体感型ゲーム「風穴」での 3D都市モデルProject PLATEAUの活用

室橋　直人
ゲーム学科

Production example of the game "Kazeana" using Project PLATEAU

MUROHASHI Naoto
Department of Game
(Received October 31, 2022 ; Accepted January 19, 2023)

キーワード：非接触型デバイス、アフターコロナ、ゲームデバイス、Unreal Engine、Project PLATEAU

Abstract

In March 2021, the company presented the experiential game “Kazeana”. Kazeana is a game content that uses the Project PLATEAU 3D city model with a game device that strives to minimize the risk of infection under with-corona and after-corona. Kazeana is an experiential game, but it is difficult to display Project PLATEAU data in such a real-time application, and it is important to select a format, build a workflow, and reduce the amount of data to be displayed. This paper describes the verification of a method to use “Project PLATEAU” data in a game engine, which was conducted in a wind tunnel. FBX is a suitable data format, and it is important to standardize textures for each region.

1.はじめに

2019年12月頃から新型コロナウィルス(COVID-19)による新型コロナウィルス感染症が世界的流行を始めた。2022年6月の段階では、日本では収束しそうに見えているものの、中国ではロックダウンが続き、完全に克服する見通しは立っていない。新型コロナウィルスによって、社会活動の多くは自粛、もしくは縮小し、他者と一定の距離を保つソーシャル・ディスタンスの確保や、感染拡大を防ぐための対策が日常生活に定着しつつある。本稿では、現在のウィルスとともに生活していく時期を「ウィズコロナ」、新型コロナウィルス感染症が流行する前の生活を「ビフォアコロナ」、対抗措置が出来上がった後の生活を「アフターコロナ」と定義する。ウィズコロナ、アフターコロナにおける感染のリスクを極力減らすことに努め、3D都市モデル「Project PLATEAU」[1)]を用いたゲーム「風穴」を制作した[2)]。本稿では、「風穴」を作成する上で直面した、問題に対する解決案を技術的に説明する。

2. 背景

2.1 Project PLATEAU

Project PLATEAUは、国土交通省主導による日本の都市の3Dモデル化を推進するプロジェクトである。都市空間に存在する地形、建物や街路に対して、航空測量等で取得したデータから3次元のポリゴンモデルを構築し、そのデータに名称や用途、建設年といった都市活動情報を付与することで、都市空間そのものの再現を目指す3D都市空間情報プラットフォームである。Project PLATEAUの建物や地形は、第三者が著作権を有しているコンテンツ以外は、クリエイティブ・コモンズ・ライセンスの表示4.0国際のオープンデータとして公開されている。さまざまな都市の活動データが3D都市モデルに統合されることで、現実空間と仮想空間の高度な融合が可能となり、都市計画立案や都市活動のシミュレーション、分析の効率化が進むことが期待されている。また、都市の風景をゲームや仮想空間に取り込むことが可能なため、エンターテインメント分野等での活用も期待されている。

XR分野では、巨大な初音ミクが出現してライブを披露するXR[3]でProject PLATEAUが利用されている。この作品にはMicrosoft社の「HoloLens2」[4]が使われている。都市の3Dデータから仮想空間の座標系を計算してHoloLens2に投影する映像を作成している。他にProject PLATEAUを用いたXR分野の事例としては、映画のシーンを再現したような映像[5]、東京駅周辺を巨大アバターが歩く映像[6]や、ARグラスの「Nreal Light」[7]と組み合わせて、都市の外観をきらびやかに彩った作品[8]が作成されている。XR分野でのProject PLATEAUの事例は複数見られるが、その使用目的は主に、仮想世界をのぞき込む際の没入感を高めるために利用されており、プレイヤーが自らの体で操作する体感型のコンテンツに使用したものが多い。

