

平成 27 年度 公立はこだて未来大学卒業論文

Leap Motion による教育をインタラクティブにする システムの提案

一ノ瀬 智太

情報アーキテクチャ学科 1012211

指導教員 美馬 義亮

提出日 平成 28 年 1 月 29 日

Interactive system for education by leap motion sensor

by

Tomohiro Ichinose

BA Thesis at Future University Hakodate, 2016

Advisor: Yoshiaki Mima

Information Architecture
Future University Hakodate
January 29, 2016

Abstract— Teachers use a blackboard and chalk in the classroom. Students has been used pencil and notebook. However, classes that use a personal computer and tablet devices have been introduced in recent years. Also, new devices has introduced. They are Leap Motion and Ring, Smart Watch. Thus, user come to not push button and wear devices. It is that the exchanges of the information from a teacher to a student are unilateral to be common when I compared the recent school education with the conventional thing. In this study, I assumed that use of leap motion when implement application which capable input method. It is not mouse and keyboard. In this study, the purpose is to implement an application which capable this function, and evaluate usefulness.

Keywords: Interaction, Leap Motion, Gesture

概要: 学校教育の場において教師は黒板とチョークを使い、学生は鉛筆やノートを使った授業が主流であった。しかし、近年ではパーソナルコンピュータやタブレット端末を使った授業が取り入れられている。また、新しいデバイスが登場している。Leap Motion や Ring, スマートウォッチなどが例である。これによりボタンを押すという動作が減ったり、デバイスを身につけるという変化が起こっている。近年の学校教育を従来のものと比較した際に共通することは、教師から学生への情報のやり取りが一方的なことである。本研究では、Leap Motion を使うことを前提として、マウスやキーボードにない入力方法でメッセージを発信するアプリケーションを実装する。そのとき提案するアプリケーションが話者の邪魔をしたり、話者の応答がないものであってはならない。本研究では、これらを踏まえたアプリケーションを実装し、その有用性を評価することを目的とする。

キーワード: 双方向性, Leap Motion, ジェスチャー,

目次

第1章 序論	1
1.1 背景	1
1.2 研究目標	1
第2章 関連研究	2
2.1 クリッカー [2]	2
2.1.1 提案するシステムとの違い	2
2.2 サイボウズ Live [3]	2
2.2.1 提案するシステムとの違い	3
第3章 提案するシステムの概要	4
3.1 提案するシステムの特徴	4
3.2 クライアント側のプログラムの役割	4
3.3 サーバー側のプログラムの役割	7
3.4 実装方法	10
3.4.1 言語・ツール・開発環境	10
3.4.2 Cinder フレームワークについて	10
3.4.3 Leap Motion について	11
第4章 実験と評価	14
4.1 実験の目的	14
4.2 方法	14
4.2.1 日時と場所	14
4.2.2 被験者	14
4.2.3 道具	14
4.2.4 手順	14
4.2.5 アンケート	15
4.2.6 アンケート結果	16
第5章 考察	21
5.1 話し手のアプリケーションに対するリアクション	21
5.2 話し手の邪魔になっていないか	21
第6章 結論と今後の展開	22
6.1 まとめ	22
6.2 今後の方針	22

第1章 序論

1.1 背景

学制百年史 [1] によると日本における大学の誕生は明治時代に遡る。これらは一斉授業の場において、教師は知識を与え学生はそれを吸収するというモデルに基づいている。こうした場において、一部の学生には頼杖をついたり、居眠りをしたりすることなどが散見される。その原因として、日々の生活の疲れだけでなく、教師の説明が理解できず、授業についていけないなどのストレスが挙げられる。こうした学生には他の学生に比べて学習意欲が低下しているといえる。

大学のゼミにおいても同じような状況があるといえる。話し手が進捗報告や自分の考えを一方的に話し、聞き手はそれに耳を傾ける。

近年では、Leap Motion や Ring, スマートウォッチなどの新しいデバイスが登場している。その結果、今までマウスやキーボードで入力する方法以外にも、ジェスチャーを使って入力したり PC を操作できるようになった。また、学校内での LAN や学外でのロボットコンテストなどが普及してきており、PC やタブレット端末などが教育でも使われるようになってきたといえる。

1.2 研究目標

本研究では、授業中やゼミの中で情報のやりとりが一方的に行われている問題に着目した。この問題を解決するために教師や発表者である話し手と、話を聞く聞き手との間に双方向性をもたせ、対話的な授業運営、ゼミの進行を図ることを研究目標とする。その方法として、Leap Motion というセンサが手や指の位置、方向、曲げ、ジェスチャー等を検知できるセンサを用いて、双方向性を実現するアプリケーションを実装することを目的とする。具体的には学生が授業中に抱いている「大きな声で話してほしい」や「もう一度説明してほしい」などのメッセージを Leap Motion で入力し、教師に伝えるということをめざす。この時、話し手には話の邪魔をしないことや話し手が応答があるようにすることを前提とし、これを満たすアプリケーションの提案を目指す。

第2章 関連研究

ここでは対話的授業運営の関連研究としてクリッカーを、意思決定ツールとしてサイボウズ Live を取り上げる。

2.1 クリッカー [2]

双方向性授業の実現としてクリッカー [4] を用いた事例がある。クリッカー (授業応答システム) とは赤外線リモコンによる学生解答システムである。授業中に教師が問題を出し、学生がクリッカーを用いて解答するというものである。クリッカーを用いた結果として、「役に立つ」という問いに対して「役に立つ」と答えた学生が 6 割を上回り、「勉強する気になったか」という問いに対しては「やる気が出た」という学生が 7 割を上回っている。

2.1.1 提案するシステムとの違い

クリッカーは教師が出した問題に対して学生が解答するためのツールであり、学生側がメッセージを発信するツールではない。先生からの選択肢に答えるだけでなく、授業に対するストレスや評価などを自然な形で発信することが重要だと考えた。

2.2 サイボウズ Live [3]

サイボウズ Live(2.1) はサイボウズ社が開発した意思決定ツールである。主な機能として、以下の 5 つが紹介されている。

- タイムライン：メンバーに聞きたいこと、伝えたいことを書く
- 掲示板：掲示板のコメント欄でメンバーとディスカッションする
- イベント：プロジェクトのイベントやスケジュールを登録・共有できる
- ToDo リスト：メンバーと ToDo を共有できる
- 共有フォルダ：1 グループにつき 1G バイトまで無料でファイルを保存可能

