

## 修士論文の和文要旨

|        |  |      |         |
|--------|--|------|---------|
| 研究科・専攻 | 大学院 情報理工学研究科 情報・ネットワーク工学専攻 博士前期課程  |      |         |
| 氏名     | 石井 拓斗  | 学籍番号 | 1731019 |
| 論文題目   | Leap Motion を用いた VR 麻雀での面白さとイカサマシステムの評価  |      |         |
| 要旨     | <p>近年、Virtual Reality(VR) 分野の発展が目覚ましいものがあり、特にエンターテインメント分野で顕著である。マウスやキーボード操作だけの古典的な 2D ゲームであっても VR 化することで面白さに繋がることが分かっており、麻雀ゲームのようなテーブルゲームも VR 化することで面白さが増す可能性がある。</p> <p>現実の麻雀ゲームではイカサマ行為が可能であり、技術的に美しいという面や、ドラマチックなゲーム状況を演出可能であることから、ゲームがより面白くなるなど非常に関心が高い。しかし通常のコンピュータゲームではイカサマ行為は不可能なことが多い。</p> <p>本研究では、VR で麻雀を行うために出来るだけ現実に近い手指の動作で麻雀牌を操作できるシステムを構築した。Leap Motion を用いた麻雀牌の操作方法やゲームに必要な他の操作を実装した。牌同士の物理演算をリアルにし過ぎると操作しにくい場面があることから、牌を手牌に持ってくる時、捨てる時、牌を積む時、など状況に応じて、手指のどの指を使うかや牌の回転を制限するなどの機能を組み合わせた。さらに、イカサマの一つであるすり替えに関し、現実よりも容易に行えるシステムを構築した。一般の人は現実ではイカサマが実現できないが、VR 空間でサポートすることによりイカサマを少し簡単に行うことが出来るようになり、結果的に緊張感を高めより面白くなることを目指した。</p> <p>Leap Motion による操作性とイカサマサポートシステムに関する評価実験を行った。麻雀牌の操作方法は、VRChat 内で既存の麻雀ゲームを模したものと比較した。この結果、操作時間や操作の快適性については本システムの方が優れていたが、操作の正確性は劣る結果となった。これは選択した牌を光らせるなどの補助的効果により改善可能であると考えられる。また、イカサマサポートシステムのバランスを探る評価実験では、位置精度を Easy, Normal, Hard の三つの難易度に分け、更に制限時間を三種類用意することで、熟練度と難易度によってどのような組み合わせがバランス良いかを検討した。この結果、位置精度は Normal 難易度、制限時間は 4 秒間程度が適していると判断した。友人同士で敢えてイカサマをやり合って楽しむような目的の際には、もう少し難易度を下げることで対応出来る。</p> |      |         |

平成30年度 修士論文

Leap Motion を用いた VR 麻雀での面白さと  
イカサマシステムの評価

電気通信大学大学院 情報理工学研究科  
情報・ネットワーク工学専攻

学籍番号 1731019

氏名 石井 拓斗

指導教員 成見 哲

副指導教員 寺田 実

平成31年1月28日

## 概要

近年、Virtual Reality (VR) 分野の発展が目覚ましいものがあり、特にエンターテインメント分野で顕著である。マウスやキーボード操作だけの古典的な 2D ゲームであっても VR 化することで面白さに繋がるのが分かっており、麻雀ゲームのようなテーブルゲームも VR 化することで面白さが増す可能性がある。

現実の麻雀ゲームではイカサマ行為が可能であり、技術的に美しいという面や、ドラマチックなゲーム状況を演出可能であることから、ゲームがより面白くなるなど非常に関心が高い。しかし通常のコンピュータゲームではイカサマ行為は不可能なことが多い。

本研究では、VR で麻雀を行うために出来るだけ現実に近い手指の動作で麻雀牌を操作できるシステムを構築した。Leap Motion を用いた麻雀牌の操作方法やゲームに必要な他の操作を実装した。牌同士の物理演算をリアルにし過ぎると操作しにくい場面があることから、牌を手牌に持ってくる時、捨てる時、牌を積む時、など状況に応じて、手指のどの指を使うかや牌の回転を制限するなどの機能を組み合わせた。さらに、イカサマの一つであるすり替えに関し、現実よりも容易に行えるシステムを構築した。一般の人は現実ではイカサマが実現できないが、VR 空間でサポートすることによりイカサマを少し簡単に行うことが出来るようになり、結果的に緊張感を高めより面白くなることを目指した。

Leap Motion による操作性とイカサマサポートシステムに関する評価実験を行った。麻雀牌の操作方法は、VRChat 内で既存の麻雀ゲームを模したものと比較した。この結果、操作時間や操作の快適性については本システムの方が優れていたが、操作の正確性は劣る結果となった。これは選択した牌を光らせるなどの補助的效果により改善可能であると考えられる。また、イカサマサポートシステムのバランスを探る評価実験では、位置精度を Easy, Normal, Hard の三つの難易度に分け、更に制限時間を三種類用意することで、熟練度と難易度によってどのような組み合わせがバランス良いかを検討した。この結果、位置精度は Normal 難易度、制限時間は 4 秒間程度が適していると判断した。友人同士で敢えてイカサマをやり合ってみようという目的の際には、もう少し難易度を下げることで対応出来る。

# 目次

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>はじめに</b>                                | <b>4</b>  |
| 1.1      | 研究背景                                       | 4         |
| 1.2      | 目的   | 4         |
| 1.3      | 本論文の構成                                     | 5         |
| <b>2</b> | <b>Head Mounted Display と Input Device</b> | <b>6</b>  |
| 2.1      | Head Mounted Display                       | 6         |
| 2.1.1    | HTC VIVE                                   | 6         |
| 2.1.2    | その他の HMD                                   | 7         |
| 2.2      | Input Device                               | 8         |
| 2.2.1    | Leap Motion                                | 8         |
| 2.2.2    | VIVE コントローラ                                | 8         |
| <b>3</b> | <b>麻雀用語</b>                                | <b>10</b> |
| <b>4</b> | <b>既存技術・既存研究</b>                           | <b>12</b> |
| 4.1      | VR 麻雀                                      | 12        |
| 4.1.1    | VRChat                                     | 12        |
| 4.1.2    | TANKI(単騎)                                  | 13        |
| 4.2      | イカサマゲーム                                    | 15        |
| 4.2.1    | イカサマ麻雀                                     | 15        |
| 4.2.2    | 咲-Saki-                                    | 15        |
| 4.2.3    | いかさまゴキブリ                                   | 16        |
| 4.3      | Leap Motion を使った入力                         | 17        |
| <b>5</b> | <b>開発したシステム</b>                            | <b>20</b> |
| 5.1      | システム全体像と使用した機材                             | 20        |
| 5.1.1    | Unity                                      | 20        |
| 5.1.2    | Head Mounted Display                       | 20        |
| 5.1.3    | Leap Motion                                | 20        |
| 5.1.4    | Leap Motion の HMD への装着                     | 21        |
| 5.2      | 牌操作の仕組み                                    | 21        |
| 5.2.1    | オブジェクトの保持方法の改良                             | 22        |

|          |                             |           |
|----------|-----------------------------|-----------|
| 5.2.2    | オブジェクト毎の持ち方の変更              | 22        |
| 5.2.3    | 複数の異なるオブジェクトの保持             | 23        |
| 5.3      | イカサマ操作                      | 25        |
| 5.4      | その他の牌操作                     | 28        |
| 5.4.1    | 洗牌                          | 28        |
| 5.4.2    | 牌山作り                        | 28        |
| 5.4.3    | 手牌                          | 30        |
| 5.4.4    | 河                           | 31        |
| 5.5      | 実装したシステムの問題点                | 32        |
| <b>6</b> | <b>システムの評価</b>              | <b>33</b> |
| 6.1      | 牌操作の評価                      | 33        |
| 6.1.1    | 実験内容                        | 33        |
| 6.1.2    | VRChat 内での麻雀牌の操作方法          | 34        |
| 6.1.3    | 評価結果                        | 36        |
| 6.1.4    | 考察                          | 40        |
| 6.2      | イカサマシステムの評価                 | 44        |
| 6.2.1    | 実験内容                        | 44        |
| 6.2.2    | 評価結果                        | 46        |
| 6.2.3    | 考察                          | 50        |
| 6.3      | デモンストレーション時の評価              | 52        |
| 6.3.1    | EC2018 でのデモンストレーション         | 52        |
| 6.3.2    | 電気通信大学オープンキャンパスでのデモンストレーション | 53        |
| <b>7</b> | <b>まとめと今後の課題</b>            | <b>54</b> |
| 7.1      | まとめ                         | 54        |
| 7.2      | 問題点                         | 54        |
| 7.3      | 今後の課題                       | 54        |

# 1 はじめに

## 1.1 研究背景

近年、スカイツリーや秋葉原等の一般的な場所での Virtual Reality (以下 VR) 体験の場が増え、VR がより身近になっている [1]。さらに、スマートフォンを使って比較的安価に VR を体験することのできる Head Mounted Display (以下 HMD) [2] も登場し、個人によるアプリケーション開発も盛んになっている。VR は医療 [3]・工場 [4]・住宅 [5] 等にも利用されているが、特にエンターテインメントでの目覚ましい発展がみられる。

トランプや麻雀等のテーブルゲームはインターネット上で手軽に対戦して遊ぶことができる。しかし、マウスをクリックするという操作が主になるため、カードや牌を自由に動かしながらプレイするテーブルゲームの臨場感や面白さを損ねることになると考えた。そこで VR で麻雀ゲームをすれば、単調な操作をすることなく臨場感を向上出来ると思った。特に、身体動作を用いた操作方法が鍵になると考えた。昨今の VR ゲームは、腕を振るなどのダイナミックな身体動作を行うものが多く、椅子に座って遊ぶゲームでどの程度身体動作を取り入れることができるか難しいところがある。しかし、古典的な 2D ゲームであるブロック崩しやパックマンであっても VR 環境で遊べるようにすることは、面白さに一定の影響を与えることがわかっている [6]。そのため、麻雀ゲームも VR 化することで面白さに良い影響が出ると考えられる。

麻雀におけるイカサマには、かつては裏芸と呼ばれたように多様な技が存在する。麻雀のプロである小島武夫氏や桜井章一氏らも裏芸を行うことができ、技術の一つであったことが伺える [7]。しかし、不正行為であることは事実なので、見つかった場合には重い罰則が科せられる。現代では、雀荘等で行った場合には出入り禁止等の措置が取られることもある。

一方で、麻雀のイカサマ行為に多くの関心があるのも事実である。ニコニコ動画にてイカサマを行うことをルールとして組み込んだ”イカサマ麻雀選手権”という動画が約 100 万再生されていることから高い関心が見られることが分かる [8]。麻雀のイカサマ行為の面白い点として、技術の一つとして行うことで完成度を競うことや、普通にゲームをしているだけでは作ることが困難な役を作りやすくなることでドラマチックな状況を演出しやすいことにあるのではないかと考えられる。

## 1.2 目的

本研究では麻雀ゲームを VR 化することで、より現実でのプレイに近づける。出来るだけ現実に近い手指の動作で麻雀牌を操作できるシステムを構築するため Leap Motion を

使用し，麻雀ゲーム内で必要な牌の操作を手指だけで実現するシステムを開発する．さらに本研究内では VR ならではの機能として，麻雀ゲーム内でイカサマを行いやすくする．麻雀のイカサマの中でもポピュラーなすり替えを，現実よりも容易に行えるシステムを構築する．現実では卓越した技術を要する動作を簡単に出来るようにすることで，一般の人には普通実現できない面白さを作り出すことを目指す．

### 1.3 本論文の構成

本論文の構成は以下の通り

#### 第 2 章 Head Mounted Display と Input Device

本研究で使用した HMD 及び入力デバイスについて解説する．

#### 第 3 章 麻雀用語

本論文内で使われている麻雀用語について解説する．

#### 第 4 章 既存技術・既存研究

既存の麻雀ゲーム，本研究をするにあたり参考にした研究を記述する．

#### 第 5 章 開発したシステム

Unity で作成した Leap Motion を用いた操作システムについて記述する．

#### 第 6 章 評価

実装したシステムの評価結果を記述する．

#### 第 7 章 おわりに

本研究のまとめ及び今後の課題について記述する．

## 2 Head Mounted Display と Input Device

### 2.1 Head Mounted Display

VR空間を映す機材として一般的なものは、Head Mounted Display(以下 HMD)である。HMDは頭に被り、両眼にあるディスプレイに映像を投影することでVR空間を映し出す。欠点として、視界全体を覆ってしまうためHMDを付けている状態では周りの状況がわからないというものがある。また、画面が急激に変わったりするためVR酔いという現象もみられる。

#### 2.1.1 HTC VIVE

本研究で主に使用したHMDは、HTCとValve Corporationにより開発されたHTC VIVEである[9](図1)。PCと接続し映像を映す方式で、本研究ではUSBを用いてPCと接続していたが、無線によって接続することもできる。ベースステーションと呼ばれる外部機器により赤外線を用いてHMDの位置を捕捉し、ポジショントラッキングすることが可能となっている(図2, 図3)。また、VIVEコントローラと呼ばれる専用のコントローラも存在し、HMDを被っている状態でもコントローラの場所を捕捉することができる。さらに、VIVEトラッカーという別売りの機器を用いることで任意の物体をトラッキングすることも可能となっている(図4)。

HTC VIVEを使用するにはSteamアカウントが必要である[10]。SteamからSteamVRをインストールすることでプレイすることができる。VRゲームの多くもSteamからダウンロードすることでプレイすることができる。

HTC VIVEを使用する際に、ルームスケールとスタンドオンリーのモードを選択することができる。それぞれで使うことに大きな差はないが、HMDを被ることにより周りの



図 1: HTC VIVE(参考文献 [9] から引用)





図 2: ベースステーション



図 3: ベースステーションの配置



図 4: VIVE トラッカー

状況がわからないことへのサポートが少し違う。設定したモードごとに淡い光の補助が入るようになる。スタンドオンリーはその場でのみ使用する場合に有効で、本研究では主にこちらのモードで行っていた。このモードでは最初に設定した場所を中心として円が見えるようになる(図 5)。ルームスケールでは、設定する際に VIVE コントローラを用いて部屋の大きさを歩いて計測する。その計測した場所に壁となるように光が入る(図 6)。これにより、プレイヤーがどこまで動くことができるのかを補助している。

HTC VIVE の欠点として、「処理がとても重くなるので高性能の GPU を積んだ PC が必要」「ベースステーションを配置するための空間が必要」という点がある。ベースステーションの配置には最低  $1.5m \times 2m$  四方の何も無い空間が必要で、ゲームをプレイすることを考えるともっと広い空間が必要である。また、ベースステーションの位置も人の頭の位置よりも高い  $2m$  以上のところに配置するのが好ましいため、使用するための準備が大変である。

### 2.1.2 その他の HMD

HTC VIVE 以外の HMD として、同様にポジショントラッキングが可能な Oculus 社により開発された Oculus Rift がある [11]。ポジショントラッキング用のセンサーが HTC VIVE と違い卓上に置くだけで利用することができ、大きなスペースを必要としない。しかし、センサーの検出範囲が狭いなどの欠点がある。また、同社により開発された Oculus Quest は外部センサーを必要とせず本体に付属されたカメラにより現在位置を把握し、ポジショントラッキングすることが可能である [12]。

スマートフォンを用いて VR 空間を映す HMD として Gear VR がある [13]。HTC VIVE 等と比べると安価に購入することができるが、ポジショントラッキングの機能がなく、視



図 5: スタンドオンリー

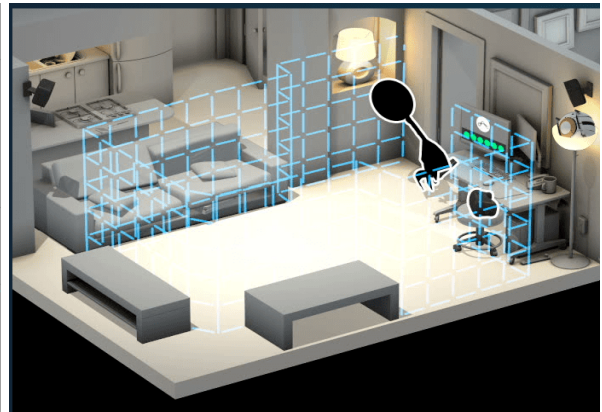


図 6: ルームスケール

界の移動はジャイロセンサーを用いた自分の周囲を見渡すことのみが可能となっている。

## 2.2 Input Device

### 2.2.1 Leap Motion

Leap Motion とは Leap Motion 社が開発した赤外線カメラを用いて手指の動きを取得することができるデバイスである [14](図 7)。検出できる範囲は肘から指の先までで、それぞれの指や右手、左手も区別して認識できる。手に何も装着する必要や持つ必要がないため、手や指を使った様々なジェスチャ入力を手軽に扱うことができる。また、複数人の腕を検出することも可能である。

