仮想影と生体情報を用いた身体能力拡張コンテンツ

小林菜摘[†1] 桑名祐弥[†1] 奥沢紀祥[†1] 宮本陸[†1]
原山透湖[†1] 増田光咲[†1] 川合康央[†1]

概要：社会は，様々な身体能力を持った人々で構成されている．本研究では，自身の身体を影で表現し，その運動を拡張することによって，体格差や年齢，身体の障害に左右されず，身体の変幻自在な体験を可能となるシステムを開発した．本システムは，生体情報として，脈拍センサと筋電位のセンサから得られた値を Arduino によって取得し，Ruby on Rails で実装した API を使用して，データのやり取りを行った．また，クライアントの開発には Unity，人体の認識については Kinect を使用した．

1. はじめに

社会は，様々な身体能力を持った人々で構成されている．個々人の身体能力の差とともに, 男性, 女性といった性差, 児童や高齢者といった年齢差, 身体などの障害の有無など, 身体能力には様々な差が存在する．一方で，スポーツなどでは，個人の筋肉量などの運動能力ではなく，動体視力や反射神経，また試合に対する戦術，戦略など，様々な要素が複雑に絡み合ってそのルールが構成されている. そこで, 個人の身体能力を拡張することによって，従来のオリンピックやパラリンピックといった分離ではない，新しい身体を用いたコンテンツを提案する．

本研究では，仮想影と生体情報を用いた身体能力拡張コンテンツ開発を行った．影という誰もが共通して持っているものを用い, これを変幻自在に変化させることによって, 自身の身体と連続性を有しつつ，その拡張について可視化することで，様々な人々が楽しめるコンテンツになるのではないかと考えた．

また，今開発では，身体能力の拡張として，Arduino を用いた脈拍と筋電位をセンサで取得するシステムを開発し，その値をリアルタイムで取得して，生体情報をシステムに反映させる．筋電位の取得には，Arduino で使用可能な Mayoware を採用した.また，身体の認識には KinectV2 を使用してプレイヤを認識し，そのアウトラインを切り取ることで画面上に仮想の影を出力している．出力したプレイヤの仮想影と取得した生体情報を組み合わせることによって, プレイヤは自身の身体である影を自由に操作することができる．

先行研究として，岩崎[1]らが影による影とのインタラクション手法の研究を行い，スクリーンに写る全ての実物影を正確にシミュレーションし，それに基づいて仮想影を制御することで，実物影を仮想影との様々なインタラクションを実現を明らかにした．また林[2]らが，発達障碍児の身体表現活動に影メディアを適用し，その有用性について研究を行い，影メディアを用いた身体表現を通して，年齢, 性別, 障碍の有無などに関わらず, 体験者同士のつ ながりを支援できる可能性が示された．Mayoware については,馬場ら[3]が肩の筋疲労測定をする際に使用している．

2. 身体能力拡張コンテンツ開発プロジェクト

2.1 コンテンツの流れ

本システムは，影と身体能力の拡張をテーマに作られている．脈拍と筋電位については，システム開始時点からの計測となっている．ユーザのコンテンツ体験の流れは，スタート画面，メイン画面，エンド画面といった構成となっている．スタート画面では，コンテンツを体験する上でのルール等の説明を行う．メイン画面では，Kinect で認識した自身の影を使用して，敵キャラクタとの戦闘を行う．腕や肩に装着した筋電位センサによって, 腕を伸ばして戦う, 羽を生やして空を飛ぶなどの体験を行うことができる．また，脈拍に応じて敵の演出も変化する．エンド画面では, 体験中の結果が表示される. 表示される情報は, 体験時間, 体験中の脈拍と筋電位の最大，最低，平均値である(図 1).

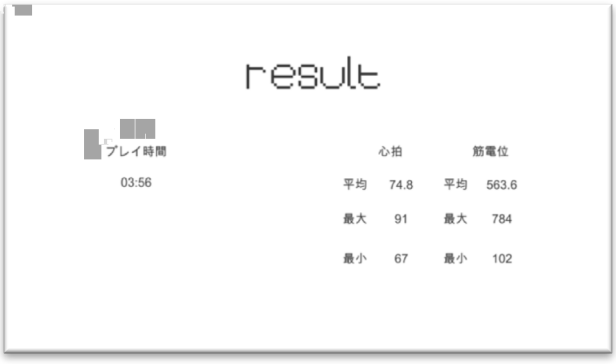

図 1 エンド画面の様子

2.2 システムの概要

本システムは，主としてゲーム開発環境であるゲームエンジン Unity を用いて開発を行った．センサとして Kinect を使用し，背景から切り抜かれたプレイヤを，黒でレンダ

†1 文教大学

リングすることによって，画面上に仮想の影を表現した．ユーザの生体情報取得には，Arduino を使用して，脈拍と筋電位の値を取得するセンサ類との接続を行った．Unity 上では，この Arduino で取得した値を使うため，ruby on rails で API サーバを構築した(図 2).

