

# ゲームエンジンの産業分野への適用

## ——スマートデバイスに適したアプリケーション

### Industrial Field Applications of Game Engine

#### —— Application Suitable for Smart Device

紺野 正宏

**要約** スマートデバイスが一般に普及してきた2012年現在、ビジネス利用も急速に進んでおり、日本ユニシス・エクセリューションズでもスマートデバイスへのアプリケーション対応が求められている。スマートデバイスアプリケーションの開発は、様々な種類のデバイスへの対応という課題がある。この課題に対して、ゲームソフト開発ツールであるUnityのマルチプラットフォーム対応を解決策の一つとした。Unityの産業分野への適用として、スマートデバイス上での3Dアプリケーションを想定した。まず第1弾として、住宅向け設計CADであるDigiDのデータを利用したiPad用ウォークスルーアプリケーションの“Room-Jumper”を開発した。本稿ではUnityを利用したスマートデバイスのビジネスモデルとスマートデバイスアプリケーション開発について報告する。

**Abstract** After all smart device has become universal, utilization of it in business is also increasing rapidly and UEL Corporation is required development of applications for smart device. Regarding the development, there is problem how to approach devices of various types. And we came up with one solution that multiple platforms development of Unity: one of game software development tool. As a result of examining industrial field application of Unity, 3D application on smart device was considered as a target. As the first stage, we developed walk-through application “RoomJumper” for iPad using data of DigiD that is design CAD (Computer Aided Design) for housing. This paper reports business model of smart device using Unity and development of applications.

## 1. はじめに

2008年のiPhone3G, 2010年のiPadやAndroidタブレットの発売を機にスマートデバイスが著しい普及をみせ、生活の一部になってきている。その流れはビジネスにも浸透しはじめ、BYOD (Bring Your Own Device) という言葉に表れているように、ビジネスのありようも変わりつつある。

本稿では、スマートデバイスに焦点を当て、ゲームエンジン「Unity」<sup>\*1</sup>を利用したアプリケーション開発事例とUnityの技術について述べる。2章にて日本ユニシス・エクセリューションズ株式会社（以降、UEL）と3Dデータ、スマートデバイスの関連について述べ、3章でUnityの概要を説明する。4章でビジネスモデルとスマートデバイスアプリケーション“Room-Jumper”の開発について解説し、5章で課題を整理する。

## 2. スマートデバイスにおける 3D データ

UEL の主な業務は CAD/CAM (Computer Aided Design/Manufacturing : コンピュータ支援設計/製造) システムの製造・販売である。住宅設計 CAD 「DigiD」\*<sup>2</sup>、製品設計 CAD 「CADmeister」\*<sup>3</sup> といった CAD ソフトウェア製品を提供している。顧客は製造業と住宅業が主であるが、近年、作業現場などへ持ち運べるスマートデバイスを利用して、UEL の CAD システムのデータを参照したいといった要望が増えている。そこで、UEL の CAD/CAM アプリケーションで作成した 3D データの現場利用を想定し、実現方法の検討を開始した。

既存プロダクトを単純に移植する方法では、スマートデバイスのハードウェアのスペックでは快適な動作が困難であることは明らかである。機能を限定し、スマートデバイスに特化したアプリケーションの開発も検討したが、一から作り上げるにはかなりの工数が必要と判明した。そこで、「データの活用」に主眼をおき、3D データと親和性が高いゲームエンジンとの組み合わせに着目した。

2012 年現在、スマートデバイスのシェアは、iPhone/iPad などの Apple 製品が主流だが、Android タブレットや Windows タブレットも相次ぎリリースされ、今後のシェア分布の予測は難しい。特定の OS やスマートデバイス向けのアプリケーション開発はリスクが高く、またスマートデバイスごとのアプリケーション開発ではビジネス上、採算が合わない。以上の条件から、アプリケーション開発環境としてマルチプラットフォーム対応のゲームエンジン「Unity」を選択した。Unity を利用すれば、3D データを制御する基本機能は Unity にまかせ、その他の、業務に特化した機能開発に注力できるため、効率よくアプリケーション開発ができるからである。

## 3. Unity とは

Unity はオブジェクトの作成やコーディングなどの作業が容易なゲームエンジンとエディタの統合環境である。2001 年にデンマークで開発が始まり、2011 年には 75 万人の開発者に利用されるまでになった。

