PDF Compressor Pro

公立はこだて未来大学 2015 年度 システム情報科学実習 グループ報告書

Future University Hakodate 2015 System Information Science Practice Group Report

プロジェクト名

シンクロ現象と音楽・映像表現

Project Name

Musical and Visual Expression of Best Synchronization

グループ名

A グループ

Group Name

A Group

プロジェクト番号/Project No.

20

プロジェクトリーダ/Project Leader

1013073　勝原望　Nozomi Katsuhara

グループリーダ/Group Leader

1013073　勝原望　Nozomi Katsuhara

グループメンバ/Group Member

1013013　田邊峻眞　Ryoma Tanabe
1013026　小澤繁輝　Shigeki Ozawa
1013032　立石純基　Junki Tateishi
1013073　勝原望　Nozomi Katsuhara
1013087　伊藤知幸　Tomoyuki Ito
1013123　松本裕史　Hiroshi Matsumoto
1013153　斎藤健　Takeru Saito
1013217　坂田聖名　Sena Sakata
1013243　土手梓　Azusa Dote

指導教員

中田隆行　花田光彦

Advisor

Takayuki Nakata　Mitsuhiko Hanada

提出日

2016 年 1 月 20 日

Date of Submission

Janualy 20, 2016

PDF Compressor Pro

概要

PBL(Project Based Learning) とは、「自立学習の育成」を目標とした教育方針のことであり、プロジェクト形式による学生自ら課題解決することを目的とした教育形態である。PBL では、具体的な課題を基に学生は意欲的に学習に取り組み、その過程で通常の座学では得られない実践的な力 (課題解決能力、論理的思考力、プレゼンテーション能力など) を獲得することができる。これは、医療や工学など実際の職場でも活用できるスキルとして多様に展開されている。本プロジェクトでは、この学習方法でシンクロ現象に関する先行研究を参考に課題を設定し、解決することを目的とする。シンクロ現象とは、動作の周期やタイミングなどが同一に揃う現象のことである。この現象は、人の意識の有無で種類別でき、自然界や日常生活の中でも現れる。意識をしない場合は、友人と並んで歩くと歩幅が合っていく現象である。意識する場合では、オーケストラの指揮者のようなある対象を軸に大勢が作業を合わせることが挙げられる。ジェノバ大学での Antonio Camuuri ら (2010) による先行研究では、同期できた人数により流れる単一音の数が変化し、それから生まれる心理的効果を研究していた。本プロジェクトでは、同期できた人数により、音楽や映像などの様々な表現を形成していくシステムを開発した。前期の活動では、まず同期判定を行うシステムを開発するために、音楽とマルチメディアに長けている Max 言語をインストールし、基礎的な技術を習得した。その後、メンバーをグループに分け、各グループからシンクロ現象を用いた様々な表現のアイデアを募った。それらのアイデアをグループごとに発表し、話し合った結果、同期判定によって音楽を形成していくシステムを作成することを前期の目標とした。前期のシステムでは、映像班、同期班、音楽班の 3 つの班をつくり、それぞれで作業を分担した。映像班では Processing を用いたカメラによる入力を基に、3 人の右手の動きを追跡し、右手を振った判定を行うプログラムを作成した。同期班では、映像班のシステムから 3 人の右手の動きのデータを受け取り、動きがそろえば同期したと判定するプログラムを作成した。この判定の際には、同期していないとき、2 人が同期したとき、3 人が同期したときの 3 つの場合に分けた。音楽班では、映像班で判定した同期の人数に合わせて、音楽を形成するシステムを作成した。この音楽は、同期していない状態のベース音から始まり、同期した人数が増えれば音楽のパートが増え、メロディとなっていくものである。この 3 つの班で作成した各システムを Open Sound Control(OSC) と呼ばれる通信プロトコルを用いて接続し、これを前期の成果物とした。前期の成果物では、同期判定のずれや音楽がリズミカルでないことから、同期したことを体感しにくいことが判明した。後期の活動では、これを改善するべく、新たに照明や映像表現を加えた班構成で活動した。映像班は Kinect 班とデザイン班に分かれて活動を行った。Kinect 班では、前期で行ったカメラ入力を Kinect を用いることで、早い動きにも対応できるシステムを開発した。デザイン班では、主に同期班の同期具合を Processing によるアニメーションを用いて映像表現するプログラムを作成した。同期班では、前期のシステムの改善と曲のテンポを変化させるプログラムを作成した。音楽班では、同期を実感しやすい曲を作成した。照明班では、同期判定よって変化する光の表現のプログラムを行った。そして、各班のシステムを接続し、最終発表会でシステムのデモを行った。この課題解決の過程より、私達はシステムを共同で開発する能力を身に付けた。

キーワード　自立学習の育成, 課題解決, シンクロ現象

（※文責: 田邊峻眞）

PDF Compressor Pro

Abstract

PBL(Project Based Learning) is educational policy for "nurturing of learning oneself" and education style for students's own problem-solving by projects form. In PBL, students can learn actively in solving specific problem. In its process, students can acquire practical skills (problem-solving, reasoning, presentation etc.) which they can't learn in usual classroom lectures. It is multiply developed as the skill of which we can make good use in actual places of work such as health care setting and engineering. Purpose of our project is solving set problems using examples from previous study of Synchronization in the method of learning. Synchronization is what cycle of action and the timing correspond. it appear in nature and daily life, We can assort it by presence or absence of human conscious. In no conscious case, the length of stride fit naturally into friend if we walk beside them. In the reverse case, everyone tune in to the object like conductor of an orchestra. Antonio Camuuri etc. in University of Genoa have developed the system which the number of musical component changes by the level of synchronization headcount. We developed the system which can make various expressions of music light box and animation by synchronized headcount based on this previous study of synchronization. We learn skills for developing the system communally in the process of problem-solving. In first term act, we aquired fundamental skill of Max that is trained in music and multimedia to develop a system of synchronization determination. we divided member into group and scout each groups for various ideas of synchronization. We made a presentation ideas for each groups. Then, we discussed and made it a goal to made a system that form music by synchronization determination. In system of previous term, we made three groups for a division of labor as visual group, Synchronization determination group and music group. Visual group made a program that chase three person's right hand and judge handwaving from information of camera by Processing. Synchronization determination group made a program that receives information of three person's right hand from visual group and judge how many people synchronize. We divided synchronization determination into "not synchronization" and "synchronization of two person" and "synchronization of three person". Music group made a system that form music by the level of synchronization. This music change from bass note to melody by the level of synchronization. Bass note come when the level of synchronization is low, melody come when the level of synchronization is high. We connected these systems through Open Sound Control (OSC) to each other. Then, this system was the deliverable in previous term. We have found that we can't feel synchronization with the deliverable in previous from gap of synchronization determination and not rhythmical music. In later term act, we acted for improving this system in new consist of group by expressions the illumination and visual. Visual group divided into Kinect group and Design group. Kinect group developed a system that keeps pace with changes of fast-paced action with Kinect. Design group made a program that can express the level of synchronization with animation of Processing. Synchronization determination group made a program that changes BPM and improved the system of previous term. Music group made some music that we can easily feel synchronization. The illumination group made music light box that light changes by synchronization. Then, We had demonstration of the system connected over OSC.

Keyword Nurturing of learning oneself, problem-solving, Synchronization

（※文責: 田邊峻眞）

PDF Compressor Pro

目次

第 1 章　背景 **1**
1.1　本プロジェクトの背景 1

第 2 章　到達目標 **2**
2.1　本プロジェクトにおける目的 2
2.1.1　具体的な目標設定 2
2.1.2　通常の授業ではなく、プロジェクト学習で行う利点 2
2.2　具体的な手順・課題設定 2
2.3　課題の割り当て 4

第 3 章　課題解決のプロセスの概要 **5**

第 4 章　課題解決のプロセスの詳細 **13**
4.1　前期 13
4.1.1　映像班 13
4.1.2　同期班 16
4.1.3　音楽班 18
4.2　後期 20
4.2.1　Kinect 班 20
4.2.2　同期班 26
4.2.3　音楽班 29
4.2.4　照明班 36
4.3　デザイン班 41
4.3.1　デザイン班ができるまで 41
4.3.2　目的 41
4.3.3　システムの詳細 42
4.3.4　システムの展望 44

第 5 章　結果 **45**
5.1　前期 45
5.1.1　前期の結果 45
5.1.2　中間発表 45
5.2　後期 47
5.2.1　後期の結果 47
5.2.2　最終発表の成果 47
5.3　成果の評価 48

第 6 章　今後の課題と展望 **51**
6.1　前期の課題と展望 51

PDF Compressor Pro

6.2 後期の課題と展望 51
6.3 今後の展望 52
6.3.1 ユーザーのモチベーションを上げたい 53
6.3.2 全く新しい表現を提案する 53
6.3.3 実験 53
6.4 中間発表のアンケート結果 54
6.4.1 中間発表の日時・場所 54
6.4.2 アンケート結果 54
6.5 最終発表のアンケート結果 55
6.5.1 最終発表の日時・場所 55
6.5.2 アンケート結果 56

付録 A 新規習得技術 57

付録 B 活用した講義 58

参考文献 59

PDF Compressor Pro

第 1 章　背景

1.1　本プロジェクトの背景

私たちの日常では、個人の動作や行為が人や自然に合わせて同期することがある。2 人で振り子を振った際に、振れる波長が自然に合っていく現象や、2 人が並んで歩くと歩幅が合っていく現象などが、同期の現象である。このような現象は、シンクロ現象と呼ばれている。また、このように無意識で起こるシンクロ現象のほかに、意識的に起こるシンクロ現象もあり、オーケストラでの指揮者と演奏者、複数人でのダンスなどが例に挙げられる。

我々は、本プロジェクトにおいて、この意識して起こる場合のシンクロ現象を取り上げる。Cirelli,Einarson,Trainor(2014) の乳児を対象とした研究では、乳児と実験者にシンクロした動きをさせた後、実験者がモノを落とした際に乳児がそれを拾って実験者に渡しに行くなどの援助行動をする可能性が高くなることがわかっている。つまりシンクロは人同士に友好的な印象を与え得る。

また、イタリアのジェノバ大学では、Varni, Mancini, Volpe, Camurri(2010) により、モバイル端末を利用したシンクロ現象の研究が行われている。彼らは、被験者数名にモバイル端末を渡し、リズムに乗ってそれを振ってもらい、同期されていると音楽が再生され、さらに同期が進むと音楽が変化していくアプリケーションを作り、シンクロ現象がもたらす心理的効果を調べる実験を行った。これによって、シンクロ現象が人の内集団に対する親和行動を促すなど心理的効果を与えることがわかっている。

また、山崎 (2015) によると、集団での歌唱とダンスを中心とするワークショップに参加することが共感性や自尊感情、個人志向性・社会志向性に及ぼす影響について調査した結果、共感性を有意に高めることが示された。また、「他者に注目しつつ、他者と体の動きや発声を同期させていく必要がある。こうした活動に従事することが、共感性を高めたのかもしれない。」と言及している。従って、音楽に合わせてリズムを取って動くことによる有意性が確認できる。

本プロジェクトではジェノバ大学の研究を先行研究とし、複数人が身体の動きでリズムを取ることで音楽が形成されるというアイディアを取り入れ、発展させる。そして、シンクロ現象によるテンポ同期が人に与える心理的な効果がどのようなものであるかを明らかにすることを課題とする。

（※文責: 齋藤建）

PDF Compressor Pro

第 2 章　到達目標

2.1　本プロジェクトにおける目的

本プロジェクトは、「心理学的なシンクロ現象に応じて複数人の右手の動きについての同期判定を行って音楽や映像が変化するシステムを構築し、人に対するテンポ同期の心理的効果を明らかにすること」を目的として活動する。

（※文責: 立石純基）

2.1.1　具体的な目標設定

上記で示した目的を達成するために私たちは、「右手の動きの同期の程度に応じて変化する表現方法を探求し、それによる心理的な効果を体験できるシステムを開発する」という目標を設定した。

また、具体的にどのようなシステムを作成するかプロジェクトメンバー内で話し合った結果、いくつかの条件に合わせたシステムを作成することになった。その条件は、以下のとおりである。

- 映像に合わせて人が動くのではなく、人の動きに合わせて表現が変化する。
- システムの利用者が楽しめる表現方法である。
- 同期を確認しやすい表現方法である。
- 同期を確認しやすい動きを使うシステムである。

これらの条件をもとに、作成するシステムは「人と人との動きの同期具合によって、音楽が流れる、もしくは映像が変化するシステム」とした。

（※文責: 立石純基）

2.1.2　通常の授業ではなく、プロジェクト学習で行う利点

通常の講義は、同じ学科・コースの人ばかり集まって行われていることが多く、意見が偏ってしまう可能性がある。しかし、プロジェクト学習は、他のコースの人が集まってメンバーが形成されているため、様々な方向から意見が寄せられる。また、システムを作成する際に、それぞれの専門分野を生かした役割分担を行うことで、効率的に成果物を作成することができる。

（※文責: 立石純基）

2.2　具体的な手順・課題設定

本プロジェクトは、以下のような手順で進められた。

1. Max (Cycling '74, San Francisco) [1] のインストールおよび技術習得
   課題：ヴィジュアルプログラミング言語である Max をインストールした。メンバー全員が

PDF Compressor Pro

Max の基礎的な技術を習得した。

2. シンクロの程度を反映する音楽・映像表現のプロトタイプの制作
   課題：メンバー全員で 1 つの制作物を決め、いくつかのグループに分かれて制作を行った。シンクロを用いた音楽・映像表現が動作するシステムの実現を目指した。また、手順 2 では、映像班・同期班・音楽班の 3 つのグループに分かれて作業を行った。各班での作業は以下のようにした。
   - 映像班
     Processing (Casey Reas, Benjamin Fry) [2] でカメラからの入力をもとに、複数の物体の動きを個別に判定するプログラムを作成した。また、その判定した情報を Max に送る仕組みを作成した。
   - 同期班
     Max を用いて、Processing から送られてくる情報をもとに、複数の物体の動きの同期具合を判定するプログラムを作成した。また、Processing から情報を受け取る仕組みを作成した。
   - 音楽班
     Max を用いて、同期班によるプログラムから判定された同期具合によって、音楽が再生される仕組みを作成した。
3. 前期に作成したプロトタイプから発展した新たなシステムの構想を練った。
   課題：プロトタイプ制作にあたっての問題点、改善点を探し、それを解決する方法を考えた。また、中間発表会でのフィードバックを集計し、これからの活動に活かした。
4. 新たなシステムの制作
   課題：前期に制作したプロトタイプをより発展、改善させたプログラムを作成した。また、手順 4 では、Kinect 班、同期班、音楽班、照明班、デザイン班の 5 つのグループに分かれて作業を行った。各班での作業は以下のようにした。
   - Kinect 班
     今回は、Kinect (Microsoft, USA) [3] のセンサーを用いて、人体の骨格を取得、追跡することで複数人の動きを個別に判定するプログラムを作成した。また、Kinect を使うために、C ♯の技術を習得した。そして、C ♯のプログラムで Max に送る仕組みを作成した。
   - 同期班
     前期に作成したプログラムを発展させ、利用者の同期具合によって小節ごとに曲のテンポを変化させる機構を作成した。
   - 音楽班
     利用者が同期しやすいと感じられるような音楽を作成した。再生される曲の音色を変更した。
   - 照明班
     利用者の同期具合によって変化する光の表現を作成した。
   - デザイン班
     利用者の同期具合によって変化するアニメーションを作成した。

（※文責: 立石純基）

PDF Compressor Pro

2.3 課題の割り当て

各人の得意分野及び関連性を基準に以下のように割り当てた。

- 前期
  - 映像班
    映像班の課題は、Processing を用いて複数人の右手の動きを追跡し、右手を振ったと判定するプログラムを実装することであった。そのため、プログラミングの講義が多い情報システムコースである立石純基、齋藤建、知能システムコースである土手梓、映像表現に関連のあるデザインコースである小澤繁輝が担当した。
  - 同期班
    同期班の課題は、映像班のプログラムで行った判定をもとに、複数人の動きが同期しているかどうかを判定するプログラムを Max を用いて実装することであった。しかし、メンバーのほとんどが Max に対する知識が不足していたため、Max に興味があった知能システムコースである坂田聖名、複雑系コースである松本裕史、情報システムコースである田邊峻眞が担当した。
  - 音楽班
    音楽班の課題は、同期具合によって再生される曲のパート変化するプログラムを作成すること及びその曲の音楽ファイルを作成することであった。そのため、音楽についての知識がある勝原望、知能システムコースである伊藤知幸が担当した。
- 後期
  - Kinect 班
    Kinect 班の課題は、Kinect を用いることで前期のプログラムよりも正確に右手の動きを判定するプログラムを作成することであった。そのため、前期で動きの判定のアルゴリズムを作成した立石純基、齋藤建が担当した。
  - 同期班
    同期班の課題は、前期に実装したプログラムをより発展させ、曲の再生中に曲のテンポが変化するプログラムを作成することであった。そのため、前期も同期班に所属していた坂田聖名、伊藤知幸が担当した。
  - 音楽班
    音楽班の課題は、同期していると実感しやすい曲を作成することであった。そのため、音楽についての知識がある勝原望、松本裕史が担当した。
  - 照明班
    照明班の課題は、同期具合によって変化する新たな光の表現を作成することであった。そのため、光の表現に興味があった土手梓が担当した。
  - デザイン班
    デザイン班の課題は、同期具合によって変化する新たな映像表現を作成することであった。そのため、デザインについて知識がある情報デザインコースである小澤繁輝、情報システムコースである田邊峻眞が担当した。

（※文責: 立石純基）

PDF Compressor Pro

第 3 章　課題解決のプロセスの概要

1. 利用者に身体を使ってリズムを取る動作を行ってもらう。その時の身体の動きをプログラム上で認識し、利用者一人一人の取っている動きのテンポを捉えるための入力装置とシステムの構築

   解決方法

   - 前期

     本プロジェクトでは、まず利用者に身体を使ってリズムを取ってもらう。その時、利用者同士の身体を動かすテンポが合っている状態を同期している。つまり、シンクロした状態にあるとし、利用者同士が同期した後に互いの間で得られる心理的効果を調べることを目的としている。　最初に必要となるのが利用者が身体を動かしているところを捉え、そのテンポを認識することである。そこで、試作段階も兼ねて一番動きを捉えやすそうな身体の部位として、腕に着目することにした。まず、利用者に各々好きなように腕を振ってもらう、そこからお互いに腕の振りが合うように意識してもらうのだが、その際に 3 人の腕の振り方が徐々に合っていく様子を捉え、それをプログラム上で判断する必要があり、なおかつ 3 人の腕の動きをそれぞれ独立して認識する必要があった。そこで、立石純基、齋藤建、土手梓の 3 名で映像班を作り、利用者一人一人の動作を独立して認識する方法について話し合った。そこで挙げられたのが、利用者の 3 人を区別するためにそれぞれに異なる色の紙を持たせる方法である。それにより、プログラム上でも色ごとに独立して認識することが可能になった。　そして、その利用者がそれぞれ色の違った紙を持っている様子をノートパソコンに内蔵されたカメラで捉え、その色を Processing によるカラートラッキング (図 3.1) という方法で色ごとにポインターを配置する。すると、そのポインターがそれぞれ指定された色を追跡出来るようになり、それによって 3 人の動きを独立して認識することに成功した。　次に、追跡した利用者の腕の動きから、利用者たちが同期しているかどうかを判定するために各々の腕を振るテンポを認識する必要があった。そこで、腕を上下に振るにあたって、利用者の間でテンポがずれた時に生じる違いというのは、腕を上から下へ、下から上へ切り返す瞬間ではないかと考えた。そこから、その色紙を持った手を上下に動かしてもらう際に、手の座標が上下に移動することと、手の加速度が腕を下から上に変化する瞬間に加速度が 0 になることを利用して、加速度が振り下ろされて 0 になる瞬間を 1 回とし、手を振った回数をカウントするシステムを構築した。　このシステムにより、利用者一人一人の動きを独立して認識するだけでなく、利用者がそれぞれ何回腕を振ったかをカウントすることが出来るようになった。

   - 後期

     前期で作成したシステムでは問題となる点がいくつか存在することが分かった。1 つ目は、素早い動きや細かい動きに対応しきれず、手を上下に振った回数を上手くカウントすることが出来ない問題であり、前期で作成したシステムを利用してもらった際に、あまり早く腕を振っているつもりでない人の腕を振った回数のカウントが上手くできていなかったり、腕を振る幅が小さすぎるとプログラムが腕を振ったと認識せずにカウン

