



VR 学習カリキュラムにおける統合開発環境の活用事例

Case Study of Integrated Development Environment in VR Learning Curriculum

橋口哲志, 松村耕平, 木村朝子, 野間春生

Satoshi Hashiguchi, Kohei Matsumura, Asako Kimura, and Haruo Noma

立命館大学 情報理工学部

(〒525-8577 滋賀県草津市野路東 1-1-1, hasiguti@rm.is.ritsumei.ac.jp)

概要:仮想現実感 (Virtual Reality; VR) の要素技術を組み合わせた作品制作を通して, VR 技術を学習する情報系学部向けカリキュラムについて報告する. 新たな試みとして, 従来の GLUT などの開発環境ではなく, HMD などの機器を効率よく利用するためにゲームの統合開発環境をプラットフォームとして採用した. 本稿ではそのカリキュラムの設計と実際に情報系学部生に授業として運用した結果について報告する.

1. はじめに

仮想現実感 (Virtual Reality; VR) または, 複合現実感 (Mixed Reality; MR) は各種のセンサ, シミュレーション, ディスプレイの各要素技術から構成されている. 近年, 各要素技術の発展により, インタフェース技術が充実し, 各機材も安価になり, 実用性が向上している. また, シミュレーションに関しては, 物理シミュレーションが組み込まれたゲームエンジンが普及してきた. 一方, VR・MR 技術の設計・実装するには, 各要素に関する基礎知識やこれらを取り扱うためのプログラミング能力が必要となり, 初学者が VR・MR 技術を学ぶことは容易ではない.

我々は学部生向けに VR・MR 技術を学習させる講義を実施している. 本カリキュラムは, まず VR を構成する各要素技術の成り立ちを学習させ, その要素技術を組み合わせた作品制作を通じて, VR・MR 技術を学習する. 本稿では, 本カリキュラムに導入した新たな試みを紹介する.

昨年までのカリキュラムでは, 作品制作として, 受講者グループに VR・MR 作品のデザイン, および Java3D や GLUT を用いたその実装を行ってもらっていた. 作品制作は, VR・MR の各要素技術を組み合わせることで効果的に技術を習得し, プログラミング能力の向上につながることを意図していた. しかし, このカリキュラムにおいては, 主にプログラミングに時間をとられ, VR・MR 技術の構成要素である, 外部のセンサなどとの接続に手が回らないなどの問題点があった.

そこで, 本年度は, 昨年までのグループワークでの作品制作実習のノウハウを活かしつつ, 新しい試みとして, UnityTechnologys 社の開発したゲームエンジンである統合開発環境 (以下, Unity) をプラットフォームとして採用した. このような統合開発環境を採用することで, 従来

は容易に実装できなかった新たな要素技術の組み合わせや GUI でのプログラミング, 内蔵された物理エンジンによる実装が可能となり, バリエーションが豊富な作品制作に取り組むことができる.

以降では, 本学情報系学部 3 回生向けの実習科目として開発した VR 学習カリキュラムの設計と実際に授業として運用した結果について報告する.

2. カリキュラムの設計方針

本実習の目的は, VR の各要素技術を組み合わせた作品制作を通じて, VR を構成する仕組みを理解することにある. 本年度から一新したゲームの統合開発環境を用いたカリキュラムでは, 利点して多くの要素技術を効率的に組み込めること, GUI による効率的なプログラミング, 物理エンジンによるシミュレーションが容易に行えることがある. 一方, 統合開発環境を用いることで, 懸念される問題点もある. 例えば, 内蔵された物理エンジンによって仮想物体の落下や衝突などのシミュレーションが容易になるが, そのときに生じる内部処理にあまり注目されない. これは, 本実験の目的に反するものであり, カリキュラムを設計する上で考慮しなければならない.

そこで, 本カリキュラムでは第一段階として, まず受講者全員に Linux 環境で C 言語を用いたボールの落下シミュレーションを実装する課題を与えた. この課題では, 簡易的な物理法則を GLUT により実装し, CG アニメーションによって実行結果を描画する物理シミュレーションの過程を学習させることをねらっている.

