

アーケードゲームのテクノロジー（その1）

宮沢 篤† 駒野目 裕久‡

†日本アイ・ビー・エム株式会社 東京基礎研究所

‡池上通信機株式会社 R&Dセンター

ピンポンをもとにした、商業的に成功した初めての業務用ビデオゲーム「ポン」(Pong, 1972年)が、米アタリゲームズ社を設立したノラン・ブッシュネルによって発明されてから、既に20年以上の歳月が流れている。当時のゲームは、TTL (Transistor Transistor Logic) を中心とした汎用ロジックICを組み合わせて設計されており、技術的に見ても未発達で、最も単純な対話型(インタラクティブ)コンピュータグラフィックスの一応用分野でしかなかった。それから現在までに、世界中のさまざまなゲーム会社から、その時代の最も進んだコンピュータ技術を取り入れた、非常にたくさんのビデオゲームが発表されてきた。ゲームデザイナーの優れた感性と、コンピュータエンジニアの高度な技術の両方が融合することによって、今日のビデオゲームは、全く新しいインタラクティブなメディアにまで進化している、と言えるかもしれない。そしてこの新しいメディアが、伝統的なメディア産業に与えている影響も無視できない。例えばビデオゲーム関連の出版物は、攻略本や情報誌から始まって、ゲームの人気キャラクターを主人公にした小説やコミック、アニメまでありとあらゆるジャンルに及んでいる。今まで単なる暇つぶし、子供向けのオモチャや遊びの感覚として扱われていたビデオゲームが、日本を代表する一つの文化として語られる時代になってきたのである。

ビデオゲームを学問的に分析しようとする試みは、アタリ社の家庭用ゲーム機「アタリ2600、通称VCS (Video Computer System, 1977年)」がアメリカで爆発的な人気を呼んだ70年代後半から、主に心理学や教育学、メディア論などの歴史社会的な領域で進められてきたが、コンピュータを含む電子的な映像表示技術の応用という立場からの調査研究は、これまでほとんど行われたことがなかった。コンピュータグラフィックス草創期の技術を、広い範囲にわたって解説した書物として有名な、ウィリアム・ニューマンとロバート・スプロールによる“PRINCIPLES OF INTERACTIVE COMPUTER GRAPHICS, 2/e” (McGraw-Hill, 1979年)は、上記のビデオゲーム「ポン」のルール説明と、「ビデオゲームは、家庭で最初に使われた対話型グラフィックスの代表である。」という印象的なメッセージで始まっている。残念なことに、技術的に未発達だった当時のビデオゲームにそれ以上の出番は与えられなかったが、このところ人気を集めている、いわゆる次世代ゲーム機の特徴となった3次元グラフィックス機能は、多角形(ポリゴン)の走査変換(スキャンコンバージョン)、幾何計算、隠面消去アルゴリズム、グーロー

(Gouraud) シェーディング、そしてテキストチャマッピングに及んでいる。今日のビデオゲームは、コンピュータグラフィックスの最新技術をふんだんに使いながら、なおかつ究極的なインタラクティブ映像への発展途上にある、と言えるかもしれない。この小文では、70年代後半から現在に至るまでのゲームマシンのハードウェアを中心に、ビデオゲームを構成するいくつかの基本的な映像表示技術について（一部推測を交えながら）解説する。

ゲームマシンのアーキテクチャ

ファミコンやメガドライブを代表とする家庭用ゲーム専用機（コンシューママシン）から、業務用ゲーム機（アーケードマシン）に至るまで、ゲームマシンのハードウェアには、これが標準といった規格はなく、各社の各機種ごとにその仕様はバラバラである。しかし、強力なグラフィックス機能や高品位なサウンドは、高度なゲームを作り上げるための必要条件であり、これらのマシンに共通する特徴となっている。多くの場合、ゲームの実行や画面処理（グラフィックス）、サウンドの発生などは、それだけを専門に担当するCPU（8080、Z-80、6809、68000などに始まって、最近ではPowerPC 602やR3000といった32ビット以上のRISCチップが使われる）によって行なわれている。この意味で、ほとんどのゲームマシンは、複数のCPUを搭載したMIMD（Multiple Instruction Multiple Datastream）型の並列計算機、という見方ができる。

もっとも、ビデオゲームの黎明期における作品、例えば上記の「ボン」や、「ブロックくずし」という呼び名のもと大ブームとなった「ブレイクアウト」（Breakout, 1972年/アタリゲームズ）などは、TTL-ICを組み合わせて設計された大掛かりなもので、当時の電卓技術を流用したと言われている。ビデオゲームに汎用CPUが初めて使われたのが、「風船割り」の通称で有名な「サーカス」（Circus, 1977年/エキシディ）においてであり、以降のゲームは例外なくマイクロプロセッサを用いている。