PDF Compressor Pro

トが増えないといった点。また、2 つ目は、色を認識する方法を取っていたことから、利用者に持ってもらっている色紙の色と似た色の背景が存在していたり、その色に近い服を着ていた際に、カラートラッキングにより色の追跡に誤作動が生じる問題が発生した。そこで、そういった動きや問題にも対応できるよう、別のプログラムで認識する方法を引き続き映像班で模索した。しかし、ノートパソコンの内蔵カメラでは、どうしても早い動きや細かいを捉えきれず、プログラムが作動しないことが分かったため、より性能の良いカメラが必要であると判断した。そこで、ノートパソコンの内蔵カメラよりも滑らかに動作を追跡でき、色に関係なく動きを追跡できるデバイスである、Kinect という製品があることを知り、Kinect を用いての検出方法を新たに作成した Kinect 班である、立石純基、齋藤健により構築していくことにした。(図 3.2 参照）Kinect では、人の動きを追跡するために利用者に色のついた紙を持たせる必要がなくなる。そのため、背景や利用者が着ている服の色などに気を遣う必要がなくなる。また、Kinect では人の骨格を認識する能力があり、腕の振りを認識する際にも右手の骨格を捉えるよう設定することで手に何も持っていない状態でも腕の振りを追跡することが出来るようになった。また、Kinect ではノートパソコンの内蔵カメラに比べて、遥かに細かいフレームレートで撮影することが可能なため、これまで捉えることが出来なかった素早い動きや細かい動きにも対応できるようになった。それらの改善点により、利用者は手に何も持たない状態でカメラの前で腕を振るだけで動作を認識でき、腕の振りを認識する範囲を設定し、その範囲内であれば前期で行ったカラートラッキングよりも自由度の高い動きを認識することが可能になったことで、利用者の腕の動きをダイレクトに捉えることが出来るようになった。

2. 利用者たちの動きを独立に入手したあと、それぞれの動きが互いに同期しているかどうかの判定

   解決方法

   - 前期

     映像班では、カラートラッキングで利用者 3 人の動きをそれぞれ独立して認識してもらい、利用者一人一人の腕を振った回数のカウントするシステムを作成してもらった。今度はその情報を用いて、プログラム上で利用者たち一人一人の腕を振るタイミングが同期しているかどうかの判定を行う必要がある。そこで、映像班では色のついた紙を上下に動かしてもらった回数をそれぞれの色ごとにカウントしているので、そのカウントが行われたタイミングが同じかどうかを判定することできれば、その瞬間利用者たちは同期しているということである。ここでは、利用者一人一人の腕を振るタイミングが同じかどうかを判定してくれるプログラムを構築したい。その方法を同期班の坂田聖名、伊藤知幸、田邊峻眞の 3 名で Max（図 3.3 参照）を用いた方法を考案したところ、まず利用者たちの動作のテンポを各々測定し、その差からテンポのずれを測ろうとした。要するにずれが少なければ、それを同期していると判定し、ずれが大きければ利用者たちの動きは同期できていないと判定するプログラムの構築に取り掛かった。しかし、そのずれを腕を振るたびに測定しようとすると、利用者間での腕の振るタイミングに大きな差が生じていたりした場合にプログラムでの測定に誤作動が発生してしまうことが分かった。そのずれを解消するために試行錯誤を繰り返したものの、解決の糸口を掴むことが出来なかった。そこで、新たな方法を考案したところ、ある一定の短い時間を設けて、その時間内に 3 人全員が腕を 1 回振るという動作が完了出来ていれば、それを 1 回とし

PDF Compressor Pro

動作を完了出来ていない人がいた場合カウントせず、そのカウントを点数として可算していくことにした。つまり、利用者たちが持っている色のついた紙を上下に 1 往復する動作を 1 拍とし、3 人のその動作が一定時間内に収まっていれば点数を 1 点可算する。収まっていなければ点数は可算されないという方法を取ることにした。すると、設けた時間内に利用者 3 人の動きが収まっている状況を同期していると見なすことができ、プログラム側で半強制的に同期している状態に持っていくこが出来た。その方法で 2 小節ごとに点数を加算していき、その可算されていった点数を 2 小節ごとにリセットする。そして、2 小節ごとの合計値によって同期していることを表現する方法が変化するプログラムを作成した。

- 後期

  引き続き同期班の坂田聖名、伊藤知幸により、前期のプログラムの改良が行われた。(図 3.4) 改良すべき点としては、1 つ目は設けた一定時間内に 3 人は腕を振るという動作をしたかどうかをカウントしていたことに対して、その方法では一定時間内に腕を振る動作が間に合わなかった人がいた場合に、プログラム上では誰が間に合わなかったかということと、同期出来ていたのはどの組み合わせの人なのかということを判別することが出来なかったことである。また、一定時間内を広く設定してしまうと、同期していると判定する範囲が過剰に易しくなってしまうため、視覚的にタイミングがあっていないように思われる状態でもプログラム上では同期していると判定してしまうことや、逆に一定時間を厳しく設定すれば、カウントが可算されるタイミングがシビアになり過ぎてしまい、実験に支障が出るといった問題が発生した。2 つ目は、最初に各々好きなようにテンポを取ってもらうことから、指定されたテンポが存在しないため、利用者たちが大きく異なるテンポで動作を始めた時に、利用者間で合わせようとするテンポに幅が生まれてしまい、腕を振った回数をカウントするシステムが上手く作動しないという事態が発生したことである。そういった点を改善するためにまず、追加要素として最初の 4 拍ではシステムを動かさずに、4 回の動作が入力される間隔の平均からテンポを決定するようにした。そうすることで、最初に利用者の取るテンポが大きく異なっていたとしても、最初の 4 回の入力でシステム上で決められたテンポで音楽が鳴り始めるようになった。これにより、利用者たちがテンポを取りやすくなっただけでなく、実験の際のテンポにバリエーションが生まれ、利用者同士がより互いを意識してテンポを取れるようになった。また、前期で作成したプログラムでは分からなかった、誰と誰が同期状態にあるかということを把握できるようにするための改善としては、1 拍ごとに利用者 3 人のタイミング情報を取得し、同期している人の組み合わせによって 0,1,2,3,4 のうちのどれかの値を与える。その値によって同期していることを表現する方法を変化させるようプログラムした。また、小節毎にカウントしていた点数を直近 8 回分の平均値を取得し続けるプログラムに変えたことにより、利用者 3 人の同期具合がずれたタイミングと表現が変化するタイミングとの誤差が前期のプログラムより、少なくすることが出来た。

3. 同期しているとプログラムが判定したことを利用者自身にも実感してもらうための音楽表現の構築

   解決方法

   - 前期

     映像班と同期班で利用者の動きを認識し、それが作成された同期具合を示すプログラムの値によって同期していると判定された時に、それを表現するための音楽の作成を音楽

PDF Compressor Pro

班である勝原望と松本裕史によって作成した。利用者たちが同期出来たかどうかを表す音楽表現として、同期している人数や人の組み合わせてによって流れる音楽を変えるといった案が挙げられた。しかし、その方法ではどの楽曲がどれくらい同期している状態を示すのか知りえない利用者にとっては、同期したことを実感できる表現としては弱いのではないかと考えた。そこで、楽曲を作成するためのツールとして Domino と呼ばれるフリーソフト (図 3.5) を使って MIDI ファイル形式で作成していくうえで、Domino での楽曲作成は全て打ち込みで行われ、楽器ごとの演奏を別々のファイルで保存することが出来るという利点に気が付いた。そういった利点を踏まえて音楽の表現の仕方を模索していったところ、作成した音楽をベース音、伴奏、メロディーの 3 つのパートに分け、利用者の同期具合によって流すパートを変化させるという音楽表現をしていくことに決定した。決定した案は、同期班より送られてくる利用者たちの動きがどれくらい多く同期しているかどうかを表す点数によって、流れる音楽を変えようというものである。同期班から送られてくる情報は利用者たちが同期した回数を表す点数であり、その点数は 2 小節単位でリセットされるので、最高点を 8 ということになる。それを利用し、点数が低いときは流れる音をベース音のみにし、点数が増えるにつれ伴奏が加わり、最後にメロディーが加わって完成するという表現方法を考案した。また、同期班から送られてくる点数は 2 小節ごとにリセットされるので、音楽班で流す音楽も 2 小節ごとに切り替わるように設定した。これらの方法により、利用者たちが同期しようと腕を振り始めてから、同期具合が向上するにつれ音楽が豊かになっていき、利用者にとってただ腕を振るのではなく音楽を完成させるという目標と音楽が完成したときの達成感を得られる表現方法となった。

- 後期

  前期で作成した音楽では、利用者にとってテンポを掴みづらいことと、曲調が本プロジェクトに合わないという意見から、新たな楽曲を引き続き音楽班の勝原望と松本裕史により作成することにした。また、前期で作成した楽曲は曲の始まりと終わりがはっきりしているもので、同期するタイミングによって中途半端な位置から音楽がスタートしてしまうといった問題があった。他にも、前期では 2 小節ごとに同期班から送られてくるプログラム上の値によって音楽を変化させるシステムだったことから、点数が高得点であった状態から急に低い点になったりした際に、急に音が消えたり増えたりと変化が著しくなってしまうことがあった。その現象により、利用者たちが同期したと思う瞬間、また同期がずれたと思う瞬間と音楽の表現との間に違和感が生じてしまい、利用者たちが互いの腕を振るリズムが合ったと思ったタイミングや合っていたタイミングがずれたと思うタイミングに音楽の変化がついてこれないことがあった。特に 2 小節ごとに同期班からの値がリセットさせるので、小節の最初で利用者同士の同期がずれた際は、音楽が変化するタイミングとのずれが顕著に出てしまい、望んだ実験結果を得られない状況に陥ることがあった。そういった反省点から、後期では同期班から送られてくる値に合わせて音量を少しずつ変化できるようにした。それにより、音楽が変化していく様子が聞こえるため、より自然な音楽表現ができるようになった。また、後期では勝原望によりオリジナル曲、特に前期の楽曲にあったような途中から音楽が始まったり、途中で切れてしまっても違和感のないような一定のメロディーを繰り返すものの作成と松本裕史により既存の曲のコピーの 1 曲ずつ新しい楽曲作成に取り掛かった。それにあたって前期で作成した音楽の反省点を踏まえ、利用者がリズムを取りやすい 4 つ打ちのもの

PDF Compressor Pro

であることと、テンポが分かりやすいよう音域の低い曲であるという 2 点に注意して作成に取り掛かった。後期で作成した楽曲も前期で作成した楽曲のように 3 つのパートに分け、同期班から受け取った値によって音楽が完成していくよう設定した。

4. 音楽表現に加えて、より利用者に同期しているという実感を得てもらうために新しい表現の考案と追加

   解決方法

   - 照明班

     同期していることを表現する方法として前期では音楽のみの表現方法を用いていたが、それだけでは同期しているという実感を得づらいという意見があり、新たな表現方法を考案する必要があった。その案の一つして挙がった RainbowCube と呼ばれるキット (図 3.6) を照明班である土手梓により作成した。この RainbowCube は 4x4x4 の照明から成る立方体の構造をしており、Processing と連携させることにより各照明を自由に点滅させることが出来るものである。これにより、同期していることの表現の幅が大きく広がった。また、その証明の色も多彩に変化させることが可能で、これまで音楽での表現しかなかったものにこのキットをこれまでのプログラムに連携させることで、同期しているかどうかを視覚的にも認識できるようにした。また、同期班によって送られてくる値によって音楽表現が変化するのと同じように、照明の演出を変化させることが出来る。音楽の表現が同期班から送られてくる値が大きくなるにつれ、曲が完成していくものであったのに合わせて、最初は光の演出が控えめな状態から始まり、値が大きくなるにつれ演出が派手になっていくよう設定した。また、同期班から送られてくる 0,1,2,3,4 の値によってそれぞれ異なる演出を行うようにした。それにより、同期具合の変化に合わせて音楽が変化するとともに、RainbowCube の光の演出が変化するようになり、RainbowCube により目を引く演出によって、利用者が同期した瞬間に得られる達成感や同期に向かう向上心を補助する働きをする狙いである。

   - デザイン班

     同期していることを表現するための表現方法として、音楽や照明よりも同期している状態を理解する手助けとして新たに映像表現を取り入れた。新たに作成した映像表現では利用者 3 人に対応したダンサーをデザイン班である小澤繁輝と田邊峻眞で Processing により作成した。(図 4.21)。ダンサーは利用者一人一人に対応させており、全部で 3 人いる。また、それぞれを服の色で区別できるように赤、青、緑の 3 人に設定した。これらはダンサーとして用意したが、これらのダンサーも同期班のプログラム上で送られてくる値に対応している。同期班のプログラムから同期している人としていない人がいると判定された場合、している人たちのみがダンスを踊るようになっている。例えば、赤と青のダンサーに対応した人が同期している場合、映像内では赤と青の服を来たダンサーがダンスを踊り、残りのダンサーは止まったままである。この映像により、システムを管理している側にしか分からなかった誰と誰がシンクロ出来ているかを利用者が自ら把握しながら実験に取り組めるようになった。これにより、利用者が実験しながらも視覚的に同期しているかどうかを瞬時に確認できるよう、また利用者が体を動かしているということをより実感してもらえるような工夫として新たな表現に加えた。

（※文責: 松本裕史）

PDF Compressor Pro

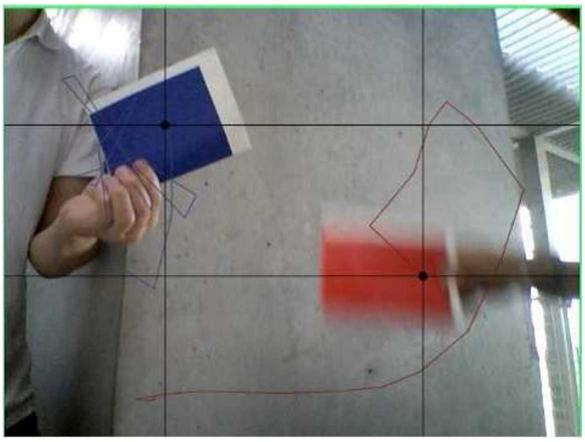

図 3.1　カラートラッキング

図 3.2　Kinect

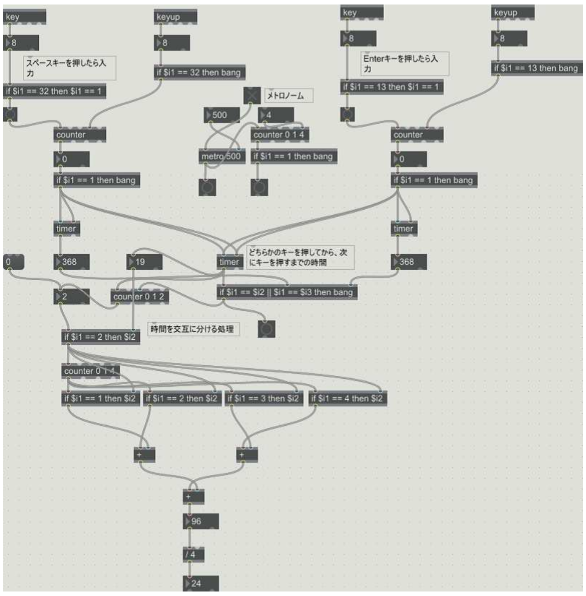

図 3.3　Max

PDF Compressor Pro

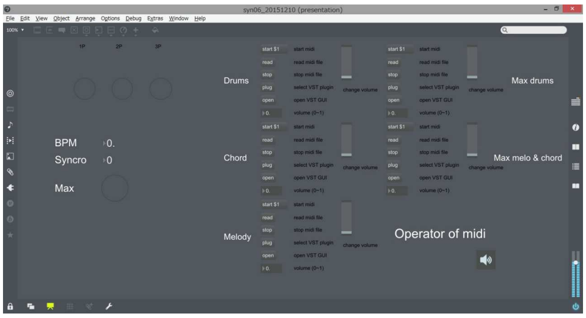

図 3.4　Max　改良版

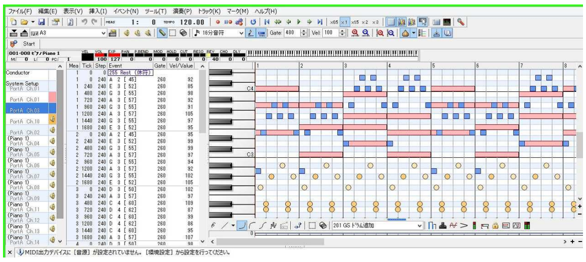

図 3.5　Domino

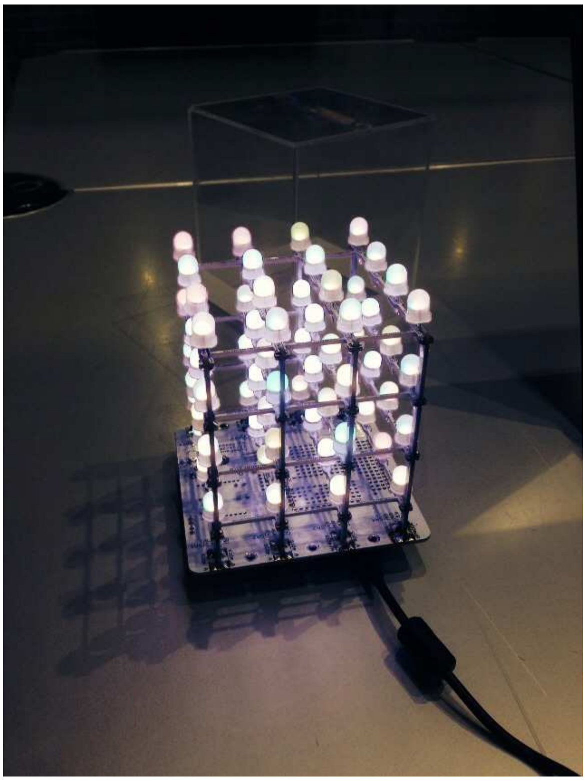

図 3.6　RainbowCube

PDF Compressor Pro

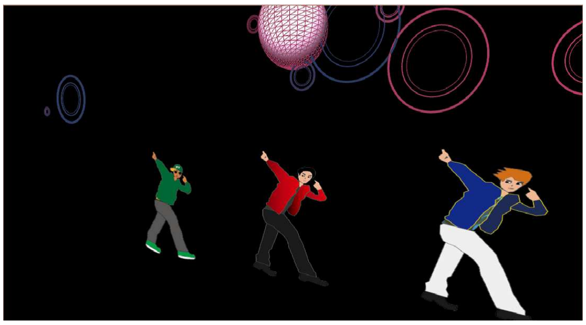

図 3.7　dancer

PDF Compressor Pro

第 4 章　課題解決のプロセスの詳細

4.1　前期

このプロジェクトは今年発足したため、当初はメンバー全体がプロジェクト活動内容を固めることができなかった。そのため前期の活動では、まずシンクロとは何か理解してからシンクロ同期を利用した。シンクロといえば最初に思い浮かぶのはシンクロナイズドスイミング等であろう。シンクロナイズドスイミングとは競泳のように、ある距離を泳ぐタイムを競うのではなく、水深 3m 以上のプール内で音楽に合わせて肉体を動かし、技の完成度、同調性、構成、芸術的な表現力などの得点で競うスポーツである。このスポーツの由来はシンクロ（同調性）から来ている。つまり、このプロジェクトでは同調性を詳しく理解する必要があった。

人と人の同調性には大雑把に 2 種類、意識して起こすシンクロ同期と意識しないで起こるシンクロ同期がある。意識して起こすシンクロ同期は前述したシンクロナイズドスイミングなど団体でのスポーツなどが挙げられる。一方意識しないで起こるシンクロ同期は普段周りからシンクロニティや意味のある偶然の一致などと呼ばれるものである。これは主に動きが一致するというよりも意識が一致するという言い方が正しく、より心理学的な現象である。今回私たちは前者の意識して起こすシンクロ同期をテーマとして取り上げ、同期した時の音楽・映像表現などの新たな表現方法を提案した。

さらに、このプロジェクト全体の目的である被験者が意識・無意識に問わず同期することにより得られる心理的効果（親和性や魅了性の向上など）を数多く見つけられるよう前期の活動に取り組んだ。

（※文責: 小澤繁輝）

4.1.1　映像班

映像班の概要

- コンセプト

  前期の活動では前述した通り私たちはまず最初にシンクロとは何か理解し、意識して起こすシンクロ同期について取り上げた。シンクロ同期について、具体的にはグループの活動方針として不特定多数の人物に同じような動きを使役しシンクロ同期を試みた。さらに、このシンクロ同期を第三者が見てもわかるような表現方法をいくつか考察した。その中で私たち映像班では、どのような方法でシンクロ現象を取得をするか、また聴衆に向けての新たな表現の方法を考察した。

- 活動内容

  映像班の前期の活動内容として人と人をシンクロ同期させるようなシステムを開発した。プロトタイプで私たちが利用した技術は主に Processing のカラートラッキングとノートパソコンの内蔵カメラである。映像班と同期判定班と音楽班人間の技術を合わせることにより、人の動きを使役し複数の被験者が同じ動きをするように試みた。映像班の役目は主に人の動きをカメラで取得しパソコン画面に映すこと、そしてそのカメラに映る人物の座標と加

PDF Compressor Pro

速度を検出し、同期判定班が使用している MAX というソフトウェアに検出したデータを送ること。

実験の際、被験者には 1 人につき 1 枚の色紙を右手に持ってもらい、計 3 人同時に実験してもらった。被験者は色紙を持った右手をパソコン上に映る自分と横にいる他の被験者を見ながら腕を上下に振ってもらった。

まず映像班同時にマーキングした色紙により、パソコンのカメラからそれぞれの被験者から色紙の位置、加速度を取得することができる仕組みである。この取得したデータを MAX に送る。

次に同期判定班が使用している MAX により映像班が取得した色紙の初期位置（色紙が最初に動き始める座標）と腕を振った時の加速度から、腕を振るテンポを読み取り被験者と被験者でテンポがあっている（シンクロしている）かを調べることができる。

