

空中ディスプレイをジェスチャーコントロールする手法の模索と提案

—— 空中タッチディスプレイを用いたゲームアートの制作 ——

室橋直人

ゲーム学科

Search and proposal of a method for gesture control of an aerial display
Production of game art using aerial display

MUROHASHI Naoto

Department of Game

(Received October 31, 2019 ; Accepted December 11, 2019)

キーワード：空中結像、空中ディスプレイ、ジェスチャーコントロール、Unreal Engine、ホログラム

Abstract

Just 20 years ago, the aerial display, which was an expression of the future in movies and animation games, can now be expressed using various materials and methods. In addition, gesture-based operation without a controller was a high-level research subject at graduate schools, but it can be obtained at a relatively low cost as a commercial off-the-shelf product represented by Microsoft's Kinect and Leap Motion. Became. In this paper, we will explore and propose a method for operating the aerial display in the air mainly for the purpose of applying it to the production of game art works by using Unreal Engine as the input device and using the aerial display for the output.

1. はじめに

空中に映像が流れる場面は20世紀以前近未来を表す表現として映画などでよく用いられていた。「バックトゥーザ・フューチャー2」では未来にタイムスリップした冒頭、未来で上映されている映画「ジョーズ19」のホログラムのサメが主人公に襲い掛かるシーンがあり、これが未来にきたことを表現している。同様にコントローラーやマウスといったデバイスを使わずにジェスチャーで操作することも空中に映像が出るのと同じように未来を表す表現として用いられている。「スターウォーズ」ではホログラムで描かれたドロイドやクリーチャーが駒として登場しDejarik¹⁾と呼ばれるチェスを行うシーンが描かれている。「マイノリティーリポート」では空中に表示されたPCの画面に対してジェスチャーにより画像ファイルの検索やフォルダの移動、画像編集を行っていた。ほんの数10年前には、未来を表す表現であった空中ディスプレイは今や様々な素材や手法によって表現が可能となっている。また、コントローラーを用いないジェスチャーによる操作も、今ではMicrosoftのKinectやLeap Motionに代表されるような民生用の既製

品として比較的安価に入手ができるようになった。本稿では、主にUnreal Engineを用いてジェスチャーシステムを入力デバイスとし、出力に空中ディスプレイを使うことでゲームアート作品の制作に応用をすることを目的とした空中ディスプレイをジェスチャーコントロールする手法を模索・提案する。

2. 空中ディスプレイの考察

通常ディスプレイ等の光源から発せられた光は、筐体内で複数回反射した後、再度交差する。その際に交差した交点に映像が浮かび上がる。空中ディスプレイはこの交点を空中にすることで映像を浮かび上がらせている。

株式会社コトのAirWitchは映像表示にスクリーンが不要なため、映像が表示される周辺には人やフィギュアなどの実体物を設置することができ、実体と映像がシームレスに重なりあう演出が可能である。iPhoneやiPadに対応したプレートを販売しているため、比較的手軽に導入することが出来る。ただし、通常のディスプレイ等の光源と比べて輝度とコントラストが落ちてしまい、屋外等の外乱光が強い環境には不向きである²⁾。

日本カーバイド工業株式会社のAerial Display

Reflector³⁾も同様の技術を用いて空中ディスプレイを実現している。どちらの技術もレンズ面以外に液晶ディスプレイの光を複数回反射させるためのリフレクターが必要となるが、液晶と線対称位置に光を集約させることで、空中に映像を表示させている。

アスカネットのASKA3Dプレートはガラスや樹脂などで出来た特殊なパネルを通過させることで、実像の反対側の等距離の空中に実像を結像させることに成功した特別なプレートである。従来のホログラムやインタラクティブフォトグラフィーなどとは全く異なる新しい空中結像技術であり2Dや3Dの概念もピントの必要性も一切なく、先の2つの技術と異なりリフレクターを必要としない。ASKA3Dプレートの仕組みは「基本的な原理としては2つの直交する反射板（ミラー）を経由して、ASKA3Dプレートを対称軸に光源（液晶ディスプレイなど）と1:1の空間上に実像を結ぶという技術である。ペッパーズ・ゴースト（英語：Pepper's ghost）は、劇場などで使用される視覚トリックであり、板ガラスと特殊な照明技術により、実像と板ガラスに写った物体を重ねて見せることで、効果を発揮する。実像と「物体」はぶつかることなく交差し、照明の調整により「物体」を登場させたり消したりすることができる。ASKA3Dプレートはペッパーズ・ゴーストとは反対側に像を結ぶので、何もない空中に映像を出現させることが可能である。プレートが1枚あれば、誰でも何時でも何処でも明るい場所でも簡単に表示可能である⁴⁾。

ASKA3Dプレート、AirWitch、Aerial Display Reflectorの3点とも特殊な透過反射板、いわゆるプレートのみの販売であるため、元の映像を投影する液晶ディスプレイやジェスチャー操作を認識するセンサーデバイスは市販品を用いることになる。今回はASKA3Dプレートを使用した。その理由はリフレクターを使わずにプレートに映像を通せば何でも、例えば紙媒体でも空中投影が可能以上に周りの明るさの影響を受けにくく、プレートのサイズのバリエー

