

自律キャラクタの遠隔多人数テストと調整のための共有 VR フレームワーク

小栗 賢章^{1,a)} 三武 裕玄^{1,b)} 杉森 健^{1,c)} 佐藤 裕仁^{1,d)} 長谷川 晶一^{1,e)}

概要：身体的なインタラクションを行う自律キャラクタの行動・動作の開発過程で求める表現に適うようになるまでには、テストプレイを行い、試行錯誤による微調整が必要である。しかし、複数人を相手とするテストプレイでは開発の現地に必要な数のテストプレイヤーを集めることは容易であると限らない。そこで本稿では、遠隔・複数人のテストプレイヤーが VR 機器を用いてテスト実行中のキャラクタプログラムに参加し、キャラクタの動作を即時確認・調整することを可能とする共有 VR フレームワークを提案・実現した。

1. はじめに

自律動作する CG キャラクタはゲーム等のインタラクティブ作品に多く用いられ、個性的かつ魅力的な振る舞いによって人を楽しませてきた。VR 空間でキャラクタと触れ合える作品 [1] や、人の振る舞いに反応するインタラクティブキャラクタサイネージ [2], [3], キャラクタ性を持つ人型ロボット [4], [5] 等の普及により、人とキャラクタが（現実・バーチャルを問わず）同一空間において身体的なインタラクションを行う機会が増えている。このようなキャラクタには、人の行動や周りの環境に合わせて自然な反応動作を行いつつ、その動作によって個性や作品を表現することが求められる。

インタラクティブな自律キャラクタの行動・動作の制作では、相手の位置に応じて変化するプロシージャルアニメーションや、行動決定のアルゴリズムを作成し調整する必要がある。このためにはテストプレイを行い、キャラクタの行動・動作が求める表現に適うようになるまで試行錯誤による動作・アルゴリズム・パラメータの微調整を行わなくてはならない。特に複数人を相手としたインタラクションが可能なキャラクタでは、テストプレイにも複数人が必要となるため、開発者一人では調整ができない。しかし、インタラクティブ作品の制作者や環境は多様であり、開発者の下に複数人のテストプレイヤーを集めることが容

易であるとは限らない。

そこで、遠隔地から複数人のテストプレイヤーとキャラクタが同一空間でキャラクタと身体的インタラクションをとるために、共有 VR 空間を用いることを考える。現在、共有 VR 空間を提供するサービスとして VRChat, cluster 等の VRSNS サービスが存在する。特に VRChat には UDON[6] と呼ばれるプログラム実行環境があり、利用者の制作したプログラムによってインタラクティブに変化する共有 VR 環境を実現できる。これにより自律キャラクタを実現すれば、多人数によるテストプレイを遠隔で行うことは可能である。しかし、動作・アルゴリズム・パラメータ等の調整結果をテストプレイ環境に反映するにはキャラクタを含む環境情報一式のコンパイル・パッケージ化とサーバへのアップロードを経るため時間がかかる。また、テストプレイ中にキャラクタ行動プログラム内の変数や状態といったデバッグ情報に触れることも容易でない。このため、試行錯誤的な微調整を行ったり、予期しない動作を修正したりするために用いることは難しい。

インタラクティブキャラクタの動作・振る舞いを試行錯誤的に微調整する行う上では、テストプレイ中に開発者が動作やパラメータを調整し・即時反映できること、キャラクタ行動プログラムのデバッグ情報に容易に触れられることが望ましい。

上記の要求を満たす手法として、本研究では、インタラクティブキャラクタの開発環境自体に容易に組み込んで利用できる共有 VR フレームワークを提案・実現する。具体的には開発環境として Unity を想定し、Unity エディタで実行中のインタラクティブキャラクタプログラムに対し、遠隔のテストプレイヤーが VR 機器で没入しインタラク