また、チャット機能の利点は以下の 3 つが紹介されている。

1 つ目は、PC でもスマートフォンでもメッセージがすぐ届く点である。サイボウズ Live のチャット機能は PC だけでなく、スマートフォンでも使うことができる。投稿したコメントは相手の画面にただちに用事されるので、オンラインのミーティングがスムーズに行える。

2つ目は、複数人でも1対1でもチャットが出来る点である。複数人のメンバーを指定して、チャットスペースを作成できる。また、サイボウズ Live 上で知人と繋がると、自動的に1対1のチャットスペースが作成されるので、簡単にチャットのやりとりをスタートできる。

3つ目は、グループ機能と併用すれば、情報の整理が楽になるという点である。チャットのやりとりで決定したことは、先に記述した ToDo リストに登録しておくことができる。



図 2.1: サイボウズ Live

2.2.1 提案するシステムとの違い

本システムとの違いは1つは入力方法である。提案するシステムは入力装置として Leap Motion を使っている。対してサイボウズ Live はスマホや PC を使って入力する。キーボードやマウスを使って入力するものであり、手や指の動きで入力操作はできない。

第3章 提案するシステムの概要

提案するシステムは、授業やゼミ等で聞き手が話者にメッセージを発信するための意思決定ツールである。

3.1 提案するシステムの特徴

このアプリケーションの主たる特徴は、2つある。

1つ目は、Leap Motion で読み取る 4つのジェスチャーを使い、リアルタイムに入力操作を行うことである。

2つ目は、クライアント側とサーバー側の二つのプログラムに分かれていることである。このアプリケーションでは、クライアント側とサーバー側間のデータを送受信する手段として、ソケット通信を行った。

このアプリケーションの機能は以下のものがある。

- ジェスチャーを使ってメッセージを発信する機能
- 発信するメッセージに重みをつける機能
- 発信されたデータを集計する機能

3.2 クライアント側のプログラムの役割

クライアント側は、聞き手が発信したいメッセージを発信するためのものである。

メッセージを発信する手順

クライアント側がメッセージを発信する際、全ての手順をジェスチャーで行うという特徴がある。このように製作した理由は、マウスやキーボードなどの入力装置との差別化を行うためである。聞き手がメッセージを送信するために行う手順は、以下の3つある。

1. ジェスチャーを行い、入力方法を定める (図 3.1).
2. ジェスチャーでメッセージの選択を行う (図 3.2).
3. ジェスチャーの回数で、メッセージに重みをつける (図 3.3).

まず、画面遷移 1 でジェスチャーを行い、入力方法を定める。例えば、このときサークルジェスチャーを行った場合、このジェスチャーがメッセージを発信するまでの入力方法にな

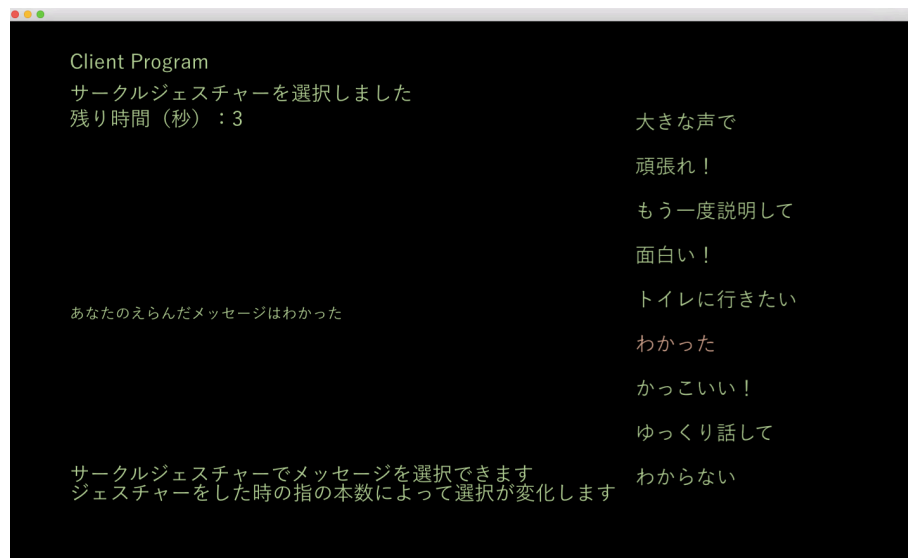


図 3.2: 画面遷移 2

る。次に、ジェスチャーでメッセージの選択を行う。30 秒間の時間が設けてあり、ユーザーはこの時間内に自分が発信したいメッセージを選択する。最後に選択されていたメッセージがサーバー側へ送信する値になる。最後に、ジェスチャーでメッセージに重みをつける。この段階でも 30 秒の値が設けてある。ジェスチャーを行った回数で、自分がどれだけメッセージを伝えたいか発信するメッセージに重みをつけることができる。この回数が次に紹介する各ジェスチャーの値に相当し、値の大小でサーバー側プログラムの描写が変わる。

