

1. はじめに

現在、教育現場に情報機器の導入が進み、児童生徒だけでなく、教師にも情報機器を扱う能力が求められるようになってきた。それに合わせて、ソフトやアプリケーションを扱う機会も多くなり、以前までは情報機器の扱いに不得手だった教師にもその能力が求められるようになってきた。そこで、情報機器の扱いに不得手の教師でも扱いやすく、また教育現場において実践的なソフトの開発を研究することにした。

次に、プロジェクションマッピングの活用について考えたい。現在学校現場でのプロジェクションマッピングの使用例は多くなく、そのほとんどが学芸会等のエンターテイメントを目的としたものである。実際に授業で教材として扱われる事例はほぼ無かった。そこで、プロジェクションマッピングを効果的に使うことで、授業の効率化、学習の質の向上が計れないかと考えた。

具体的な研究内容として、始めに初心者が扱うのを目的とした「動画編集ソフトの開発」、次に「プロジェクションマッピングの教育現場での活用」を進めてきた。



図 1.ピアノ廊下の様子

2. 動画編集ソフトの開発

編集環境を OSにとらわれないようにするため、Web上で動作するもの。編集機能は、初心者でも扱いやすいように、動画のカットやテキストの挿入等簡素で必要最低限のものとする。レイアウトを見やすくし、直感的な操作で扱えるようなデザインにすること。以上のような動画編集ソフトの開発を考えていた。

動画編集ソフトの開発は、現在まで JavaScript と HTML5 の勉強をしてきたが、現時点で完成の見通しが立てられていないのが現実であり、実際に Web 上で動作する編集ソフトを作ることが可能なのかも調べたが、確認できなかったため、開発を断念した。

3. プロジェクションマッピングの活用

今回、プロジェクションマッピングを活用した教材を作ろうと考えたが、プロジェクションマッピングを目的としたものを作成するのは今回が初めてであったので、始めに、自分が制作できそうな作品を作成することで、プロジェクションマッピングの作品作成に慣れ、最終的な作品作りにつなげていこうと考えた。

今回は、以前研究室で行われていた「ピアノ廊下」をもとに、ピアノの鍵盤を映像にして、プロジェクションマッピング作品にしてみる。ピアノ廊下は、Arduino と近接センサを用いて、近接センサの前を障害物が通過すると、そこから、キーボードの信号を受け取って、対応するキーボードに応じたピアノの音を鳴らすというものであった。

今回、プロジェクションマッピングとして「ピアノ鍵盤」を作成することにあたって、Processing を使用した。Processing とは、キャセイ・レアスとベンジャミン・フライによるオープンソースプロジェクトで、プログラミングによる視覚的な技法を簡単に表現するために作られたプログラミング言語である。Processing によるプログラムはすべて Java で出力され、Windows、Mac ともに使用でき、どちらで動かしても同じファイルを開くことができる。



図 2. Processing

<https://processing.org/>

今回の研究ではまず初めに、Processing に慣れるための練習として簡単なプロジェクションマッピング作品の制作をした。それが、「ピアノ鍵盤」である。以前鶴川研究室で行われていた「ピアノ廊下」では、Arduino でプログラムされ、近接センサをつなぐことで、センサの前を障害物が通ると、それに反応してピアノの音が鳴る仕組みだった。そこで、それにピアノの鍵盤の映像を組み合わせようと考えた。

始めは、既存の「ピアノ廊下」を用いて、近接センサとピアノの音を Arduino で、新たにピアノの鍵盤の映像を Processing で作成し、組み合わせようと考えた。この方法だと、Arduino と Processing を連携させて動かす必要があり、そのために、シリアル通信による 2 つの連携を試みたが、うまくいかず、この方法は断念した。

次に、Processing でピアノの鍵盤の映像と、各鍵盤の音を一緒に制御できないかと考えた。調べたところ、Processing に Minim ライブラリというものインポートすることによって、Processing でも音声ファイルを扱えるようになることが分かったので、今回作成する「ピアノ鍵盤」では、すべて Processing でプログラムすることにした。

