
目次

- [謝辞](#)
 - [序文](#)
 - [第一章 ゲームとは何か](#)
 - [第二章 人はなぜゲームをするのか](#)
 - [第三章 コンピュータゲームの分類](#)
 - [第四章 ゲーム環境としてのコンピュータ](#)
 - [第五章 ゲームデザインの各工程](#)
 - [第六章 さまざまなテクニックとアイデア](#)
 - [第七章 コンピュータゲームの未来 \(未訳\)](#)
 - [第八章 ゲームデザインの実際](#)
 - [最終章 クリス氏へのインタビュー](#)
-

Web版の編集にあたって

このテキストはもともと、1982年にコンピュータゲームデザイナーのクリス・クロフォード氏 (Chris Crawford) によって執筆されたものです。1997年にスー・ピーボディ教授 (Prof. Sue Peabody) は、ゲームと物語が本質的にどう関わり合っているのかという研究のために文献を探すうちに、この『クロフォードのゲームデザイン論 - コンピュータゲームは芸術たりうるか -』 (The Art of Computer Game Design) という、絶版のため長く入手困難となっていた本を発見しました。そこで、スー・ピーボディ教授は、彼女の講座の学生やゲームデザインに興味を持っている人たちのために、この本を電子化してWeb上で公開する許可をクロフォード氏に対して願い出、受け入れられました。また、ワシントン州立大学バンクーバー校 (Washington State University Vancouver) の好意により、イラストレーターのドナ・ローパー (Donna Loper) の協力を得ることができました。現在、[このサイト](#)はワシントン州立大学バンクーバー校 (WSUV) のサーバー上に置かれて管理されています。

このサイトに関する感想等は、ワシントン州立大学バンクーバー校 歴史学課 スー・ピーボディ教授 <peabody@vancouver.wsu.edu> までお願いします。

クリス・クロフォード氏の最近のコメントに興味をお持ちの方は、巻末のクロフォード氏へのインタビューをご覧ください。また、氏のWebサイト [Erasmatazz](#) を訪れてみるのも良いでしょう。

また、Acrobat形式のテキストが[ここ](#)に置かれています。

日本語版の編集にあたって

日本語版は、クリス・クロフォード氏の許可のもと、@nifty RPGフォーラム (FRPG) の有志によって翻訳、作成されたものです。日本語訳等に関するご意見は、代表のShino <shinoaki@a2.mbn.or.jp> までお願いします。

なお、第七章「コンピュータゲームの未来」に関しては、クロフォード氏が語る未来を現実（2000年）がすでに追い越してしまっているため、あえて翻訳していません。もし、氏の予言の成否に興味があって自分で訳されたという方がいらっしゃいましたらご一報下さい(^_^)

日本語版作成スタッフ

翻訳

[Shino <shinoaki@a2.mbn.or.jp>](mailto:shinoaki@a2.mbn.or.jp)

[O J <oj@v7.com>](mailto:oj@v7.com)

協力（敬称略 ABC順）

Genich! <SDI00769@nifty.ne.jp>

Infrared <JBD04156@nifty.ne.jp>

JSTAK <SDI00659@nifty.ne.jp>

The SiNner <QZF07402@nifty.ne.jp>

タナック <KGG02762@nifty.ne.jp>

西上 柁 <QYM03433@nifty.ne.jp>

著者紹介

アタリゲームズ社でいくつかのシミュレーションゲームのデザインに関わった後、アタリ社の倒産とともに独立フリーのデザイナーとなる。コンピュータゲーム学会 (Computer Game Developers Conference) の創設者としても知られ、いくつかの大学でゲームデザイン論を教えている。

代表作にMac版のシミュレーションゲーム『バランス・オブ・パワー』（Balance of Power : 1986年作品）がある。米ソ冷戦時代の外交を扱ったこのゲームでは、プレイヤーは米国の大統領（またはソ連の書記長）となり、両者の間に起こるさまざまな事件に関する折衝を行う。弱気になれば自陣営が衰退し、かといって強気に出すぎると全面核戦争が勃発しゲームオーバーというそのバランスは、まさに冷戦を再現したゲームとして話題を呼んだ。

参考

- クリス氏のWebサイト : [Erasmatazz](#)
- アタリ社のゲーム一般 : [ATARI 2600/2800 "VCS" ソフトレビュー](#)
- コンピュータゲーム学会 : [Game Developers Conference](#)

謝辞

この本を書くにあたり、詳細かつ惜しみない批評を与えて頂いたマドリン・M・グロス嬢 (Madeleine M. Gross) に深く感謝いたします。彼女は多くの場合において、私の原案に対して私以上に一生懸命に厳しい批評を加えてくれました。ともすれば誇張となりがちな私の表現を引き締め、論理基盤をより厳密なものとしてくれたのは彼女の努力の賜物です。この本の論理が首尾一貫し、信頼できるものになったとすれば、それは彼女のおかげでしょう。また、妙な論理的飛躍が残っていたとすれば、それはすべて私の責任であります。

序文

この本は「コンピュータゲームは斬新な、しかしながら未熟な芸術の一形態をなしており、それはゲームデザイナー、プレイヤー両者にとって大きな可能性を秘めている」という前提のもとに構成されている。

この前提は、あまりに不遜であり嘲笑の対象となるべきように思えるかもしれない。いったい誰が、『スペースインベーダー』(SPACE INVADERS)や『パックマン』(PAC-MAN)を芸術だなどといえるだろう。『テンペスト』(TEMPEST)や『ミサイル・コマンド』(MISSILE COMMAND)を、ベートーヴェン(Beethoven)の『交響曲第五番』(Fifth Symphony)やミケランジェロ(Michelangelo)の名作『ピエタ』(Pieta)、ヘミングウェイ(Hemingway)の小説『武器よさらば』(A Farewell To Arms)と同列に比較する者などいるだろうか。コンピュータゲームは芸術と呼ぶにはあまりに矮小でくだらない。せいぜい、時間潰しのお遊びといったところだろう。懐疑論者はそう言う。

しかしながら、コンピュータゲームの現状を見て、一過性のポップカルチャーとして汚物ために放り込むのは早計だろう。コンピュータゲームは、まだあまりにも若く、あまりにも急速に変化を続けている。そう簡単に、コンピュータゲームを見捨ててしまっただけではないのだ。現状ではなく可能性を見なくてはならない。結論を出す前に、コンピュータゲームの本質とは何かということを考えなくてはならない。そうすることで、コンピュータゲームが、他の一過性の流行のように、時と変化の中で朽ち果ててしまわないようにするための道を見いだすことができるだろう。

芸術にはさまざまな定義があるが、芸術論について不慣れな初心者がそのまま理解できるようなわかりやすい定義はほとんどない。ここでは、私なりの定義をあげておこう。芸術とは、イメージを通して感動を呼び起こそうとするものである。芸術家は、観客の知覚にさまざまな経験を与えることで、彼らと芸術家が一般に共有しているイメージを刺激する。それが感動を生むのだ。芸術が存在するのは、あくまで我々人類が共有しているイメージが豊富であるためなのである。にもかかわらず、何かが芸術となることは困難である。なぜなら、人の心の奥深くに沈むイメージを呼び起こすことを疎外する多くの問題があるからである。ひとつに、観客の注意を引いて芸術への参加意識をもたせることが難しいことがあげられる。観客の直接参加が可能な芸術はほとんどない。観客に許されているのは音楽家が演奏する音楽をただ静かに座って聞くことや、美術館を歩き回って他人が作った絵画や彫像を眺めること、じっと座って小説や詩を読むことだけなのだ。いずれの場合も観客の立場は受動的なものである。芸術家はその感情の最大限を投資して、すべての能動的な行動を行い、観客はただじっとそれを鑑賞することですべての利益を得ることを期待されている。観客の能動的な参加は極めて限られている。そして、参加意識がなければ、観客の集中力は徐々に減退し、いつしか感動も消えていってしまうのである。

これまでの芸術家たちが作り上げてきた芸術を批判しようという気はさらさらでないが、従来の芸術で使われてきた技術そのものが観客の参加を拒んでいるのである。音楽を演奏中のオーケストラピットへ飛び込んだり、オペラのステージへ駆け上がったたり、はてはピカソ(Picasso)の名画に絵の具をぶちまけたりすれば、確かにドンチャン騒ぎに参加していることにはなるだろうが、それはもはや芸術ではない。芸術家たちがどんなに自らの作品に関して語っても、観客が参加意識を持たないために耳を貸そうとしないというのは、芸術に課せられた呪いということが出来るだろう。

話をコンピュータに戻そう。コンピュータ、考える機械という発想そのものは非常に古い。

それは、数々の戦争の中で産声をあげ、ビジネスに利用されることで成長してきた。そしていま、このまだ幼い技術は、コンピュータールームの中からショッピングセンター、ピザ・パーラーそして各家庭へとその居場所を広げてきている。コンピューターのイメージそのものも、全知全能、冷血な巨大計算機というイメージから、100円玉ひとつでビデオ映画並みのスリルを与えてくれるものというように変わってきている。生まれたばかりのころは、数字をばりばり処理するだけの機械だったのに、グラフィックとサウンドを処理する能力を得て、まったく違ったものとなったのである。グラフィックとサウンドの処理が可能になったことで、コンピューターは非常に強力な利点を持つことになった。従来冷たくわかりにくい数字の羅列ではなく、直接人間の感情に訴えることのできる表現方法を身につけたのだ。これにより、以前には想像もしなかったような可能性が出てきた。コンピューターを、人間の感情に働きかける芸術の媒体として考えることが可能になったのである。コンピューターゲームは、芸術の媒体としてのコンピューターの利用法の先導者として現れたのだ。コンピューターゲームは、私の定義のもとでは芸術の一形態である。なぜなら、それはゲームをプレイする観客に、感動を呼び起こす仮想体験を与えるのだから。

残念なことに、いまのパソコンレベルではオーケストラや映画並みの臨場感を与えることはできない。しかしながら、この欠点を補って余りある利点がコンピューターゲームには存在する。ゲームは本質的に観客自身が参加するものなのである。芸術家は従来の芸術に比べ、より繊細で間接的な道具を手に入れたのだ。従来の芸術では、芸術家は自らの手で直接、観客に披露する体験を作り出す。それは、非常に精密に計画、実行され、観客はその進行を乱すことを許されない。つまり、本当の意味でそれに参加することはできないのだ。しかし、ゲームでは芸術家が作るものは、体験そのものではなく、観客たちが自ら体験するための環境とルールなのである。芸術家に要求されるものは、以前の芸術と比べても多大である。なぜなら、彼／彼女らは観客に与えようとする体験を間接的に設計しなくてはならず、体験に参加する観客のすべてのアクションとリアクションを計算に入れておかななくてはならないからだ。もちろん、見返りももっと大きい。実際に参加することで、その体験が与える感動はますます大きくなるのだから。誰かの芸術を受動的に鑑賞しているときには、そこからわざわざかばかりの感動を引き出しているだけなのに対し、実際にゲームに参加することで、自意識の一部をその仮想体験の中に送り込むことができるのである。臨場感が大きくなるのと比例して、そこから受ける感動も大きなものとなる。実際、アマチュアの芸術家たちと一緒に芸術に参加しているときのほうが、プロの芸術をただ眺めているときよりも感動が深いではないか。すなわち、ゲームは本質的に参加型であるために、それを利用する芸術家たちがより観客に近づくことを可能にしたのである。

いまのところ、ゲーム一般、特にコンピューターゲームはそれほど印象深い芸術形態にはなっていない。コンピューターゲームはまだまだ幼稚な代物である。これは表現の手段としてのコンピューターゲームが、芸術家ではなく技術者の手にあったからである。彼ら（男性代名詞を使ったが、プログラマーのほとんどが男性であることに文句をつける人はいないだろう）にとっては、美しいOS、プログラム言語、リンカといった技術的に優れたものを作り出すことが第一で、芸術的なセンスなどというものは二の次だったのだから。

それに加えて、市場が立ち後れていることも、現在のコンピューターゲームがぱっとしない理由である。コンピューターはまだ一般の人々にとって目新しいものであり、ゲームを供給する側もこれほど急速に発展しつつあるものを、一般の人々に押しつけることにためらいを覚えているのだ。そのために我々デザイナーは、冒険をさげ、つまらない、観客にこそそとささやいてありきたりの感動をよびさまそうとするようなゲームを作るようになってしまっている。本当は、強烈な感情を呼び覚ますような状況、すなわち、パトス、エクスタシー、荘厳なイメージ、狂喜、カタルシス、悲劇といったものこそ、ますますゲームの題材として取

り扱うべきなのだ。我々が作っているのは芸術ではなくエンターテインメントなのだという言葉は、芸術をその大本から誤解している者たちにしか通用しない。確かに、パリッと糊のきいた服を着ているエリート主義の者たちからも芸術は生まれることがあるが、大衆たちが足を踏み鳴らすリズムが素晴らしい芸術となることもあるのだ。芸術を研究している者たちは、えてしてエリート主義になりがちだが、芸術のインパクトは感情に素直になるところから生まれるのである。

幸か不幸か、時は激しく流れている。すでにあちこちでコンピュータゲームに対する反動勢力が現れてきているのがわかるだろう。たとえば、アーケードゲーム機の設置場所に対する規制などというのもそうだし、ゲームに夢中になっている子どもたちを厳しく叱る両親などというのもそのひとつだ。こういった批判は、ゲーム業界の小心者たちには悩みの種となっているし、もっと先が見える連中すら、ゲームを守ろうという立場には立たずに、むしろ興味深く成り行きを見守っているだけだ。ゲームを弾圧しようとしているアメリカ国民は、われわれゲーム業界の者たちに何かを伝えようとしているのだ。そう、とても大切な何かを。それは彼らにとって、人の仕事を邪魔しようとしているという汚名を浴びるに足るほど、大切なことなのだ。学校の崩壊、不良集団の発生といった話題が公開の場にあがるたびに、彼らのゲームに関する嫌悪感、ゲームは時間の浪費であるという漠然とした印象が、どこからとなく表面に現れてくる。いつまでも、彼らをだまし続けているわけにはいかない。彼らは、すでに気付き始めている。ゲームは未開で不毛なものであるということに。

コンピュータゲームは、キャンディやマンガ、アニメのようなものだ。これらは皆強い刺激を与える。砂糖を加えたり、感嘆符を使ったり、大爆発を起こしたりするのは、すべて強い刺激を与えるためである。子どもたちはあらゆる面において新鮮味を感じる力が強いために、そういった誇張された表現を楽しむことができる。一方、長い間にそんな刺激に飽き飽きした大人たちは、もっと巧妙で奥深いものを好む。子どもにとってのキャンディ、マンガ、アニメといったものは、豪華な料理のフルコース、ずらりと並んだ文学、超大作の映画などに取って代わられた。しかし、コンピュータゲームの代わりになるものはまだ生まれていない。そう、ゲームデザインを洗練させていくことで、この穴を埋めることができるかもしれないのだ。

コンピュータゲームを芸術たらしめようという、いま起きつつあるこの大変革には、実は近年の急速な技術革新はまったく関係ない。いまの技術革新の速度を考えれば、もはやハードウェア上の限界からやりたいことができないということはなくなっていくと思われる。真の問題点は、我々がゲームを芸術たらしめる上で、いったいどこに力を注ぐべきなのかということに関して、ほとんど考察したことがないということなのだ。ゲームとは何なのか、人はなぜゲームをするのか、いったい何が素晴らしいゲームを作るのか……。こういった根源的な命題に関して、我々は本当は何もわかっていない。ゲームを通して芸術を表現することは不可能ではないはずである。しかしそれは、ゲームの本質を理解することなくして、決して達成することはできないだろう。我々は、ゲームに関する審美眼を磨き上げ、評価基準を選定し、ゲームデザインのモデルを作り上げなくてはならない。ハードウェアの進歩は確かにゲームを進化させてくれる。だからといってそれがゲームを芸術にしてくれるわけではないのだ。いくらオーケストラを揃えたってベートーヴェンが生まれてくるとは限らないように。コンピュータゲームがシェークスピア (Shakespeare) の戯曲やチャイコフスキー (Tchaikowsky) の交響曲、ファン・ゴッホ (Van Gogh) の自画像と肩を並べて比較しうるようになるまでには、はるか長い道のりが必要だろう。これらの芸術は、はるか昔から数々の先人たちが切り開いてきた遺産の上に成り立っている。我々、コンピュータゲームデザイナーも、後に続く者たちのために少しでも道を開いておかななくてはならない。この本は、そのために書かれたのだ。

[目次に戻る](#) | [各章へ: 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9](#) | [次章に進む](#)

ボードゲーム
トランプゲーム
体を使うゲーム
子どもたちのゲーム
コンピュータゲーム

世界の再現
インタラクティブ性
対立関係
安全性

ゲームとゲームデザインについて理解するためには、まず、基本的な定義をはっきりさせておかななくてはならない。いわゆる「ゲーム」という言葉の表すものを、明確に示すことが必要である。さらに、すべてのゲームと呼ばれるものの基本的な特徴を示さなければならない。まずゲームを定義することの難しさについて述べた後に、個々のゲームをその特徴によって分類しよう。その上で、すべてのゲームに関して共通する、ゲームの本質といえるものを提示していきたい。

ゲームは、人間の存在そのものと根源から関係している。そのため、ゲーム用語はいつしか、本当の意味でゲームとはいえない活動を表すのにも使われるようになってしまった。特に英語のplayやgameといった単語には慣用句が多く、協力するふりをするとか、計略をめぐらすとか、場合によってさまざまな意味に使われている。このように、人間の活動の中にゲームの概念があまりに浸透しているため、ゲームの本質を理解する上でふたつの問題が出てきてしまった。

第一の問題は、ゲーム用語があまりにあちこちに広まってしまったために、ゲームとは何かということを理解したつもりになってしまっていることである。そのために、ゲームに対して学術的な分析を行うときに必要な、注意深い細かな見方ができなくなってしまい、ゲームデザインがいかに複雑なものであるかということをやたらと無視しがちになっている。その結果、プログラミングの技術しか持っておらず、プレイヤーとしての経験しかないような素人の素人が、ゲームデザインを引き受けることになっている。なまじ、ゲームについて理解した気になっているために、正しいゲームデザイン論を学ぶ機会を失っているのだ。