最後に被験者はパソコン画面に映る自身や横にいる他の被験者を見ながら、腕を上下に振っているため被験者自身がシンクロ同期しているか分からない。この問題を解決するため音楽班による表現を利用した。音楽班で midi 音楽を自作し、MAX でテンポが一致した時（被験者同士がシンクロ同期した時）にその音楽を流すことで、被験者同士がシンクロ同期したと理解できる工夫をした。もう一つ工夫として被験者一人に対し曲のパートを分けることで被験者自身が曲のテンポと同調しているかも理解できるようにした。

前期の映像班の反省課題としては、ノートパソコンのカメラを利用したことにより被験者全員がカメラに収まらなかったり、パソコンの処理能力に耐えられず被験者全員の動きを取得することができなかったりと、活動環境として適切ではなかった。また視覚的表現について、前期ではまずノートパソコンの内蔵カメラで複数の物体を読み取り、その後 Processing のカラートラッキングを用いることで、ノートディスプレイからの視覚的同期認識を可能としたが、読み取ったデータからの映像効果は特に作成しなかったため、前期の成果では実際に上記の技術を用いてデモンストレーションを行ってもらい、ユーザーにノートディスプレイを見せるみすぼらしい形となり、聴衆から見てつまらないというアンケート結果から頂いた。

よって映像班全体の課題として被験者のデータを取得するための環境を十分に用意すること、また被験者が利用していて楽しいと思う表現をさらに追求し、聴衆が見ていて面白いと思わせられるような表現方法を実現し、被験者と聴衆全体のモチベーションを高めたいと思う。

（※文責: 小澤繁輝）

右手の動きの検出

- 概要

我々は動きを検出する方法としてカラートラッキングを用いた。カラートラッキングとは、特定の色を検出することで、物体を追跡するという方法である。追跡する色が、周囲の色とはっきり区別のつく色であれば複数の色を使うことで個別の認識が安易に可能である。

今回、単一の色で塗りつぶされた紙を数枚用意し、これを被験者それぞれに 1 枚ずつ持ってもらい、それを振ることで動きを検出した。画像の読み込みは PC に接続されたカメラで行った。被験者の服装は、色が派手でないものとし、背景がカラフルでない場所を選んで実験を行うものとした。また、逆光など、色がはっきりと確認できない環境で利用しないこと

PDF Compressor Pro

を想定した。

- アルゴリズム

我々が採用したカラートラッキングのアルゴリズムは、映像班で試行錯誤して考案したもので、我々独自の技術である。そのアルゴリズムは次のようなものになった。

カラートラッキングを行う際に必要となるデータは、カラー画像、追跡する物体の色、色の検出範囲（閾値）の3つである。まず、カラー画像の全ピクセルに1ピクセルずつアクセスし、追跡する物体の色との差を求めた。この差が閾値以内であった場合、このピクセルを物体の一部であると見なし、その座標を配列に記憶していった。画像全体に対してこの処理が終わった後、配列に保存された座標群から平均座標を求め、この実数値を物体の中心点とした。我々はこのようにしてカラートラッキングを行った。

次に具体的なプログラムの実装と実行プロセスを解説する。まず、あらかじめ閾値を設定した。これは定数として宣言して、プログラム実行中は変更しないものとした。この値は計算で求めるのが不可能であったため、試行を繰り返すことで適切な閾値を導いた。アプリケーションを実行した際、被験者は最初に持っている紙を、画面上ではっきりと色がわかるようにカメラの前に提示する。読み込まれたカメラ画像上でそれぞれの紙の中心をクリックすることで、追跡する色をRGB（Red, Green, Blueを組み合わせて1つの色を作り出すためのデータ）で取得した。その後、30フレーム毎秒でカラートラッキングを実行した。取得された中心座標と2秒分の軌跡をカメラ画像上に表示し、検出の状況を確認できるようにした。また、誤検知を防ぐために、取得されたピクセル数が少ない場合に、画面上に物体は存在しないと判定するようにした。カラートラッキングは図4.1のようにして行われる。

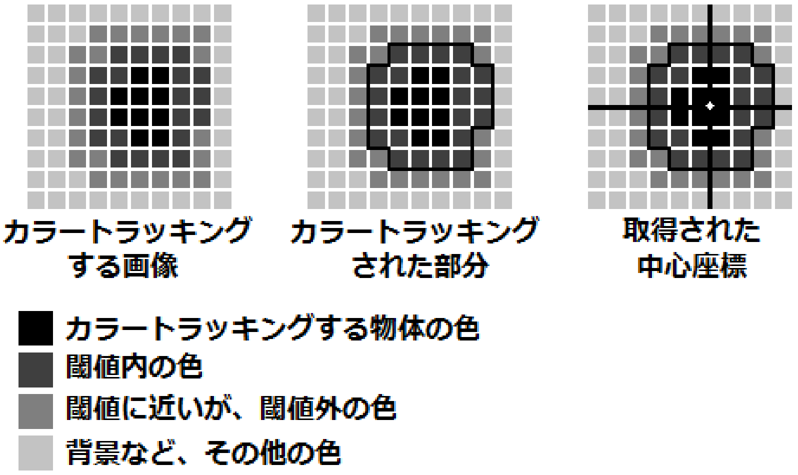

図 4.1　カラートラッキング

（※文責: 齋藤建）

判定方法

私たちは、物体を上下に振りその動きを追跡することを想定し、マーカーの上下の動きを判定できるようにしようと考えた。そこで、上下の動きを追跡するための方法としてカラートラッキングを用いた。

まず初めに、上下の動きを判定するにあたり、マーカーが動いた速度を求める必要があると考え

PDF Compressor Pro

た。マーカーの速度は、現在の中心座標と 1 つ前の中心座標の差を求めることで計算することができると考えた。そこで、現在のマーカーがある場所のの x 座標, y 座標を一時的に記録した。そして、マーカーが動いた後の x 座標, y 座標と、先ほど記録した x 座標, y 座標との差をそれぞれ求め、それを速度として記録していった。これを繰り返すことで、マーカーの速度を求めることが可能になった。

次に、先ほど保存していった速度のうち、縦方向の速度をすべて検査していき、その絶対値が 3 以下である成分が一定数以上存在していた場合、マーカーが一定時間動いていないと判断した。そのとき、保存していたすべての速度を消去した。マーカーが動いた場合は、上下の動きの判定を行う。

上下の動きを判定するとき、ある一定の速度を超えていなければ振ったと判定されないために、基準値となる定数を設定した。まず、縦方向の速度の絶対値と基準値の絶対値を比較し、縦方向の速度の方が小さい場合は、処理をスキップした。これにより、微小な動きでも判定してしまうことを防ぐことができる。

今の縦方向の速度が正のとき、次の速度が負の基準値よりも小さい場合、上から下にターンしたと判定する。そうでない場合、今の縦方向の速度と、それよりも後の全ての速度を比較する。後の速度の方が大きい間は上方向に動き続けていると判断し処理を続行し、後の速度ベクトルが負の基準値を下回った時点で、上から下にターンしたと判定する。

今の縦方向の速度が負のとき、次の速度が基準値よりも大きい場合、下から上にターンしたと判定する。そうでない場合、今の縦方向の速度と、それよりも後の全ての速度を比較する。後の速度の方が小さい間は下方向に動き続けていると判断し処理を続行し、後の速度ベクトルが基準値を上回った時、下から上にターンしたと判定する。

以上のようなアルゴリズムを用いることにより、マーカーの上下の動きを判定することができるようになった。

（※文責: 立石純基）

4.1.2 同期班

同期班の概要

同期判定班の前期の活動では、映像班から 3 つのタイミング情報を受け取り、それがすべて同期しているかを判定する機構を作成した。プログラム作成にあたって、Max 7 を使用した。また、映像班から情報を受け取るにあたって OSC を使用した。

5 月、6 月では映像班から送られてくる 3 つのタイミング情報のうち 2 つずつ同期しているか判定を行うような機構を作成した。そして、3 つのタイミング情報を A,B,C、音楽班が作った音楽のパートを A',B',C' とすると、A と B が同期していれば A'、B と C が同期していれば B'、C と A が同期していれば C' が再生されるような機構を作成した。

しかし、様々なエラーが出たので 7 月頃から 3 つのタイミング情報が拍ごとに収まっているかどうかを判定して、二小節ごとに点数を情報として送る方法に切り替えた。これにあたり、他のグループのメンバーからの補助もあり、Max 内で javascript を使用した。最終的に、3 つの信号が一定時間内に収まっているかどうかを拍ごとに判定して、二小節ごとに点数を情報として送るプログラムを作成することに成功した。

（※文責: 坂田聖名）

PDF Compressor Pro

Max7 について

Max7 とは Cycling'74 が開発している音楽とマルチメディア向けのグラフィカルな統合開発環境 (ビジュアルプログラミング言語) のことである。オブジェクトと呼ばれる箱のようなものがそれぞれ機能をもっており、そのオブジェクト同士をパッチコードと呼ばれる線で繋いでプログラムを作成していくものである。一つのウインドウに表示されたマックスのプログラムをパッチャー、またはパッチと呼ぶ。必要に応じて Max 内で javascript としてプログラムを書くことも可能である。処理が速いため、細かいタイミングで処理を行うシステムに対応することができる。これにより、様々な音楽アーティストも Max 7 を使用している。

（※文責: 坂田聖名）

プログラムの説明

メインパッチ　前期の活動では、映像班からタイミング情報を受け取り、そのタイミング情報が同期しているかを判定して判定結果を音楽班に送信するプログラムを作成した。

まず、OSC-route オブジェクト等を用いて、3 人分のタイミング情報を「turn0」、「turn1」、「turn2」として受け取る。select オブジェクトを用いてそのタイミング情報のうち最初の 4 回はメインの機構を動かさず、クリック音が鳴るようにした。5 回目の入力で、start というメッセージオブジェクトを使用して 3 パート全ての音楽を再生、metro オブジェクトや counter オブジェクトを使用してシステム内の時間をカウントし始めて、同期判定を開始する。if オブジェクトとシステム時間を用いて、それぞれのタイミング情報をシステムが動き出してから何ミリ秒経過したかを ave.js という javascript を使用しているオブジェクトに送信する。ave.js は 3 人のタイミング情報がほぼ同じタイミングで入力されているかを判定するものである。このオブジェクト内で 2 つの情報を出力する。 1 つは 3 人が 200 ミリ秒以内に収まって 3 人とも入力していれば 1 を送り、3 人揃っていなかったりタイミングが極端にずれていれば 0 を送る出力。もう 1 つは 3 人が入力したときの平均システム時間の出力である。2 つ目の出力は、1 つ前の出力と減法を用いて前の拍の入力から、現在の拍の入力の時間を算出し、tempo.js という javascript オブジェクトを用いて 1 小節分を平均して算出して BPM に変換して出力できるようにしたが、最終的なシステムの操作性の都合により使用しなかった。ave.js の 1 つ目の出力は sum.js という 2 小節分の点数を計算する javascript オブジェクトに送られて、 2 小節分まで加法で合計されてそれがその時点での点数として出力される。この点数をサブパッチの judge という名前の p オブジェクトに送信する。この中で 2 小節分の音楽が流れた時点で評価されて、その点数によって音楽班の 3 つあるパートのうちどこに再生の合図を送るかを決定する。もし点数が 0 ～ 2 の場合はベース音のみが再生される。 3 ～5 の場合はベース音とコード音が再生される。6 以上の場合はベース音、コード音、メロディ音全てが再生されるように指示を送る。2 小節が終わるタイミングは音楽班の midi ファイルが 2 小節ごとに分かれているので、seq オブジェクトから検出することが可能である。

5 秒間なにも操作しないと、図の左やや下にある delay オブジェクトが作動して、最初の 4 回の入力を数える counter オブジェクト、システム時間の counter オブジェクト、サブパッチ judge 内の counter オブジェクトの値を初期化する。

（※文責: 坂田聖名）

PDF Compressor Pro

ave.js　このオブジェクトは 3 つのタイミング情報が同期しているかどうかを判定する。3 つのタイミング情報をシステムが動き出してから何ミリ秒経過したかという数字として受け取る。入力は 3 つでそれぞれの時間を受け取り、出力は 2 つである。

このタイミング情報を ave.js 内で a, b, c として受け取る。まず if 文を用いて、a, b, c の内容を小さい順に並び替え、一番小さい値を min、一番大きい値を max に代入する。そして、min と max の差が 200 ミリ秒以下であれば、1 つ目の出力に 1 を送信して、3 つの時間の平均を 2 つ目の出力に送信する。もし、min と max の差が 200 ミリ秒を上回った場合は 1 つ目の出力に 0 を送信して、2 つ目の出力には 3 つの時間の平均を送信する。入力が 3 つ揃わなかった場合は 1 つ目の出力に 0 を送信して、2 つ目の出力には何も送信しない。

（※文責: 坂田聖名）

sum.js　このオブジェクトは ave.js の 1 つ目の出力から送られてくる 1 か 0 の数字を 2 小節分足していって、点数として出力する。入力は 1 つで、出力も 1 つである。入力を受け取るたびにその入力の値を受け取り、sum という変数に加算していく。sum の値は 1 拍ごとに出力されて、2 小節ごとに初期化される。ここでの点数は基本的に最高 8 点ということになる。

（※文責: 坂田聖名）

サブパッチ (judge)　このサブパッチでは sum.js から送られてきた点数をもとに 3 つある音楽班の作ったパートのうちどのパートを再生するかを決定する。この judge サブパッチは 2 つあり、1 つはメロディパート、もう一つはコードパートの判定を担当している。入力は 3 つで、1 つ目は前の 2 小節が終わったタイミング情報を受け取る。2 つ目は点数の情報を受け取る。3 つ目はパッチ内のカウンターの値を初期化するためのものである。出力は 4 つで、曲が 8 小節分ある中で、それぞれ最初の 2 小節、その次の 2 小節、またその次の 2 小節、最後の 2 小節の判定を担当している。

if 文を用いて、2 小節の切れ目のタイミングで点数が何点になっているかを確認する。0〜2 点の場合、どちらの judge サブパッチも反応せず、ベースパートのみが流れている状態になる。3〜5 点の場合、コードパートの judge サブパッチが反応して、ベースパートとコードパートが流れている状態になる。6 点以上の場合はベースパート、コードパート、メロディパートすべてが流れている状態になる。そして、1 つ目の入力が来た時に judge 内の counter オブジェクトを用いて、前 8 小節のうち現在何小節目を再生しているかをカウントする。この部分の数字を用いて、パートを再生する合図を適切な小説に送信する。

（※文責: 坂田聖名）

4.1.3　音楽班

出力する音楽の提案

音楽班では同期班が作成したプログラムから受け取った信号によって音楽の出力を行うシステムを構築した。その際、出力するための音楽も作製した。受け取った信号からどの程度同期がなされているかを直感的に分かるような曲を MIDI ファイルで構築しようと考えた。その結果、今回は”パッヘルベルのカノン”を選曲した。カノンを選んだ理由としては、多くの人たちに認知されている曲の一つであり、また輪唱であるため複数の異なるパートが同時に流れていたとしても同時に流

PDF Compressor Pro

れているのであれば、不協和音になることなくハーモニーのある曲が流れると考えたからである。また作曲経験のある人があまり多くはなく、MIDI ファイルで曲を作る際に旋律やコード進行を学習する上で簡単な曲の方がいいと考えたからである。したがって、今回のプロトコルの出力する音楽はカノンを提案した。

（※文責: 伊藤知幸）

MIDI ファイル作製

まず初めにカノンを MIDI ファイルで作成することにした。MIDI ファイルを作成するために MIDI 作製フリーソフトである”sakura[4]”を用いた。このフリーソフトは、テキストで音階を書いて作るタイプのため音階が分かっていればテキストに音階を書き込むだけで容易に曲を作ることが可能であった。したがって、今回はカノンのため旋律やコード進行が明確であり、またその一部分を使ってループさせることを考えた。そのため、このソフトで作る方が容易であると考えた。

次に sakura で出力する曲を作る時、どのくらい同期しているかを滑らかに表現するにはどうしたらいいかを考えた。どのくらい同期しているかを分かりやすくするために複数の入力の組み合わせにそれぞれベース、コード、メロディにパートを割り当て、その組み合わせが同期しているときに同期したパートが流れるように作ることで同期具合を滑らかに表現できるを考えた。また、それによって徐々に音楽が作られていくと考え、先行研究ではされていなかった音の変化を作ることで先行研究から発展させたシステムを作成した。

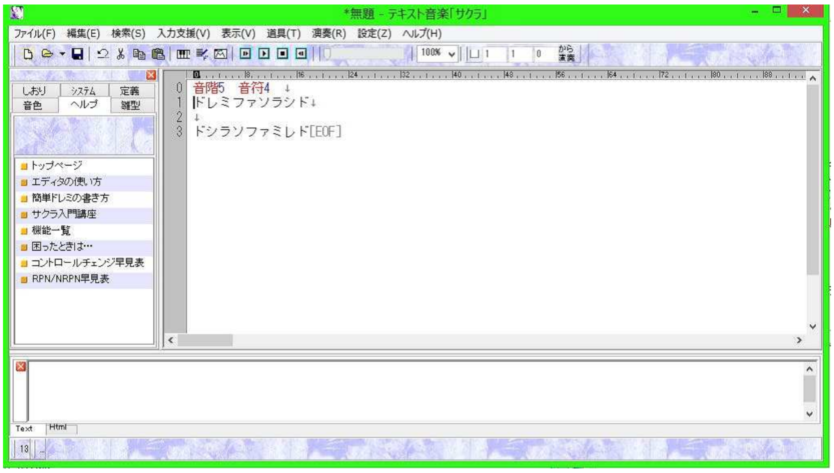

図 4.2　MIDI 作成ソフトの sakura

（※文責: 伊藤知幸）

MIDI ファイルの出力

作製した MIDI ファイルを出力するため Max の MIDI 用のパッチを作成し、受け取った信号が変化する事に合わせて MIDI の出力を割り当てた。受け取る信号は 3 種類あり、同期具合の点数が 0〜2 点の場合に送られてくる信号を受け取るとベースのみが流れ、同期具合の点数が 3〜5 点の場合に送られてくる信号を受け取るとベースとコードが流れ、同期具合の点数が 6 点以上の場合に送られてくる信号を受け取るとベース、コード、メロディが流れるようなシステムを作成した。そして作製した MIDI ファイルは図 4.1 の E で囲まれた部分のように、それぞれのパートを 2 小節で

PDF Compressor Pro

分けて作成した。このことによって曲の完成度の変化が細かく変化しているため、より分かりやすくシンクロがなされていると判断した。

（※文責: 伊藤知幸）

4.2 後期

4.2.1 Kinect 班

Kinect 班の概要

前期に映像班で作成したプログラムでは、判定の精度や遅延などによって、あまりシンクロが感じられないことが分かり、後期ではこの Kinect 班で、よりシンクロを体感できるアプリケーションを開発することになった。

Kinect 班で制作するアプリケーションには、大きく分けると、以下の 2 つの機能が要求された。また、これに基づいて我々は設計を行った。

- 1. 複数人のある動作を前期のプログラムよりも正確に判定
- 2. 判定されたデータを外部（Max の Patch）に出力

Kinect 班では、動きを取得する方法として、Kinect v2 を用いた。Kinect（キネクト）は、Microsoft 社から発売されたジェスチャー・音声認識によって操作ができるデバイスである。当初のプラットフォームは、家庭用ゲーム機 Xbox 360 のみであったが、のちに Windows 用の SDK や、USB 接続のアダプタなどが公式に開発された。Kinect v2 は、Xbox One と共に再設計された Kinect であり、仕様やコードの記述が大きく異なる。また、Kinect v2 を PC で動作させるには、表 4.1 のスペックが要求される。

表 4.1　Kinect v2 のシステム要件

OS	Windows 8, Windows 8.1
プロセッサ	2.66GHz 以上のデュアルコアプロセッサ
RAM	2.0GB
GPU	DirectX 11
USB	USB3.0 端子

次に、取得する部位を右手に限定した。これは、多くの人の利き手が右手であることから判断した。プログラミング言語は C ♯で、統合開発環境である、Visual Studio Express 2013 for Desktop を使い、WPF アプリケーションで開発した。また、Max への出力は、Open Sound Control(以下 OSC）の技術を用いた。

（※文責: 齋藤建）

実行環境の設定

Kinect センサーについて調べていくことで、制約などが分かり、またプロジェクト全体で要求されるアプリケーションに基づいて、実行環境を設定した。取得する人数は 3 人である。また、取得する部位はそれぞれの右手である。この時、利用者は右手の動きをはっきり取得するために、

PDF Compressor Pro

Kinect から約 1.5m 離れた場所に立つ。利用者は、上下に振り幅約 40cm で右手を振る。(図 4.3)