欠点として、指が重なってしまうと誤認識が起きてしまい、予想外の結果を返してしまう点である。また、デバイス自体が小さいため検出範囲が広くないという点がある。

現在、Leap Motion は複数使用に対応し、試験的な運用がなされている。これにより、一方向からでは認識できずに誤検出してしまう問題点を解消できることが期待される。しかし、複数使用するにはフルスピードで動作する複数のデバイスをサポートするのに十分な CPU パワーと十分な USB 帯域幅があることが必要となっている。また、現在のサポートでは手が Leap Motion の数だけ重なって見えてしまう (図 8)。

### 2.2.2 VIVE コントローラ

HTC VIVE には初めから付属されているコントローラがあり、片手で一つずつ持てるように二つ付属されている (図 9)。このコントローラはベースステーションによってポジショントラッキングされていて、HTC VIVE を被っている場合でも位置を捕捉することができる。主な VR ゲームはこのコントローラを用いて遊ぶことになる。



図 7: Leap Motion(参考文献 [14] より引用) より引用)



図 8: Leap Motion の複数使用 (参考文献 [14] より引用)



図 9: VIVE コントローラ

入力には、トリガー、トラックパッド、グリップボタンの3つを使う。トリガーはコントローラ下部についており、引くことで入力を行える。マウスにおけるクリックのような役割を担う。トラックパッドは表面にある円い円盤上のもので、その上で指を動かすことで入力とする。また、ボタンのように押すことも可能である。グリップボタンは側面にあるボタンで、握りこむとちょうど押せるようになっている。また、VIVE コントローラは位置だけでなく方向もトラッキングされているので、コントローラの角度を変えるとHMD内のコントローラの角度も変わるようになっている。

### 3 麻雀用語

本章では，本論文内で使われている麻雀用語について解説する (図 10)。

- 手牌

プレイヤーが持っている牌。本研究内では基本的に自プレイヤーが持っているものとして，卓の手前に並んでいる。13 牌元々持っていて，山から一牌自摸り，14 牌で役を作る。

- 河

打牌した牌を置く場所。基本的に卓の中央付近に置く。

- 牌山，山

牌が積んである状態。一列に並べて裏返した牌の上に同数の牌が乗っている。ここから牌を自摸り，手牌に加える。

- すり替え

イカサマの一種。手牌と山を入れ替える。通常は一度にすべての牌を入れ替えるのではなく，片手に持てる範囲で入れ替える。様々なやり方があり得るが，例えば以下のような方法がある。牌を二牌掴み，山の右端の奥側へ移動，山に持っている牌を当てるように手前にスライド，手間に持ってきて，置く。

- 自摸

山から牌を引き，手牌に加える行為。山の牌を掴み，手前に持ってきて，置く。表面を手前側に向けるため，牌を掴んでから回転させる必要がある。

- 打牌

自摸の後，手牌から 1 牌河へ捨てる行為。手牌を掴み，奥に持っていき，置く。牌の表面が上へ向くように置くため，回転させる必要がある。

- 洗牌

牌を混ぜる行為。ゲームごとに牌をランダム配置させるために行う。両手で無造作に牌を混ぜる。

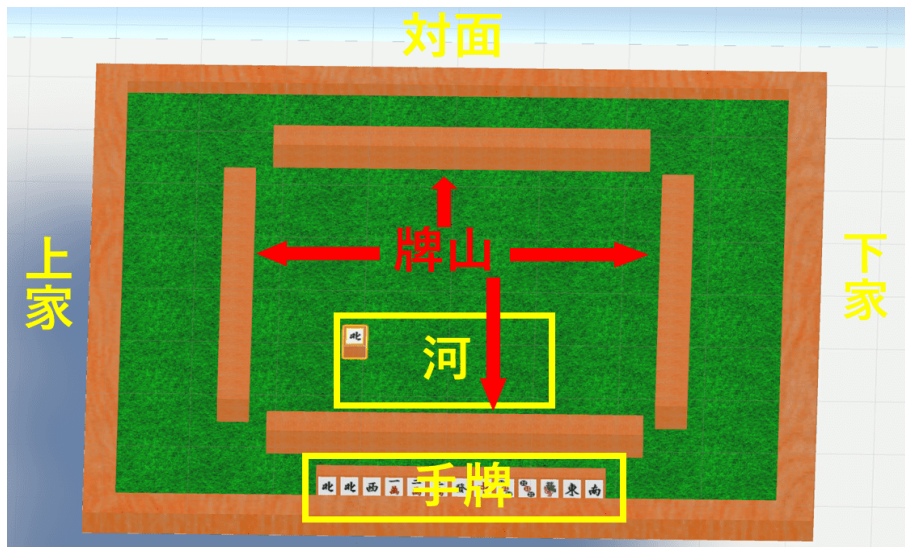


図 10: 麻雀用語の説明図

- 理牌

手牌を並べ変えること。種類や数字の大きさごとに整頓する。牌を掴み、好きな位置に置く。その際、邪魔な牌は掴みどける。その後、牌の間隔をつめて整頓する。牌の方向は常に手前を向いたままである。

- 上家，対面，下家，自家

プレイヤーのいる位置。自家を自分がいる位置としたとき，上家は左，対面は正面，下家は右側とする。

## 4 既存技術・既存研究

### 4.1 VR 麻雀

VRChat-TANKI(単騎)は、VRChat 内にあるワールド”TANKI”のことで、VR 空間で麻雀ができるものである。

#### 4.1.1 VRChat

VRChat とは、VRChat Inc. が開発、運営している仮想空間内で他の人とコミュニケーションが取れるサービスである [15](図 11)。名前の通り、VR で Chat を行うもので世界中の人とコミュニケーションを取ることができる。Steam からダウンロードすることが出来、基本的に無料で利用することができる。また、動作に制限が出来てしまうが VR 機材がなくても利用可能である。

VR でプレイする場合には、VIVE コントローラを両手に持った状態が基本となる。さらに、VIVE トラッカーを足や腰に装着することでさらなる没入感を得ることができる。

VRChat 最大の特徴として、プレイヤーのアバターやチャットルームであるワールドを自作し、使用することができる点である。例えば、アバターは 3D スキャンした自分の体に骨格のスクリプトをあてることで自分自身をアバターとして使用することが出来る。ワールドは自分の部屋の家具等を再現して配置することで、離れている友人とあたかも自分の部屋で遊んでいるかのような感覚を味わうこともできる。もちろん、バーチャルならではの現実ではありえないような空間を作成することも可能である。



図 11: VRChat(参考文献 [15] から引用)



図 12: 部屋の様子 (参考文献 [16] より引用) 図 13: 壁のボタン (参考文献 [16] より引用)

#### 4.1.2 TANKI(単騎)

Lieselotte 氏が作成した VRChat で麻雀を行えるワールドで一般に公開されていて誰でも遊ぶことができる [16]. 部屋自体は麻雀卓が置いてあるシンプルな構成で壁にある配牌ボタンを押すことで牌が配られる (図 12, 図 13). 手牌を置く部分は牌が必ず手前を向く機能がついていて, 手牌の範囲内であれば左右に滑らかにスライドする. この機能により牌を置きたい場所にある他の牌を掴んで動かす必要がない. さらに, 一列に整列している状態が崩れないため, 牌を置いたらそのままが良い. 以上のことから理牌しやすくなっている. これ以外の自摸等の動作は手動で行うようになっている (図 14).

牌の操作方法は VIVE コントローラを用いるもので, 一般的な VRChat 内のオブジェクトを持つものと変わらない. 持ちたいオブジェクトの方にコントローラを向けると接続線が出て, 現在持てるオブジェクトが灰かに光るようになっている. その状態でコントローラのトリガーボタンを押すことで指のところまで移動して持つことができる. ただし, この機能には角度調整機能がなく, 持つ前のオブジェクトの角度のまま保持することになるため, 自摸や打牌する際には手首を捻り角度を変える必要がある.

TANKI を用いた麻雀大会も開催されておりその様子が YouTube に配信もされている [17]. ただし, 2019 年 1 月 21 日現在, VRChat がサポートしている Unity のバージョンが更新されたことで利用できなくなっている.

本研究とは, 入力デバイスの違いやイカサマを行えるかという点で差異がある.



図 14: プレイ中の様子 (参考文献 [16] から引用)





図 15: イカサマ麻雀 (参考文献 [18] から引用)

## 4.2 イカサマゲーム

### 4.2.1 イカサマ麻雀

メディアファクトリーにより 2000 年に発売されたプレイステーション用のゲームである [18](図 15)。多様なイカサマ技を使って遊べるが、「相手の隙を見てイカサマする」「逆に相手のイカサマを見破り勝つ」ことに主眼をおいたゲームで、麻雀というよりイカサマゲームという側面の方が強い。また、自摸の度にイカサマするかどうかの選択肢が出るためテンポの悪さがある。

本研究とは、イカサマを行う麻雀という点で同じであるが、VR で行う点、イカサマ動作を身体操作で行う点、イカサマを行うかはプレイヤーが任意に選ぶという点で違いがある。

### 4.2.2 咲-Saki-

小林立による麻雀漫画である。作中内でキャラクター達はそれぞれ特殊能力を駆使しながら麻雀をプレイする。また、PS Vita で本作を題材としたゲームが発売されている [19]。プレイヤーはキャラクターを選択し、そのキャラクター毎の固有の能力を使うことができる。例として、

- 自分が欲しい牌を引きやすくなる
- 他プレイヤーの欲しい牌がわかる
- 特定の牌のみ引くようになる



図 16: いかさまゴキブリ (参考文献 [20] から引用)

が挙げられる。これらはイカサマではないが、現実ではありえない力が働くことでプレイヤーに有利な状況をつくることができ、創作物ならではのドラマチックな状況を演出しやすい。現実では味わうことができない状況を作るという点で、本研究で扱うイカサマと似ているところがある。

#### 4.2.3 いかさまゴキブリ

Emily Brand と Lukas Brand により制作されたカードを用いたボードゲームの一種である [20](図 16)。プレイヤーは配られたカードを順番に出していくことが基本ルールで、一番最初に手札を無くした人が勝ちである。このゲームの特徴は以下の通り

- 出したくても合法的に出せないカードがある
- 気づかれなければ、カードをどこかに捨てるか隠してもよい

出したくても合法的に出せないカードというのは手番で捨てることが出来ないカードが存在するという点で、このカードを持っているとずっと勝つことが出来ない。そのため、他のプレイヤーにばれないようにどこかに捨てる必要がある。勝つためにはイカサマを行わなければならない、イカサマを行うことを前提としたゲームである。本研究とはゲームで用いる道具が違うが、「イカサマによってゲームを優位に進める」「プレイヤーが任意にイカサマを行える」点が似ている。

### 4.3 Leap Motion を使った入力

Spyros Vosinakis らは Bare hand interaction(以下 BHI) の普及には、物を掴む、放すのような基本動作の最適化が必要であると考えた [21]。BHI とはユーザが手に何も持つことなく手の動きを入力とすることである。Leap Motion を用いた操作方法がこれにあたる。

Spyros Vosinakis らは掴む、放すの相互作用については触覚フィードバックがないと、ユーザがオブジェクトの位置の把握が困難になるとしている。しかし、追加の視覚的フィードバックを施すことで改善できる可能性があるとした。そこで、Leap Motion を使った掴む、放す動作における 4 つの視覚的フィードバック手法によるユーザビリティの評価を行った。

4 つの視覚的フィードバックは「色付け」「接続線」「影」「ハロー」と呼ばれるものを用いた。図 17 はフィードバックがない時の表示である。

- 色付け (図 18)

ユーザが持ちたいオブジェクトに手を近づけるとオブジェクトの色が変わる。

- 接続線 (図 19)

持つことが可能なオブジェクトと手の間に光の線が一本入る。

- 影 (図 20)

本来、オブジェクトや手には影が設定されていないが、影ができるようになる。

- ハロー (図 21)

持つことが可能なオブジェクトの周りがほのかに光る。

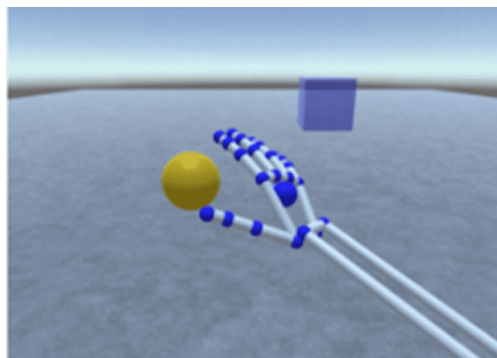


図 17: フィードバックなし (参考文献 [21] から引用)

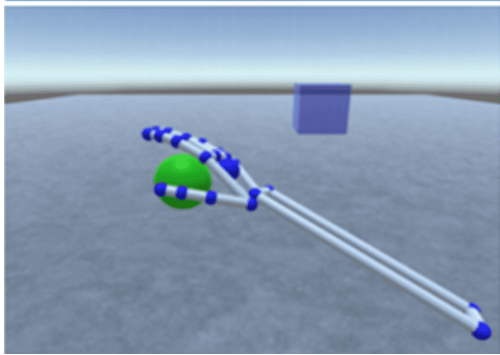


図 18: 色付け (参考文献 [21] より引用)

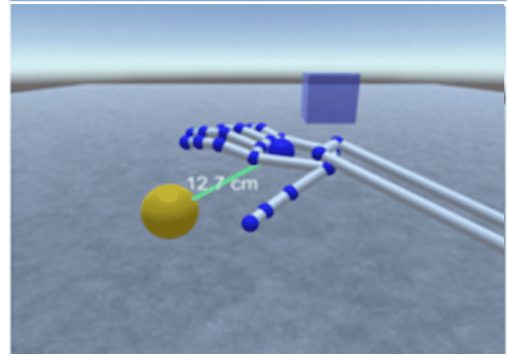


図 19: 接続線 (参考文献 [21] より引用)

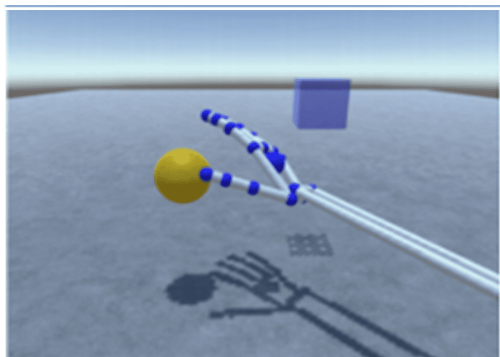


図 20: 影 (参考文献 [21] より引用)

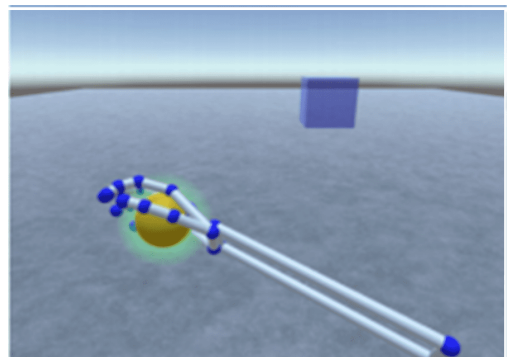


図 21: ハロー (参考文献 [21] より引用)

評価する入力デバイスは Leap Motion を用いるが，ディスプレイは HMD と通常のディスプレイの二つを用いて評価した．タスク時間，精度，エラー，満足度の項目で評価したところ表示方法の評価では HMD を用いたときの方が良い結果となった．また，ほとんどのフィードバック手法で何もフィードバックがないよりもユーザビリティに有益な結果を残した．しかし，HMD を用いて接続線を使った場合は，しばしばユーザの操作を妨げる結果になっている．そのため，視覚的フィードバックを入れた方が良くなるとは強く言えないが，一部のフィードバック手法は良い結果となっている．

本研究では，Leap Motion を用いた評価を行っている点が同じであるため，評価の項目について参考にした．ユーザ満足度は 4 つのアンケートを取っており快適性，正確性，直観性，好感度を用いている．本研究でも同様の観点で評価した．

## 5 開発したシステム

### 5.1 システム全体像と使用した機材

システムの全体図を図 22 を示す。Leap Motion で手指の位置情報を取得し、PC 内の Unity で処理を行う。処理結果を HTC VIVE へ出力する。Leap Motion からの入力によって、Unity 内の麻雀牌を操作する。また、ベースステーションが取得した HTC VIVE の位置によって、Unity 内のポジションを変更する。