第二の問題は、ゲーム用語が指し示す範囲があいまいなことである。言葉の意味を広げすぎてしまったため、その本来の意味は薄められてしまっている。もはや、いま我々が理解しようと努力している「ゲームとはこれこれだ」などという、はっきりとした概念は存在しない。ゲームデザイナー同士がゲームデザイン論について語り合うことができるような、明確に定義された言葉がないのだ。言葉の意味があいまいであるがゆえに、ゲームデザイン論はしばしば不毛な論争となったあげく、崩壊してしまう。ゲームデザイン論を始める前に、ゲーム用語のまわりに生い茂っているあれこれを、メスなりブルドーザーなりを使って取り除いてやる作業が必用なのである。

というわけで、本論に入る前に少し後戻りして我々の立場について論じておこう。ゲームと呼ばれているものをざっと見渡して、その大まかな分類について簡単に紹介する。個々のゲームに関する記憶を呼び起こすことで、ゲームの本質に関する細かな分析に入る前に、いくつかの点を指摘することができるだろう。ここでは、ゲームを以下の五つに分類する。ボードゲーム、トランプゲーム、体を使うゲーム、子どもたちのゲーム、そしてコンピュータゲームである。

ボードゲーム

まずは、ボードゲームから始めよう。ボードゲームには、いくつかの領域に分割されたゲームボードが用いられ、その上で複数のコマが配置され、移動させられる。よくあるルールでは、一個一個のコマが直接プレイヤーと関連付けられている。そして、ボードはゲームを行う環境を提示するだけで、プレイヤーは直接ボードそのものに影響を与えることはできない。プレイヤーは自分のコマを動かすことで、人のコマを取ったり、ゴールに向かったり、領土を広げたり、その他いろいろな勝利条件を目指す。ボードゲームにおけるプレイヤーの最大の関心事は、各自のコマがボード上でどんな位置関係にあるかを分析することである。

トランプゲーム

次は、トランプについて見てみよう。一般的なトランプゲームは、ふたつの要素（1から13までの数字と、スペード、ハート、ダイヤ、クラブの四つのスート）の組み合わせからなる、52枚のカードを使う。ゲームは、そのふたつの要素の組み合わせを中心として回っていくのだ。プレイヤーは、ランダムにしる、何かの順番にしる、ルールで定められた方法にしたがってカードをやりとりする。ルールで決められたそれぞれのカードの組み合わせが勝利点となり、ゲームの結果に影響する。プレイヤーは、いま持っているカードから期待できる組み合わせを考えて、それを完成させるのに必要なカードを手に入れる可能性を吟味しなくてはならない。もちろん、その可能性には、その組み合わせで得られる勝利点がどれほどかということも考慮されている必要がある。しかしながら、考えられる組み合わせの数があまりに多いために、正確な確率の評価は人の手に余ることが多く、結局はプレイヤーの直感を鍛えることになることが多い。ここでは、プレイヤーの第一の関心事は、カードの組み合わせを分析することである。

体を使うゲーム

古くからある別のゲームのひとつに、体を使うゲームがある。こういったゲームでは、精神的な能力よりもむしろ肉体的な能力の優劣が強調される。ここでは、ルールによって厳格にプレイヤーがやって良い、あるいはやらなくてはならない体の動かし方が決められている。いかに勝つために自分の体を効率良く動かすかが、体を使うゲームにおけるプレイヤーの第一の関心事である。

ここで、ゲームと競争の違いには注意しておかなくてはならない。たとえば、マラソンなど、一般にレースと呼ばれている競技は、ゲームではなく競争である。競争とゲームを分けるものこそが、すべてのゲームに共通するゲームの本質のひとつである。それは、プレイヤー間の相互作用、インタラクティブ性の大小である。理屈の上では、徒競走中の選手は相互作用をしていない。個々の選手がそれぞればらばらに、ストップウォッチと戦っているだけなのだ。他の選手がいるかどうかは大して重要ではないはずだ。ところが、現実には、個々の選手が精神面で相互作用するため、ある選手の記録が別の選手の記録に影響することになる。レースの種類によっては、選手（ドライバーでもパイロットでも艇長でも何でも良い）は、自分が優位に立つために、他の選手の進路を塞ぐこともできる。すなわち、個々の参加者が他者と関りを持たずに、自分の成績を上げようと努力するようなものが最も単純な競争であり、参加者同士の相互作用があるような競争こそが、すなわちゲームなのである。

子どもたちのゲーム

子どもたちのゲームも、ゲームの一形態である。かくれんぼ、おにごっこ、かんけりなどが

その例だ。こういったゲームはしばしば、単に体力だけが強調された集団ゲームのような形を取る。これらのゲームも単純な形ながら体力、精神力の両面を持ってはいるが、子どもたちにその能力の限界に挑ませようとか、そういう意図を持ったものではない。子どもたちのゲームにおけるプレイヤーの関心事は、いかに社会的なスキルを使うかである。それは、本当の人間社会における人間関係への基礎となっている。

子どもたちのさまざまな行動が、しばしばゲームであるといわれることがある。たとえば、子どもが、剥ぎ取った木の皮に向かって話しかけ、それを振り回し、びゅんびゅんという音をさせるとき、いかにもゲームをしているように見える。しかしながら、この本では、このような遊びはゲームとしては考えない。このような、ルールのない即興的な遊びは、ゲームの本質論を論ずる上で不適當だからである。

コンピュータゲーム

いよいよこの本の主題である、いま流行のコンピュータゲームである。まずは、その概略だけを簡単に見ていこう。いま、コンピュータゲームが走っているコンピュータには大きく分けて次の5種類が有る。高価なアーケード機、ゲームウォッチに代表される安価なハンドヘルド機、アタリ2600 (ATARI 2600) やアタリ5200 (ATARI 5200) のようないわゆる家庭用ゲーム機、そして、パソコンとメインフレーム機である。多くの場合、コンピュータはゲームの中でプレイヤーの対抗者やレフリーをつとめる。グラフィックアニメーションを動かすのもコンピュータだ。コンピュータゲームの代表はアクションゲーム (the skill and action : "S&A" game) であり、動体視力と手先の素早さを競うゲームである。アクションゲームの中には本質的に暴力的なものも多い。その他にも多種多様なコンピュータゲームが存在する。たとえば、アドベンチャーゲーム、ファンタジーロールプレイ、ウォーゲームなどだ。しかし、コンピュータゲームをざっと見た感じでは、そういったその他の種類のゲームは、アクションゲームと比べればその影に隠れてしまうほどの量になってしまうだろう。

さて、これでとりあえずゲームの世界をざっと見て、そのおおざっぱな分類について述べたことになる。コンピュータゲームの分類と、その本質の抽出は後に回そう。ここで、我々の最初の質問に戻らなければならない。そう、「これらすべてのゲームに共通するものとはいったい何だろうか」という質問に。この質問に対して、私は四つの解答を提示しようと思う。それは、世界の再現、インタラクティブ性、対立関係、そして安全性である。

世界の再現

ゲームとは、クローズドで形式的なシステムであり、現実世界の一面を主観的に再現するものである。難しい言葉を並べてしまったが、それぞれの言葉について細かく説明して行こう。

クローズド

「クローズド」とは、ゲームがそのルールの中で完全であり、それだけで成立しているということである。すなわち、ゲームが表現しようとしている世界がそのルールだけで表現され、ゲームの外にいる者たちが他に資料を探したりしなくて済むということだ。デザインに失敗したゲームの中には、この条件を満たしていないものもある。そのようなゲームでは、ルールに書かれていないような状況に陥るたびに、ルールの解釈に関して論争が始まってしまう。状況に合わせていちいち、ルールを拡張しなくてはならないのだ。喧嘩にもなるだろう。きちんとデザインされたゲームならそんな心配はない。クローズドなゲームは、ゲーム

中に起こりうるすべての事象をルールがカバーしているのである。

形式的

「形式的」とは、単に、ゲームに明確なルールが存在するということの意味している。中には、ルールがおおざっぱに記述されていたり、故意にあいまいになっていたりする、形式的でないゲームもある。しかし、そういったゲームは、ゲームの本流とはとてもいえないだろう。

システム

「システム」という言葉はとかく誤用されがちであるが、ここではまさに正しい意味で使われている。互いに、ときには非常に複雑に相互作用する、ゲームの構成要素の集合体。それこそまさに「システム」である。

現実世界の一面

私は「現実世界の一面」という言い方をした。「一面」という言葉の意味は容易に理解できるだろう。ゲームが現実そのものでない以上、ゲームの中に本当の現実があるはずはない。すなわち、ゲームが表現するのは、たかだか「現実世界の一面」なのである。その中から、どの部分を取り上げるかによって、ゲームの目指すものが決まってくる。むやみにいろいろなものを詰め込めば、再現される現実世界が大きすぎてやたらとややこしいものになり、プレイヤーの理解を越えてしまうだろう。それは、ゲームから、最も大切な「焦点」を奪ってしまうことになる。

主観的に再現する

世界の再現は、コインのようにその両面を持っている。すなわち、その世界を客観的に見るか、主観的に見るかという両面である。これらは決して相容れないものではない。主観は客観から生じ、客観は主観から影響を受けるものだからである。ゲームにおいてもその二者は互いに絡み合っているが、主観的な面のほうが強調される。たとえば、ゲームの中で数百匹ものエイリアンを虐殺したところで、それがプレイヤーの現実世界にそのまま反映すると考える者はいないだろう。しかし、彼がプレイしているゲームには、彼が現実世界をどう思っているのかが非常によく表れているかもしれない。ここで、「彼のやっている殺戮は、彼の心に深く染み付いている不満のはけ口となっている」などという精神分析を始めたりして、この論文をあさっての方向に持っていく気はない。しかし、彼の心の中で単なるエイリアンの殺戮以外の「何か」が起きているのは間違いのないところだろう。それが何であるかを明確に示す必要はない。プレイヤー自身が心の中に想像した世界を反映した何かである、ということを理解しておくだけで十分である。

すなわち、ゲームはプレイヤーにとって主観的な、決して客観的ではない何かを再現するものなのである。ゲームの世界は、客観的に見れば決してリアルではない。ゲームの中で起きているようなことは、現実にはできないのだから。しかし、プレイヤーの主観の中では、それでも十分にリアルな世界を再現することができる。客観的に見て荒唐無稽な状況を臨場感あふれるものに変えるのは、プレイヤーの想像力である。このように、すべてのゲームにおいてプレイヤーの想像力は重要な役割を果たす。ゲームが再現するのは、科学的モデルではなく、プレイヤーの想像力を刺激する世界なのである。

ゲームとシミュレーションの違い

主観的な世界の再現と、客観的な世界の再現の違いは、ゲームとシミュレーションの差を考えるとわかりやすい。シミュレーションは、実際に起きる現象をさまざまなパラメータを用いて精密に表現しようとする。一方、ゲームでは、その現象をできるだけシンプルに表現しようとするのだ。シミュレーションの研究者は、あまりに複雑で計算が追いつかないとか、現象がややこしすぎて理解できないという場合に、仕方なしに現象の単純化を行う。それに対して、ゲームデザイナーはデザイナー自身が一番大事だと思っているパラメータにプレイヤーの意識を集中させるために、喜んで現象を単純化するのである。両者の目的には明確な差があるのだ。シミュレーションは、何かを計算したり評価したりするために行われるのに対して、ゲームは娯楽のため、そして何かを教育するために行われるのである（もちろん、その中間的なものも存在する。たとえば、教育のためにシミュレーション的な要素の多いゲームが行われる場合など）。シミュレーションには正確さが不可欠であり、ゲームには簡潔さが要求される。もちろん、シミュレーションの技法は、ゲームと深い関係を持っている。絵画を描くためには、筆の使い方を知る必要があるように。しかし、ゲームは決して単なる小規模のシミュレーションではないのだ。そこには、シミュレーションに不可欠な細部の精密な描写が抜けているのだから。ゲームでは、精密な描写は、デザイナーがプレイヤーに伝えたいと思っているものを際立たせるためにあえて省略されているのである。すなわち、シミュレーションは詳細な表現を重んじ、ゲームは様式的な表現を重んじているのである。

たとえば、パソコン用のフライトシミュレータとアーケードゲームの『レッド・バロン』(RED BARON)の違いを考えてみよう。どちらも、プレイヤーが航空機を操縦するし、マイクロコンピュータ上で動いているという点でも同じだ。しかし、フライトシミュレータが失速、ロール、錐揉みといった非常に多彩な現象を再現しているのに対して、『レッド・バロン』では何ひとつとして再現できない。事実、『レッド・バロン』でプレイヤーが操る戦闘機は極めて非現実的なものである。失速、ロール、錐揉みはおろか、地面に墜落させることすらできないのだ。飛行中に操縦桿を放しても、自動的に水平飛行に移るのである。だからといって、『レッド・バロン』がフライトシミュレータよりも劣っているというわけではない。『レッド・バロン』は、本物の飛行機の操縦を楽しむゲームではなく、空中戦を楽しむゲームなのだから。もし、ただ空を飛ぶためだけに複雑な操縦テクニックが必要だとしたら、ほとんどのプレイヤーがゲームを楽しむことができなくなってしまうだろう。『レッド・バロン』のデザイナーは、細かな操縦テクニックの部分をすっかり削ぎ落とすことで、プレイヤーの意識を空中戦に集中させたのである。その飛行が現実的ではないということは、ゲームの目的をプレイヤーにはっきりと提示する上で、欠点ではなく利点なのである。もちろん、フライトシミュレータでそんなことをすれば、欠点になってしまうことは間違いない。

世界の再現に関するまとめ

ゲームは主観的で、あえて単純化されたイメージとしての現実を再現する。それは、客観的に見て現実を忠実に再現したものではない。現実の精密な描写は、プレイヤーの想像力を助ける場合にのみ必要とされる。プレイヤーの想像力こそが、ゲームをリアルだと感じさせるための鍵なのである。

インタラクティブ性

世界を静的なものとして再現する表現手段が存在する。たとえば、絵画や彫刻は、刻々と流

れる時間のある一瞬を切り取って表現する。一方、世界を動的なものとしてあらわす表現手段も存在し、映画、音楽、そしてダンスなどがそれにあたる。時間の中での変化を表現するのだ。動的な表現は、静的な表現よりもより豊かに変化を表現することができる。しかしながら、現実世界を表現する上で最も魅力的なのは、現実世界がある一瞬にどうだったかでも、それがどう変わるかでもない。それがどうやって変わって行ったかなのである。互いに複雑に絡み合った原因と結果のもつれあい、それが最も魅力的なのだ。観客たちになぜそうなったのかを十分に伝えるためには、彼らに隅から隅まで覗き込むことを許して、事件の原因を作り、その顛末を観察することを可能にするしかない。そう、最も完全で高度な世界の再現手法は、観客が参加できるインタラクティブな方法なのである。ゲームは、インタラクティブな一面を持っており、これが非常に重要な役割を果たしている。

ゲームとパズルの違い

ゲームのインタラクティブ性を理解してもらうために、ゲームと、パズルやその他のインタラクティブ性のない競技を比較してみよう。ルービックキューブとまるばつ、あるいは走り高跳びとバスケットボールを、それぞれ比べてみると良い。どちらの場合もプレイヤーに課題を与えるという点では同等である。大きな違いは一方はインタラクティブであり、他方はそうでないということだ。ルービックキューブはプレイヤーが歩いていようが座っていようが関係ないし、走り高跳びのバーは選手がどんなに努力していたかを知って落ちずにふんばるわけでもない。それに対して、まるばつやバスケットボールでは、相手の知識や反応がプレイヤーの行動に直接関係してくるのである。

ゲームとパズルの差は、状況を少し変えるだけでほとんどなくなってしまう。パズルや運動競技をゲームにすることもできるし、その逆も可能だ。たとえば、チェスはまさにゲームの一種だが、それが元になった詰めチェスはパズルである。ゲームは、他のいろいろなものを取り込むことができるのと同じように、パズルをその一部として取り込むことができる。ほとんどの場合、パズルはゲーム全体の中のほんの一部である。パズルに重きをおいているゲームは、そのパズルがいったん解かれてしまうとその価値を失ってしまうだろう。

ゲームと物語の違い

ゲームにおけるインタラクティブ性の役割を別の面から見るために、今度はゲームと物語を比較してみよう。物語とは、言い替えればいくつもの原因と結果を時系列に並べたものである。そこに現れる出来事は、フィクションであることも多い。なぜなら、物語の中で個々の事象は本質的には重要でないからである。事実、個々の事象は重要ではないという観点に立てば、フィクションという言葉の持つ本当の意味、すなわち「真実ではないが嘘ではない」ということが理解できるだろう。個々の事象の因果関係こそが、物語において重要な役割を果たすのである。たとえば、『スター・ウォーズ』(STAR WARS)を見て、「ルーク・スカイウォーカー」(Luke Skywalker)や「デス・スター」(Death Star)が実在するものであるかどうかを気にする者などいないだろう。そこには「ルーク」の純粹で正義感にあふれた行動、「デス・スター」の邪悪さ、そして、「ルーク」が「デス・スター」を倒す姿が描かれている。我々はこの物語を見て、「正義は必ず悪に勝つ」という因果関係を思い浮かべるのである。このように、物語とは現実を再現するためのひとつの方法であるが、それ自身が現実を再現するのではなく、その中で描写される事象の因果関係によって現実を再現するのである。

ゲームもまた、現実を再現しようとする。ゲームと物語の差は、物語では描写される事象が固定されているのに対して、ゲームではプレイヤーの選択によって枝分かれしうるという点

である。物語を鑑賞するときは、一本道に沿った事象の連なりによってその因果関係を追っ
ていかなくてはならないのに対し、ゲームのプレイヤーは、その事象の逆、裏、対偶を探す
ことで、よりその因果関係をはっきりさせることができる。ゲームプレイヤーはより多くの
角度から物事を見ることができるといえるのだ。

事実、プレイヤーは違う展開になることを期待して、毎回違った戦略でゲームをプレイす
る。それに対して、物語はどうしても一度きりになりやすい。二度三度と繰り返し読んで
も、新しい情報がなければ感動は薄れていってしまうのである。ゲームの与える感動は、プ
レイヤーがすべての別れ道を行きつくすまで尽きることはない。

私は、ゲームが物語よりも優れているなどと言いたいのではない。物語の中で描き出される
事象は、確かに一過性のものだが、それゆえにゲームよりもより複雑に、より詳細に描写す
ることができる。細部の描写こそ、物語の成功の鍵である。観客は、細かな描写から産み出
される現実味により、作者が期待した印象へと導かれる。一方、ゲームデザイナーは、悪知
恵をめぐらしてたったひとつの真実をあちらこちらから見せるような複雑な幾本もの筋道をつ
むぎだす。この点から、物語は、彫像的であり、ゲームは宝石的であるといえることができる
かもしれない。彫像は、その細部の描写の巧みさに価値があるが、宝石にとって細部は重
要ではない。それこそ、宝石の表面は完全につるつるだって構わないのだ。宝石の価値は、
いかにさまざまな方向の光を反射して輝くのかにあるのである。彫像は、ある場所に置かれ
る事によって鑑賞され、宝石は身につけて持ち運ばれることでさまざまな表情を見せる。そ
う、物語は静的であり、ゲームは動的なのである。