Unity での開発作業は基本的にドラッグ&ドロップのみである。Unity は、3.2 節で述べるゲームに必要な各種ミドルウェアを取りそろえており、基本機能だけの使用ならば、短期間でアプリケーション実装が可能で、アルゴリズムなどゲーム自体の構築作業に集中できることが特徴である<sup>[1]</sup>。

### 3.1 マルチプラットフォーム

Unity は、iPhone/iPad、Android や Flash、WEB、PC などの様々なプラットフォームに対応した統合型開発環境であり、アプリケーションを一度構築するだけで、前述のプラットフォーム全てに移植可能である<sup>[2][3]</sup>。

Unity はベースに Mono を採用している。Mono とは .Net Framework のオープンソース実装であり、さらにソースコードを CIL (Common Intermediate Language) と呼ばれる中間言語にコンパイルし、それが VM (Virtual Machine) 上で動く (図 1)。このため、移植性が高くマルチプラットフォーム対応を実現している。

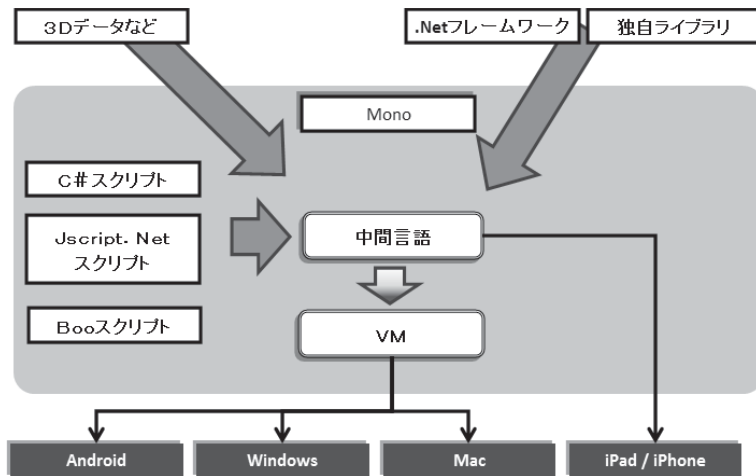


図1 Monoの構成

### 3.2 ミドルウェア

Unity 開発環境には高性能な各種ミドルウェアがあらかじめ統合されており、高機能の実装が低コストで実現できる。代表的なミドルウェアは以下の二つである。

ライティングには Beast ライトマッピング<sup>\*4</sup> が組み込まれている。画面上に表示される影を事前に固定表示しておけば、影に対する実行時の計算処理が必要ない。これによりパフォーマンスを劣化させずに、光と影を表現できる。

物理演算処理を司る物理エンジンが、PhysX<sup>\*5</sup> である。剛体物理演算に対応しており物体の落下および回転、衝突シミュレーションをプログラミングせずに実装できる。さらに布表現や柔らかな物質のシミュレーションなども取り扱い可能である。

### 3.3 AssetStore

Unity の開発において利用する各種パーツを Asset という。画像ファイルや 3D モデル、スクリプトなどすべて Asset として扱われる。Unity で利用可能なこれらの Asset は AssetStore と呼ばれるオンラインストアから、無料もしくは低価格で入手できる。それらのパーツを組み合わせるだけで、コードを書かずにアプリケーションを実装することも可能である。本稿で説明するアプリケーションの開発でも、アニメーションを伴う 3D キャラクターなどは AssetStore から入手し利用した。

このようにして、開発コストを削減できることも Unity の特徴の一つである。またユーザは AssetStore で自ら作った素材を販売することもできる。

## 4. ビジネスシナリオ

2章で述べたように、UEL は住宅設計用の DigiD、製品設計用の CADmeister といった CAD ソフトウェアを開発・販売している。3D データとスマートデバイスを組み合わせ、顧客の求めるアプリケーションを具現化する環境は整っている。本章では 3D データとスマートデバイスの組み合わせによるビジネスモデルについて考察する。