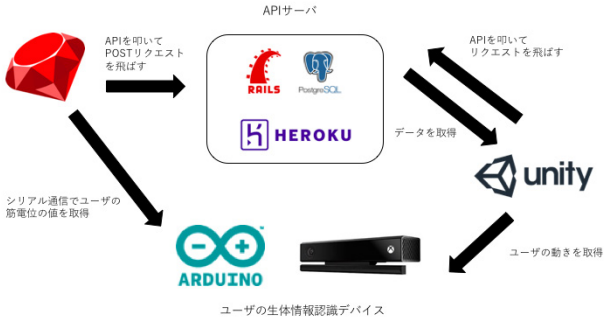

図 2 システム概要図

2.3 センサによる計測について

Arduino では，筋電位センサ及び脈拍センサで取得した身体情報の数値を計測する．筋電位センサは，筋肉の動きにかかわる皮膚表面の電位の数値を取得するものである．皮膚表面の電位は筋肉に力が入ると上昇するため，どの程度値が上昇したかを取得することで，肩に羽が生える，腕が伸びるなど，影の演出が変化するようになっている．また，脈拍センサは脈拍を数値にて取得するものであり，これらは 1 秒ごと値を取得している．センサを装着している様子を以下に示す(図 3).

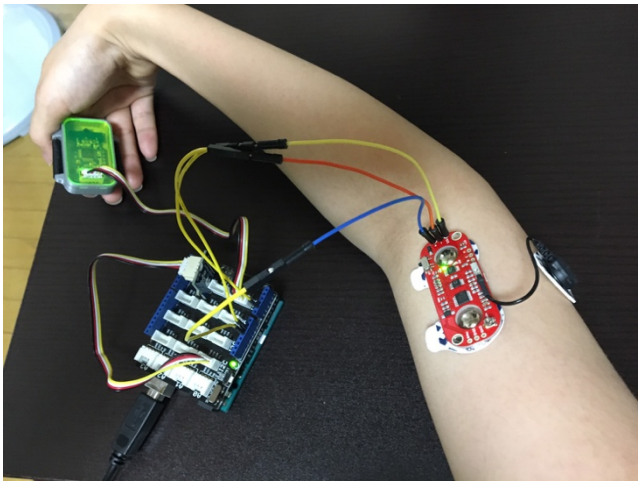

図 3 センサによって身体情報を取得している様子

2.4 Arduino の API

Heroku, Ruby on Rails, postgrSQL を使用して，API サーバを構築した．この API を実行することによって，Arduino で取得した値を DB に登録し，数値計算や加工を容易にすることとした．データ送受信する際のデータ形式は json で行っている．今回は，ローカル上で ruby によって記述されたソースコードを実行した．コードの内容として，シリアル通信で Arduino の値を取得して json を生成，API に POST リクエストを実行という流れで実装した．API を使用するメリットとして，http を経由しているため，Unity の実行と Arduino の値を取得する PC を分離することが可能な点があげられる．複数台の PC を使うことにより，処理の負荷分散をすることができる．また，プレイ後のプレイヤのデータも API を実行して，DB に登録をしているため，取得したデータの事後活用も可能である．今回実装した API は下記の通りである(表 1).

表 1 API の URI

URI	メソッド	役割
api/v1/one_arduino_values	GET	登録されている 1 台目の Arduino の全件表示
api/v1/one_arduino_values	POST	データ登録
api/v1/one_arduino_values	DELETE	データ全件削除
api/v1/one_arduino_values/new	GET	最新のデータを 1 件取得
api/v1/multi_arduino_values	GET	登録されている登録されている 2 台目の DB の全件表示
api/v1/multi_arduino_values	POST	データ登録
api/v1/multi_arduino_values	DELETE	データ全件削除
api/v1/multi_arduino_values/new	GET	最新のデータを 1 件取得
api/v1/play_summary	GET	登録されている DB の全件表示
api/v1/play_summary/:id	GET	指定されたユーザの id のデータを表示
api/v1/play_summary	POST	プレイデータ登録

図 4 システムの動作画面

本システムでは，脈拍，腕の筋電位，肩の筋電位を取得するため，arduino を 2 台使用した．これらは，筋電位のみを取得する one_arduino，筋電位と脈拍を取得する multi_arduino と命名した．