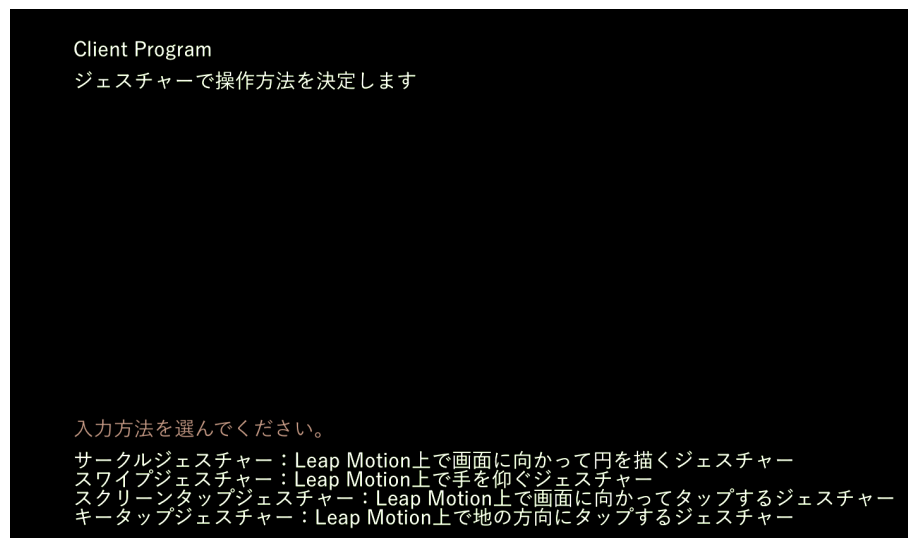


図 3.1: 画面遷移 1

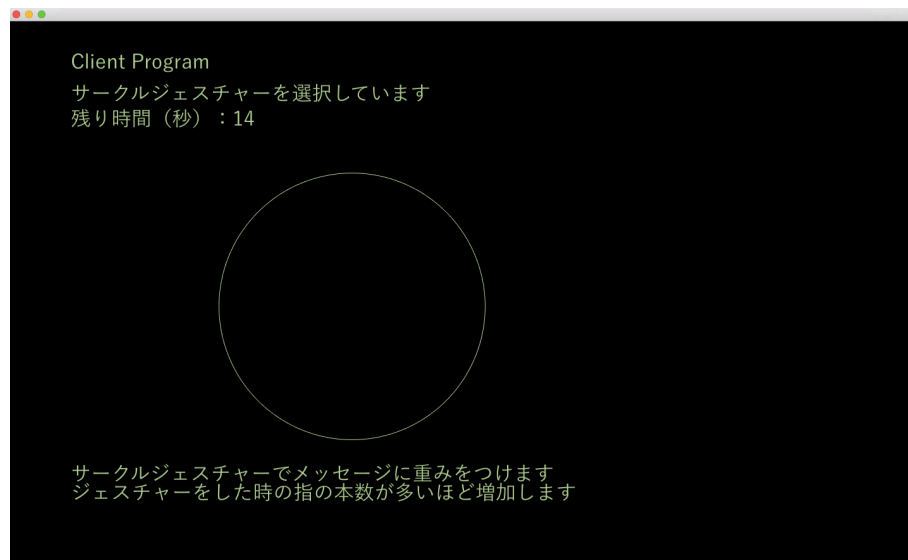


図 3.3: 画面遷移 3

クライアント側から発信する値について

- アカウントナンバー
- Leap Motionにかざされている手の数
- サークルジェスチャーの回数
- スワイプジェスチャーの回数
- スクリーンタップジェスチャーの回数
- キータップジェスチャーの回数
- メッセージナンバー

発信される値は全部で7つある, これらの値は, カンマで区切られた1つの文字列になり, サーバー側へ発信される (図 3.4).

まず, アカウントナンバーはユーザーを特定するための固有な値であり, それぞれの聞き手には予め番号が振り分けられている. これにより誰がどのような値を送ったかを特定しサーバー側で処理する.

次に, 手の数は聞き手が Leap Motion に手をかざしているかの判定に使う. 発信されている値が0の時は, 手をかざしていないことを発信し, 0よりも大きい値の時, 聞き手が手をかざしていることを表す.

各ジェスチャーの回数は, メッセージに重みをつける際に, 聞き手が行ったジェスチャーの回数である. 発信された値は, サーバー側で処理され, 描写に反映される (図 3.5 参照). 最後にメッセージナンバーは, 選択したメッセージを表す. 発信するメッセージは9つ用意されており, それぞれ対応した値が発信される. メッセージナンバーは-1と0から8の値

で構成されており, -1 のときは選択されていないことを表し, 0 から 8 の値ではそれぞれのメッセージ送られる.

Leap Motion で読み取る値と発信するメッセージについて

Leap Motion で行うことはジェスチャーの判定とジェスチャーのカウントである. Leap Motion では, ユーザーが行ったジェスチャーの種類と, ジェスチャーの状態を判定することができる. ジェスチャーの状態とは, ジェスチャーが始まった時, ジェスチャーをしている時, ジェスチャーが終わった時のことを指す. メッセージの選択の際は, 以下の手順で行う.

クライアント側プログラムの値の発信について

発信する値 (例)	
アカウントナンバー	1
手の数	1
サークルジェスチャーの回数	6
スワイプジェスチャーの回数	-1
スクリーンタップジェスチャーの回数	-1
キータップジェスチャーの回数	-1
メッセージナンバー	6

1. カンマを入れ, 一つの文字列にする

1, 1, 6, -1, -1, -1, 6

入力操作で選択されなかったジェスチャーの値は -1 になる

2. サーバー側へ発信する

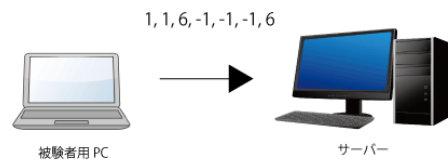


図 3.4: クライアント側から発信する値

3.3 サーバー側のプログラムの役割

サーバー側プログラムの描写 (図 3.5) はのように行われる. クライアント側プログラムから送られてきた値を集計したものが円の大きさや色で反映される. 以下, それらの役割を記述する.

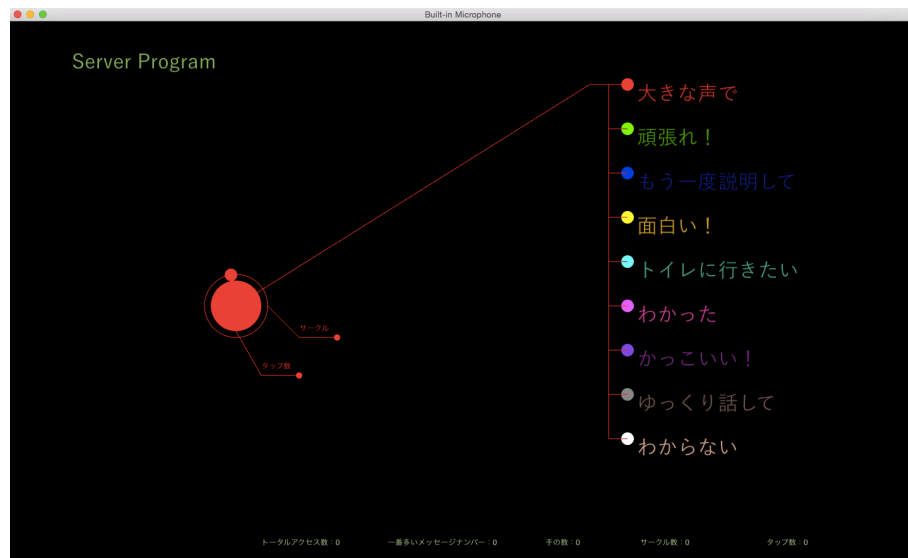


図 3.5: サーバー側プログラムの描写

描写処理について

図 3.6 は集計結果を表す円である。円の色は送られてきたメッセージの多かったものの色に変化する。変化する色は9つあり、それぞれの意味を図 3.7 が示すメッセージの色に対応する。円の直径は発信されたタップするの数によって変化する。クライアント側からメッセージの送信時に送られてきたタップの回数が多いほど、円は大きくなり、メッセージの重みを示す。円の外側の小さな円は、内側の円の円周に沿って回転する。クライアント側からメッセージの送信時に送られてきたサークルジェスチャーの回数が多いほど、円周を周回する速さが速くなり、メッセージの重みを示す。