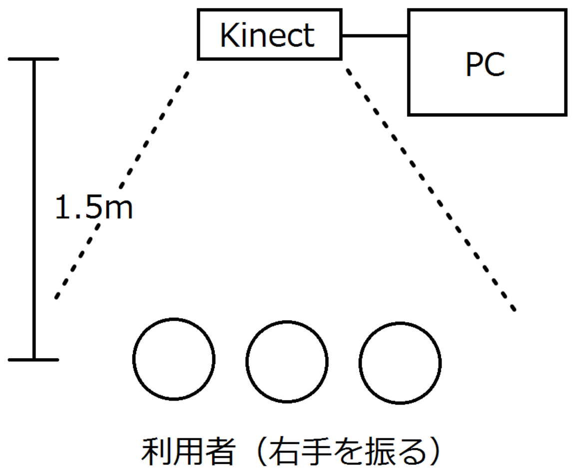

図 4.3　想定される配置

我々は Kinect v2 の深度センサーを利用し、人体の骨格を検出し、右手の座標を取得した。プログラミング言語は C ♯で、統合開発環境である、Visual Studio Express 2013 for Desktop を使い、WPF アプリケーションで開発した。実験中に画面上で取得の状況を確認できるように、Kinect のカメラの画像を表示し、その上に右手部分に円を表示した。

（※文責: 齋藤建）

OpenSoundControl

Kinect で得られたデータを Max へ出力ために、Open Sound Control[5] の技術を用いた。OSC とは、電子楽器やコンピュータなどの機器で、演奏データを共有するための通信プロトコルである。ネットワーク経由でリアルタイムにデータの共有が可能であり、電子楽器の演奏データを機器間でデジタル転送するための世界共通規格である「MIDI」の代替となるとして、近年注目を集めている。C ♯のライブラリ「Rug.Osc」[6] を使用した。また、OSC をより扱いやすくするために OpenSoundControl クラスを作成した。これによって、次のように記述して使用することができるようになった。

- OpenSoundControl osc = new OpenSoundControl(”IP アドレス”, ポート番号);
  使用する IP アドレスとポート番号を指定して宣言する。
- OpenSoundControl.Send(”OSC メッセージ”, ”送信する文字列”);
  OSC メッセージと文字列をセットで送信する。

具体的には、1 人目が振ったときは、osc.Send(”turn1”, ”1”); と記述し、受信側では/turn1 で”1”を受け取った際に、1 人目が振ったと判定する。また、プログラム実行後すぐに空のデータを OSC で送ることで、OSC が利用できることを確かめられるようにした。

（※文責: 齋藤建）

PDF Compressor Pro

計画

Kinect 班で話し合いをして、作業を明確にし、分担を行って作業した。その結果、表 4.2 のように作業を進めることができた。

基本的に、上下の動きの判定を立石が担当し、Kinect に関するプログラムの実装を齋藤が担当した。一方が行き詰った際は、2 人で解決することで柔軟に課題に取り組んだ。

Kinect を注文してから届くまでに時間がかかってしまったため、開発開始が遅れてしまったが、順調に開発することができていた。Kinect のカメラと、深度センサーの取得範囲が異なるため、どちらか一方に合わせる必要があった。このプログラムに関する説明がインターネット上にあまり無く、実装に多少時間を要した。

また、試験をし、不具合などを発見して修正することで安定したプログラムを作ることができた。

Kinect v2 は比較的新しい技術であるため、資料がまだ少ないが、これから資料が充実していき、発展していくと考えられる。本プログラムはその手助けになるだろう。

表 4.2　Kinect 班の開発工程

内容	担当者	完了日
Kinect・C ♯の技術習得	齋藤・立石	10/23
カラー画像の入力・表示	齋藤	10/28
深度センサーで右手の座標を取得	齋藤	10/28
OSC でデータを送信するメソッドを作る	齋藤	10/29
システム時間を取得	齋藤	10/29
複数人の右手の座標を取得	齋藤	10/30
右手の座標とシステム時間を配列に記録	齋藤	11/2
Processing で加速度を用いて上下の振った判定をする	立石	11/2
Kinect を使って、右手を上下に振った判定をするプログラムを実装	齋藤・立石	11/6
振った判定を Max に送る	齋藤・立石	11/6
試験	齋藤・立石	11/6
座標を取得する間隔を少し長くする	齋藤	11/6

開発工程を補足する。

- Kinect・C ♯の技術習得
  主にサンプルプログラムを入手したり、簡易的なアプリを作ることで Kinect を学んだ。
- カラー画像の入力・表示
  Kinect のカメラから読み込まれた画像をディスプレイに出力した。
- 深度センサーで右手の座標を取得
  Kinect の深度センサーを使って人体の骨格を検出し、そのうちの右手の位置を二次元座標として出力した。
- OSC でデータを送信するメソッドを作る
  OSC を簡単に扱えるようにするために、既存のライブラリを用いてクラスを作成した。
- システム時間を取得
  アプリケーション開始からの経過時間を取得するクラスを作成した。

PDF Compressor Pro

- 複数人の右手の座標を取得
  上記の右手の座標を取得するプログラムを、3 人にまで対応させた。
- 右手の座標とシステム時間を配列に記録
  右手の座標を配列に記録し、連続したデータとして扱えるようにした。
- Processing で加速度を用いて上下の振った判定をする
  Processing 上で上下の動きを判定するプログラムを作成した。
- Kinect で右手を上下に振った判定をするプログラムを実装
  立石が作成した Processing のプログラムを齋藤が Kinect 用に書き換えた。
- 振った判定を Max に送る
  Max が受信できるようなデータで送信し、Max で受信できるか確かめた。
- 試験
  動作チェック等を行った。
- 座標を取得する間隔を少し長くする
  試験した際に、座標を取得する間隔が短いことが分かったため、取得時間を少し長くした。

（※文責: 齋藤建）

前期に作成したプログラムとの比較

前期に作成した判定プログラムでは、カメラから取り込んだ映像の中でクリックした場所の色を取得することで、右手だけではなく他の様々な物体の座標を取得した。一方、今回作成したプログラムでは、人体の骨格を検出することにより右手の座標を取得した。Kinect を用いることにより、前期に作成したプログラムよりも正確に座標を取得することに成功した。しかし、骨格を検出するためには体全体を映す必要があるため、少し Kinect から離れる必要がある。

また、前期に作成したプログラムでは、取得した色の動きを追跡することにより、1 フレームごとの座標の変化から速度を求めることで、上下の動きを判定した。一方、今回作成したプログラムでは、Kinect で取得した右手の Y 座標と、C ♯のプログラムの中で取得した時間を用いて右手の動く速度を求めた。前期に作成したプログラムでは、1 フレームごとの速度を求めていたが、1 フレームごとの時間に誤差が生じる可能性があり、正確に速度が求められない可能性があった。しかし、今回は時間を用いて速度を求めているため、より正確に速度を求めることが可能になった。

（※文責: 齋藤建）

プログラムの構成

Kinect 班が実装したプログラムは図 4.4 のようになった。

私たちは、右手を振った判定を行うために、Time, Data, DynamicPoint の 3 つのクラスを実装した。

Time クラスでは、プログラムを実行してから現在までの時間を取得する。その方法として、名前空間 System.Diagnostics の Stopwatch クラスを利用し、プログラム実行時にストップウォッチをスタートすることとした。そして、取得したいタイミングで経過時間をミリ秒で取得した。Time クラスを初期化することでストップウォッチがスタートされるようにしており、Time クラスの中にある GetTime メソッドで現在の経過時間を取得できる。

PDF Compressor Pro

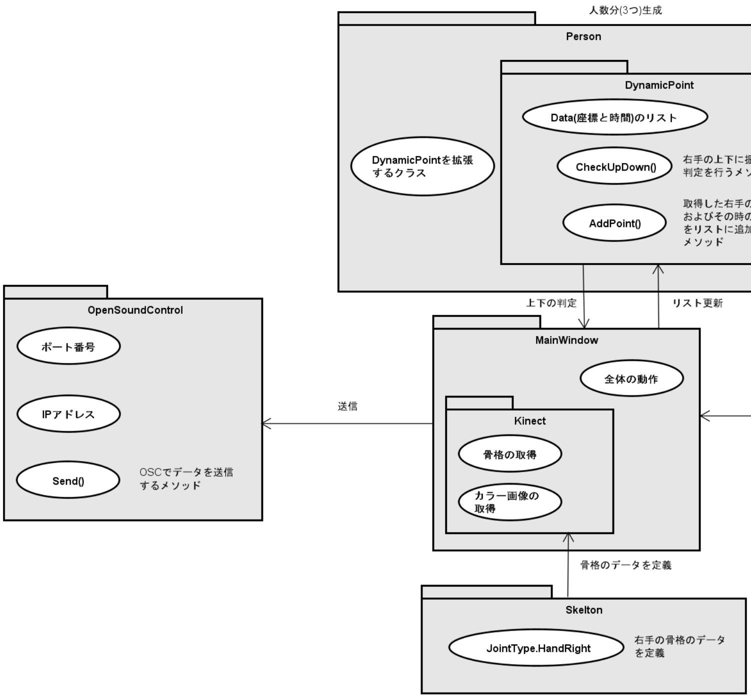

図 4.4　プログラムの構成

MainWindow クラスでは、Kinect を用いて右手の座標を取得した。

Data クラスでは、取得された右手の座標と取得した時間を一つにまとめた。

DynamicPoint クラスでは、実際に右手を振った判定を行う。AddPoint メソッドで取得した Data を List に格納し、それをもとに CheckUpDown メソッド内で判定を行った。

（※文責: 立石純基）

振った判定のアルゴリズム

私たちは、右手を振った際の縦方向の速度と高さと時間の関係は図 4.5 のようになると想定した。

上のグラフが速度と時間の関係、下のグラフが高さと時間の関係を表したものである。下向きへの速度を正、上向きへの速度を負とした。　右手を振り下ろし始めたとき、ある地点までは速度が正の方向に加速していく。その後、速度が減速していき、完全に振り下ろした時に速度が 0 になり、静止する。

反対に、振った右手を元に戻すときは、ある地点までは速度が負の方向に加速していき、その後減速し、元の位置に戻った時に速度が 0 になり、静止する。この予想をもとに、私たちは図 4.6 のように動作する CheckUpDown メソッドを実装した。

- 下がり待ち
  初期状態を表す。この状態で右手の速度が正になったとき、つまり右手を振り下ろし始めたとき、「下がり開始」状態に遷移する。
- 下がり開始
  この状態のとき右手が一定の速度を超えると、「下がり判定待ち」状態に遷移する。このと

PDF Compressor Pro

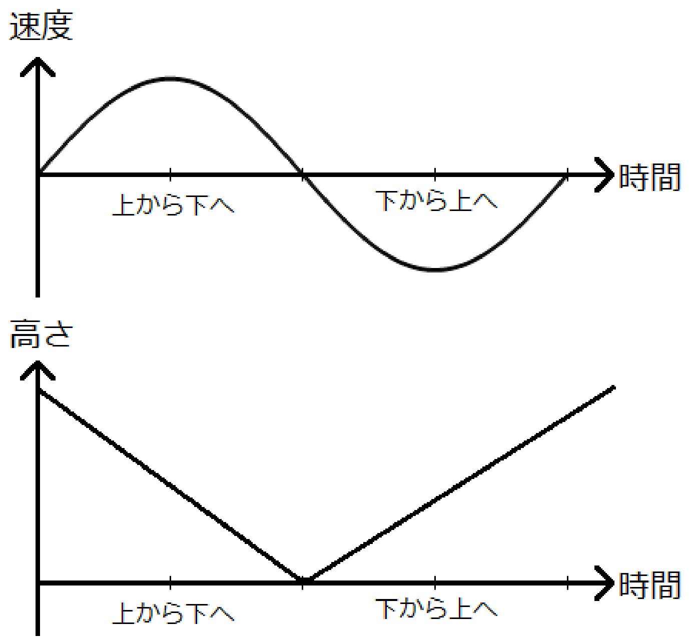

図 4.5　速度・高さ・時間の関係

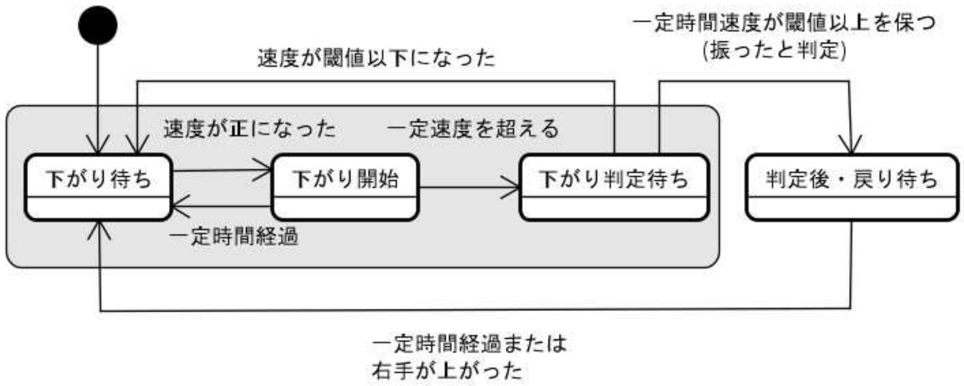

図 4.6　CheckUpDown の動作

き、現在の右手の y 座標を記録する。また、この状態のまま 1 秒が経過すると、再び「下がり待ち」状態に戻る。これにより、微小な振動で判定が行われることを防ぐことができる。

- 下がり判定待ち
  この状態のときに、15 ミリ秒の間、速度が閾値以上であったときに、はじめて右手を振ったと判定される。また、15 ミリ秒の間に 1 度でも速度が閾値以下になった場合は、「下がり待ち」状態に戻る。これにより、一瞬だけ一定速度を超えたという場合にも振ったと判定されてしまうような誤判定を防ぐことができる。
- 判定後・戻り待ち
  判定後は、「戻り待ち」状態に遷移する。このとき、右手が「下がり判定待ち」状態に遷移した時の y 座標以下になったとき、もしくは速度が一定以下になったとき右手を上げた (戻した) と判定し、再び「下がり待ち」状態に遷移する。また、振ったと判定されてから 2 秒以上経過した場合も「下がり待ち」状態に遷移する。

PDF Compressor Pro

上記のように、ただ単に状態が遷移していくだけはなく、前の状態に戻るための条件を設定することにより、いくつかの誤判定を防ぎ、より正確な判定を行うことが可能になった。

（※文責: 立石純基）

Kinect 班のまとめ

今回作成したアプリケーションは、複数のクラスで構成されており、メインのクラス・時間を取得するクラス・OSC の接続と送信のクラス・上下の判定のクラスなどを実装した。このアプリケーションは、Kinect の深度センサーによって複数人の右手の座標を取得し、その動きを読み取って振ったかどうかの判定がされ、振ったと判定されたときに OSC で振った人に応じたデータが送信されるという流れで動作する。また、このアプリケーションは、単に Max にデータを送るだけでなく、OSC で受信が行えるあらゆるデバイスやアプリケーションに対して使うことができ、拡張性が高いことがわかった。

（※文責: 齋藤建）

4.2.2 同期班

同期判定班の概要

プログラム作成にあたって、私たちは Max 7 を使用した。前期の活動で作成したプログラムを改良していく形で作業を行った。まず前期のプログラムからあった最初の 4 回の入力では機構を動作させずクリック音を鳴らせるものを利用して、基本のテンポを変動できるようなものにした。次に 3 つの入力の平均を取り、その間隔をもとに曲中にテンポを変更することを試みた。しかし、11 月中旬から (vstVirtual Studio Technology) という音楽に対してエフェクターなどの役目を果たすものを使用し始めたことで曲中のテンポ変動は難しいものとなった。

その後、ave.js という 3 人が同期しているかを判定する javascript ファイルを編集して何人が同期したかを判定する機構を作成した。また sum.js という ave.js から送られてくる点数を加算していく javascript ファイルを編集して、ave.js で出した値を 2 小節分記録してその平均を出すという機構を作成した。またこの頃に音楽班から音楽を vst を通して再生して、変数に応じて音楽のボリュームを変更して音を出すか出さないかを動かす機構を受け取った。これにあたり、様々な部分からエラーが出たのでそれを処理する作業を行った。さらに、音楽班から発展バージョンの音楽を再生したいという要望があったのでこれを同期の評価が最高の状態のときにそれを再生する機構を作成した。また証明班、映像班の視覚的な表現をわかりやすいものにするために誰と誰が同期しているかを判定する機構を作成した。

最終的に、Kinect 班から動きのタイミング情報を受け取り、それが何人同期しているか、誰と誰が同期しているかを判定する機構を作成することができた。また、最初 4 回の入力で元のものから曲のテンポを変動させる機構を作成することに成功した。

（※文責: 坂田聖名）

プログラムの説明

sum.js　この javascript オブジェクト (図 4.7) では、ave.js から送られてくる 8 拍分の同期人数の平均を求めたものを出力する。入力と出力はどちらとも 1 つである。 8 拍分の入力を入れること

ができる配列を用意し、1拍ごとに送られてくる入力をその配列に入れていく。８拍分の入力が配列に入っていて、さらに１拍分の入力が来た場合は、１番古い入力の情報を配列から出し、新しく来た入力の情報を配列に入れていく。１回の入力があるごとに同期人数の平均を出し出力する。その平均の値が３の場合、cnt という変数に１を足していき、それ以外の値の場合は cnt を０にする。cnt の値が６４以上になった場合、re1 という変数の値を４にし、同期人数として出力する。

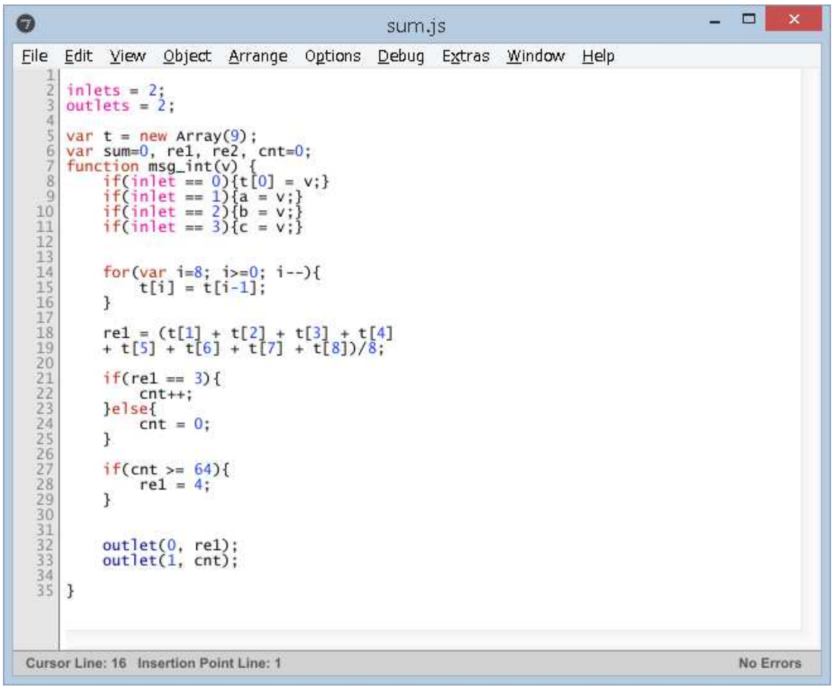

図 4.7　sum.js

（※文責: 伊藤知幸）

サブパッチ (judge)　このサブパッチ (図 4.8) では、sum.js から送られてきた同期している人数の値をもとに音楽班が作成した３つのパート、発展パートの再生パターンを決定する。入力は３つある。１つ目は前の１小節が終わったタイミング情報を受け取る。２つ目は同期人数の値を受け取る。３つ目はサブパッチ内にあるカウンターの値を初期化するためのものである。出力は４つあり、ドラムパート、コードパート、メロディパート、発展パートをそれぞれ度のパートを再生させるかを決定するためのものである。if 文を用いて、１小節の切れ目のタイミングで同期人数を確認する。同期人数が３から１、１から３のように一気に上がったり下がったりしないように１小節前の同期人数の値の情報をサブパッチ内に記憶させ、その情報とリアルタイムで送られてきた同期人数の値を比較させ、同期人数が３から１ではなく、３から２、２から１になり、１から３ではなく、１から２、２から３になるように１段階ずつずらすようにする。

同期人数によっての曲の再生パターンは、同期人数が０または１の場合、出力１に１の値、出力２、３、４に０の値を送りドラムパートだけを再生させる状態になる。同期人数が２の場合、出力１、２に１の値、出力３、４に０の値を送りドラムパートとメロディパートを再生させる状態になる。同期人数が３の場合、出力１、２、３に１の値、４の出力に０の値を送りドラムパート、メロディパート、コードパートを再生させる状態になる。同期人数がしばらく続き同期人数の値が４になった場合、出力１、２、３に０の値、出力４に１の値を送り再生されていたドラムパート、メロ

PDF Compressor Pro

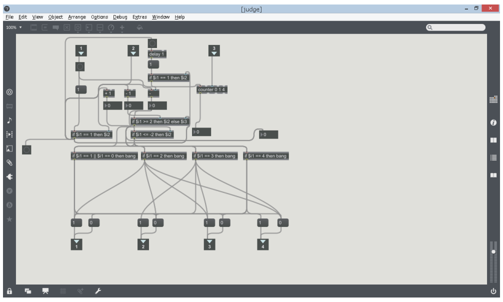

図 4.8　サブパッチ (judge)