画面処理

ビデオゲームにとって必要なのは、高速で動く、たくさん色鮮やかなキャラクタである。グラフィックスの表示速度が遅ければ、いくら緻密で美しい画面を出したとしても、もはやゲームとしては成り立たない。IBM-PCやMacintoshのような、一般的なパソコンで主流を占めているビットマップ方式のグラフィックスは、速度上の問題から、現在では全く使われていない。過去の例としては、有名なタイトー社の「スペース・インベーダー」（Space Invaders, 1978年）[4]やナムコ社の「リブルラブル」（Libble Rabble, 1984年）などがある。（なお、前出の“PRINCIPLES OF INTERACTIVE COMPUTER GRAPHICS, 2/e”では、「ボン」がフレームバッファを用いてプログラムされているように説明してあるが、このインプリメンテーションはアタリVCS版のものだと思われる。）ゲームマシンの画面構成をひとことで言えば、PCG（Programmable Character Generator）

のような疑似グラフィック画面（通常、2面から7面くらいあって、重ね合わせて表示される）と、それらを独立して上下左右に1ドット単位で動かせるスクロール機能である。これに、後述するスプライトの機能が加わって、あのスピード感あふれる映像が作り出される。〔5, 6〕

スプライトは、オブジェクトとも呼ばれる、ある大きさをもった特殊なキャラクタパターンのことである。通常のグラフィック画面で、キャラクタを移動させようとする、前のパターンを消してから、移動した後のパターンを新しい位置に表示しなければならないため、高速で物体を動かすのはほとんど不可能である。このスプライトは、座標を変更することによって、表示画面内を自由に移動することができ、ソフトウェアによるパターンの再定義は一切必要としない。また、スプライトには優先順位があり、2つ以上のスプライトが重なった場合、優先順位の低いスプライトが自動的に消えるようになっていて、手前にある物体が後方の物体の前を通り過ぎるように見え、画面に奥行き感を持たせることができる。さらに、スプライトどうしの1画素が一致すると、プロセッサ内部の衝突フラグがセットされるなど、ビデオゲームの製作には欠かせない機能の一つである。スプライトは、もともとMSXパソコンなどに採用された、テキサスインスツルメンツ社のグラフィック用LSI（TMS-9918-A）にユニークな機能だったが、現在のゲームマシンには、性能面でそれをはるかに上回る、カスタムのディスプレイプロセッサが搭載されている。

例として、「ドンキーコング」（Donkey Kong, 1981年/任天堂）に使用したハードウェアのスプライト制御の仕様を示す〔5〕。

- ・スプライト表示アドレス 256×256
- （表示画面有効区間 224×256）
- ・スプライトのサイズ 16×16ドット
- ・パターン数 128個
- ・一走査線スプライト表示最大数 16個
- ・一画面スプライト表示最大数 96個
- ・左右、上下反転機能
- ・カラー 3色+黒（カラーコードにより、16種）

ハードウェアとして、一走査線分のラインバッファを持っているだけで、表示画面のドットに対応した2次元の画面メモリを持っているわけではない。CPU側からは、各スプライトに対し、 x , y アドレス、パターン番号、カラーコード等の表示データをハードウェアに渡す（スプライト表示データバッファ）。通常、この処理は、表示画像の乱れを防ぐために垂直ブランキング期間（約1.2ms）に行われる。

スプライト表示データバッファに書き込まれたデータは、水平ブランキングのタイミングごとに、表示判定アドレス演算部によって、走査線表示アドレスを示すTV水平信号カウンタ値 Xa をもとに、スプライトの x アドレスから、表示ライン Xa 上に存在する

スプライトの判定を行う。表示ライン X_a 上にスプライトが存在すれば、パターン番号より該当パターンをパターンROMから読みだし、表示ライン上で表示するデータを y アドレスを参照してラインバッファにパターンデータをセットする。この処理を、すべてのスプライト表示データに対し、水平ブランキング期間 ($11.4 \mu s$) に行う。水平ブランキング期間では、処理時間が短いため表示用と処理用を並列にできるように、ラインバッファを2つ設けて、水平期間 ($63.55 \mu s$) で行っているものもある。最終的にできあがった、ラインバッファの内容は、ビデオ表示装置によりディスプレイに表示される。この一連の処理を走査線ごとに行う事により、画面上に2次元のスプライトの表示が行われる。

「ドンキーコング」では、最初にすべてのスプライトに対し、表示ライン上に存在するかのチェックを行い、該当するスプライトにマークを付け、その後、マークの付いたスプライトデータのみを処理している。存在のチェックでは、96個のスプライトデータから、一走査線上で表示できる最大の16個のスプライトの抽出処理を行っている。

特殊効果

画面処理に触れたついでに、背景やキャラクタの拡大・縮小や回転機能など、ビデオゲームの表現力を高めるための特殊効果についても、見逃すわけにはいかない。これらの機能は、数学で言うところの座標幾何の応用であるが、定義通りまともに座標計算をしていたのでは、とても限られた時間内に絵を作り終えることができない。そこで、高速化のためのさまざまな工夫が必要になってくる。まず拡大・縮小については、キャラクタパターン内の1画素を、規則的に繰り返したり間引いたりしながら表示することでとりあえず実現できる。この処理が、実は直線発生アルゴリズムを用いて高速に実現できることについて、以下に詳しく説明する。