そのため、体感型のコンテンツと相性の良いゲームにProject PLATEAUを利用することには意義があると考える。体感型のコンテンツとしてはNTTドコモが、松竹、歌舞伎座、歌舞伎座サービス、ポーラ、日産自動車と連携し、バーチャル空間上に再現された銀座、バーチャル銀座において、地域の魅力度向上や経済活動の活性化につなげる実証実験を2021年9月～2022年3月まで行っている[9]。数あるコンテンツの中のひとつに、ユーザ自身の3Dアバターで街全体をダイナミックに駆け回る体験を提供することを通じて、銀座の歴史・文化に触れながらその魅力を発見するゲームコンテンツがある。バーチャル銀座のようにリアルな世界を表現するには、さまざまなデータをもとにデジタル空間上にそっくりそのままに再現するデジタルツインを利用して3D空間が再現されており、ここでProject PLATEAUが使われている。

このようにProject PLATEAUを利用したコンテンツを組織や法人が公開しているものはあるが数が少ない。その理由は、デジタルツインで再現するためには、提供されている建物テクスチャの解像度が低く流用するのには困難なことや、Project PLATEAUのデータはクリエイティブ・コモンズ・ライセンスの表示4.0国際のオープンデータとして公開されているが、三者が著作権を有しているコンテンツ、例えばビルの看板に使われている広告や企業のロゴについては利用者自身の責任で利用の許諾を受けなければならない[10]ことも大きく関係していると考える。Project PLATEAUはすでにいくつかの利用例が公開されているが、本稿執筆時では個人が情報発信している段階のものが多い。例えば、大橋直記氏によるProject PLATEAUを使ったゲーム作品「Silhouette Guessr」[11]は、一人称視点の街の景色から自分が地図上のどこにいるかを当てるゲームだ。これらの個人発信の作品はProject PLATEAUの今後の可能性を感じさせるものが多く見受けられる。

2.2入力デバイスについて

デジタルゲームの入力に用いられる周辺機器の代表的なものとして家庭用ゲーム機に備え付けられるゲームパッドがある。「風穴」は、美術館等での展示を主たるターゲットとした。展示やアーケードゲームでは、不特定多数の人がプレイするため、念入りなアルコール除菌等の対策が必要であり、家庭用ゲーム機以上にデバイスの制限を受ける。ゲームパッドは、方向または位置を入力するためのスティックや指で操作する各種ボタンで構成されており、手でしっかりと握り続ける必要がある。しかし新型コロナウィルスは、飛沫や接触によって感染することから、不特定多数の人がプレイする状況ではゲームパッドは感染症を拡大させやすいため、非接触型のデバイスの方が望ましい。

感染リスクや消毒作業を減らすことができる可能性のある接触型デバイスとして、フットペダルなど足を使った入力が考えられる。しかしながら、不意に接触したり、ふざけてペダルを手で触る可能性がある。したがって、接触型デバイス自体を用いない方が感染症対策としては望ましい。ただし、非接触型デバイスであれば何でもよいわけではない。声を出すようなゲームは、唾を飛ばす機会が増えやすく、飛沫感染の可能性が高まる。したがって、マイク入力は避けるべきである。飛沫感染をさらに検討すると、コロナウィルスは汗で感染することはないため、体を動かすゲームはアフターコロナで忌避すべき存在ではない。ただし、汗で足元を滑らせて転倒した場合に唾などを飛ばす可能性があるため、感染リスクの点から考えると大きな移動を伴う運動を用いたコンテンツはアフターコロナでは好ましくない。

3.「風穴」の概要

3.1 コンセプト

大規模な3Dモデルを使う事で広大な都市空間内を自由自在に探索できるため、自由に移動することと体験を結び付ける方法を模索した。「自由な移動」というテーマでの移動方法をいくつか考えた末、通常はできない体験として、飛ぶ体験を選択した。本来、人は飛ぶことができないので、夢のような体験として自由に3D都市空間内を飛び

回ることが本作品のコンセプトとなった。

初期コンセプト決定後に大規模都市から連想されるキーワードを練った。ネオン、新宿、高層ビル、人工物と自然物、首都高速道路、乱立、圧迫感、閉塞感、煌びやかなど複数のキーワードが出てきたところで、首都高速道路を車で運転する際に閉塞感や圧迫感を感じながら、乱立するビルの間を左に右に走り抜けていく経験を思い出した。このことから、「密集」を第2のキーワードとして、乱立するビルや建物の間を右に左にかわしながら進む作品へとコンセプトを変更した。広大な風景をゆったり自由自在に飛ぶ作品も魅力的ではあるが、素早く判断して左右に移動するような自由度が少ない体験の方が「密集」していることを表現できると考え、素早いテンポのゲームとした。