#### 5.1.1 Unity

シーンの作成は Unity2018.1.0f2[22] を用いた。Unity とは Unity Technologies が提供するマルチプラットフォームに対応したゲームエンジンのことである。開発環境としては、Windows, Mac, Linux の他に Web, iOS, Windows Phone, PlayStation3, Xbox360, WiiU に対応している。また、VR アプリの作成にも利用されている。Unity は近年非常に人気が高まっており、ゲーム開発への利用が増加傾向にある。

#### 5.1.2 Head Mounted Display

本研究では、VR 空間を投影する HMD として HTC VIVE を用いた。入力デバイスである Leap Motion は検出範囲が広くないが、ヘッドトラッキングされている HMD に装着することでその欠点を補うことができる。そのため、本研究ではヘッドトラッキング可能な HMD の一つである HTC VIVE を用いた。

#### 5.1.3 Leap Motion

本研究では、Leap Motion を入力デバイスとして扱う。触覚フィードバック機能はないが、手に何も持つ必要がないため手軽に扱え、手指の形の取り方の工夫次第で多くの入

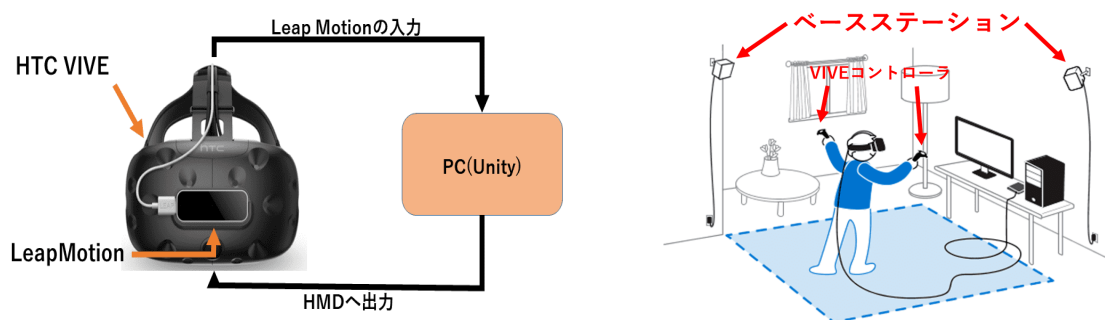


図 22: システムの全体図



図 23: HTC VIVE と Leap Motion

力を行うことができる。また、検出範囲の問題も後述の使い方によって改善される部分があるため、このデバイスを選択した。

Leap Motion で取得した値から手のモデルを作成するのは、Leap Motion からディベロッパー用に提供されている Orion[23] という Asset を用いた。これは、Unity にインポートするだけで、ハンドモデルがあらかじめ用意されており、Leap Motion が検出できるように手をかざし動かすと Unity 内のハンドモデルも同じ動作をするというものである。

#### 5.1.4 Leap Motion の HMD への装着

HTC VIVE の本体の前面に Leap Motion を設置する (図 23)。仮に、Leap Motion を現実空間内に固定した場合、画面の奥にあるオブジェクトを掴みたいとき、手を目一杯伸ばすと検出範囲から外れてしまう。

Leap Motion を HMD に装着すると、仮想空間内での Leap Motion の位置は顔の前にあると仮定できるので、頭を前に動かしオブジェクトに近づくことで、手を Leap Motion の検出範囲に入れ前述の問題を解決することができる。

## 5.2 牌操作の仕組み

本システムのうち、Leap Motion を用いて VR 空間内で麻雀を遊ぶために必要な牌の操作について説明する。

麻雀牌の操作は Magnetic\_Pinch スクリプト [24] を元に改良して実装した。ピンチ (親指と他の指をくっつける動作) を検出するスクリプトと、持ちたいオブジェクト毎に処理を変えるスクリプトに分けて設計した。Magnetic\_Pinch スクリプトは親指とその他の指

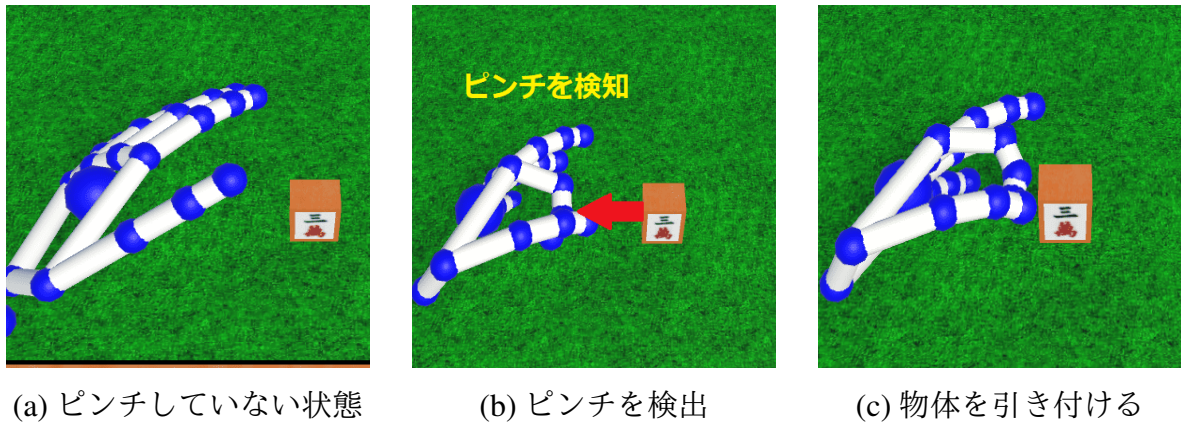


図 24: Magnetic\_Pinch スクリプトの説明図

が一定距離よりも近づいた場合にピンチしたと判断し、ピンチ位置から球状範囲で最も近いオブジェクトを AddForce 関数により近づけることでオブジェクトを持つスクリプトである (図 24)。

AddForce 関数は、物体を指定した座標へ引き付ける関数である。変数は引き付ける位置である座標と、引き付ける力の大きさである。

改良した部分を以下に示す。

### 5.2.1 オブジェクトの保持方法の改良

既存の Magnetic\_Pinch スクリプトは保持対象のオブジェクトには、AddForce 関数のみを作用させている。しかし、そのままでは Unity 内の物理エンジンによる重力と反発しあい、オブジェクトが上下に動き続けることになり、結果ホバリング (机と手の間で浮いた状態) してしまう。そのため、ピンチしている間の保持対象のオブジェクトの重力作用を無効にし、放すときに改めて、重力作用を有効にする手法を取っている。

また、保持しているオブジェクトが回転してしまう現象を解消するために、Unity の FreezeRotation を有効にすることで角度を一定に保つようにしている。さらに、保持したときに表面をプレイヤー側に向けることで、他プレイヤーから見られてしまうことを防止している (図 25)。

### 5.2.2 オブジェクト毎の持ち方の変更

オブジェクト毎に持てる手と指の数を制限した。麻雀においては、一つの牌のみを掴むことが主な動作となるが、牌山を作る、配牌を取る、今回実装したイカサマ動作等、複数の牌を同時に持つことがある。



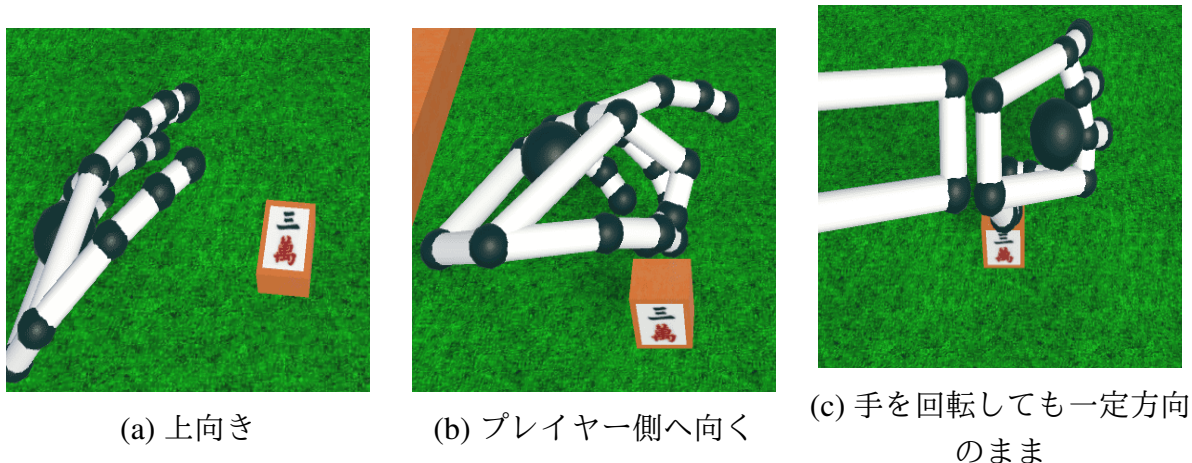


図 25: 角度を一定に保つ

表 1: 持ち方

| 牌の状態, タグ   | 使う手 | 使う指          |
|------------|-----|--------------|
| 単牌 (図 26)  | 片手  | 親指, 人差し指     |
| 列 (図 27)   | 両手  | 親指, 人差し指     |
| 複数牌 (図 28) | 片手  | 親指, 人差し指, 中指 |

Magnetic\_Pinch スクリプトは同時に複数の異なるオブジェクトを持つことに適しておらず, Unity の物理エンジンも現実的な力学作用を完全に再現出来ないので, 圧力で挟み, 持ち上げることも難しい. Unity の仕様上, 複数オブジェクトをまとめて持とうとする場合, 持ちたいオブジェクトを子として, 一つの親にまとめる必要がある. このとき, 親オブジェクトのタグを変えることで牌の状態ごとに持てる指を制限できるようにした. 十以上の牌を片手で持つことは現実空間では出来ないことから, オブジェクトのまとまり毎に使う指の本数や, 手の数を決めた. 表 1 にオブジェクトのタグと手, 指の数を示す.

### 5.2.3 複数の異なるオブジェクトの保持

前述した複数牌は親オブジェクトに紐づけされた一つのオブジェクトとして定義されている. それとは別に, 単牌を一つ持っている状態から, もう一つ別の単牌を保持できるようにした. 単牌は, 親指と人差し指を使って保持するが, その状態から中指を曲げることで, もう一つ別の牌を保持することができる. 保持する位置は, 最初に持っている牌の右隣に配置される. 以下に状態の遷移を画像で示す (図 29).

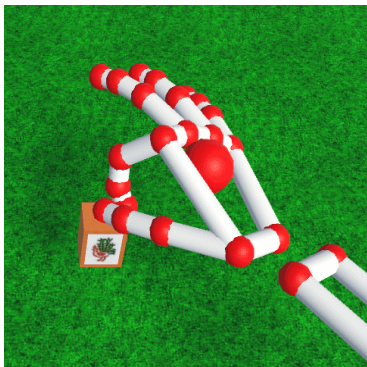


図 26: 単牌

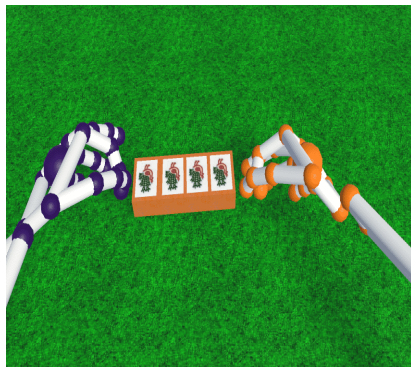


図 27: 列

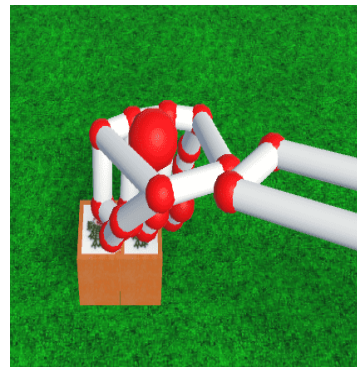
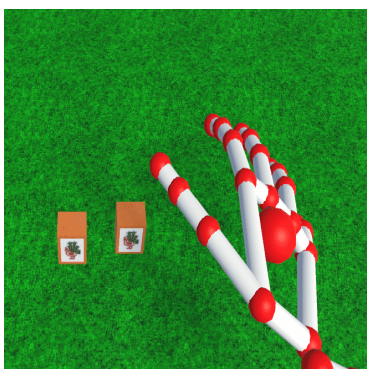
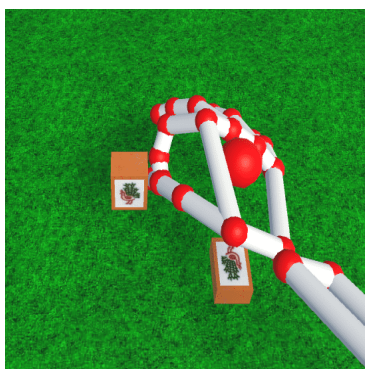


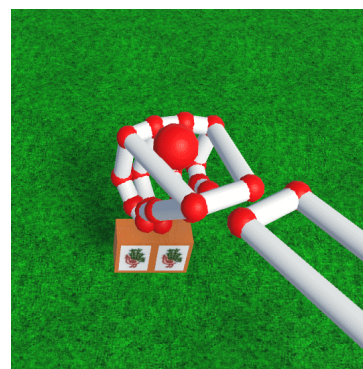
図 28: 複数牌



(a) 何も持っていない状態



(b) 牌を一つ持っている状態



(c) 二牌目を持っている状態

図 29: 二牌保持

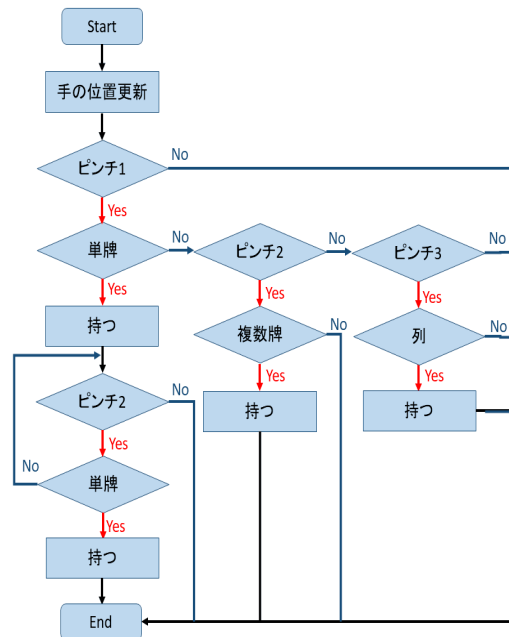


図 30: ピンチ動作の状態遷移

ピンチ操作の状態遷移図を図 30 に示す。図中のピンチ 1 とは右手の親指と人差し指でのピンチ、ピンチ 2 は右手の親指と中指のピンチ、ピンチ 3 は左手の親指と人差し指のピンチをそれぞれ表している。

### 5.3 イカサマ操作

イカサマの中で最もポピュラーなすり替えを実装した。手牌の一部(二牌)と自分の前の牌山をすり替えることができる。奥から手前に牌同士を当てるようにスライドさせることでイカサマを行う(図 31)。5.2 の牌操作におけるオブジェクト毎の手の形の制限、別オブジェクトの保持を基に、二牌もつ動作をイカサマ開始の条件とする。現実より容易に行える点として以下を挙げる。

- 牌の持ち方を工夫する必要がない  
現実では牌の角度や持つ個数により牌を重ねる必要があり、技術を要する。
- すり替え途中で牌を持ち変える必要がない  
現実では牌を一瞬で持ち替える必要があり最も難しい部分である。本システムではスライド動作のみで持ち替えられる。