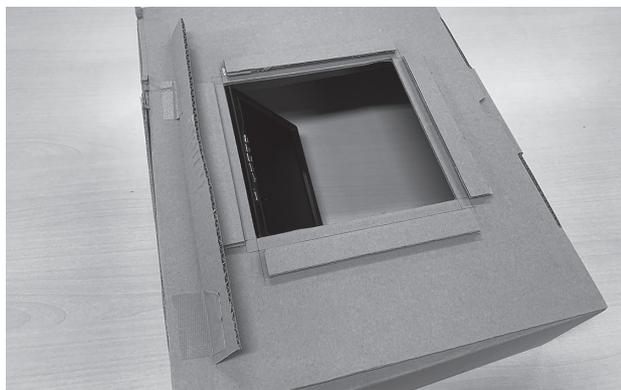


図1. ディスプレイとプレートの位置関係

ションも豊富だからだ。ASKA3Dプレートは投影するディスプレイとASKA3Dプレートを挟んだ反対側に同じ角度で空中結像を行うため、図1のような配置でディスプレイとASKA3Dプレートを配置することとなる。使用するASKA3Dプレートだが、ガラス製の非量産型タイプと樹脂製の量産型タイプにわかれており、今回は量産型を使用した。非量産型サイズは4種類あり、8インチから最大32インチまで表示することが可能で、照度も使用するディスプレイに対して約50%と明るく、ミラーピッチも細かいため20cm程度まで浮かび上がらせて結像することができる。それに対して量産型は約7インチのサイズが1種類で、照度もディスプレイに対して約15%と暗くミラーピッチも量産型より荒いため1cm程度までしか浮かび上がらせて結像させることができない。ただ、非量産型と量産型との価格差は値段の桁が一桁かわってしまうほどなので、比較的安価な量産型であるASKA3D-200NTを採用した。

3. システムの中核となるゲームエンジンについて

注目されているゲームエンジンにEpic Gamesの「Unreal Engine」とUnity Technologiesの「Unity」がある。UnityはC言語または、C++をベースとしながら、基本的にはスクリプト記述によってシステムの構築を行う⁸⁾。Unreal EngineもC++がベースとなっているが、スクリプトの記述をしなくても制作できるようにブループリントが採用されている。ブループリントとはノードをベースとするシステムで、スクリプトがGUI（グラフィカルユーザインタフェース）である為、視覚的にわかりやすく、そのノードを繋げていくことでシステムを構築することができる。また、デバックの際にも処理を行っているノードがグラフィカルに変化するため、システムのどの部分に不具合があるかを容易に見つけ出すことが可能となっている。UnrealとUnity両方のSDK（開発環境）を検討したが、どちらも本質的には同じであるが、Unityは新しい技術やデバイスに比較的早く対応するのにに対して、Unreal Engineの方は対応が遅く、対応しても正式な対応ではなく有志によるプラグインがGitHub¹⁰⁾に公開される。Unreal Engineは比較的早くアップデートが行われるため、公開されたプラグインが古いバージョンのまま更新されないことも多く、自力でプラグインを自分の使っているバージョンに合わせてビルドする必要があり、プログラマー以外は敷居が高い。筆者がデザイナーでありプログラム経験が少ない点からも視覚的にも判断しやすく、よりデザイナー視点で制作が可能なUnreal Engineを採用した。

4. 入力デバイスの考察

Leap Motion（リープモーション）は、2012年にLeap Motion社から販売されたコンピュータの手前に置かれるように設計された小型USB周辺装置で、手のジェスチャーによってコンピュータの操作ができる入力機器である。マウスや画面タッチを用いずにジェスチャーによって直観的に操作することができるシステムである。2基の赤外線カメラと赤外線照射LEDから構成されており、赤外線LEDに照らされた手や指を2基の赤外線カメラで撮影し、画像解析によって3D空間での手や指の位置を割り出す。検知できる範囲は半径50センチ程度、中心角110度の空間で、手、指、ペンのようなポイントを指し示すツールを0.01ミリの精度で認識する。両手と指10本をキャプチャすることが可能で、解析は専用のソフトウェアによって行なわれる。手や指の動きは3次元的に捉えることが可能で、上下左右、前後への移動はもちろん、ひねりなどの動作もしっかり捉えられる。Leap Motionコントローラーは単なる距離画像計測装置というにすぎず、姿勢推定アルゴリズムは明らかにされていない。価格は11,000円程度で簡単に入手することができる⁵⁾。

MicrosoftのKinectは「カラダまるごとコントローラー」のキャッチコピーの通り、モーションキャプチャや人物認識などを行う「NUI」（ナチュラルユーザーインターフェイス）の一つである。元々はゲーム機用の周辺機器として登場したが、後にPCに接続して使える一般用も発売され、ゲームだけでなく医療用やロボットなどの研究用にも活用されている。RGBカメラ、深度センサー、マルチアレイマイクロフォン、および専用ソフトウェアを動作させるプロセッサを内蔵したセンサーがあり、プレイヤーの位置、動き、声、顔を認識することができる。これにより、プレイヤーは自分自身の体を使って直観的な操作が可能となり、常にプレイヤーの位置、身長をKinectが測定し、最適なプレイができるよう上下の角度の自動調整が行われる。Kinectは、特殊なマーカー付きスーツの着用や、マーカー検出時に使用するトラッカーなどを必要としない。カメラに被写体を映すことでプレイヤーからKinectまでの距離を計測し、プレイヤーの骨格のさまざまな動きを検出して、ゲーム内のキャラクターの動きにリアルタイムに反映させることが可能となる。多人数による同時マルチプレイにも対応しており、プレイ人数はソフトによって異なる⁶⁾。このようにLeap Motionに比べると全身の読み取りや、多人数にも対応しているため、大掛かりなジェスチャーコントロールに向いている。機材自体もLeap Motionにく