ディーパート、コードパートの 3 つのパートが消え、発展パートを再生させる状態になる。

（※文責: 伊藤知幸）

メインパッチ　まず、OSC-route オブジェクト等を用いて、3 人分のタイミング情報を「turn1」、「turn2」、「turn3」として受け取る。select オブジェクトを用いてその情報のうち最初の 4 回はメインの機構を動かさず、クリック音が鳴るようにした。この際、サブパッチの time と、ctempo.js という javascript オブジェクトで最初の 4 回の入力間の平均を取り、それが元の音楽と比べて何倍になっているかを計算する。元の音楽は BPM120 であるので、1 拍は 500 ミリ秒である。これを元に除算で元の BPM に対する倍率を求め 1024 に掛ける。それを start1024 というメッセージオブジェクトに送信する。これで、再生される音楽を求めた倍率のテンポで再生することが可能である。

5 回目の入力で、start というメッセージオブジェクトを使用してドラムパートのみの音楽を再生、metro オブジェクトや counter オブジェクトを使用してシステム内の時間をカウントし始めて、同期判定を開始する。サブパッチの input を用いてシステムが起動してから何ミリ秒経過したかを ave.js という javascript オブジェクトに送信する。このオブジェクト内には 3 つのタイミング情報の他に、その時のテンポをミリ秒に変換し、2/5 を掛けた数字と、その時のシステム時間を送信している。出力について、1 つ目は 3 つのタイミング情報の平均を出力する。これは今後音楽を再生中に曲のテンポを滑らかに推移させるために使用しようと考えたが使用しなかった。2 つ目はその拍で何人が同期していたかを判定してその人数に応じて、0 人なら 0、1 人なら 1、2 人なら 2、3 人なら 3 を出力する。3 つ目は誰と誰が同期しているかを数値として出力する。これは turn1,turn2,turn3 を a,b,c として受け取り、a,b,c が全て同期できていたら 7、a,c が同期していたら 6、b,c が同期していたら 5、a,b が同期していたら 4、c のみが入力していたら 3、b のみが入力していたら 2、a のみが入力していたら 1、誰も入力していなければ 0 を出力する。3 つ目の出力はこのままデザイン班、証明班に送信する。2 つ目の出力は sum.js という javascript オブジェクトに送信する。

（※文責: 坂田聖名）

サブパッチ (time)　このサブパッチはシステムが起動する前の 4 回の入力間の時間をミリ秒として出力するものである。入力は 2 つで 1 つは入力のタイミングが送信されてきて、もう一つは time 内の counter を初期化するための入力である。このサブパッチ内では、メインパッチで使っているシステム時間を測るものと同じ仕組みでタイムを計り、if オブジェクトを用いて、入力間の

PDF Compressor Pro

時間を出力する。

（※文責: 坂田聖名）

ctempo.js　この javascript オブジェクトではサブパッチの time から送信された時間の情報を平均して BPM に変換してから出力する。入力は 1 つで、出力も 1 つである。入力で入ってくる値は array という配列変数に代入していき 4 つの平均を算出する。その値で 60000 を割り BPM に変換し、その値を出力する。

（※文責: 坂田聖名）

サブパッチ (input)　前期の制作で作った入力の部分と同じものであるが、後期の作成時、見やすさなどの都合でサブパッチとしてまとめた。

（※文責: 坂田聖名）

ave.js　この javascript オブジェクトではタイミング情報を受け取り、その時間の平均、何人が同期したか、誰と誰が同期したかを出力する。入力は 5 つで、3 つはそれぞれ 3 人分のタイミング情報を受け取り、1 つはその時のテンポをミリ秒に変換して 2/5 を掛けた値を受け取り、もう 1 つはその時のシステム時間を受け取る。出力は 3 つで、1 つはタイミング情報の平均、1 つは何人が同期したかを 0 人なら 0、1 人なら 1、2 人なら 2、3 人なら 3 を出力する。最後は誰と誰が同期しているかを数値として出力する。

　これは turn1,turn2,turn3 を a,b,c として受け取り、a,b,c が全て同期できていたら 7、a,c が同期していたら 6、b,c が同期していたら 5、a,b が同期していたら 4、c のみが入力していたら 3、b のみが入力していたら 2、a のみが入力していたら 1、誰も入力していなければ 0 を出力する。まず、a,b,c のうちどの値が実際に入力されたかを if 文で場合分けする。このとき、システム時間が 0 から、その時のテンポをミリ秒に変換して 2/5 を掛けた値までの場合は入力されたと判定されない設定にした。これは同期判定を表迫に近い部分でのみ行うためである。その後その時入力された値の平均を ave という変数に代入する。また、何人が同期したかの情報を re という変数に代入する。さらに誰と誰が同期したかの情報を hu という変数に代入する。

　そしてこの ave,re,hu の 3 つをその時のテンポをミリ秒に変換して 2/5 を掛けた値分時間を空けてから出力する。こうすることで、オブジェクト内で判定は何度か行われるが、出力は 1 拍で 1 回になる。

（※文責: 坂田聖名）

4.2.3　音楽班

音楽班の概要

　音楽班の前期の活動ではバッハルベルノカノンをもとに 2 小節ごとにパートごとそれぞれ分けた MIDI ファイルを作成し、それを同期判定の判定によって出力される音楽のパターンを変更できるようしていた。同期判定班から送られてきた同期具合によって再生させるかさせないかを変化させていただけであったため、メロディやコードに当たるパートが突然途切れるということが発生していた。また音楽データは 8 分主体のメロディーによって構成されているためにリズムが分かりやす

PDF Compressor Pro

いはずであるが、そのような結果を得られにくいという問題があった。

カナダの Varni, Marie, Bruce & Trainor (2010) の研究によると人は小周波の音と低周波の音では、低周波の音のが寄りリズムを認知しやすいという効果を持っていると示唆している。この研究では高音低音それぞれピアノで演奏を同時に行い、被験者にはそれを聞いてもらう。その演奏者が演奏する音のうち一つのビートを 50 ミリ秒というわずかな時間ずらして演奏した。その変化に被験者は気づけるかどうかということを確かめる実験である。その結果、低音の音を変化させた時の方がリズムの違いが分かる人が多かった。

この研究から低周波の音がよりリズムに影響していると言うことがわかる。前期で作成した MIDI ファイルを見てみると、伴奏やメロディの音階が C4 以上、ハ長調における起点である「ド」の音よりも 1 オクターブ以上高い音で構成されていた。そこで今回は低音の音を中心に構成されるリズムがより分かりやすい音楽を作成することで振るリズムによって音楽の速さが変化してもリズムが取りやすいような音楽を作成することを目標として活動を行った。

新たに音楽班の中でドラム、コード、メロディのパートで構成される楽曲を作り、それをそれぞれのパートで分けた MIDI ファイルを作成した。その時 MIDI ファイルの作成手段として、オリジナルで新たに作成したものと、既存の曲から手動で MIDI ファイルの書き起こす方法の 2 通りを用いた。そしてそれを Max で同期班の作成した同期具合を判定する機構と接続した。同期具合に合わせて送られた信号によってドラムのみ、ドラムとメロディ、ドラムとメロディとコードのパートがそれぞれ再生されるようにすることができた。また、その同期具合の判定が変化した時でも、急激に再生が切断されるのではなく、音量が少しずつ同期の判定に合わせて変化するようにできたために、自然に曲を変化することができた。また、3 人の同期が確認できているとき、その同期が持続していた場合に新たに更に変化を加えた MIDI ファイルを再生できるように機能を拡張した。

（※文責: 勝原望）

図 4.9　音楽出力がなされている画面

PDF Compressor Pro

音楽ファイルを作成する

この課題を達成するための MIDI ファイルを作成するにあたって MIDI 編集フリーソフトである「Domino」を用いて作成を行った。Domino とは TAKABOSOFT によって開発されたピアノロールをメインとした MIDI 専用の音楽編集ソフトの一つである。フリーソフトであり手軽に作曲環境を整えることができるのが特徴である。

前期では sakura を用いて作曲を行っていた。Domino と sakura の相違点としては Domino ではレイヤー管理ができるということが挙げられる。前期に sakura で作成していた時はそれぞれのパートで sakura のデータを起動してファイルを作成する必要があった。しかし Domino ではそれぞれのパートの曲のデータをレイヤーで分割して作成ができる。そのため一つの Domino のデータで全てのパートを作成することができた。また sakura では具体的な音を解析してそれを文字としてデータを置かなくてはならない。確かに前期ではカノンの曲で作成していたため音階について理解していたためにそれを用いて MIDI ファイルを作成することができていたが、今回全く新しいものでは作成が難しい。Domino は下の図のように音の高低とピアノの位置が対応していて、さらに時系列でデータを配置するようになっているため、楽譜に似たように音のデータを配置することができる。また最後にその音のデータから MIDI ファイルを作成するにあたっては、Max でも動作するように format 0 のシングルトラックデータで保存することに注意した。

後期の活動では Domino を用いたことでより効率的的に MIDI ファイルを作成することができた。次にそれぞれ作成したドラム、コード、メロディのパートを解説していく。

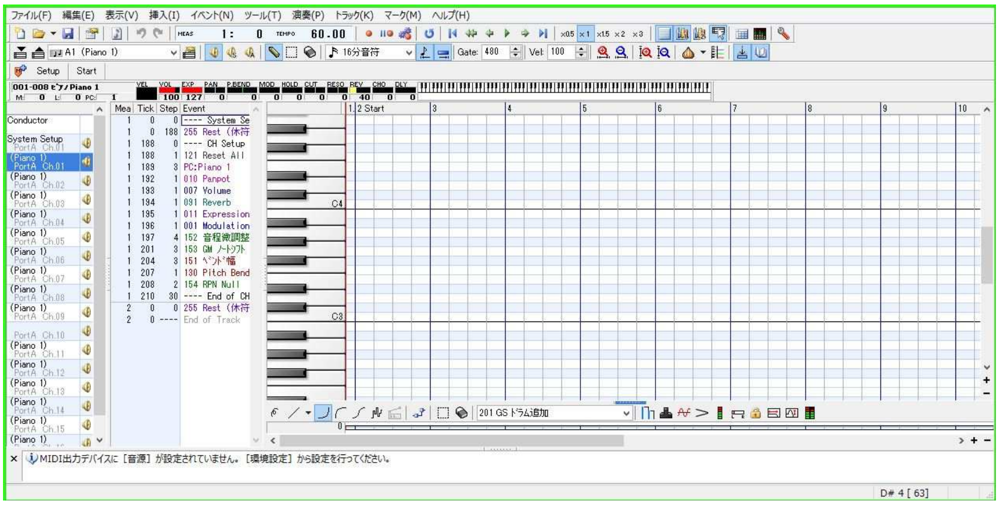

図 4.10　Domino の初期画面

（※文責: 勝原望）

新規で作成した音楽での各パートについて

まずはドラムから説明を行う。Domino では下図の様に、基本となるピアノの音源の他にもドラムの音源があり、専用のピアノロールがある。そしてその音源は鍵盤一つ一つに対応している。そのため、このようにドラムを配置することによってもドラムのデータとして MIDI ファイルを作成することができる。ドラムのパートについては一般的なハウスミュージックに使われているドラムを参考に使用した。ハウスミュージックではバスドラムを 1 小節に 4 つ等間隔で打ちこむように

PDF Compressor Pro

配置する。これを 4 つ打ちと言い、これによって同期の腕の降る速さが変化することによってリズムが変化したとしてもよりはっきりと変化が理解できるようにすることができる。そしてバスドラム単一だけでなく、スネアドラムやハイハットを追加することによってより 4 つ打ちのバスドラムを基調としたハウスミュージックの典型となるドラムパートとなった。

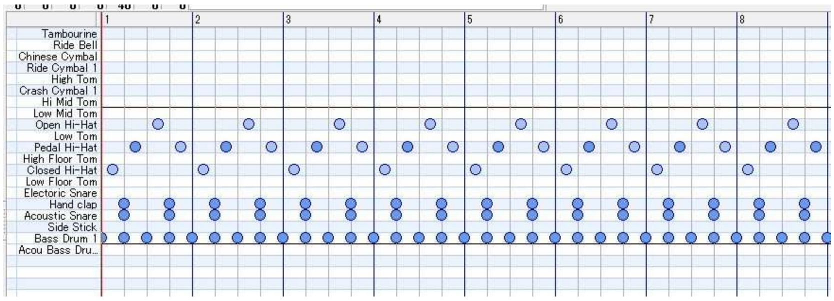

図 4.11　ドラムのデータ内容

次にメロディのパートに移る。この作成したファイルは 8 小節で 1 ループするようにしている。そのためメロディは何回も同じパートを繰り返しても違和感がが無いように作る必要があった。またそれであり低音が強く、リズムが分かりやすいということにも気を付ける必要があった。底で作成した MIDI ファイルの中身が下の図の通りである。音は全て 8 分の構成であるためにリズムが取りやすいと考えられ、また音の高さは C2 の音階を始点としているために前回よりも 1 オクターブ近く全体的に音が下がっている。また小節ごとに見ても同じような音の撮り方で、それが 4 小節で 1 セットになっているような繰り返し再生されていても違和感なく繰り返し再生することができる。このことからメロディパートではリズムが分かりやすい繰り返しても違和感の少ないパートが作成された。

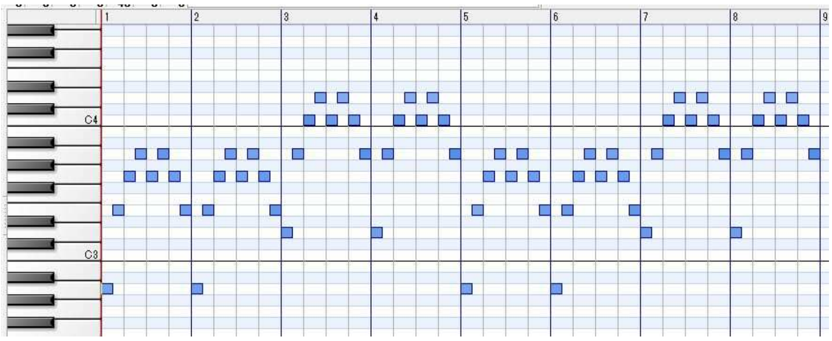

図 4.12　メロディのデータ内容

最後にコードである。コードは全て 1 小節分ずつの長さに統一し、また最初と最後で同一の音を配置することによって繰り返し再生していても違和感なく再生できる構成である。また、メロディは 4 小節で同一の音を繰り返す短いループであったが、コードではそのような短い繰り返しを使っていない。そのことで同一のメロディが再生されていても変化がみられるようになった。

　また、3 人が同期が持続していた時に新たに音が追加されるように拡張した MIDI ファイルも作成した。これは 3 人とも同期が確認できているため、メロディ、コード、ドラムが全て再生されて

PDF Compressor Pro

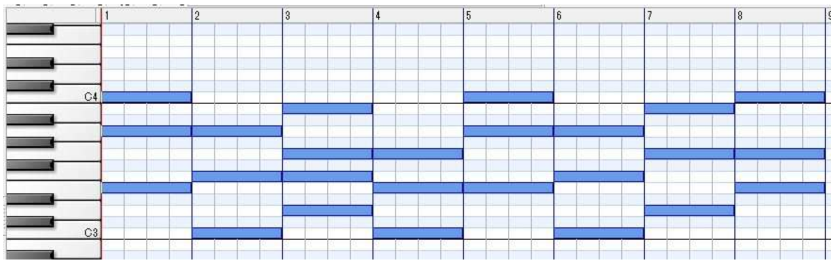

図 4.13　コードのデータ内容

いる状態から遷移するため、その MIDI ファイルを作成する時は、ドラム、メロディとコードの 2 つの MIDI ファイルで分割して作成した。下図はドラムのデータの内容である。上で記述したドラムと比較すると、確かに音の数は増えているものの、基本となるバスドラムは 4 打ちのことや、ハイハットやスネアも最初の場合と同様の配置に音を追加しているだけである。

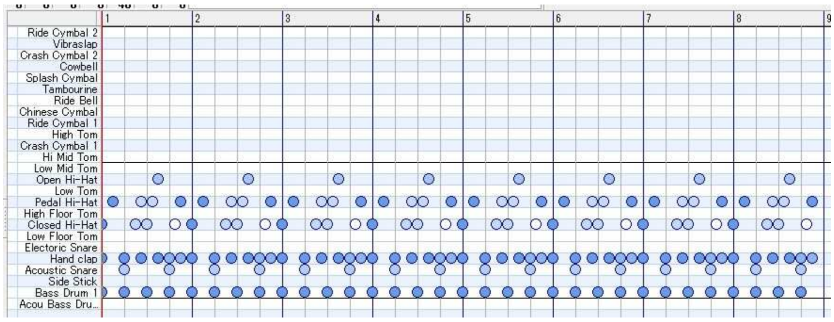

図 4.14　ドラムのデータ内容

最後にメロディとコードの内容では、メロディには低音の音をさらに一つ追加している。メロディの追加音やコードに関しては 8 分ではない音が使われるようになっている。しかし基本となるメロディには手を入れずに全く同じ状態にしている。そのため、メロディの主旋律は 8 分のみの構成でありリズムは変わらず取りやすいと考えた。

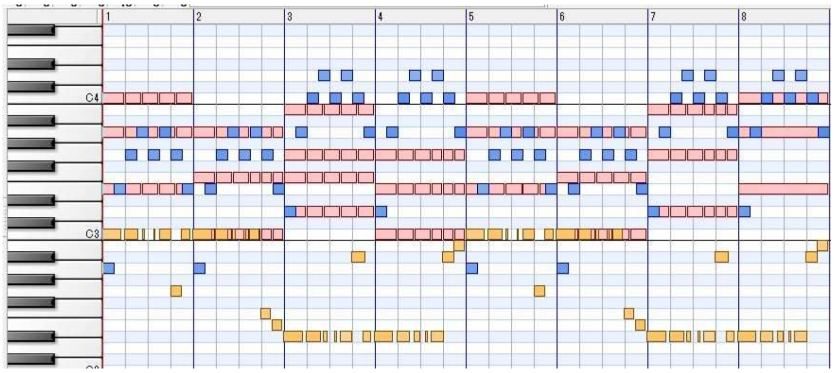

図 4.15　メロディのデータ内容

（※文責: 勝原望）

PDF Compressor Pro

既存の音楽から作成する方法について

オリジナルで音楽表現としての音楽を作成したほかに、既存の音楽を用いて MIDI ファイルに抽出して作成した。既存の音楽から作成する時も同様に、Varni ら (2010) の研究から、基本となるリズムは 1 小節に等間隔に 4 回バスドラムが置かれるようなドラムパート、またメロディに関しては低音の音が強く、また同じような音が繰り返すような音楽を探した。その結果今回 Royksopp の only this moment を採用した。

既存の音楽から MIDI ファイルを作成する方法として、今回はフリーソフトである「wavetone」を用いて音声ファイルから音を解析し、そのデータを用いて Domino を使って MIDI ファイルを作成した。wavetone とは Ackie Sound によって提供されている採譜支援ソフトのことで、MP3 や wav ファイルを等の音声ファイルを解析して、音階ごとの音量のグラフをピアノロール形式で表示できるソフトウェアである。

解析する音楽のファイルをこのソフトで開くと、ピアノロールにたいして赤から青で音量の大きさに対応した色の時系列データが表示される。only this moment の音楽ファイルを解析すると以下のような音楽の時系列データが表示された。

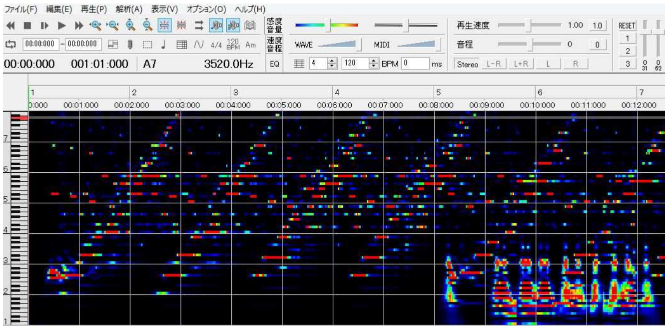

図 4.16　解析された音楽の画面

また、ここから必要なパートを抽出する。赤から青に対する音量の色の変化、表示される音を上の 2 種類のタブを用いて変更した。そのようにすることで、余分な音が省かれて、メインで流れているパートの音が抽出ができた。下の図を見ると、これは曲初めの部分であるが、およそ C5 の音階で繰り返し同じような音が流れていることが分かる。これは今回のシステムで使われる何回も同じ MIDI ファイルを繰り返し再生しても違和感ないような音楽に向いていると考えられる。これによって必要な音が抽出された時系列データからメロディとなる部分が採譜できた。

また、コードについてはこの wavetone の機能で採譜することができた。メニューバーにある解析を開き、コード解析を開いてコード解析を行ったあと、ファイルタブを開き、コードを MIDI ファイルに出力という操作を行うことで、この音楽にあったコードを作ることができた。

最後に Domino を用いてドラムパートを作成し、抽出したメロディと自動採譜されたコードを書き出すことで最終的にこのような音楽が作成された。メロディに関しては、抽出された音楽では C5 でハ長調の始点よりも 1 オクターブ高音であったため、全体を 1 オクターブ下げている。