いまのところ、物語がコンピュータゲームに対して優位に立っている点が存在する。それ
は、物語は観客に驚きを与えるということである。良くできた物語は、いくつもの面白い筋
書きが絡み合っていて、語り手は、物語をつむぐ中で我々読者にこうなるのだろうな
という期待を抱かせ、それをうまく裏切るような、まったく新しいドラマチックな状況を作り
出してみせる。こういったどんでん返しは、一連の物語の中で何度も行われる。一方、コン
ピュータゲームでこういったあっと驚かすような筋書きを与えてくれるのは、アドベン
チャーゲームくらいのものだ。残念なことに、こういったどんでん返しを仕掛けるために
は、プレイヤーを特定の状況に誘導するために、プレイヤーの行動を制限する必要がある。
その結果、アドベンチャーゲームの多くは安易な一本道になりがちである。本当なら、プ
レイヤーの行動に応じてさまざまに変化すべき物語が、押しつけの安易な一本道になっ
てしまっているのである。本当にプレイヤーを驚かそうと思ったら、プレイヤーの行動を分析
し、そこから彼がいま何を思っているのかを読み取り、彼に不満を与えないような方法で、
物語的に無理のないどんでん返しを提示する必要がある。いまだ、こんなことのできる人工
知能は存在しないが……。

ゲームとおもちゃの違い

ゲームは、その使用法の点で物語とおもちゃの中間に属する。観客は、物語の中で提示され
た事物に手を出すことは許されない。ゲームの中でなら、空想世界の事物にプレイヤーが参
加することもできる。しかし、その空想世界を形作っているルールは明確に提示されてい
る。一方、おもちゃでの遊び方はもっと自由である。おもちゃは想像力の許すかぎり、どん
な風に扱っても構わない。物語の語り手は、聴衆の反応を見ながら彼らの経験を直接制御す
ることができるし、ゲームデザイナーは、プレイヤーの反応を想像することによって間接的
に制御することができる。おもちゃのデザイナーは、まったくそんなことはしない。

インタラクティブ性の重要性

インタラクティブ性が重要だということには、いくつかの理由がある。ひとつには、インタラクティブ性がゲーム中のイベントに社会性、人間関係といったものを持ち込むということである。その結果、ゲーム中の障害が単に技術的なものから人間関係を反映したものへと変化する。ルービックキューブの色を揃えるのは、単に技術的なパズルに過ぎないが、チェスの対戦は人間的なものである。前者は単に状況からのみ作られた論理パズルを解いているのに対し、後者では対戦相手の心理を読むといった要素が含まれているのである。

第二に、相互作用は受動的な挑戦を能動的な挑戦へと変化させる。同じパズルは何度解いても同じものだが、対戦相手のいるゲームは、相手の手によって毎回変化する。この差をプレイヤーの感情面から考察してみるのは非常に興味深い。パズルを解こうとするものは、パズルのデザイナーが隠したそのパズルの鍵となるポイントを見つけようとして、考え、推理し、神に祈る。心の中では、パズルの解答者はそのデザイナーと戦っているのである。そのため、いったん、その謎が明らかになってしまえばパズルはその魅力を失ってしまう。一方、ゲームはプレイするたびに違ったクリアすべき障害を与えてくれる。パズルは不変だがゲームは変化する。プレイヤーはそのときどきに、自分と対戦相手の両方のことを考えて最善手を見つけなくてはならない。ゲームとパズルの最大の違いは、パズルではあらかじめデザイナーが準備している解法を見つければ良いのに対して、ゲームでは自分で作り出さなくてはならないということである。ゲームはプレイヤーの存在を認め、その手に対応する。一方、パズルは死んだ魚のようにただ解かれるのを待っているだけなのだ。

コンピュータゲームで人間の対戦相手がいることはあまりない。そのため、他のゲームと比べると社会的側面は少なくなっている。しかしコンピュータは、対戦者のふりをする疑似的人格を演ずることができる。この部分こそ、コンピュータゲームが最も優れており、また、いまのところ最も開発が遅れている部分である。最もコンピュータに社会性を持たせることが可能かどうかという点さえ目をつぶれば、コンピュータはプレイヤーにたやすくインタラクティブな体験をさせることができる。なにしろ、プレイヤーの動きに、即座に確信を持って対応することができるのだから。

インタラクティブ性の本質

インタラクティブ性は、あるかないかで表せるような二分木的な概念ではなく、どの程度インタラクティブであるかということの評価すべきものである。パズルには、ほとんど（あるいはまったく）インタラクティブ性は存在しないが、ゲームはそれに比べてずっと人間同士の相互干渉が多い。言い替えれば、インタラクティブ性の大小を、「ゲーム度」(gaminess)のパラメータとして評価しようということである。ゲームの中には、ブラックジャックや、コンピュータテニスの『ポン』(PONG)のように、プレイヤー間の相互作用がほとんどないものもある。プレイヤーがインタラクティブ性を求めたとしても、それは本当に限られている(ブラックジャックでは、カードを引くか勝負するかのどちらか、『ポン』では、ゲームを走らせてパドルを動かすことのどちらかしかできない)。このようなゲームでは、あの手この手でプレイヤーに対することで、プレイヤーの能力すべてを要求するというわけにはいかないのである。その他のゲーム、ブリッジやアメリカンフットボール、『リージョネア』(LEGIONNAIRE: アヴァロンヒル(Avalon-Hill Game Company)の商標)などでは、明らかにもっと多くのプレイヤー間の相互作用が存在する。プレイヤー同士、さまざまなレベルで勝負しようるのである。インタラクティブ性の少ないゲームは一般に飽きが来るのが早く、インタラクティブ性の多いゲームのほうが面白いことが多い。

ここで注意しないといけないのは、ここでいうインタラクティブ性というのは、ゲームのメ

カニックな部分ではなく、むしろそれを見たときの印象というべきものであるということである。『ポン』が無味乾燥なものに見えるのは、私が跳ね回るドットに感情移入できないからなのだ。ブリッジが『ポン』よりも勝れているのは、そこにはペア同士のチームワークやトリックといったインタラクティブな要因が含まれているからなのである。私は、ブリッジにならぬのめり込める。このようにインタラクティブ性は「ゲーム度」をあらゆるパラメータといえるのである。

対立関係

すべてのゲームに存在する第三の要素は、対立関係である。対立は、ゲーム内でのプレイヤー間の相互作用により発生する。個々のプレイヤーはゲーム中、各自の勝利を目指す。一方で、ゲームシステムが設置した障害が、プレイヤーが簡単にゴールに入ってしまうのを阻止する。この障害が静的でプレイヤーの行動に反応しないものなら、それはパズルか運動競技に分類されるべきものである。一方、動的でプレイヤーの動きに反応して動く障害を設けるためには、知性を持ったものがその障害を動かす必要がある。意識してプレイヤーがゴールに入ることを邪魔しようとする者がいるなら、そこに対立が生ずる。このように、対立関係は、すべてのゲームに共通するものなのである。

対立関係のないゲーム

世の中には対立関係というものを嫌う人々もいて、対立のまったくない「おきれいな」ゲームをデザインしようとする試みも何度となく行われた。そういったゲームの中では、プレイヤー間の対立関係よりもむしろ協調関係がクローズアップされている。しかしながら、そういったゲームで売れたものはほとんどない。つまり、対立関係のないゲームはつまらないということなのだろう。

それ以上に重要なのは、対立関係を排除した時点で、すでにそれはゲームではないということだ。対立関係を排除しようとするれば、必然的にプレイヤーの行動に反応して何かをすることができなくなってしまう。反応をさせることができなければ、当然インタラクティブ性もなくなるのだ。対立関係のないゲームはゲームではないのである。

もちろん、ゲームから対立関係を排除することは不可能だといっても、その対立関係を少しずらすことで、ゲームに協調関係を入れることはできる。チーム内のメンバーは、別の対抗者と対立しながら、互いにチームメイトとして協調することができる。ここでいう対抗者とは、対立チームでも、強力な個人でも、コンピュータが担当しているプレイヤーでも良い。いずれにせよ、対抗者は、プレイヤーたちには一個の人格を持ったプレイヤーとして感じられなければならない。少なくとも、プレイヤーの行動に対して手応えのある反応を示しているとプレイヤーに思わせることができなければ、ゲームは崩壊してしまう。

このような「どぎつい」対立関係がゲームで描かれるのには、社会的な背景がある。現実社会での対立関係は、間接的で長きに渡り、法律でがんじがらめに規制されている。それに加えて、現実社会での対立関係がプレイヤーの完全勝利といった形ですっきりと決着がつくことはほとんどない。たまに小さな勝利は味わえても、決着はつくことなく、対立関係はずっと続くのである。ゲームは現実世界の一面であるから、現実世界での何かを強調してそこにプレイヤーの意識を集めようとする。ゲーム中での対立関係が、多くの場合（必ずしも必要というわけではないのに）暴力的で直接的になりやすいのはそのためである。暴力はゲームにとって不可欠なものではない。ゲームに暴力的傾向が出てきやすいのは、それが対立関係を最も直接的かつ明示的に表現する方法だからである。

対立関係に関するまとめ

対立関係は、すべてのゲームにおいて不可欠な要素である。直接的であれ間接的であれ、暴力的であれ非暴力的であれ、それはすべてのゲームの中に存在する。

安全性

対立関係があれば、そこには危険が生ずる。どちらかが傷つくという危険である。だからといって、ゲームの中で本当にけがをさせるわけにもいかない。そこで、ゲームにおいてはプレイヤー同士を空想世界の中で対立させ、危険を味わわせることで、実際に肉体的にぶつかり合うことの代わりにするのである。一言でいえば、ゲームとは現実世界の危険を安全に味わう手段なのだ。正確を期すなら、ゲームの結果は、そのゲームがモデル化しようとした現実と比べて、常により安全なものなのである。モンスターを一日中倒しまくっていたところで、プレイヤーが失うものはたかが100円玉が何枚かである。さっきまで巨大企業を切り回していたプレイヤーが、あっという間に破産に追い込まれたとしても、彼女は子豚の貯金箱を割る必要はない。強大な軍隊を国歌の浮沈がかかった絶望的な戦争に投入しても、一滴の血も流れない。無常な因果応報、次々と来る悲劇、堪えがたい結果。皆、ゲームの中ではよく使われる題材である。

このことは、ゲームの結果は重要ではないということの意味しているわけではない。ゲームに負けるかもしれないという思いは、ときにゲームをやりたくないという感情へとつながる。他人に負ければ、多かれ少なかれ名誉を失うのだ。この点ではコンピュータゲームのほうがましだろう。コンピュータに負けるほうがまだしも恥ずかしくはない。敗者は、面目を保ったまま、何度でも再挑戦できる。それに加えて、コンピュータを完全に打ち負かすような完全な勝利は普通不可能なので、負けてもまあいいかという気持ちになりやすいのである。

敗者に課せられた別のペナルティとしては、そのゲームに勝っていたら貰えたかもしれない報奨を失うということがあげられる。多くのゲームにおいて、勝利したときに得られる報奨は、負けたときに失うものよりも大きい。すなわち、負けた者を打ちのめすというよりも、勝者をほめたたえる構造となっているのである。敗者が失うものは、ゲームに参加するためのチップのようなものでしかない。これは、通常、非常に安価で、敗者に対するペナルティというよりは、ゲームを開催するためのサービス料というべきものだろう。

ゲームの安全性というものを考える上で、ギャンブルは少々厄介な問題となる。ギャンブラーたちは、彼らの現金や持ち物を、完全にランダムかあるいはそれに近いような現象に賭ける。敗者は掛け金を失い、勝者はそれをごっそり手に入れる。そう、ギャンブルはギャンブラーたちを実際に破産の危機に誘う可能性があるのだ。しかしながら、ここでふたつの例を考えておかねばならない。第一の例は趣味のギャンブラーで、彼らは本当に小額しか賭け事には使わない。第二の例は、確率の法則を無視したがるギャンブラーである。彼らは、確率をコントロールできるという幻想に耽っている。ダイスを振るときの気合いのかけかた、スロットマシンのレバーの握り具合、そんなもので勝敗が左右できると信じているのだ。つまり、こういうことになる。趣味のギャンブルは、本道からは離れているとはいえゲームの仲間ということが出来るだろう。しかしながら、本当のギャンブル、大金が動き趣味よりも金儲けが目的となるようなギャンブルは、ゲームとの境界線の外にあると考えるべきなのだ。

ギャンブルの一例ではあるが、ポーカーに関しては特に考察しておく価値があるだろう。

ポーカーは、ブラフ、はったりをいかに使うかというゲームである。ポーカーで勝つためには、対戦相手に自分の手札を現実よりもより良く、あるいはより悪く見せなくてはならない。お金がかかっているからこそ、プレイヤーは懸命に対戦相手をだまそうとする。他のギャンブルでは単なる賞金に過ぎない掛け金が、ポーカーにおいてはそのゲーム構造の根幹に関わっている。この点を考慮すると、ポーカーもゲームの一種とすべきだろう。

安全性に関するまとめ

ゲームは、現実を仮想体験するための安全な方法を与えてくれる。例外は多々あるけれども、基本的にはゲームは安全である。この章では、私が考える「ゲーム」に共通するいくつかの特徴について論じてきた。この章では、かなりの部分を使って、プレイヤーの動機よりも、ゲームの本質とは何かということを強調してきた。しかしながら、このようにプレイヤーとゲームの関係を無視して考えていくのは、不自然であるし誤解を招きやすい。プレイヤーとゲームは切っても切り離せないのである。次章では、プレイヤーにスポットをあて、その動機について考察して行こう。

[目次に戻る](#) | [各章へ: 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9](#) | [次章に進む](#)

空想世界の探求

反社会的行動

優越性の誇示

社会の潤滑油

能力の維持

認識の欲求

動機に関するまとめ

動機と選択の違い

ゲーム性

感覚的満足感

個別の嗜好

ゲームには、ふたつの要素が必用である。プレイするゲームそのものと、それを遊ぶプレイヤーだ。ゲームデザイナーの目的はゲームの制作であり、当座の関心がゲームそのものに向けられるのは仕方がないことだろう。しかし、プレイヤーにゲームを楽しんでもらったり、ゲームから何かを学んでもらい、啓発したりすることこそが、ゲームデザイナーが目指すものなのである。当然、人間のプレイヤーこそ、ゲームデザイナーの最終的な関心事となるべきなのである。人はなぜゲームをするのか？彼らを動機付けるものは何か？ゲームを面白くするのは何か？これらの問いに答えていくことは、良いゲームデザインを目指す上で極めて重大である。

この問題に取り組むひとつの方法は、ゲームの歴史を調査することである。現在のゲームは、あまりにも多彩で複雑になってしまい、ゲームとは何かという明確な解答を示す目的には向いていない。歴史をゲームの原初にまで遡ることができれば、ゲームの根本的な性質は、おそらく、より明白なものとなるだろう。どれくらい遡る必要があるのだろうか？大恐慌の間に作られた『モノポリー』(MONOPOLY)までだろうか？いや、トランプはもっとずっと昔から遊ばれていた。それどころか、ツタンカーメン王の墓の発見者は、その財宝の中から、規則的な区切りが書かれた木の板を見つけ出した。これは何らかのボードゲームと思われる。けれども考古学さえ、我々を十分な過去へ連れて行ってはくれない。ゲームの起源まで溯ることを望むのならば、考古学者の守備範囲を越えて古生物学者の守備範囲に行かなくてはならない。最古のゲームを見つけるためには、何千年どころか何百万年も溯らなくてはならないのだ。ゲームは、歴史どころか人類のすべてに先行するのだから。そう、ゲームを発明したのは人間ではないのである。

幸い、実際に古生物学を学ぶ必要はない。動物園へ出かければ十分だろう。そこで我々は、二匹のライオンの子どもが、母親の近くで取っ組み合っているのを見つける。彼らはどうなりながら、お互いを爪で引っかいたり、噛んだり、蹴ったりする。ある子どもは、歩き回ってチョウを見つけると、草の中にかがんで、その獲物に向かってゆっくり忍び寄り、腰をもち上げてぴくぴく動かし、そして飛びかかる。いかにもユーモラスな情景を見て、「ほら、ライオンの赤ちゃんがゲームをしているよ。楽しそうだね。のんきでいいねえ」などと言うのだ。

第一の点は正しい。ライオンの子どもたちは、本当に一種のゲームをしているように思われる。それらの行動の中には、確かに、第一章で述べた四つのゲームの基本特性（世界の再現、インタラクティブ性、対立関係、安全性）のすべてが見られる。第二の点もおそらく正しい。もっとも、ライオンが本当に楽しんでいるかどうかわかる者など、どこにもいないだろうが。

しかし、最後の点は大間違いである。ライオンの子どもたちはのんきではない。彼らは、子ども時代を気ままに過ごすためにゲームをしているのではない。これらのゲームは極めて真剣な仕事である。彼らは狩りの技術、生き残りの技術を勉強している。獲物に見つからずに接近する方法、飛びかかりかた、反撃を受けずに獲物をつかんで殺す方法といったものを学習しているのだ。彼らは実際の行動によって、しかし安全な方法で学習している。失敗しても良いように、ヌーの角ではなく、チョウや兄弟を相手にするのである。

このように、ゲームは最も古来の、由緒ある教育手段である。自然淘汰の中で承認されてきた、天然の教育技術である。母親ライオンは、黒板を使って子どもたちに講義したりしないし、高齢のライオンが子孫のために自叙伝を書いたりもしない。これを踏まえると、「ゲームに教育的価値はあるか」という質問は、質問そのものがナンセンスなのである。新奇な概念、試されていない一時的流行、伝統の違反者であるのは、ゲームではなく学校のほうである。ゲームは、学習能力を持つどんな生物にとっても、極めて重要な教育機能なのである。

どんな動物がゲームをするのかということを考えてみると、また有益な情報が得られる。ゲームと思われる行動は、哺乳類および鳥類においてのみ観察された。それ以前の系統（魚類、昆虫、両生類、爬虫類）には、ゲームの兆候は見られなかった（オックスフォード大学出版局、ロバート・ファージェン (Robert Fagen) の『Animal Play Behavior』参照）。ゲームは、我々が脳の大きさと関係があるのではないかとあいまいながら考えている知性、学習能力といったものと結び付けられるように思われる。この一致は偶然ではない。ゲームは明らかに、多くの生物が成長するための重要な要素なのである。