#### 4.1 スマートデバイスの活用

スマートデバイスが利用されるのは、主にお客様と対面する営業活動の場や、工場や住宅施工などの製造現場であると想定した。例えば、お客様と商談しながら画面を見せて補足説明したり、工作機械を前にしての操作など、他の作業と並行して利用する際に、スマートデバイスの特徴を活かすことができる。

#### 4.2 クラウドを利用したマスビジネス

Apple の App Store や Google の Android マーケット\*<sup>6</sup> などを利用すれば、世界中のユーザーにアプリケーションを提供できる。データ活用にはクラウドコンピューティングを利用し、次々に作成されるデータを社内LAN接続という閉じられた環境ではなくクラウド上に置けば、いつでも、どこでも最新データを参照できる。

3D データを操作するためのアプリケーションは App Store 等で提供し、ユーザーごとに異なるデータはクラウドに保存して必要な都度入手できるようにする。この仕組みがスマートデバイスを活かすモデルだと言える（図2）。

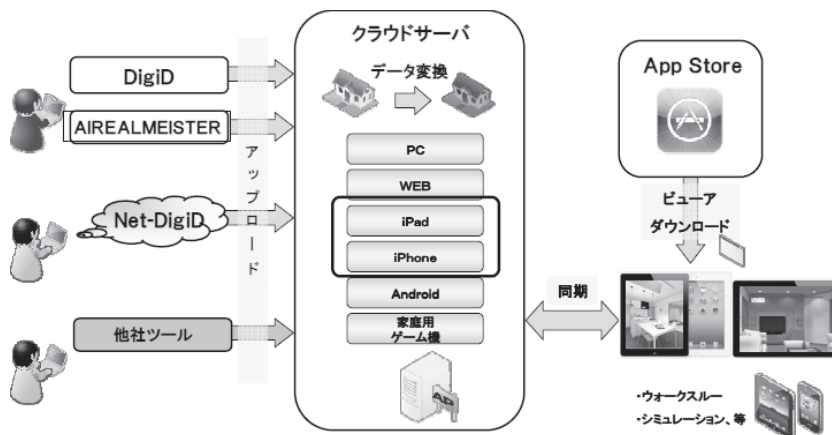


図2 クラウド利用イメージ

Unity では 3D データは独自のフォーマットである AssetBundle 形式（4.3.1 項で述べる）で取り込む。そのためユーザーが自分の持っているデータを扱いたい場合は、AssetBundle 形式に変換する必要がある。クラウド上に任意のデータをアップロードし、AssetBundle 形式に変換するクラウドサービスの提供により、ユーザーは外出先で任意のデータをスマートデバイスにダウンロードできる。

#### 4.3 住宅分野での活用事例

前節のスマートデバイスとクラウドを組み合わせたビジネスモデルにおいて、UEL の住宅用 CAD システム「DigiD」が扱っている CAD データを利用して、家の内観・外観を確認できるウォークスルー機能をスマートデバイス上で実現させるアプリケーション“Room-Jumper”を開発した（図3）\*<sup>7</sup>。基本要件は下記の通りである。

- ① DigiD で出力した家モデルデータを入力する

- ② 住宅の内観・外観を確認することができる
- ③ iPad で扱えるアプリケーションとする
- ④ 家モデルのデータは随時入れ替えることができる
- ⑤ 内観・外観確認はネットワーク非接続状態でも可能とする

これらの要件の実装にあたり利用した Unity の利用技術を本節で述べる。

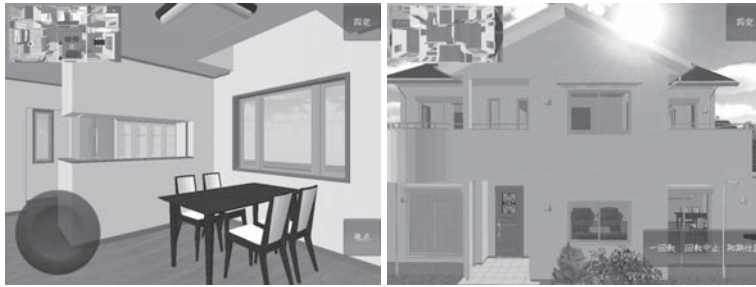


図3 Unityで作成したスマートデバイス用アプリケーション“RoomJumper”