次に第二段階として, 統合開発環境でも同様にボールの落下シミュレーションを実装させる. 簡易的なシミュレーションの一例ではあるが, その過程を事前に実装させることで以後の統合開発環境で実装時に内部処理を意識させ

る。この課題の中で物理シミュレーション時の内部パラメータに敢えて違いを施すことによって、Linux 環境での落下シミュレーションとの差異や考察させた。

第三段階では、Unity そのものの習熟を目的として、用意した教材を使って簡易的なゲーム環境を実装させた。

カリキュラムの最終段階では、グループワークによる VR 作品制作を行う。各班のアイデアで VR の要素技術を組み合わせ制作する。作品制作の前に行った課題によって得られる Unity の基礎知識や VR 環境の内部処理を意識させることは重要な意味をもつ。作品制作では、役割分担を行い、作業を進めていく。Unity の基礎知識は作業の効率を上げ、VR 環境の内部構造を意識することで、自分の役割を認識し、作業を進めていくことができる。

3. カリキュラムのスケジュール

本実験は本学情報系学部 3 回生の 70 名を対象とし、1 週に 2 時限分 (1 時限: 1 時間 30 分) を 7 週かけて行う。対象の学生は、VR に関する講義を別途受講している。

受講者は 1 週目に、前述の第一、二段階、個別課題に取り組み、VR シミュレーションの内部処理と Unity の基本操作について学習する。2 週目に第三段階の Unity の基本操作を学習する。図 1 の実験室で教員・TA の教示をうけながら、課題を作成する。3~6 週目にかけてグループワークで VR 課題作品を制作する。1 グループは 6~7 名から構成され、合計 10 グループに分かれて制作した。最終週に作品を発表する。各週の実施内容は以下である。

第 1 週目 VR シミュレーションの実装

Linux 環境で C 言語を用いてボールの落下シミュレーションを実装する。CG 表示には GLUT を用いる。次に、Windows 環境で Unity を使い、全く同じ自由落下シミュレーションを実装する。両者の結果を比較し考察する。

第 2 週目 Unity によるゲームの実装

Unity による VR 環境の実装について学ぶために、用意した教材を使ってゲーム環境を実装する。まず図 2 に示すようなキャラクタを投げつけ、敵キャラクタに衝突させる 2D ゲーム[1]、および、図 3 に示すような 1.5 人称型の障害物競走ゲームを実装する[2]。

第 3~6 週目 グループワークの実装

第 3 週に目指す課題作品の目標とシナリオを決定し、その後 6 週目まで開発を進めていく。開発が早期に完了した班は、当初の目標以上に機能を追加し、作品の質を向上させる。グループワークの様子を図 4 に示す。また、役割分担は以下を参考にしよう指示した。

[ディレクタ: 1 人]

グループの進行状況を確認したり、メンバに必要な指示を出したりするプロジェクト管理業務を行う。目標設定・シナリオ作成の指揮も積極的に行う。

[プログラマ(インタフェース担当): 2 人]

作品制作作業のうち、実験に用意した入出力インタフェースについて調査し、これを Unity 上で利用するためのプロ



図 1 第 1, 2 週目の実習の様子

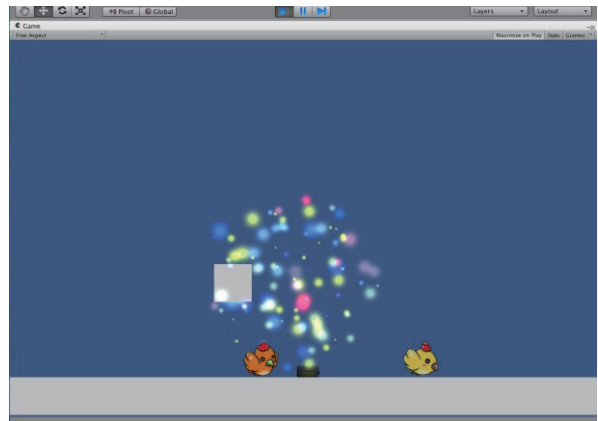


図 2 制作させた 2D ゲーム

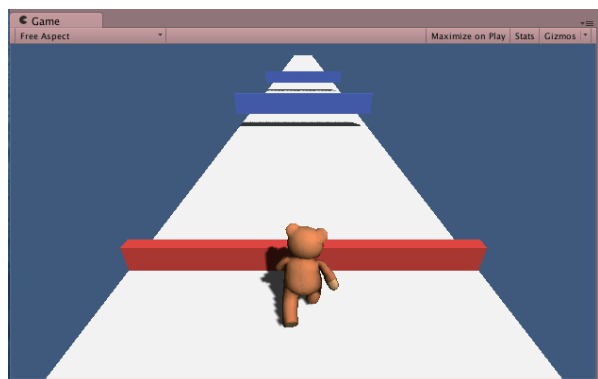


図 3 制作させた 1.5 人称型の障害物競走ゲーム



図 4 グループワークの様子

グラミングを担当する。

[プログラマ(シミュレーション担当): 2人]