以上のようなコンセプトに対して、複雑にならない操作方法を模索した。飛ぶ体験や、飛んでいる物体・生物を操作すると考えたときに、最初に思い浮かんだのは飛んでいるポーズである。スーパーヒーローが拳を握って片手を上につきだし、もう一方の手を折り曲げて拳を握るポーズがある。しかしながら、このポーズでは、体を横にする台のような接触する機材を用いるか、上を向いて天井を見るなど不自然な態勢での体験となる。考えをさらに進めたところ、子供のころに空を飛ぶマネとして、両手を広げて飛んだ気になって走っていた姿を連想した。腕の上下を左右操作にも用いることができることから、手の操作による入力方法を採用した。最終的に、翼にたとえた両手を使ってキャラクターを操作して、迫ってくるビルの間を素早く飛び抜けるコンテンツとした。

3.2 コース設計

ビルの間を素早く抜けていくためには、自由に飛び回れるよりはある程度システム側で誘導してルートを決めておくと演出などが行いやすい。そのためコースの設計が重要となる。今回は首都高速道路をコースとして選択した。特に首都高速道路の中央環状線は、密集した都市部に高低差を伴いながら右に左に曲がるようなレイアウトのため、都市の中を飛び抜けるコースとしてふさわしい。また、作品展示が東京工芸大学中野キャンパスのラウンジであったため、東京の地形が多くの体験者にとってはなじみ深いという点もコース決定の要因の一つである。

ルートの作成において、ビルを利用した閉塞感や圧迫感の制御を考慮している。スタート地点は、周りのビルが比較的低く全体が見渡せて圧迫感が少なく、また、操作方法に慣れてもらうための比較的真っすぐな道であることを考慮して、首都高速7号線の江戸川区とした。首都高速7号線を新宿方面に進み、錦糸町を通過して両国橋ジャンクションから、首都高速6号線に入る。右に左に曲がりながら密度高いビル群の間を進み、江戸橋ジャンクションから首都高速環状線外回りへ進む。皇居の周りを通って、トンネル内で首都高速4号線へと進み、新宿出口でおりて一般道で中野区の東京工芸大学を目指す経路とした。各エリアにはチェックポイントがあり、制限時間内にチェックポイントを通過することで次のエリアに行くことができる。そのため、コース上に制限時間を延長するための加速装

図1　加速装置「風穴」

置を模した図1のようなアイコンを配置した。

加速装置である「風が吹き抜けるポイント＝風穴」を通過することで加速して、持ち時間が増えるシステムとし、この「風穴」をコンテンツのタイトルとしてロゴの制作を行った。「風穴」を通過しにくくするための障害物も見た目を変更した物を数種類置くことにした。ただし、障害物に当たっても時間が減るようなルールは廃した。これは飛び抜ける感覚を長く経験してもらいたいためである。スピードは若干落ちるが、スタート時の速度を下回らないように最低速度を設定しているため、素早い感覚を軽減しないルールとした。障害物は、上記のコース設計にあわせて作成し配置している。例えば、スタート直後は遠景まで見渡せるレイアウトになっているため、障害物は高く見やすく配置した。また、スピードの乗るエリアでは左右にテンポよくかわせるようにし、高密度なビル群ではコースが上下から合流・分岐するため、車線変更を促す障害物を置いてバランスを取った。

4.「風穴」の実装

4.1 入力デバイスの選定

非接触型の入力デバイスには、カメラやセンサーを使用した機器がある。Leap Motion[12]は、2012年にLeap Motion社から販売されたUSBでPCと接続する小型のデバイスである。Leap Motionに手をかざすことで、手のジェスチャーによってコンピューターへの入力が可能となり、マウスや画面タッチを用いずにコンピューターを操作できる。

Microsoft社から2010年に発売されたKinect[13]は、マーカーレスモーションキャプチャーにより体をコントローラーとする「NUI」(ナチュラルユーザーインターフェイス)の一つである。RGBカメラ、深度センサー、マルチアレイマイクロフォン、プロセッサで構成されており、特殊なマーカーの付いたスーツの着用や、マーカー検出時に使用するトラッカーなどが無くてもプレイヤーの位置、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っている。