また、図 3.8 のマリオネットは図 3.6 の下面に表示され、接続されている人数を表す。このシステムは最大で7人が接続されるようになっているので、最大で7体のマリオネットが表示される。

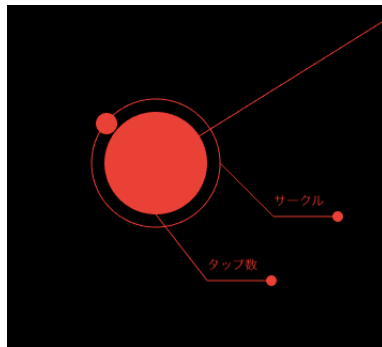


図 3.6: 集計結果を表す円

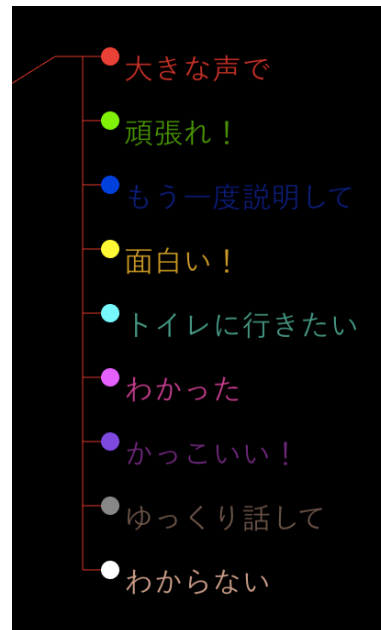


図 3.7: 円の色が示すメッセージ

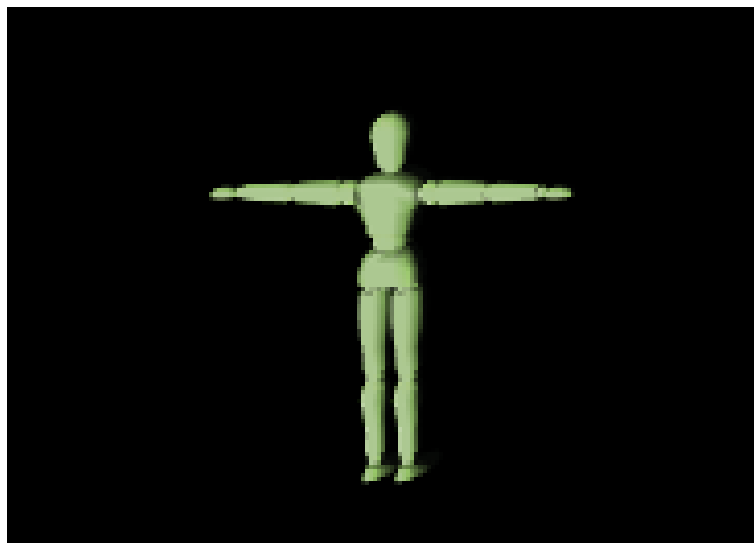


図 3.8: 接続されている人数を示すマリオネット

メッセージナンバーの集計

サーバー側では、聞き手のメッセージを集計し、その集計結果を発信するためのものである。まず、送られてきたメッセージを区切りで分解する。その値を保存する。送られてきたメッセージは保存後、どのメッセージが多く発信されたかを算出する。発信されてくる

メッセージナンバーは-1 から 8 の値であり, -1 のときは代入されない. これにより聞き手が発信した最新のメッセージナンバーが保存される. 一番多く受け取ったメッセージナンバーを算出後, その値にあった描写処理を行う.

3.4 実装方法

3.4.1 言語・ツール・開発環境

使用言語は主に C++ である. 一部, C 言語での記述もある. IDE は Xcode で Leap Motion を入力装置として使う. また開発環境は表 3.1 の通りである.

表 3.1: 開発環境	
OS	OS X Yosemite Macbook Air
プロセッサ	1.6 GHz Intel Core i5
メモリ	4GB 1600 MHz DDR3

3.4.2 Cinder フレームワークについて

Cinder は C++ で書かれたのフレームワークである. 画像や動画, 音声処理に長けており, ライブラリが用意されている. 今回は, Cinder を使ったの導入事例が多くなってきていることから, Cinder を使って開発を進めることにした.

openFrameworks との違い

ここでは openFrameworks[] について述べる. openFrameworks は C++ のオープンソースツールキットである. openFrameworks は, Cinder と同様に, 描写処理等に長けたフレームワークでもあり, Leap Motion を使ったアプリケーションで使われるフレームワークとして, この 2 つが使われていることが多い. openFrameworks は Cinder よりも歴史がある. openFrameworks は 2004 年に開発され広まっていったのに対し. Cinder は 2010 年から広まった. goolge トレンド (図 3.9) では, 近年, Cinder の方が openFrameworks よりも広く利用されてきていることがわかる.

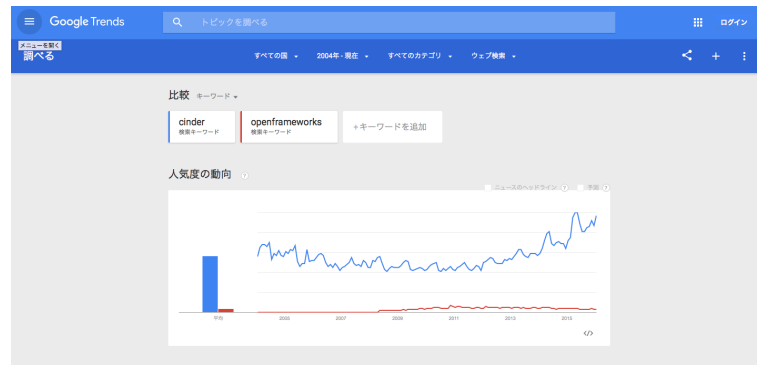


図 3.9: Cinder と openFrameworks

3.4.3 Leap Motion について

Leap Motion(図 3.10)とはアメリカの Leap Motion 社が 2013 年にリリースした手や指を検知することに特化したセンサである. Leap Motion は教育に対応するものや医療など幅広く使われている. 開発の可能な言語としては, Java や C++などの 6 つが用意されている (図 3.11 参照).