らべると大掛かりな装置となっている。価格は15,000円程度だが、2017年に生産が終了しており入手が困難な上プラグインやSDKの開発も終了している。

NeonodeのzForce AIRタッチセンサーは、さまざまなアプリケーションで統合して使用できるIRレーザーベースのマルチタッチセンサであり、高い走査周波数と較正を必要としない安定性、短い待ち時間と、あらゆるディスプレイまたは任意の表面で使用することができる。zForce AIRは、乱反射した赤外光を検出してオブジェクトを検出し追跡し、放出されたIRビームと受信されたIR光の視野を同じ開口内に組み合わせるように配置された光学系のセンサーで構成されており、乱反射した赤外光を検出して指やオブジェクトを検出し追跡することが可能である。zForce AIRタッチセンサーは、標準コネクタを介してホストシステムに接続し、標準のI2CまたはUSB HIDプロファイルインターフェースを介して通信ができる。NeonodeのzForce AIRタッチセンサーの特徴は赤外光の送受信によるあらゆるディスプレイや物体の表面で使用することができる点であるが、赤外光の送受信であれば空中での使用が可能である。最大7点までのオブジェクトの検出・追跡が可能であるが、その数の設定は任意で行えるため、マルチタッチを行いLeap Motionのようなジェスチャーとして使用することもできれば、シングルタッチによるマウス操作の代用として使うことも可能である⁷⁾。加えて、長細いセンサーとコネクタで構成されているため、様々な機器に組み込みやすい構造となっている。半面、コネクタがむき出しなのでLeap Motionほどの耐久性はないと思われる。価格はセンサーの長さによって異なるが6,000円～8,000円程度であるが海外からの発送なので入手に多少の時間がかかる。

表1. 入力デバイスの長所と短所

Leap Motion

長所	手のジェスチャーによる直観的な操作
短所	姿勢推定アルゴリズムは不明

Kinect

長所	全身を読み取り、さらに多人数にも対応
短所	生産が終了しており入手が困難

zForce AIR

長所	エンジン側にプラグインが必要ない
短所	コネクタむき出しのため、耐久性を考慮

5. 入力デバイスの予備実験

3つの入力デバイスの比較から、空中ディスプレイであるASKA3D-200NTにあう入力デバイスの実験を行った。実験の方法だが、Unreal Engine4を中継して入力デバイスからASKA3D-200NTにジェスチャーを表示する方法を取った。その際、導入方法やデバイスの安定運用、ジェスチャーの調整の点を評価した。コンテンツの内容にもよるが今回約7インチサイズの空中ディスプレイに大きなジェスチャーは不向きであること、Kinectのプラグインの開発も終了しており自力でビルドしなければならない点からKinectのテストは行わなかった。

Leap Motionは、Unreal Engineとの相性は良くOrionの最新版とLeap Motion SDK Leap_Motion_Developer_KiとLeapUnreal 最新版をダウンロードして、プロジェクトのルートフォルダにPluginsというフォルダをそのまま入れる。Unreal Engineを起動しコンテンツブラウザ内のLeap Desktop Actorをシーンに配置すればLeap Motionでキャプチャされた手を表示することができる。指はもちろんのこと両手の甲やひらまでもが簡単にキャプチャされUnreal Engine内で操作することができ、各指や手のひらに当たり判定であるコリジョンを設定することで、空中に浮かんだ物体やメニューを操作することが可能となる。

zForce AIRは、USBでPCに繋ぐだけでドライバが自動でインストールされ、操作が可能となる。マウスの代わりに使うことが可能だが、キャプチャの範囲やマルチタッチの設定などは別途「Neonode Workbench」をインストールして設定を行う必要があるが、特別なプラグインが必要ではなく、Windowsの標準機能の一部が置き換わるので機材やシステムを安定して運用することができる。

6. 入力デバイスの実験

Leap Motionは姿勢推定アルゴリズムが公開されていない為、手のひらがひっくり返ったりした場合の調整が出来なかった。おそらくはレンズ面の汚れや向きなどに影響していると考えられるが、調整するパラメータが存在しないため、調整することができなかった。これは単にブループリントによる調整ができただけで、C++によるプログラム処理では調整は可能だと思われる¹¹⁾。また、表示される手のひらのオブジェクト自体リアルスケールとの関連が曖昧なため、手のひらのオブジェクトをスケールしてメニューに合わせると、他の指や手の甲やひらが干渉するなど、コリジョンの調整が繰り返し必要となった。

次にzForce AIRだが、薄い板状のデバイスな為空中に浮かんだ角度に合わせて容易に配置することができる。Leap Motionのように手や指全体の動きをキャプチャするわけではないので、複雑なジェスチャーには対応できないが、マルチタッチの機能があるため複数タッチによるスマートフォンでの操作のような、パンやピンチを行うことができた。