普通、ゲームというと子どもたちのものと思われる。それどころか、活動としての「遊び」はほとんど子どもたちの特権的領域であると考えられており、遊びという言葉は成人に対しては軽蔑的に、あるいはふざけて用いられる。子どもたちがゲームをすることを期待されるのは、我々が、ゲームが教育の道具として有益だということ（おそらくは無意識に）認識しているからである。子どもたちが大きくなるにつれて、文化的圧力は変化し、より真剣な活動ができるように、ゲームをする時間を減らすことを期待される。

私は、すべてのゲームの基本動機は学習であると主張する。これはゲームの根源的な動機であって、ゲームの重要性の多くがここにあるのは確実である。この主張は、コンピュータゲームが新しい芸術形式を形成するという主張とは矛盾しない。たとえば、人間と食べ物を考えてみよう。食べ物を食べる基本動機は栄養摂取の欲求である。だからといって、栄養摂取とは無関係のテーブルマナーや、食前の祈りの儀式、調味料、付け合わせで食べる、などといった装飾的な活動が制限されるわけではない。これは食べ物が芸術形式であるという意味ではなく、我々人間だけが、本来の目的を否定せずに、そこから離れた活動をすることができるということである。

では、すべてのゲームの基本動機が学習であるという主張を吟味して行こう。まず、学習という動機は必ずしも意識的なものではなく、意識的には、ゲームが好きだというあいまいな感情が主たる動機となっているという問題がある。しかし、動機が無意識のものだからといって、それが重要でないということにはならない。むしろこの事実は、学習が真の基本動機であるという主張に説得力を与えるだろう。

次に、ゲームの動機には、学習とほとんど関係のないものが多く存在するということについて考えてみよう。ときには、それらの副次的な動機が、局所的には学習という基本動機よりも重要である場合がある。それらの動機には、以下のようなものが含まれる。空想世界の探求、反社会的行動、優越性の誇示、社会の潤滑油、能力の維持、認識の欲求の六つである。

それぞれ順番に見ていこう。

空想世界の探求

ゲームの動機で非常に重要なのが、空想願望の充足である。ゲームは、映画や本や音楽と同じように、プレイヤーを低俗な現実世界から離れさせ、悩みを忘れさせてくれる空想世界へと連れて行くことができる。ゲームは、プレイヤーの直接的な参加を許すという点で、伝統的な逃避の手段（映画、本、音楽）よりも潜在的に優れている。ただ映画を観たり、本を読んだり、音楽を聴く代わりに、ゲームには能動的に参加することができる。プレイヤー自身がゲームをコントロールするようなことは、従来の受動的なメディアでは絶対に不可能なことである。この逃避の願望を充足してくれる空想は、確実に重要な動機である。

空想願望の充足は、しばしば象徴的な探求という形式をとる。現実世界では、十分に刺激的な事物や人、場所はなかなかない。我々の大部分は、アスファルトとプラスチックと紙で作られた世界に閉じ込められている。多くの芸術形式は、別の世界へと観客を送り込み、日常生活では得難い経験や感動を与えようと試みるのである。

たとえば、ディズニーランドがなぜ成功したのかを考えてみよう。ディズニーランドは疑う余地なく、そのジャンルの中で最も成功している。ディズニーランドのような施設は、しばしば「アミューズメントパーク」あるいは「テーマパーク」と呼ばれる。こういった用語は紛らわしい。なぜなら、ディズニーランドが成功したのは、その娯楽性のためだけではないからだ。個々の要素は技術的に優秀ではあるが、その点ではその点では他のアミューズメントパークも、ものすごい乗り物をこれ見よがしに並べている。ディズニーランドの成功理由は一語で現せる。「空想」だ。ディズニーランドは、空想世界を作って、その存在を支援している。ディズニーランドという空想世界の存在を信じさせるためにパークすべての活動がある。訪問者は、パークに足を踏み入れた瞬間、別世界に入ったように感じる。道標、壁、窓、手すりにまで及ぶ細かな気配りが、その世界を信ずるに足る環境を作ったのである。

空想世界の探求は、人間の遊びの重要な要素である。それは我々の気晴らし、そして芸術とゲームにとっても不可欠なものである。

反社会的行動

ゲームの共通の機能は、社会的制限を、せめて空想の中で克服する手段を提供するということである。多くのゲームは、プレイヤーを、現実では社会的に受け入れられることのない、海賊や盗賊といった役割に置く。この端的な例は、Automated Simulations社の『クラッシュ・クランブル・アンド・チャンプ』（CRUSH, CRUMBLE, AND CHOMP）である。このゲームで、プレイヤーは、お気に入りの街でのしのしと暴れ回る、1950年代風の怪獣の役を割り当てられる。パトカーを踏みつけたり、ビルを押しつぶしたり、ヘリコプターを叩き落としたりして、街を大混乱に陥れるのである。このゲームの箱には恐怖におののく市民たちが逃げ惑う中、いまにもIRSのビルをたたき壊そうとしている怪獣が描かれている。こんな反社会的行動もあくまでゲームの中であって、市民もプレイヤーも安全だということで許されているのだ。

ときには、プレイヤーの役割そのものは社会的に受け入れられるが、その行動が現実では歓迎されないものである場合もある。『モノポリー』は、米連邦通商委員会 (Federal Trade Commission) が「略奪的な貿易慣行」 (predatory trade practices) と呼んでいる行為に携わるように仕向ける。ウォーゲームは、戦争を起こしてそれに勝利するようにけしかける。また、

性的な内容を扱い、プレイヤーが実世界ではできないような行動にふけることを許すゲームもある。

この反社会的行動の最も印象的な例は、アーケードゲームに見られる。暴力を強調するゲームだ。アーケードゲームにおいて、何者かを破壊するというテーマは、至るところに存在する。とどめの一撃は、控えめだったり上品だったりすることは絶対になく、犠牲となる敵キャラは、思い切り派手にアニメーションする爆発とともに殺される。サム・ペキンパー (Sam Peckinpah) の映画のように、暴力は極めて重要な要素であり、目的そのものなのである。さらに、我々が不快な感情を抱かないように、外装を工夫することで、暴力の不快な面を巧妙に覆い隠すようになっている。我々は決して、決して人間を殺してはならない。その代わりとして我々は、醜悪な宇宙のモンスターをやっつけるのだ。モンスターたちは星間の重大な罪を犯したことになっていて、プレイヤーは、防御者、護衛者、復讐者といった役割を当てられる。しばしば見られるのは、「人類の運命は危機にある！！」というような緊急事態を表現するケースである。こういう設定はプレイヤーの緊迫感を高めるし、その上好都合なことに、罪の意識なしに暴力を振るうことをプレイヤーに許し、極端な暴力の使用を正当化してくれる。プレイヤーは、社会的な非難のリスクを背負わずに、暴力と大量殺人に携わることができる。ゲームは、反社会的行動を行う安全な方法を提供するのである。

優越性の誇示

ゲームのもうひとつの機能は、能力の優越性を誇示する手段を提供することである。すべてのゲームは、多かれ少なかれこの機能をサポートしている。多くのゲームサークルが、トーナメントとかプレイヤーの成績評価といったものを提供する。アーケードゲームは、ハイスコアを収めたプレイヤーのイニシャルを記録し、それを表示することによって、この機能をサポートする。これを極端に推し進めるプレイヤーも存在する。彼らの主目的は、ただ単に勝つことではなく、誰かを、それもなるべく倒す価値のある相手を打ち負かすことである。チェスなどには、こういった「サメ」たちが非常に多く集まる。ウォーゲームも然りである。ウォーゲームのプレイ中に尋ねられる、良くある質問がある。「君は楽しむためじゃなくて、対戦相手を負かすためにゲームをしているの？」というものだ。このようなプレイヤーは普通、彼らのスキルが直接勝負に反映されるような、偶然性が最小限に押さえられたゲームを好む。

このようなプレイヤーが集まるのは、推論的なロジックゲームだが、実はほとんどすべてのゲームがサメたちにとっての餌をまいている。サメの餌とは、純粋に楽しむためにゲームをするプレイヤーのことである。サメが重大な報酬（たとえば社会的権利）のためにプレイし、そして失敗のリスクが重大なものであるとき、安全性の要素はゲームから排除され、そのゲームはゲームでなくなる。闘争そのものになるのである。

すべてのゲームが過度に競争的な方法でプレイされる可能性を持っているために、闘争としてのゲームの社会的リスクにとりわけ敏感な若干の人々は、ゲームをすることを拒否する。なぜなら、彼らはゲームが安全であると考えないからである。彼らがゲームをするときは、純粋な運のゲームを好む。サメを封じ込めるとか落胆させるとかいうよりは、むしろ、勝利はプレイヤーの能力に明らかに無関係であるという状況を示すために。勝利がまぐれにすぎないのであれば、社会的リスクは排除され、安全性が復活するわけだ。

勝利の唯一の決定要素が純粋な運ではないようなゲームを、絶対的な安全性を確保して（すなわち、サメのつけいるスキがないように）デザインすることは不可能である。どんな面であれ、ゲームの勝敗に個々の能力が影響を及ぼす可能性があるのなら、その結果が個人の能

力の優劣を反映したものとして受け取られるのも、ある意味当然だからである。ゲームにおける社会的リスクからはプレイヤーの態度によって避けられることが多い。その態度とはすなわち、「これはただのゲームなのだ」という自覚である。

社会の潤滑油

ゲームは、特に大人にとっては社会の潤滑油として用いられることがある。その場合、ゲームそれ自体は、プレイヤーにとってそれほど重要な物ではなくなる。本当に重要なのは、社交的な夕べを醸し出す余興としての機能である。トランプや、いくつかの軽いボードゲームがこの機能を果たす。このような「社会の潤滑油」としてのゲームの良い例は、大きなプラスチックのボードを利用するゲームである。ボードは約1.2メートル四方の大きさで、さまざまな色のマークがついている。それぞれのプレイヤーの順番がくると、ランダムに、四本の手足のどれをどのマークに置くべきかが決定される。彼らは必然的に、悪意なしに、ユーモラスな方法でお互いの体と触れ合うことになる。社会との交流が、それによって促進されるわけである。

能力の維持

能力の維持は、ゲームのまた別の動機である。それは頭脳的なものであっても、肉体的なものであっても、その両方の組み合わせであっても構わない。ゲームは、頭脳的能力や肉体的能力を維持するための面白い方法だ。認識力を働かせることを好むプレイヤーもいれば、一方で直感力の使用を好む人もいる。運動能力を鍛えることを好むプレイヤーもいる。さらに、プレイヤーたちは、各々に合ったレベルの課題を必要とする。チェスのプレイヤーがまるばつをしても、簡単過ぎてほとんど鍛錬にはならないだろう。同様に、まるばつを難しく感じるような人がチェスをして、これも良い鍛錬とはならない。このような理由から、プレイヤーたちはそれぞれ、各々にとって有用な、違ったゲームに取り組むのである。

認識の欲求

我々は皆、他者から認識されたいという欲求を持っている。我々が切望する認識とは、単に存在を認められることではなく、個性を認められることである。たとえば我々が知人に会うとき、普通は、「やあ、ジョーンズ」というような気のない挨拶をされる。それよりも、たとえば「やあ、ジョーンズ。ひざの調子はどうだい?」というように、特定の個人として認められた挨拶のほうが、我々は嬉しく感じる。

ペットの人気の高さは、この「認識の欲求」のもうひとつの好例である。いったいなぜ我々は、エサや衛生管理などを必要とする動物を、わざわざ家に置いておくのだろうか。それは、ペットが我々を認識してくれるからだ。我々は、ペットと相互作用することができる。話しかけ、ともに遊び、感情を表に出すことができる。犬は、とりわけ応答的な動物であり、我々の表情を読んだり、声色を解釈したりすることができる。微笑みかければしっぽを振ってくれるし、声をかければ、跳んだり、なめたり、吠えたり、その他の愛情表現をしてくれるだろう。金魚などは、それと対照的に、飼い主を認識したり、感情を表現したりはしてくれない。したがって、金魚のほうが世話をするのはずっと容易であるけれども、たいいてい人は、ペットとして犬のほうを好む。人々は、外界による認識という要素を、それを得るのに必要な努力と比べて、十分に高く評価するわけである。

これは、なぜインタラクティブ性がゲームにとってそれほど重要であるのかという理由のひとつである。ゲームは、二人のプレイヤーがお互いに認識しあうことを可能とする。本当に

優れたゲームには、プレイヤーの個性の大部分が現れてくる。そのようなゲームは、個々のプレイヤーによってもまったく違った様相となるだろう。対戦相手は、ゲームを通して、相手の頭のよさ、早急さ、不正直さ、その他すべての性格を認識することになる。ゲームが終わったとき、対戦相手は互いにゲームを始める前よりも、お互いのことを良く知っていることだろう。

動機に関するまとめ

さまざまな要因が、ゲームの動機として働いている。根源的な（そしてほとんど本能的な）動機は学習であるが、他の要素もそれに続いて、同様に動機として考えることができる。

動機と選択の違い

人々がゲームをする第一の動機と、実際にどのゲームをするか選択する際の要因とは、区別するように気をつける必要がある。つまり、「人はなぜゲームをするのか」という質問の答は、「ゲームをより面白くする要素は何か」という質問の答からは大きく異なりうる。ゲームをする動機を与える要因はいくつかあるが、その人が特定のゲームを選択する理由はまた別にあるのである。たとえば、感覚的満足感とは、このような選択にかかわってくる。特定のタイプのゲームをすることに決めたプレイヤーは、下手なグラフィックのゲームよりも美しいグラフィックのゲームを好むだろうが、グラフィックだけでは、多くの人々にとってゲームをする動機とはなり得ない。ある動機によってゲームをやろうと思ひ、その後、好みによって特定のゲームを選択するのである。

動機と好みを区別するといっても、そのふたつの間に相関関係があることを否定するわけではない。動機を満たすような経験を提供しないゲームを楽しむことはできないのは明らかだ。すなわち、好みと同じようにゲーム選択の決め手となる動機というの、すべてではないが存在するのである。もしプレイヤーが、頭脳的な能力の維持という動機のためにゲームをするのなら、そのプレイヤーは恐らく、他のゲームよりも多くの頭脳的な課題を提供するゲームを、より好むだろう。ゲームに含まれる要素がプレイヤーの動機を満足させないとすれば、そのゲームは面白いはずがないのだ。さて、動機に含まれることのない、ふたつの好みの要因は、ゲーム性および感覚的満足感である。

ゲーム性

ゲーム性 (Game play) は、とりわけアクションゲームにおいて決定的な要素である。この用語は数年にわたって使われてきたが、その意味について明確な意見の一致を見ることはなかった。ゲーム性はプレイヤーのインタラクティブ性に関係し、優れたゲーム性がゲームの成功に欠くことができないということについては、皆が同意するところである。しかしながら、その意味のニュアンスは幅広く、まさに十人十色だ。ゲーム性という用語は、そのあいまい性のために、記述的な価値を失っているわけである。そこで私は、ゲーム性という用語の、より正確で、限定された、そして（私が望むに）いっそう有用な意味を提案する。ゲーム性という捉え難い特徴は、ゲームのテンポと、必要とされる認識力の結合から生ずるものである。『テンペスト』(TEMPEST)のようなゲームは、非常に速いテンポで進行する。他方、『バトルゾーン』(BATTLEZONE)のようなゲームは、はるかにゆっくりしたテンポで進行する。この相違にもかかわらず、両者とも優れたゲーム性を持ったゲームである。なぜなら、それぞれがゲームの要求する認識力に対して適切なテンポを持っているからである。『テンペスト』は『バトルゾーン』よりも、必要とする計画性や概念化がはるかに少ない。プレイヤーに要求されるものは単純で直接的であり、その代わりテンポが速い。『バトルゾーン』

の場合は、より多くの認識力をプレイヤーに要求するが、その代わりテンポが遅い。ゆえに両者は、テンポが非常に異なっているにもかかわらず、おおむね同等のゲーム性を持っているといえる。テンポと認識力が結合し、ゲーム性をもたらすのである。

感覚的満足感

感覚的満足感は、もうひとつの重要な好みの要因である。優秀なグラフィックやアニメーション、サウンドといったものはすべて、ゲームプレイヤーを引きつける。それらは、ゲームの現実感の「証拠」を提供することによって、空想を補助するのである。映画の特殊効果においても、同様の現象を見ることが出来る。映画は、優秀な特殊効果を駆使して、刺激的な面白さを演出する。我々を宇宙戦争のさなかに放り出し、未知の不思議な生き物に合わせ、はるか遠くまで連れて行ってくれる。我々が見るものは、空想世界を信ずるに足るほど真に迫って見えるため、我々は（主観的には）空想世界が真実であると認識する。同様の過程がゲームにも適用できる。特殊効果、グラフィック、サウンド、アニメーション。これらの要素はすべて、良いゲームとそうでないゲームを区別するための要因となりうる。しかしながらその役割を混同してはならない。感覚的満足感あくまで補助機能であり、中心的な特徴ではない。感覚的演出はゲームあるいは映画によって作られた空想の効果を高めるが、素晴らしいグラフィックやサウンドが、それだけで成果を上げるわけではないのである。空想世界抜きの映画は、綺麗な映像コレクションに過ぎないのだし、空想世界抜きのゲームは、綺麗な映像コレクションがインタラクティブ的になっただけに他ならない。

個別の嗜好

これまでのところ、動機と好みについて、それが個々のプレイヤーから独立している絶対量であるかのように論じてきた。しかし実際は、特定のゲームに対する反応は、個々のプレイヤーの性格に著しく依存する。このような性格の相違を、我々はどのように扱えば良いのだろうか？

この問題に対するひとつの理論的な回答は、ゲームの嗜好を決定する莫大な数の性格の特徴の存在を仮定することである。次に、ゲームの特徴についても同様に仮定し、それと合わせて、ゲームの心理的プロフィールを完全に定義する。そして、万能の「性格計量器」を使って、個人の性格の特徴をすべて測定した上で、同じく強力な「ゲーム計量器」を使って、問題のゲームのすべての特徴を洗い出す。最後に、性格の特徴とゲームの特徴の間で行列演算を行うのである。いつの日か、太陽が赤色巨星になってしまう前には、我々の化け物コンピュータが、その人がそのゲームをどれくらい楽しむことができるかを、数値にして答えてくれることだろう。

