レイアウトと触感をカスタマイズ可能なゲームコントローラの提案

川尻 千遥 *　　塚田 浩二 *

概要. 本研究は，スマートフォンと触感シートを組み合わせて，レイアウトや触感を自由にカスタマイズできる汎用ゲームコントローラを提案する．既存のゲームコントローラは，ボタンの物理的な配置が固定されており，スマートフォンベースのコントローラでは触感フィードバックに乏しい問題があった．本研究では仮想ボタンを自由にレイアウトできる Android アプリと，そのレイアウト情報を元に造形できる触感シートを組み合わせることで，こうした問題の解決を図り，実用可能なゲームコントローラを構築する．

1 はじめに

PC ゲーム用コントローラには，ボタン・スティック等を組み合わせた多様な製品が存在する．しかし，プレイヤーの手の大きさやゲームジャンルによって最適なボタン配置は異なり，既存のコントローラではボタンの大きさ／配置などを自由にカスタマイズすることは困難であった．

一方，スマートフォンを PC のコントローラとして利用できるアプリケーション（例: Monect[1]）を用いれば，柔軟なボタンレイアウトに対応できる．しかし，ボタンは平坦な画面上に配置されるため，指先の感覚でボタンを捉えることは難しく，物理的なフィードバックに欠ける問題があった．

そこで本研究では，スマートフォンと触感シートを組み合わせて，レイアウトや触感を自由にカスタマイズできる汎用ゲームコントローラを提案する．仮想ボタンを自由にレイアウトできる Android アプリと，そのレイアウト情報を元に造形できる触感シートを組み合わせることで，こうした問題の解決を図る．提案システムを用いることで，ユーザ個人の特性や，格闘ゲームや音楽ゲーム等の用途に応じた，多様なコントローラを作成できる．さらに，低遅延の PC コントローラとして実用できるように実装を工夫する．本稿では，システムの設計とプロトタイプの実装を中心に述べる．

2 関連研究

タッチパネルの入力体験を向上させるための製品や研究事例は多数存在する．

スマートフォン等のタッチパネルに張り付けて利用可能な導電性のボタン型アタッチメントは市販されている[2] が，コントローラのボタン配置とは連動しておらず，カスタマイズや取り付けには手間がかかる．研究事例としては、PuyoSheet and PuyoDots[1] が存在する．タッチパネルにソフトジェル状の透明フィルムを貼り付け，ボタンを押したような感覚を与える研究である．また，ShiftTouch[2] は，画面を覆う領域を少なく抑えつつ，電極を通じてスクリーンへの入力を拡張する入力方式を提案している．これらの先行研究は，タッチパネルに物理的なフィードバックを追加したり入力領域を拡張したりする点で有用である．一方，本研究では，ユーザがボタンを自由にレイアウトして，ボタン配置に合わせた触感シートを作れる点や，PC 用ゲームコントローラとして実用できる点に特徴がある．

3 提案手法

3.1 提案概要

本研究の概要を図 1 に示す．ハードウェアとしては Windows PC と Android スマートフォンを中心に構成され，両者は USB ケーブルで有線接続される．スマートフォンアプリ上では，ユーザは自由に仮想スティックやボタンをレイアウトすることができる．さらに，レイアウトをベースとして触感シートを造形し，保護フィルムのようにスマートフォンに張り付けることで，ボタンやスティック部分に触感を付与できる．次に，ソフトウェアと触感シートの詳細について述べる．

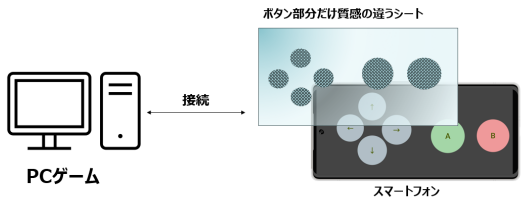

図 1．本研究の概要

Copyright is held by the author(s). This paper is non-refereed and non-archival. Hence it may later appear in any journals, conferences, symposia, etc.

