

デジタルゲームの設計手法

Design Method for Digital Games

川合 康 央*
Yasuo KAWAI

1. はじめに

本稿では、デジタルゲームの構成要素とそのデザイン手法について論じる。ここでは、ゲームを遊ぶユーザーの視点ではなく、ゲームを制作するデザイナーの視点から考察する。

1.1. ゲームの定義

「ゲーム」という語の持つ意味は、遊戯や競技等を指し示すものであるが、その範囲は明確ではない。将棋やチェス等のボードゲームからトランプ等のカードゲーム、野球やサッカー等のスポーツやかくれんぼ等の遊戯等が含まれる幅広い概念である。しかしそこには、緩やかな類似点が存在している。ゲームとは、ルールと競争があり、目標に対して意思決定を行い、不確定な未来に対して活動を行うものである。また、それらは生産活動を含まない行為であるとも言える。

これらのうち、プログラミング言語を用いてデジタルでコンテンツを制御するものを、本稿で扱う「ゲーム（デジタルゲーム）」とする。このデジタルゲームの分類もまた、明確なものではない。文献や資料により用語等の定義や解釈が異なり、また、各地域において独自の市場を展開していることから国際的なルールもない。さらに、コンテンツによっては分類の範疇に収まらないものも少なくなく、技術進化のスピードとともに、これらを完全に定義することはできない。本稿では、CESA¹⁾によるゲームの定義をもとにし、以下のように再構成し分類する（図1）。

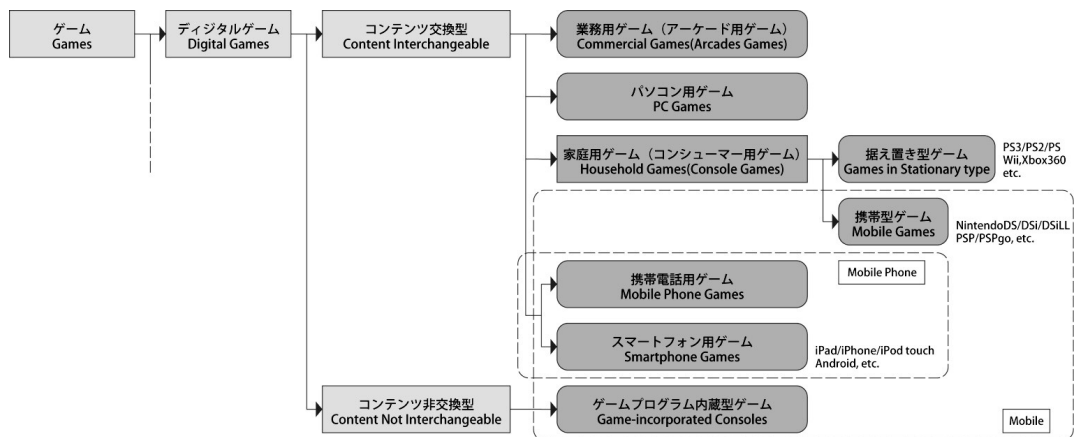


図1：ゲームの分類

* 文教大学情報学部准教授

1.2. デジタルゲームの変遷

コンピュータによって処理を行うデジタルゲームは、そのゲームプログラムであるコンテンツについて、交換型と非交換型に大きく分類される。コンテンツ非交換型は、ゲームプログラムが予めハードウェアに組み込まれているものである。家庭用ゲーム機としてこのコンテンツ非交換型は、黎明期の家庭用据え置き型ゲームや、「ゲーム&ウオッチ（任天堂、1980）」等の携帯型液晶ゲームとして広く普及し、1980年代には様々な電子LSIゲームが市民権を得た。その後、「たまごっち（バンダイ、1996）」等の携帯型コンテンツとして、他の機能と融合しつつ独自の進化を遂げることとなる。

一方、テレビモニターに接続する方式の家庭用据え置き型ゲーム機も、デジタルゲーム史の最初期にはコンテンツ非交換型のものが主流であった。アーケードゲームを家庭で楽しむことを目的としたこれらのゲーム機は急速に普及していったが、コンテンツごとの専用筐体は高価であり、またゲーム内容も価格に見合わないといった問題を抱えていた。その後、ROMカートリッジ交換型の「Atari VCS（Atari, 1977）」、「カセットビジョン（エポック社,1981）」等が市場に流通し始め、「ファミリーコンピュータ（任天堂,1983）」以降、ゲームプログラミングを外部メディアに保存するコンテンツ交換型が主流となる。カートリッジやフロッピーディスク等の外部メディアによるゲームプログラミングの流通が始まり、家庭用ゲーム機はハードウェアとソフトウェアに分かれ発展を遂げていくこととなる。

PCの普及とともに、PCゲームも独自の発展を遂げてきた。据え置き型ゲーム機とともに1980年代から急速に普及し始めたPC市場でもまた多くのゲームプログラミングが流通していた。1990年代にPC端末がインターネットへと接続するとともに、様々な形のコンテンツがネットワークを通じて発展していくこととなる。特に近年の海外市場では、ゲーム専用機ハードウェアよりインターネットに接続されたPCの方が普及している地域も多く、また開発環境の整備も安価なことから、ゲーム開発のプラットフォームはPCが主流になりつつある。また、インターネットに接続されたPC環境において、ゲームは必然的にネットワーク化され、MMO（Massive Multiplayer Online）型等のネットワークゲームの市場が急激に拡大した。これらのゲームプログラミングは、ネットワークからダウンロードする形での流通が増えてきており、課金方法も従来のパッケージ型から課金型へと移行しつつある。