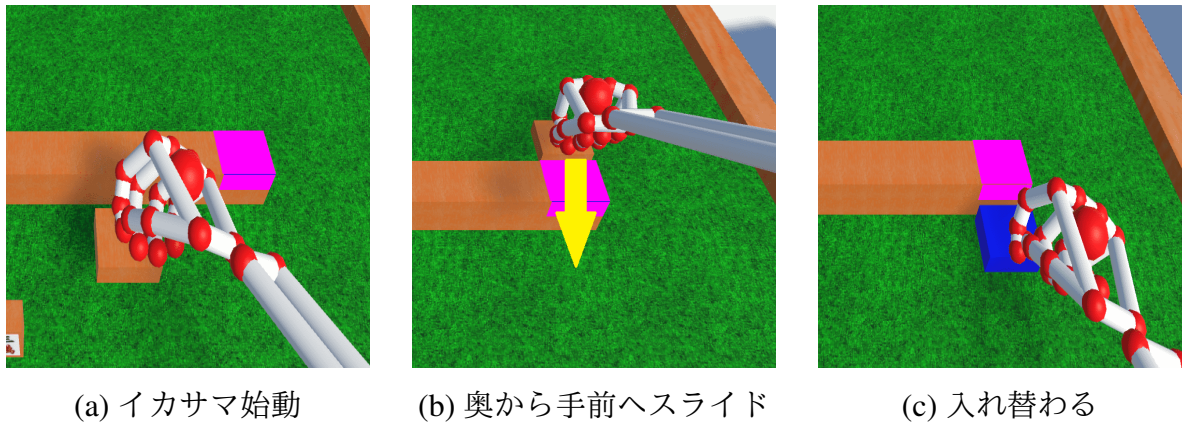


図 31: イカサマの説明図

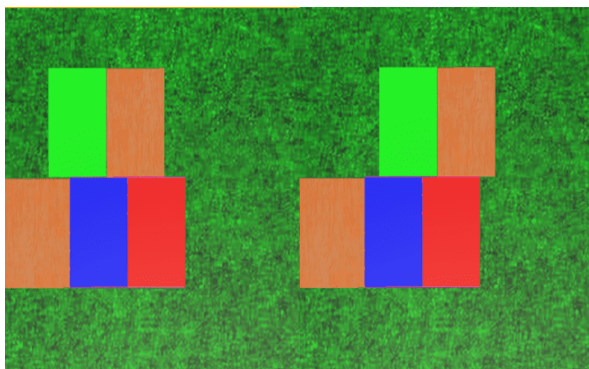


図 32: 牌同士の距離 0.020

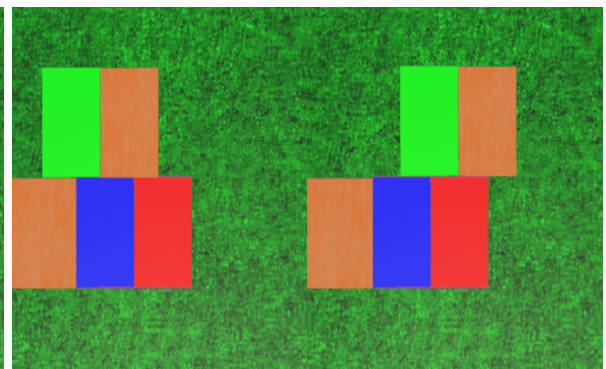


図 33: 牌同士の距離 0.023

しかし、すり替えが容易すぎると面白くなくなってしまう可能性がある。例えば、コントローラのボタン入力ですり替えが実行できるシステムでは、イカサマを行うスリルが減って面白くなくなってしまうと考えられる。そこで、熟練度によっては、牌山にある他の牌が動いてイカサマがばれてしまうようにした。熟練度の判定は、牌同士を当てた時の距離によって決まり、持っている牌の立体としての中心と牌山の平面としての中心で判定している(表 2)。そのため、図 32、図 33 では左右のずれのみを表しているが、上下にずれ過ぎてもイカサマは成功しない。

判定する場所は牌の前面と後面に用意した(図 34)。イカサマ時の牌同士の距離が大きいときは、すり替え対象の横にある牌に左方向への力を加えることで、牌山を崩している(図 35、図 36)。

表 2: イカサマの判定

| 判定     | ミス無し            | 小ミス                | 大ミス             |
|--------|-----------------|--------------------|-----------------|
| 牌同士の距離 | 0.020 未満 (図 32) | 0.020 以上, 0.023 未満 | 0.023 以上 (図 33) |
| 牌山の状態  | 変化なし            | 小さく崩れる (図 35)      | 大きく崩れる (図 36)   |

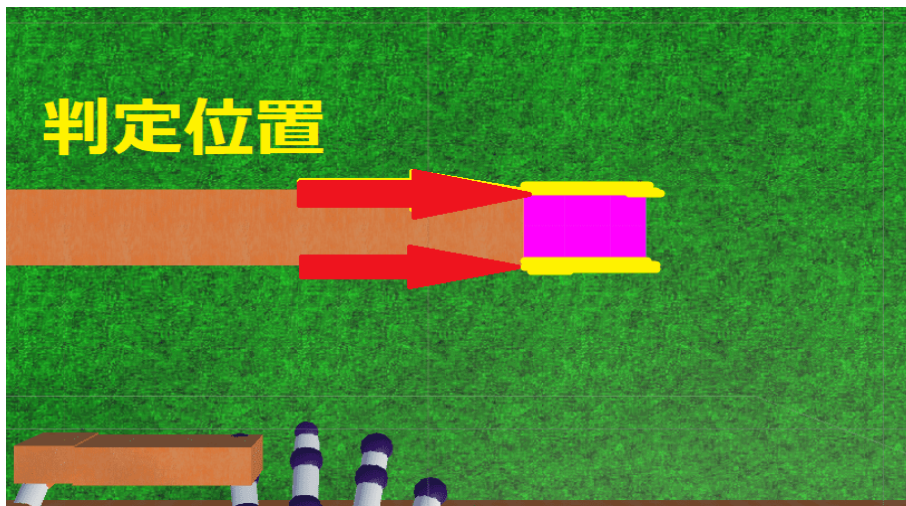


図 34: 上から見た判定範囲

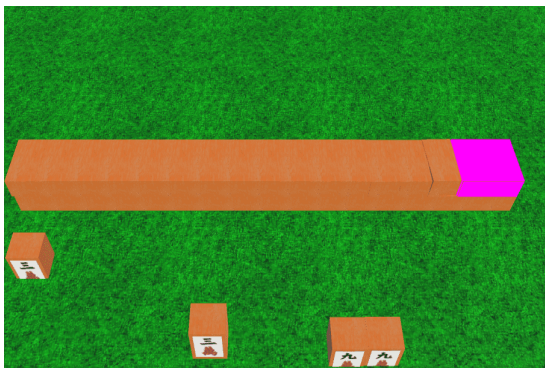


図 35: 小さく崩れた状態

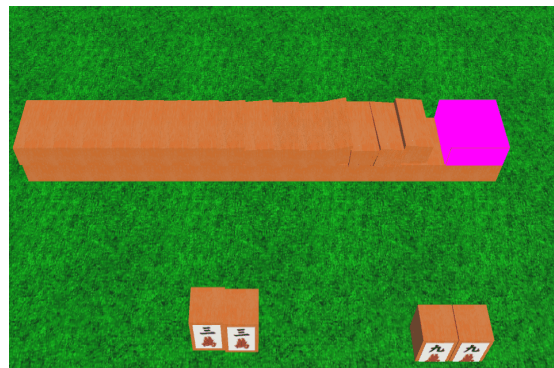


図 36: 大きく崩れた状態

表 3: 牌の状況と角度

| 牌の状況        |      | 角度制限 | 角度変更   | 牌の状態 |
|-------------|------|------|--------|------|
| 保持          | 牌一つ  | あり   | プレイヤー側 | 単一   |
|             | 牌二つ  | あり   | プレイヤー側 | 単一   |
|             | 列    | あり   | なし     | 列    |
|             | 複数   | あり   | なし     | 複数   |
|             | イカサマ | あり   | 下側     | 複数   |
| 手牌ゾーン       |      | あり   | プレイヤー側 | 単一   |
| 河ゾーン        |      | あり   | 上側     | 単一   |
| 洗牌 (一番手前以外) |      | なし   | なし     | 単一   |
| 牌山作り        |      | あり   | 下側     | 列    |
| それ以外        |      | なし   | なし     | 単一   |

## 5.4 その他の牌操作

Leap Motion を使った麻雀で遊ぶためにその他の牌操作機能を実装した。以下で説明する各状況における牌の角度の制限を表 3 に示す。

### 5.4.1 洗牌

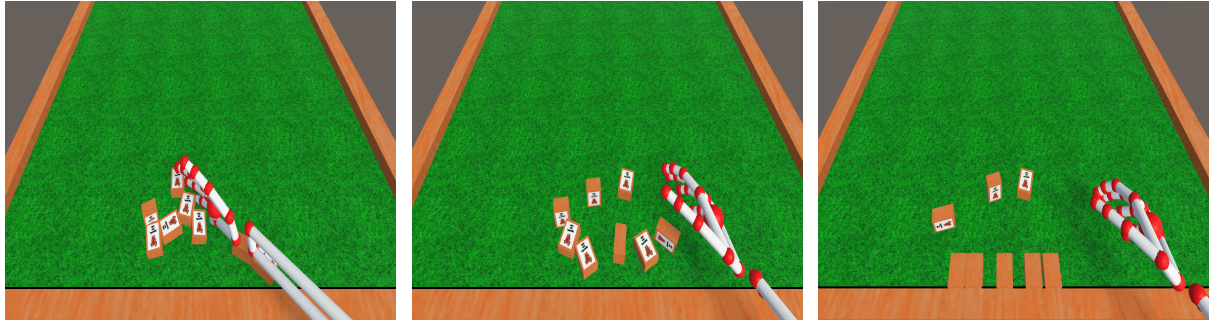
Leap Motion を Unity で使う場合、コライダーと呼ばれる物理衝突のためのコンポーネントがアタッチされている。そのため、初めからオブジェクトと接触することができるのでその機能を用いる。

牌山を作る際にプレイヤー側へ牌を寄せるが、牌の向きはそれぞれバラバラなので、壁に持ってきた際に一定の角度にし、裏返すようにした (図 37)。

### 5.4.2 牌山作り

手前に牌を寄せただけでは、隙間があり積むことができななので、手をスワイプさせることで隙間を無くし、整列させる機能を実装した (図 38)。このとき、寄せた牌を一つの親に入れることで、列オブジェクトとして動かせるようにしている。

二列目は、手前に寄せた牌を一行目の端から順に詰めて並べるようにし、一行目と同数並んだ場合、一つの親オブジェクトに入れ、列オブジェクトとして動かせるようにしている (図 39)。

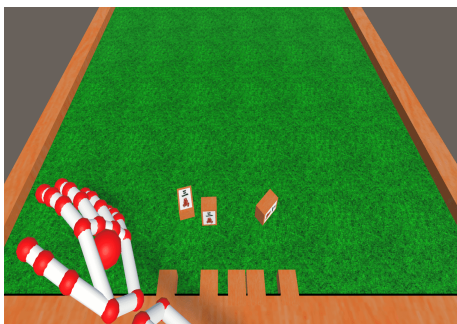


(a) 混ぜる

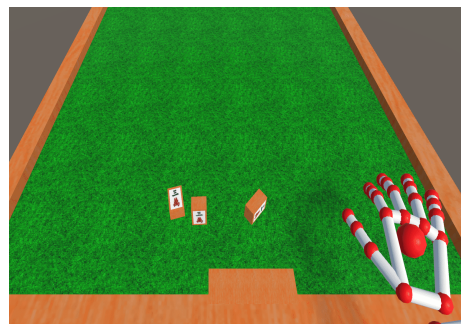
(b) 手前に寄せる

(c) 裏返る

図 37: 洗牌時, 手前に寄せる



(a) スワイプする



(b) 隙間が埋まる

図 38: 整列動作

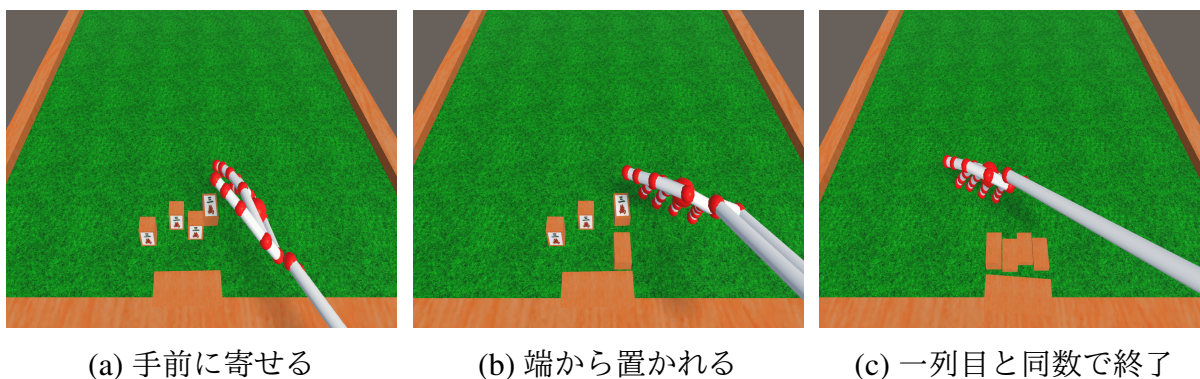


図 39: 二列目の作成

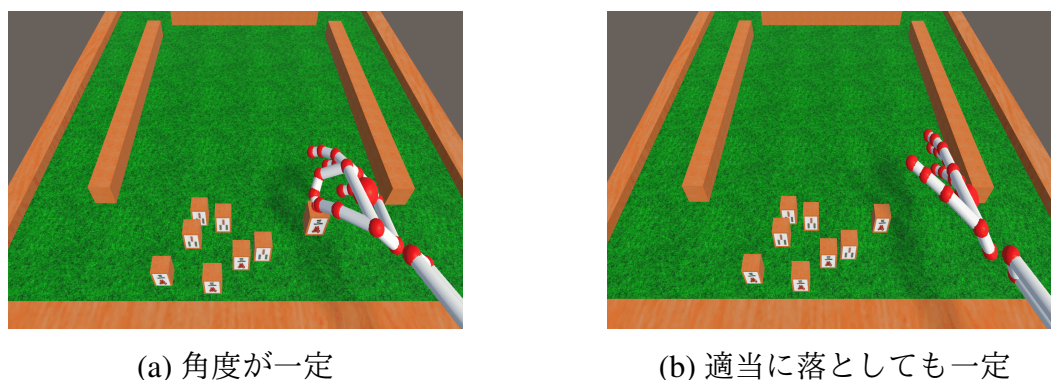


図 40: 手牌-角度が一定に保たれている様子

### 5.4.3 手牌

手牌を置く場所を実装した。プレイヤーの前の机を手牌ゾーンと仮定して、その上に置いた牌に対して、必要な作用を働かせる。実装してある機能は以下に示す。

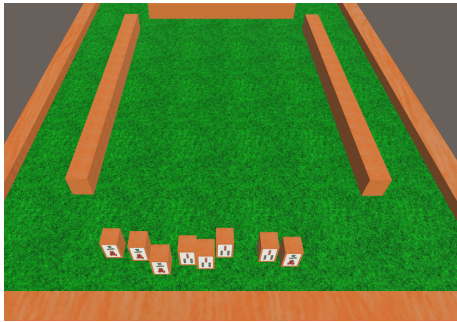
- 角度を一定に保つ

置いた牌をプレイヤー側に向け、立てた状態で置かれるようにしている(図 40)。これにより、誤って牌を落としてしまっても、自分の手牌ゾーン内なら他プレイヤーへ見られてしまうのを防ぐことができる。さらに、理牌するとき角度を気にする必要がないため、牌の並べ替えが簡単である。

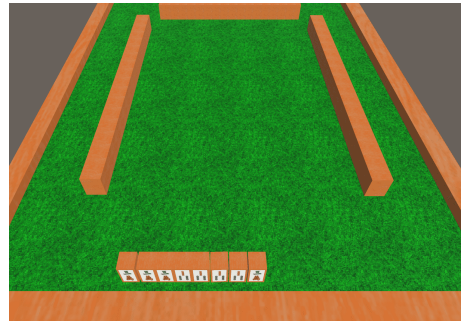
- 理牌

固定されるのは向きのみなので、牌を動かしていると手牌ゾーン内でバラバラになってしまう。そのため、牌を一列に整列させる機能を実装した。手牌の上で手を



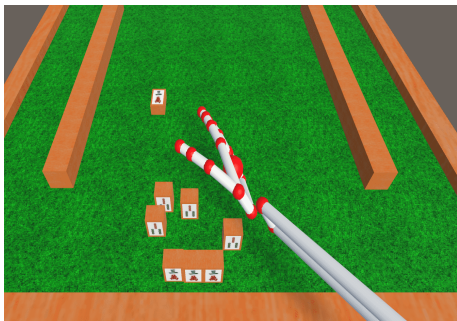


(a) 手牌内でバラバラ

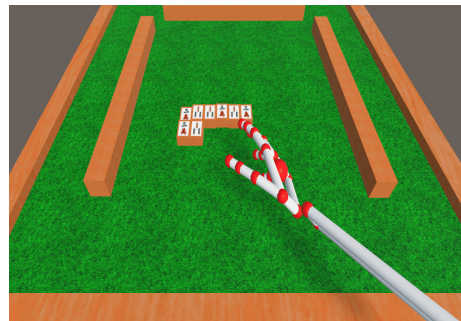


(b) きれいに整列

図 41: 理牌動作



(a) 表を上向き



(b) 6 牌毎に整列

図 42: 河

スワイプさせると整列させることができる (図 41)。あらかじめ設定した手牌ゾーン内の中心へと手牌ゾーンにある牌を移動するようなシステムになっている。そのため、牌同士が縦に固まっていると、整列した際に重なってしまうので、牌同士が重なり合わないようにより離しておく必要がある。