7. 入力デバイスの実験結果

2つのデバイスの導入実験の結果、Leap Motionは姿勢推定アルゴリズムが不明なためコリジョンの調整のみでコントロールする必要がある。そのためおよそ7インチの画面と操作範囲ではうまく機能することができなかった。より大型なASKA3Dプレートを導入することで上記の問題は解決すると思われるが、複雑なジェスチャーコントロールを小型のプレートで行うには調整がかなり難航する。

zForce AIRは、シングルタッチにすることで単にマウス操作によるメニューの選択やドラッグなどの代用とすることができ、単純なタッチ操作には向いている。仮に、平面上に動きを認識するため多少の角度の誤差があったとしても、指の先での操作が指の腹、もしくは第一関節での操作となる程度で操作自体に大きな問題が出るのが少ない。上記の「Neonode Workbench」を使うとzForce AIRのセンサーの幅に対してどこまで感知・追跡を行うのか、追跡するオブジェクトの数はいくつにするのか等の細かな調整が可能である。

この結果、より細かな調整が可能で安定運用のできるzForce AIRを採用することとなった。



図2. zFORCE AIR

8. 空中タッチディスプレイの制作

ASKA3D-200NTの表示できる領域は約7インチであるが、上記の「Neonode Workbench」を使いディスプレイの側で表示する領域を抑えることで、2560×1660の解像度を使うことができた。実際に使ったディスプレイは2560×1660の8.9インチものを使用した。仮にディスプレイを4kディスプレイに変更すれば4kの画像が空中に浮かび上がることになる。

「Neonode Workbench」の設定はDevice Configurationで行うのだがSub Area Hight Bound Xをディスプレイの解像度2560に対して2124、Sub Area Hight BoundYをディスプレイの解像度1660に対して1100としてタッチできる範囲を狭めることとした。同じくHid Display Size Xを2088、Hid Display Size Yを1100とすることで、本来ASKA3D-200NTを使用しているため表示可能範囲は約7インチであるが、使用しているディスプレイは8.9インチと大きいため、表示領域を狭めることで表示可能としている。MiscellaneousのNumber Of Reported Touchesを1とすることで、シングルタッチとすることができ、指の動きとUnreal Engine内でのカーソルの動きを合わせることができた。また表示に使用した液晶



図3. 空中タッチディスプレイ専用テーブル

ディスプレイも本来は上下逆さまに設置しないとASKA3Dプレートを通して正常に見えないのだが、ディスプレイそのものを逆さまに設置するには不安定となるため特別な固定具が必要となる。そこで、Windows10のディスプレイ設定の向きを横反対向きにすることで、表示自体を逆さまにすることで対応できた。このASKA3DプレートとzForce AIRの組み合わせは、大きめのセンサーを用意して「Neonode Workbench」で反応する適切なサイズに変えることでどんなディスプレイの映像も空中結像することができる利点がある。制作したこのシステムによって約7インチの映像を空中に投影し同範囲で空中タッチすることで反応する「空中タッチディスプレイ」を作ることができた。この試作機を元に、天板がくりぬかれてテーブルにて俯瞰による空中ディスプレイが鑑賞でき、使用するディスプレイの交換ができること、zForce AIRの交換・メンテナンスがしやすくケーブル類がなるべく露出しない構造となること、zForce AIRの角度の調整ができることを考慮した空中タッチディスプレイ用の機器の製作を行った。

9. 空中タッチディスプレイの考察

今回ASKA3DプレートとzForce AIR、Unreal Engineの組み合わせによる空中ディスプレイシステムの構築だが、次のような運用・利点が考えられる。画面に直接触れることなく操作が可能なので、感染症の防止などの衛生面での効果やタッチパネルに指紋が残ることがないので、パスワード認証時のセキュリティ面でも効果的である。加えて、空中結像はその視野角が狭く正面から角度がつくにしがって見えなくなる性質があるため、逆に第三者に見られないといった面でもセキュリティ対策に効果的と考えられる。また、Unreal Engine等のゲームエンジンを用いることでインタラクティブなコンテンツを空中に投影して直感的に操作することが可能となる。kinectの生産は終了してしまったが、後続機であるAzura Kinect¹²⁾やIntel RealSense¹³⁾などのデバイスを用いることで全身によるジェスチャー操作、Leap Motionを使った両手でのジェスチャーによるコントロール、AR画像の実態化によるMRのインタラクティブアート、これらの要素にゲームシステムを組み込んだゲームアートの制作が可能と考える。これらの考察を元に実際にコンテンツの制作を行った。

10. 空中タッチディスプレイの考察

10.1 コンセプトの制作

2019年度の芸術学部フェスタのテーマ「結ぶアート、築くアート」に合わせて「結ぶ」と「築く」をテーマに

空中ディスプレイのコンテンツ制作を試みた。簡単な操作方法で結ぶを表現しようと考え、ひと筆書きを採用した。ひと筆書き（ひとふでがき）とは、広い意味では「筆記具を平面から一度も離さず線図形を描く」ことであるが、狭い意味では、これに加えて「同じ線を二度なぞらない（点で交差するのはかまわない）」という条件が加わる。後者の意味を取って複数存在するブロックと開始地点を表すブロック、終了地点を表すブロックと3種類のブロックを用意した。開始地点のブロックから、終了地点のブロックまで同じブロックをなぞることなく全てのブロックを一筆書きでなぞっていくことでステージクリアとして、次のステージに進みだんだん難しく、ブロックの数を増やしていくゲームシステムが完成した。一筆書きで選択していった順にブロックが積みあがって階段となり、下に控えている「かぐや姫」がその階段を上がって月からのお迎えが来る前に月に帰っていくという設定でコンテンツの制作を開始した。