今回は全身を用いるKinectを採用した。Kinectのような全身を使った大きなコントロールの方が、Leap Motionでの手のジェスチャーによる小さなコントロールよりも大規模な都市空間内を飛んで移動する感覚を持たせやすいものと考えた。しかし、Kinectの導入には問題がある。Kinectは2017年に生産を中止しており、新品の入手は困難である。後継機としてAzure Kinect[14]が発売されているが、代替品として使用する場合は機能の検証が必要である。今回、開発ツールとして、Unreal Engine 4.26(以下UE4)とAutodesk社のMayaを採用した。UE4は標準ではKinectをサポートしていないが、UE4でKinectを扱えるようにするための拡張プラグインである「Neo Kinect」[15]は開発が続けられており、制作で使うバージョンのプラグインを導入することができた。Neo Kinectは、ビジュアルスクリプトのブループリントを使うことで最大5人までの手首を含む全身25の部位の位置を計測することが可能である。「風穴」では、一番近くにいる人物を同定し、その両手首の位置の計測結果を入力に採用した。 人によって手首の最大、最大位置は異なる。したがって、タイトル画面の後に手首の位置を計測するためのジェスチャーを用意し、画面に合わせて両手首を上下させる指示を提示することで、プレイヤー毎の手首の位置の可動範囲を計測した。計測した可動範囲を用いて手首の位置を分析することで、どちらかの手を上に上げているのか、もしくは両手が水平なのかを検知することが可能となった。この仕組みを応用することで、ゲーム中のキャラクターを、右手を下に左手を上に上げた場合は右方向、右手を上に左手を下にした場合は左方向にキャラクターを動かす制御が可能となった。Kinectにはマイクがあるが、アフターコロナにおいてデバイスとしては好ましくないため、機能をオフにしている。

4.2 パスに沿いながらのプレイヤーの自由な移動

コースに沿ってプレイヤーキャラクターを動かすシステムを構築した。

最初にコースをなぞるようにパスを作成して、そのパスの上をキャラクターが移動する方法を検討した。しかし、この手法ではパスの上に位置情報が拘束されてしまうため、コースから外れたりすることはないが、パスから離れて移動することができない。 次に検討した手法は、キャラクターの階層の上に不可視な空のオブジェクトであるロケーターを配置して、親であるロケーターをパスに沿って動かし、子であるプレイヤーは親であるロケーターに引っ張られながらも自由に動かす方法である。この手法も移動の自由度は向上したものの、大局的にはパスに拘束され

てしまい自由に動かすことができない。これらを解決する手法として、親子関係を構築するのではなく、ロケーターをプレイヤーの前方のパスに拘束する形で配置し、プレイヤーは常にロケーターの方向を見ながら、近づこうとする手法とした。この手法により、パスに沿いながらも自由に移動するという課題が解決された。この手法の利点は、パス（UE4ではSpline Mesh）の各ポイントは3方向の軸をもっているため、単に飛行予定のルートをなぞるだけでなく、上下を伴った立体的なルートを作ることが可能となることである。ルート作成に使用したSpline Meshを複製して、コースに使用する道路や街灯などのオブジェクト（UE4ではStaticMesh）を配置することができる。今回のように道路を設置する場合はコース用のパスと道路となるStaticMeshを用意するだけで、Spline Meshに沿って道が構築される。ただしこの機能はSpline Mesh上にあるポイント間にStaticMeshをスケールしながら配置する手法であるため、街灯や鉄柱のような長い形状のオブジェクトを配置すると、形が歪む。Spline Meshに沿って配置するというよりもSpline Meshで押し出された結果となるため、コースの制作に習熟が必要である。