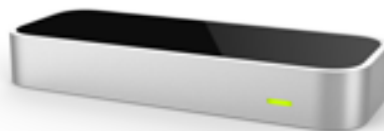


図 3.10: Leap Motion



図 3.11: 開発の可能な言語

Leap Motion の評価

Leap Motion の入力装置としての特性を追求した．ここでは Leap Motion の良い点と欠点をあげる．Leap Motion の良い点は，3 つある．

1 つ目は，Leap Motion では手や指の位置や方向，ジェスチャーなどを検知し，リアルタイムで入力操作ができるという点である．ジェスチャーは 4 種類あり，以下に示す．先にも述べたが今回のアプリケーションではサークルジェスチャーとスクリーンタップジェスチャーの 2 種類を検知する．

- サークルジェスチャー：スクリーンに向かって円を描くジェスチャー (図 3.12)
- スクリーンタップジェスチャー：スクリーンに向かってタップするジェスチャー (図 3.13)
- スワイプジェスチャー：スクリーン方向にスワイプするジェスチャー (図 3.14)
- キータップジェスチャー：Leap Motion 方向にタップするジェスチャー (図 3.15)

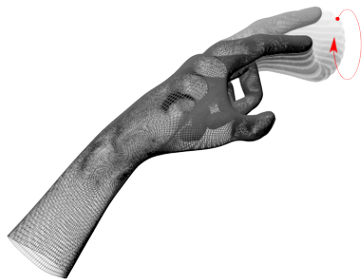


図 3.12: サークルジェスチャー

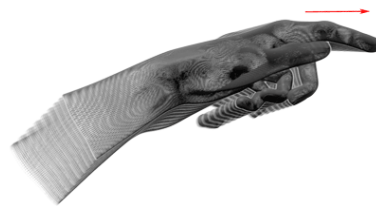


図 3.13: スクリーンタップジェスチャー



図 3.14: スワイプジェスチャー



図 3.15: キータップジェスチャー

2つ目は、指の曲げを判定できるので、入力パターンが豊富であることがいえる。Leap Motion はどの指が曲げられているか、曲げられている本数は何本か、どのくらい曲げられているか判定することができる。また、右手と左手の判定もできるので、以上のことを使えば入力パターンを増やすことが可能である。

3つ目は、Leap Motion をキーボードやマウスなどの入力装置と比較したとき、音を出さずに入力できるという利点がある。キーボードやマウスにはボタンがあり、押すとカチッと音になる。しかし、Leap Motion は指の曲げやジェスチャー等が入力操作として扱われるので音が出ることはない。

一方で、Leap Motion の欠点として次の2つのことを述べる。

1つ目は、特定の方向ではセンシングが難しいことである。Leap Motion は赤外線センサーで感知しており、図3のような逆四角錐の範囲で検知される。そのため手を地と垂直に向けると手のひらで指を隠してしまうため、誤検知される危険性がある。よって、意識して地と平行にかざすことでLeap Motion が誤操作が発生しにくいことがわかった。

もう1つは、複数接続が不可能であることである。1台のPCに接続できるのは1つのLeap Motion で、2つ以上繋いだ場合、1台目のLeap Motion が優先され、2台目は機能しないことがわかった。

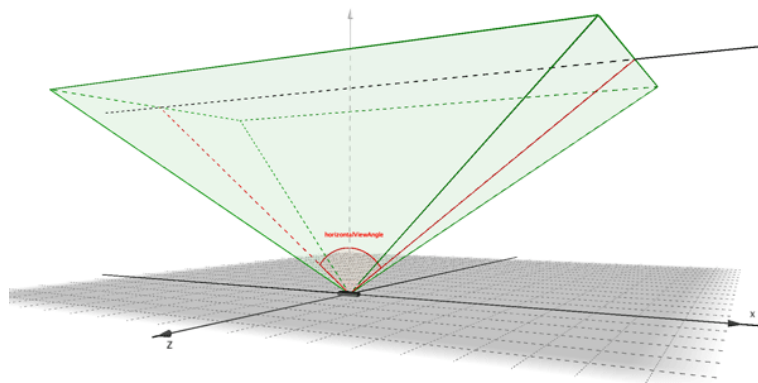


図 3.16: Leap Motion のセンサ範囲

クライアント側プログラムからサーバー側プログラムへ値を送受信する手段としてソケット通信を行った。

第4章 実験と評価

4.1 実験の目的

本実験の目的は提案するアプリケーションが話し手にどのようなアクションを起こすのか, 話し手の邪魔になっていないかどうかを調査するために行った.

4.2 方法

4.2.1 日時と場所

本実験は1月27日, 公立はこだて未来大学の美馬研究室で行った.

4.2.2 被験者

被験者は公立はこだて未来大学の大学生6人で, 男性が3人, 女性が3人である.

4.2.3 道具

本実験で使用した道具は表4.1のとおりである.

表 4.1: 実験器具	
道具名	台数
Macbook	6 台
Leap Motion	6 台
ビデオカメラ	1 台

4.2.4 手順

本実験を行うにあたり, 以下の手順で行うことを被験者に説明した.

1. Leap Motion が使用できることを確認する (付録2 参照).
2. visualiser で一度手が検知されていることを確認する (付録2 参照).
3. 被験者は話し手と聞き手に分かれる.

4. 実験者は被験者全員にクライアント側のプログラムを配布する.
5. 実験者はサーバー側のプログラムを起動する.
6. 聞き手ははぜミが始まると同時にプログラムを起動する.
7. 両方のプログラムの起動を確認後, 話し手が進捗報告を行う.
8. 被験者には伝えたいメッセージがあるときにアプリケーションを使って発信してもらう.
9. 話し手の進捗報告が終わると, 聞き手は話し手に質問やコメントを述べる.
10. 被験者はアンケートに回答する.

4.2.5 アンケート

アンケートは5段階評価の質問を8つと自由記述を2つ設けた.