図4. 「ひと筆」のロゴマーク

10.2 Unreal Engineでの制作

Unreal Engineにはゲームの代表的なシステムである一人称視点のシューティングや三人称視点、横スクロールといったシステムをデフォルトのテンプレートとして持っているので、この一人称視点シューティングゲームのテンプレートを利用することでシステムのベースを構築した。Unreal Engine 4で壁や衝突時の情報を得るために「Line Trace」というものがある。主に「レイキャスト」呼ばれているが、視点の先や銃口の先にレイを飛

ばし、衝突判定を行い、衝突したアクターやオブジェクトの詳細情報を知ることができる。この「LineTrace」系のノードを使用することで、ブロックをなぞることが可能となる。なぞる操作自体はマウスで行うが、2Dのマウスの座標を「ConvertMouseLocationToWorldSpace」のノードを使うことで3DのWorld座標に変換することができる。先のライントレースと併用することで、3D空間内にあるブロックを2Dマウスで選択することができる。このマウス操作をzForce AIRを使って「指でなぞる」に置き換えることで、空中でひと筆書きすることができる。ブロックの方になぞられた際のトリガーを用意しておき、なぞられたブロックはその瞬間に上下左右に隣接しているブロックになぞられたことを通知する。これは衝突判定となるコリジョンを作成しておいて、なぞる判定が行われた際にコリジョンに触れているブロック、すなわち隣接しているブロックに「Apply Damage」ノードを使ってダメージを与える。隣接しているブロックはダメージを受けた直後になぞられた判定を受け取れば、自身が選択されたと判断するブループリントを作成することで、ただなぞるだけで選ばれることなくひと筆書きで選択することを可能とした。また、大きなブロックとブロックの間に小さな細長いブロックを挟むことで、ブロック間を埋めて、どのようになぞってきたかを視覚的にわかりやすくした。このシステムも同様にコリジョンとダメージの2つで自分が選ばれたかどうかの判定を行うが、小さなブロックは2つの大きなブロックに挟まれているので、2回連続でダメージを受けたかどうかのフラグを入れることで、誤動作を防ぐことが可能となった。

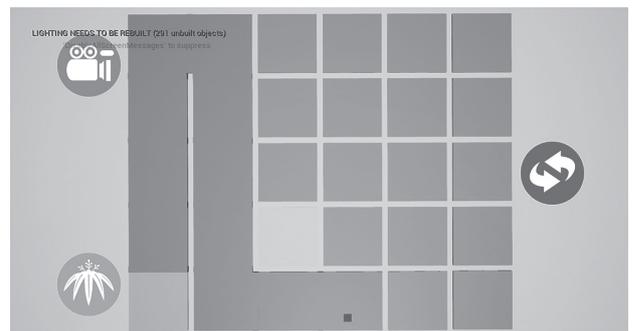


図5. 小さなブロックを配置

Unreal Engineにはレベルの考え方があるが、これはゲームにおける面・ステージと同じ考え方であり、このレベル自体にブループリントを記述できる。レベルブループリントと呼ばれるが、このレベルブループリントにひと筆書きに必要なブロックの数を記入していくこと

で、さまざまな組み合わせのひと筆書きのステージを作ることが可能となり、複数のステージをゲーム開始画面時にランダムに呼び出すことで、複数回遊んだ際に違うひと筆書きを楽しめるように構築した。「築く」のテーマに合わせた階段作りだが、選択したブロックを配列に入れて選んだブロックの順番を記録する。全てのブロックをひと筆書きして終了ブロックに到着すると、配列に記録された順番通りにブロックが下へと落下していき、階段を形成するようにした。そこにAI管理されている「かぐや姫」のスケルトルメッシュが一段ずつ階段を上っていく。この階段を上るスピードは変数にて管理しており、ステージが上がるにしたがいより早く歩かせた。かぐや姫が待機する場所は「和風」の感じとして「枯山水」を模した風景として、その外側には竹と秋の紅葉した木々を配置した。階段をあがっていくと向こう側に新宿副都心のビル群と東京工芸大学1号館が見えてくることで、2019年芸術学部フェスタの展示予定場所であるメディアラウンジから月に帰っていくような演出とした。

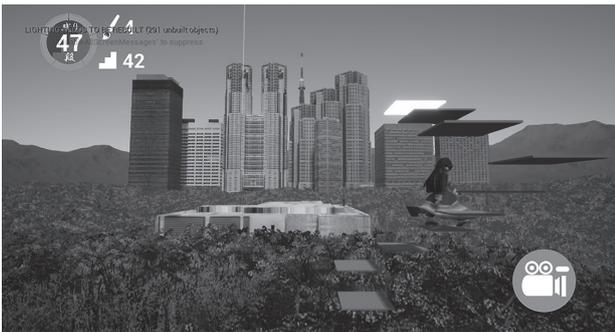


図 6. 月に帰るかぐや姫

AI管理のかぐや姫の制作だが、Unreal EngineではAIによる移動、すなわちコンピュータによる自動操作を行うには「Character Movement Component」もしくは「Floating Pawn Movement Component」のどちらかが必要となる。「Character Movement Component」は人型キャラクターを移動させるための非常に多彩な機能を持ち、「Floating Pawn Movement Component」はベクトルを与えるだけで移動するシンプルな移動機能を作ることができる。さらに「Nav Mesh」を使うことで、AIの移動する範囲を指定することが可能だ。移動に必要なノードは、「Simple Move to Location」「Simple Move to Actor」の2つがありControllerと位置、もしくはActorを指定するだけで簡単に移動させることが可能だが、移動後はゴールに辿りつけるか関係なく、その後にコールバックイベントが発生するといったことがないため、単純に移動したい場合には有効である。「AI