（※文責: 勝原）

PDF Compressor Pro

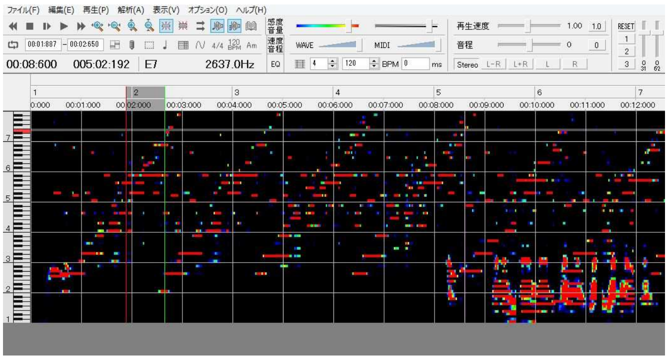

図 4.17　抽出された音楽の画面

Max との接続

Max と接続する際、MIDI ファイルの音源をそのまま流すだけではその作成したソフトの音源に依存していて、フリーソフトである Domino では音が乏しく、弱いということが懸念されている。そのため音源を追加するプラグインである VST を Max の中で動作できるようにして、音源を追加することで MIDI ファイルを編集して出力できるように接続した。そのことによって同期具合の変化に合わせて寄り音楽表現の変化が分かりやすくすることができた。

VST とは Virtual Sound Technology の略で有り、音楽制作ソフトの一つである Qubase を開発するメーカーである Steinberg が提唱するプラグイン形式のことである。音源を追加してそれを使用する VST instrument(VSTi)、ソフトとしてぞんざいするエフェクト以外に、エフェクトを追加して使用できるようにする VST effect(VSTe) がある。この規格に沿って製作されたプラグインは多くが操作を容易にするために GUI を採用している。VST はフリーソフトとして提供しているものもある。今回は VSTi フリーソフトを 2 種類使用した。ドラムの音源の追加に適している DR-II Fusion と、汎用性や自由度が高く使用がの多さから参考できる資料の多い Synth1 の 2 種類を用いた。

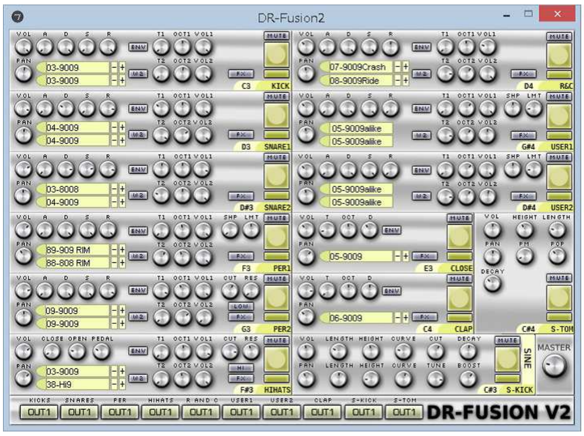

図 4.18　DR-FUSION

（※文責: 勝原望）

PDF Compressor Pro

図 4.19　Synth1

音楽班でのまとめ

最終的に作成された音楽データは、ドラムパートを追加し、メロディやドラムの音の使いの他に、音程にも注意して作成したことによって心理学的にもリズムがわかりやすくなるように作成できていると考えられる。また同期具合の信号に合わせて流れる MIDI の音量を変更するようにしたり MIDI ファイルに新たなパターンを追加して作成するなどといった機能を拡張した。その結果、動的に同期具合が変化する中で音楽表現が自然に変化できるようにできた。今回はこの音楽データを作成して機能を拡張するまでで活動は終了してしまった。今後は心理実験を通してより心理的にも効果的な音楽を探求したい。

（※文責: 勝原望）

4.2.4　照明班

照明班の概要

前期の成果物では、音楽でのみシンクロ現象を表現していた。しかしながら、人間には聴覚、視覚、味覚、触覚、嗅覚などの五感が備わっており、産業教育機器システム便覧によると、その知覚の割合は視覚が 8 割、聴覚が 1 割、その他の感覚器で 1 割 [14] といわれている。つまり、前期までの表現方法では様々な感覚器官のうち、約 1 割の感覚器官にのみ頼って表現していることになる。そこで後期では、聴覚を使う音楽表現に加え、視覚を用いた表現を作成することとした。また、視覚の表現には平面の映像に加え、立体の照明などの幅広く存在するが、デザイン班との区別化のために、照明班では立体の照明器具を用いたシンクロ現象の表現方法を作成にすることにした。

（※文責: 土手梓）

照明班の目標

照明班では、上記の通り、立体照明器具を用いたシンクロ現象を表現することを目標とした。しかし、一口に照明表現と言えども、多岐にわたるため更に条件の絞り込みが必要であることが分かった。そこで以下の 3 つの条件を満たした、シンクロ現象の表現方法の作成を目標とした。

1 つ目に参加者が一目で楽しめる表現である。これは表現が奇怪すぎて楽しさ以外の要素に参加

PDF Compressor Pro

者が集中してしまった場合、疑念が生まれ、シンクロしているということが楽しめなくなってしまうからである。そこで、照明班が目指す条件として、一目でわかりやすく参加者が楽しめる表現にする必要がある。

2 つ目に、今までの講義で学んだことを生かせる立体の照明器具を用いた表現である。これは、プロジェクトの期間中にプロトタイプを完成させ、より改良し易いように、初めの段階で技術の取得を早めるためである。

3 つ目に、テンポ同期班の Max から受け取った信号によって変化する表現である。これは、手動で表現を切り替えるのではなく、参加者のシンクロ具合によって得られた値によって自動で表現が変化する様に設定するためである。

以上三つの条件を満たす、シンクロ表現の作成を照明班の目標とした。

（※文責: 土手梓）

目標達成のプロセス

照明班の目標達成に際し、照明班は以下の活動を行った。その活動とは、表現方法の案だし、材料の調達、照明器具の作成、開発環境の整備、プログラムのプロトタイプの作成、発表時の準備、2 月の発表のための改良である。

（※文責: 土手梓）

表現方法の案出し

一目で楽しめる表現作成のために、まず、インターネットを用いて照明について検索をした。照明の中には、日常生活でインテリアとして使われる照明器具から、LED を用いた電光掲示板まで様々な種類が存在した。その中で今回の目標の条件である、一目でわかりやすく、楽しい照明表現にするために、さらにいくつか条件を加えた。一つ目の条件は、照明の電球は白色のみでなく、カラフルな電球を使うことである。これは、光の点滅のみでなく、色合いの変化も参加者が知覚し、楽しめる表現を作るためである。二つ目の条件は、立体の照明器具を使用することである。これは立体化することにより、変化を動きで認識しやすくするためである。上記の結果より、最も適した照明器具を考えた結果、Rainbowcube[12] に決定した。

（※文責: 土手梓）

材料の調達

照明器具の作成のため、Seeed Studio 社製 Rainbow cube kit[12] を購入した。また Rainbow cube は縦横高さが 4 個である計 64 個の立方体 LED 照明であったため、Arduino[8] では制御することができなかった。そこで Rainbowcube を制御するために定電流回路と LED ドライバが必要となり、新たに Arduino 互換ボードである Rainbowduino[13] を購入した。

（※文責: 土手梓）

PDF Compressor Pro

開発環境の整備

今回、Rainbowduino の開発環境として Arduino 1.6.5 を使用した。そのために Arduino1.6.5 を公式ホームページからダウンロードし、それぞれのパソコンにインストールした。次に Rainbowduino のライブラリをダウンロードし、プログラムを開発する Arduino 1.6.5 のライブラリに追加した。次に Arduino 内のボードとプロセッサ、ポート番号を設定した。Rainbowduino は Arduino より少し古い型だったのでボードを Arduino uno[9] から Arduino Duemilanove[10] or Diecimila[11] に変更した。またこの動作を行わないと、マイコンボードに書き込まれないことが分かった。以上が Rainbowduino の開発環境の整備に掛かる工程である。また、この工程は seeed studio wiki[13] ページを参照に行った。

（※文責: 土手梓）

プログラムのプロトタイプの作成

照明班では、Max と同期させる前に、表現方法の拡張として、いくつかプロトタイプの作成した。プロトタイプ作成の際はアイディアの拡張を目標としたため、arduino 内のみで変化する様にプログラムした。いくつか作成した中でインパクトのあるプログラムを紹介する。

maru.ino　このプログラムは、立方体の中に球体を描き、その中で順送りで円を描くように色が変化するプログラムである。まず、立方体内の球体点灯させるプログラムについて説明する。Rainbowduino のライブラリ内にある、Rb.setPixelZXY 関数で 1 個 1 個の LED を点灯させる。そこで、球体になるように、色を変え、球体の座標と色を上書きで点灯させた。次に順送りに弧を描くように点灯するプログラムの説明する。これは、for 文と delay 関数を用いて数ミリ秒ごとの遅れに、弧を描く軌道上に一個ずつ LED の色が変化しつつ点灯するプログラムである。また、色の変化には random 変数を用いて 24bit の RGB カラーコードの変化を行った。

（※文責: 土手梓）

dannMove.ino　このプログラムは、立方体の高さを段として、値が大きくなるごとに点灯する段が上に動くように LED が点滅するプログラムである。これは Rb.setPixelZXY 関数を用いて段の軌跡上に 1 段ずつ、点滅しながら上昇するようにプログラムを設定した。またその際に色が徐々に青から明るくなり白色に代わった後、カラフルな色に変わる様に設定した。またこの色の変化には Rainbowduino のライブラリ内にある、PlasmaCube のプログラムを参考とした。

（※文責: 土手梓）

suuji.ino　このプログラムは、立方体のある一面に 0 から 9 までの数字を表示するプログラムである。数字の表示方法として、縦 4 個横 4 個の LED 平面上にそれぞれ数字一個ずつ表示し、ある一定時間経過後次の数字を表示することとした。象るために Rb.setPixelZXY 関数を用いて LED を 1 個ずつ点灯させ座標を取得した。その座標を全て表示させた後、各数字に誤差がないか比較した。誤差を修正した後、次の数字が表示されるように象る作業を繰り返した。以上が数字を表示するプログラムである。またその際に数字の色にも変化を与えた。色の変化には虹の配色を用いて 0 を紫、5 を黄色、 9 を赤になるように変化をさせた。

PDF Compressor Pro

（※文責: 土手梓）

guruguru.ino　このプログラムは、ある一点が段々と底辺から頂点を螺旋階段上にまわるように見えるプログラムである。まず、Rb.setPixelZXY 関数と for 文を使い、すべての LED を点灯させせた。その後、一番底辺のある一点に違う色で LED を点灯させ、移動する点を作成した。次に delay 関数を用いて点を移動させせるプログラムを作った。実際に LED を動かすことができないので、ミリ秒ごとに 1 個ずつ点滅しながら移動するようにプログラムした。

（※文責: 土手梓）

発表時の準備

照明班では、前述の通り、Max と同期させる前に、いくつかプロトタイプの作成した。しかし、そこでは Max との同期は一切考えず、表現方法を作成したため、送られた数値に沿って変化するようにプログラムを組みなおす必要があった。また、Max と同期させるため、データ通信の方法も考える必要があった。また Max との連携以外に外見上の問題点も見つかった。担当教員である中田先生から、機材を地面にそのまま置くことで参加者が屈んでしまうことになり、傍観者の視点奇怪な状況であるのではないかという意見が出た。今までは外見に気を使うことはなかったが、その意見を踏まえ発表用に外見を整えることにした。最終成果発表前に上記の三つの問題点が存在しため、以下の三つの工程を行った。

（※文責: 土手梓）

Max とのデータ通信　今回、Max と連携させるために二つの手段を使った。一つ目の手段として、Maxuino を使い Max と連携させる方法を使った。Maxuino とは、Max 上で arduino を連携することができるオブジェクトセットである。この Maxuino を使い、Arduino の制御ができ、同期具合の数値を直接 Arduino に転送使用と試みた。しかし、Maxuino を使い、連携させる方法には大きな欠点が存在した。その欠点とは、Maxuino のバージョンが Rainbowduino に対応していないという点である。つまり、Maxuino では主に Arduino uno に対するデータ通信を行っており、Rainbowduino である Duemilanove or Diecimila には対応しておらず、Maxuino を使う連携方法はあきらめざるを得なかった。そこで二つ目の手段として、Serial 通信を用いて、Max と連携させる方法を使った。この方法は Serial 通信で Max 側からある通信速度である数値を送信し、Rainbowduino 側で受信する方法である。この場合、Max 側に送信するためのプログラムを追加し、Rainbowduino 側からも受信するためのプログラムを追加する必要があった。そこでこの方法を用いるか判断するためにプロトタイプで仮想的に実装し、実行した。結果は、ポート番号に注意しなければならない点を留意すれば、問題なく動作することが分かった。これらの理由から Max とのデータ通信に用いる方法として、Serial 通信の使用を決定した。

（※文責: 土手梓）

アクリルボックスの作成　外見を整えるために、アクリルボックスを二つ作成した。作成にあたり、工房職員の西野さんに協力を依頼した。一つ目の Rainbow cube を覆うアクリルボックスについて説明する。始めに、設計図を作成した。大きさとして、一辺が約 15 ☒になる立方体を目安した。この際、Rainbow cube の大きさに板厚が被らないこと,Rainbowduino の USB ケーブルが

PDF Compressor Pro

外に出ることの 2 点に注意し設計した。材料である、2mm のアクリル板を使用した。アクリル板の断面を垂直にするために, アクリル板を設計図通りに切断する必要があった。そこで、アクリル板の切断にはレーザーカッターを使用した。レーザーカッターの使用に際し、工房でレーザーカッター講習を受け、レーザーカッター使用許可証を取得した。またレーザーカッター用データを作成する際に、Adobe Illustrator の使用方法も学んだ。断面を切断した後、セロハンテープで仮止めし、アクリル版用接着剤で接着した。その後、接着するまで数日保管し、機材が見えないように乳白色のスプレーをかけ完成させた。次に二つ目の Rainbowcube をのせるアクリルボックスについて説明する。大きさとして、約 25 ⊠四方になる平面を持ち、高さが 55 ⊠となる直方体を目安した。工程は一つ目のアクリルボックスとほぼ同じであるが、上に荷重をかける分強固にしなくてならないため、板厚の違う 5mm アクリル版を用意した。また一つ目と板厚に変更が見られたため、設計図を再度作成した。その後、同じ工程を経て、アクリルボックスを完成させた。

（※文責: 土手梓）

発表に向けた表現の改良　最終成果発表に向けて、照明班では表現とそれに伴うプログラムの改良、及び付加を行った。まず表現の改良点として、他の班員から参加者の誰と誰が同期しているのかわかりにくいという指摘を受けた。今までの照明班では、シンクロ具合を数値のみで表現の対象としていた。しかし、この指摘によって、全体の数値のみでなく、個々のかかわりにもシンクロの表現が存在することを知ることができた。そこで成果発表に向けて、今までの表現に加え、個々のかかわりも追加された表現にするべく、プログラム作成を行った。

（※文責: 土手梓）

最終発表用のプログラム作成

doai3.ino　このプログラムは、参加者同士の組み合わせやシンクロ具合によって色合いと LED の点灯数が変化するように見えるプログラムである。まず、参加者同士の組み合わせを表現するために、Rb.setPixelZXY 関数で参加者を表す 3 つの塔を Rainbow cube 上に表現したプログラムを作成した。それぞれの参加者に光の三原色である、赤、青、緑の色を割り当て点滅するプログラムを作成した。今後それぞれの参加者を A、B、C と割り当てることとする。具体的には、参加者 A と参加者 B がシンクロした場合、赤と青の二つの塔が点滅し、参加者 B と参加者 C がシンクロした場合、青と緑の二つの塔が点滅する。参加者 A と参加者 C がシンクロした場合、赤と緑の二つの塔が点滅し、全員がシンクロした場合、赤青緑の三つの塔が点滅するプログラムである。これで、参加者同士の組み合わせを表現したプログラムは完成した。次にシンクロ具合によって変化するプログラムについて説明する。シンクロ具合を表現するために、三つの塔の高さが数値に反映するように設定した。具体的には塔の高さを引数として、for 文を用いて引数の高さにまで LED を点灯させるプログラムを作成した。この際、xy 座標において回路上で上下が反転するため、引数の高さの数値を反転させることに注意した。これで、シンクロ具合を表現したプログラムは完成した。次にシンクロ具合が最高潮になったときカラフルに点滅するプログラム説明する。このプログラムは、参加者 A,B,C のすべてがシンクロし、しばらく間シンクロが続いた場合に、表現を変えるプログラムである。具体的には、Rainbowduino のライブラリ内にある PlasmaCube1 のカラーシフトを参考に、Rb.setPixelZXY 関数内にある RGB のカラーコードを 100 ミリ秒ごとにランダムに切り替え点灯させるプログラムを作成した。これらより、参加者の組み合わせとシンクロ具合を表現するプログラムが完成した。

PDF Compressor Pro

（※文責: 土手梓）

Serial-Arduino.ino　このプログラムは、Serial 通信を用いて Max から送信された同期具合の数値を Rainbowduino で受信する Arduino 側のプログラムである。このプログラム作成にあたり、通信速度を 9600bps に設定した。また Serial.read() で送られた文字を読み込んだ際に、通信時以外に LED が点灯することを防ぐために Serial.available() で条件分岐をした。その後、条件分岐内で受け取った数値によって表現が変化させるために doai3.ino を用いて LED の点灯を行った。

（※文責: 土手梓）

2 月の発表における改良点

12 月の行われた成果発表において、大きく分けて参加者から意見が出た。一つ目に、光っているか分からなかったという指摘を受けた。この原因として照明班のプログラム内 delay 関数が影響し、信号受信内で遅延をおこなわれていたことが考えられた。このことにより、参加者自身がシンクロ同期を感じた瞬間に表現を変化させることができず、分からないという意見につながったと考えられる。二つ目に機材が小さく地味であるという指摘を受けた。これは照明の器具としてRainbowcube を用いたが、この照明器具の大きさが 15 ☒に満たず、やや小さいものなってしまったため、参加者からインパクトの少ないものとなってしまった可能性がある。また発表時が 12 月ということもあり、他の班員から 2m サイズのツリーを用いたほうが方がインパクトが強いという意見が存在した。以上二つの意見を検討した結果、一つ目の意見を 2 月の発表までに改良する必要あると感じ、2 月までの改良点として、プログラムを改編した。

プログラムの改良　2 月の発表にまで達成できる意見のうち、一つ目の光の遅延について改良を加えた。今回問題になっていた、delay 関数の代わりにタイマー割り込みを用いて、信号受信内の遅延を減らすよう改良した。それにより、今までの大幅な遅延が解消された。

4.3　デザイン班

4.3.1　デザイン班ができるまで

デザイン班は、映像班の Kinect 班から派生した班である。後期の 1 回目の話し合いでは、斎藤、田邉、小澤の 3 人で Kinect 班を担当しており、kinect や C ♯の基礎的な技術習得に努めていた。そこに前期のシステムで右手の動きを検出するシステムを担当していた立石が加わり、kinect による映像を使った動作検出を行うプログラムを作成する班と映像表現のシステムを作成する班に分かれ、役割を分担することとなった。役割分担の理由としては、映像班には、一から基礎的な技術を学び、それを応用したシステムを作成することと、同期具合の結果を映像表現することの 2 つの作業があり、作業を分担することで、各システムの質を高めることや作業の効率を上げるためである。動作検出のシステムでは、立石、斎藤が担当し、映像表現のシステムには、田邉、小澤が担当した。そして、動作検出のシステム開発を Kinect 班とし、映像表現のシステム開発をデザイン班として後期の活動を行った。

PDF Compressor Pro

4.3.2 目的

デザイン班では、本成果物における Max の同期判定の結果から、同期具合を映像として出力することを目的として活動した。同期具合を映像として視覚化することは、人に「楽しい」や「面白い」など何らかの心理的効果を与えることを示唆した。映像は Processing を用いてプログラムした。

前期の成果物では、同期判定の結果を出力する方法として、音楽による表現のみであった。識別した色の動きを同期判定することによって同期が確認されれば、初めに設定されたベース音にレパートリーを増やし、メロディを形成していくシステムを開発した。しかし、このシステムでは同期判定の際にずれが生じてしまって、再生される音楽がリズミカルでないことから操作した人が同期を体感しにくいことが分かった。これを改善するべく、図 4.20 のような人の得る情報量が最も多い視覚を刺激することが効果的と考え、映像表現を用いた。従って、このシステムに映像表現を加えることで、万が一形成された音楽がリズミカルに感じなくても、同期を体感させやすることを考えた。

図 4.20　五感の取得情報量

デザインにおいて、以下の 2 つが重要であると考えた。

- 同期を体感できるか
- システムから心理的効果を得られるか

前期の成果物は、音楽のリズミカル性の無さや同期判定のずれが原因で同期を体感しにくいものであった。これは、デザイン班でも言えることであり、操作する人が同期を感じることができなければ、映像表現の意味がなくなってしまうことをシステムを開発するうえでの重要な問題とした。

デザイン班の作業として、主なシステムの作成を田邊が担当し、デザインコースである小澤が本システムのキャラクター画像を作成した。次では、システムの詳細について説明する。

4.3.3 システムの詳細

初めに、出力する映像のデザインを考えるにあたり、同期の判定を聴衆に理解させやすい色の演出を考えた。同期判定により色を変化させるというものである。色には三原色である「緑」、「赤」、「青」の 3 つを使って、「同期していない」、「二人が同期した」、「三人が同期した」のそれぞ

