータを生成すること。今回は 3 つのインスタンスを使用する。

①midiIn：MIDI の入力を受け取るためのインスタンス。

②midiMessage：MIDI 入力データを保存するためのインスタンス。受信した midiMessage の情報のうち、

・キーボードが押されたか離されたか

・キーボードの押されたキーの高さ

の 2 つを今回は使用する。

③manager：音を出力するためのインスタンス。

(2) MIDI の入力を受け取る。

MIDI キーボードからの入力があれば入力データを保存する newmidiMessage という型を呼び、audioRepuested という型で MIDI キーボードの入力を元に音を生成する。これらの型を用いてキーが押された時、キーの高さによる音のデータを作成する。

(3) 出力の設定

出力する音の波形に関する設定として、ofSoundStreemSetup という関数を使用する。波形を滑らかに調節して音の波形に近づける。この関数が含む内容は以下である。

ofSoundStreemSetup(出力チャンネル, 入力チャンネル, コールバック, サンプリングレート, バッファ数)

(4) 音データの構築

出力する音の波形を生成して再生できる準備が出来次第、新しい音の波形を作って再生する。

＜結果＞

音がパソコンから出力されることが確認できた。

３.３ofxBox2D に関するシステムの作成

３.３.１　MIDI からの入力に合わせて物体を落下

＜方法＞

(1)物理世界の生成

物理パラメータ(密度・弾性係数など)を設定し、物理法則を適用した世界を生成する。

(2)物体を追加

ofxBox2D にはあらかじめ用意されている物理法則を適用可能な基本図形を生成する。

<結果>

図 3: 物理法則を適用した物体の生成

(3)複数の物体を追加する

Vector 配列を使用して、大量の物体を追加する。

(4) MIDImessage の pitch 数に合わせて物体を生成する

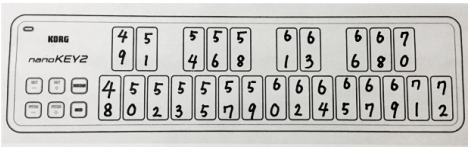


図 4: pitch 数(48～72)

If 文で条件分岐し、pitch ごとに異なる形が落下するように設定する。

＜結果＞

生成された物体は自由落下しながら，画面下方へと消えた。

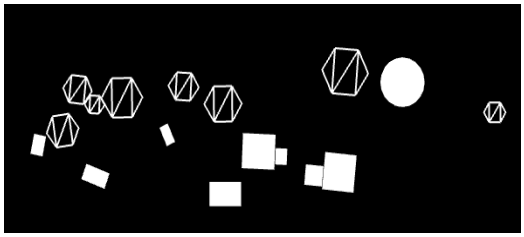


図 5: 実行結果

３.３.２ 障害物を作る

＜方法＞

createGround (始点の x 座標，y 座標，終点の x 座標，y 座標) を使用する。

＜結果＞

生成したボールは線上に溜まったあと、左右から落ちていった。

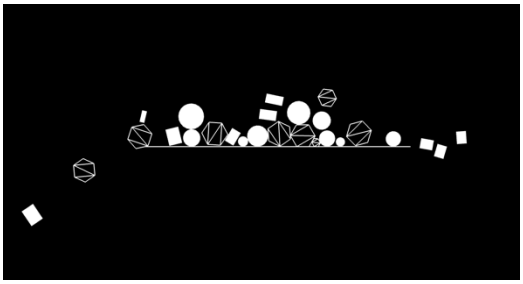


図 6: 実行結果

３.３.３障害物となる線を観客が自由に引けるよう設定

トラックパッドの操作によって線を表示できるように設定する。

＜方法＞

トラックパッドを押しているか、離しているかでそれぞれ型を用意する。

＜結果＞

生成した形は線で囲ったことにより、画面下方に消えず引かれた線上にとどまった。

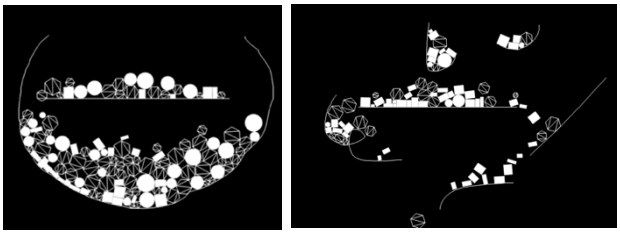


図 7: 実行結果

４．今後の展望

①生成された物体をリセットできる設定の作成を行う。

②一定数の物体がたまれば映像が変化する仕掛けの作成を行う。

③プロジェクターによる映像の投影を行う。

５．参考文献

[1] 小嶋秀樹 | 研究室

https://www.ei.tohoku.ac.jp/xkozima/lab/ofTutorial5plus2.html

[2] Life is Tech! Members

https://members.life-is-tech.com/techfile

６．謝辞

本論文の作成にあたり、終始適切な助言を賜り、丁寧に指導して下さった顧問の藤野先生に感謝します。また、美術部教師として長谷先生からは本研究の制作におけるデザインについて重要な示唆を賜りました。厚く感謝を申し上げます。そして研究の相談にのってくれた美術部員の皆様とサイエンス研究会の皆様にはこの場を借りて深く御礼申し上げたいと思います。本当にありがとうございました。