作品制作作業のうち、主に Unity におけるオブジェクトの挙動ディスプレイ機能や操作結果に対するシミュレーション機能のプログラミングを担当する。

[モデラ: 2人]

作品に用いる CG オブジェクトやキャラクターのモデリング、音響素材などの作成を行う。

第7週目 課題作品の発表・体験・評価

各班が、約 5 分間で自分たちが作り上げた課題作品について、全員の前で口頭発表とデモをする。

4. 開発環境と機器構成

4.1 開発環境について

本実験では、Unity Technologys 社の開発したゲームエンジンである統合開発環境の Unity を用いた。Unity は、OSX と Windows 環境で実行され、ゲーム空間を CAD のように GUI ベースで構築しながら、C#や JavaScript でスクリプトを加えて開発を進める。Unity は、OpenGL に基づいたグラフィックスエンジンと、nVidia の PhysX 物理エンジンを内蔵しており、実行環境の能力に応じたクオリティで CG のリアルタイムレンダリングと衝突判定処理、物理挙動のシミュレーションを実行できる。また、内部で用いる CG オブジェクトや追加機能であるプラグインを自由に導入可能であり、極めて拡張性が高い。さらに、それらのデータやプラグインを開発者間で流通させるマーケットとしての AssetStore も活用できる。

本実験で Unity を採用した理由の一つとして、HMD などの VR 機器を効率よく使用できることにある。今回、使用できる機器は以下の機器となる。

- ・Oculus Rift HMD
- ・Wii リモコン
- ・バランス Wii ボード
- ・USB ゲームパッド
- ・Kinect
- ・MR システムとタブレット

本実験で用意した各種インタフェース機器を接続し、Unity を用いてこれらを統合して課題作品を開発できる。各機器のサンプルデモを参考にして、作品制作を進めていく。課題作品の制作には原則として OS X のデスクトップ PC 環境を利用して開発する。この環境に加え、Windows のデスクトップ PC とラップトップ PC を準備し、CG モデリングソフトであるメタセコイアを使用できるように準備した。

4.2 機器構成

本実験で用いる機器について、以下に説明する (図 5)。

- ・Oculus Rift HMD
両眼立体視が可能であり、視野角も 110 度と広い。頭の動きを内蔵するモーションセンサにより取得できる。
- ・Wii リモコン、バランス Wii ボード



図 5 機器構成

Bluetooth によって OS X 環境に接続し、ゲームパッドや動作入力として利用可能である。Unity には専用のプラグインを導入して接続する。

・USB ゲームパッド

Unity においては汎用のゲームパッドとして利用され、移動方向やアクションのボタン入力として利用できる。Unity ではスクリプトによってボタン入力を取得できる。

・Kinect

Unity で Kinect を使用するためのライブラリ zigfu ZDK を用いて、様々なジェスチャを認識したりすることができる。人を検知するだけでなく、レンジファインダとしても利用できる。

・MR システムとタブレット

Unity の統合開発環境内で Qualcomm 社が提供する AR 開発ライブラリ Vuforia を用いることによって、MR 空間を構築する。Unity の環境では、タブレットへのアプリ転送も容易である。Web カメラやタブレット 内蔵のカメラからマーカを認識し、MR 空間を構築できる。

・実施結果と考察

4.3 VR シミュレーション課題と Unity によるゲーム課題 (第一、二、三段階)

ボールの落下シミュレーションを Linux 環境で GLUT を用いた実装と Unity による実装の課題を行った後、インタビューを行った。

第一段階の Linux 環境で GLUT を用いた落下シミュレーションを実装に関して以下の意見がみられた。

- ・オイラー法による落下アルゴリズムをプログラムに変換することが難しいと感じた
 - ・落下シミュレーションは実装できたが、内部パラメータまで確認できていなかった
- また、第二段階の Unity による落下シミュレーションを