PDF Compressor Pro

れの条件で色を変えていく表現方法を用いた。この表現を A とする。

次に A にキャラクターを加えることで、より同期具合の変化を分かりやすくすることを考えた。今回の同期の現象は、人と人の間に起こすものであったため、キャラクターを人型で表した。A にあった三色を基に、それぞれの色に対応したキャラクターを作成した。キャラクターには同期具合により踊る人数が変化する仕様を考え、「緑」、「赤」、「青」それぞれのダンサーを用意した。キャラクター 1 つにあたり 16 コマの画像をイラストレーターで作成し、合計 48 コマ作成した。図 4.21 は各色のキャラクターの 1 コマ目である。

[1] 緑のダンサー　[2] 赤のダンサー　[3] 青のダンサー

図 4.21　各色のキャラクター画像

これらの画像は各色で 16 コマの画像が次々に変化していくものである。同期判定により、画像の変化を制御し、A にあった条件に加えて、踊るキャラクターの組み合わせを変化させた。Max の同期判定では、0 から 7 までの数字により同期具合を分けており、その例を表 4.3 に示す。

表 4.3　各同期判定の映像出力

同期判定	キャラクターの組み合わせ
0,1,2,3	赤
4	緑と赤
5	赤と青
6	緑と青
7	緑と赤と青

表 4.3 は、Max による同期判定の結果から踊り始めるキャラクターの組み合わせである。同期班手が 0,1,2,3 のときでは、同期していないことを表すため赤のキャラクターのみであるが、ここでの 0 以外の数字の時は、照明による表現の出力で使用する数字であるので、映像表現では 0 と同様の条件にある。4 から 7 までの数字では、2 人以上同期した結果のキャラクターの出力である。これらの結果から、誰と誰が同期しているのかを分かりやすく判断できるようになった。

図 4.22 は、実際の出力映像である。同期具合の結果を映像として出力するためには、Processing と Max 間で同期判定のデータを送受信するが必要があった。その接続には、OSC と呼ばれるデータをネットワーク経由でリアルタイムに共有するための通信プロトコルを使用した。本成果物で

PDF Compressor Pro

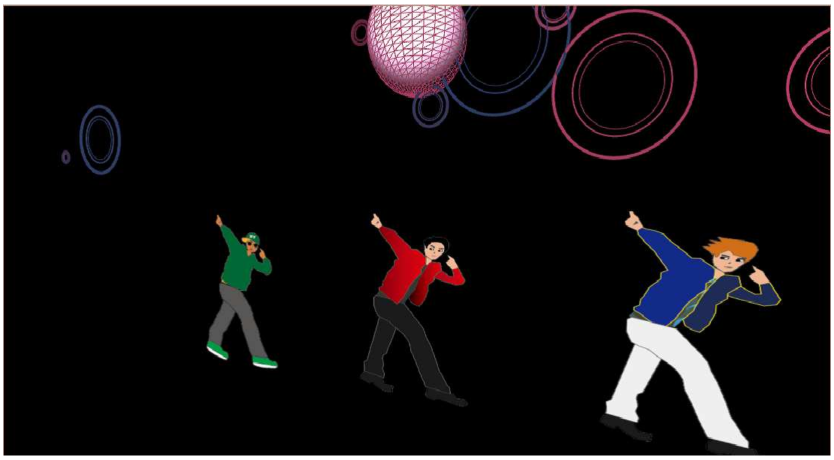

図 4.22　出力映像

は、1 つの PC 上でデータの送受信を行ったが、システムを改良することで 2 つ以上の PC でもデータの送受信が可能である。

以下では、このシステムの具体的な内容について説明する。このアニメーションプログラムには、二つのクラスが存在する。running man クラスと Ripple クラスである。

running man クラスは、同期班の同期判定を行うシステムから OSC によって同期した人数のデータを受け取り、それをアニメーションとして映像出力するメインのプログラムである。アニメーションの主役となる緑、赤、青の 3 つのキャラクターが踊っているように見せるために、frameRate() 関数を用いて、毎秒表示するフレーム数を指定し、画像の表示を連続して行った。本システムは、4 フレームで設定した。各キャラクターの画像を連続して表示する際に、各々の踊りに跳ぶ様子を表現するため、跳ぶ様子の画像が表示される場合にのみ image() 関数の Y 座標で変数の値を変動させた。また、このクラスでは、キャラクターを引き立たせるためにミラーボールに見立てたオブジェクトを作成し、ダンスホールの雰囲気をイメージしたデザインを考えた。このオブジェクトを作成するにあたり、sphere() や translate() を用いて、立体的な球を表現した。一派的なミラーボールは、色の変化が特徴的であると考え、オブジェクトの色を設定する specular() の各値をランダムで変動させた。オブジェクトをより立体的に見せるため、ambientLight() や lightSpecular()、directionalLight() などの光を設定し、陰影を表現した。アニメーションのデザインにおいて、聴衆に飽きを感じさせない工夫として、camera() を用いた視点の横移動によって色々な角度からの視聴も可能にした。

Ripple クラスでは、アニメーションの背景をデザインしたプログラムを組んでいる。こちらは、様々な大きさの波紋をランダムで表示するものとなっている。この波紋は、3 つの ellipse() による円の表示を用いて表現している。3 つの円はそれぞれストロークの幅が異なり、それぞれの大きさをフレームに合わせて同時に大きくすることで、動きのある波紋の表現がなされている。この波紋も running_man クラスのミラーボール同様、色をランダムで変化させている。波紋の変化をわかりやすく表現するために running_man クラスでは、background を黒に設定した。

以上の構成で、同期具合の結果を映像表現で出力した。

PDF Compressor Pro

4.3.4 システムの展望

これらのシステムから、同期具合をアニメーション映像で表現することに成功した。しかし、デザイン班のシステムでは映像による表現を行ってきたが、操作をした人の間で親和効果を与える具体的な結果は得られなかった。また、音楽や照明など他の表現との連動もできなかったために、操作する人の間で同期したことを実感することがまだ不十分であることが分かった。従って、音楽に合わせたアニメーションや照明とアニメーションの明るさを合わせることが必要である。例として、照明の光が強くなれば、アニメーションで同期している人のキャラクター同士が同じ色で光出すなどのエフェクト効果を追加するなど、アニメーションにも照明を意識した効果を作り出すことなどが、今後のデザインとして考えられる。

今後、デザイン班でのシステムを改良するにあたり、人と人との間で親和効果を生み出すために、より強い心理的効果を与える表現を考える必要がある。それには、他の表現のシステムと完全に連動させることが重要な部分となる。デザインの課題としては、本成果物のキャラクターは 2D であったので、より人間に近づけるために 3D で作成した場合の結果も期待できる。また、今回は単調なキャラクターの踊りだけであったので、ストーリー性を加えた新たな表現も、「楽しい」や「面白い」といった感情を促すことに繋がる可能性がある。また、本システムにゲーム性を加えて、ひとつの作業を操作する人全員で力を合わせてゴールに向かうといった、操作する人の間で団結力を高めることも親和効果を与えることも考えられる。本システムでは、Kinect から右手の振った情報を受け取り、その同期具合に合わせて、キャラクターが踊りを始めるようデザインした。実際の操作の動きやアニメーションの表現がリンクしていないのも、心理的効果や操作する人の間での親和効果を与えにくい要因であったと考える。今後のデザインでは、それらを踏まえたシステム開発を期待できる。

（※文責: 田邊峻眞）

PDF Compressor Pro

第 5 章　結果

5.1　前期

5.1.1　前期の結果

前期では、PC のカメラから 3 人の動きを検出して、その 3 人の動きが同期しているかを判定して、その結果を音楽として表現することに成功した。これにより人と人との間に同期現象を起こして、それを音楽によって実感できるようなものを制作することができた。私たちは前期の制作物として、人と人との同期具合によって、音楽が流れるシステムを制作した。

私たちはそれぞれ映像班、同期班、音楽班に分かれて作成した。映像班は PC のカメラから Processing を使用して動きの検出を行った。方法として、カラートラッキングという技術を使った。これは指定した色をカメラで追い、その速度、加速度から動きを検出した。しかし、カメラやノート PC のスペック・背景の色合いに依存しており、誤判定が多くなってしまうことが多かった。同期班は映像班から送られてきた 3 つのタイミング情報が同期しているか（だいたい同じタイミングで送られてきているか）を判定した。

方法として、MAX/MSP を用いて OSC で映像班から情報を受け取った。ただ、同期班は 3 人が同期しているか否かを判定していたが、誰と誰が同期していたか、2 人の同期はできていたかは判定できなかった。また、他の班からの助けもあり MAX 内で javascript を使用した。音楽班は同期班からの情報を元にして流すための音楽を作成した。音楽は既存の曲を使用した。パートごとに分割して作成し、さらに同期判定のために音楽を入れた midi ファイルは 2 小節ごとに 4 分割した。

（※文責: 坂田聖名）

5.1.2　中間発表

平成 27 年 7 月 10 日の 4，5 限にプロジェクト学習の中間発表を行った。発表場所は 1 階のプレゼンテーションベイの食堂側を使用し、発表場所のすぐ近くにポスター (図 5.1) を 1 枚展示した。発表者の両サイドにスクリーンを 1 つずつ設置し、発表者から見て右側に発表用のスライドを表示させ、左側にはデモムービーを表示させるようにした。デモムービーはパソコンを使用して再生した。発表する前に評価者に発表評価シートを配布し、評価者に発表評価をしてもらった。発表は前半 4 人、後半 5 人に分かれ 1 回の発表を 2 人ずつで行った。スライドを使用しながら、最初にシンクロとは何かについて軽く説明をし、どのようなものを作ったのかを一目で分かってもらえるようにデモムービーを流した。次に、私たちの前期の活動内容や製作物についての軽い説明、製作物の実演の順に行った。最後に製作物の実演をやるための時間を多く設けるために、それまでの説明はなるべく簡潔に丁寧に説明をした。実演の際には評価者から 2 人出てきてもらい、発表者 1 人評価者 2 人の計 3 人で実演を行った。実演は 1 回も失敗することなく、すべて成功した。その後に質疑応答をし、質疑応答が終わってから発表前に配布した発表評価シートを回収した。

発表評価シートに書かれている評価内容は発表技術についてと発表内容についての 2 つの評価である。この 2 つの評価をそれぞれ 1 ～ 1 0 の 1 0 段階で評価してもらい、さらにそれぞれの評価内

容のコメントを書いてもらった。まず発表技術についての評価である。発表技術についての評価の平均は 6.6 点であった。評価内容には、「練習不足が感じられた」、「声が小さい」、「もっと堂々と発表してほしい」などネガティブなコメントが多くあったが、「実際に実演をやっていておもしろかった」、「デモの動画があり、分かりやすかった」など実演やデモムービーに対しての評価はとてもよかった。他にも「スライドの説明が分かりやすかった」、「楽しいプレゼンだった」というコメントがあった。

次に発表内容についての評価である。発表内容についての評価の平均は 6.9 であった。評価内容には、「シンクロしているようには見えなかった」、「プロジェクトの目標がよく分からない」というネガティブな意見が多くあったが、他にも「楽しそうなプロジェクトだと思った」、「アプリ化してスマホで気軽に遊べるようにしたら楽しそう」などのポジティブな意見もいただいた。

中間発表の全体的な評価としては、評価内容のコメントにも多くあった発表の練習不足によっていい発表だったとは到底言えなかったと考える。プラグラム作成や調整に多くの時間を与えてしまったことによって、発表練習をする時間をあまりとれずに練習不足のまま中間発表の本番をやってしまい、声の大きさが十分ではなかったり、スムーズに発表を行うことができなかったりとグダグダな発表になってしまった。また、デモムービーの途中から急に音量が上がったり、余計な音が入っていたりと編集不足であった。他にも、評価内容のコメントで「プロジェクトの目標がよく分からない」と指摘されたように、あまりプロジェクトの目標の部分を伝えることができなかったと考える。スライドにはぼんやりとしたことしか書いてなかったり、口頭ではさらっと軽く説明しすぎたりとしていて伝わりにくくなってしまった。後期で行われる最終発表のときには、この中間発表で出た反省を生かして見に来てくれた人たちにがっかりされないようないい発表になるように発表練習の時間を多く設けてしっかりと発表練習をしていきたいと考えている。また、評価者に指摘されたシンクロのタイムラグについては、同期判定のプログラムを改善し、シンクロのタイムラグをなくし、体験者や評価者にシンクロしている実感を与えられるようにしていきたいと考えている。

図 5.1　前期ポスター

（※文責: 伊藤知幸）

PDF Compressor Pro

5.2 後期

5.2.1 後期の結果

後期では Kinect のカメラから 3 人分の右手の動きを検出して、何人が同期しているか、誰と誰が同期しているかを判定して、その結果を音楽、映像、証明によって表現することに成功した。これにより前期より正確な動きの検出を行うことが可能になり、また細かく同期判定を行い、3 つの要素で判定結果を表現することでより同期現象を実感できるようになった。私たちは後期の制作物として、人と人との同期具合によって、音楽が流れたり、光や映像で同期具合が表現されるシステムを作成した。

私たちはそれぞれ Kinect 班、同期判定班、音楽班、証明班、デザイン班に分かれて作成した。Kinect 班は Kinect センサーを用いて、3 人の右手の座標を取得して、3 人のうち誰かが右手を振ったときに、OSC を使って外部にデータを送信判定は、座標と時間から速度を求め、その加速度を記録していき、その変化をもとに行った。送信されるデータによって、受信側は誰が振ったかわかるように設定した。前期からの改善点はまず、Kinect を使用して、プログラムを新しく書き直したことである。前期で用いたカラートラッキングでは判定の正確さが背景の色に依存したり、カメラの性能から速い動きに対応できない場合があったが、Kinect を使用し、骨格の情報を取得することで基本的に背景の色合いにはあまり依存しなくなり、速い動きにも対応できるようになった。同期判定班は 3 つの入力のうち何人が同期したか、誰と誰が同期したかを判定した。前期からの改善点は何人が同期したか判定することによって同期したか、していないかではなくどの程度同期したのかを判定できるようになった。また、誰と誰が同期したかが判定できるようになったため、光表現や映像表現で同期を実感しやすいものが制作可能となった。音楽班はそれぞれのパートごとに midi ファイルを分割して作成した。音楽はオリジナルの物を作成した。そして、音楽が入っている midi ファイルの音量を調節できるようになった。また、深く同期が見られた場合のために通常の音楽を発展させたものも作成した。前期からの改善点はまず楽曲をオリジナルで音程やリズムなどがこのシステムに適しているものを作成したため、非常に前期よりわかりやすい楽曲となった。また、楽曲の音量を調節できるようになったため midi ファイルを 2 小節ごとに分割して、同期判定させる必要がなくなった。証明班は RainbowCube を使用して、同期具合に合わせて光表現を行うもの、プログラムを作成した。デザイン班は Processing を用いて同期具合によってキャラクターが動くアニメーションを作成した。証明班とデザイン班は後期からの新しい試みであった。これによりシステムの使用者により強く同期したという実感を与えることが可能となった。

（※文責: 坂田聖名）

5.2.2 最終発表の成果

西暦 2015 年 12 月 11 日金曜日午後 3 時 20 分からプロジェクト学習最終発表が行われた。日程はこの時間から 20 分区切りで 3 回前半の発表を行い、その後 10 分の休憩があってから 20 分区切りで 3 回後半の発表が行われた。

最初の前半の発表では実演が 3 回中 2 回失敗（音楽が流れないなどの不具合）が起こり、後半の発表では 3 回中全て失敗した。当日、発表についてのアンケートを取り、発表技術と発表内容について 10 段階の評価とコメントを記述していただいた。これは基本的に発表を見に来ていただいた

PDF Compressor Pro

方々全員にアンケート用紙を渡して学生、教授、企業の方々などから評価をいただいた。10 段階評価について、平均を取ったところ、発表技術は平均約 6.06 点、発表内容は平均約 6.77 点という結果であった。コメントについて、発表技術では声の大きさを指摘しているコメントが多くみられた。発表内容では目的がわかりにくい、実演が見られなくて残念という意見が多くみられた。

（※文責: 坂田聖名）

5.3 成果の評価

1 つ目の課題である Processing と Max の学習は、プロジェクトメンバー全員で行った。まず初めにほとんどの人が今まで扱ったことのなかった Max の学習をすることにした。Max の学習は、プロジェクトメンバー全員で Max に詳しいメンバーに聞いたり、Max のチュートリアルにあるものを実際に作ってみて動かしてみたりして Max の基本的なプログラムの組み方を身に付けることができた。さらに、映像・音楽表現を実現させるために最初に学んだ基本的なプログラムの組み方を応用し、本やサイトを見るなどして映像・音楽表現の実現に有効な技術を身に付けることができた。また、Processing を用いた画像処理の学習は、Processing を使用する映像班が主に行い、教員から配布された画像処理のプログラムの解説を聞いたり、実際に自分自身でプログラムをいじったりして画像処理について理解することができた。この 2 つの学習を行ったことでプロトタイプの製作に取り掛かることができた。したがって、1 つ目の課題である Processing と Max の学習は達成された。

2 つ目の課題であるプロトタイプ製作は、映像班、同期班、音楽班の 3 つのグループに分かれて作業を行った。製作物の提案については、メンバー全員で話し合い、ジェノバ大学で行われたシンクロ現象の先行研究を発展させたものを製作することに決定した。プロトタイプを製作するにあたってそれぞれのグループに課題を割り当てた。まず映像班の課題は、Processing でカメラからの入力をもとに複数の物体の動きを個別に判定するプログラムとその判定した情報を Max に送る仕組みを作成することであった。この課題を解決するために教員から解説されたプログラムに活用されていた色を追跡するカラートラッキングを使用した。色を追跡するカラートラッキングを用いることでカメラの入力から物体の動きを色で判定するプログラムを作成することができた。物体の動きの判定を Max に送る仕組みについては、OSC を導入することにより解決された。しかし、この物体の動きを判定するプログラムには少し問題があった。カメラの画面内に動かす物体の色と似たような色があるとそちらにも反応してしまいうまく判定してくれないということである。この問題を解決するためにはさらに判定の精度を高める必要があったと考えた。これは後期に作成するプロトタイプの時に生かしていくことにした。

次に同期班の課題は、Max を用いて、Processing から送られる情報をもとに複数の物体の同期具合を判定するプログラムを作成することであった。この課題は、システムが動き始めたと同時に時間とカウントし、タイミング情報が送られてきた時の時間をそれぞれ記録し、その時のそれぞれの時間の差が 200 ミリ秒以内であれば物体が同期していると判定し、これを 2 小節ごとに物体の同期具合を点数化することで同期具合の判定するプログラムを作成することができた。しかし、このプログラムには問題点があった。それは 2 小節ごとに同期具合の判定を行うことで物体が同期した直後に音楽が流れるのではなく、少し遅れて音楽が流れるので、これをシンクロ現象とは言えるのかというのが問題点であった。シンクロしていると実感してもらうためにはこのタイムラグをなくす必要があった。この問題点は、2 小節ごとに同期具合の判定を行ったいたものを 1 拍ごとに同期

PDF Compressor Pro

判定を行うことで解決していく。

最後に音楽班の課題は、同期班によるプログラムから判定された同期具合によって音楽が再生されるシステムと音楽の作成であった。音楽の作成については、再生させる音楽をほとんどの人が聞いたことのあるカノンにし、MIDI 作製フリーソフトである"sakura"を使用することで MIDI ファイルの作成を行った。音楽をベース、コード、メロディの 3 つのパートに分け、パートごとに MIDI ファイルとして作成し、その 3 つのファイルを Max で流せるようにした。そして、同期具合の点数によってベース音、伴奏、メロディーの順で音楽が形成されるようにし、音楽班の課題は解決された。以上の 3 つのグループの課題解決によりプロトタイプが完成された。

3 つ目の課題である前期で作成したプロトタイプをもとにより発展させたシステムの構想を練ることについては、2 つのグループに分かれ、それぞれのグループでアイディアを考えた。意見を出し合った結果、様々なアイディアが提案されたが、キネクトというモーションセンサーデバイスを用いて全身の動きを読み込めるようにしたり、照明器具を用いてシンクロ現象に関する新たな表現を考え、体験者が楽しさを得られるようなゲームの製作をし、そのゲームで表現される音楽や映像が体験者にどのような心理的な効果を持つのかを心理学的に証明していくことに決定した。また、体験者にシンクロの実感を与えることができるように前期で作成した製作物の問題点であったシンクロのタイムラグを改善することを試みることにした。

最後の課題である前期に制作したプロトタイプをより発展、改善させたプログラムの作成は、Kinect 班、同期班、音楽班、照明班、デザイン班の 5 つのグループで作業を行った。発展したプロトタイプを作製するにあたってそれぞれのグループに課題を割り当てた。まず Kinect 班の課題は、Kinect センサーを用いて、人体の骨格を取得、追跡することで複数人の動きを個別に判定するプログラムの作成と Kinect を使うために C ♯の技術を習得、C ♯のプログラムで Max に送る仕組みを作成することであった。この課題は、Kinect v2 の深度センサー利用しすることで人体の骨格を取得し、右手の座標を取得することができた。さらに動きを判定するアプリケーションである Visual Studio Express 2013 を使用し追跡することで、複数人の動きを個別に判定するプログラムの作成は解決された。また、OSC を導入することによって C ♯のプログラムで Max に情報を送信する仕組みを作成することができるようになった。C ♯の技術習得に関しては、C ♯に詳しいメンバーに聞いたり、インターネットや本で調べたりしてそれぞれで学習することで C ♯を習得することができた。したがって、Kinect 班に与えられた課題は達成された。