MoveTo」はControllerを必要とせずに、到達半径等ある程度細かい設定を行うことが可能である。位置、もしくは対象Actorを設定するだけで移動し、移動開始後すぐに一番上のピンが実行され、目的地に到達するとOn Success、到達できないことが確定すると"On Fail"がそれぞれ返ってくる。このノードは遅延 (Latent) ノードとして実行されてバックグラウンドでも処理が行われるが、遅延の調整ができないため、処理全体のスピードのコントロールができない。「Move to Location」「Move to Actor」は「Simple Move to」と同様に、AIControllerを渡して、位置もしくは対象Actorを設定すると移動することができる。到達半径の設定や他にも色々設定可能だが、「Use Pathfinding」のチェックを外すと、NaviMeshがなくても移動することができる。NaviMeshはAIの障害物や移動範囲を決めるため、AIを使う際には必要不可欠ではあるが事前計算が必要なため、今回のように後から階段が作成されるといった動的なイベントを苦手とし、また高さに関してもある程度の高さまでしか対応できない。ただしこれらのノードはコールバックイベントがないため、実行後に移動が開始できたかの値のみを返すだけだ。「Move To Location or Actor」は上記のノードのほぼすべてを内蔵しておりAIControllerを渡して、位置か対象Actorを設定すると移動開始し、「Use Pathfinding」を外すとNaviMeshなしで移動が可能で、到着したか失敗したかのコールバックイベントも帰ってくる。ただし、このノードは遅延 (Latent) ノードなため、処理スピードのコントロールができない。今回かぐや姫が階段を上るにあたり、階段の位置に正確に到着する、到着までのスピードのコントロールが可能である、次の段に進むために到着に成功したかのコールバックが必要である、以上の点から「Move To Location」を使ってAIのブループリントを構築した。このノードだけが遅延がなく移動が可能だが、コールバックがない。そのため、到着判定は目的のブロックの位置座標を取得して、

表 2. 移動ノードのまとめ

Simple Move to	AI MoveTo
コールバックイベント無 単純移動	コールバックイベント有 細かな設定が可
遅延の調整可能	遅延の調整でない
NaviMesh 必要	NaviMesh 必要

Move to
コールバックイベント無 単純移動
遅延の調整可能
NaviMesh 必要なし

自身の位置座標と幅を持たせた値とが一致したかで到着判断のブループリントを構築することで解決した。また、動的に作成される高さのある階段にはNaviMeshが不向きであるが、このノードはNaviMeshを必要としない。

ブロックの落下による階段の形成だが、各ステージの終了ブロックが次のステージの開始ブロックにならないと綺麗に繋がった階段を作ることができない。ただし、この場合階段の形成される方向によっては、次のステージのブロックが階段ではなく天井となり行き止まりになってしまうため、階段を形成するタイミングを2段階に分けて、終了ブロックまでかぐや姫が来ることで次のステージのブロックが落下し階段を形成する仕組みに変更した。かぐや姫と背景の制作はAutodesk Mayaで行い、テクスチャとマテリアルはSubstance Painterを使用して、Autodesk Motion Builderにてかぐや姫のモーションを制作した。

モーションは階段を上る前の待機、歩き、ゲーム終了時の待機と3つのモーションを制作し、Fbx形式でUnreal Engineに読み込み、ステートマシンにてモーションの切り替えを行っている。ステートマシンは、スケルトンが組み込まれているオブジェクト（Unreal EngineではActorと呼ぶ）であるスケルトルメッシュのアニメーションを一連のステート（連続アニメーション）に分割するグラフィカルな手法である。

ステートは、あるステートから別のステートにブレン

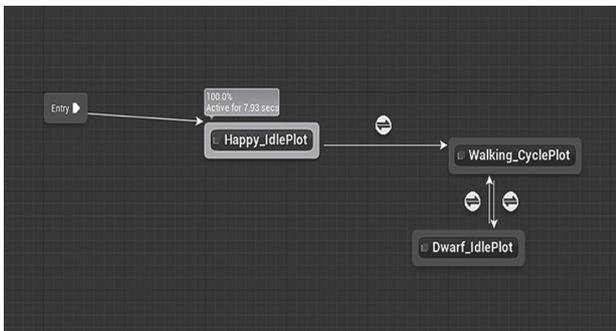


図7. ステートマシン

ドする方法を制御する「遷移ルール」によって管理され、複雑なブループリントを作成することなくアニメーションのタイプによってキャラクターがどのように動くかを簡単に制御することができる。そのため、スケルトルメッシュのアニメーションのデザインプロセスを非常に単純化することが可能である。「遷移ルール」自体はとってもシンプルな式によって構成されているため、ステートマシンを用いればブループリントと連動させる必要がなく独立して機能することができる。