映像は、ピアノの「ドレミファソラシド」、1 オクターブ分の鍵盤の映像を作成した。背景を黒色に設定し、

その上に鍵盤の形の長方形で白く塗りつぶすようプログラムすることで、再現した。鍵盤の映像は、起ボードで“a”～“k”をそれぞれ入力することで、“ド”～“ド”まで対応する鍵盤が灰色に塗りつぶされるようプログラムしたため、特定のキーボードを押すことによって、映像の鍵盤が押されているように見合えるようにした。今回は初めに「ピアノ廊下」を参考にプログラムを作成していたため、キーボードを押したときと、離れたときに実行される関数、KeyPressed 関数と KeyReleased 関数を使用した。

```

45
46 void draw(){
47   if(keyPressed){
48     if(key == 'a'){
49       noStroke();
50       fill(#808080);
51       rect(0,0,50,288);
52       rect(0,288,76,192);
53
54       minim = new Minim(this);
55       player = minim.loadFile("piano_do.mp3");
56       player.play();
57
58     }
59   }
60   if(keyPressed){
61     if(key == 's'){
62       noStroke();
63       fill(#808080);
64       rect(90,0,40,288);
65       rect(80,288,78,192);
66     }
67   }
68   if(keyPressed){
69     if(key == 'd'){

```

図 3. KeyPressed 関数によるプログラム

ピアノの音に関しては、前述した Minim ライブラリから音声ファイルを読み込むことで解決した。Minim ライブラリとは、Processing に対応した標準サウンドライブラリのこと、これを導入することで、Processing でも短いコードでオーディオのコントロールができるようになる。今回は、“ドレミファソラシド”の各音階の音を音声ファイル (MP3) で準備し、Processing と同じ階層のフォルダの中にそのファイルを置いておくことで、呼び出せるようにした。こちらも鍵盤の映像と同様に、KeyPressed 関数を用い、特定のキーボードを押したときに対応する各音階の音声ファイルが再生されるようプログラムした。

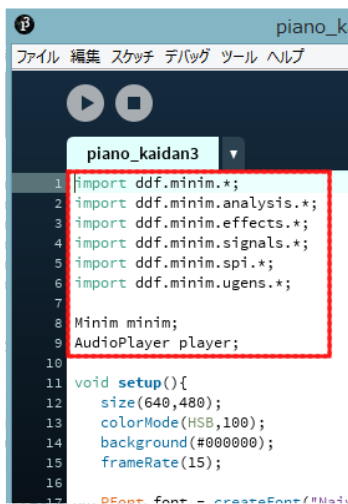


図 4. Minim ライブラリ

実際に動かしてみると、キーボードの“a”を押したときに、映像上の“ド”の鍵盤が灰色に表示されるのと同時に“ド”の音声ファイルが再生され、“s”を押した

ときには、鍵盤の“レ”が灰色に表示されるのと同時に“レ”の音声ファイルが再生される、といったように、簡単にはあるが、プロジェクションマッピングによって擬似的にピアノを演奏できるようなプログラムを作成することができた。

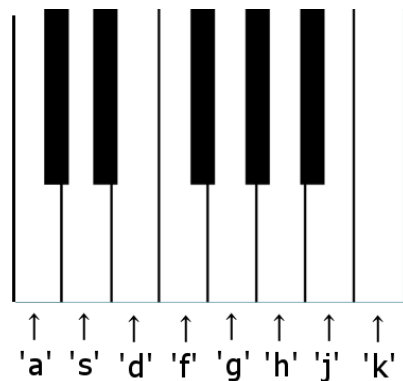


図 5. キーボードを押す前の状態

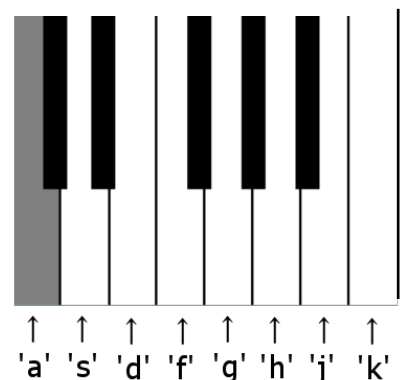


図 6. キーボード“a”を押した状態

「ピアノ廊下」では、近接センサによって、センサの前を障害物を通るだけで、音を鳴らすことができたが、「ピアノ鍵盤」では技術不足もあり、センサをつなげることができなかった。そこで、映像を投影するためにプロジェクターとつないでいるパソコン本体と離れてもプロジェクションマッピングを制御できるよう、「ピアノ鍵盤」では Bluetooth キーボードをつなぐことで、離れた位置からでも制御できるようにした。

4. 今後の展望

今後は、プロジェクションマッピングの方を進めていきたい。まずは、「ピアノ鍵盤」の完成を目指す。現段階ではキーボードによる遠隔操作で動かしているが、この後、近接センサをつなげたり、ペダルのようなものを準備してそれを接続することによって足で操作できるようにしたりして、現実的な使用を目指す。

その後、授業で使えるプロジェクションマッピングの教材や、学芸会等で使えるプロジェクションマッピング作品の研究・制作を進めていきたいと考えている。

5. 参考文献

[1] Processing
<https://processing.org/>