このアプローチは差し当たり、思考実験の領域から出ることはないだろう。個別の相違に対処できる、もっと簡単な方法を考えなくてはならない。ひとつの方法は、ゲームプレイヤーのグループを観察して分類し、それぞれのグループに高く評価されたゲームを識別することである。この方法は、ゲーム産業の若さゆえに困難である。我々はいま、広範囲であいまいで、しかも部分的に重なっている、少数のグループしか識別することができない。アクションゲームの支持者、D&Dの支持者、戦略ゲームの支持者というように。ゲームのジャンルは他にもあるが、これ以上細かくしたところで、プレイヤーのグループを定義するための役には立ちそうにない。時間が経ち、研究が進めばきっと、もっと多くの有用な情報が手に入るはずである。

個々のゲームに対する嗜好は、不変のものではなく、人間が成長し、変化するにつれて、そ

の嗜好も同じく変化していく。音楽に喩えることで、この点を説明しよう。

子どもの頃は、皆、いろいろな形で音楽を聴かされる。だが、嗜好がまだ不完全な状態であるため、音楽からの影響はわずかしか受けることがない。簡単な歌に合わせて歌ったり踊ったりはするものの、音楽の表現している感情を十分に理解できていないわけではない。音楽のパワーは、我々の持っている、音楽的表現を感情と結び付ける能力から生ずるのである。この連想能力を身につけるまでには何年も必要とする。それは経験の中で徐々に作られていくものだからである。私と同世代の人々にとって、最初に深い感銘を受けた音楽は、60年代のロックンロールだった。激しいビートや、簡素な主題、短い音符といったものは、青春期の素朴な知性に容易に染み込んでいった。我々はこの音楽を理解することができたのだ。さらに、音楽を聴いて楽しむという行為は、学習経験そのものだった。音楽的経験が拡大していくにつれ、音楽のより複雑な構成要素を学び、いっそう広い範囲の連想能力を身につけた。まもなく我々は、それまでの訓練されていない耳では理解できなかった音楽を理解し、正当に評価することができるようになった。ロックはこの成熟を反映して変化し、あるものはロックに留まり、他の人たちは、ジャズやカントリー、フォークなどに移っていった。私も、他の人たちと同じように、一連のステージの中で、ロックからクラシックに移った。私が進化の道をたどったように、ひとつのステージで学んだことが、次のステージの材料を理解することを可能にしたのである。他の人たちは、各自が最も気に入った音楽表現の分野を探求して学び、彼ら自身の道をたどった。共通しているのは、選んだ方向にかかわらず、音楽的な嗜好が進展したということである。ロックは、我々すべてが共有した広い土台であり、入り口であり、多くの分岐を発生させるがらくた、宝の山だった。

ちょうどロックンロールが音楽の入り口だったように、アクションゲームがゲームの入り口となる。初期のロックンロールと同じように、アクションゲームには幅広い魅力があり、理解することが容易である。そして、人々がゲームに慣れていくにつれて、彼らの嗜好は枝別れしていくだろう。ロックンロールと同じように、アクションゲームもまた消えることはない。それは大衆の進化する嗜好を反映して、変化していく。すでにこの現象は起こっている。初期のアーケードゲームは、最近のシューティングゲームなどと比較すれば、飼い慣らされた猫のようにおとなしい。もし『テンペスト』が1977年にリリースされていたら、その難易度はプレイヤーを脅えさせ、受け入れられなかったはずである。時代が変われば、人々も変わる。アクションゲームはいつでも存在し、常に新しいプレイヤーの入り口となるだろう。しかし大衆はそれをずっと遊び続けることはない。多くの人々が、ゲームの他のエリアを探求し始めることだろう。

人々がゲームをする理由はさまざまである。この章で、私はそれらの動機の種類について簡単に触れた。この問題に対する私の分析は、薄っぺらで不確かなものであって、説得力は強くない。人間は複雑な生物であり、ゲームをする人間の動機を完全に理解することは決してできないだろう。だとしても我々は、その重要性を無視することはできない。少なくとも、コンピュータゲームを芸術として極めていくためには、そのことを理解しておく必要がある。

[目次に戻る](#) | [各章へ: 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9](#) | [次章に進む](#)

アクションゲーム	ストラテジーゲーム
シューティングゲーム	アドベンチャーゲーム
迷路ゲーム	D&D
スポーツゲーム	ウォーゲーム
パドルゲーム	運だめしのゲーム
レースゲーム	教育用ゲーム
その他のゲーム	マルチゲーム
	結論

すでにさまざまなハードウェアで、膨大な数のゲームが発売されている。これらのゲームは、途方に暮れるほどバラエティに富んでいる。しかし、その一方で共通点を持つゲームも数多い。ほとんどのゲームが複数の特徴を持っている。発売されている多数のゲームをサンプルとして、コンピュータゲームの分類法を確立することは、ゲームデザインを学ぶために大いに役立つだろう。分類することによって、特定のジャンルのゲームに共通するものは何で、決定的に異なっている部分は何かということが明らかになる。ゲームをうまく分類することができれば、未開拓となっている新しいゲームを提案できるかもしれない。最も重要なことは、分類することによって、ゲームデザインの基礎をなしている原則が明らかになるということである。他の学問を例にあげれば、進化論はチャールズ・ダーウィン (Charles Darwin) がビーグル号において行った細かな分類作業から自然に導き出されたものだ。コンピュータゲームを分類しても、進化論ほど劇的な成果を上げることは期待できないかもしれないが、努力する価値があるのは間違いない。

ここであげる分類法が唯一絶対でないことは強調しておこう。この章で提案する分類が正確なものだとは主張しないし、正確な分類法を公式化することが可能だという主張も受け入れない。分類というのは、単に、多数の関連した対象を整理するための便宜的な方法にすぎない。もしも、何か自然の力や潜在的なプロセスが働いていて、対象のグループを作り出すというのであれば、構造中に根源的な組織化原則を具体化した、唯一絶対の分類法の発見を期待するのも合理的なことではある。たとえば、この地球上の生物の広範な分布は、偶然に発生したものではなく、自然淘汰の産物である。自然淘汰は合理的で理解可能な、非恣意的プロセスである。ゆえに、自然淘汰の原則を具体化する、唯一絶対の、地球上の生物の分類法が存在するわけである。飛行機の形状から空気力学の原則を想像できるのと同様に、生物の分類法から自然淘汰の働きを想像することもできる。

このことはコンピュータゲームには当てはまらない。何らかの組織化原則は存在しているかもしれないが、業界はあまりに若く、サンプルはあまりに少ないため、それを断言することはできない。現在のゲームは、確立した力による必然的な結果というより、むしろ偶然の産物である。もっと、本当の意味で広範なゲームが存在しないことには、ゲームを選択する機会がない。そして、選択なくして自然淘汰はありえない。したがって、唯一絶対の分類法を考案することは不可能なのである。多くの分類法が許容可能である。それどころか、違った分類法を構築しようという試みは、コンピュータゲームの共通の特徴を考察する上で有用な方法ですらある。私はそれほど意欲的ではないので、ただひとつの分類法を提案するだけで満足である。私はコンピュータゲームを、大きくふたつのカテゴリーに分けようと思う。知覚と運動神経を強調する「アクションゲーム」と、認識力を強調する「ストラテジーゲー

ム」である。それぞれのカテゴリの中に、いくつかのジャンルが存在する。

アクションゲーム

アクションゲームは、コンピュータゲームの大部分を占めるポピュラーなカテゴリである。というより、たいていの人は、コンピュータゲームといえばこれを思い浮かべるだろう。アーケードゲームは全部アクションゲームだし、アタリ2600 (ATARI 2600) のゲームも、ほとんどすべてがアクションゲームである。このカテゴリの特徴は、リアルタイムであること、グラフィックやサウンドによる演出があること、キーボードよりジョイスティックやパドルでの操作に向いていることなどである。プレイヤーに要求される主な能力は、手と目の共同作用と、反射神経である。

私は、アクションゲームを、以下の六つのジャンルにまとめてみることにする。シューティングゲーム、迷路ゲーム、スポーツゲーム、パドルゲーム、レースゲーム、そして、その他のゲームだ。

シューティングゲーム

シューティングゲームは、すべて、直接的で暴力的な対決を表現するものである。プレイヤーは、コンピュータが操る敵キャラを、撃って破壊しなくてはならない。要求されるのは、弾を撃ちながら、自機を適切な位置に移動させて、敵の攻撃をうまく避けることである。このようなゲームは非常に人気があり、アタリ (Atari) の強みとなっている。この種のゲームには、視点や武器の違いなどから生ずる、多くのバリエーションが存在する。

『スター・レイダーズ』 (STAR RAIDERS) と『スペースウォー』 (SPACEWAR) を、このような観点から比較してみよう。これらのゲームでは両方とも、プレイヤーはロケットに乗って宇宙を飛んで、敵の宇宙船とリアルタイムの戦闘を繰り広げる。『スター・レイダーズ』は一人称的な視点を提供する (すなわち、画面には、パイロットが見ているのと同じ光景が映し出される)。『スペースウォー』も、同じような武器やメカを使っているが、決定的に違うのは、三人称的な視点を提供することである (つまりプレイヤーは、離れたところから、自機と敵機の両方を見る形になる)。両方のゲームをプレイすれば、その違いは誰にでもわかる。一人称的なゲームは三人称的なゲームよりも、エキサイティングで熱中できるのだ。残念ながら、一人称的視点のゲーム制作は技術的に困難であるため、それを実装しているのはごく少数のゲームだけである。たいていのゲームは、三人称的な視点を採用している。

『アステロイド』 (ASTEROIDS) は、『スター・レイダーズ』と同じように宇宙を舞台とするゲームである。これらのゲームの違いは、敵の性質にある。『アステロイド』の敵は、プレイヤーと同じような武器で武装している少数の知的存在ではなく、ただぶつかってくるしか能のない、多数の岩石である。

『ミサイル・コマンド』 (MISSILE COMMAND) は、いくつかの面白い工夫がされたシューティングゲームである。第一に、プレイヤーは、降ってくる核爆弾から、自分だけではなく街も守らなくてはならない。第二に、プレイヤーは決して敵を攻撃することがない。つまり、純粋な防衛のゲームなのである。第三に、他のゲームでは弾を撃つ動作が非常に速いのに対して、このゲームではより遅く、ミサイルが飛んでいって実際に爆発するまでに時間を必要とする。よってプレイヤーは、先の見通しを立てて多数のミサイルを発射するという計画性を要求されるのだ。したがって、これはアクションゲームでありながら、このジャンルの多くのゲームよりも、多くの戦略的要素を伴っているわけである。

『スペースインベーダー』(SPACE INVADERS: タイトー (Taito America Corp.) の商標) は、これまでで最も成功したシューティングゲームのひとつである。それは、最初の大ヒットゲームであり、1979年に始まったコンピュータゲームブームの火付け役となった。『スター・レイダース』や『アステロイド』では自機は自由に動くことができ、『ミサイル・コマンド』では自分の位置が固定されているのに対して、『スペースインベーダー』では、一次元、つまり左右にのみ動くことができる。プレイヤーは、『アステロイド』のように、自機に接触(着地)しようとして迫ってくる、どちらかという愚かな多数の敵と対決する。さらに、インベーダーは『スター・レイダース』のように、向こうからも攻撃をしてくる。インベーダーは、画面を横切ったり来たりしながら、プレイヤーに向かってゆっくりと降りてくる。プレイヤーが多くのインベーダーを殺すにつれて、それらの動きは加速していく。このことが、ゲームに催眠術的な促進効果を与えているのだ。『スペースインベーダー』はやはり名作である。

『スペースインベーダー』の成功によって、多くの模造品や二番煎じが登場した。これらのうちの多くのゲームの目的は、ただ、オリジナルの成功に便乗して儲けることだけだった。ただし、単なる二番煎じに終わっていない、本物のゲームもいくつか存在する。たとえば『ギャラクシアン』(GALAXIAN: ミッドウェイ (Midway) の商標) は、『スペースインベーダー』の単純なバリエーションだが、このゲームのインベーダーは、オリジナルよりも獰猛で、急加速してプレイヤーに襲い掛かる。『センチピード』(CENTIPEDE) も、同じく『スペースインベーダー』の派生形である。デザイン的にはかなり違っているが、内部的なゲームの構造はオリジナルに酷似している。インベーダーは、いくつもの節からなる巨大なムカデ一体となった。上下左右への移動は、画面のふちによってではなく、画面内に散らばったきのこによって制限されている。多数の脇役(クモ、ノミ、サソリ)が、ゲームをより複雑なものとする。『テンペスト』(TEMPEST) は、ワイヤーフレームを使用して、『スペースインベーダー』を三次元の一人称視点にしたようなゲームである。『スペースインベーダー』のデザインが多くのゲームに参考にされたのは、それだけこのゲームが独創性に優れ、魅力がある証拠である。

このほか、実に多くのシューティングゲームがある。『バトルゾーン』(BATTLEZONE) や『レッド・バロン』(RED BARON) は、ワイヤーフレームを使用した、一人称的視点のシューティングゲームである。『キャヴァーン・オブ・マーズ』(CAVERNS OF MARS)、『ヤーズ・リベンジ』(YAR'S REVENGE)、『クロスファイヤー』(CROSSFIRE: オンライン・システムズ (On-Line Systems) の商標)、『ディフェンダー』(DEFENDER: ウィリアムズ (Williams) の商標) など、皆シューティングゲームである。

なぜこれほど多くのシューティングゲームが、舞台を宇宙空間に設定しているのかということに不思議に思うかもしれない。これには三つの理由がある。第一に、宇宙空間は、コンピュータで描画するのが簡単だということである。ゲームデザイナーは、真っ黒な画面に白いドットを描画するだけで良いのだ。第二に、宇宙空間は、プレイヤーの期待によって妨害されることがない。これはどういうことかということ、ゲームデザイナーがデザイン上の問題に遭遇したとき、それを何か強引な手段で解決したとしても、誰も、それが非現実的であるという理由で異議を唱えることができないということだ。これが地球を舞台にしたゲームだと、非現実的なことをすると目立ってしまうため、ゲームデザインが制約され、そのぶん負担が大きくなるのである。第三に、宇宙空間は、本質的に空想がたくさん詰まった環境だということだ。宇宙が未知の世界であるため、不信を抱きにくいからである。

シューティングゲームは、これまで常にコンピュータゲームの中心にあった。プレイヤーたちは決して、それに飽きてしまうことはないだろう。これからも長い間、存在し続けるに違

いない。

迷路ゲーム

アクションゲームの第二のジャンルは、迷路ゲームである。なかでも成功しているのは『パックマン』(PAC-MAN: ナムコ(Namco)の商標)である。迷路ゲームを特徴付けるのは、プレイヤーが通る迷路である。敵キャラが迷路を通過してプレイヤーを追いかける場合もある。いくつかの迷路ゲーム(アタリ2600の『メイズ・クレイズ』(MAZE CRAZE)が良い例だ)では、プレイヤー自身が出口への道を切り開く。その他の迷路ゲームは、プレイヤーが迷路の各部を通過して移動する。『ドッジ・エム』(DODGE 'EM)が、このようなゲームの良い例だろう。どちらの場合も、追ってくる敵キャラの数や速さや頭の良さが、ゲームのペースと難しさを決定する。『パックマン』は、こういった要素のバランスが非常に良く取れている。敵キャラは、プレイヤーよりも少しだけ遅く設定されており、頭の良さや数でそれを補うようになっている。全体的にハイペースなゲームであり、プレイヤーが自キャラと4匹の敵キャラの位置関係をリアルタイムに把握することを難しくしている。

ひとつ成功したゲームがあると、その模倣品やバリエーション、派生物が次々に作られる。『パックマン』も例外ではない。アタリの家庭用ゲーム機における『パックマン』の最初のバリエーションのひとつに、『ジョーブレイカーズ』(JAWBREAKERS: オンライン・システムズの商標)の初版がある。このゲームは、いまは市場から消えてしまっているが、構造的な変化と演出的な変化の違いを、はっきりと証明している。構造的には『パックマン』と区別がつかない。ゲーム性もほとんど『パックマン』と同じである。しかしながら、演出的には多くの違いがある。追いかけてくる敵キャラは、幽霊ではなく顔であり、プレイヤーは口ではなく歯である。迷路の構成は違うし、サウンドも異なっている。このゲームは、著作権上の問題を最小限にとどめてゲームを模倣するのに使える方法の好例といえるだろう。

『パックマン』の派生物には、『ムスクアタック』(MOUSKATTACK: オンライン・システムズの商標)というものもある。このゲームは、『パックマン』と比べて、構造的にも若干の変更が加えられている。プレイヤーが、迷路の中でコンピュータが操る4匹の敵キャラに追いかけるのは同じだが、多くの装飾による味つけがなされている。まず、迷路のすべてのポイントを通り過ぎただけでは十分ではなく、コンピュータによってランダムに選択されるいくつかのポイントは二回完全に通過しなくてはならない。つぎに、追いかけてくる敵キャラに対抗する方法がまったく違う。パワーアップした自キャラで食べ返すのではなく、ネズミ取りを使うのだ。戦略・戦術的様相が異なるため、まったく違ったゲーム性が生まれている。最後に、とても面白い2人用のゲームが用意されていて、協力、あるいは対戦ゲームとして遊ぶことができる。『ムスクアタック』(MOUSKATTACK)は、『パックマン』の基本的な構造はそのままながら、多くの装飾と拡張を加えることで、別のゲームに仕上がっているのである。

迷路ゲームの魅力は、迷路自身が分岐構造を純粋な形で内包するという特性にある。分岐構造は、すべてのゲームにおいて重要な要素である。読者は、ゲームはツリー構造を持っているということを知っているだろう。第一章で述べた、ツリーの各々の分岐点、プレイヤーによってなされた決断を表すという考え方である。迷路ゲームにおいて、各々の分岐点は、迷路の交差点という形で簡潔に描写され、プレイヤーの利用可能な選択肢は、交差点で通行可能な道として、視覚的に提示される。つまり、迷路ゲームは、ゲームの分岐的な構造の、明確で視覚的な「世界の再現」を提供するのである。

さらに魅力的なのは、迷路ゲームでは、ループする構造が可能だということである。プレイ

ヤーは、迷路において、何度でも交差点に戻ることができる。そのたびに、他のキャラクターは異なった位置に動いているため、選択肢は毎回異なった意味を持つわけだ。このようにして、表示上は少数の交差点が、ゲームのツリーにおいては莫大な数の分岐点を表すことができる。コンピュータのプログラムに喩えると、少数の命令が、ループや条件分岐を通して、多数のケースを扱うことができるということと似たようなものである。