¹ 東京工業大学

a) oguri.k.aa@m.titech.ac.jp

b) mitake@hi.pi.titech.ac.jp

c) sugimori.k.ac@m.titech.ac.jp

d) satoh.h.aa@m.titech.ac.jp

e) hase@haselab.net

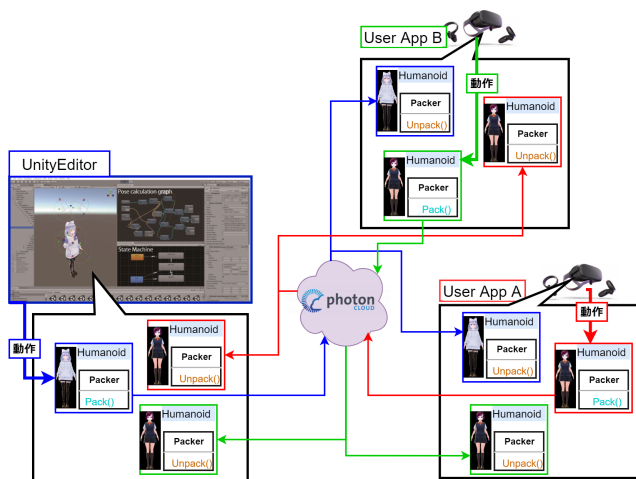


図 1 システム概略図

ションできるようにする。遠隔プレイヤーが体験しているキャラクタープログラム自体を、開発者がUnityの開発環境により即時確認・調整することができる。

また、こうしたツールは調整時のテストプレイだけでなく、インタラクティブキャラクタープログラムの小規模な遠隔体験提供にも有用と考えられる。試作物のデモや実験、趣味の作成物を公開し少数に楽しんでもらうといった用途も想定する。

2. 提案システム

提案するフレームワークは、Unityで開発中のキャラクタープログラムに組み込むライブラリと、テストプレイヤー用のVRアプリケーションからなる。ライブラリは、UnityとVRアプリケーションとの間で、人・キャラクターの動作を相互に同期するものである。また、VRアプリケーションは、ネットワークを通じて送信されてきたキャラクターや他のテストプレイヤーのアバターをVRデバイスを用いて表示し、コントローラを用いて自分のアバターを動作させることができるものである。

テストプレイヤーがVRアプリケーションを使用すると、自律キャラクターを開発するUnity上にテストプレイヤーによって操作される身体モデルオブジェクトが出現する。複数のテストプレイヤーが同時に一つのUnityシーンにアクセスすることができ、同時に複数名によるキャラクターとのインタラクションを行うことができる。開発者は、テストプレイにおけるインタラクションの様子とその際のキャラクター動作プログラムの実行状況を観察し、インタラクションの最中でも動作・行動のパラメータを変化させることで試行錯誤による振る舞いの微調整が可能である。

本システム全体の概略図を図1に示す。以降ではシステムの実現について述べる。

2.1 通信手段

リアルタイム相互同期通信にはExit Games社からリ

リースされているUnityアセットPUN2(Photon Unity Networking)を利用する。Photonはネットワークゲーム等でのシーンやキャラクターの同期に広く用いられているライブラリであり、ゲームシーン内の各オブジェクトの状態(位置・姿勢・値など)を相互に同期できる。

2.2 通信内容

Unityでは、人型の自律キャラクターやプレイヤーのアバターの身体モデルオブジェクトは、Humanoidと呼ばれるデータ構造で管理される。Humanoidは人体の各関節に相当する回転関節を持つ。

そこで、開発者の用いるUnity内のシーンおよび各テストプレイヤーが用いるVRアプリケーション内のシーンのそれぞれの間で、シーン内の対象とするHumanoidの全関節姿勢を相互に同期する。また、キャラクターがプレイヤーの行動に反応するために、プレイヤーがUnity内のHumanoidで表現されていることを前提にしてキャラクターの行動決定プログラムを書く。これにより、UnityおよびVRアプリケーション間でHumanoidの状態をネットワークで同期するだけで共有VR空間におけるプレイヤーとキャラクターとのインタラクションが可能となる。

ただし実際には、Humanoidに含まれない追加パーツを持つキャラクターや、Humanoidに合致しない形態のキャラクターも考えられる。また、関節姿勢以外にもキャラクターの状態を表す変数(表情など)を同期する必要が生じることも考えられる。

そこで提案フレームワークでは、キャラクター1体分の同期データを表すインタフェースクラスを用意し、そのインタフェースクラスを通じてキャラクターの状態の同期を行う。このキャラクター1体分の同期データを表す抽象クラスを本手法ではPackerと呼ぶ。

Packerは、同期に先立ってキャラクターの状態の同期に必要な情報を収集して内部の変数に格納(pack)する。同期後は、受信した値をシーン内の対応するキャラクターに反映(unpack)する。収集・反映する典型的な情報として、Humanoidの姿勢・表情と単一オブジェクトの位置姿勢の3種類を基本機能として提供する。

- HumanPosePacker ... Humanoidの位置・姿勢と関節姿勢パラメータ
- BlendSharpPacker ... Humanoidの表情アニメーションパラメータ
- TransformPacker ... 任意のゲームオブジェクトの位置・姿勢

これらのPackerが実装されたコンポーネントを同期したいオブジェクトにセットして用いる。

PUN2の役目はといえば、Packerを同期することである。このように抽象化することで、PUN2以外のネットワークライブラリを用いるよう書き換えることも比較的行きやす

い。PUN2は通信量が多い場合は有料となるため、オープンソースのgRPCフレームワークであるMagicOnionや、プライベートネットワーク内UDP通信とVPNを組み合わせたものを用途に応じて利用可能にすることは将来の課題である。