5 段階評価のアンケート

1. 自分のメッセージが発信されたことについてわかりやすかったと思う
2. サーバー側プログラムの変化がわかった
3. 表示されているものの役割・意味についてまで理解することができたと思う (クライアント側)
4. 表示されているものの役割・意味についてまで理解することができたと思う (サーバー側)
5. 話し手の進捗報告は十分に理解できたと思う
6. メッセージの発信後, またはサーバー側プログラムの表示の変化後, 話し手にリアクションがあったと思う
7. アプリケーションは話し手の邪魔をしていたと思う
8. メッセージの候補の数は適当だと思う

自由記述

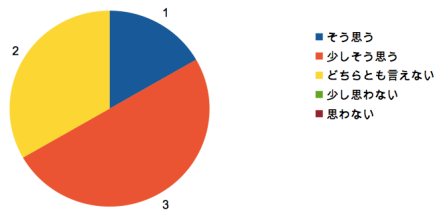
1. 他にも話し手に伝えたいメッセージがあれば教えてください
2. 意見や感想, アドバイスなど

4.2.6 アンケート結果

ここではアンケートの集計結果と自由記述で書かれていたことを記す.

アンケート集計結果

自分のメッセージが発信されたことについて、
分かりやすいものであったと思う



サーバー側プログラムの表示の変化がわかった

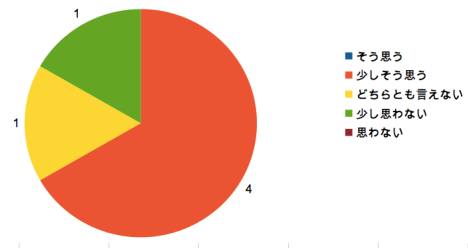
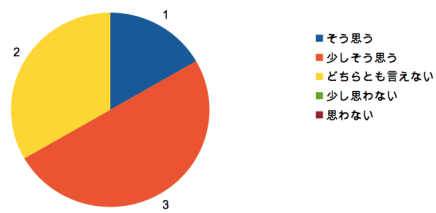


図 4.1: 自分のメッセージが発信されたこと
についてわかりやすかったと思う

図 4.2: サーバー側プログラムの変化
がわかった

表示されているものが示す役割・意味についてまで理解
することができたと思う (クライアント側)



表示されているものが示す役割・意味についてまで理解
することができたと思う (サーバー側)

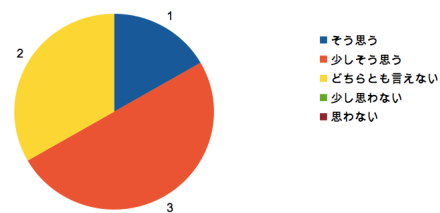


図 4.3: 表示されているものの役割・意味
についてまで理解することができた
と思う (クライアント側)

図 4.4: 表示されているものの役割・意味
についてまで理解することができた
と思う (サーバー側)

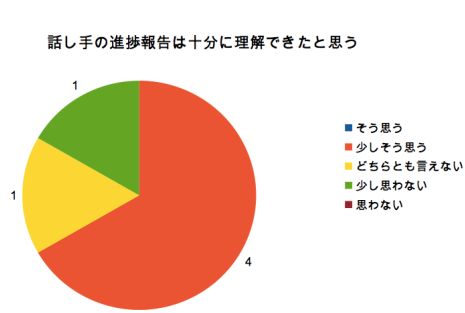


図 4.5: 話し手の進捗報告は十分に聞けたと思う

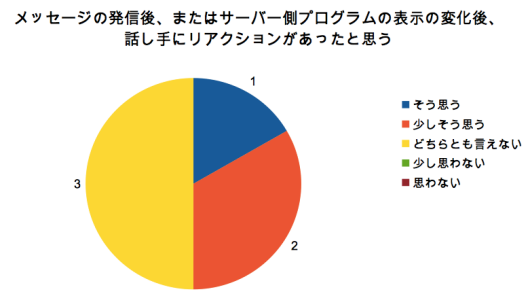


図 4.6: メッセージの発信後、またはサーバー側プログラムの表示の変化後、話し手にリアクションがあったと思う

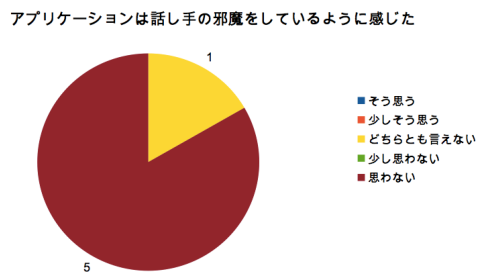


図 4.7: アプリケーションは話し手の邪魔をしているように感じた

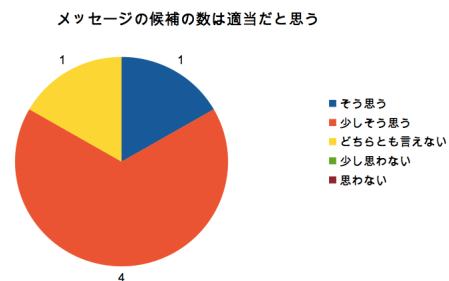


図 4.8: メッセージの候補の数は適当だったと思う

図 4.1 より、自分のメッセージが発信されたことについてわかりやすかったと思うという問いに対して、4 名が思うという方に回答をし、2 名がどちらとも言えないと答えた。

図 4.2 より、サーバー側プログラムの変化がわかったという問いに対して、4 名がわかったという方に回答をし、1 名の人がどちらとも言えない、1 名がわからなかったと答えた。

図 4.3 より、表示されているものの役割・意味についてまで理解することができたと思う (クライアント側) という問いに対して、4 名の人が思うという方に回答をし、2 名の人がどちらとも言えないと答えた。

図 4.4 より、表示されているものの役割・意味についてまで理解することができたと思う (サーバー側) という問いに対して、4 名の人が思うという方に回答をし、2 名の人がどちらとも言えないと答えた。

図 4.5 より、話し手の進捗報告は十分に理解できたと思うという問いに対して、4 名の人が思うという方に回答をし、2 名の人がどちらとも言えないと答えた。

図 4.6 より、メッセージの発信後、またはサーバー側プログラムの表示の変化後、話し手にリアクションがあったと思うという問いに対して、3 名の人が思うという方に回答をし、3 名の人がどちらとも言えないと答えた。

図 4.7 より、アプリケーションは話し手の邪魔をしていたと思うという問いに対して、5 名の人がそう思わないという方に回答をし、1 名の人がどちらとも言えないと答えた。

図 4.8 より、メッセージの候補の数は適当だと思うという問いに対して、5 の人が思うという方に回答をし、1 名の人がどちらとも言えないと答えた。

自由記述の結果

問い 1：他にも話し手に伝えたいメッセージがあれば教えてください

- 講義などのシーンに関しては現状で良いと思います。
- スライドが見易い。
- 話すスピードが速い。
- 伝えたいものというより、30 秒待たないと発信できないもどかしさがあった。メッセージの種類は丁度良かったと思う。
- メッセージの種類は丁度良いと思います。
- おなかがすいたとか休憩したいとか送りたいかった。
- トイレに行きたい、おなかがすいたなどを全部まとめて「休憩したい」にしても良いように感じた。