Unreal Engineでは通常使うブループリントの他にア

ニメーション専用のアニメーションブループリントがある。同じブループリントの名前ではあるが、役割が異なるため通常のブループリントとアニメーションブループリントとの連携には少々手間がかかる。そのどちらのブループリントも複雑に組み上げることなくモーションのブレンドであるステート間の移動を行うことができる。例えば、歩きからゲーム終了時の待機モーションへの移行もかぐや姫が階段を登り切ったか、制限時間が来たのかをブループリントによって判断しなければならないが、かぐや姫は階段がない場合は自動的に歩くのをやめる仕様をブループリントで作成してあるので、追加で上記の条件に当てはまるブループリントの記述をすることなく、かぐや姫の移動速度が0.02以下ならば終了の待機モーションへ移行という簡単な式をステートに記述するだけで実装が容易に行える。

10.3 Unreal Engineでの制作上の問題点

Unreal Engineでは複数のカメラを切り替えて表示することも2つのカメラを1つの画面に同時に表示することも可能だが、2つの画面に違うカメラを同時に表示することはできない。空中結像の技術の欠点の一つに視野角の狭さがあり、正面から外れた位置からは空中結像を見ることができない。そのため、本コンテンツでは空中ディスプレイでひと筆書きを楽しんで頂くプレイヤー以外の方は前方の大型スクリーンにてかぐや姫が月へと上がっていく様を鑑賞できる映像コンテンツとしても制作したのだが、上記の理由から単独で2画面出力ができない。そのため、ネットワークを構築して2人によるマルチプレイを構築した。Unreal EngineのネットワークについてはServer Client（クライアント・サーバーモデル）を採用しているがエンジンがクライアント、サーバー両方をこなせる仕組みを持っており、最初からエンジンコアの部分でネットワークに対応する設計となっているので、シングルプレイヤーからマルチプレイヤーにするのも比較的簡単にブループリントのみで作れるように設計されている。サーバーについてネットワークにおける最高権限を持ち、ゲームプレイフローを管理し、クライアントはサーバーにログインしてから、サーバーからの情報を共有する。ListenサーバーモードとDedicatedサーバーモードがあり、どちらの手段であっても役割自体は同じで、サーバー機能を有しつつもクライアントのように動作することが可能である。Listenサーバーモードは見た目にも動作がわかりやすく、テストプレイもしやすいため特に問題がなければListenサーバーモードで行う。Dedicatedサーバーモードは、完全なサーバー専用モードであるためクライアントのような画面はなく、コマンドプロンプトのような真っ黒い画面

上でも動作し、最低限のリソースで動作するが、クライアントを処理するのにフルスペックで動作するため比較的大規模なネットワークのマルチプレイヤー向けのモードである。今回はListenサーバーモードによりマルチプレイを実行した。

展示予定の東京工芸大学2号館「メディアラウンジ」は7m×9mの台形型の大きな展示会場なため、100インチのスクリーンと4kプロジェクターによるプレイ画面の表示、空中タッチディスプレイの展示を行う予定であった。メディアラウンジ内は暗幕に囲まれ、薄暗い照明となる。メディアラウンジの天井にはダクトレールがあるため、照明にphilips社のhueを使用した。hueはデバイスから簡単にライトを制御することのできるパーソナルワイヤレス照明である¹⁴⁾。このhueは同じネットワーク内にコントローラーとしてのデバイス（iPhone等のスマートフォン）とその電球をコントロールするブリッジを同じWiFiネットワーク内に置くことで、Apple社のiPhoneやGoogle社のAndroidの携帯端末からアプリを利用して、電球を好きな色の好きな明るさにコントロールすることが可能となる。主な機能としては、電球のOn/Off、照明の明るさ（256レベル）・彩度（256レベル）・色相（65536レベル）の変更、複数の電球をグルーピングしてまとめて操作・一括管理が可能、タイマー機能による時間帯・時間での操作などがあるが、このhueのブリッジにはRESTful APIが搭載されている。RESTful APIとは、URL/URIですべてのリソースを一意に識別し、ステートレスにセッション管理や状態管理などを行わないが、リソースの操作は取得ならGETメソッド、書き込みならPOSTメソッドのようにHTTPメソッドによって指定されるため、結果はXMLやHTML、JSONなどで返され、処理結果はHTTPステータスコードで通知される。このRESTful APIを使うことで、Unreal Engine4から電球をコントロールすることは可能になるが、Unreal Engine4には制作段階でのバージョン4.22では、ブループリントを使ったhttpのリクエストをすることが出来ないため「LE Http Request」を導入した。このプラグインによってブループリントを使ってHTTPメソッドのGETやPOSTを扱うことができるため、hueに対して細かな電球の制御をブループリントで行うことが出来るようになった。このHueを使って選択したブロックの色に合わせて電球の色が変わり、合計9個の電球がブロックの選択した量に応じて光、クリア時には点滅するように構築した。これにより暗い空間内をゲームの内容に合わせて照らし出すことが可能となった。

10.4 展示方法の変更

このように展示に向けての準備を行っていたのだが、メディアラウンジの入り口には扉があり、防犯上の観点から常に開けた状態にしておくことはできない。暗幕で囲まれて入り口が閉まった状態では、同じく防犯上の観点から好ましくないと判断し、設営当日に展示方法の変更を行った。変更箇所は、メディアラウンジ外側ではポスターによる展示が行われているため、外側に面する暗幕以外の暗幕を撤去することで、もともとガラス張りで作られているメディアラウンジを外から見渡すことが可能となった。メディアラウンジ内が外光により明るくなったため、プロジェクターによる投影から65インチ4kテレビモニターへと変更した。使用したASKA3Dプレートは量産型であるASKA3D-200NTは、ディスプレイに対して照度が約15%と暗いが、このプレートの特性上明るい場所での空中結像は可能である。そのため、暗幕を取り除くことでの空中結像に対するデメリットはほとんどなかった。ここまでの作業で2019年芸術学部フェスタのレセプションの時間となってしまったため、機器の調整が未調整のまま作品のプレオープン形となってしまった。プレオープンを通じて以下の問題点を発見した。当初は複数人で訪れた場合を想定して、プレイヤー以外が画面を見られるように2台のディスプレイ