2.3 キャラクタモデルの共有

Unityシーンおよび接続するすべてのVRアプリケーションの間で、利用するキャラクターモデルを相互に共有する必要がある。キャラクターモデルをあらかじめVRアプリケーションに組み込むことが最も簡便であるが、開発中にキャラクターモデルを変更する可能性を考えると好ましくない。また、テストプレイヤーのアバターは各テストプレイヤーが選択できることが望ましい。

本手法では、キャラクターモデルの共有の方法として以下の4種類の方法を用意した。

- (1) あらかじめUnityプロジェクト及びVRアプリケーションのUnityゲームシーン中に利用したいキャラクターモデルゲームオブジェクトを配置しておく方法
- (2) あらかじめUnityプロジェクト及びVRアプリケーション内に全キャラクターモデルを格納したプレハブを共有しておき、名称指定によりロードすべきキャラクターモデルを指定する方法
- (3) あらかじめ全キャラクターモデルのファイル(VRM形式)を共有しておき、ファイル名によりロードすべきキャラクターモデルを指定する方法
- (4) キャラクターモデルをオンラインで共有するサービスであるpixiv社のVRoidHubを用い、各々が利用したいキャラクターモデルをあらかじめVRoidHubに登録しておき、UnityプロジェクトおよびVRアプリケーションからはVRoidSDKを用いてVRoidHubからモデルデータをダウンロードして用いる方法

2.4 開発者側の利用法

自律キャラクター開発者は、まず使用したい通信方式クラス(標準ではPUN2)のコンポーネントをゲームシーンに配置する。次にPackerを同期したいすべてのオブジェクトにセットする。その際に、受信側におけるHumanoidモデルのロード方式の指定も行う。これにより、テストプレイヤーのVRアプリケーションに自律キャラクターを出現させ動作を同期することができる。

2.5 テストプレイヤー用VRアプリケーション

テストプレイや体験のためのVRユーザアプリケーションはコントローラ付きHMDを用いることを前提としており、基本的なものとして以下の2種類を用意している。

- (1) Humanoidキャラクターモデルをアバターとするもの
- (2) 頭部と両手のみを表示したもの

(1)は自律キャラクターと同様の形式で同期され、自律キャラクターがその動作生成プログラムによってHumanoidモデルを動かすのに対して、こちらはユーザのHMDの動きとコントローラの入力から全身IKを解くことによってHumanoidモデルを動かす。全身IKにはRootMotion社からリリースされているUnityアセットFinalIKを用いている。(2)はプレイヤーのアバターに全身が不必要な場合に用いる。データの同期としては頭部と両手の位置や必要な指の関節にTransformPackerがセットされており、HMDとコントローラの動きをダイレクトに伝えることができる。自律キャラクターの開発者はテストプレイヤーにVRアプリケーションを提供する際に、これらの基本的な実装に加えて開発したキャラクターのインタラクションに合うようにコントローラのボタン入力を加えるなどの改変を加えることができる。

さらに、ユーザアプリケーションはHMDを用いたVRアプリだけでなく、2.1で示した通信フレームワークを実装することで開発者が任意のプラットフォームに向けて作成することができる。例えば、スマートフォンを用いたARアプリケーションであったり、デジタルサイネージ向けにセンサを用いたディスプレイに組み込むことができる。この場合、自律キャラクター側に同期したいインタラクション対象の任意オブジェクトにTransformPackerをセットする実装や、任意の手段でHumanoidを動作させるような実装を行う。

3. 動作例

3.1 多人数インタラクション

本システムをUnityEditor内の自律キャラクター作成環境に組み込み、実際に動作させた様子を図2に示す。この動作例では自律キャラクター作成環境には我々の研究室で開発中のVGentEditor[7]を用いている。ここでは2人のテストプレイヤーにVRアプリケーションを使用してもらい、VGentEditor上で動作の調整を行った。その様子を図3に示しており、Unityゲームシーンにテストプレイヤーによって動かされるHumanoidキャラクターモデルが出現していることがわかる。

3.2 触れ合えるVRインタラクティブキャラクター

ここには我々が[8]で発表したインタラクションを遠隔地で体験できるように、本システムを組み込んだ例を示す。図4には、[8]の実装に本システムを組み込んだ場合のシステム全体の図を示す。図中左側の自律キャラクターとプレイヤーの物理シミュレーションを行う過程の詳しい説明は[8]を参照されたい。図中右側では本システムを使用しており、VRアプリケーション中で全身IKによって計算された結果をリアルタイムに自律キャラクター側へと同期している。プレイヤー自身のアバターは全身IKによって計算