携帯型ゲーム機は、主に家庭用据え置き型ゲーム機の機能を、屋外へ持ち運べるようにしたハードウェアである。携帯型ゲーム機向けのカートリッジを据え置き型ゲーム機向けとは別途開発し、軽量で手軽なハードウェアとして広く普及した。ゲーム専用のハードウェアであることから、携帯電話と比して描画性能もよく、コンテンツのデータ容量も大きいものとなっている。

一方、携帯電話は携帯型ゲーム機とは異なる種類のコンテンツが普及した。特に通信端末である携帯電話の特徴を活かしたネットワークを用いたゲームが挙げられる。しかし、ハードウェア性能や通信速度等の問題もあり、データ容量の小さいダウンロード型ゲームや、限定的な規模のネットワーク型ゲームが主流である。またスマートフォンは、携帯電話の通信機能とともに、携帯型ゲーム機と比して遜色のない描画性能や処理速度を併せ持つものである。これらはPC同様に世界共通のプラットフォームを持つネットワーク型のハードウェアであることから、今後さらに従来のゲームの枠組みに収まらない様々なコンテンツが開発されると予測される。

デジタル化されたゲームは、業務用アーケードゲームから家庭用コンシューマーゲームへ、据え置き型ゲーム機から携帯型ゲーム機へ、ゲーム専用機から携帯電話やスマートフォンへと市場を拡大してきた。そこでは単なる各プラットフォーム間におけるコンテンツの移植ではなく、夫々のハードウェア特性を活かしつつ役割を分化させながら発展してきたものである。また、開発環境もオープン

になりつつあり、ネットワークを通じてコンテンツを配信可能なことから、今後も多くの新しいデジタルコンテンツが現れると考えられる。

2. ゲームのデザイン

デジタルコンテンツを作成する際、どのような技術で制作するかとともに、どのような手法で表現するかが重要なものとなってくる。ゲームに限らず、創作物はいかに新しいアイデアを多くの人に共感してもらうかが鍵である。しかしその新しいアイデアが、これまでのコンテンツの約束事から隔絶した余りにも異質なルールであった場合、多くのユーザーに受け入れてもらうことは難しい。

建築設計の場合、空間をデザインするためのスキルを修得するために、既存の優れた建築を詳細に調査する。多くの建築を観るだけでなく、実物に対して測量やスケッチを行い、図面を模写し詳細な模型を作成することで、空間における見えないルールや共通する要素とその関係性を理解する。同様に、ゲームデザインにおいても、多くのゲームをユーザーとして体験するのではなく、既往のゲームについて詳細に調査し、正確にトレースすることで、コンテンツに共通のルールと要素の構造を理解し、ゲームデザイナーの思考を追体験することが重要である。

2.1. ゲームのルール

ゲームを作るには、まずそのゲームにおけるルールをデザインしなければならない。ルールのバランスが悪いと、他に優れた要素が多くあったとしても、そのコンテンツは価値のないものと看做される。しかし、歴史と経験から改良されてきたスポーツやカードゲーム等の普及しているゲームルールと、比類するレベルで緻密かつ魅力的なものを、全くの白紙から作り出すことは容易ではない。

一方で、既存のスポーツやカードゲーム等をデジタルでシミュレートしたゲームタイトルも少なくない。既に普及しているルールを用いることで、複雑なルールを白紙から定義せずとも、魅力的なコンテンツを作成することが可能である。最初期のコンピューターゲームである「Tennis for Two (William Higinbotham, Brookhaven National Laboratory, 1958)」やアーケードゲームの「PONG (Nolan Bushnell, Atari, 1972)」も、テニスや卓球をもとにして作られたゲームである。

既存のデジタルゲーム内においても、多く使われ普及しているルールがある。デジタルゲームの構造を読み解き、そのルールを用いることによって、新しい約束事を全て作成し、多大の説明を用いて普及させていくことなく、ユーザーに自然な操作をさせるコンテンツの制作が可能である。

2.2. ユーザビリティのルール

これまで、デジタルゲームはその短い歴史の中において、他のメディアや現実のメタファから援用した様々なルールを取り入れてきた。また、デジタルゲームにおいて、使いやすい操作方法や表示方法は、改良が繰り返された上で広く普及し、様々なゲームにおけるルールの基盤となった。

最初期のアーケードゲームである「PONG」や「スペースインベーダー (タイトー, 1978)」、「ギャラクシアン (ナムコ, 1979)」等では、自機 (プレイヤーキャラクター) の操作は1次元的な動きであった。これをもとにして「ゼビウス (ナムコ, 1983)」等のような2次元的なXY軸の操作が可能な手法へと発展した。これらは、コントローラー等の物理的操作デバイスを用いて、自機をXY平面上に移動させる操作を行うことが可能であり、コントローラーを倒した方向へそのまま画面上を自機が移動するものである。一方、「バンゲリングベイ (Will Wright, Brøderbund, 1984)」では、コントローラーの上下で前進と後退、左右で旋回を行うものであった (図2)。これはラジコンの操作方法をもとにしたものであり、全方位スクロールシューティングゲームの操作方法として妥当であるように思われた。しかし、上方からの視点における2次元空間上での操作方法としては、発表当時のユーザー

にとって他のゲームの操作方法と比して直感的ではなく、その後広く普及することにはなかった。現実のメタファを取り込む手法は、ゲームに限らず多く存在するが、コンテンツの内容によってその操作手法は慎重に選択されるべきである。この「バンゲリングベイ」のような操作手法に近い手法は、のちに3次元空間を移動する操作方法として異なる形で普及することとなった。リアルタイム3Dによってキャラクターがステージ上を自由に動けるコンテンツでは、キャラクターを背面から表示し、前進と左右の方向転換により操作する。現在のユーザーにとってこの「バンゲリングベイ」のような操作手法は、2次元空間上のコンテンツであっても発表当時より抵抗が少ない可能性がある。また、高さ方向を含む3次元空間上を自由に移動する際の操作手法としては、2軸を操作する十字型コントローラーだけでは扱いきれず、ボタンとの併用なども見られ、ゲームルールとともに入力デバイスを含めて、現在も様々な手法の試行と改良がなされている。