次に同期班の課題は、前期に作成したプログラムを発展させ、利用者の同期具合によって小節ごとにテンポを変化させる機構を作成することであった。この課題は、1 小節ごとに同期している人数で判定するプログラムや小節ごとにテンポを変化させるプログラムに変更し、前期に作成した 2 小節ごとに物体の同期具合を点数化して同期具合の判定をしていたプログラムを発展させようとした。しかし、小節ごとにテンポを変化させることができなかったため、最初の 4 回分のテンポの平均を出し、その平均のテンポでずっと音楽を流すプログラムに変更した。当初に与えられた課題である小節ごとにテンポを変化させる機構の作成は達成することができなかったが、最初の 4 回分のテンポによって音楽のテンポを変えることができた。したがって、音楽班に与えられた課題はおおむね達成された。

次に音楽班の課題は、利用者が同期しやすいと感じられるような音楽を作成することと再生される曲の音色を変更することであった。この課題は、前期で作成した曲ではなく、より同期しやすいと感じられるように自分たちでオリジナルの曲を作成することで解決した。オリジナル曲の作成には、MIDI 編集ファイルである「Domino」を用いてメロディ、ドラム、コードの 3 つのパートにわけて作成を行った。また、同期している人数が MAX の状態がしばらく続くと、それぞれのパー

PDF Compressor Pro

トが少し変化して曲が再生されるように音色が変化したバージョンを作成した。しかし、Dominoの音源では音が乏しく、音が弱かったので、音源を追加するプラグインである VST を Max 内で動作するようにし、音源を追加、または編集して出力できるように接続した。そのことにより、同期具合によって音楽の表現を変化させることに成功した。したがって、音楽班に与えられた課題は達成された。

次に照明班の課題は、RainbowCube kit を用いて、参加者が楽しめるような光の表現方法を探求することと利用者の同期具合によって変化する光の表現を作成することであった。この課題は、照明器具を立体化にすることで変化を動きで認識しやすくなり、さらに照明の電球を白色だけにするのではなく、カラフルな電球を使用することで光の点滅のみでなく、色合いの変化も参加者が知覚し、楽しめる表現をつくることができた。また、参加者が楽しめるように同期具合によって色彩を変化させることで表現方法を探究することは解決された。RainbowCube kit の開発環境としてArduino を使用し、Arduino と Max を連携させるために Serial 通信で Max 側からある通信速度である数値を送信し、Rainbowduino 側で受信する方法を使用した。このことによって、Max 側と Rainbowduino 側にそれぞれ少しプログラムを追加するだけで Max と Arduino を連携させることに成功した。したがって、照明班に与えられた課題は達成された。

最後にデザイン班の課題は、Processing を用いて、アニメーションを作成することと利用者の同期具合によって変化するアニメーションを作成することであった。この課題は、同期具合を視覚で捉えることができるように Processing で同期具合によって「赤」、「緑」、「青」の色が付いた 3 人のダンサーが踊っている人数が変化するアニメーションを作成することで解決された。また、3 人のダンサーの色をそれぞれ分けて誰と誰がシンクロしているのかを分かりやすくすることができた。Max と Processing を連携させるために OSC を用いて 1 つの PC 上でデータの送受信を行うことが可能となった。しかし、音楽と照明など他の表現と連動することができず、シンクロしたことを実感することが不十分になってしまった。作成された音楽に合わせたアニメーションや照明に合わせたアニメーションの明るさを合わせることが必要であったと考えた。したがって、デザイン班に与えられた当初の課題はおおむね達成されたが、新たな課題ができた。以上の 5 つのグループの課題解決によりプロトタイプが完成された。

（※文責: 伊藤知幸）

PDF Compressor Pro

第 6 章　今後の課題と展望

6.1　前期の課題と展望

前期の活動までは 3 人の被験者がノート PC を利用したカラートラッキング技術を利用により、被験者の動きの一部の座標や加速度を大雑把に認識できた。この情報を Max というアーティストなど様々な場所で利用されているバーチャルプログラミング言語を利用することで、認識した座標や加速度から音楽表現という異なる表現によりシンクロ同期を聴衆に表現した。前期は被験者同士のシンクロ同期が安定しない、聴衆が見ていてつまらないという結果になった。
今後の課題として、まずシンクロ同期を安定させるようなシステムを構築する。前期ではノート PC のカメラを利用したので、人を認識し取得できるカメラを使用したい。更に聴衆が見ていてつまらないというアンケート結果から、聴衆も楽しめるような新しい表現方法を提案したい。

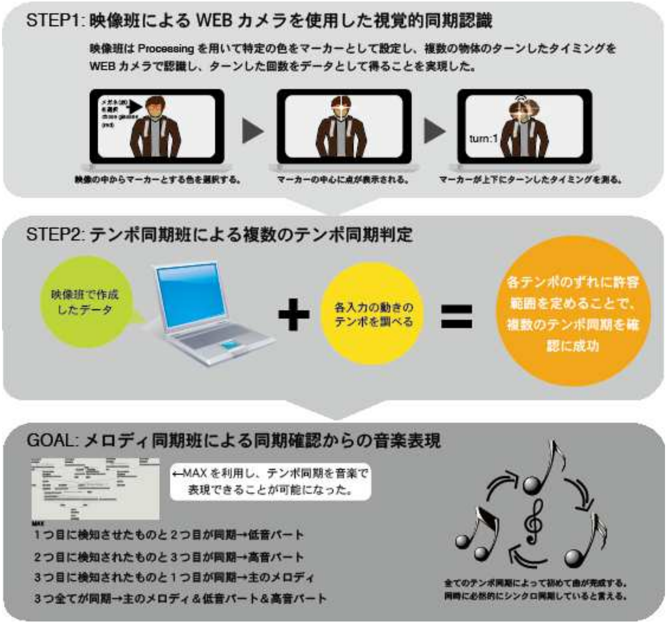

図 6.1　前期のシステム

（※文責: 小澤繁輝）

6.2　後期の課題と展望

前期の展望から、後期では Xbox などゲーム機にも使用されている kinect というデバイスを利用した。このデバイスはジェスチャーや音声認識による操作が可能なため様々な場面で利用できる。私たちは kinect の RGB カメラと深度センサーを利用し、被験者の位置、動きをより繊細に認識できるようになった。

また、聴衆が楽しみながら被験者の同期を理解できるような表現方法として、音楽表現以外に映像表現と照明による表現方法を提案した。映像表現として聴衆に見てもらうために Processing に

PDF Compressor Pro

よるアニメーションを作成した。アニメーションの中では被験者の数だけプレイヤーを表示するようにしてあり、被験者に特定の色を付与することで、聴衆がそのリズミカルなアニメーションを見て楽しく理解てもらおうとした。照明の表現方法として RainbowCube を利用し、被験者には RainbowCube を見ながら実験してもらった。RainbowCube も複数の被験者に特定の色を付与することで被験者の動きを使役し、被験者同士のシンクロ同期が安定しないという問題を解決しようとした。発表の時には被験者に色付きのサイリウムを持たせることで被験者に特定の色を付与した。

これらの表現方法やシステムの改良によりシンクロ同期が被験者と聴衆に対して前期に比べて格段に理解しやすいものとなった。アンケート結果からも表現方法に関しては評価が高かった。しかし、このプロジェクトの本来の目的であるシンクロ同期がもたらす被験者同士の親和効果がこの発表から理解することができなかった。また、発表での実演の際に被験者が照明の表現を見ながら手を振るかアニメーションを見ながら手を振るかで迷う時があったため、表現方法として完成出来ていなかったのも問題の一つとして挙げられる。

今後の展望としては、まず誰に向けての表現方法なのかを定めるようにする。そしてプロジェクト本来の目的であるシンクロ同期がもたらす被験者同士の親和効果を確認するため、他の場所で実験してデータを取得したり、親和効果が高まるような心理的効果をもたらすシステムを構築したい。

（※文責: 小澤繁輝）

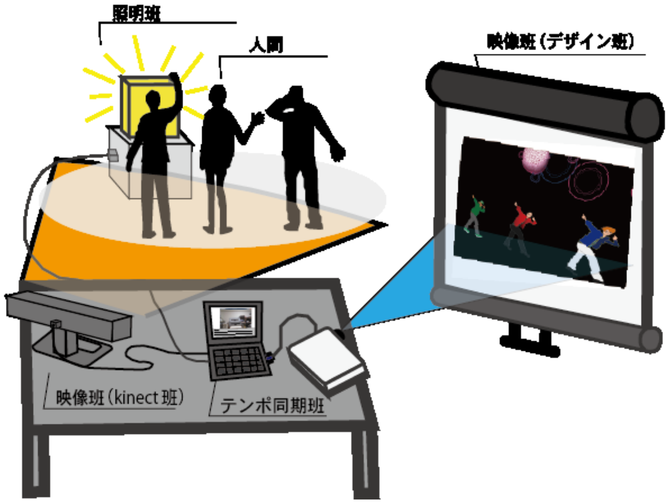

図 6.2　後期のシステム

6.3　今後の展望

このプロジェクトは今年発足したばかりで、私たちの活動のほとんどはプロジェクト本来の活動内容を理解することに力を注いだ。私たちは今回シンクロ同期には意識する同期と意識しなくても起こる同期が存在することを理解し、全体を通してこのプロジェクトでは意識させて起こる表現方法が取り上げた。今後はよりわかりやすく楽しい表現方法を見つけたい。

（※文責: 小澤繁輝）

PDF Compressor Pro

6.3.1　ユーザーのモチベーションを上げたい

例えば、誰でも簡単にライブ形式で踊ることができるゲーム『DanceEvolution』がある。これは聴衆が見ていて楽しい、ゲーム形式であるためユーザーのモチベーションが上がる、そしてこのゲームは協力によってゲームをクリアするので、必然的に協力性により親和効果がもたらされるなどの私たちが抱えている問題をクリアできる（図 6.3）。なので、今後はゲーム形式にしたり協力するようなシステムを構築してみたい。

図 6.3　DanceEvolution

（※文責: 小澤繁輝）

6.3.2　全く新しい表現を提案する

シンクロ同期を応用したシステムを構築してみたいです。普段は気づかぬうちに他の人とシンクロしていたり、他の人とシンクロするために動きを合わせようとします。しかし、このシステムは他の動きを見ながらではなく無意識にシンクロしているか確認できるものです。ここで「シンクロニシティー」（共時性）という心理的効果があります。特にその中で有名なのがボディーシンクロニー（姿勢反響）があります。心の近い人同士（友達、恋人）が一緒に会話をしているといつの間にかしぐさが一致してしまう現象のことです。会話中にしぐさが一致するだけでなく言葉遣いや表情、そして呼吸のリズムなども同調してくる傾向があります。このシステムでは表情に注目して無意識に同期しているかを確認できるものです。実装方法として具体的には提案できませんが、例えば表情を kinect 等で認識しその表情を画面上に表現する。

（※文責: 小澤繁輝）

6.3.3　実験

今回のプロジェクト活動の成果はほとんどアンケート結果から得られた意見をもとに改良した部分が大きいので、この活動を続けるのであれば、函館のまちづくりセンターや青年センターでの発表により一般市民にも利用してもらう機会を作れたらいいと思う。

（※文責: 小澤繁輝）

PDF Compressor Pro

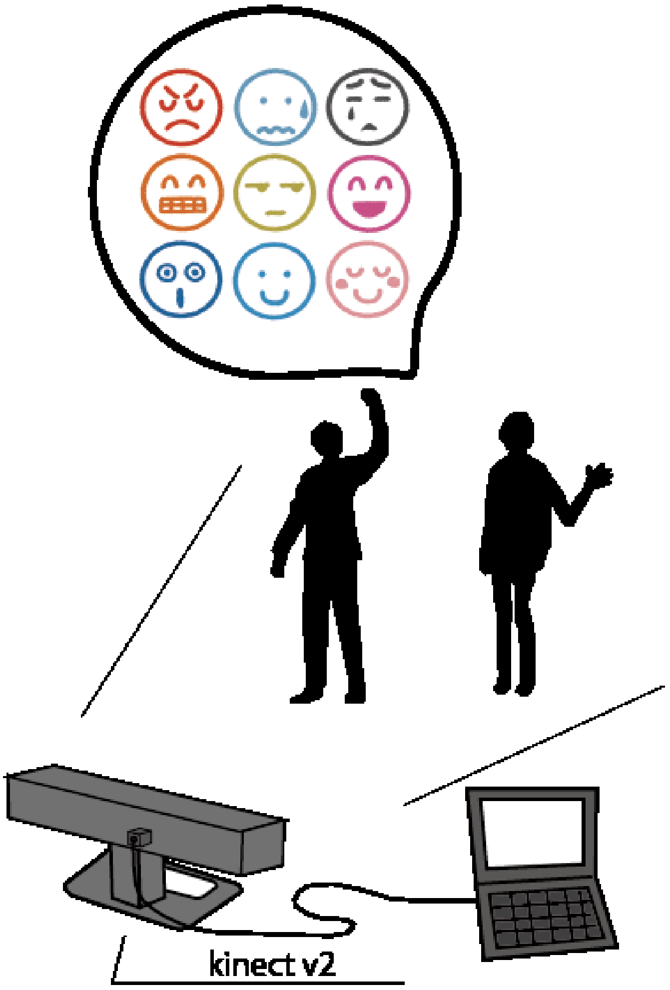

図 6.4　表情を利用した新しいシンクロ同期

6.4　中間発表のアンケート結果

6.4.1　中間発表の日時・場所

平成 27 年 7 月 10 日の 14:50 〜 18:00 に 1 階プレゼンテーションベイにて行われた。

（※文責: 小澤繁輝）

6.4.2　アンケート結果

発表技術について

- デモがあるのがいい
- デモがスムーズに動いていなくて残念だった
- 発表者の動きが気になる
- 色紙を持っていなくても、音が流れ続ける
- 動画を流したのが良かった
- 例を多く上げていて良かった

PDF Compressor Pro

- 1 枚のスライドに対して文字が多い気がした
- スライドに工夫がほしい
- 映像スクリーンの位置をもう少し高くしてほしい
- 少し早口だったが、身振り手振りがわかりやすかった
- 映像と発表者の発表がかぶっていた
- 分からない用語が出たため理解できなかった
- 時間が来たら始めよう (ダラダラしているように見えた)
- 発表練習をしてほしい

発表内容について

- まず「シンクロ」の説明ではなく、どういう目的のプロジェクトなのか説明してほしい
- なぜ Max なのか気になる
- 研究の背景が紹介されていて、プロジェクトの内容がわかりやすかった
- 「シンクロする」ということをカメラ等でどこまで判別しているかわかりにくく感じた
- カラートラッキング技術がわかりやすかった
- 発表の際にカラートラッキング技術を利用していたので、どう実装するか気になった
- 色とテンポから音を形成するという発想が面白かった
- 目標の一つのゲームが楽しみ
- 心理学的にどのような効果を持つのか明らかにした後、どう活かすか気になる
- 是非キネクトを用いてほしい
- Web カメラのディレイを少なくしよう
- 上下以外の動き (左右、前後等) でもシンクロしてみてほしい
- 伴奏のみとメロディー入りの違いが分かりにくい
- 対象者をだれか明確にしてほしい
- デモから感じることを伝えてほしい
- なぜ同期を用いると面白いのか?
- アプリ化もありかも
- 班の分担がよく分からなかった
- 最終目標がわからなかった
- 映像表現がほしい
- UX, 体感のデザインがほしい

（※文責: 小澤繁輝）

6.5 最終発表のアンケート結果

6.5.1 最終発表の日時・場所

平成 27 年 12 月 11 日の 14:50 ～ 18:00 に 1 階プレゼンテーションベイにて行われた。

（※文責: 小澤繁輝）

PDF Compressor Pro

6.5.2 アンケート結果

発表技術について

- 発表者が変わるなら言ってほしい
- 作ったシステムを見せてもらい、どんな活動か理解しやすかった
- リハーサルをもう少し行ってトラブルを減らしてほしかった
- スライドに写真などを用いると分かりやすいと思う
- デモによって何を伝えたいのかが見ている人に伝わりにくい
- ムービーを用意してほしかった
- 1 枚のスライドに対して文字が多い気がした
- 強調している部分がわかりづらい
- 専門用語をわかりやすく説明してほしい
- スライドに時が多いので、block diagram で表現したほうがわかりやすいと思う

発表内容について

- ダンスのシンクロとかなら、将来性がありそうだと思った
- 開発してどうするのか
- 心理的な効果を与える理由は
- 体の動きから何かを発見するのは面白かった
- 展望を具体的にしてほしかった
- 様々な動きでシンクロ同期を試してほしい
- プロジェクト全体として何を成し遂げたのか理解しにくく少し残念
- 照明が光ったのがよく分からなかった
- まだディレイが少し残っていた
- 目的を定めるまでのプロセスがほしかった
- 班と協力しているのが伝わってきた
- キューブ LED をもっと大きくしてほしい
- スライドに目標を書いてほしかった
- 自信をもって「こういうことがやりたかった!」とアピールしてください
- デモが参加型だったのでやってみたいと思った
- Kinect を使った面白いコンテンツだと思った
- デモでの同期中被験者の目線が気になった
- なぜ音楽なのか
- 使うときのシチュエーションを考えるべきだと思った
- 実験データがほしい
- 判定が厳しい
- 同期の視覚化がダンサーなのはなぜ
- 体の動きによる曲調やテンポのバリエーションが増えるといい

（※文責: 小澤繁輝）

PDF Compressor Pro

付録 A　新規習得技術

- Max
- processing
- JaveScript
- sakura
- TeX
- Kinect V2
- Arduino
- C ♯
- Open Sound Control

PDF Compressor Pro

付録 B　活用した講義

- 情報表現基礎
- 画像工学
- Tex 講座
- ポスター講習
- 認知心理学

PDF Compressor Pro

参考文献

[1] Cycling '74 | MAX 7 — エレクトロニックアート&ミュージックの共通言語 - MI7 Japan. http://www.mi7.co.jp/products/cycling74/max/. 最終更新日 2016 年 1 月 13 日. 最終閲覧日 2016 年 1 月 13 日

[2] Processing.org. https://processing.org/. 最終更新日 2016 年 1 月 13 日. 最終閲覧日 2016 年 1 月 13 日.

[3] Xbox One Kinect センサー. http://www.xbox.com/ja-JP/xbox-one/accessories/kinect-for-xbox-one/#fbid=e1ZZ4JJjA4Q. 最終更新日 2016 年 1 月 13 日. 最終閲覧日 2016 年 1 月 13 日.

[4] テキスト音楽「サクラ」. http://oto.chu.jp/top/. 最新更新日 2015 年 9 月 27 日. 最終閲覧日 2016 年 1 月 13 日.

[5] openFramewoks - OSC (Open Sound Control) を利用したネットワーク連携 — yoppa.org. http://yoppa.org/ma2_10/2279.html. 最終更新日 2016 年 1 月 13 日. 最終閲覧日 2016 年 1 月 13 日.

[6] rugcode / Rug.Osc - Bitbucket. https://bitbucket.org/rugcode/rug.osc. 最終更新日 2016 年 1 月 13 日. 最終閲覧日 2016 年 1 月 13 日.

[7] Hove, MJ, Marie, C, Bruce, IC, and Trainor, LJ (2014). Superior time perception for lower musical pitch explains why bass-ranged instruments lay down musical rhythms.. Proc Natl Acad Sci U S A, 111(28), 10383-8

[8] Arduino-Home. https://www.arduino.cc/. 最新更新日 2016 年 1 月 10 日. 最終閲覧日 2016 年 1 月 10 日.

[9] Arduino UNO & Genuino UNO. https://www.arduino.cc/en/Main/ArduinoBoardUno. 最新更新日 2016 年 1 月 10 日. 最終閲覧日 2016 年 1 月 10 日.

[10] Arduino Duemilanove. https://www.arduino.cc/en/Main/ArduinoBoardDuemilanove. 最新更新日 2016 年 1 月 10 日. 最終閲覧日 2016 年 1 月 10 日.

[11] Arduino Diecimila. https://www.arduino.cc/en/Main/ArduinoBoardDiecimila. 最新更新日 2016 年 1 月 10 日. 最終閲覧日 2016 年 1 月 10 日.

[12] Rainbow Cube kit RGB 4X4X4. http://www.seeedstudio.com/depot/Rainbow-Cube-kit-RGB-4X4X4-Rainbowduino-Compatible-p-596.html. 最新更新日 2016 年 1 月 10 日. 最終閲覧日 2016 年 1 月 10 日.

[13] Rainbowduino v3.0. http://www.seeedstudio.com/wiki/Rainbowduino_v3.0. 最新更新日 2015 年 8 月 25 日. 最終閲覧日 2015 年 12 月 21 日.

[14] 教育機器編集委員会. 産業教育機器システム便覧. 日科技連出版社. 1972.

[15] ビジュアルの効果 | テクニカルイラスト・テクニカルライティングのキテックス. http://www.kitex.co.jp/visual/visual.html. 最終更新日 2011 年 12 月 14 日. 最終閲覧日 2016 年 1 月 13 日