#### 4.3.1 AssetBundle 形式

Unity では AssetBundle 形式という外部素材フォーマットが用意されている。その中には 3D データは勿論、各種文字情報やアルゴリズムをセットで保持できる。AssetBundle 形式のメリットは、アプリケーションサイズを小さくできることである。iPhone アプリケーションなどを公開する App Store では、3G 回線でアプリケーションをダウンロード可能とするには、サイズは 20MB (2012 年 3 月より 50MB まで可能) 以下という制限があるからである。Unity ではこの AssetBundle ファイルを利用してアプリケーション本体のサイズを抑制している。さらにアルゴリズムなども AssetBundle ファイルに取り入れることができるため、最適なデータ提供が可能である。

AssetBundle 形式のデメリットは、プラットフォーム間でデータの互換性がないことである。Unity ではプラットフォームごとにビルド作業が必要になるが、AssetBundle 形式のファイル出力でも同様にビルド作業が必要になる。このため、プラットフォームごとに外部ファイルも用意する必要がある。

この AssetBundle 形式のファイルをクラウド上で管理して、必要な都度、デバイスへダウンロードする構成とした。

#### 4.3.2 カメラ設定

Unity では容易にカメラ設定ができる。RoomJumper ではこのカメラ機能を利用し、家モデルに対して一定の高さから俯瞰するカメラを配置して家の間取り図を用意した (図4)。さらに移動物体の現在の座標値と連携して、家の中を移動しながら現在位置を地図表示させるなど、ゲームエンジンならではの機能も簡単に実装できる。

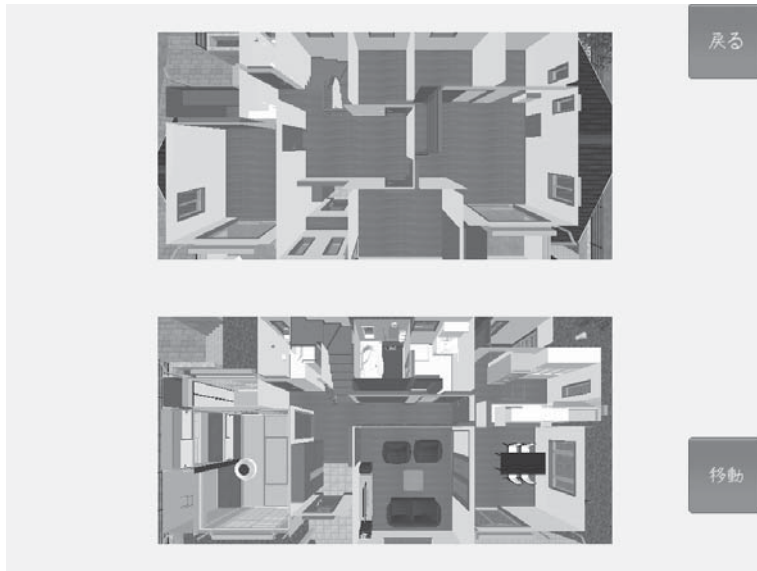


図4 RoomJumperの間取り図機能

## 5. スマートデバイス用アプリケーション開発における課題

本章では、Unityの特徴を利用してアプリケーションを実装する過程でいくつか見えた課題を整理する。

### 5.1 操作性

スマートデバイスの入力には基本的に指によるタッチ操作である。スマートデバイスはデータの視聴に重きを置いているのでこれが最良の方法と考えるが、ビジネス利用では、入力操作が