#### 5.4.4 河

手牌ゾーンと同様に河ゾーンを実装した。手牌ゾーンとは違い、他プレイヤーにも牌を見せる必要があるため、牌の表面を上に向けた状態で角度を固定するようになっている (図 42)。また、六牌を一行として、七牌目以降は一段ずらして配置されるようになっている。河ゾーン内であればどこに置いたとしても、きれいに整列されるようになっている。

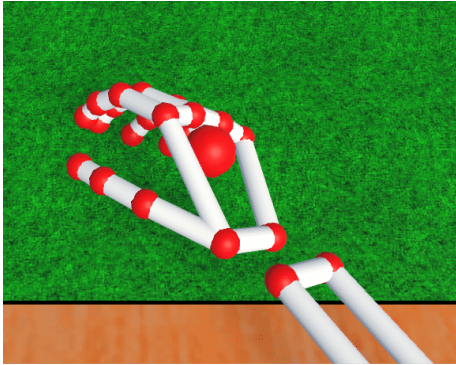


図 43: 平常時の手

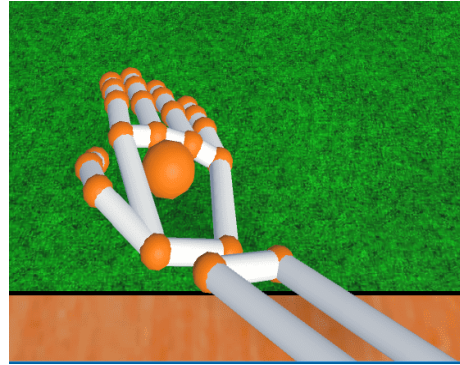


図 44: 指をそろえた状態

## 5.5 実装したシステムの問題点

オブジェクトを持つためのトリガーであるピンチ動作の取得が、ピンチしていない場合でも誤認してしまうことがある。これは、Leap Motion が一方向からしか手を検出することができないためであり、平常時の手の状態でも角度によってピンチしていると認識してしまうからである (図 43)。

また、手のスワイプ動作を行うために、全ての指をそろえた状態をトリガーとして実装しているが、この状態もピンチしたと判断されてしまうことがある (図 44)。スワイプスクリプトを `Magnetic_pinch` スクリプトとは別に実装したが、両方のスクリプトを有効にしているとスワイプ動作を行うために、全ての指をそろえると `Magnetic_pinch` スクリプトのピンチトリガーも発火してしまう。

対処法として、スワイプ動作に関しては `Magnetic_pinch` スクリプトとスワイプスクリプトを、同時運用しないことで解決することができる。スワイプ動作をトリガーとしている動作は洗牌と理牌であるので、そのときには `Magnetic_pinch` スクリプトを無効にすることで対処が可能である。誤認してしまう点に関しては、オブジェクトを持ちたいとき以外は手の形をパーにしておく等の方法がある。しかし、プレイヤーへの負担が大きい。

## 6 システムの評価

ここでは、Leap Motion を使った牌操作及び作成したイカサマサポートシステムの有用性について評価する。イカサマサポートシステムについては評価結果から実際に麻雀ゲームに組み込んだ時への影響を考察する。また、デモンストレーションを行った際の体験者からのフィードバックについても述べる。

### 6.1 牌操作の評価

今回実装した Leap Motion を使った操作方法について評価する。牌を自模る、打牌する部分に対する評価実験を行った。

#### 6.1.1 実験内容

実装した本システムの Leap Motion を使った操作方法 (図 45) と、既存技術の VRChat 内の麻雀で使われている操作方法 (図 46) を再現したものとを比較した。上家、対面、下家、自家の 4 箇所 17 牌 2 列の山を用意して、時計回りに牌を自模り、打牌動作を行う。つまり、牌を自模る山の順番は、上家→対面→下家→自家となる。各山からは 4 牌ずつ自模り、4 牌自模ったら次の山に移る。自模った牌は、必ず手牌の右端においてもらうものとする。また、操作方法の実施順は本システムを行った後に、VRChat のシステムを行った。

自模って手牌に加える時間、手牌の牌を打牌する時間を計測し、評価する。また、自模るべき牌以外の牌を持ってしまった場合ミス動作としてカウントする。また、操作方法についての満足度のアンケートを取った。評価する項目は、快適性、正確性、直観性、好感度の 4 項目を 5 段階で評価してもらう。快適性は牌操作を行っていてストレスを感じなかったか、操作が楽だったか、正確性は持ちたい牌以外を掴んでしまうことがなかったか、直観性は操作方法がわかりやすいか、好感度は使っていてよかったかや実際にゲームをするとしたらどうかを表すものとする。

実験手順は以下の通りである。

1. 牌の操作方法について説明し、慣れてもらう
2. オブジェクトを実験用の配置にする
3. 上家が牌を打牌したのを確認する、(計測開始)
4. 指定された山から牌を自模り、手牌に加える、(時刻を記録)

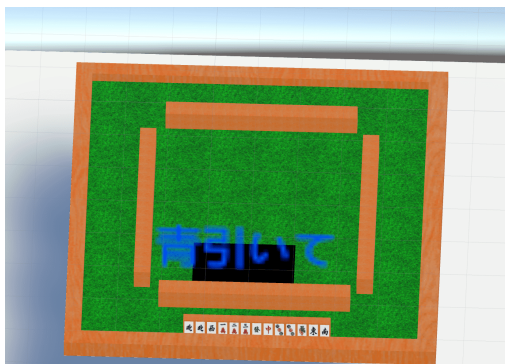


図 45: Leap Motion を使った麻雀

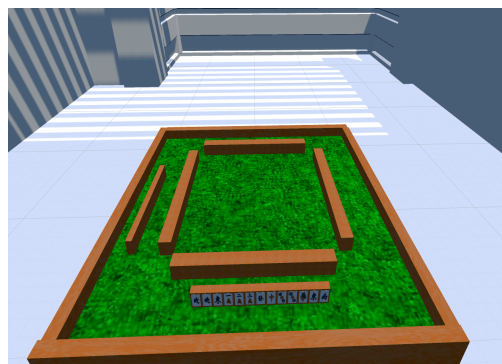


図 46: 自作した VRChat 内の麻雀

5. 加えた牌以外の牌から好きな牌を選び打牌する, (時刻を記録)
6. 3 ~ 5 を 4 回繰り返す
7. 牌を自模る山の位置を変更し, 2 から再度始める
8. 4 箇所, 全ての山で行う
9. 行ったデバイスの操作方法について満足度のアンケートを取る
10. 違うデバイス (VRChat) での操作方法に切り替えてもう一度 1 から行う

### 6.1.2 VRChat 内での麻雀牌の操作方法

VRChat がサポートしている Unity のバージョンが更新されたため, VRChat 内で元々あった麻雀を行うワールドの”TANKI(単騎)”が利用出来なくなりました. そのため, 似たような牌操作ができるワールドを自作した(図 46). Unity のバージョンは, Unity2017.4.15f1 を用いた. ワールド内の部屋は公開されているアセット VRChat Home Kit をそのまま使い, オブジェクトへの作用も VRC.PickUp スクリプトを用いた [25].

元々あった”単騎”内では, 手牌を保持する部分があったが, その部分はスクリプトが無く再現出来なかった. そのため, Leap Motion の操作と同様に自分の前に持ってくる動作のみを自模動作とする. しかし, VRCPickUp の仕様上, 牌の角度は持ったときのまま固定されている. そのままでは自模った牌が見えないので, 手牌として加える前に VIVE コントローラを捻り角度を調整する必要がある.

牌の保持は以下のようなになる. 持ちたい牌の近くに VIVE コントローラを近づけると, 現在持てる対象の牌の周りが仄かに光り, アバターの手とオブジェクトの間に線が出現



図 47: VIVE コントローラの操作方法

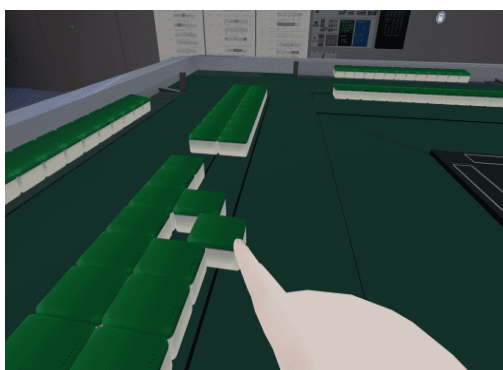


図 48: VRChat 麻雀”単騎”の牌山 (参考文献 [16] より引用)

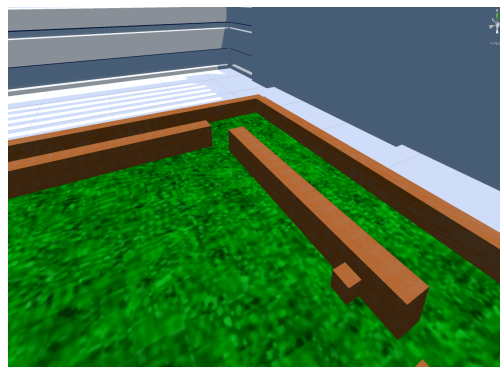


図 49: 実験で使った牌山

する。その状態で VIVE コントローラ下部にあるボタンを押すと牌を持つことができる (図 47)。押している間は牌を保持し続け、ボタンを押すのをやめればその時点で牌はアバターの手から離れて落ちる。

VRChat 麻雀”単騎”では、山の牌は積まれておらず並べてあった (図 48)。しかし、Leap Motion の評価実験でも現実と同様牌は積んであるので、敢えて”単騎”と同じ状況を再現しなかった (図 49)。

表 4: 自摸・打牌の操作時間 (秒)

|       | 自摸場所   | 自摸 (s) |     |     |     | 打牌 (s) |     |     |     | ミス |
|-------|--------|--------|-----|-----|-----|--------|-----|-----|-----|----|
|       |        | 上家     | 対面  | 下家  | 自家  | 上家     | 対面  | 下家  | 自家  |    |
| 被験者 A | 本システム  | 4.5    | 5.7 | 5.1 | 2.1 | 2.9    | 2.0 | 2.1 | 1.9 | 3回 |
|       | VRChat | 6.8    | 7.2 | 5.7 | 4.7 | 2.2    | 3.1 | 2.7 | 2.4 | 0回 |
| 被験者 B | 本システム  | 3.4    | 5.8 | 4.5 | 2.4 | 2.0    | 2.0 | 1.4 | 1.6 | 4回 |
|       | VRChat | 5.7    | 6.4 | 5.9 | 5.0 | 1.5    | 1.8 | 2.1 | 2.1 | 1回 |
| 被験者 C | 本システム  | 3.8    | 7.3 | 5.6 | 3.0 | 2.7    | 3.2 | 3.3 | 2.2 | 9回 |
|       | VRChat | 10.3   | 7.2 | 9.0 | 4.0 | 4.4    | 6.0 | 5.0 | 4.1 | 0回 |
| 被験者 D | 本システム  | 3.4    | 3.8 | 3.7 |     | 2.0    | 2.0 | 2.2 |     | 1回 |
|       | VRChat | 12.1   | 7.2 | 6.4 | 5.3 | 4.4    | 4.7 | 3.3 | 3.6 | 3回 |
| 被験者 E | 本システム  | 4.7    | 5.5 |     |     | 2.5    | 2.5 |     |     | 3回 |
|       | VRChat | 15.4   | 5.2 |     |     | 3.4    | 5.2 |     |     | 1回 |
| 被験者 F | 本システム  | 3.8    | 4.5 |     |     | 4.7    | 2.8 |     |     | 0回 |
|       | VRChat | 8.1    | 4.6 |     |     | 5.9    | 4.6 |     |     | 0回 |
| 被験者 G | 本システム  | 5.2    | 6.3 |     |     | 2.5    | 2.7 |     |     | 1回 |
|       | VRChat | 16.2   | 7.8 |     |     | 2.8    | 3.6 |     |     | 0回 |

### 6.1.3 評価結果

それぞれの被験者の操作時間の4回の平均を表4に示す。表4において上家と自家、対面と下家はそれぞれ位置が近いいため実験データとして大きな違いが出ないと判断し、一部の被験者では実験を行わなかった。また、自摸った場所の区別を付けない場合の平均として上家と対面の合計8回の平均を表5にそれぞれ示す。表4には実験中、ミスした回数も示す。ミスとは、自摸る際に本来自摸るべき牌以外を持ってしまうことである。

表6にはそれぞれの操作方法に対しての満足度のアンケートを示す。1~5段階評価で数字が大きいほど良い評価である。

表 5: 上家と対面に関し平均した場合の自摸・打牌の操作時間(秒)

|       |        | 自摸   | 打牌  |
|-------|--------|------|-----|
| 被験者 A | 本システム  | 5.1  | 2.4 |
|       | VRChat | 7.0  | 2.6 |
| 被験者 B | 本システム  | 4.6  | 2.0 |
|       | VRChat | 6.1  | 1.6 |
| 被験者 C | 本システム  | 5.6  | 2.9 |
|       | VRChat | 8.7  | 5.2 |
| 被験者 D | 本システム  | 3.6  | 2.0 |
|       | VRChat | 9.6  | 4.6 |
| 被験者 E | 本システム  | 5.1  | 2.5 |
|       | VRChat | 10.3 | 4.4 |
| 被験者 F | 本システム  | 4.1  | 3.7 |
|       | VRChat | 6.4  | 5.2 |
| 被験者 G | 本システム  | 5.7  | 2.6 |
|       | VRChat | 12.0 | 3.2 |

表 6: 自摸・打牌の満足度の評価

| 対象    |        | 快適性 | 正確性 | 直観性 | 好感度 |
|-------|--------|-----|-----|-----|-----|
| 被験者 B | 本システム  | 3   | 3   | 5   | 4   |
|       | VRChat | 4   | 4   | 4   | 4   |
| 被験者 C | 本システム  | 4   | 3   | 3   | 3   |
|       | VRChat | 3   | 4   | 4   | 3   |
| 被験者 D | 本システム  | 4   | 3   | 5   | 4   |
|       | VRChat | 3   | 4   | 3   | 3   |
| 被験者 E | 本システム  | 3   | 2   | 5   | 5   |
|       | VRChat | 2   | 4   | 4   | 5   |
| 被験者 F | 本システム  | 4   | 4   | 5   | 4   |
|       | VRChat | 3   | 4   | 3   | 3   |
| 被験者 G | 本システム  | 4   | 2   | 5   | 4   |
|       | VRChat | 2   | 5   | 2   | 2   |