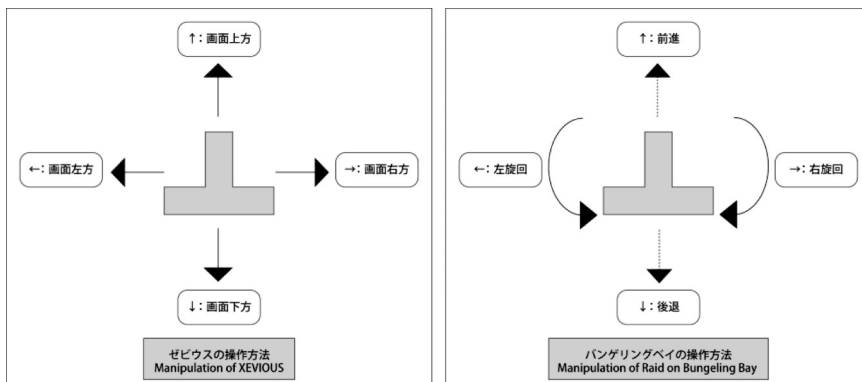


図2：操作方法の違い

2.3. インタフェースデザインのルール

表示系のルールとして、シューティングゲームにおける背景のスクロール方向が挙げられる(図3、図4)。シューティングゲームでは、キャラクターとは独立したレイヤーを背景として描画し、これを進行方向とは逆にスクロールさせることによって、大きなステージを移動させているように見せている。最初期の「ギャラクシアン」において、星が画面下方へ移動することでプレイヤーキャラクターが前進していることを表現するようになった。その後の「ゼビウス」では、地上を表す背景画像をスクロール表示するとともに、敵キャラクターを空中だけでなく地上にも配置することで高さの概念をゲームに取り入れている。これらの縦スクロール型シューティングゲームは、自機が下方から上方へと向かって進んでいることを表すため、背景画像を上方から下方へとスクロールさせている。

また、「グラディウス(コナミ, 1985)」や「ダライアス(タイトー, 1986)」等では、背景が横方向にスクロールする。これら横スクロール型シューティングゲームでは、プレイヤーキャラクターが右方へと向かっていることを表すため、背景を右方から左方へとスクロールさせている。この横スクロール型は、モニターが通常横向きであることから、家庭用コンシューマーゲームやその移植を念頭に置いたコンテンツでは有利であったことから、数多く開発された。また「魔界村(カプコン, 1985)」等のアクションゲームにおいても同様の横スクロール処理を行っている。

縦スクロール型や横スクロール型等の2次元的なステージを持つものに対して、「スペースハリア

ー（セガ, 1985）」、「アフターバーナー（同, 1987）」等、疑似的な3Dを表現した奥スクロール型シューティングゲームも、デジタルゲーム史において比較的早い時期から存在していた。これらは、自機を背面から表示し、敵キャラクターや背景等の要素を徐々に拡大することによって奥行きを表現するものである。奥スクロール型は迫力のあるシーンを表現することが可能であり、これらはリアルタイム3Dコンテンツへと展開していくこととなる。

これらのスクロールの背景の移動方向であるが、縦スクロール型は上から下へ、横スクロール型は右から左へ、奥スクロール型は奥から手前へと、スクロールする方向は多くの場合暗黙の裡に決定されている。横スクロール型で背景画像が左から右へとスクロールする「スカイキッド（ナムコ, 1985）」や、斜め上方からの平行投影表示で3Dを表現する「ザクソン（セガ, 1982）」等のコンテンツはいくつか存在したが主流とはならなかった。

Webページのスクロールにおいて、現在では縦スクロール型が主流となっている。一般的に表示モニターは水平方向が垂直方向に比して長いこともあり、横に情報を配したほうが有利であるとも考えられなくはない。情報を垂直方向の表示範囲で提示したものを水平方向に連続的に配置し、横スクロールでページを遷移させる方式のような横スクロール型サイトデザインは考えてみると使いやすそうではあるが、実際に運用されているサイトはほとんど存在しない。縦スクロール型が主要である要因として、見易さや使い勝手が挙げられるが、同時にユーザーが使い慣れている表示方法であるという理由も大きいものと考えられる。

同様に、デジタルゲームにおけるスクロールの方向も、わかりやすさとともに慣れ親しんだものが好まれる傾向にあると思われる。使いやすいデザインを考える際、これらのような既存の見えないルールに従うことによって、ユーザーに混乱させないインタフェースデザインが可能である。一方で、新しい発想に基づくデザインは、使いやすいデザインであるとともに、それらが受け入れられるまでに経験に基づく、慣れと親しみが必要である。また、全く新しいゲームのルールを独自に作成する際には、ゲームはシミュレーションであるという観点から、日常にある活動や事象をよく観察し、これをわかりやすい形に情報を整理し抽象化するという手法を用いることが多い。