問い 2：意見や感想、アドバイスなど

- メッセージ選択までの時間が 30 秒だと少し長いように感じました。

- 前回と比較して、メッセージを決定するまでの動作が簡単になったため、もう少し選択時間が短くても良いのかな、と感じました。でも、選択した瞬間にサーバー側にすぐに行っているみたいなので、その点はリアルタイムなコメントがもらえて良いと思いました。
- 2つ目の画面の円の大きさを送る場面は、時間が長すぎる。15秒くらいでいいと思う。
- 入力画面の時間制限を目立つ形で表示すると、誤送信を防げると思う。
- かっこいいがくると、とても嬉しかった。色でメッセージが判断できるのは、わかりやすく良いと思います。
- 真ん中の円の視認性をあげた方がいい。発表者は自分の発表に集中しながらその画面を見るわけだから、視野に入った時点で、聞き手が何を合図したかわかるように円を大きくしたり工夫をするといいかも。
- 私のPCがみんなと画面サイズが異なるせいか、下のメッセージ(小さく書いていた文字)が全て見えなかった。そのため、なにが送られたか謎だった。
- 選択したメッセージを取り消す方法がわからなかった。
- サーバー画面(話し手が見るべき画面)の置き場によって、結果が変わってしまうように感じた。話し手と話し手が使うモニターの間にサーバー画面を置いた方がもっとこのツールの良さが出てくると感じた。サーバー画面は話し手だけではなくゼミ生みんなに見えてもおもしろいかも！
- わかったとかっこいいの色の区別がしにくかった。
- 以前よりも格段に操作がしやすかった
これをゼミ生みんなで使えたらゼミがもっと充実したものになると思う
- 自分の意思で「送信」できないのは「自分のメッセージが発信されたことについて、分かりやすいもの」かと考えたときに分かりづらいと思う。
- あと、時間制限があるためメッセージの選択権利と重要性が自分の意思と反して伝わってしまう場合が多いと感じられた

- 話し手が見る画面. Help が欲しい
- UI はシンプルでいいと思う. 逆に伝わりづらくなってる部分もある. (例えば, サーバー画面で発信されたメッセージは色でわかりやすいがメッセージの重さがシステムを知らない人からしたら味方がわかりにくいと感じた.)

第5章 考察

5.1 話し手のアプリケーションに対するリアクション

本実験のアンケート結果から、話し手のアプリケーションに対するリアクションは半数の人があったと答え、半数がどちらともいえないということがわかった。この結果に関して、実際の実験の様子から、話し手がアプリケーションを見るのは説明しているスライドや聞き手の顔を見る時間と比べて極めて少ない時間であり、送られてきたメッセージ全てを確認できなかったのではないかと考えられる。また、発信されてきたメッセージに気付いても、瞬時に対応するのが難しかったとも考えられる。自由記述の部分では、かっこいいなどの話し手を賞賛するメッセージが発信された時、話し手が嬉しくなることがわかった。

5.2 話し手の邪魔になっていないか

本実験のアンケート結果から、話し手の邪魔になっていないかという問いに対して、5名が思わないと答え、1名がどちらとも言えないということがわかった。このことから概ね話し手に与えるストレスは少ないということがいえる。ただ色の判別や文字の大きさなど、視認性をあげることで話し手がメッセージに気づく可能性が上がるのではないかと考えられる。

第6章 結論と今後の展開

6.1 まとめ

本研究では、授業中や大学のゼミ中を想定し、聞き手側から Leap Motion を操作し、話し手側にメッセージを発信するためのアプリケーションを開発した。これにより、聞き手が話し手にメッセージを送信することで、話し手がリアクションを起こし、対話的な授業運営、ゼミの進行を目指すことを目的とした。実験結果から聞き手から話し手にメッセージの発信した際、話し手がリアクションを行うことや、それが話し手にストレスを与えるものではないということがわかった。また、視認性をあげることやメッセージを発信するまでの時間について課題がみられた。

6.2 今後の方針

本研究では、実験の結果から視認性に対して課題があると感じた。文字の色や大きさなどである。これを改善することで話し手がメッセージに気づく可能性が上がり、聞き手が発信したメッセージに対して、話し手からの良いリアクションが得られると考えられる。

謝辞

本研究を進めるにあたり, ご指導いただいた美馬義亮先生に感謝いたします. また, アドバイスやご指摘をしてくれた美馬研究室の学生の皆様, 実験に協力してくださった学生の皆様に感謝申し上げます.

参考文献

- [1] 学制百年史, 明治初期の高等教育, http://www.mext.go.jp/b_menu/hakusho/html/others/detail/1317598.
- [2] 武田直人 田口忠緒, クリッカー (授業応答システム) を用いた双方向性授業の比較と評価: 学生中心学習の構築を目指して, 名城大学薬学部薬学教育開発センター, <http://www.keepad.com/jp/casestudies/docs/Comparison%20and%20Evaluation%20of%20Clikers%20> 2012.
- [3] 株式会社サイボウズ, ”サイボウズ live”, <https://live.cybozu.co.jp/overview.html>, 参照 2016 年 1 月 25 日.
- [4] 中村薫, 「Leap Motion プログラミングガイド 改訂版」, 2015.
- [5] Leap Motion SDK, 2015.

付録その1 アンケート

Leap Motionによる教育をインタラクティブにするアプリケーションの提案

公立はこだて未来大学システム情報科学部情報アーキテクチャー学科4年の一ノ瀬智太です。
現在、私の提案するシステムについてのアンケートを取らせて頂いております。

ご回答頂きましたデータについては、今回の卒業論文のみで利用することとし、
それ以外に活用することはありませんのでご安心下さい。

***必須**

性別 *

- ☐ 男
☐ 女

あなたの役割 *

- ☐ 話し手
☐ 聞き手

自分のメッセージが発信されたことについて、分かりやすいものであったと思う

1 2 3 4 5

そう思う ☐ ☐ ☐ ☐ ☐ そう思わない

サーバー側プログラムの表示の変化がわかった *

1 2 3 4 5

よくわかった ☐ ☐ ☐ ☐ ☐ 全くわからなかった

図 1: 使用したアンケート 1

表示されているものが示す役割・意味についてまで理解することができたと思う(クライアント側)
★

1 2 3 4 5

そう思う ○ ○ ○ ○ ○ そう思わない

表示されているものが示す役割・意味についてまで理解することができたと思う(サーバー側) ★

1 2 3 4 5

そう思う ○ ○ ○ ○ ○ そう思わない

話し手の進捗報告は十分に理解できたと思う ★
(文字の大きさ、色、円や線の大きさ etc...)