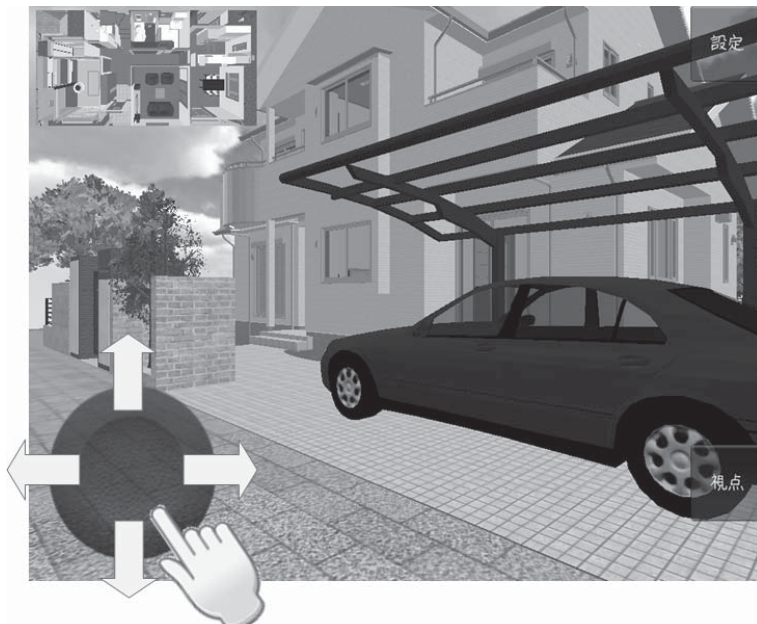


図5 ジョイスティックイメージによる移動指示

必要な場合が多い。今回開発した RoomJumper でも、指の操作による移動指示が必要であり、ジョイスティックイメージを画面上に表示して移動量と移動方向をタッチ操作で直感的に操作できるようにした(図5)。ただし、指の動きに対する感度が最適化できておらず、思うように移動できない問題が発生している。最適化にはデータの蓄積とゲームソフト開発のノウハウが必要と思われる。

## 5.2 データサイジング

顧客はこれまでの PC 上でアプリケーションを操作してきた感覚でスマートデバイスでも同様に操作できることを望んでいる。当然の要望であるが、2012年現在のスマートデバイスのスペックでは自ずと限界がある。これまでの実験ではおよそ50万ポリゴンが限界で、これ以上となるとスムーズな3Dデータの操作は難しくなる。

ゲーム業界ではFPS (Frames Per Second) 値は30~60と言われている。ただし、RoomJumperのウォークスルー機能では30万以上のポリゴン数からなる3Dデータを使用するため、スマートデバイス上では、このFPS値の維持は困難であった。実測で、10FPSを維持できるように開発したが、これが使用に耐えられる最低限の数値である。

## 5.3 テライン

RoomJumperではDigiDから出力した敷地を含めた家モデルの3Dデータを、あらかじめUnityで作成したテラインと呼ばれる地形オブジェクト上に配置した。このテラインの作成もUnityの特徴といえるが、今回は単なる草原での表現にとどめた。実際の家の着工現場は住宅街など人工物の中にあることが多い。リアリティを出すためには、実際の住宅街に近い地形に対応する必要がある。

## 6. おわりに

スマートデバイスの普及に伴い、ビジネス分野への対応の必要性が高まりつつある。本稿ではビジネス分野でのスマートデバイス活用の取り組みの一例として、3Dコンテンツと親和性が高いゲームエンジンを利用した開発手法を取り上げた。ゲームエンジンの中でもUnityはマルチプラットフォームに対応したスマートデバイスアプリケーションの開発に有効である。UnityはGUIに不向きな面もあるが、利用シーンを精査し、スマートデバイスの利用環境に最適化させて、その能力を引き出したい。UELは今後も、Unityを利用して様々なスマートデバイスアプリケーションを開発し、3Dデータの現場利用という顧客の要望に応じていく所存である。

- 
- \* 1 Unity Technologies 社のゲーム開発用ツール
  - \* 2 日本ユニシス・エクセリューションズの住宅CADシステム
  - \* 3 日本ユニシス・エクセリューションズの3次元統合CAD/CAMシステム
  - \* 4 Autodesk 社のライティングツール
  - \* 5 NVIDIA 社の物理エンジン
  - \* 6 2012年3月、Google Play に改称された。
  - \* 7 2012年10月31日現在、App Store への登録はまだ審査中である。

- 参考文献** [1] 和泉信生, Unity マスターブック, 株式会社カットシステム, 2011年9月  
[2] 高橋啓治郎, Unity 入門, ソフトバンククリエイティブ株式会社, 2011年10月  
[3] 山崎透, 野口基之, Unity で作る iPhone/Android アプリ入門, 株式会社マイナビ, 2012年6月

**執筆者紹介** 紺野正宏 (Masahiro Konno)

1993年日本ユニシス・エクセリュションズ(株)入社。流通、金融業務のシステム開発に従事。2010年より各種先端技術の研究・開発グループに所属。