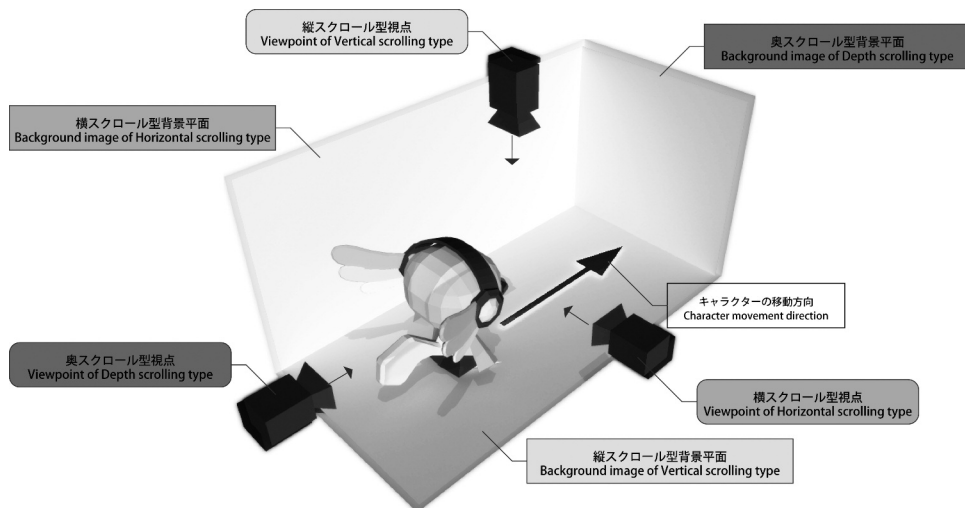


図3：スクロール型の分類

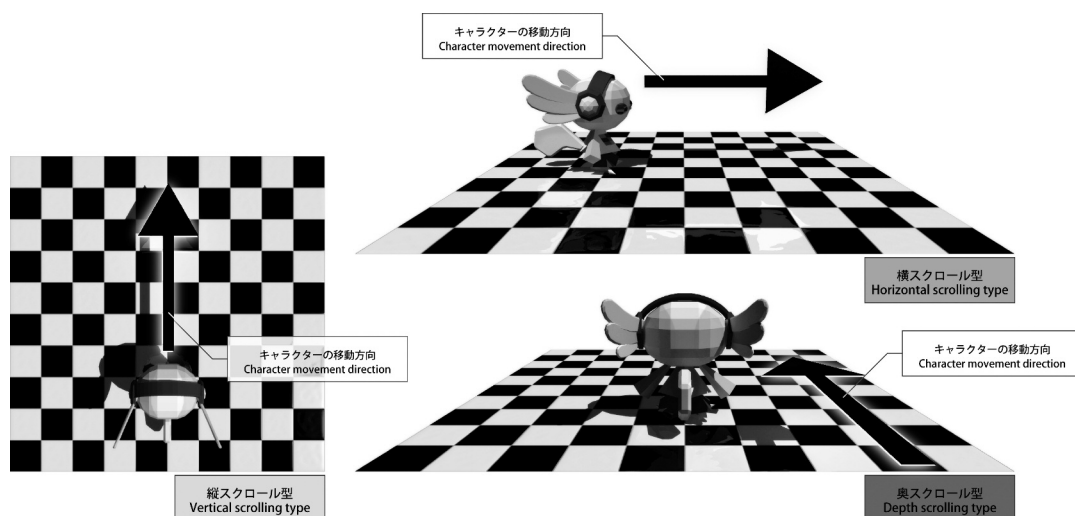


図4：スクロール型による視点からのスクリーン表示

2.4. デジタルトレース

既存のスポーツやカードゲームをシミュレートしてデジタルゲームを作成した時、ユーザーは本来のゲーム自体のルールは知っているが、デジタルゲームのルールに迷うことも少なくない。レースゲームにおいて普段の運転方法と異なる入力デバイスに戸惑い、カードゲームに至ってはカードの切り方すら迷うこともある。

インタフェースデザインを考える際、既存の多くのコンテンツを調査研究することは有効である。デジタルゲームにおける共通のルールを把握することで、デザイナーのユーザーへの配慮と工夫を理解することができる。そこでは、単にユーザーとしてゲームをプレイするのではなく、コンテンツの構造を要素ごとに分類し、その意味を読み解くことが必要である。

デザインを深く読み解くには、トレース（模写）することが有効である。模写は、優れた作品の技法や作風を忠実に写し取ることによって、技術を修得するとともに、作品の読み解きとして古くから美術等の世界で行われてきた手法である。単なるコピーではなく、作品の背景にある作家の思考を追体験することで、多くのことを学ぶものである。

デジタルゲームのトレースでは、キャプチャーした動作画面を要素ごとに分解し、レイヤーを分けて忠実にPixel単位で再現する。如何なる要素がどのように配置されているのか、素早い判断が必要な場面でいくつかの情報をどのような形で表示させているのか等、デザイナーの思考を追体験することから理解できる事柄は多い。同様に、動きのある要素に関しても、Flash等のオーサリングツールを用いてトレースすることが可能である。要素ごとに画像をシンボル化し、アニメーション処理やプログラムであるActionScriptを記述することで、限定的に再現することができる。実際に同じような動きを再現させようとすると、敵キャラクターの発生パターンや行動パターン、自機キャラクターの入力に対する反応の遅延等、気付く点が多い。

いくつかの類似コンテンツを正確に再現することで、コンテンツ間の類似する要素とルールを整理する。それらを踏まえたゲームデザインを行うことによって、これまでのデジタルゲームのデザイン遺伝子を持った、ユーザーに分かりやすいコンテンツの作成が可能となる。

3. ゲームの制作手法

ゲーム開発は、企業が商品として大規模なソフトウェア開発を行うものから、個人が趣味の範囲で制作するものまで、様々な規模のものがある。複数人のプロジェクトで数か月を費やして開発するものもあれば、Webアプリケーション等で個人がプログラミング知識を一切用いずとも容易に作成できるものまで、様々なゲームを作成する環境が用意されている。制作する際には、コンテンツの企画に合わせてそれに応じたツールを選択する。環境によっては限定的な機能を持つものもあり、ツールによってコンテンツの幅が決定されないよう注意する必要がある。ここでは主に、一般的なPCや携帯電話等のWebブラウザで動作するFlashでのゲーム開発について記述する。