図8. 外向きの展示への変更

としたが、空中タッチディスプレイがメインであるはずが、2台のディスプレイを置くことで、どちらを見てよいかの判断が困難となっていた。また、空中タッチディスプレイで作業すれば当然もう片側のディスプレイはみることはないため、かぐや姫の設定は生きていない。フェスタのテーマである「結ぶ」にこだわったので、開始ブロックと終了ブロックの間をひと筆書きで結ぶことになりゲームの楽しさ、自由さを失っていた。プレイヤーは空中タッチディスプレイの視野角の悪さを前方にある4kテレビを見ることで補おうとしてしまう。そこで、かぐや姫が階段を上るシステムを廃止して、ひと筆書きにより集中できるシステムに変更した。ブループリントにより各アクターに役割を与える設計となっているUnreal Engineはこのような場合に大いに役に立つ。かぐや姫に関するアクターとブループリントを削除し、同時にネットワークも取り外しシングルプレイとすることで系統的にも身軽となった。その結果階段を登る都合上どうしても終了ブロックが必要であったが、終了ブロックも削除することができたため、ひと筆書きの方法が増えゲームの自由度が増した。暗幕を撤去して明るくなったメディアラウンジ内だが、天井のダクトレールからの照明の効果は高く、一部は床に別の照明器具として配置することで、外から覗いている人に対して作品のアピールを、さらに4kテレビモニターもプレイヤーから見えない外側に配置することで、外から覗いている人に対しての広告とした。



図9. 上下及び角度の調整が電動で行える機構

11. まとめ

今回制作した空中タッチディスプレイの問題点の1つに、上下の視野角がある。左右の視野角については、機器の正面に立つことである程度回避できるが視野角の上下に対しては、機器自体が上下する必要がある。空中ディスプレイの試作機作成の段階では上下及び角度の調整が

電動で行える機構の作成も行ったのだが、安定運用と耐久性の面から採用を断念した。また、踏み台のようなものを設置してプレイヤー自身で調整してもらう方法が考えられたが、プレイヤー自身と機器に対して衝突等の安全面からも課題点があるため、上下の視野角に関しての解決には今回は至らなかった。今後空中タッチディスプレイの精度とともに上記の問題点の改善とさらなる技術の洗練を行わなければならない。空中タッチディスプレイの発展としてはスクリーンの大型化による全身ジェスチャーによる入力操作やPC本体を機器に内蔵してのスリム化などが挙げられる。さらに、今回の入力デバイスのテストには、Kinect、Leap Motion、zForce AIRの3つを用いたが、intel Real Sense、Azure Kinectなど様々な入力デバイスがあり、今後も増えていくと考えられる。どのデバイスも長所と短所があり、それらを中継するゲームエンジンもUnity、Unreal Engine以外にもCRYENGINEやOROCHI、PhyreEngineなど多数存在する。これらの組み合わせによって比較的安価に、特別な加工や手間を必要とせずにコントローラーなどの特別なデバイスを必要としないインタラクティブなゲームアートの制作が可能となった。今後はUnreal Engineを中核としながら、複数のデバイスを組み合わせ予備知識がなくても誰もが楽しめるインタラクティブなゲームアートの発展に寄与したいと考える。

参考文献

1. Dejarik (閲覧日：2019年4月15日)
<https://starwars.fandom.com/ja/wiki/%E3%83%87%E3%82%B8%E3%83%A3%E3%83%AA%E3%83%83%E3%82%AF>
2. AirWitch (閲覧日：2019年4月15日)
<https://airwitch.jp/>
3. Aerial Display Reflector
https://www.carbide.co.jp/product/airial_display/
4. ASKA3Dプレート (閲覧日：2019年4月15日)
<https://aska3d.com/ja/>
5. Leap Motion (閲覧日：2019年10月26日)
<https://www.leapmotion.com/ja/>
6. Kinect (閲覧日：2019年10月26日)
<https://developer.microsoft.com/ja-jp/windows/kinect>
7. zForce AIR (閲覧日：2019年5月1日)
<https://www.neonode.com/zforce-air-touch-sensor/>
8. Unity Technologies (閲覧日：2019年10月26日)
<https://unity3d.com/jp>
9. Unreal Engine (閲覧日：2019年10月26日)
<https://www.unrealengine.com/ja/features/>
10. GitHub (閲覧日：2019年10月26日)
<https://github.co.jp/>
11. 永江孝規：“Leap Motion を用いたアート制作”、東京工芸大学芸術学部紀要,25,31-38 2019
12. Azure Kinect (閲覧日：2019年10月26日)
<https://azure.microsoft.com/ja-jp/services/kinect-dk/>

13. Real Sence (閲覧日：2019年10月26日)
<https://www.intel.co.jp/content/www/jp/ja/architecture-and-technology/realsense-overview.html>
14. Philips Hue (閲覧日：2019年10月26日)
<https://www2.meethue.com/ja-jp/philips-hue-benefits>