1 2 3 4 5

そう思う ○ ○ ○ ○ ○ そう思わない

メッセージの発信後、またはサーバー側プログラムの表示の変化後、話し手にリアクションがあったと思う ★

1 2 3 4 5

そう思う ○ ○ ○ ○ ○ そう思わない

アプリケーションは話し手の邪魔をしているように感じた ★

1 2 3 4 5

そう感じた ○ ○ ○ ○ ○ そう感じなかった

図 2: 使用したアンケート 2

アプリケーションは話し手の邪魔をしているように感じた *

1 2 3 4 5

そう感じた ☐ ☐ ☐ ☐ ☐ そう感じなかった

メッセージの候補の数は適当だと思う *

1 2 3 4 5

そう思う ☐ ☐ ☐ ☐ ☐ そう思わない

他にも話し手に伝えたいメッセージがあれば教えてください

意見や感想、アドバイスなど

こうすれば良くなると思う、ここが伝わりにくい、こういう使い方はどうか etc...

送信

Google フォームでパスワードを送信しないでください。

100%: 完了しました。

図 3: 使用したアンケート 3

付録その2 Leap Motionを使うための準備

Leap Motion を使用するために

Leap Motion を使用するために、まず自身の PC にインストーラがインストールされていなければならない。ここではその手順を示す。

Leap Motion を使うためには、3つの手順が必要である。

1. インストーラを公式ホームページのセットアップ (図 4) からダウンロードする。
2. ダウンロードしたものを解凍する。図 5 のような画面が表示されるので、指示に従ってインストールする。
3. メニューバーにアイコン (図 6) が表示されていたら完了。

Leap Motion を使用する前の確認事項

Leap Motion のアプリケーションを使用する前に確認しておくべきことを2つあげる。以下に述べることは Leap Motion のアイコン (図 6) から設定を選択し Leap Motion の設定画面 (図 7) から確認することができる。これらの設定後、Visualizer 等で確認するとなお良い。

1つは、操作の高さを調節しておくことである。Leap Motion では自動調節機能があり、操作の高さが可変になるが、初めて Leap Motion 使うとこれに気づきにくい。実際に使ってもらい、手が疲れるという声があった。ユーザーによって使いやすさに違いが出るので確認しておくべきである。

もう1つは、道具の設定がオフになっているかどうかを確認することである。Leap Motion ではペンなど棒状の道具を検知できる。今回のアプリケーションでは、道具を使って操作することを想定しなかったため、道具の検知を行っていない。もし、道具を使ってのアプリケーションを製作する場合は、チェックされていないと検知できないため確認する必要がある。

Leap MotionSDK のダウンロード

Leap Motion を使ったアプリケーションを開発するために SDK を自分の PC にインストールする必要がある。Leap MotionSDK も Leap Motion インストーラーと同様に、公式ホームページのセットアップ (図 4) よりダウンロードできる。



図 4: Leap Motion のセットアップ

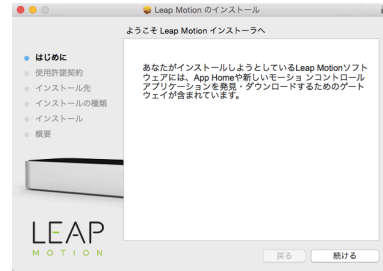


図 5: インストール画面

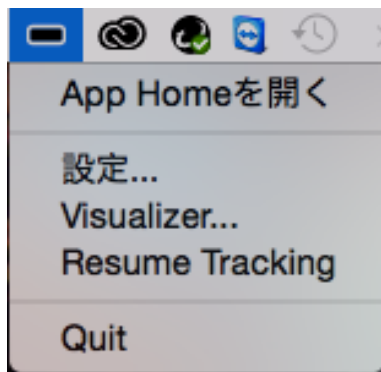


図 6: Leap Motion のアイコン

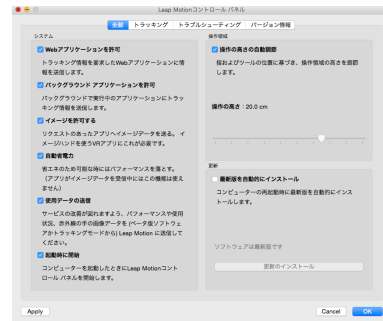


図 7: Leap Motion の設定画面

目 次

2.1	サイボウズ Live	3
3.2	画面遷移 2	5
3.1	画面遷移 1	5
3.3	画面遷移 3	6
3.4	クライアント側から発信する値	7
3.5	サーバー側プログラムの描写	8
3.6	集計結果を表す円	9
3.7	円の色が示すメッセージ	9
3.8	接続されている人数を示すマリオネット	9
3.9	Cidner と openFrameworks	11
3.10	Leap Motion	11
3.11	開発の可能な言語	11
3.12	サークルジェスチャー	12
3.13	スクリーンタップジェスチャー	12
3.14	スワイプジェスチャー	12
3.15	キータップジェスチャー	12
3.16	Leap Motion のセンサ範囲	13
4.1	自分のメッセージが発信されたこと についてわかりやすかったと思う	16
4.2	サーバー側プログラムの変化 がわかった	16
4.3	表示されているものの役割・意味 についてまで理解することができた と思う (クライアント側)	16
4.4	表示されているものの役割・意味 についてまで理解することができた と思う (サーバー側)	16
4.5	話し手の進捗報告は十分に聞けた と思う	17
4.6	メッセージの発信後, またはサーバー側プログラムの表示の変化後, 話し手 に リアクションがあったと思う	17

4.7	アプリケーションは話し手の邪魔	
	をしているように感じた	17
4.8	メッセージの候補の数は適当だったと思う	17
1	使用したアンケート 1	25
2	使用したアンケート 2	26
3	使用したアンケート 3	27
4	Leap Motion のセットアップ	29
5	インストール画面	29
6	Leap Motion のアイコン	29
7	Leap Motion の設定画面	29

表 目 次

3.1 開発環境	10
4.1 実験器具	14