3.1. キャラクターデザイン

キャラクターは、そのデザインのベースとなるものが人間であっても機械等の無機物であっても、複数の構成要素の集合で成立している。人間であれば、頭と体、腕、脚と分解可能であり、これをさらに目や口等に詳細に分割できる。下書き等をもとに全体を大まかに描いた後、早い段階でこれらのパーツに切り分け、夫々シンボル（グループ）化しておくことで作業が容易である（図5）。また、目や口等の要素を入れ替えることで表情のバリエーションが作成可能であり、髪形や目等のパーツパターンや色を組み合わせることで複数のキャラクターを共通のパーツを用いて作成できる。さらに腕や脚などにアニメーションを施すことで、様々なポーズや動きを取らせることも可能である。自分なりの基本となる素体を作成しておくことで、様々な場面で重宝する。

3.2. アニメーション

デジタルでアニメーションを作成する際の利点として、キーフレーム法が挙げられる。キーとなるフレームを設定すれば、キーフレーム間のアニメーションをコンピュータが自動的に補完するものである。Flashによるキーフレームアニメーションでは、オブジェクトの位置や大きさ、回転、色等の要素を変更することができる。キーフレーム間は同一のシンボル要素で構成されていなくてはならない等のルールがある。

アニメーションを作成する際は、動きの表現についても考慮が必要である。無思慮にキーフレームを用いた平面的な動きではなく、オブジェクトの押しつぶし等の誇張表現や、次の動作に対する予備的動作等、アニメーション特有の動きの表現がある。特に細かい動きが必要な場面では、フレームごとに動きを設定し、メリハリをつけるとよい。

Flashでアニメーションを作成する際は、ムービークリップシンボルを用いる。このシンボルは、アニメーションを付したシンボルを入れ子にすることが可能であり、より複雑なアニメーションを作成することができる。また、このムービークリップへActionScriptを記述することも可能である。

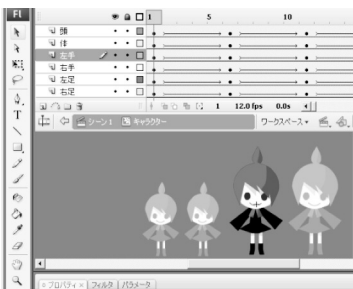


図5：キャラクターデザインとモーショントウィーンアニメーション



図6：ボタンシンボルを用いたインタラクションデザイン (1)

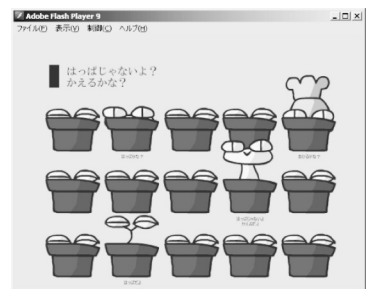


図7：ボタンシンボルを用いたインタラクションデザイン (2)

3.3. インタラクシオンデザイン

ユーザーからの入力を受けるものとして、Flashではボタンシンボルが挙げられる。これは、自シンボルに対するポインタの位置を確認するもので、予め設定した判定範囲内にポインタが発生した場合、それに応じたフレームの切り替えを行うものである。Web上のボタンデザイン等に多く用いられている他、当たり判定のあるゲームの自機や、デジタル絵本などにおけるアニメーション再生のキーにキャラクターシンボルを配することができる。このボタンシンボルに対しても、ムービークリップシンボルを入れ子にすることは可能であるが、ボタンシンボルを二重に入れ子にすることは判定や表示が混乱するため避けるべきである。ボタンにActionScriptを記述することで、フレーム遷移のためのページめくりボタンや、びっくり箱のような仕掛け等、様々なインタラクシオンを作成できる(図6,7)。ユーザーが起こした操作にどのような結果を返すかというデザインは、インタフェースデザインとしてのルールを守るとともに、ユーザーが理解できる反応を返すよう注意しなければならない。

3.4. コンテンツデザイン

3.4.1. シナリオゲーム

絵本をデジタル化することで、紙媒体では表現し難かった動きとインタラクシオンを付すことができる。動くデジタル絵本は、フレームごとにアニメーションを付したキャラクターシンボルとともにセリフやページ遷移ボタン等を配置して作成される。シークエンス型のシナリオだけでなく、分岐イベントを設け、その選択行動によって異なる任意のフレームへ移動させることで、分岐シークエンス型のコンテンツを作成することも可能である(図8)。

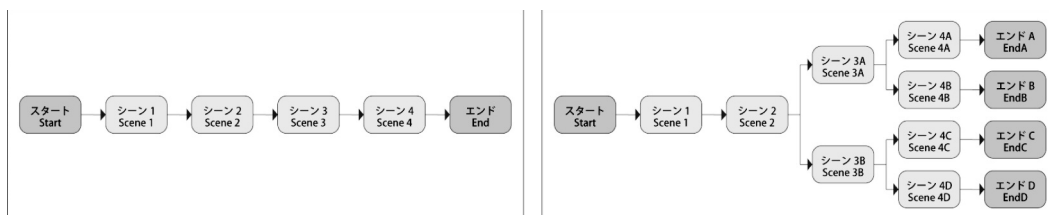


図8：シークエンス型と分岐シークエンス型

3.4.2. 占いゲーム

乱数を発生させ、その値によって任意のフレームへ移動させることができる。ボタンシンボルにActionScriptを記述することで、インタラクシオン発生時のキーとすることができる。ランダムにフレームを分岐させることで、おみくじやタロット占い等のコンテンツの作成への展開が可能である(図9,10)。またサイコロやじゃんけん等、ゲーム内に用いる要素としての応用も可能である。

