

独自ツールキットを用いた ゲーム制作教育への取り組み

渡辺大地

東京工科大学

ゲーム制作教育とプログラミング

筆者の本務校である東京工科大学メディア学部では、2005年度よりゲーム制作の教育と研究に取り組んでいる。今でこそ、ゲーム制作を教育や研究で扱う大学は随分増えたが、当時は希少であった。大学でゲーム制作教育を行うことが難しい理由はいくつかあるが、その1つは大学の多くの学部学科が専門領域ごとに縦割りに構成されていることである。近年のゲーム制作では多様な専門家が必要であり、従来の学部学科の中だけでは人材がまかなえないことが多い。メディア学部では多様な専門性を持った教員が揃っており、ゲーム制作を実習を通して学ぶ土壌があったことが幸いした。

しかしその一方、学際型の学部では学生の素養が「広く浅く」となってしまう面も否定できない。ゲーム制作において、それが最も切実となるのはプログラミングである。ゲームは、遊ぶ立場としては大変親しみやすいが、プログラミングという観点からいうと実は初学者にはかなり難しい対象である。特にリ

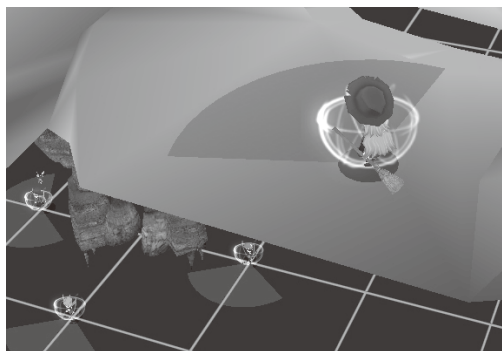


図-1 FKによる開発画面

アルタイム 3D を利用する場合、プログラム言語だけではなく数学や物理の素養も必要となるため、挫折する場合も多々見受けられる。

これに対応する方法にはさまざまな教育手法が考えられるが、本学での取り組みの1つとして独自ツールキットである「Fine Kernel Toolkit」(以下「FK」)を用いて学生に教育を行うという方式を採用している。図-1は、FKを用いて制作されたゲームの開発途中の画面出力の様子である。

独自ツールキットについて

FKは筆者らが開発しているリアルタイム 3DCG用のツールキットであり、オープンソースソフトウェアとして一般に公開している。FKはC++言語での利用を想定したクラスライブラリ群であり、内部 3D API として OpenGL を採用している。基本的には、C++ と OpenGL が利用できる環境においては FK を構築することが可能であり、Windows, Mac OS X, Linux といった各種 OS 上で動作確認と検証を行っている。同様の目的を持つライブラリの多くは特有のプラットフォームでしか開発や実行ができないものが多いが、FKにおいてはそのような制約はない。これは学生同士での共同開発のように、さまざまな異なる環境下で開発を進める必要がある場合に都合が良い。

FKは、もともとは筆者が自身や所属組織の 3DCG 研究を支援する目的として構築したものであり、現在でもゲーム制作だけでなく 3DCG や人工

知能などの研究で多く用いられている。研究においては、さまざまなほかのライブラリやシステムと連携を取る必要が多いが、コンテンツ作成を目的とするツールキットやシステムではその点が念頭に置かれておらず、不可能であったり困難である場合が多い。FK では、できるだけほかのシステムとの共存が可能であることを前提として設計を行っており、シェーダ技術^{☆1}や GPGPU 技術^{☆2}といった 3DCG の最新技術と同時に利用することができる。図-2 に FK を用いた研究の実行例を示す。

FK の設計としての特徴に座標変換の扱いがある。座標変換は、3DCG プログラミングの初学者が最初に遭遇する難関といえる。これは、平行移動・回転・拡大縮小の各種変換を組み合わせるために線形代数の知識が必要となることや、空間内で扱っている対象物と座標変換の関係が複雑になりがちなためである。そこで FK では、空間内の個別の座標系に「モデル」という概念を与え、そのモデルの方向や位置を具体的に指定することで想像を容易にし、かつ各座標系間の関係 (= モデル間の関係) をインスタンス同士で明確に指定する機能を持つこととした。これにより、利用者は座標系概念を意識することなくオブジェクトの移動制御を行える。たとえば、モデルを中心とした任意軸の回転は通常は困難であるが、FK の場合は点 A, B を通る軸を中心に角度 θ 回転する操作を

```
model.glRotateWithVec(A, B, theta);
```

と 1 行記述するだけで表現できる。また、モデルと幾何形状を階層的に扱うことができるため、オブジェクトのグルーピングやモーション制御といった技法も容易に扱うことができる。具体的には、2 つのモデル parent と child に対し、

```
child.setParent(parent);
```