* 公立はこだて未来大学

[1] https://www.monect.com/

[2] https://gametech.co.jp/smartphone/sgf2129-2/

3.2 ソフトウェアの構成

ソフトウェアの構成を図 2 に示す．大きく分けて，Android アプリと PC 側のサーバから構成される．Android アプリは「レイアウトモード」と「コントローラモード」を備える．レイアウトモードでは，ユーザはボタンやスティックを任意の場所／サイズで配置できる．さらに，配置したレイアウトのアウトラインデータを SVG 等で書き出すことで，後述する触感シートの造形に利用する．コントローラモードでは，配置したスティックやボタンに対するタッチ入力を検出し，その状態が変化したタイミングで WebSocket 通信を介して PC 上のサーバに情報を送信する．ボタンの場合はボタン ID と押した/離したという状態を送信する．スティックの場合は，指のドラッグ位置を，スティックの中心を原点とする整数座標（X， Y）に変換して送信する．PC 上のサーバは，受け取った入力を元に，仮想 Xbox コントローラの入力に変換し，一般的な PC ゲームを操作できる．なお，WebSocket 通信を利用しているため原理的には Wi-Fi 等を用いた無線環境でも動作するが，入力遅延を考慮して USB を用いた有線通信を採用している．

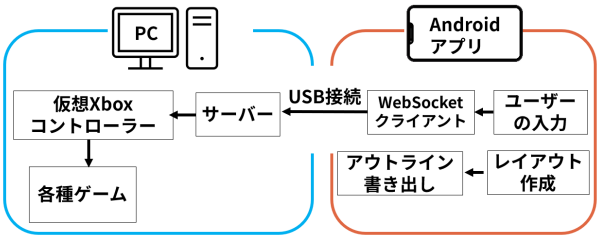

図 2. ソフトウェアの構成

3.3 触感シート

ここでは，アウトラインデータと触感シートについて紹介する（図 3）．前述したように，ユーザが作成したボタンレイアウトのアウトラインデータを出力する．このデータを Adobe Illustrator で読み込み，任意のテクスチャ(スウォッチ) を指定して，UV プリンタでグロスインクを用いて印刷する．保護フィルム上に印刷してスマートフォンに貼り付ければ，ボタン等の位置に任意の触感パターンを付与できる．このように，ユーザは自分のデザインしたコントローラに触感を付与して，指先の感覚でボタンやスティックの位置等を把握して利用できる．

図 3. アウトラインデータと触感シート

3.4 レイアウト例

ここでは，提案システムを用いて実装したコントローラのレイアウト例を示す (図 4)．左は，一般的な Xbox コントローラをイメージした事例であり，左側それぞれにスティックと 4 つのボタンを配置し，補助的なボタンも全て配置している．右は，格闘ゲーム向けコントローラをイメージした事例である．左側にスティックを一つ，右側に 6 つのボタンを配置し，押しやすさを考慮してボタンサイズを大きくしている．このように，ゲームジャンルや，プレイヤーの手の大きさ等に合わせた多様なコントローラを設計することができる．

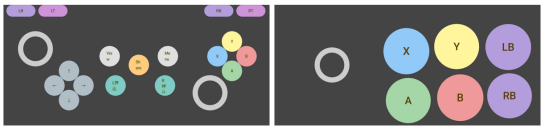

図 4. 一般的なコントローラの配置 (左) と格闘ゲーム向けの配置 (右)

4 議論

プロトタイプを用いて Minecraft をプレイしたところ，操作の遅延は体感されず，基本的なゲームプレイは可能であることが確認できた．

触感シートは，スティックの操作範囲を明確にし，ボタンを押したフィードバックが得られるという利点があった．しかし，ボタン領域全体が均一な質感であるため，指先がボタンの境界を認識しづらいという課題も明らかになった．この結果，ボタンが密集するレイアウトでは正確な位置を瞬時に把握できず，操作が難しくなっている．この課題を解決するため，今後はボタンのフチ部分のみ質感を変化させるなどの改良を行う．

現在レイアウトツールで出力したアウトラインデータを UV プリンタで出力するためには，Adobe Illustrator でテクスチャ(スウォッチ) の割り当て等の作業が必要であり，一定の知識が必要となる．今後は，こうした印刷過程の支援や自動化も検討していきたい．

これらの機能開発と改良を経て，最終的には一般的なゲームコントローラや，スマートフォンベースのコントローラと比較した評価実験を行い，提案手法の有効性を検証したい．

参考文献

[1] M. Fukumoto. PuyoSheet and PuyoDots: simple techniques for adding "button-push" feeling to touch panels. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, p. 3925–3930, New York, NY, USA, 2009. Association for Computing Machinery.

[2] K. Ikematsu and K. Kato. ShiftTouch: Extending Touchscreens with Passive Interfaces using Small Occluded Area for Discrete Touch Input. In *Proceedings of the Seventeenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '23, New York, NY, USA, 2023. Association for Computing Machinery.