3.4.3. 着せ替えゲーム

キャラクターの衣装や小物などのアイテムを、要素ごとにシンボルとして分類する。その各要素シンボルをキャラクターに設定した座標軸へ吸着させることで、着せ替えゲームが作成できる(図11)。この吸着はパズルゲーム等でもよく用いられる手法である。また、これを応用することで、ゲーム内においてユーザーが自由にキャラクターをデザインできるようなコンテンツへの展開も可能である。

3.4.4. 落ちものゲーム

X軸に対する移動をユーザーが任意に操作できるように、自機となるキャラクターシンボルへActionScriptを記述する。キーボードでフレームあたりの移動画素数を記述することで、左右への移



図9：占いゲーム



図10：デジタルタロットカード

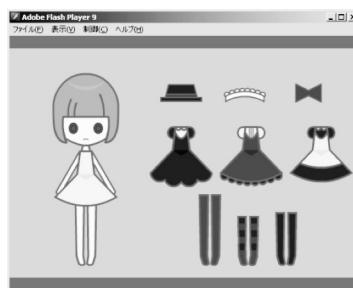


図11：着せ替えゲーム



図12：落ちものゲーム

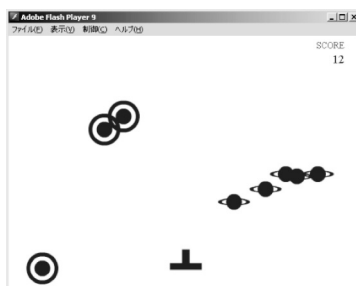


図13：縦スクロールシューティング



図14：横スクロールシューティング

動が可能となる。これに、画面上方からアイテムや障害物となるキャラクターシンボルを下方へと移動させるとともに自機との当たり判定を加えたものを、シンボル上にActionScriptで記述する。シンボルによって得点となる数値の値を変更することで、オブジェクトに役割や性質を与えることができる。このシンボルは、画面下端を通り過ぎた後、再び画面上端へ再表示させる。その際、再表示させるX軸座標に対してランダムな値を取得させることで、分散して落下してくるよう構成することができる(図12)。

3.4.5. シューティングゲーム

落ちものゲームに、弾の発生と敵キャラクターとの当たり判定を加えると、シューティングゲームとなる。自機キャラクターシンボルに、X軸だけでなくY軸への移動を記述することで2次元的な空間の移動が可能となる。マウスで操作させる際、自機を画面上のマウスポインタと同座標に直截的に表示させるのではなく、マウスとの相対座標を取得しその方向へ移動させるといった工夫を加えることで、より自然な操作感を与えることができる。

自機進行方向と逆方向に背景画像を移動させると、画面のスクロールを表現できる。また、画面に対する夫々のキャラクターのサイズや移動速度、移動方向、出現頻度等、ゲームバランスを計ることも必要である(図13, 14)。

3.5. ライブラリの活用

ゲーム等のコンテンツをFlashで作成する際、簡単な構造のものであれば一人でも作成できるが、やや複雑な構造のものの場合、チームで開発を行うこととなる。ActionScript2.0の場合、各種シンボルに直接プログラミングを記述することができることから、直感的に制作できる環境といった大きな利点があり、簡単なコンテンツや開発に伴うプロトタイプ作成等において有利である。その反面、

大規模コンテンツにおいて構造が複雑になると、どこにActionScriptが記述してありそれらが互いにどのような関係にあるのか等、情報を整理しておかなければ記述した開発者ですら把握し難く、機能の追加や修正も容易ではない。また、プロジェクト間で基本的な処理を共有することで、多くのコンテンツを効率的に作成することが可能となる。

そこで、よく使われるコードを整理し、再利用可能な形に情報を整理しておく。ライブラリは、多くのプロジェクトで必要とされる機能を、汎用的な形に整理したものである。Flashはデザイナー側からのオーサリングツールとして機能を実装してきたものであるが、近年、画像処理や物理演算、3D等の複雑な処理を行うActionScriptライブラリが急速に普及し発展してきた²⁾。

3.5.1. 物理演算

ボールが重力によって地面に落ちるよう表現するには、初速と重力加速度を設定し、夫々のシンボルの相対的な距離を考えれば可能である。しかし、ボールが斜面や他の動いているオブジェクトに対しての跳ね返り等を考えると、処理は複雑になってくる。実際のゲームでは、ピンボールのようにボールが複数のシンボルに衝突する。この時、その都度シンボルごとに判定していたのでは処理速度が低下する要因となり、また整理されていない構造のプログラムではゲームバランスを取る上での数値変更も容易ではない。このような処理を行うためには様々な技術的な発想と工夫が必要であるが、物理演算ライブラリを利用することで落下や衝突の運動をシミュレートする処理を扱うことも可能である^{3) 4)}。ライブラリを利用するには、ActionScriptの知識が必要であり、また、設定項目の多さや複雑さ、動きの精度と処理速度等の限界はあるが、これらを利用することで物理法則に則った滑らかな動きが可能となる魅力的なものである。

3.5.2. 3D

ゲーム専用機やネットワークゲーム等では、リアルタイムでレンダリングを行う3Dのコンテンツは珍しくない。しかし近年まで、Flashを用いたWebコンテンツ上でリアルタイムに3Dを表現する手法は一般的ではなかった。Webでの表現は、予めレンダリングした大量の画像を用意するものや、これらの画像をベクタ表現によって滑らかに表示するもの、オブジェクトの拡大縮小や変形を用いたもの等によって、疑似的な3D表現手法が採られてきた。その後、計算速度の向上により複雑な処理が現実的なものとなるにしたがって、Papervision3D⁵⁾等の3Dを扱うActionScriptライブラリが多数発表されてきた。

ライブラリを用いることで、3DCGモデラーで作成された複雑な3次元モデルを、テクスチャー情報とともにCollada等の形式で直接Flashで読み込むことが可能となった。Colladaは、PS3の開発において3DCGアプリケーション間でデータを互換するために策定されたものであり、XMLで3Dデータを記述するファイル形式である。様々な3DCGアプリケーションが対応を始めたが、新しいフォーマットであるためデータの入出力時の不具合も多い。