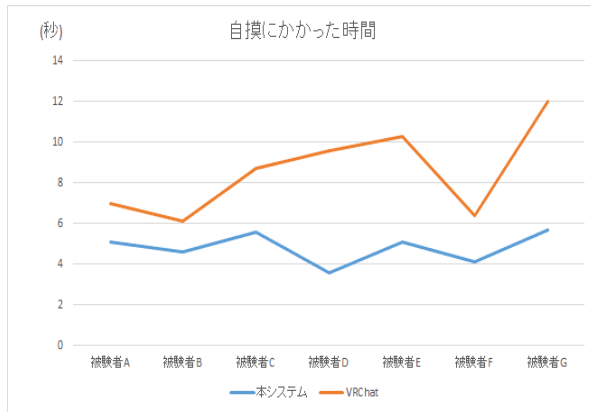


図 50: 自摸にかかった時間の比較

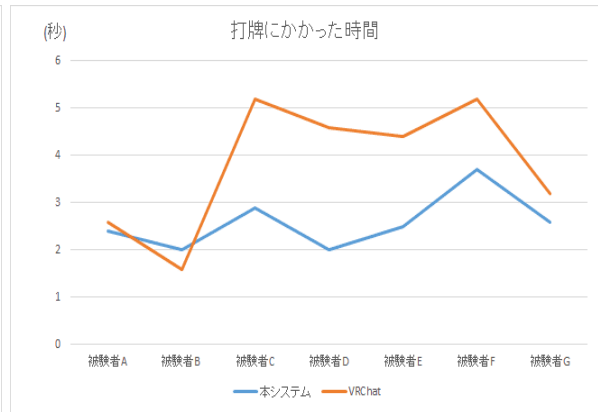


図 51: 打牌にかかった時間の比較

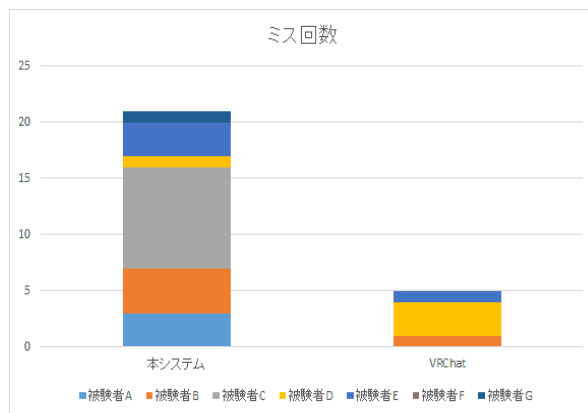


図 52: ミス回数の比較

自摸にかかった時間と打牌にかかった時間の比較をそれぞれ図 50, 図 51 にグラフで示す。また, ミスをした回数の合計の比較を図 52 に示す。

満足度の人数比を図 53 に示す。



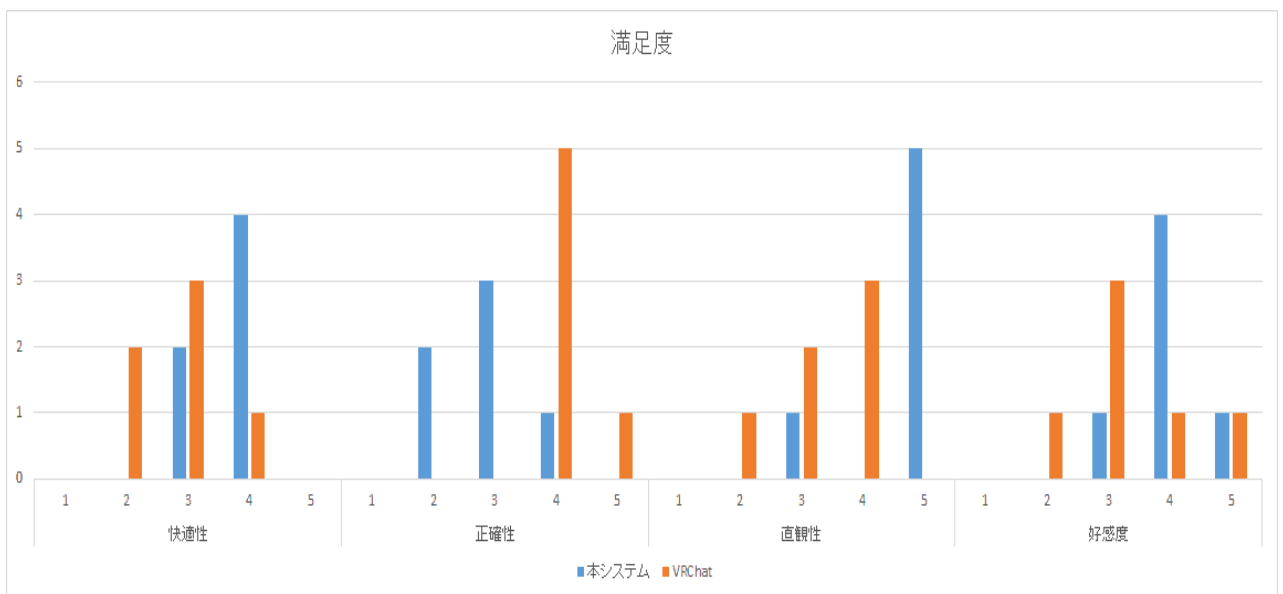


図 53: 満足度の比較

#### 6.1.4 考察

##### 操作時間について

図 50 と図 51 のグラフより、自摸にかかった時間は本システムを用いた方が VRChat のシステムより短くなっている。また、打牌の時間もほとんどの被験者が本システムを用いた場合の方が短い結果となっている。本システムを用いた自摸にかかった時間は 4 秒から 5 秒程度となっている。一方で VRChat を用いた場合は 7 秒から 8 秒程度と本システムより 3 秒程度時間がかかっている。しかし、人によっては 10 秒以上自摸にかかっていることもある。本システムの打牌にかかる時間は 3 秒程度で自摸よりは短くなっている。VRChat を用いた場合では長くて 5 秒程度で、人によっては本システムと変わらなかったり速い場合もある。

全体的に自摸、打牌ともに本システムの方が速く行うことができている。本システムと VRChat のシステムの違いの一つに牌の角度を気にしなければならないという点がある。自摸や打牌する際に、本システムでは手牌に持ってきた牌をそのまま放せばよいところ、VRChat のシステムでは角度を変えるために手首を捻り牌の角度を調整する必要がある。しかし、打牌時にも角度の調整が必要であるにもかかわらず、打牌時間は本システムと VRChat では大きな差はない。よって、角度の調整が本システムとの大きな差になっているとは考えにくい。

実験中、被験者が時間をかけていた部分は牌を取るまでの部分である。VRChat のシステムは持つことができるオブジェクトに接続線が発生する。この線ができていないオブジェクトを持つことができるが、牌山のように選択可能オブジェクトが細かく並んでいると近くの別オブジェクトを選択してしまう。このとき、直ぐに持ちたいオブジェクトに選択を移せばよいが、手元がブレることで周りの牌を行ったり来たりすることになりなかなか選択することができない。さらに、接続線はかなり早い段階から出現してしまう。その上、持ちたい牌にコントローラを近付けていったとしても接続線が近い牌に変わるわけではなく、再選択の位置は接続線が出現するところである。そのため、少し遠い距離でコントローラを微妙に動かし続け接続線の選択を変更するしかない。10 秒以上かかっている被験者の原因はこれにあたると思われる。一方で、打牌の場合は牌が一行に並んでいる上に、自摸った牌以外ではどれを打牌してもよいため、選択にあまり時間を取られないことが大きな差にはなっていない要因になっていると考えられる。これは、特に操作に慣れていない被験者に多くみられた。

操作方法に慣れていない被験者でも、本システムのほうが数秒短い。角度調整、牌選択以外の時間がかかる要因として、対面等の遠い牌を取りに行くときの時間が考えられる。本システムでは手を伸ばすだけで自摸することができるが、VRChat ではそのままでは牌が

遠すぎるため取りに行くことができない。そのため、被験者自身が一步前へ進んで、自模り、一步下がるという工程が必要となる。このような動作の積み重ねが本システムとの差になっていると考えられる。

自摸や打牌は麻雀というゲームの中では一番多く行われる行動である。一回につき2秒ほど多くかかると、それだけテンポが悪くなり面白さを損ねる原因になると考えられる。反復動作を短い時間で行えるのは本システムの大きな利点であると考えられる。

### ミスについて

図 52 のグラフより、操作ミスの回数は本システムを用いた方が、VRChat のシステムより 4 倍多い。これは、牌を持つ前の段階に視覚的フィードバックが存在するかが大きな原因の一つとなっていると考えられる。

VRChat のシステムでは、持つことができるオブジェクトに接続線とハローのフィードバックが入る。そのため、持つべき牌をわかった上で保持動作に入ることができるため、誤って違う牌を持ってしまう可能性が低くなっている。しかし、この機能が牌を持つための時間がかかってしまう原因にもなっている。VRChat ではこのフィードバックがあってもミスが少しあるが、これは持つタイミングで手振れ等で違うオブジェクトを持ってしまう現象による。特に、なかなか持つべき牌を選択できなかった場合などに起きる。

本システムには視覚的フィードバックがない。さらに、触覚フィードバックもないためピンチ動作を行い牌を持つ前にどの牌を持つことになるのかわかりづらい。ピンチした場所から一番近いオブジェクトを保持するスクリプトになっているが、遠くにある牌を持つ場合は手と牌が重なりあうためわかりづらい。その上、牌同士は非常に近くに隣り合っているため、持つべき牌の隣の牌を持ってしまうというミスが多発することになる。また、手が牌山に触れると牌山が崩れてしまうというデメリットがある。VRChat のシステムでは掴む、放す以外の方法でオブジェクトに接触することができないようになっていたのでこの心配はないが、本システムでは手全体に接触判定があるためこの現象が起きてしまう。現実でも、牌山に手を触れると崩れてしまうため臨場感があるが、現実と違い牌山を積みなおすのも容易ではないためテンポが悪くなってしまう。この問題は牌山の接触判定を自摸時以外になくすことで解消することはできる。

打牌時に違う牌を持ってしまうことがあっても、手牌内で牌を持ち直せばいいため大きな問題にはならないが、自摸時に違う牌を持ってしまうことは大きな問題になると考えられる。そのため、VRChat より 4 倍ほどミスがある状況はよくないと考えられる。このミスに関しては、できる限りゼロにすることが望ましい。現実で自摸りたい牌の隣の

牌を誤って自摸ってしまうというシチュエーションはほぼありえないが、現行のシステムでは、熟練者がどんなに気を付けたとしても一定のミスが出てしまうため解決の必要がある。

ミスを減らす方法として、牌山の接触判定を自摸る牌以外は無くしてしまうという方法がある。次に自摸る牌はゲーム中常に決まっているため、打牌された段階で次に自摸られる牌の接触判定を有効にするという方法を取ることで自摸時に他の牌を自摸ってしまうことを防ぐことができる。しかし、この方法はゲームの進行方法による対処法なので、オブジェクトを間違えて持ってしまうという問題点の根本的な解決にはなっていない。他の方法として、牌を保持する前に視覚的フィードバックを入れることが挙げられる。VRChatのシステムでは視覚的フィードバックが操作時間の問題にはいるが、ミスが起きにくい要因の一つが視覚的フィードバックの有無であることは確かである。問題は、早い段階で出てしまう接続線と選択の変更が利きづらい点である。本システムは指で掴むという性質上、早い段階から視覚的フィードバックを入れる必要がない。基本的に持ちたいオブジェクトとゼロ距離になった段階で持つはずである。そのため、色付けやハローのフィードバックを入れることでミスを減らすことは可能であると考えられる。また、手を近付けるだけで選択を変更できることから、視覚的フィードバックの影響で操作時間が長くなることはないと考えられる。さらに、手振れにより違う牌を掴んでしまうことを無くすためにオブジェクトを大きくする方法も考えられる。

## 満足度について

図 53 のグラフより、満足度のそれぞれの項目について考察する。

快適性については、本システムを用いた方が好印象であった。VRChatのシステムではよくない評価を下した人も一定数いた。これは、VRChatの視覚的フィードバックによる選択の難しさが原因であると考えられる。また、角度を調整するひと手間も要因の一つであると考えられる。VRChatのシステムではVIVEコントローラを常に両手で持っている状態であるため、両手が塞がる点でも快適性の評価を下げていると考えられる。本システムでは、良い評価の方が多いがVRChatのシステムと比較して飛びぬけて良いというわけではない。選択はスムーズに行うことができるが、Leap Motionの検出範囲の問題により手のトラッキングがうまくいきづらい点が要因であると考えられる。特に、遠くのものを取りに行くときにはトラッキングが消えやすい。頭を手前に動かし手とLeap Motionを近づけることで解消することができるが、逆に言えば頭を手前に動かす必要があるということで、この点は評価を下げていると考えられる。

正確性については、VRChatのシステムが非常に高評価で、本システムはあまりよくな

い評価であった。選択に難はあるが、正確に持ちたいものを持つという点では視覚的フィードバックのある VRChat のシステムの方が印象がよいと考えられる。また、本システムでは牌山が崩れてしまう等が評価が低い原因であると考えられる。

直観性については、現実の掴む動作をそのまま使っている本システムの方がよい評価となっている。また掴み方のコツとして、オブジェクトを持つのではなく、近くで指で輪を作るイメージを被験者に話していたがほとんどの被験者が行ってくれていた。VRChat のシステムも低い評価が多いわけではないが、VIVE コントローラを用いるためボタンの使い方等説明が必要な部分があり、本システムより低くなっていると考えられる。

好感度について、本システムの方が評価が良い。他の満足度の評価から、快適性は本システム、正確性は VRChat、直観性は同程度として考えると快適性と正確性が同じくらいの好感を得られていると考えられる。視覚的フィードバックにより選択に難が出ている点と持つ牌がどれかわからない点は相殺しあっていることがわかる。麻雀に限らず VR でゲームをプレイする場合どちらでプレイしてみたいかという質問の仕方をしたが、若干本システムのほうが良い評価のところから、正確性よりも快適性を求めていることがわかる。

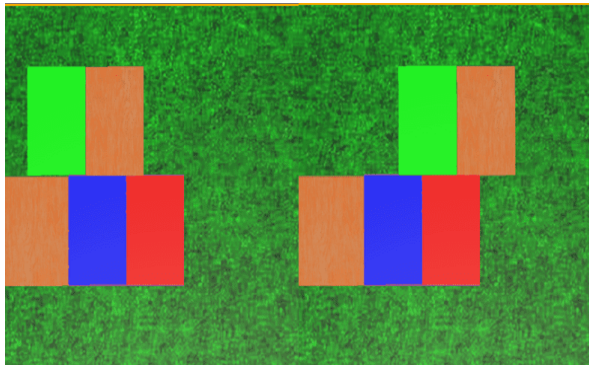


図 54: 牌同士の距離 0.025

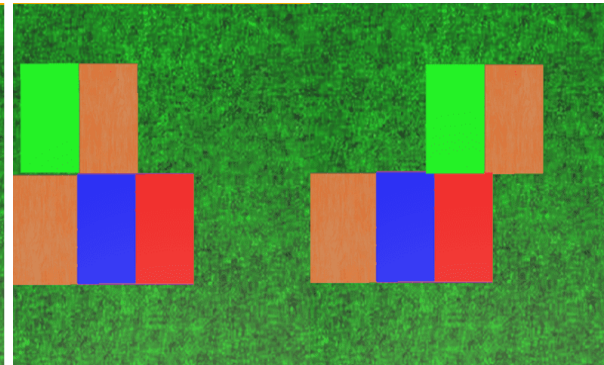


図 55: 牌同士の距離 0.030

## 6.2 イカサマシステムの評価

実装したイカサマシステムの難易度を変えて麻雀ゲームに適するバランスを探る実験を行った。実験条件として、位置精度を3種類、制限時間を3種類の合計9種類の組み合わせに対し、イカサマがどの程度成功したかによって得点をつけた。一つの組み合わせにつき三回イカサマを行う。

位置精度は、牌同士の距離を基準として、表7のようにしてミス进行定義した。

### 6.2.1 実験内容

実験は、被験者と被験者の動作を監視する観測者で行った。被験者は、赤く着色された牌を自摸することで実験を開始する。手牌に加えたタイミングで右側から板がスライドしてくる(図58)。これは、観測者の視界を遮るもので、イカサマ動作自体を隠す効果はないが、その前の牌を二牌持つ動作を隠す役割がある。観測者に牌を二牌持っていることを見られないように被験者はイカサマする必要がある。観測者が二牌持っていると判断するとフィードバックを行い、被験者の画面に警告文字が入るようになっている(図59)。

制限時間は、10秒、7秒、4秒の3種類について実験した。位置精度と制限時間の組み

表7: 位置精度

| 判定     | ミス無し            | 小ミス                | 大ミス             |
|--------|-----------------|--------------------|-----------------|
| Easy   | 0.025 未満 (図 54) | 0.025 以上, 0.030 未満 | 0.030 以上 (図 55) |
| Normal | 0.020 未満 (図 32) | 0.020 以上, 0.023 未満 | 0.023 以上 (図 33) |
| Hard   | 0.019 未満 (図 56) | 0.019 以上, 0.020 未満 | 0.020 以上 (図 57) |

