

# PlayStation によるプログラミング教育について

河野真治

e-mail:kono@ie.u-ryukyu.ac.jp

PRESTO, Japan Science and Technology Corporation  
Information Engineering, University of the Ryukyus  
Nishihara-cyo 1, Okinawa, 903-01, Japan

August 18, 2001

## 1 メタ技術としてのプログラム教育

プログラミングは、数学の問題を解くのと異なり、完璧な答が一番良いとは限らない。限られた時間と資源、あるいは人的資源の中で最良の結果を出すことが要求されている。そこでは、新しい技術を習得することそのものの時間も限られている。

従って、

新しい開発環境への習熟  
マニュアルや WWW からの情報収集  
非常に早い開発サイクルへの慣れ

こと自体を技能として習得する必要がある。特に英語の能力が情報収集では重要である。さらに、実用規模のプログラミングは、多人数で行われるのが普通であり、学生同士でグループを作る場合でも、能力差、知識差が問題となる。

ツール作成  
データ作成  
練習、習熟

などをメンバの状況に応じて割り振る必要がある。また、

技術をライブラリや文書の形で保存し、伝えること

も重要である。これらは、個々の人材の固有の技術ではなく、それらを支える、または、技術を作っていくメタな技術であると言える。

また、ゲームなどは、最終的な製品に近く、完成度が問われる。完成度とは、

ユーザの目に作成されていない部分が見えない  
ゲームの始まりから終わりまでが一通りそろっている  
絵と音が出る  
あらゆる入力に対して対処がなされている

などで判断される。この判断力そのものを身につける必要がある。また、他のグループのアウトプットに対する評価をする能力も重要である。

これらのメタな技術は、教科書を作成し、教えるよりも、実際の作業を通して見つけていくことが効率的だと思われる。また、予算の限られた大学で、教育が陳腐化するのを防ぐ意味でも、このようなメタな技術を教えることが重要である。

## 2 PlayStation 開発キット

PlayStation は、95年に発表された Sony のテレビゲーム機である。

2Mbyte Memory  
37MHz Mips CPU  
Geometry Engine

という、当時 Unix ワークステーション並と言われた構成は、今から見ると、貧弱といって良い。「ねっとやろうぜ」と呼ばれるカスタマ向けゲーム作成環境は、96年に発売された。

パソコン上のクロスコンパイラ  
パソコンと接続可能な特別製 PlayStation  
3D Lightwave 3次元モデリングツール  
WWW サーバによるサポート

という構成は、プロ用の機器とは隔たりがある。しかし、基本構成としてはほとんど同じだといって良い。プロ用の開発機器との差は、ライブラリ内部をいじれないことと、デバッグ環境の差であると考えられる。また、開発キットでは、CDROM を作成することはできず、ダウンロードのみとなる。

コンパイラは gcc であり、Windows だけでなく Unix 上でも開発することが可能である。コンパイラを持っていれば、市販の PlayStation でも拡張バスを使うことにより開発をおこなうこともできる。ただし、最新の PlayStation には拡張バスは存在しない。

## 3 PlayStation によるプログラミング

PlayStation のプログラムは、テレビの走査線の周期にそってダブルバッファを用いて、行われる。ダブルバッファとは、二つの画面を切り替えて、表示しているのと別な画面側に書き込むことにより、スムーズな表示を可能にする手法。描画は GPU が行い、描画コマンドを登録すれば良い。

```
初期化ルーチン
while(1){
    パッド等、入力データ受け取り
    foreach ゲームオブジェクト{
```

```
        ゲーム上のオブジェクトの状態計算
        描画登録処理
    }
    画面切替え操作
    登録描画処理実行開始
}
```

という形をとることが多い。3次元と2次元での本質的な差は、描画登録処理で何を登録するかだけである。

3次元の決まった形は、3D Lightwave でモデルを作り、それを PlayStation に読める形 (TDM 形式) にしてダウンロードする。その形を描画登録処理すれば良い。ただし座標系の計算は3次元なので視点などを含めるとかなり複雑である。線形代数、行列に関する知識は必須である。

このように PlayStation のプログラムは、ハードウェアに密接に結び付いたプログラムであり、限られた時間内に全ての処理を終わらせる必要がある。特に、クロックに依存した形と状態計算からなるので、ハードウェア記述に近い構造を持っている。