元来、2Dでの表現を念頭にFlashは開発されてきたこともあり、3D表現は未だ発展の途上である。だが、これらのライブラリによって表現される3Dインタラクティブは、表現の幅を大きく広げ、そこに様々なアイデアと工夫を施すことが可能であり、非常に魅力的な表現手法である。FlashやWeb3D開発環境は、今後もより使いやすい形へ洗練されていくものと思われる。

3.5.3. AR

Augmented Reality (拡張現実感、以下AR) とは、現実の環境に対して人間が情報を得る視聴覚等の知覚に対し、コンピュータによる情報を重ね合わせるものである。視覚情報の場合、その提示手法としてHMD (ヘッドマウントディスプレイ) 等を用いることが主流であったが、近年、身近なデ

イバイスで気軽に利用できるよう、カメラ画像を用いてPCモニターやスマートフォン等で視聴する形のコンテンツが増加している。利用用途も、医療や機械等の分野から、娯楽や広告の分野へと急速に広がりを見せている。

視覚情報に対するARは、現実環境の位置情報を、ARマーカ―やGPSを用いて認識し、コンピュータで描画された画像を表示させるものである。この視覚情報へのARを実現するためのC言語ライブラリであるARToolkit⁶⁾を、ActionScriptライブラリに移植したものとして、FLARToolkit⁷⁾が挙げられる。

Flash上で3Dモデルを扱うことから、FLARToolkitを使用時には3DActionScriptライブラリが必要である。又、Flashへの3Dモデルの取り込みやインタラクションは、3Dライブラリの手順に準ずることとなる。FLARToolkitを利用して作成されたswfファイルを実行すると、Webカメラで読み込んだARマーカ―を空間の基準として、3Dモデルが表示される。この3Dモデルに様々なインタラクションをActionScriptによって付すことができる。

3.5.4. フィジカルコンピューティング

コンピュータに人間の動作情報を入力する際、一般的にはマウスやキーボード、コントローラー等のデバイスを用いて行う。しかし、入力装置の制約に基づき、コンテンツ表現の幅は自ずとして制限されてきた。入力デバイスのユーザインタフェースはライトペンやデータグローブ等、古くから試行錯誤され発展してきたものであり、現在ではペンタブレットやタッチパネル、Wiiリモコン等によって、より直感的な入力環境を支援するデバイスが普及しつつある。

入力デバイスにWiiリモコンを用いるActionScriptライブラリとしてWiiFlashがある。Wiiリモコンは、3軸加速度センサーやモーションセンサーを用いて、直感的な動きを可能とし、それに応じた様々なコンテンツが展開されてきた。WiiFlashはこのWiiリモコンの各種センサーから取得された、傾きや加速度、位置、距離、各種ボタンのon/off信号等の情報を、Bluetoothを介してFlashへ受け渡すものである。これらを用いることで、直感的な操作に応じたコンテンツの作成が可能である。またWiiリモコン自体を分解し、プロダクトそのものを再構成することで新しい入力装置を作成することが可能である。

他にもFlashコンテンツを操作するデバイスとして、ぬいぐるみの両手にペンタブレットのセンサーを分解して埋め込み、画面上のポインタ座標を認識させることでコンテンツを操作させるもの⁸⁾や、マウスやキーボードを分解し新しいデバイスを作る等、発想に応じて既存のデバイスを改造することで新しいインタフェースを作成することが可能である。また様々な外部環境とFlashの入出力を行う際、オープンソースのArduino⁹⁾やGainer¹⁰⁾等を用いることができる。これらにセンサーやアクチュエータ等を組み合わせることで、より自由度の高いコンテンツの制作が可能である。

4. デジタルゲームコンテンツ

4.1. Web絵双六

絵双六は、サイコロを振ってその出た目の数によって、自分のコマを紙の上に描かれたマスに沿って進めていくものである。これは、古くから遊ばれているものであり、老若男女ともに遊ぶことが可能で、ルールが明確である等の点から、デジタルゲームの題材として有用である。そこで、Flashを用い、Web上で動作するデジタルゲームの制作を行った(図15)。素材として藤沢市文書館に所蔵されている古い絵双六を対象とした。この絵双六は東海道中膝栗毛を題材にしたものであり、日本橋を振り出しに、戸塚、藤澤等の宿をマスとし、京都が上りとなる弥次喜多の旅が描かれたものであ



図15：Web 絵双六初期画面



図16：Web 絵双六操作画面

る。この絵双六をデジタル画像に加工し、Flashを用いて乱数を発生させるサイコロと、それに連動してマスを進むコマを作成した。藤沢市文書館によるキャラクターデザインを、3Dキャラクターとして制作し、これをコマとして2人で対戦できるように設計を行った。コマがマス目に止まるとそのマスの画像が正位置で拡大表示される（図16）。Web上からプレイすることが可能であり、公開を前提として作成されたものである。

4.2. Kolam文様

南インドの伝統的文様であるKolamを用いたインタラクティブコンテンツを作成した^{11) 12)}（図17, 18）。Kolam文様を構成要素に分解し、6種類のタイルパターンと各ブロック間の接続パターンに分解する。このタイルパターンを各面に配した立方体ブロックをFlashで作成した。立方体ブロックをフィールド上に複数個に設置し、プレイヤーはこれらを回転させることで上面に一筆書きのパターンを成立させる。判定ボタンを押下すると、選択されたブロックを起点として、フィールド上のパターンがトレースされ、一筆書きが成立しているかどうかを判定する。操作回数やブロックの数等、制限を設けることで、より魅力的なゲームコンテンツへと展開することが可能である。