直線発生器は、もともとコンピュータグラフィックスの分野において、直交座標系の格子点上に誤差最小に離散化された直線を、微分解析器の原理を用いて高速に発生させるために使われるものである。加減算とシフトのみを含む直線発生アルゴリズムは、傾きが n/m (ただし $n < m$) の直線に対して、

```
static int y, e;

initLine(int m, int n)
{
    y = 0;
    e = (n << 1) - m;
}
```

```

genLine()
{
    if (e >= 0) {
        y++;
        e += (n - m) << 1;
    }
    else
        e += n << 1;
}

```

で与えられる。ここで、`initLine(m, n)`によって初期化された直線発生器は、それ以降 `genLine()`が呼び出されるたびに、直線の座標値を順次発生していくものとする。このとき得られる傾き n/m の直線を、 n 個の画素列と m 個の画素列間の対応付けのための規則ともみることができる。これは、デジタル画像に対して倍率 n/m の拡大・縮小をすることと等価である。アルゴリズムは加減算とシフト操作の範囲内で行えるために、ハードウェア化も容易である。ひとたび、画像の拡大・縮小が実時間で実行できるようになると、これを使ってゲーム画面に疑似的な 3 次元効果を出すことが可能になり、その結果、例えば「ポール・ポジション」(Pole Position, 1982 年/ナムコ) や、セガ社の大型可動筐体による「体感ゲーム・シリーズ」として、「スペース・ハリヤー」(Space Harrier, 1985 年) や「ハングオン」(Hang On, 1985 年)、「アウトラン」(Out Run, 1986 年) など、数々の名作と呼ばれた疑似 3 次元ゲームが輩出することになる。

回転については、2 次の変換行列をただだか 4 つの単純な行列の積に分解し、それぞれを整数演算の範囲内で、高速に計算するようなアルゴリズムがいくつか知られている。しかし、現実のゲームマシンでは、各パターンを実行時にリアルタイムで回すのではなく、あらかじめ回転しておいたパターンを適当に選びながら、それらを決められた位置に表示することで、あたかも回転しているかのように見せているものがほとんどである。戦闘機のドッグファイトをシミュレートした「アフターバーナー II」(After Burner II, 1987 年/セガ・エンタープライゼス) では、敵の攻撃を避けながら、空中旋回のグラフィックで 360 度自機がぐるっと回る。これは、ゲーム画面のほとんどすべてをスプライトだけで構成し、スプライトの表示位置を調整することによって、背景そのものが回っているような臨場感を出すことに成功している。同様の技法を用いたのが、下に落ちるボールを画面全体を回転させてゴールへ運ぶ画期的なアクションゲーム「キャメルトライ」(Cameltry, 1989 年/タイトー) である。このゲームでは、迷路を構成する一つ一つのブロックのすべての回転パターンをスプライトとして用意することで、回転機能を持たないシステムへのインプリメンテーションを可能にしている。

また、数学では射影変換 (Projective Transformation) と呼ばれる、長方形の背景パ

ターンを台形に変形することによって、奥行き感を持った画面を作る手法が、任天堂のスーパーファミコン用ソフト、「パイロットウイングス」(Pilotwings, 1990年)や「スーパーマリオカート」(Super Mario Kart, 1992年)で使われている。これは一見すると、長方形パターンの上部を縮小、下部を拡大していくことによって作れそうな気がするが、射影変換自体が非線型なため、実際にはうまくいかない。この変換にも、やはり整数演算の範囲内で行なえるアルゴリズムが存在する[7]。ちなみに、上記のソフトでは、ゲームカセットの内部にDSP(Digital Signal Processor、NEC社製のD77P25C)を搭載することによって、処理の高速化を図っている。

なお、ビデオゲームを支える技術に関しては、この他にも立体視やポリゴナイザー、画像処理やサウンドなどの興味深い話題もあるが、また別の機会とさせていただきます。

参考文献

1. テレビゲーム・ミュージアム・プロジェクト編：テレビゲーム／電機遊戯大全／TV-GAMES, UPU (1988年)。
2. 月刊ゲームスト増刊：ザ・ベストゲーム, 新声社 (1991年)。
3. コンピュータ局開発：新世代ゲームビジネス, 日経BP社 (1995年)。
4. テレビジョン学会編：応用マイクロコンピュータ, 昭晃堂 (1980年)。
5. 社内資料：ドンキーコング プログラム仕様書, 池上通信機株式会社 (1981年)。
6. ラジオライブ別冊：ゲームマシン大研究, 三才ブックス (1989年)。
7. A. Miyazawa: SYSTEM FOR SOURCE IMAGE TRANSFORMATION, U. S. Patent 5, 077, 681 (Dec. 31, 1991)。

本 PDF ファイルは 1996 年発行の「第 37 回プログラミング・シンポジウム報告書」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトに、下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載し、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

https://www.ipsj.or.jp/topics/Past_reports.html

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者（論文を執筆された故人の相続人）を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間： 2020 年 12 月 18 日 ~ 2021 年 3 月 19 日

掲載日： 2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>