と記述することによりモデル間に親子関係が設定さ

.....
^{☆1} グラフィクス描画ユニット (GPU) の内部描画アルゴリズムを変更する技術のこと。

^{☆2} GPU を描画以外の計算に用いる技術のこと。

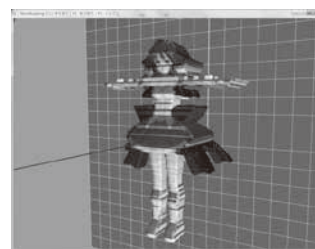
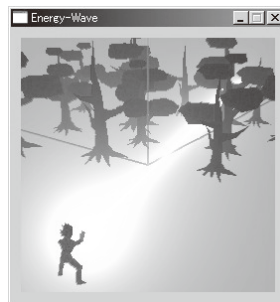


図-2 FK を用いた研究例

れ、parent の位置や方向を操作すると child も連動するようになる。

■ モーション・アニメーション作成ツールについて

リアルタイム 3DCG を用いたコンテンツにおいて、重要な要素の 1 つにアニメーションがある。FK は、形状変形に関しては高度な処理を提供しているが、形状全体の平行移動、回転移動に関してはプリミティブな命令しか用意しておらず、一連のモーションとしての動作は逐次プログラミングを行う必要がある。ゲームや映像作品においてキャラクターやオブジェクトを表現する際には、1 つのモデルに対してさまざまなモーションを用意することが一般的であるため、直接モーション制御をコーディングすることは非効率的である。そのため、モーションをデータとして保持し、任意のタイミングで再生できるような構造が望まれる。

3DCG におけるアニメーションを作成する際には、モーションキャプチャを用いた手法や、Digital Content Creation (DCC) ツールと呼ばれるソフトウェアによって手付けを行う手法などが多数存在する。しかしこれらの高機能なソフトウェアは、単純なモーションの製作用途には過剰ともいえるスペックを持っているため、操作方法が困難になりがちである。また、DCC ツールが出力するデータ形式は扱いが困難であることが多く、自作のプログラム上で自在に制御可能な環境を整えるには多大な労力を要する。

そこで、筆者の研究グループは FK と連携が可能なモーション&アニメーション作成ツールを開発し、



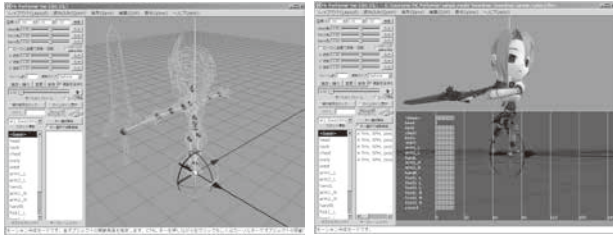


図-3 FK Performer の実行画面

ツールキットの利用者に提供した。このツールを「FK Performer」と呼称する。FK Performerによって作成したモーションデータを簡易に再生可能なAPIを用意することで、モーションを伴うゲームや映像などのコンテンツ制作が容易な環境を構築した。なお、このFK Performer自体もFKを用いて開発した。

□ FK Performer 実装の概要

図-3に、今回開発したFK Performerの実行画面を示す。

FK Performerは、フリーのモデリングソフトである「メタセコイア」のファイル形式に対応しており、複数のオブジェクトで構成されたテクスチャ付きのモデルデータを扱うことができる。一般的なDCCツールではリギングと呼ばれている工程に相当するが、FK Performerではアニメーションをオブジェクト単位に限定し、とにかく簡単にポージングができることを最優先とした。

□ FK との連携

FK Performerによって作成したモーションデータは、アスキー形式による独自データ形式で出力する。FK Performerを構成するクラスコンポーネントを再利用することで、FKによるプログラム中で独自データを容易に再生できるAPIを提供した。コンテンツ開発者が利用するのは、メタセコイア形式のモデルデータと親子関係の構築データを読み込んだ後、作成したモーションデータを複数読み込ませることが可能なクラスである。モーションの再生処理は、1フレームごとにモデルの状態を更新する形式で行うため、コンテンツの内容に応じてモーションの中断や切り替えなどが柔軟に制御可能に



図-4 演習科目学生作品の実行画面

なっている。

授業・演習での利用について

CGの学習では、プログラミングによる演習が大きな効果を持つ。本学においても、多くのCG系の科目においてプログラミングを用いた教育を行っている。その一例として、前章で紹介したFK Performerを利用し、アニメーションコンテンツを制作する授業を2年次生向けに開講した。この科目では、理論的な面よりもモーションに関する基本的な機能を教え、その後に3、4人のグループごとに自由作品を制作する。図-4は本演習科目での学生制作作品のスナップショットである。

標準的なカリキュラム体系とは別に、本学部では「プロジェクト演習」というユニークなカリキュラムを設置している。これは、標準的なカリキュラムだけでは物足りない学生を対象とした発展的学習を進めるために、各教員が独自に内容を設定する演習であり、感覚的には「道場」や「部活動」に近い。