図 2 多人数インタラクションの様子

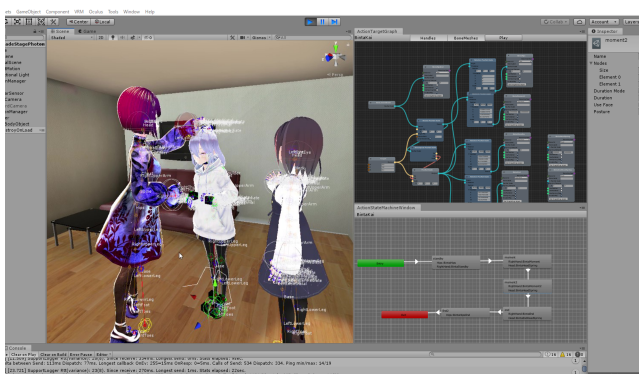


図 3 VGenEditor 上での調整

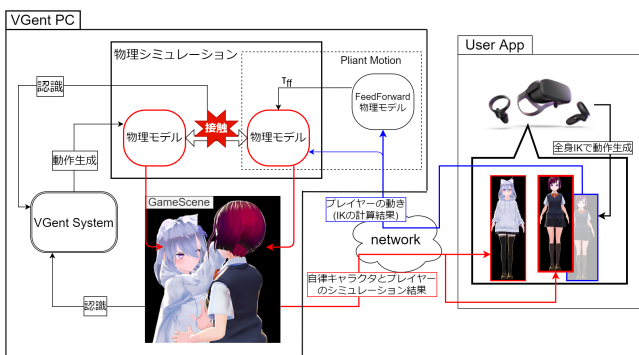


図 4 [8] に本システムを使用した場合のシステム図

されたものは描画せず、通信によって送られてくる物理シミュレーションの結果を描画することで、VR アプリケーション中に物理シミュレーションを組み込まなくても疑似的な体験ができるようにした。プレイヤー自身が自分の体の動きを描画から確認できるまでに往復分の通信遅延が発生し、ローカルで動作する場合に比べて体の動きの追従が遅れることが確認されたが、我々の環境では大きな違和感とはならなかった。しかし、低速な通信環境では違和感が大きくなってしまふことが十分に考えられるため、体験者に高速な通信環境が備わっている必要があると考えられる。

4. 結論と今後

本稿では開発環境として Unity を想定した、インタラクティブキャラクターの開発環境自体に容易に組み込んで利用できる共有 VR フレームワークを提案・実現し、提案フレームワークを活用したインタラクティブキャラクタープログラムの遠隔多人数テストと小規模な遠隔体験システムの開発を確認できた。

今後は 2.2 項にも述べたように PUN2 以外の通信フレームワークを使用した実装を行うとともに、開発中の VGenEditor とともに本格的な公開に向けて準備を進める予定である。また、本システム用いて VGenEditor によって開発されたインタラクティブキャラクターとの小規模体験会なども行い、フィードバックを得ることでより機能の洗練を図りたい。

参考文献

- [1] バンダイナムコエンターテインメント: サマーレッスン, , 入手先 (<https://summer-lesson.bn-ent.net/>) (参照 2020-08-05).
- [2] 名古屋工業大学: メイ&タクミ, 名古屋工業大学 (オンライン), 入手先 (<http://mei.web.nitech.ac.jp/>) (参照 2020-08-05).
- [3] 株式会社イージュワークス: inforoid, 株式会社イージュワークス (online), available from (<https://www.ejworks.com/inforoid/index.html>) (accessed 2020-08-05).
- [4] スピーシーズ株式会社: 世界初! AI と“仕草”によるエンパシーコミュニケーションを実装した等身大アイドルロボットのデモンストレーションを実施, , 入手先 (<https://www.atpress.ne.jp/news/197285>) (参照 2020-08-05).
- [5] ソフトバンク株式会社: pepper, , available from (<https://www.softbank.jp/robot/pepper/>) (accessed 2020-08-05).
- [6] VRChat: Getting Started with Udon, VRChat Inc. (online), available from (<https://ask.vrchat.com/t/getting-started-with-udon/80>) (accessed 2020-08-05).
- [7] 佐藤裕仁, 三武裕玄, 杉森健, 長谷川晶一: VGenEditor: 操作部位と空間目標点を動作表現として用いたインタラクティブキャラクターの動作生成, VRSJ2019(第 24 回日本バーチャルリアリティ学会大会) 論文集 (2019).
- [8] 小栗賢章, 三武裕玄, 杉森健, 佐藤裕仁, 長谷川晶一: インタラクティブキャラクターのフレームワークと触れ合える VR インタラクションシステム, 情報処理学会インタラクション 2020 予稿集 (2020).