スポーツゲーム

このタイプのゲームは、一般的なスポーツを再現しようとしたものである。これはもともと、コンピュータゲームが独自性を持っていなかった時代に作られたものである。独創的なアイデアを持っていない人がゲームをデザインするのに、そのモデルとしてスポーツを採用したのだ。これはマーケティング的にも有効な方法だった。保守的な消費者は、まったく経験したことのない、えたいの知れないゲームをなかなか買おうとはしない。すでに精通しているゲームのほうが安心できるのだ。というわけで、バスケットボール、アメリカンフットボール、野球、サッカー、テニス、ボクシング、その他のスポーツに基づいたゲームの登場となった。これらすべてのゲームは、ゲーム性を確保するために、元になったスポーツを大きくねじ曲げている。コンピュータゲームが提供するゲーム性は、元のスポーツとはかけ離れたものである。これは良いことだ。なぜなら、スポーツを完全に模倣しようとするれば、それはかえってつまらないゲームになってしまうからである。オリジナルのスポーツを大きく変更することによってのみ、きちんとしたデザインを作り出すことが可能だった。とはいえ、スポーツゲームは、依然としてコンピュータゲームの世界ではマイナーな壁の花である。私は、将来スポーツゲームのデザインはあまり注目されなくなると考えている。コンピュータゲームの独自性が認められている現在、既知の素材を扱う必要性は減少したからである。

パドルゲーム

『ポン』(PONG)に基づくゲームをカバーするために、パドルゲームという名前を使おう。『ポン』は間違いなく、最も成功していて応用の利くゲームデザインのひとつであり、そのため多くの孫やひ孫が存在する。ゲームの核である、パドルで何かを操作して弾を受け止めるという形は、果てしないバリエーションをもって使用された。オリジナルの『ポン』は、二人でプレイする、ピンポン(卓球)の電子バージョンだった。ゆえにその名がある。『ブレイクアウト』(BREAKOUT)は、ボールで壁を破壊する、一人用のゲームであり、破壊したブロックに応じて点数がもらえた。『スーパーブレイクアウト』(SUPERBREAKOUT)はこのテーマを拡張し、動く壁や、追加のボールや、その他のトリックを取り入れた。『サーカス』(CIRCUS)は、放物線状の軌道をとるピエロの姿をした弾や、動く壁(風船)の複雑な軌道を導入した。『ウォーローズ』(WARLORDS)は、いささか遠いジャンルだが、四隅に位置する最高四人のプレイヤーが、フィールドの端で跳ね返った弾から、パドルを操作して自分の城を防御する。

以上のゲームでは、プレイヤーはボールを武器として用いるが、単にボールを受け止めるだけのゲームもある。『アヴァランチ』(AVALANCHE)はそのようなゲームのひとつである。このゲームでは、プレイヤーは画面の一番下におり、上から多数の岩が落下してくる。プレイヤーは、落ちてくる岩を受け止めなくてはならない。ゲームが進むにつれて岩の落ちるスピードが速くなり、ペースが上がるに従い、ゲームは半狂乱的になっていく。『チキン』(CHICKEN)は、このテーマを拡張している。岩を卵に置き換え、地面に落ちると卵がかえってしまい、主人公のめんどりは、動き回るときにそれを飛び越えなくてはならなくなる。

パドルゲームのシステムは非常にシンプルであり、まだ発展の可能性があるかどうかについては疑問があるけれども、このような古くからある普遍的なシステムに死を宣告してしまうのはためらわれるところである。

レースゲーム

直線コースを使ったレースゲームもある。ほとんどの場合、プレイヤーの移動スピードは一定であり、障害がうまく乗り切れないと、時間的ペナルティが与えられる。APXのスキーゲーム『ダウンヒル』(DOWNHILL)では、プレイヤーは木や岩を回避しなくてはならない。プレイヤーのスコアは、コース完走時点のタイムに基づいている。ジェベリ・ソフトウェア(Gebelli Software)の『マッチ・レーサー』(MATCH RACER)は、オイルや障害物のあるカーレースだ。また、『ナイト・ドライバー』(NIGHT DRIVER)は、一人称視点が特徴的なカーレースゲームである。これらすべてのゲームにおけるひとつの問題は、それらが真のゲームではなく、パズルであるということである。というのは、プレイヤーとその対抗者の間にインタラクティブ性がないからだ。それどころか、これらのゲームで対抗者を識別することは困難である。

レースゲームのいっそう進化したバリエーションが、『ドッグ・デイズ』(DOG DAZE)である。これはパズルではなく、真のゲームである。二人対戦式のゲームで、毎回位置が変わるゴールと障害物がフィールドにおかれる。それぞれのプレイヤーは犬を操作する。給水栓が画面上のランダムな位置に出現するのだが、プレイヤーは、その給水栓に先に接触して自分のものにするために、競争しなければならない。相手の給水栓には触れてはならず、触れた場合は一時的に動きが止まってしまう。ゲームは多くの面白いひねりが加えられているが、かといって過度に複雑でもない。これは、レースゲームが、ゲームデザインにおいて融通の利く媒体であることを示している。

その他のゲーム

私の分類には欠陥があり、この分類にうまく当てはめることができない多くのゲームが存在する。たとえば『ドンキーコング』(DONKEY CONG : Nintendoの商標)は、知的な障害物が入り入れられたレースゲームのように見える。『フロッガー』(FROGGER)なども私の分類法では扱いきれない。壁や障害物を動かすことから、迷路ゲームの一種と呼ぶことはできるかもしれないが、適合度は低いだらう。同じく、ブローダーバンド・ソフトウェア(Broderbund Software)の『アップル・パニック』(APPLE PANIC)もまた私の分類を拒絶する。ある意味では迷路ゲームのようであり、またある意味ではシューティングゲームのようでもある。ゲームのペースは中途半端に遅い。私はこのゲームと何と呼んだら良いかわからない。これらのゲームが私の分類法に適していないという事実は、それほど悩むようなことではない。私は、個別のゲームのために特別なジャンルを作ることは望まない。新ジャンルの発展を静かに見守ることにしたい。

ストラテジーゲーム

ストラテジーゲームが、コンピュータゲームの第二のカテゴリーである。これらのゲームは、操作よりも思考を強調する。アクションゲームが戦略性を欠いているというわけではない。いくつかのアクションゲームは、しっかりした戦略的要素を持っている。ストラテジーゲームとアクションゲームを区別するのは、アクション性の強調の度合いである。アクションゲームはすべて運動神経を必要とするが、ストラテジーゲームはそうではない。リアルタイムなストラテジーゲームは希少である(これは変わりつつある。『リージョネア』

(LEGIONNAIRE) は、注目すべきリアルタイム・ストラテジーゲームである)。ストラテジーゲームは、典型的に、アクションゲームよりも多くのプレイ時間を必要とする。ストラテジーゲームはゲームセンターには存在しない。アタリ2600 (ATARI2600) でも珍しい。ほとんどパソコンゲームの独壇場である。私は、ストラテジーゲームを六つのジャンルに分ける。アドベンチャーゲーム、D&D、ウォーゲーム、運だめしのゲーム、教育用ゲーム、マルチゲームである。

アドベンチャーゲーム

アドベンチャーゲームは、最も古いコンピュータゲームのひとつ『アドベンチャー』(ADVENTURE) から始まっている。このタイプのゲームでは、冒険家は、複雑な世界を動き回り、さまざまな道具を手に入れ、障害を乗り越えて、ゴールにたどり着かなくてはならない。スコット・アダムス (Scott Adams) は、パソコンで広く稼動する最初のアドベンチャーゲームを作った。彼のソフトハウス、アドベンチャー・インターナショナル (Adventure International) は、それらのゲームによって有名になった。彼のゲームは、少量のメモリで動作する純粋なテキストアドベンチャーであり、ディスクドライブを必要とせず、違うマシンに容易に移植することができた。その後、ケン (Ken) と ロバータ・ウィリアムズ (Roberta Williams) は オンライン・システムズ (On-Line Systems) を創業し、その代表作となる『ウィザード・アンド・プリンセス』(THE WIZARD AND THE PRINCESS) を発売した。ゲーム自体は特に新しくなかったが、革新的だったのは、グラフィックを使用して、冒険家が見たものを絵で表示することだった。両社とも、そのシステムで多くのゲームを作るために、ラインを拡張した。これらの派生的なゲームの大部分は、構造的にはオリジナルに類似しており、ディテールや規模が異なっている。

アドベンチャーゲームの次のバリエーションは、超大作ゲームだった。『タイム・ゾーン』(TIME ZONE) などがそのひとつである。これらの巨大なアドベンチャーゲームは、多数のディスクを使う。プレイヤーは、あるディスクで謎を解いたら、別のディスクに入れ替えてゲームを進めるわけである。ゲームは構造上、以前のゲームとまったく同じであり、唯一の相違はその規模である。これらのゲームは、解くのに何週間ものプレイを必要とした。

インフォコム (Infocom) の『デッドライン』(DEADLINE) は、アドベンチャーゲームの新しいバリエーションを示した。これは、いくつもの面白いひねりが加えられた探偵アドベンチャーゲームだ。まったくグラフィックを使っていないこと、優秀な構文解析能力を持っているという点で『アドベンチャー』(ADVENTURE) の後継者といえる。プレイヤーは、殺人事件を解決する探偵となる。ゲームの進行がリアルタイムになったことで、ゲームの面白さと難易度が上がっている。プレイヤーは、宝物ではなく殺人事件を解決する手がかりを捜す。このゲームは、アドベンチャーゲームの可能性を示している。すなわち、同じシステムが、筋書きを変えて異なった目的に使うことができるということである。

いままでで最も巧妙なアドベンチャーゲームのひとつが、ウォーレン・ロビネット (Warren Robinett) の『アドベンチャー』(ADVENTURE) である。このゲームは、他のゲームと同じ基本的なフォーマットに従っているが、文章をまったく使わないのが特徴である。その代わりに、シンプルなグラフィックで表現されたいくつもの部屋を行き来する。グラフィックと操作方法は根本的に異なっているけれども、アドベンチャーゲームのシステムの基本的な手触りは一緒である。『スーパーマン』(SUPERMAN)、『ホーンテッド・ハウス』(HAUNTED HOUSE)、『ガラハド・アンド・ホーリーグレイル』(GALAHAD AND THE HOLY GRAIL) などは、すべてこのゲームの派生形である。

アドベンチャーゲームは、ゲームよりパズルに近い存在である。第一章で論じたように、パズルは、プレイヤーに与える障害が静的であるということで、ゲームとは区別される。アドベンチャーゲームは複雑な障害を提供するが、いったん解かれたらおしまいである。何らかの方法でプレイヤーに反応する障害物（飢えたドラゴンなど）を含めれば、少しはゲームに近くなるが、そういう工夫をしても、やはり基本的にはパズルのままである。

D&D

D&Dから、ゲームの歴史の中で完全に独立した流れのスタイルが作られた。ファンタジー・ロールプレイングゲームは、ゲイリー・ガイギャックス (Gary Gygax) によって作られたTSRの『ダンジョンズ・アンド・ドラゴンズ』 (Dungeons and Dragons) に始まった。お城があって、ドラゴンや魔法使いやドワーフがいるおとぎ話の世界で、探検、協力、そして対立といった要素が複雑に絡み合う、非コンピュータゲームだ。D&Dでは、「ダンジョンマスター」の誘導のもと、プレイヤーのグループが宝探しを始める。ゲームに必用なハードウェアは最小限ですむ。プレイヤーがテーブルのまわりに集まって、メモ用紙があれば良い。ダンジョンマスターはゲームのルールをその都度応用して審判をする。ダンジョンマスターはすべてのイベントを判定する権利を持っており、これによって、複雑なルールに悩まされることなしに、複雑なシステムを実現できるのだ。雰囲気は非常にルーズで、くだけた感じである。こういった理由により、D&Dは、果てしないバリエーションと派生形を生み、人気ゲームとなった。

D&Dは70年代半ばに出現した。人々が、それがふたつの重大な限界を持っていることに気付くのに、長くはかからなかった。第一に、ゲームは複数のプレイヤーとダンジョンマスターを必要とするため、一人で遊ぶことはできなかった。第二に、プレイ中に長い計算や多数のダイスを振ることが必用で、ゲームはしばしば長たらくて飽き飽きするようなものになってしまうことがあった。多くの人々が、これらの問題はマイクロコンピュータで解決できるということを知った。D&Dスタイルのコンピュータゲームを最初に利用可能にしたのは、オートメテッド・シミュレーションズ (Automated Simulations) だった。彼らの『テンプル・オブ・アプシャイ』 (TEMPLE OF APSHAI) は非常に成功していた。そのほかにも彼らは、同じようなD&Dスタイルのゲームを多く発売している。

しかし、これまでのところ、本当にD&Dの精神を継いだゲームが発売されることはほとんどなかった。これには、いくつかの理由がある。まず、たいていのD&Dプレイヤーが若者であり、そのような製品を買うための金を持っていなかったということ。次に、アドベンチャーゲームが、D&Dのアイデアの多くをゆっくりと吸収していったことだ。もともと、アドベンチャーは純粋な文章だけのゲームであり、D&Dはグラフィックを使っていた。アドベンチャーはパズルであり、D&Dは真のゲームだった。アドベンチャーは概して非暴力的であり、D&Dは非常に暴力的な傾向があった。だが最近では、アドベンチャーゲームがD&Dの特徴の多くを取り入れており、それらの明確な相違点を示すことが難しくなっている。

この現象の良い例は、『アリババと40人の盗賊』 (ALI BABA AND THE FORTY THIEVES : クオリティ・ソフトウェア (Quality Software) の商標) である。このゲームは、アドベンチャーとD&Dの両方の要素を持っている。プレイヤーは広大な迷路を歩き回って、お姫様を救わなくてはならないが、途中で怪物や盗賊とも戦わなくてはならない。プレイヤーは、アリババとして、D&Dゲームのような能力値（器用さ、素早さなど）を持っているが、一方、アドベンチャーのように迷路を探検しなくてはならない。このため、私はこのゲームをアドベンチャーとD&Dのどちらかに分類することはできない。むしろ「ファンタジー・ロールプレイングゲーム」とでも分類されるべき、新しいジャンルと言ったほうが適当だと思われる。こ

れからは、アドベンチャーゲームの「探索・発見」という面と、D&Dの「敵の撃破」という面を兼ね備えたゲームが、もっとたくさん出てくることだろう。

ウォーゲーム

ストラテジーゲームの第三のジャンルは、ウォーゲームである。ゲームとしての非コンピュータ・ウォーシミュレーションには、長い歴史がある。商用ウォーゲームは、1880年代の、木製ブロックを使うアメリカのゲームまで遡る。イギリスでは長い間、ウォーゲームのマニアたちが、兵隊のミニチュアの模型を使った、非常に複雑なルールのゲームをプレイしていた。それらのゲームはミニチュアゲームと呼ばれ、人気が広がってゆき、いまではアメリカでもプレイされている。けれども、近年、ウォーゲームといえば、ほとんどはボードゲームである。それは、1950年代後期に、チャールズ・ロバーツ (Charles Roberts) によって生み出された。彼はアヴァロンヒル (Avalon-Hill Game Company) を設立し、60年代には、『ブリッツクライグ』 (BLITZKRIEG)、『ワートルロー』 (WATERLOO)、『アフリカ・コープス』 (AFRIKA KORPS) といった名作ゲームを作った。そして70年代になって、新しい会社、シミュレーション・パブリケーションズ (Simulations Publications) が、ボードゲーム形式のウォーゲームを、ウォーゲームの大半を占めるまでに発展させたのである。

ウォーゲームは、一般に売られているすべてのゲームの中で、最も複雑で高度なゲームである。ルールブックは、企業の合併の契約書のごとく熟読しなければならないし、プレイ時間が3時間を超えることもざらである。したがって、ウォーゲームをコンピュータ上に実装することは非常に困難と思われるのだが、にもかかわらず、それは存在する。

現在のコンピュータ上のウォーゲームは、ふたつのグループに分けることができる。第一のグループは、従来のボードゲームをそのままの形で直接コンピュータ上に移植したものである。たとえば、『コンピュータ版 ビスマルク』 (COMPUTER BISMARCK)、『コンピュータ版 アンブッシュ』 (COMPUTER AMBUSH)、『コンピュータ版 ナポレオン戦争』 (COMPUTER NAPOLEONICS) といったゲームだ。これらのゲームは、ある形式のゲームをそのまま別の形式に変換することの愚かさを教えてくれる。良質なボードゲームを真似していても、それ自身がオリジナルと同じくらい良くできているかということ、そうではない。なぜなら、元のボードゲームをそのまま複製しようとするものだから、コンピュータを使っているのに、ボードゲームのように遅くて遊びにくいゲームになってしまっているからである。

コンピュータ上のウォーゲームにおける第二のグループは、ボードゲームをコピーすることにそれほど盲目的ではない。私自身の『東部戦線 1941』 (EASTERN FRONT 1941) が、主にグラフィックや操作性などにおいて、このグループの中でベストなものだと考えられる。このジャンルでは実験的な作品が多く、それゆえ、失敗作のほうが多いくらいである。アヴァロンヒルが最初に投入したウォーゲームも、そのような実験作だった。私の『タンクティクス』 (TANKTICS) は、かつて最も先進的なゲームだった (私が1978年にそれをリリースした当時は、製品では唯一のウォーゲームだったのだ)。いまとなっては平凡なゲームである。コンピュータ・ウォーゲームは、まだ発展途上の分野だといえるだろう。ウォーゲームというと、プレイヤーにとってもデザイナーにとっても、いまはまだあまりにもボードゲームとの結びつきのイメージが強すぎる。しかし、ウォーゲームはいつかボードゲームの呪縛から開放され、独自のアイデンティティを確立し、少しずつ進化していくはずだ。

運だめしのゲーム

運だめしのゲームの歴史は非常に古く、何千年にもわたってプレイされてきた。それをコン

コンピュータ上に作ってみようと思うのは当然である。プログラミング的にも非常に容易であるため、クラップスやブラックジャックを始めとして、この手のゲームには多くのバージョンが存在する。だが、そういったゲームはさほど人気が高いわけではない。その大きな理由は、コンピュータの長所が生かされていないということだろう。それどころか、オリジナルのゲームの利点をも失ってしまっている。これらのゲームは、他のメディアのゲームを不用意に持ってくることの愚かさを証明している。