プロジェクト演習では、まず1年次前期にアナログな題材で「遊び」についての本質を学ぶ。具体的には、新たなトランプゲームの創作や、割り箸鉄砲を題材とした遊びの提案などを通じて「面白さ」の原因を探求する。1年次後期では、全員にプログラミングを必修とし、インタラクティブなゲームに必要な技術的基盤を身に付ける。この段階では全員がFKを用いて少人数グループにてゲーム制作を行う。2年次からは、企画、プログラマ、グラフィック、音音楽などの役職を明確にしつつグループに分かれて制作を行っていく。最終的には、3年次の秋の段階での作品を毎年開催されている「東京ゲー

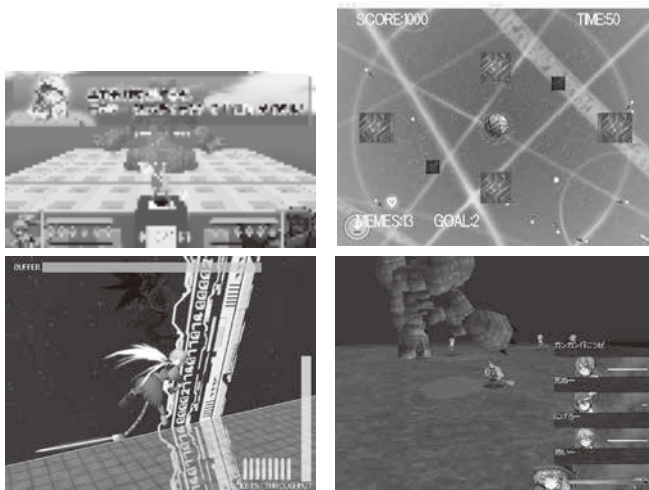


図-5 FKを用いたゲームの実行画面

ムショウ」に出展する。また、世界的なゲーム開発イベント「Global Game JAM」へも毎年多数の学生が参加している。

2年次から取り組むゲーム制作では、開発に利用する環境、言語、ツールは全て学生が自由に選択することとしている。調査や学習もすべて個々に任せるものとしている。ただし、教員や上級生による助言はある。時には、本来の企画に沿わない技術を採用することや、学生の持つ素養では対応できないような難しい技術を採用することもあり、それが主因となって制作自体を取りやめてしまうことも生じる。プロジェクト演習は必修科目ではないので、そのような「成功の保証がない道程」というのを経験する貴重な機会であるというのが、本科目の指針の1つとなっている。

図-5は、FKを利用したプロジェクト演習やGlobal Game JAMでの成果作品である。

ゲーム制作教育についての今後の展開

近年においては、ゲーム制作の現場ではライブラリを用いたプログラミングよりも、ゲームエンジンを用いるケースが増えてきている。ゲームエンジンを用いることで、プログラムの記述量をかなり減らすことができ、制作工程は大きく短縮された。このような状況から、専門学校や大学におけるゲーム制作教育でも、ゲームエンジンを採用する例が増えて

いる。これは、プログラム開発を主体とするゲーム制作と比較し、開発工程の削減が期待できることが理由である。

本学のゲーム教育グループにおいても、上級学年でゲームデザインを学ぶことを主目的とする科目においては、積極的にゲームエンジンを採用している。しかしながら、初学段階ではFKを用いたプログラミングによるゲーム制作を行うという方針をとっている。ゲームエンジンはたしかに企画案のプロトタイプには優れた効率性を発揮するものであり、簡単なゲームを完成させるために必要な学習量はかなり少なくすむ。しかしながら、このことから学生側がゲームエンジンの機能に沿った企画案しか発想しなくなるということや、高度なプログラミング技術が必要となるようなシステム構築を避けるような懸念が生じる。実際、ゲームエンジンを用いて制作を行う学生のプログラミングスキルは、以前のプログラミングによるゲーム制作を行っていた学生よりも低い傾向が見られる。ゲームエンジンを用いたゲーム制作教育は、コンテンツデザインを学ぶ上ではメリットも大きいですが、少なくともプログラミングを自主的に学ぶという目的の場合は一考の必要があると考えられる。

FKは現在C++だけでなく、C#やF#といった.NETを主体とする新しい言語でも記述できるように拡張を行っている。ツールの自主開発は、教育や研究の本来の目的と道具としての整合性を自らコントロールできるという強みを持つ。今後もゲームエンジンのような新ツールとの併用を踏まえつつ、より良い教育や研究が行える環境として、さらに有用となるよう開発を続けていきたいと考えている。

(2015年8月24日受付)

渡辺大地 (正会員) earth@gamescience.jp

1994年慶應義塾大学環境情報学部卒業。1996年同大学院政策・メディア研究科修士課程修了。1999年より東京工科大学メディア学部講師。CGやゲーム制作に関する研究に従事。