4.3 ProjectPLATEAUのデータフォーマット

Project PLATEAUでは、公開する3D都市モデルの主たるデータフォーマットに「CityGML 2.0」[16]が採用されている。このフォーマットは、地理空間情報分野の国際標準化団体であるOGC（Open Geospatial Consortium）が策定した標準フォーマットである。CityGML形式ではフォルダにデータが分かれており、udxフォルダ内にあるbldgフォルダに建物が、demフォルダに地面が、bldgフォルダに橋などの建造物が収められている。各フォルダには、CityGML形式の.gmlファイルが収められており、.gmlファイルにLOD1、LOD2の情報が記録されている。また、appearanceフォルダにTIFFフォーマットのテクスチャが収められている。gmlファイルは無料で入手できるFZKViewer[17]で閲覧可能である。都市の3Dデータは、CityGMLのフォーマットだけではなく、3D Tiles、obj、FBX等の各種データ形式でも公開されている。これらのフォーマットを選択することで、特別なコンバーターを自作ないしは用意することなしに、各データをさまざまなソフトウェアやツールで用いることが可能である。CityGMLフォーマットのデータでは、3D都市モデルの詳細度に応じてLOD（Level of Details）が設定されており、4つのレベルに分けられている。

- LOD1：建物の2D形状に高さ情報を付与したシンプルな箱モデル
- LOD2：LOD1に屋根形状を追加したモデル
- LOD3：LOD2に窓やドアなどの外構（開口部）を追加したモデル
- LOD4：BIM/CIM等で建物内部まで再現したモデル

このうちLOD1ではテクスチャは存在しない。Project PLATEAUのデータは、大規模であるが現時点では均質化されていない。大都市のモデルでは、地域ごとに用意されているLODが異なっている。例えば、東京23区では区に応じてテクスチャやモデルの詳細度が異なっており、一番低いLODしか用意されていない場所に差し掛かると、一部分だけ簡素なモデルがテクスチャなしの状態で表示されることになる。また、テクスチャには太陽光や影などの映りこみが多い。撮影されている時間帯はモデルによって異なっており、この意味でもデータをそのまま使用することは難しい。また、使用できる都市は日本全域ではない。2021年度の制作時点では56都市となっている。

4.3.1 CityGML形式

今回、UE4でデータを扱うための各種フォーマットを検証した。CityGML形式は、UE4で使うためにDataSmithフォーマットに変換する必要がある。DataSmithフォーマットへの変換は、ソフトウェア FME-Desktopで可能である。FME-DesktopはProject PLATEAUの公式GitHubで提供されているプリセットファイル「citygml2datasmith.fmwt」[18]を開くことで変換することができる。ただし、変換処理に数十時間を要した。変換後のDataSmithフォーマットは、数十のアセットを持ち、マテリアル及びテクスチャ数も1000を超えた。この状態では、UE4で読み込むことがかなり難しい。読み込めたとしてもそのままの状態でデータをインタラクティブに操作することはできないと推測できる。従って、大規模なCityGML形式のデータをUE4で取り扱うには、アセットのマージ等のデータの最適化が必要となる。

4.3.2 FBX形式

FBX形式ではメッシュは建物毎に分割されており、LODによってはさらに建物内でも分割されている場合が

ある。建物ごとにマテリアルの設定とTIFFテクスチャが設定されている。しかしながら、TIFFはUE4で読み込むことができない。同じファイル名でPNGも提供されているため、MEL等で全て変更する事も可能ではあるが、工数が増加するため好ましくはない。 区画ごとに建物をまとめることでポリゴン数・アセット数を軽減することは可能である。

「風穴」を制作していた2021年10月の段階ではテクスチャはTIFFで提供されていたが、2022年10月現在はJpegで提供されておりUE4に読み込むテクスチャに関する弊害はなくなっている。

4.3.3 OBJ形式

OBJ形式では、建物は1つのメッシュに統合されて保存されている。どちらの形式もマテリアルとテクスチャ数は膨大であり、整理が必要である。

4.3.4 「風穴」で採用した形式

CityGML形式は、UE4で使うためにDataSmithフォーマットに変換する必要がある上に、マテリアル及びテクスチャ数も膨大であるため、採用を見送った。全ての建物を使う場合は、区画ごとに統合されているOBJ形式が扱いやすいが、一部の建物を削除する場合などに建物が個別になっているFBX形式の方が扱いやすい。また、データを軽量化するために建物をまとめていく必要があるが、「風穴」では区画を設けて建物等を管理しているため、区画ごとに建物をまとめる場合は、1つの区画に1つのマテリアル・UV・テクスチャとなるのが望ましく、これはOBJ形式では困難である。以上の理由から今回はFBX形式を採用した。