教育用ゲーム

ストラテジーゲームの第五のジャンルは、教育用ゲームである。すべてのゲームは何らかの意味で教育的であるけれども、ここで取り上げるゲームは、始めから教育を目的としてデザインされたものである。こういうゲームはまだそれほど多くない。それはおそらく、コンピュータを使った教育に関心を持つ人々が、まだあまりゲームデザインに注目しなかったからだろう。ソーンEMI (Thorne EMI) のパズルが、このジャンルの良い例である。また、APXは、いくらかの教育的価値を持った、非常にシンプルな子ども向けゲームのセットを販売している。名作コンピュータゲームのいくつかは、教育用ゲームである。特に『ハングマン』(HANGMAN)、『ハンムラビ』(HAMMURABI)、『ルナ・ランダー』(LUNAR LANDER)の3本は最も注目に値するだろう。『スクラム』(SCRAM: 原子力発電所シミュレーション)と『エナジー・ツァー』(ENERGY CZAR: エネルギー経済学シミュレーション)は、教育用ゲームの中でも特に複雑なプログラムである。私のお気に入りには『ロッキーズ・ブーツ』(ROCKY'S BOOTS: ラーニング・カンパニー (The Learning Company) の商標) というゲームで、ブール論理とデジタル回路を扱った子ども向けゲームである。プレイヤーは、仮想的な論理マシンを作るために、論理ゲートを組み立てる。このゲームは、コンピュータゲームの計り知れない教育的価値の可能性を示している。教育者は、ゲームが子どもたちに与える動機付けの能力に気付き始めている。じきに我々は、もっとたくさんの教育用ゲームを目の当たりにすることになるだろう。

マルチゲーム

ここでは、個人やグループ間の人間関係に焦点を置いたゲームについて考察していこう。うわさを扱ったゲームがある。プレイヤーは、コンピュータが演じる最大7名の他のプレイヤーたちとうわさを流しあうのだ。交わされる会話は、そのときの印象、すなわちそれを良いと思うか悪いと思うかに限られる。抜け目ない態度を取っていれば多数派でいられるだろう。同様のゲームは、会社経営、お茶の間ドラマ、中世恋愛物、外交、謀略といったさまざまな分野を表現できる。また、この分野のゲームは成熟しているとはとてもいえないが、プレイヤーたちに非常に現実感のある空想世界を与えてくれるという点で、重要な分野のゲームとなるだろう。ゲーム以外の多くの芸術分野で、人間関係を表現しようとするのにいかに多くの努力が注ぎ込まれているかを思えば、コンピュータゲームが人間関係を扱えるようになるまでの時間はほんの一瞬だったといえるだろう。

結論

これにて、私の提案する分類法の記述を終える。明らかに、この分類法には多くの欠陥がある。これは主に、雄大な自然法則などではなく、歴史的な偶然を根拠として区分されているためである。どうしてウォーゲームとD&Dを別に扱うべきであるのかという、根本的な理由がないのだ。しかし、両者は別々に発展し、歴史的にまったく別の道を歩んできた。同じように、私が教育用ゲームを独立したジャンルとしたのは、教育者たちが、教育用ゲームを作

るための努力をしたからである。時間が経つにつれ、市場原理の力がそれらのジャンルを確定し、いっそう体系的で隙のない分類法が可能となるだろう。人々が教育用ゲームを作ろうとしたから、それらがいまあるわけだ。私の分類法は寄せ集めにすぎない。なぜなら現在のコンピュータゲーム全体が、寄せ集めのようなものだからである。

この分類法は、コンピュータゲームデザインの状態について、多くの観察結果を示している。たとえば、アクションゲームにおいて、それぞれのジャンルの系譜が極めて少数しかないということは明白だろう。それぞれのジャンルでの典型的なゲームは、模倣、バリエーション、改良の一群全体を生んだ。さらに、それぞれの典型的なゲームというのは、それ自体が大きなドル箱となることはめったになく、それを改良したゲームが、打ち止めとなるまで後に続いた。シューティングゲームでは『コンバット』(COMBAT)が『スペースインベーダー』を導いたし、迷路ゲームでは『ドッジ・エム』が『パックマン』を導き、パドルゲームでは『ポン』が『スーパーブレイクアウト』を導いた。

この分類法のもうひとつの教訓は、ストラテジーゲームが、アクションゲームと比較して、まだ非常に不完全な発展状態にあるということである。アクションゲームにはかなりはっきりしたジャンルがあるのに対し、ストラテジーゲームのジャンルは、それほど満足できるものではなく、それぞれのジャンルの区別もあいまいである。だが、あいまいであるということは、ストラテジーゲームに、よりいっそうの発展の可能性が残っていることを示している。

分類法というものは、組織化しようとする素材の主要な部分を反映する。コンピュータゲームデザインが日進月歩である以上、ここで提案した分類法も、短期間で、時代遅れで不適当なものになってしまうと思われる。数年のうちに、市場の変化を反映して、新しい分類法が作られるに違いない。しかしながら、差し当たりこの分類法は、雑多なゲームを体系的に観察する方法を提供してくれるのである。

[目次に戻る](#) | [各章へ: 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9](#) | [次章に進む](#)

ゲーム環境

コンピュータ

コンピュータゲームデザインの六つの鉄則

鉄則1：適材適所

鉄則2：ちょっと拝借でいいの？

鉄則3：まずは使い勝手

鉄則4：シンプルが一番

鉄則5：少なく貯めて大きく使おう

鉄則6：二足のわらじをはけ

結論

ゲーム環境

すべての芸術表現には、何らかの素材が必用になる。素材の取り扱いに習熟しない限り、芸術家はその作品で自己表現をすることはできない。彫刻家は、自分が使おうとする素材、たとえば、大理石、真鍮、その他もろもろが何に向いていて何に向いていないかを知っていないてはならない。画家は、筆の使い方や光がどう振舞うのかを理解している。音楽家は、どうすれば良い音が出るのかを知っている。コンピュータゲームデザイナーも、素材、コンピュータについて良く理解していないてはならない。コンピュータは、ゲームをデザインする上で素晴らしい可能性を与えてくれるが、一方ではコンピュータを使うがゆえの新たな負担も生じてくる。この章では、コンピュータゲームをデザインする上での可能性と問題点について論じていこう。まずは、いくつか簡単な例をあげて、コンピュータゲームの本質を明らかにしていきたい。

トランプは、非常に単純なゲーム環境のひとつである。環境を構成する素材は、非常にシンプルなものである。裏側には全く同じ模様が、そして表側には一枚一枚別々の模様が印刷された⁵²枚のカード、それがトランプである。その特徴は以下のように表すことができるだろう。

- 1) 枚数が多い。
- 2) 個々のカードは互いに異なっている。
- 3) カードには数字が振られている。
- 4) カードは4種類のスーツに分かれている。
- 5) 裏表を選ぶことで、何のカードであるかを隠すことも表示することもできる。
- 6) セットで販売されており、すべてのカードを一人で所有することが容易である。

以上の六つがトランプの本質であり、すべてのトランプゲームはこれを基本としてデザインされている。すべての特徴がトランプゲームのデザインに深く関わっている。トランプで実現しやすいゲームもあれば、実現しにくいものもあるだろう。たとえば、確率が絡んでくるようなゲームは、トランプに向いている。トランプは2値（数字とスーツ）を持っているから、これを組み合わせれば、まさに無数の確率の組み合わせが実現できる。カードの裏面から得られる情報は非常に限られているから、当て推量や直感を働かせるようなゲームも作りやすい。事実、直感が重要なゲームの代表ともいえるポーカーでは、面白みのない確率論で

はなく、相手のカードが隠されているという点がゲームの本質となっている。

さまざまな他のゲーム環境と同じく、トランプにも欠点がある。たとえば、52枚のカードしかないのに、参加者が52人以上いるようなゲームをデザインするのは難儀である。トランプを使ったアクションゲームとか、体を使うゲームをデザインするのも無理がある。たとえそのようなゲームがデザインできたとしても、それが素晴らしいものになるとはとても思えない。

もちろん、これはトランプがゲームに向いていないということを意味しているわけではない。トランプでできることもあれば、できないこともあるというだけの話である。ボードゲームは、トランプよりもより多彩なゲームが作りやすい。ボードゲームは、カードよりもっと複雑な要素をゲームに盛り込めるから、その特徴をトランプほどすっきりと定義することはできない。ボードゲームとはこのようなものだという説明はできても、厳密な定義にまでは至らないのである。ボードゲームでは、何らかの模様が描かれた大きな紙やゲーム板を使う。その模様は簡略化された何らかの地図を表していることが多い。多くの場合、その地図は、（四角形なり六角形なりの）規則正しいマス目、こまかく区切られた道、不定形の領域、分岐点といったものでさまざまな領域に分割されている。普通、地図そのものはゲームの間には変化しない。プレイヤーはその地図の上を動かすことができるいくつかのマーカーを操ることで状況を変化させていく。偶然の要素を表すために、ランダムな状況を作り出す装置が使われることも多い。ルーレットやサイコロが良く使われる。ランダム性を作り出すために専用のカードが使われることもある。

ボードゲームというゲーム環境は非常にうまくいった。さまざまな嗜好を持つプレイヤーたちに合わせたゲームをデザインするのも簡単で、非常に広範囲のゲームをデザインすることが可能である。チェスは古典的で不朽というべきボードゲームだし、パーカーブラザーズ (Parker Brothers) の『モノポリー』 (MONOPOLY) は、不動産売買を扱った素晴らしいボードゲームである。その他にも、人生、殺人事件の推理、レースといったさまざまなテーマを扱った、数多くのボードゲームが存在する。いま現在、最も大がかりなボードゲームは、ウォーゲームだろう。中には1.5m四方もあるボードを何枚も使うような巨大なゲームも存在する。動かすコマは数千個にのぼり、50ページに及ぶルールブックが付属している。ここまでくると、ゲームを作るのもちょっとした事業になる。歴史的な調査が行われ、スターデザイナーたちが集まり、連中にしか理解できないような言葉で打ち合わせが行われるだろう。

このように、ボードゲームというゲーム環境は非常に自由度が高い。しかしながら近年、多少停滞気味のようなのである。数こそ多く出てはいるが、皆『モノポリー』の安易なコピーになってしまっている。ウォーゲームも60年代、70年代にブームを迎えて以来、すっかり停滞してしまっただけで、本当の意味で新しいものは、ほとんど出てきていない。もはや、このボードゲームという鉱脈は掘り尽くされてしまったのかもしれない。

ボードゲームの欠点といったら何だろうか。第一に、誰か特定のプレイヤーだけに固有の情報を与え続けることが難しいということがあげられるだろう。それぞれのコマがゲームボード上でどこにあるのかは、すべてのプレイヤーから一目瞭然なのだ。第二に、細々としたコマをひとつずつ、プレイヤーたちの手で動かさなくてはならないということである。これは、前に例にあげたようなモンスター級ウォーゲームなどでは特に、相当面倒くさいことになる。そのようなわけで、ボードゲームは1ゲームが長引きやすく、一晩がそれだけで終わってしまうことも多い。1ゲームが20分以下で終わるような、短いボードゲームは極めてまれである。さらに、誰かがボードをひっくり返してしまったり、コマが目茶苦茶になってゲーム

が成立しなくなってしまうという問題もある。

ここまでの議論では、すべてのゲームは何らかのゲーム環境のもとでデザインされており、そのゲーム環境にはそれぞれ固有の利点と欠点があるということを述べてきた。ゲーム環境にはそれぞれ、得意、不得意があるのである。目端のきくゲームデザイナーになろうと思ったら、使おうとするゲーム環境の利点と欠点をしっかりと把握していなければならない。では、ゲーム環境としてのコンピュータとはいかなるものであるかを見ていこう。

コンピュータ

コンピュータをゲームに使えるようになり、最も大きく変わった点は、コンピュータはプレイヤーの打つ手に対応できるということだろう。こちらの動作に対応してくれるというのは、コンピュータとプレイヤーの相互作用の中で最も重要なことであり、すべてのゲームにおいて重要な意味を持つ。コンピュータは、プレイヤーの希望をさまざまな方法で満たしてくれる。たまに、カードゲームやボードゲームで膠着状態に陥ってしまい、続けるのがいやになってしまうことがあるが、人間同士のプレイでは、いつまでもその苦行を続けるか、いっそやけくその大攻勢にでるかくらいの選択肢しかない。コンピュータ相手であれば、一気に時間を進めてしまえば良いのだ。ゲームの所要時間、難易度、ルールですら変えることができる。『アタリ2600』(ATARI 2600)用の『スペースインベーダー』(SPACE INVADERS)が良い例となるだろう。プレイヤーが一人か二人か、透明のインベーダーを出現させるかどうか、要塞が動くかどうか、弾のスピード等々、本当にさまざまな設定が変更できる。要するに、プレイヤーがルールを決めることができるのである。これは、コンピュータゲームがプレイヤーの希望を細かく反映できるということに他ならない。

これは、コンピュータの自由度が非常に高いところから来ている。コンピュータはその設定を動的に変化させることができる。すなわち、ゲームから固定要素を排除することができるのだ。ボードゲームやカードゲーム、あるいは、スポーツにしる、それをデザインする上で何らかの固定要素はどうしても必要だった。10万枚のゲームボードが刷り上がってきたあとで、マップに調整を入れようとしてもちょっと難しいだろう。作れと言われても、53枚セットのトランプを作ることはできない。よほどの技術的進歩がなければ、より遠くまで飛ぶフットボールは作れないし、スタジアムの拡張には巨万の富が必要だ。そのようなものと比べたら、コンピュータは制限なしも同様だろう。すべてのパラメーターをゲームプレイ中に自由に書き換えることもできる。フットボールゲームで、敵軍のゴールポストを遠ざけてやることだって簡単だ。ウォーゲームで自軍の占領地を広げてやることは、リビングルームのイスの場所を変えることより簡単だろう。このような自由度は、ゲームをデザインする上で最も重要なことである。しかしながら、これまでのところ、このコンピュータの自由度がうまく活かされているといえるようなゲームはあまりない。

コンピュータをゲーム環境として考えたとき、次にあげるべきはゲームにおけるレフリーとしての働きだろう。従来のゲーム環境では、誰かがゲームを管理することに時間を割く必要があった。スポーツはその最たるものである。厳格にルールを適用し、ルール解釈の違いによるもめ事を採決するために、複数の公平な審判員が必要になる場合もある。トランプやボードゲームにおいても、プレイヤー自身がレフリーの役を果たす必要がある。レフリーはトランプにおいてはそうでもないが、ボードゲームでは結構大変だ。ウォーゲームのような複雑なゲームではなおさらだろう。複雑なルールをいかにげんに解釈したことがもとで、いさかいが起こり、ゲームそのものが成立しなくなる場合もある。コンピュータを導入すれば、こういった厄介ごととはですべて解決される。プレイヤーがゲームに集中することが可能になるのだ。さらに、大きな利点がある。複雑な計算が入るようなややこしいルールが導入

できるということだ。従来のゲーム環境では、プレイヤーの管理能力を越えてしまうため、それほど複雑な計算をルールに含めることはできなかった。コンピュータにはそのような制限は存在しない。

たとえば、『東部戦線1941』(EASTERN FRONT 1941)のオリジナルヴァージョンにおいて、私は非常に複雑な計算式を使うことができた。これまでの第二次世界大戦の東部戦線を扱ったゲームでは、勝利点を評価するために、占領された都市の数と後はせいぜい互いに与えあった被害の量、程度しか使われてこなかった。これ以上のややこしい計算は人の手にあまるからである。しかし、『東部戦線1941』のオリジナルヴァージョンでは、占領地や被害の多寡だけでなく、個々のドイツ軍ユニットがどれだけ西進したか、個々のロシア軍ユニットがどれだけ東進したかということまで計算に入れることができた。これにより、ゲームはより現実的になり、プレイヤーの力量を明瞭に示すようになったのである。

ゲーム環境としてのコンピュータにおける三つめの利点は、リアルタイム処理が可能だということである。その他のゲーム環境では、何か判定すべきことが出てくるたびに、いったんゲームの進行を止める必要がある。コンピュータは、人間の判断よりもずっと高速にさまざまな判定をくだすことができる。リアルタイムゲームが可能なのだ。アクションゲームはその一番の例だろう。コンピュータの高速性を生かすことで、カードゲームやボードゲームにおいてすらターン制を採用せずにリアルタイム処理を行っているものも出てきた。

ゲーム環境としてのコンピュータにおける四つ目の利点は、コンピュータに知性的な対戦相手を演じさせることが可能になったということである。その他のゲーム環境では、どうしても生身の対戦相手が必要になる(トランプのソリティアのような例外もあるにはあるが、これはゲームよりもパズルに分類すべきものだろう)。この面で、いまのところ、最大の成功というのは、コンピュータにチェスを指させることが可能になったことだろう。パソコン用のチェスプログラムはたいていの素人チェスプレイヤーには勝てるくらいにはなっている。ゲームにおける最高の人工知能といえるだろう。チェス以外のゲームはそこまで知的なゲームだとはいえないが、それでも人間の持つ知性かわりに強力な計算力を使って対戦相手を演じることができる。近い将来、もっと手強い敵を演じられるようなアルゴリズムが作り出されるのを期待しよう。

ゲーム環境としてのコンピュータにおける五つ目の利点は、プレイヤーから特定の情報を隠しておくことが容易だということである。これは、非常に重要な利点となりうる。情報を隠すことでプレイヤーに推理力を使わせることができるのだ。推理はもともと非常に面白い挑戦になりうる。1から10までのランダムな数字をあてることはそれほど面白いことではないが、相手の行動や性格を観察して対戦相手がいま何を狙っているのかを推理することは、はるかに面白い。非常に複雑で限定された情報から何かを推理するのは、プレイヤーにとって本当に面白いことなのである。

プレイヤーに提供する情報を制限できるということは、別の重要な利点にもなりうる。さきにも述べたように、ゲームは現実世界の非現実的な再現であり、それにリアリティを与えるためにはプレイヤーが自らの想像力を駆使させる必要がある。与えられる情報が制限されていることで、かえって想像力を刺激することができるのである。必要な情報がすべて与えられていれば、推理の働く余地はほとんどない。ほんの取っ掛かりだけが知らされているからこそ、そこにしがみつこうとして、ありったけの想像力がかきたてられるのである。ゲームが再現しようとしているものの中で、リアリティほど重要なものがあるだろうか。ゲーム内の情報をあえて制限することで、想像力を刺激し、ゲームで表現された事象を現実社会としっかりと結びつけるのだ。その過程の中で、このゲームはリアルであるという幻想が生ま