4.3. やわらかいインタフェース

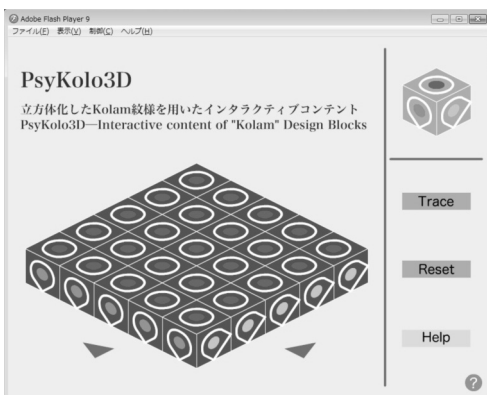


図17：PsyKolo3D 初期画面

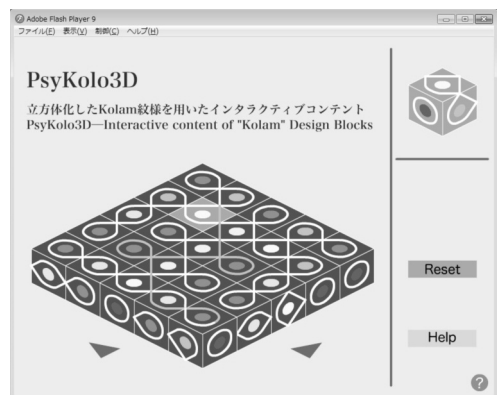


図18：PsyKolo3D 操作画面



図19：やわらかいインタフェース

楽曲のテンポや音階を緩やかにコントロールする、やわらかいデバイスの開発を行った¹³⁾(図19)。袋状の布の中にビーズとともに Wiiリモコンをセンサーとして埋め込み、3方向への傾き情報を BluetoothによってFlashに取り込み楽曲を操作させた。マウスやキーボード等の端末ではなく、自然な操作でユーザーの環境を変化させるものとして、枕やクッションとして使える程度の比較的大きなサイズのデバイスとやわらかいテクスチャーでインタフェースを構成した。ユーザーは数回の操作でデバイスと環境との関連を理解することが可能であり、またその操作方法も直感的なものであることから、児童などでも容易に操作可能である。今後、このやわらかくて大きいプロトタイプと比して、テクスチャーやサイズを変更したものを開発し、そのユーザビリティについて比較分析する。

5. 今後の展望

現在、デジタルゲームの開発環境は身近なものとなり、多くのゲームデザイナーが制作することが可能となった。また、そのコンテンツもインターネットを通じて個人が世界に配信することができる。しかし、面白く魅力的なゲームを作り出すことは、開発環境の大小に関わらず難しい。デジタルゲームはその概念を拡大しつつ、さらに発展する傾向にあることから、これからも次々と様々な立場のゲームデザイナーが新しいコンテンツを発表していくことになると思われる。

魅力的なコンテンツというものは、ユーザーがその世界観に親しみを持つものではないかと思われる。コンテンツ制作が身近になった現在、自分にとって最もなじみ深い、身の回りのローカルな情報を対象としたコンテンツ、自分の住んでいる街や地域を対象としたものや、家族や友人が喜んでくれるようなコンテンツを、丁寧に緻密に作り込むことによって、結果的にそのコンテンツが多くの人々の共感を得られるものになるのではないかと思われる。

参考文献

- 1) CESA: 2010CESAゲーム白書, 社団法人コンピュータエンターテイメント協会, 2010.
- 2) Spark project, <http://www.libspark.org/>, (参照2010-10-07).
- 3) Box2DFlashAS3, 2.0.1, <http://box2df.flash.sourceforge.net/>, (参照2010-10-07).

- 4) Alec Cove: APE - Actionscript Physics Engine, <http://www.cove.org/ape/>, (参照2010-10-07).
- 5) Carlos Ulloa, John Grden, Ralph Hauwert, Tim Knip, Andy Zupko: Papervision3D -Open Source realtime 3D engine for Flash, 2.1.932, 2009-12-01, <http://code.google.com/p/papervision3d/>, (参照2010-10-07).
- 6) 加藤博一: ARToolKit, 2.72.1, 2007-05-18, <http://www.hitl.washington.edu/artoolkit/>, (参照2010-10-07)
- 7) 小 山 智 彦 : FLARToolKit, 2.5.4, 2010-07-10, <http://www.libspark.org/wiki/saqoosha/FLARToolKit>, (参照2010-10-07)
- 8) 益岡了, 尾崎洋, 渡部陽佳: むいぐるみを用いたビデオゲームのユーザインタフェースデザイン, 平成22年度日本デザイン学会第57回研究発表大会, デザイン学研究 研究発表大会概要集(57), pp.52-53, 2010.
- 9) Arduino, <http://www.arduino.cc/>, (参照2010-10-07).
- 10) Gainer, <http://gainer.cc/>, (参照2010-10-07).
- 11) Yasuo KAWAI, Kosuke TAKAHASHI, Shojiro NAGATA: PsyKolo3D -Interactive Computer Graphical Content of "Kolam" Design Blocks, FORMA Vol.22 No.1 (The Society for Science on Form), pp.113-118, 2007.
- 12) 川合康央, 高橋宏輔, 長田昌次郎, 池田岳史, 益岡了: PsyKolo3D -立方体化したKolam紋様を用いたインタラクティブコンテンツ, 平成18年度日本デザイン学会第53回研究発表大会, デザイン学研究 研究発表大会概要集(53), pp.364-365, 2006.
- 13) 小森久栄, 益岡了, 川合康央, 尾崎洋, 門屋博: 玩具を用いたサウンドインタラクシオンデザイン, ヒューマンインタフェースシンポジウム2010 DVD-ROM論文集, 4p(DVD-ROM), 2010.