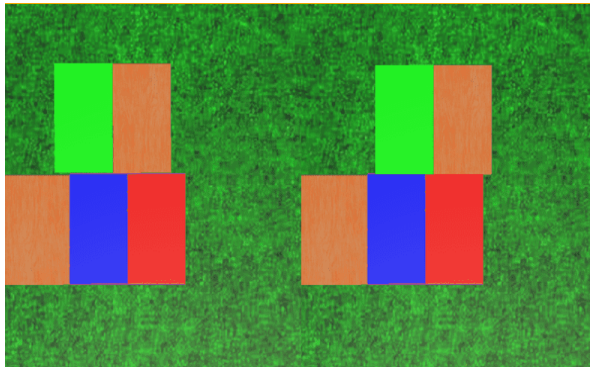


図 56: 牌同士の距離 0.019

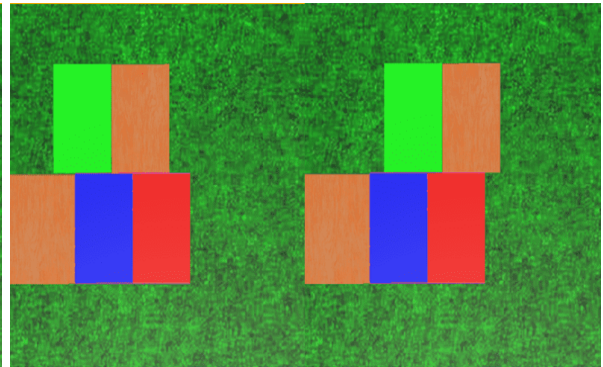


図 57: 牌同士の距離 0.020

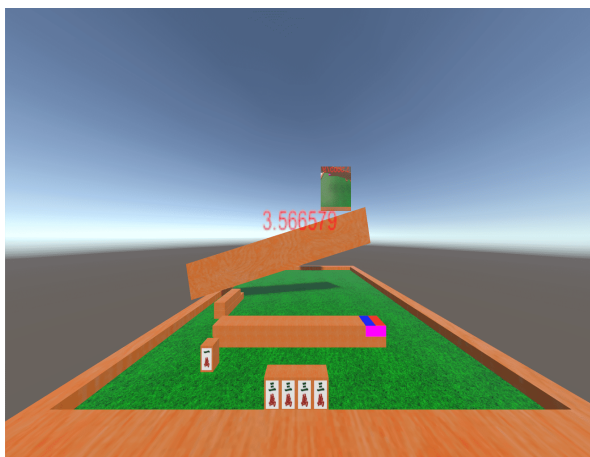


図 58: 板がスライドした状態



図 59: 警告文字が入った状態

合わせは、(位置精度 Easy, 制限時間 10 秒) → (位置精度 Easy, 制限時間 7 秒) → (位置精度 Easy, 制限時間 4 秒) → … → (位置精度 Hard, 制限時間 4 秒) の順で行った。

実験手順は以下の通りである。

1. 牌の操作方法, イカサマ動作のやり方の説明を行い, 操作方法に慣れてもらう
2. オブジェクトを実験用の配置にする
3. 左にある山から牌を一つ自模り, 手牌に加える
4. 対面の視界を塞ぐ板がスライドしてくるので, スライドしきったタイミングでイカサマ操作を行う
5. スライドしきる前に牌を二つ持った場合, 警告を与える
6. 3~5 を 3 回繰り返す
7. 時間内にイカサマ出来たか, 山が崩れたかを記録する
8. 全ての位置精度, 制限時間のパターンで 2 から繰り返す

### 6.2.2 評価結果

制限時間の結果に関しては, 時間点として, 制限時間内に終了していた場合 1 点, していなかった場合 0 点. さらに, 警告を貰っていたら一回につき -0.5 点を与える. それぞれの組み合わせについて 3 回の平均を取った.

山の崩れ方に関しては, 山崩れ点として, 山が崩れていなかった場合 1 点, 小さく崩れていた場合 0.5 点, 大きく崩れていた場合 0 点を与える. それぞれの組み合わせについて 3 回の平均を取った.

結果を以下の表 8 に示す. 各制限時間ごとの全被験者の時間点の平均と, 位置精度を比較したグラフを図 60 に示す. さらに, それぞれの時間点を位置精度の区別なく比較したグラフを図 61 に示す. 各難易度ごとの全被験者の山崩れ点の平均と, 制限時間を比較したグラフを図 62 に示す. さらに, 各難易度ごとの山崩れ点の比較を図 63 示す.



表 8: イカサマの評価結果

| 制限時間  |      | 10 秒 |        |      | 7 秒  |        |      | 4 秒  |        |      |
|-------|------|------|--------|------|------|--------|------|------|--------|------|
| 位置精度  |      | Easy | Normal | Hard | Easy | Normal | Hard | Easy | Normal | Hard |
| 被験者 A | 時間点  | 1.0  | 1.0    | 1.0  | 1.0  | 0.67   | 1.0  | 0.33 | 0.67   | 0.67 |
|       | 山崩れ点 | 1.0  | 1.0    | 1.0  | 1.0  | 1.0    | 0.5  | 0.83 | 0.67   | 0.5  |
| 被験者 B | 時間点  | 0.67 | 1.0    | 0.83 | 0.33 | 0.83   | 1.0  | 0.67 | 1.0    | 1.0  |
|       | 山崩れ点 | 0.83 | 0.83   | 0.5  | 0.83 | 0.67   | 0.33 | 1.0  | 0.83   | 0.33 |
| 被験者 C | 時間点  | 0.67 | 0.67   | 1.0  | 1.0  | 1.0    | 1.0  | 0    | 0      | 0    |
|       | 山崩れ点 | 0.5  | 1.0    | 0.33 | 1.0  | 0.67   | 0.5  | 1.0  | 1.0    | 1.0  |
| 被験者 D | 時間点  | 0.83 | 1.0    | 1.0  | 0.5  | 1.0    | 0.83 | 0.33 | 1.0    | 0.67 |
|       | 山崩れ点 | 0.67 | 1.0    | 0    | 0    | 0.67   | 0    | 0.83 | 0.83   | 0    |
| 被験者 E | 時間点  | 0.83 | 0.83   | 1.0  | 0.5  | 0.83   | 1.0  | 0.67 | 0.67   | 0.67 |
|       | 山崩れ点 | 0.67 | 0.5    | 0.5  | 1.0  | 0.5    | 0.5  | 1.0  | 0.5    | 0.33 |
| 被験者 F | 時間点  | 0.67 | 1.0    | 1.0  | 1.0  | 0.67   | 1.0  | 0.67 | 0.33   | 0.67 |
|       | 山崩れ点 | 1.0  | 0.5    | 0.67 | 0.83 | 0.17   | 0    | 0.83 | 0.33   | 0    |

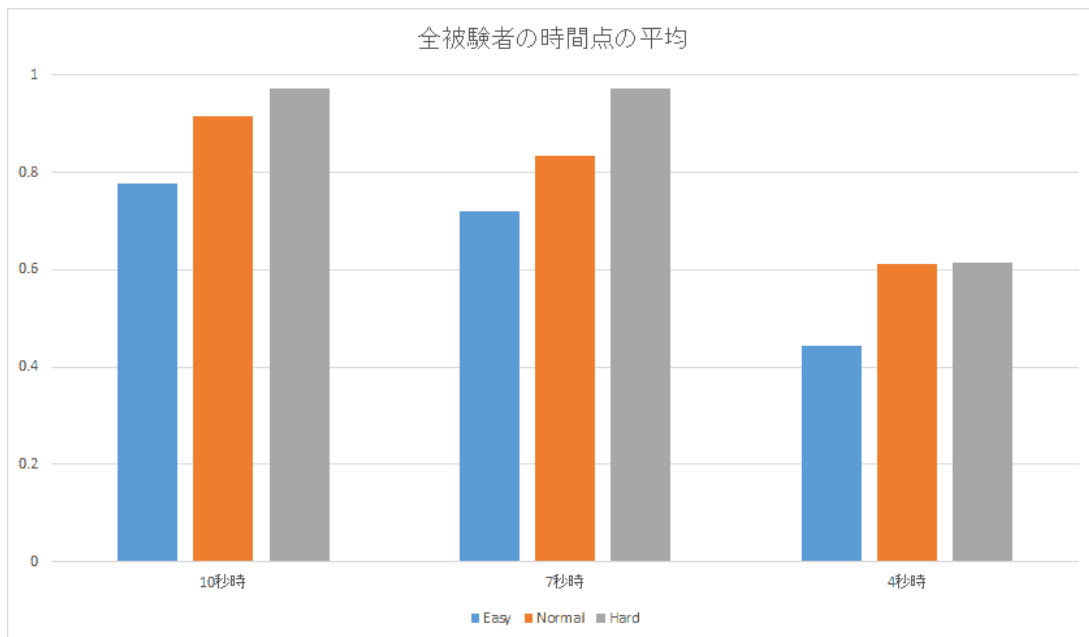


図 60: 時間点の平均と位置精度の比較

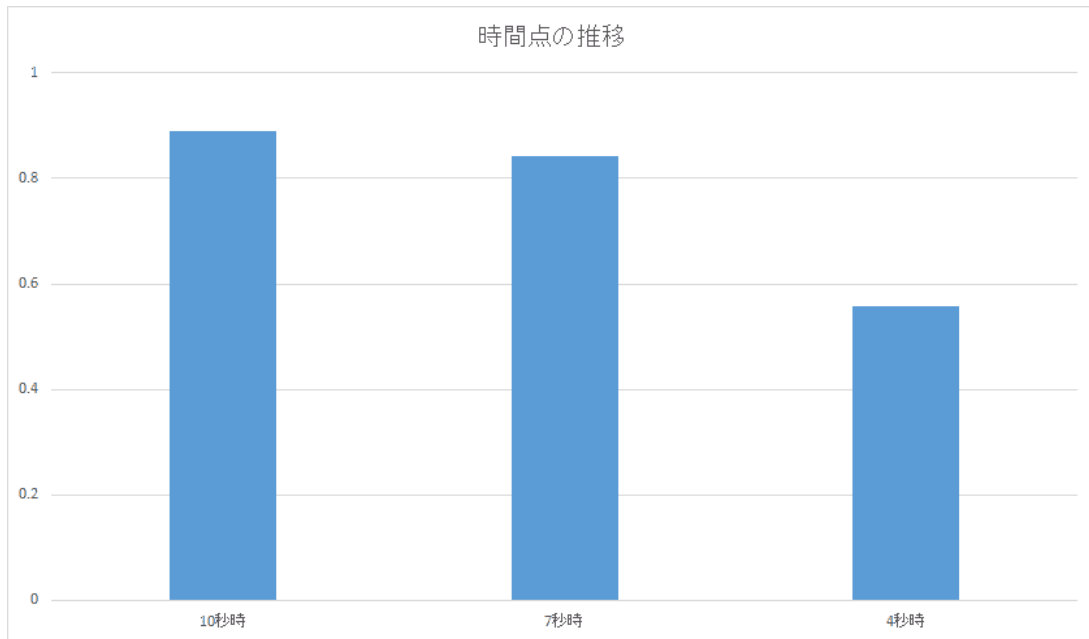


図 61: 時間点の制限時間毎の比較

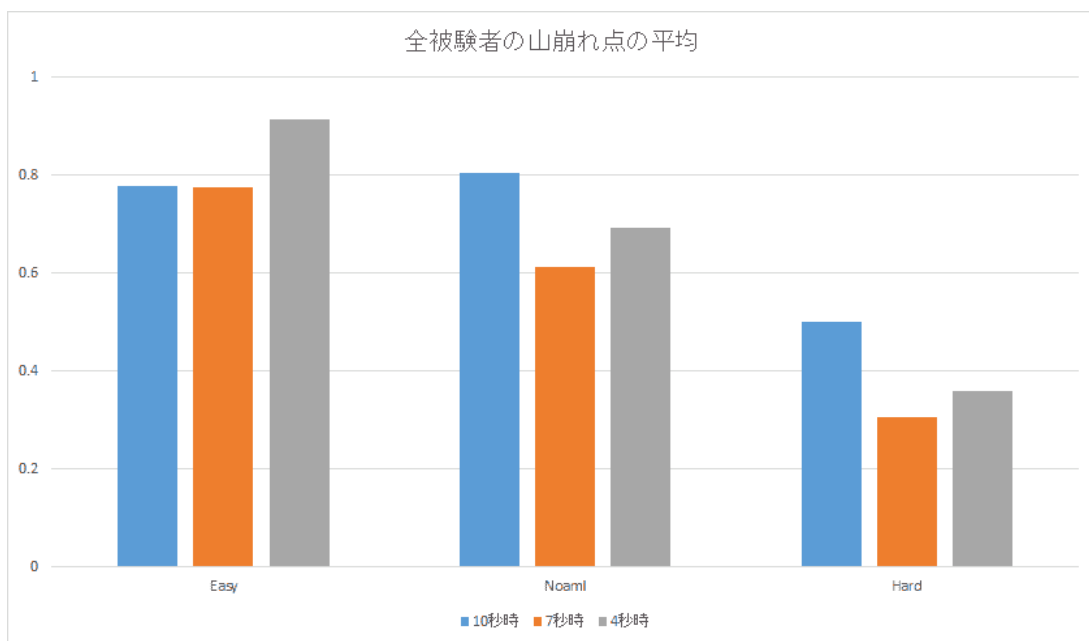


図 62: 山崩れ点の平均と制限時間の比較

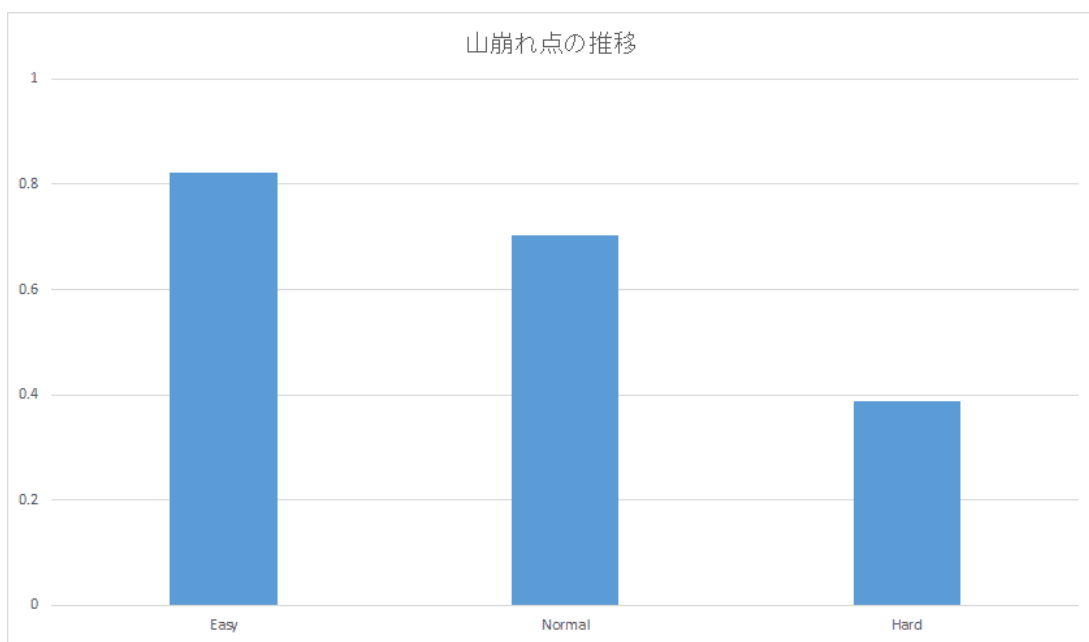


図 63: 山崩れ点の位置精度ごとの比較

### 6.2.3 考察

#### 時間内にイカサマを終了しているかについて

時間点の大きさについては、制限時間が短いほど時間点は小さいと予測できる。また、難易度との関係は難易度が難しくなれば時間点が小さい可能性がある。

図 60 より見られる時間点と位置精度の関係として、直感に反して位置精度が上がると時間点が大きくなるというものである。個人間で見れば全員がこの傾向を示しているわけではないが、平均では全ての時間においてグラフが右肩上がりのもが多く、位置精度が上がると時間点が増えているのがわかる。しかし、これは位置精度によるものではなく評価実験の順番による部分も大きいのではないかと考えられる。Easy 難易度から順に実験を行ったため、最初の方は被験者が時間のタイミングを掴めず、やっていくうちに慣れてきているのではないかと考えられる。

直感に反する他の要因として、被験者が山を崩さないように慎重に行う度合いによるのではないかと考えられる。Easy 難易度では、簡単が故にゆっくり丁寧に行い出来る限り山が崩れないように意識するため時間内に終了できない状況が生まれる。一方で、Hard 難易度ではどれだけゆっくりやったとしても結果崩れてしまうため多少雑に行っているという部分もあると考えられる。この場合、Hard 難易度が難しすぎるのが問題となり、かなりの熟練度が必要となると容易にイカサマを行えるサポートの意味が無くなってしまおうと考えられる。

図 61 は制限時間ごとに分けた全ての位置精度での時間点の平均である。制限時間が短くなるにつれて時間点が減ってきてはいるが、10 秒と 7 秒のときに大きな違いが見られない。10 秒の方が時間点が大きいのが、7 秒のときも制限時間内に間に合っていると考えられる。また、4 秒のときは時間点が 0.5 強であるから、制限時間内に成功しているのは半分程度であると考えられる。このことから、制限時間は長くて 7 秒あれば十分であり、4 秒では少し短すぎる可能性がある。本実験中では、イカサマを主に行っているため、イカサマ行為にのみ集中することができる。実際に麻雀ゲームを行う場合にはもっとイカサマ成功率は下がるであろう。