2.5 クライアントの実装

クライアントの実装には，ゲームエンジン Unity を使用した．また，人の認識部分には，Kinect を使用している．Kinect を Unity で使用する際に，Rumen の開発した Kinect v2 Examples with MS-SDK[4]といった Assets を使用した．本 Assets を利用することによって，背景を切り抜き，人体のアウトラインを抽出することが可能となった．取得した人体を黒でレンダリングを行い，これに物理判定を付すことによって，画面内で他のオブジェクトとの接触判定が可能となるよう実装を行った．また，敵キャラクタにはアニメーションを設け，プレイヤの影との間でモーションによるインタラクションを作成した．(図 4)

クライアント側では，毎秒 API の GET リクエストを実行している．このことによって，毎秒 DB に登録されている Arduino の最新の値を取得できるようにしている．毎秒取得した最新の値が一定値を超えると，腕が伸びる，肩に羽が生えるなどのインタラクションを体験できるように実装を行った．また，ゲームをスタート時に DB に登録されているデータを削除するため，DELETE リクエストを実行している．また，ゲーム終了時には，END のシーンで結果を表示する際，表示された値を DB に登録するため POST リクエストを実行している．

3. まとめ

本研究は，ユーザの身体情報に基づいて，その形状が様々に変化する影を用いた身体能力の拡張についてのシステム開発である．実装したシステムでは，生体情報を脈拍センサと筋電位のセンサから Arduino を介して取得し，Ruby on Rails で実装した API を使用して，データのやり取りを行った．また，クライアントの開発には Unity，人体の認識については Kinect を使用した．ステージにおいては，木々などを配置したが，今開発では，敵キャラクタを表すオブジェクトが木々のオブジェクトと影が重なってしまい，判別しづらいといったフィードバックが得られたため，UI デザインの改善も必要だと考えられる．

今回，身体拡張として実装した要素として，腕に力をいれると腕が伸びる，肩に力をいれると羽が生えるといったものであったが，影の特性を十分利用しているとは言い難い．今後，影を利用して，何かのオブジェクトと身体が同化する，オブジェクトを操作できるといったコンテンツへと拡張を行うことが考えられる．

システムの実装方法において，Arduino で取得した値は，ローカルでシリアル通信を取得し，API サーバに POST リクエストを行い DB に保存をかけている．Unity 側では GET リクエストでデータを取得しているが，API サーバを使わず websoket を使用して，Unity と連携するといった手法を採用したほうが，システム構成がより整理されるのではないかと考えられる．今回，Arduino でシリアル通信を採用したが，ruby でシリアル通信を取得する際に，値が途切れることがあった．これは，シリアル通信では 1 秒ごとに受信できる文字に限りがあったためであった．また，ローカル上で API を実行することはシステムが煩雑になるため，ESP32 などのシールドを使用して，Arduino 上で API や websoket を使用したいと考えた．

今後，DB にデータを登録した際に，脈拍と筋電位の相関を測りたいため，年齢と性別を取得したいと考えている．その際に，web カメラを使用して IBM が提供している visual recognition を採用する．これは，画像認識でプレイヤの年齢と性別を判別することができるため，ユーザが入力をする手間を省くことが可能となる．また，プレイ後のプレイヤのデータを DB に登録をしているため，web 上で可視化できるようなツールについても検討する．

Kinect で取得した影については，プレイヤの形をそのまま取得することが可能であるため，プレイヤが何かを身につけていた場合，その身につけた要素に対しても物理判定を取り入れられるように実装する方法についても検証するとともに，現時点では Kinect で 1 人しか認識ができないため，複数の人間を認識して，対戦や協力して進められるようなものについても実装を進めている．今後，本システムの開発と利用者からのフィードバックで得られた知見を用い，更なるシステムの拡張をはかっていく．

参考文献

[1] 岩崎妃呂子, 伊藤玲, 近藤桃子, 杉浦沙弥, 大葉有香, 水野慎士. 実物の影による仮想の影とのインタラクション手法の提案. エンタテインメントコンピューティングシンポジウム 2015 論文集, 2015, vol.2015, p40-45.
[2] 林真誘, 西洋子, 三輪敬之. 影メディアを用いた発達障碍児を対象とする身体表現ワークショップ, ヒューマンインターフェースシンポジウム 2017 論文集. 2017, vol.2017, p593-591.
[3] 馬場南実, 嵯峨智, 志築文太郎, 高橋伸. 肩の筋疲労測定を用いた姿勢改善支援システムの検討. 情報処理学会研究報告ヒューマンコンピュータインタラクション (HCI), 2017, p.1-8.
[4] ” Kinect v2 Examples with MS-SDK”. https://rfilkov.com/2014/08/01/kinect-v2-with-ms-sdk/, (参照 2018-12-24).