5. 制作時の問題の解決

5.1 データ変換のワークフロー

マテリアル・UV・テクスチャをまとめるためのワークフローを構築した。今回整備したワークフローは、Mayaを用いた以下の手順である。

・必要となる区画のデータをFBX形式で読み込む
・Mayaにインポートした後、不必要な建物を削除する
・シーン内にある必要な建物のオブジェクトをレイヤに入れた後に、複製して新しいレイヤに複製した建物のオブジェクトを入れて非表示にする
・可視化されているレイヤ内にある建物のオブジェクトを結合して一つにまとめてからマージを行って頂点を結合する。この際に一つの建物に一つのUVアイランドとすることで、UVシェルでの選択による建物単位で作業が可能となる
・全ての建物のUVを0〜1の範囲内に収める
・新たなマテリアルを作成して設定を行う。 不可視になっているレイヤを可視状態にして、テクスチャレンダリングを行い既存のテクスチャを新しいUVに転写する。転写の際にファイルの形式をPNG等のUE4で読み込める形式に指定する

このワークフローを用いることで、一つのUV、一つのマテリアル、一つのテクスチャでUE4に読み込める形式に変換することができる。また区画ごとのオブジェクトとなるため、管理が容易となる。

5.2 テクスチャがない地域に関する対応

Project PLATEAUのデータは、地域によって最も詳細なLODレベルが異なり、テクスチャが存在しない地域や、テクスチャの解像度が低い、または濃い影が入っているなどでテクスチャが使えない地域がある。都市部ではテクスチャが入っている場合が多いが、それ以外の地区は入っていないことが多い。「風穴」では、コースレイアウトはプレイ体験から事前に決定したため、都市部と都市部を繋ぐようなコースレイアウトとなっており、テクスチャの入っていない地区も多数存在した。 メッシュの選定として、どの地域でもデータが提供されているLOD1で進めたが、テクスチャが用意されていないため、別途テクスチャを用意する必要があった。統一感を考えると、ゲーム中で使用するすべてのテクスチャを用意すべきであるが、既存テクスチャを一切使わずに新たに全てのビル群のテクスチャを零から制作するのは、膨大な工数を必要とするため現実的ではない。そこでゲームの世界観を幾何学的なビジュアルデザインとした。コース沿道にあるビル群とそれ以外のビル群とにオブジェクトを分けてマージし直した。この際に、Maya内でゲームのプレイ状態と同じアングルのカメラを設置し、このカメラから見えない建物のオブジェクトは削除し、ビルの間などの間隔を調整して密度をあげる対応を行った。修正を行った建物は、ビューポートのオブジェクト数を参考におおよそ同じ建物数となるよう

図2　建物のエッジが光るビル群

に1地区内を複数のブロックに分割した。1ブロックに1マテリアル・1テクスチャとして、UVを自動で展開したUVをテクスチャとして書き出し、マスク用のテクスチャとして適用することで、建物のエッジが光る煌びやかなビルを表現することができた(図2)。

なお、地面のオブジェクトdem.fbxは、膨大なポリゴンを使って細かな起伏まで再現されているため、非常に扱いにくい。そのため読み込んだ後に地面を削除して、簡単な平面で代用した。その後、高速道路の写真を参考にしながらコースを修正した。

5.3 フレームレートの向上

シンプルなテクスチャにしたといえども、Project PLATEAUのデータをリアルタイムに表示するのは負荷が高い処理である。画面に表示されるポリゴン数を削減することで、処理速度を向上することができる。そのような仕組みを実現するために、用意されている地域番号ごとにActorBluePrintを作成した。各地域のActorBluePrintは、変数によって自分自身がどの地域であるのかを認識すると共に、プレイヤーが各地域を通過した際に自身が消滅するように組まれている。さらに地区を管理するブループリントを別途用意して管理をしている。タイトルが表示されている間にシーン内に地区データを読み込んでいるが、読み込まれる地区はスタート時に見える地区のみとしている。 地区を管理するブループリントは、通過した地区に応じて、その先のいくつかの地区だけを表示することで、ポリゴン数の多いアセットである街並みの表示の負荷を抑えている。同様な処理を風穴や障害物などのコース以外のオブジェクトにも設定している。ActorBluePrint内の変数は「編集可能」に設定するとエディター上で変数の入力が可能となる。例えば、障害物を配置する際に、配置したエリアの番号をその場で変数に入力することができる。これによりActorBluePrint内に変数を書き込むことなく、どのエリアにある障害物なのかを障害物自身が認識することが可能となる。プレイヤーが障害物の設置してある地区を通過した場合、不要となった障害物は削除される。これにより、シーン内のオブジェクト数は削減され、応答性が高いゲームを実現することができた。