#### 山の崩れ方について

イカサマがばれないためには山崩れ点が 1 である必要がある。図 62 より山崩れ点はそれほど制限時間と大きな関係はないと考えられる。制限時間が長いとしても山崩れ点が良くなるという傾向が見られるわけではなく、短いから山崩れ点が低くなるとは言えない。これに関して、ユーザが「制限時間内に終わらせることを優先」させるか、「山を崩さないように行うことを優先」させるかという個人の考え方によるものであると考えられる。

制限時間や位置精度が変化してもイカサマ動作に変化を与えず、一定の行動速度で実験を行う被験者がいた。制限時間が変化しても一定の速度で行うため、位置精度が上がると全く山崩れ点が伸びない代わりに時間点が高い。制限時間が短い時は丁寧にイカサマをしようとするため山崩れ点が高いが、時間点が高い。本実験の性質上本当のゲームを行っている訳ではないので、「制限時間内に終わらせる」「山を崩さないようにする」ことに対する緊張感がなかった可能性がある。

図 63 より位置精度が上がるごとに山崩れ点は減っている。Easy から Normal への推移はあまり大きくないが、Hard では半分以下になっている。Easy 難易度では平均で 8 割以上の得点率があることから熟練度によるスリルを出すことが難しいと考える。逆に、Hard では難しすぎるためイカサマを行うことが困難ではないかと考える。山崩れ点が 0.5 という状況ではイカサマを行うことでほぼ小さく崩す以上のことが起きていると考えられる。成功することを前提とするならば、0.7 点の Normal に設定し、今後熟練して 1 点を目指すくらいに設定することが考えられる。しかし、イカサマは現行犯で押さええないといけない点から小さく崩す場合ならば、すぐにはばれない可能性がある。その場合 1 点を目指す必要はない。その上、被験者は本システムのイカサマの初心者である。初心者がほぼ成功する状態では熟練度によりスリルは味わえないと考える。そのため、Hard よりもう少しだけ緩和された難易度がちょうどいいという考え方もある。一方で、友人同士で遊ぶ場合イカサマの失敗による指摘も面白さに繋がる部分であると考えられる。そのため、ある程度失敗前提で Hard でプレイすることにも意味がある。

### ゲームに組み込む

麻雀ゲーム内へ本研究で実装したイカサマサポートシステムを組み込む場合の難易度のバランスを議論する。

イカサマを行う上で難しい点の一つとして、周りの他プレイヤーの視線である。他プレイヤーに警戒され常に見続けられるとどんな熟練者でもイカサマを行うことは困難となる。注意がそれたときにイカサマを行うべきだが、他プレイヤーに注意を配り続けるのも難しい。本研究では、制限時間の評価によりプレイヤーがどの程度の時間あればイカサマを行うことができるかを評価した。さらに他プレイヤーから見られないような不自然な妨害板を出現させた。もう少し自然な妨害として、一定時間他プレイヤーの視界を防ぐような光を出すことを考えた。これにより、他プレイヤーの注意がそれた状態を意図的に作り出すことができる。さらに、この視界妨害は他プレイヤー全員に行う必要はないと考える。例えば、自摸番のプレイヤーは自摸行為に集中しているはずであるため、こちらへの注意はそれている。さらに、自摸番のプレイヤーが自摸のために手を伸ばす

ことで他プレイヤーの視界を防ぐ役割を担う。以上のことから視界を防ぐ光は、一人ないし二人のプレイヤーへ行えば十分であることが分かる。視界を防ぐ時間であるが、長すぎるとゲームのテンポを損ねる可能性がある。そのため、7秒程度を基準とするのは難しく、4秒程度を基準とするのがいいと考えた。4秒時の時間点から半分近くが間に合っていないことになる。しかし、熟練度により速くできる可能性を考慮するとこの程度がいいのではと考えた。さらに、自摸にかかる時間が4秒から5秒であることを考慮すると、長くても5秒程度が限界であると考えた。

位置精度については、Hard 難易度では難しすぎる可能性はあるが、熟練度により解消できる可能性もある。また、ある程度崩れる前提のほうが友人内でプレイする場合は面白い可能性もある。しかし、視界を防ぐ時間がイカサマ可能時間であることを考えると4秒強では素早く行う必要があるため、山を崩さず行うことは難しいのではないかと考える。そのため、位置精度は Normal が適当ではないかと考えた。イカサマを成功することを前提とし、ミスが起きたとしても小さく崩れるで済むことで、イカサマがばれずにごまかせる可能性が残っている方が良く考えた。

複数人でプレイするときの問題点として、光等による妨害を行っている間にイカサマを行うため、手元が見えていなくてもイカサマを行っていたということがばれてしまうことである。しかし、この問題は妨害を行う人数を減らすことで解決することができると考えられる。例えば、一人にのみ妨害を行い、その人物が妨害があったことを訴えたとしても残りの3人にはそのような事実がないため白を切り通すことができる。このように妨害を行う人数を減らすことで、妨害を行ったことでイカサマが露見することを防ぐことができる。また、本実験で行った妨害板は視界を遮っているプレイヤー以外からも見えてしまうため、非常に不自然となる。そのため光等による妨害方法の改善を行うが、このとき光は視界を遮るプレイヤーのみに見えるものとするすることで他プレイヤーから見たときの不自然さを解消できると考えた。

## 6.3 デモンストレーション時の評価

### 6.3.1 EC2018 でのデモンストレーション

2018年9月13日から15日にかけて開催されたエンターテインメントコンピューティングシンポジウム2018(EC2018)へ参加し、14日にデモンストレーション発表を行った。ユーザは Leap Motion を前面に装着した HTC VIVE を被ってもらい、Leap Motion を用いた VR 空間での牌操作とイカサマ操作を行ってもらった。ユーザには、手牌にある牌を一つ掴み、操作に慣れたら二つ目を掴んでもらう。その後、イカサマ操作をしてもらうというデモンストレーションを体験してもらった。

## 評価結果

10名程度のユーザに体験を行ってもらった。最初、多くのユーザは牌を持つときに現実と同じように親指と人差し指で挟むように持つ傾向にあり、持っている途中で牌を落としてしまうことが多かった。その後、指で輪を作るようにしてもらおうようアドバイスをしたらほとんどのユーザが途中で落とすことが無くなった。また、イカサマ操作に関しては牌の持ち方からすり替えまでほとんどのユーザが簡単に行うことができた。しかし、難易度の設定がきちんとなされていなかったため、山が崩れることがほとんどなく、只々牌を入れ替えるだけということになってしまっていた。そのため、面白さが伝わりにくい結果になっていた。

## EC2018 後の研究

位置精度を見直し、熟練度判定をきちんに行えるように調整した。また、イカサマを麻雀ゲーム内に組み込んだ場合に適するバランスを探るために難易度を複数用意し、評価を行えるような実装を行った。

### 6.3.2 電気通信大学オープンキャンパスでのデモンストレーション

2018年11月25日の電気通信大学オープンキャンパスにて研究のデモンストレーション発表を行った。ユーザは Leap Motion を前面に装着した HTC VIVE を被ってもらい、Leap Motion を用いた VR 空間での牌操作を行ってもらった。牌山から牌を自摸り、手牌に持ってきたのち、打牌する操作を行ってもらった。

## 評価結果

10名程度のユーザに体験を行ってもらった。牌山から牌を自摸ってもらうため、頭を動かすことで顔を近づけて、Leap Motion の検出範囲に入れる方法を行ってもらったが指示が通りにくかった。しかし、自摸や打牌動作は問題なく行ってもらうことができた。また、牌を持つ方法を最初から輪を作るように持つと説明をすることで、VR 空間ではそういう風に牌を持つものだとして認識してもらうことができ、すぐに牌操作に慣れてもらうことができた。しかし、牌山から自摸る際手を前に突き出すときに、手牌の牌を誤って持った状態になってしまいますことが多々見られた。これは、人が平常時に指を軽く曲げていることが原因で Leap Motion がピンチと誤認識してしまうことが原因である。そのため、自摸の際は手をパーの状態の手開いた状態にってもらおうようなアドバイスを行った。

## 7 まとめと今後の課題

### 7.1 まとめ

本研究では、Leap Motion を用いた VR 空間で麻雀ゲームを行うためのシステムを実装した。Leap Motion を用いた牌の操作方法やゲーム内において必要な動作を実装した。牌操作に関して、既存の VRChat での麻雀システムと比較し、操作速度や操作の快適性について良い結果が得られた。一方で、持ちたい牌以外を持ってしまうミスが目立ち、正確性では良い結果を得られることが出来なかった。

また、Leap Motion を使ったイカサマサポートシステムを実装し、ゲームに組み込む場合のバランスについて考察した。他プレイヤーの注意を逸らすために、視界に光の妨害を入れることを検討し、その時間に関して考察した。時間は4秒程度とすることで、ゲームのテンポを損なうことなくゲームを行えるとした。さらに位置精度に関しては、Hard 難易度では難しいとし、時間が短いことも考慮し Normal 難易度が適当であるとした。しかし、友人内で山を崩す前提の面白さを追求する場合は Hard 難易度も良い可能性がある。

### 7.2 問題点

牌操作に関して、「持ちたい牌以外を持ってしまうことが多い」「遠くの牌を取る場合頭を前に動かす必要がある」ことが挙げられる。また、麻雀ゲームに必要な動作を独立して実装したため、一つの流れで行うことが困難な点がある。

イカサマサポートシステムについて作成した機能のバランスについて探したが、この機能を用いれば面白くなるかの検討が十分になされていない。また、イカサマ対象の牌のうち山にある牌はあらかじめ決めてある状態のため、好きなどころの牌をすり替えられない点がある。

### 7.3 今後の課題

牌操作の正確性を上げる必要がある。ミスをなくすために視覚的フィードバックの色付けやハロー等を実装することで、どの牌を持つことになるか事前にわかるようにすることで誤操作を減らすことができる。

麻雀ゲームとして遊べるように、機能を充実させる必要がある。積んだ牌を配る部分を実装出来ていないので、その部分の実装を行う。また、多人数が部屋にログインできるような機能の実装を行う。その他、ゲームをするための細かい機能の実装を行う必要がある。

本研究では、すり替えのみができるようなシステムを実装したが、他のイカサマを行



えるようにするのも課題である。面白さを追求するには、まず実際にゲーム内で遊べるようにすることが重要である。

## 参考文献

- [1] 株式会社コニカミノルタ, "VirtuaLink - VR 体験 in TOKYO SKYTREE TOWN | コニカミノルタ - VR サービス"(online), <https://planetarium.konicaminolta.jp/VirtuaLink/TokyoSkyTreeTown/>, (最終アクセス 2018 年 8 月 1 日).
- [2] "Cardboard - Google VR", [https://vr.google.com/intl/ja\\_jp/cardboard/](https://vr.google.com/intl/ja_jp/cardboard/)(online), (最終アクセス 2019 年 1 月 14 日).
- [3] Robert Gabriel Lupu, Nicolae Botezatu, Florina Ungureanu, Daniel Ignat : "Virtual Reality Based Stroke Recovery for Upper Limbs Using Leap Motion", 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), October 13-15, Sinaia, Romania, pp.295-299 (2016).
- [4] Mikhail Giorgini, Jacopo Aleotti : "Visualization of AGV in Virtual Reality and Collision Detection with Large Scale Point Clouds", 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), July 18-20, Porto, Portugal, pp.905-910 (2018).
- [5] Yi-Kai Juan, Hsing-Hung Chen, Hao-Yun Chi : "Developing and Evaluating a Virtual Reality-Based Navigation System for Pre-Sale Housing Sales", Appl. Sci., Volume 8, Issue 6, June 5, Taipei 106, Taiwan, pp.1-12 (2018).
- [6] 伊藤直紀, 橋本剛 : "VR 技術を加えた古典的 2D ゲームに関する考察", エンターテインメントコンピューティングシンポジウム (EC2016), AR/VR-3, (2016).
- [7] "麻雀のイカサマ技 玄人技 が凄すぎる【小島武夫 桜井章一のツバメ返し 積み込み】 - NEVER まとめ"(online), <https://matome.naver.jp/odai/2136275881794596101>, (最終アクセス 2019 年 1 月 14 日).
- [8] "第 1 回イカサマ麻雀選手権 東 1 局 (実況: モナカ王国 解説: OR) - ニコニコ動画"(online), <https://www.nicovideo.jp/watch/sm29921124>, (最終アクセス 2019 年 1 月 14 日).
- [9] "VIVE 日本 | 想像を超えたバーチャルリアリティの体験"(online), <https://www.vive.com/jp/>, (最終アクセス 2019 年 1 月 16 日).
- [10] "Steam へようこそ"(online), <https://store.steampowered.com/?l=japanese>, (最終アクセス 2019 年 1 月 16 日).

- [11] ”Oculus Rift” (online), <https://www.oculus.com/>, (最終アクセス 2019 年 1 月 27 日).
- [12] ”Oculus Quest |Oculus” (online), <https://www.oculus.com/quest/>, (最終アクセス 2019 年 1 月 27 日).
- [13] ”Gear VR | ウェアラブル - Galaxy Mobile Japan 公式サイト” (online), <https://www.galaxymobile.jp/gear-vr/>, (最終アクセス 2019 年 1 月 27 日).
- [14] ”Leap Motion” (online),<https://www.leapmotion.com/?lang=jp>, (最終アクセス 2019 年 1 月 16 日).
- [15] ”VRChat” (online), <https://www.vrchat.net/>, (最終アクセス 2019 年 1 月 17 日).
- [16] ”だったら VR で勝負だ！リアルと同じ感覚で遊べるバーチャル麻雀にハマろう【週間 VRChat】 - PANORA” (online), <https://panora.tokyo/64947/>, (最終アクセス 2019 年 1 月 10 日).
- [17] ” 【VRChat】VRChat 麻雀大会！-YouTube” (online), [https://www.youtube.com/watch?v=NUdaBrivTp0&feature=player\\_embedded](https://www.youtube.com/watch?v=NUdaBrivTp0&feature=player_embedded), (最終アクセス 2019 年 1 月 17 日).
- [18] ”イカサマ麻雀 - アイデアファクトリー | 製品情報” (online) <https://www.ideaf.co.jp/game/spec/?hard=ps&title=majyan>, (最終アクセス 2019 年 1 月 10 日).
- [19] ”PS Vita『咲-Saki-全国編』全キャラクター能力一覧 |Uunz” (online), [https://uunz.org/game/saki\\_zenkokuhen/all\\_characters\\_ability\\_list](https://uunz.org/game/saki_zenkokuhen/all_characters_ability_list), (最終アクセス 2019 年 1 月 24 日).
- [20] ”い か さ ま ゴ キ ブ リ” (online), [http://www.mobius-games.co.jp/DreiMagier/Mogel\\_Motte.html](http://www.mobius-games.co.jp/DreiMagier/Mogel_Motte.html), (最終アクセス 2019 年 1 月 24 日).
- [21] Spyros Vosinakis, Panayiotis Koutsabasis : ”Evaluation of visual feedback techniques for virtual grasping with bare hands using Leap Motion and Oculus Rift”, Virtual Reality (2018) 22:pp.4762 (2018).
- [22] ”Unity - Game Engine” (online), <http://japan.unity3d.com/>, (最終アクセス 2019 年 1 月 17 日).

- [23] "Orion — Leap Motion Developer"(online), <https://developer.leapmotion.com/orion/#105>,  
(最終アクセス 2019 年 1 月 17 日).
- [24] "LeapMotionCoreAssets/MagneticPinch.cs at develop ·  
leapmotion/LeapMotionCoreAssets · GitHub" (online),  
[https://github.com/leapmotion/LeapMotionCoreAssets/blob/develop/Assets/LeapMotion/  
Scripts/Utils/MagneticPinch.cs](https://github.com/leapmotion/LeapMotionCoreAssets/blob/develop/Assets/LeapMotion/Scripts/Utils/MagneticPinch.cs), (最終アクセス 2019 年 1 月 17 日).
- [25] "VRChat Home Kit"(online), [https://assets.vrchat.com/kits/VRChat\\_HomeKit\\_v1.unitypackage](https://assets.vrchat.com/kits/VRChat_HomeKit_v1.unitypackage),  
(最終アクセス 2019 年 1 月 17 日).

## 謝辞

本研究を進めるにあたり，ご指導いただいた主指導教員の成見哲教授，副指導教員の寺田実准教授及び実験に協力して下さった被験者の皆様に感謝いたします。