図6 成果発表の様子

表1 作品制作で使用された機器

使用された機器	使用した班の数
Oculus Rift HMD	17
Wii リモコン	8
バランス Wii ボード	5
USB ゲームパッド	5
Kinect	1
タブレット	0

実装に関して、以下のような意見がみられた。

- ・ GLUT を用いた場合より、簡単に作成できた
- ・ オブジェクト構造が把握しやすい

以上、インタビューの結果から考えて、GLUT を用いた実装はサンプルコードを用意しても、数式のコード化などに手間がかかり時間がかかった。それに比べて、Unity で落下シミュレーションを実装した場合、Unity を使用したことのない初学者でも、30 分程度で落下シミュレーションを実装できた。Unity の場合、コードの記入なしで実装ができ、より簡易的に感じた可能性がある。

また、レポート課題で各シミュレーション結果と解析解を比較させ、VR 空間でシミュレーションについて理解をより深めさせようという意図した。GLUT を用いた実装の場合、実装時には内部パラメータの差に気づけなかったが、レポートにより再度確認することで差異を認識していた。

4.4 グループ課題について（最終段階）

最終段階の実装について述べる。成果発表の様子を図6 に示す。表1 は計 20 班中、実際に作品制作で使用された機器の数となる。大半の班で Oculus Rift HMD が使用され、学生のほとんどが HMD を用いた開発に興味があったと考えられる。VR 空間とインタラクションで、最も多く使用されたのが Wii リモコンであった。Wii リモコンは、ゲームパッドとして多くの学生に馴染み深い機器であり、多くの班が使用した。バランス Wii ボードを使用した班は、主にスケートボードなどに乗ったときなどの体重移動での操作入力に使用していた。Kinect はダンスゲームを作

成した班だけ使用していたが、技術的に導入が難しいという意見もあり、使用されることは少なかった。MR 用に準備したタブレットは、Oculus Rift HMD がほとんどの班で使用されたこともあり、利用されなかった。

複数の装置を組み合わる班もあった。Wii リモコンを複数使用したり、バランス Wii ボードと USB ゲームパッドを組み合わせた。要素技術を複数組み合わせることによって、作品の新規性を出すように工夫されていた。

Unity を用いた作品制作について、学生にインタビューを行い、意見を聴取した。インタビューから Unity を使用した際の利点・欠点を抽出した。

以下は、Unity を使用した際の利点・欠点について述べられたものである。

- ・ GUI ベースなので、オブジェクトの構造がわかりやすい
- ・ 導入知識があまりなくても取りかかれる
- ・ 実装に時間がかからない
- ・ サンプルも多く、すぐに反映できる
- ・ コーディングがすぐ反映されるので理解しやすい

以下は、Unity を使用した際の欠点となる。

- ・ 操作手順が多く、間違ってしまうことがある
- ・ 個別で作成したプログラムを統合する際、手間がかかる
- ・ 高度な機能を拡張しようとすると、急に難しくなる

以上の実装結果とインタビューから、統合開発環境を用いることで、導入知識があまりなくとも習得でき、多くの要素技術を取り込みやすい環境が構築できることがわかる。Unity では GUI を用いて構造から制作していくため、見た目にもわかりやすく、プログラムが苦手な学生でも取り組みやすいと考えられる。

しかし、Unity を使用する上で2つの問題点があった。

1つ目は、グループワークを行う際、個別で作成したプロジェクトを統合しにくいことがわかった。Unity ではグループプログラミングの環境が用意されておらず、グループワークを行う際に手間がかかってしまうことである。2つ目は高度な実装になると急に難しくなることである。特に C# や JavaScript でスクリプトを加えて開発する際、スクリプトで高度な機能を拡張しようとすると難しいという意見が見られた。作品制作していく上で、機能の拡張は必須である。拡張する機能の数に従ってスクリプトの量が多くなった。

5. まとめ

情報系学部生に授業として、統合開発環境を用いた VR 学習カリキュラムの事例を紹介した。統合開発環境を用いることで、導入知識があまりなくとも習得でき、複数の要素技術を組み込み合わせた作品制作が可能となった。また、プロジェクトの統合や高度な機能拡張が困難な点など、内包された問題点もあることがわかった。

参考文献

- [1] <https://gist.github.com/4586393>
- [2] <https://gist.github.com/makoto-unity/6094663>