6. 結果

実際のゲームをプレイしている様子が図3である。65インチのディスプレイの上部にKinectを配置して、2m程度離れた場所でゲームを操作する。体全体を使ってコントロールするのはかなり難しいが、高速でビルの間を飛び抜けていく疾走感が感じられる。ただし1回のプレイ時間は約3分であるが、腕を伸ばした状態を保つのが難しいという問題があった。

図3　ゲームプレイ時

7. まとめと今後の課題

Project PLATEAUが提供しているデータはFBXやOBJなど汎用性の高い形式で提供されているが、データの扱いは決して容易ではない。適切なフォーマットを選択し、データを扱いやすいものに変換するワークフローが構築できれば、一から構築するのが困難な正確な街並みの3Dデータをコンテンツで利用しやすくなる。

Project PLATEAUを使用してアフターコロナにおいて体感型により適した非接触型の入力デバイスであるKinectを用いたゲームを制作することができた。しかしながら、完成したコンテンツが閉塞感漂う中、ビルの間を素早く飛んで抜けていく体感が得られたかのゲームデザインとしての検証が行えていない。今後、効果検証を進めながら、さらなるコンテンツを制作していきたい。

参考文献

1.国土交通省：“Project PLATEAU” (最終確認日:2021年5月3日)
https://www.mlit.go.jp/plateau/
2.室橋直人：“ウィズコロナ、アフターコロナ時代におけるProject PLATEAUを利用した体感型ゲーム「風穴」の提案”, 映像情報メディア学会技術報告, vol. 46, No 10, pp.285—288 ,2022
3. YORIMIYA (最終確認日:2022年1月30日)
https://twitter.com/jav6868/status/1375416195101327362
4. Microsoft HoloLens2 (最終確認日:2021年12月30日)
https://www.microsoft.com/ja-jp/hololens/hardware
5.龍lilea / Ryo Fujiwara(最終確認日:2022年1月30日)
https://twitter.com/lileaLab/status/1376853912775323648
6.Michigari(最終確認日:2022年1月30日)
https://twitter.com/Michigari/status/1377184149212078080
7.Nreal “Nreal Light” (最終確認日:2021年5月13日)
https://www.nreal.ai/light/
8.jojomon(最終確認日:2022年1月30日)
https://twitter.com/jojomonvr/status/1376475300263993347
9.国土交通省：“Project PLATEAU Use Case” (最終確認日:2022年12月9日)
https://www.mlit.go.jp/plateau/use-case/uc20-023/
10. 国土交通省：“Project PLATEAU Site Policy” (最終確認日:2022年12月9日)
https://www.mlit.go.jp/plateau/site-policy/
11.大橋直記 “Silhouette Guess” (最終確認日:2022年1月31日)
https://sorabatake.jp/21929/
12.Leap Motion(最終確認日:2019年10月26日)
https://www.leapmotion.com/ja/
13.Kinect(最終確認日:2019年10月26日)
https://www.playstation.com/ja-jp/accessories/playstation-move-motion-controller/
14.Azure Kinect (最終確認日:2022年12月10日)
https://azure.microsoft.com/ja-jp/products/kinect-dk/#overview
15.Rodrigo Villani “Neo Kinect” (最終確認日:2020年7月12日)
https://www.unrealengine.com/marketplace/ja/product/neo-kinect
16.Open Geospatial Consortium：“OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0. Technical Report”, (2012).
17.FZKViewer(最終確認日:2021年5月22日)
https://www.kkaneko.jp/dblab/citygml/fzkviewer.html
18.citygml2datasmith.fmwt(最終確認日:2021年5月22日)
https://github.com/Project-PLATEAU/Data-Conversion-Manual-for-3D-City-Model