## 4 PlayStation によるプログラミング教育

さらに、PlayStation 用に次の4種類のライブラリを作成した。これらは、学生によって作成されたものである。

```
2次元シューティングゲーム専用ライブラリ
PlayStation 用 Tuple 通信ライブラリ
VRML 表示ライブラリ
TMD ダウンローダ
```

これらを用いて、

```
2次元ゲーム
3次元ゲーム
3次元ネットワークゲーム
```

というステップを踏んで1年で、2回プロトタイプを作ると言うのが最初の2年の標準的な課題である。後半2年では、より多くの学生に PlayStation Programming を経験させるために、半年で二つのグループに独立に同様の課題を課した。

## 5 課題の運営

通常、簡単な例題、あるいは、先輩の作った例題を読むことから始め、同時に、3次元モデリングについても学ぶ。だいたい3か月で最初のプロトタイプが完成する。

1グループ、3-6人から構成して、課題の提出やレポートはグループ単位で提出ということにした。これは、プログラミング能力差を吸収するためでもある。一人だけ、あるいは、二人とすると、完成度の高いプログラムを作るのには人数が足りず、また、プログラミング能力の低い学生が集まった場合に、課題が破綻してしまう。

全体的な完成度は、1年通して実験したグループの方が当然高い。例えば、オープニングや音などをつけることにかなりの時間がかかってしまう。

しかし、一番重要なのは「どんなゲームにするか」であり、これをまじめに行うためには、数か月かかるようである。従って、独創的なゲーム作成をめざすならば、一年かける方が良い。半年では、かろうじて動くゲームを作成するのが限界である。

また、プログラミングコンテストに、2年続けて参加し、銅賞一つを獲得した。

## 6 学生のプログラミングの特徴

銅賞を獲得したゲームは、技術的にはそれほどでもないが、オリジナリティのあるゲームであった。これを作成したグループは、扇子なげや中国ごまなどを普段から練習するなど、テレビゲームではない遊びにも凝ったグループであり、プログラミングが技術だけではないことを示している。また、3次元ゲームを考えるのに、紙で作った模型を作成していた。ゲームの内容を考える時には、有効な手法だと思われる。

ゲームを作る技術の低いグループは、ゲームの面白さをシナリオの導入で補うことを考えることが多い。しかし、技術の高いグループが作ったゲームが面白いとは限らない。「すごい」と人をおぼせることができるが、それと面白いゲームとは独立なもののようなものである。

最初に簡単な例題、あるいは、完成したプログラムを渡してしまうので、どうしても、ほとんどのプログラムは、人のをそのまま移したものとなっている。

ライブラリを作ったり、それを人に使わせたりするのは、かなり高度なことであり、ごく一部のグループでしかおこなわれなかった。

特に指導しないと、PlayStation では巨大な main 文一つでサブルーチンがほとんどないプログラムを書くことが多い。これは一つは、PlayStation のプログラム構造が巨大な while 文になっているので、それに引きずられているのだと思われる。

また、例えば、ミサイル5つの処理などを、「一つのミサイルの処理をカットアンドペーストして、5つ並べる」というスタイルが良く見られる。これが、全体の構成をみずらくしてしまう。

また、Cのポインタと構造体は、Cの初心者のおぼえであり、配列のみを使って、構造体を避けていることが多い。例えば、オブジェクトの位置を、

```
int object_x[5];
int object_y[5];
```

のような二つの配列で持ってしまう。これを、

```
struct object { int x; int y; } object_list [5];
```

のように書くことができない。従って、malloc() がほとんど出て来ない。

一方で、malloc() のちゃんとした初期化は非常に難しいらしく、malloc() の使用を強制すると、プログラムミスが増え、デバッグが大変になる。だとすれば、配列を使うこと自体は、プログラミング能力に見あった妥協だとも言えるかも知れない。

一部のグループは、Playstation で動かす前に、Windows 上でプロトタイプを作成している。VC++ + Direct-x を用いることが多い。PlayStation 上でも C++ を用いたグループが一つあった。ただし、C++ を効果的に使っているコードではなかった。

PlayStation 開発キット「ねっとやろうぜ」はドキュメントのバグが多く、それが大きな障害となった。さまざまに、試しながらプログラムしていくようなことが必要になっていた。

Tuple 通信ライブラリを、理解することができず、通信を使った同期を行う方法もわからないので、ネットワークゲームを作ることができない。一つは、ドキュメントが不備だからかも知れない。一つのグループなどは、自分で通信ライブラリを別に自作していた。