れ、より効果的に空想世界への道を開くのである。

ゲーム環境としてのコンピュータにおける六つ目の利点は、たとえば電話回線といったネットワークを通じてデータのやりとりができるということである。ネットワークが使えることでその他のゲーム環境では全く手の届かなかったようなゲームが可能となる。たとえば、とんでもない数のプレイヤーが同時に参加するようなゲームだ。これまでは、管理上の問題からゲームに参加するプレイヤーの数には必ず限度があった。専門のレフリーを置かないゲームでは、プレイヤー数は6人程度が限界だろう。プレイヤーが12人にもなれば、レフリーも一人では足りなくなるし、20人を越えればもっとたくさんのレフリーが必要になってくる。したがって、プレイヤーが何百人もいるようなゲームは、どうしても、いかにそれを管理するかという問題に直面するのである。加えて、どうやってそれだけの人数を集めるかという問題も、ゲームの開催を困難にする。これらすべての問題は、ネットワークに繋がったコンピュータを利用することによって解決できる。ネットワークを用いることで、世界中に散らばった何千人ものプレイヤーが互いに相互作用しあうようなゲームをデザインすることも可能だ。プレイヤーはやりたいたときに自由にゲームに参加し、抜け出すことができる。全参加者の数が十分大きければ、特定の個人の出入りはゲームの大勢に影響を与えないのである。

その他のゲーム環境と同じく、コンピュータも利点だけでなく欠点も持っている。第一にあげるべき最大の欠点は、たいていのコンピュータはインターフェースが非常に貧弱だということである。コンピュータは、それ自身は非常に高速である。しかし、プレイヤーが自分のしたいことをコンピュータに告げられなかったり、コンピュータのいつていることを理解できなかったりすれば、そのスピードは全く無駄になってしまう。換言すれば、コンピュータは何とかしてプレイヤーと意思疎通する必要があるのである。これを実現するインターフェースは非常に難しい。現状のコンピュータからの出力はほとんどの場合、グラフィックとサウンドであり、入力、キーボードやジョイスティック、パドルといったものを使うしかない。

コンピュータからの出力としてまずあげるべきはグラフィックだろう。高品質のグラフィック環境は、まだなかなか手に入らない。パソコンレベルでは世界一のグラフィックをうたうアタリ (the Atari Home Computer System) ですら、ゲームをデザインする上では厳しい制約がある。自分が描きたいと思う細かな画面をそのまま表現することさえできない。たとえば、このコンピュータ上に一画面で表示できるボードゲームがどれだけあるだろう。使える色数にしる、高解像度モード上に表示できるテキストの行数にしる、ゲームボードの大きさを考えると、とてつもなく大変なことだとわかる。ボードを塗る色数を減らし、文字数を削り、画面よりも大きなボードをスクロール表示するなど、ありとあらゆるテクニックを使ってはじめて、同じようなゲームボードが再現できるのだ。『東部戦線¹⁹⁴¹』では、こういった手法をすべて使った。その結果は、もちろん有益だっただろう。しかし、コンピュータ上でグラフィックを使うのは一筋縄ではいかないのである。

もちろん、コンピュータのグラフィックにしかできない利点というものもある。コンピュータゲームでよく見られるような、アニメーションを使ったり、ボードそのものが変化していったりするようなボードゲームは見たことがない。また、グラフィックがあるというだけで、人が受けるインパクトは非常に大きくなるのも間違いない。グラフィックがすべて悪いということはないのだ。

入出力に関するもうひとつの問題点は、もちろん入力である。コンピュータへの入力はいまのところキーボードか何か他のコントローラーを使わなくてはならない。これは、ゲームデザイナーにとって頭の痛い問題である。第一に、ジョイスティックにしるキーボードにしる

そうそう多くの情報を一度に入力できるわけではない。ジョイスティックから入力できるのはたかだか五つ、「上」、「下」、「右」、「左」、そして「ボタンが押された」という情報だけである。もちろん、キーボードならもっと複雑な入力が可能だが、それには長い間違えやすいキーシーケンスを打つ必要がある。コンピュータと何か実のある会話をしたいと思うなら、とにかく長くて不器用なコマンド列を間違いなく入力しなくてはならないのだ。キーボードにしるジョイスティックにしる、入力方法としてはまわりくどくややこしい。そこには、現実世界のアクションと直接結びつけられるようなものはほとんどない。その他のゲーム環境では非常に簡単で明白な行動ですら、コンピュータ上で再現しようとするときまさに神秘のベールに包まれてしまうのである。バットを渡されて、「あのボールを打つことが野球の目的だ」と言われたら、「このバットをボールに向かって振ればいいんだな」ということはすぐにわかるだろう。ところが、コンピュータ上の野球ゲームではそう簡単にはいかない。一体どのキーを押したら良いのだろう。「ヒット」だから[H]? 「スイング」だから[S]? それとも「バット」の[B]だろうか? ジョイスティックのボタンを押せば良いのかもしれないし、スタートキーを押すのかもしれない。いっそのこと、ジョイスティックのケーブルをつかんで、画面に映っているボールめがけて振り回してみたらどうだ。

ゲーム環境としてのコンピュータを考えたときに、入出力について問題になるのは、コンピュータが基本的に一人用として作られているということである。普通、コンピュータは、コンピュータデスクに鎮座していて、それを使うときにはその前に座る必要がある。これを二人で使おうと思ったら、いちいち席を変わらなければいけない。まったく面倒でややこしいことだ。ジョイスティックやパドルを使えば多少は楽になるが、それで十分というわけでもない。そういうわけで一人用のゲームが増えて、コンピュータゲームは社会性を疎外するなど非難されることになる。ボードゲームなら、友達を何人も呼んでテーブルを囲むことができるのに、コンピュータゲームになると普通は一人、せいぜい二人用で、それ以上人が集まるとどうしようもないというわけだ。

ここで取り上げるゲーム環境としてのコンピュータにおける最後の欠点は、コンピュータゲームを作るにはどうしてもプログラムが必用だということである。デザイナーにここまで負担をかけるゲーム環境は他にはないだろう。たとえばボードゲームのデザインを考えてみよう。とりあえずは、ラフにゲームボードをスケッチして、コマを適当にでっち上げればそれでプレイは可能だ。いざ発売というときには、プロのグラフィックデザイナーにおまかせするだけで、きれいな商品に仕上げしてくれるだろう。ゲームデザイナー自身が、きれいに絵が描ける必要はどこにもない。

コンピュータゲームデザイナーはそうはいかない。ゲームをデザインするためにはどうしてもそれをプログラムしてコンピュータ上で動かしてやる必要がある。プログラミングはそれそのものが厄介で難しい。しかも、そう簡単に人に任せるわけにもいかない。なぜなら、プログラミングの努力自体が、コンピュータゲームのデザイン過程に大きな影響を及ぼすからだ。そのゲームをいかにプログラムするかというのは、コンピュータゲームデザイナーにとって非常に高いハードルなのである。

コンピュータゲームデザインの六つの鉄則

ここまでに述べてきたような、ゲーム環境としてのコンピュータの利点や欠点をゲームデザイナーの卵たちへ伝える指針とするにはどうしたら良いだろう。ここでは、これらを六つの鉄則としてまとめてみよう。

鉄則1：適材適所

いかにも頭が鈍そうなコミックのヒーロー。近くにはロケットが停泊している。すぐそばには、車輪のとれた乳母車。我々がヒーローは車輪ととんかちを手にして、乳母車からロケットへとえっちらおっちら歩いている……。

第一の鉄則は、コンピュータという素材をうまく活かそうという意味である。非現実的な目標を立てるゲームデザイナーが多い。彼らは、コンピュータが苦手とすることをさせたがるのである。プログラミングで手抜きをしろという意味ではない。コンピュータはあくまで人間が使う道具であり、コンピュータの都合はデザイナーにとって重要ではないということをおぼわすてはいけないのだ。我々の目標は、コンピュータから最大のパフォーマンスを引き出してベストなものを作ることである。そのためには、コンピュータが得意なことをさせるしかない。

この鉄則にあてはまる例をひとつあげておこう。ウォーゲームのプレイヤーは、昔から、六角形のマス目でしきられたゲームマップに慣れ親しんできた。このヘクスシステムは、コマの移動方法と位置をうまく表すことができる。ヘクスシステムは、単純な四角のマス目よりもいくつかの点で優れている。まず、四角いマス目には対角線上の位置関係が存在する。二個のユニットが互いに斜向かいの位置になれるのだ。ルールをデザインする上で、この斜向かいの位置関係というのは厄介である。たいていの場合、難しくて例外的なルールを作ることが必要になる。ヘクスシステムなら、どんな場合でも位置関係は等価であり、このような問題は起きない。次に、ヘクスシステムでは、六方向好きな方向にコマを動かすことができるのに対し、四角いマス目では四方にしか動けない。移動先の選択肢が多いということは、より細かなコマの移動が可能になるということなのである。

こうして考えてみると、コンピュータウォーゲームにもヘクスシステムを持ち込もうとしたというのは、非常に自然な考え方といえる。ヘクスシステムを採用したコンピュータウォーゲームは非常に多い。しかし、それは大きな間違いなのだ。確かにヘクスシステムには利点も多い。しかし、コンピュータ上では、紙のボードゲームにはなかった問題が出てきたのである。紙の上になら何だって好きなものを印刷できるだろう。しかし、コンピュータディスプレイの解像度や色数ではそうはいかない。もともとコンピュータディスプレイの座標系は四角いのだ。水平方向の走査線を縦に積み重ねて表示しているのである。

つまり、コンピュータディスプレイは、六角形を描くよりも四角を描くほうが得意なのだ。六角形を描くためには、斜め線を四本も引かなくてははいけない。ところが、ディスプレイ上にかかれた斜め線は、ガタガタの四角いドット列になってしまう。はっきりとわかる線（少なくとも1ピクセルの太さは必要だ）で区切られたヘクスシステムをディスプレイ上に再現してみよう。小さいヘクスを多数並べると、画面の大部分が線で埋まってしまうだろう。だからといってヘクスを大きくすれば、今度は一度に表示できる数が少なくなってしまう。さらに、ジョイスティックもヘクスシステムには向いていない。もともと、ジョイスティックは四方向の入力用の装置なのだから。もちろん、コンピュータゲームでヘクスシステムが使えないというわけではない。実際、ヘクスシステムを採用したコンピュータゲームは数多い。ただ、この問題によって、グラフィックが簡略化され、非常に使いにくいものになってしまっている。とても、スムーズに働いているとはいえないのである。ヘクスシステムと等価な利点を持ち、この問題を解決したシステムを採用したゲームもいくつか存在する。ヘクスの替りに、互い違いの四角形を採用したシステムである。このシステムでは、ヘクスシステムの利点を保ったまま、ディスプレイにとっても表示しやすくなっている。もっとも、ジョイスティックに向いてないという点は変わらないが……。

以上のようなことを考えて、私は『東部戦線¹⁹⁴¹』では、あえて古臭い四角いマス目を採用

した。もちろん、手抜きをしたかったとか、ヘクスシステムの問題点を解決するだけの気力がなかったというわけではない。実際、別のゲーム『タクティクス』(TACTICS)では、ちゃんとヘクスシステムを使っているし、そのためのプログラムも書いた。しかしそれでも私はヘクスシステムのためのプログラムを書いているうちに、やはりそれはゲームにとって絶対に必要なものではないということを確認するに至った。事実、『東部戦線¹⁹⁴¹』はヒットし、ヘクスシステムを使っていないということは決してハンデには成らないということを証明している。

鉄則2：ちょっと拝借でいいの？

おお、我らがヒーローは、崖からまっ逆さまに落ちていく。腕に間に合わせの翼をつけ、狂ったように打ち振りながら……。

コンピュータゲームの世界には、胸が悪くなるほどたくさんの移植ゲームがある。もともと別のゲーム環境で作られたゲームを、何を勘違いしたのかコンピュータ上に生まれ変わらせたものである。いくら皆がやっていることだといっても、これが本質的に愚考であることは間違いない。彼らの言い分はこうだ。「コンピュータゲームはまだ生まれたばかりだ。コンピュータゲームをデザインするためのガイドラインが存在しない以上、既存のゲームのガイドラインをあてにするしかないだろう」と。いつの日か、ゲームデザイナーたちは、そんな安易な移植ゲームの山を見て笑う日が来ることだろう。大昔の羽ばたき飛行機的设计図をあざ笑うように。

どうしてそこまで移植ゲームを忌み嫌うのかというと、移植ゲームは、ちっともその気のない二つのゲーム環境の間に生まれてしまった厄介者の私生児のようなものだからである。一番ひどい例をあげてみよう。コンピュータゲーム版のクラップスだ。ルールは、普通のクラップスと同じで、コンピュータが二個のダイスを振って結果を表示してくれる。確かにうまく動いている。しかし、なぜ他のゲーム環境で完成しているゲームをコンピュータで再現するために、わざわざ面倒くさいことをする必要があるのであるのだろうか。ダイスを二個買うのだって一ドルもしない。コンピュータ・クラップスがオリジナルと比べて圧倒的に劣っている点がある。クラップスの売りのひとつは、プレイヤーが自分でダイスを振ることができるということだ。皆、ダイスを握り締め、「頼んだぜ、サイコロ君」とささやき、ときにはそれにキスして運が向いてくるのを願う。プレイヤーは皆、そうやって、ツキをコントロールできると信じている。それが、ただ見ているのではなく、ゲームに参加しているってことだろう。ところが、コンピュータ・クラップスにはそれが無いのだ。確かに、ダイスを支配している数学は同じでも、プレイヤーに空想や幻影を与えることはできないのである。

多かれ少なかれ、移植ゲームはオリジナルの何かを移植する過程で失ってしまう。たまに何かを得ることはあるが、必ず何かを失っているのである。なぜなら、別のゲーム環境で作られたゲームは、そのゲーム環境の中で最適化されてきたからだ。特定のゲーム環境下で、最も利点が多くなり、欠点が少ないように作られているのである。移植ゲームは、異なる利点と欠点を持つゲーム環境に、わざわざ同じゲームを再現しようとする。これでは、まずオリジナルに及ばないのは間違いないだろう。すべての印象的な芸術作品は、それに見合ったイメージ媒体で形作られている。シェイクスピア(Shakespeare)を本当に理解しようと思うなら、エリザベス一世の時代の英語(Elizabethan English)で読まなければならない。現代英語に訳してしまえば、シェイクスピアの持つ華麗な言い回しや語感が失われてしまう。ギリシャ語のリズムで謡ってこそ聴衆をぞくぞくさせるソクラテス(Isocrates)の演説も、現代英語になつては、くすんで退屈なものになってしまうだろう。本を読んで感動したからといって、映画化されたものを見て満足したことがあつただろうか。この法則がコンピュータゲームに

当てはまらないわけではない。他からちょっと拝借してきたところで、面白いゲームは作れないのである。

鉄則3：まずは使い勝手

我らがヒーローは、ついに巨大な装置の前にたどりついた。とても複雑な装置のようだ。あちこちに、パイプやバルブ、煙突にケーブルが走っている。装置の正面には「選択してください」と書かれているぞ。そのすぐ下には「選択A」、「選択B」とかかれた二つのボタンがある。その隣には、「あなたの勝ち!」、「あなたの負け!」と書かれたランプが並んでいる……。

これまでも述べてきたように、コンピュータの計算力は非常に大きい、しかし、こと入出力となると貧弱としかいえない。すなわち、コンピュータゲームデザイナーにとっての当面の問題は、コンピュータの計算能力に関してではなく、I/Oまわりに関することになる。コンピュータと人間の情報のやりとりをどうするかが一番の問題なのだ。

コンピュータからの情報が、ディスプレイやサウンドから自然と受け取れるようにデザインする必要がある。ゲームシステムそのものは素晴らしいのに、I/Oまわりがダメでどうしようもなくなったゲームを数多く見てきた。表示があまりに乱雑だと、せっかくきれいなグラフィックを作っても、プレイヤーには見てもらえない。もっとひどいのは入力がダメなゲームである。特に安易にキーボード入力を採用したものがひどい。キーボード入力となるとたじたじになってしまうゲーマーがほとんどだろう。もちろん、困難に挑戦することそのものが面白い場合もあるが、キーボード入力がうまくいかないのはイライラするだけだ。ゲームの大半はI/Oで決まってしまう。何が表示できて何が表示できないのか、何が入力できて何が入力できないのか、ゲームの仕様はこういうことで決定されるのである。

I/Oまわりが仕様を決めた良い例として、私のデザインした二つのゲームをあげてみよう。

『東部戦線1941』と『タンクティクス』(TANKTICS)はどちらも第二次世界大戦を扱ったゲームである。どちらのゲームも、結構手ごわい敵、細かなシミュレーション、多彩なオプション、やりがいのあるステージといったものを持っている。そういった点ではどちらもだいたい等価だ。両者の違いはI/Oにある。『東部戦線1941』では、I/Oを中心にゲームをデザインした。見やすく十分な情報が得られる画面と、直感的にわかりやすいジョイスティックによる入力を採用している。一方、『タンクティクス』では、ゲームそのものがデザインの主眼となった。その結果、使いにくくてややこしいキーボード入力とまるで暗号のような画面ができあがった。『東部戦線1941』は、絶賛を持って迎えられいくつもの賞をとることになったが、一方『タンクティクス』は酷評されることになった。I/Oまわりの差がすべてを決めたのである。

鉄則4：シンプルが一番

我らがヒーローは特注バイクを走らせている。全長6メートルのこのバイクの装備は、数え切れない。バックミラー、パワーステアリング、ブレーキにスロットル。もちろんシートやハンドルの高さ調整も自由自在だ。フロントガラスだけでなく、すべてのミラーにワイパー完備。テレビにハンバーガーの自販機まで、思いつく限りの装備がこのバイクにはついている……。

ゲームをデザインするにあたって、いま作っているゲームを自分の理想に近づけようとして、かえって失敗するゲームデザイナーは数多い。何か問題が出てきたときに安易にその場

しのぎの変更を加えてしまい、その結果、ゲームシステムがシンプルさを失ってしまうのだ。もし、良いゲームだとプレイヤーを感動させようとするなら、ゲームは芸術的なまでに統一性を保っていないとてはならない。それには、ゲームは表現しようとするテーマにそって作られており、気を散らすような枝葉が可能な限り排除してある必要がある。

ここで、なぜ、ゲームのメインテーマとして複雑なものを持ってきてはいけないのかについて述べておこう。複雑さがゲームに悪影響を与えるということはほとんど認識されていない。なぜなら、適度な複雑さはゲームに彩りを加え、より現実味のあるものにしてからである。適度な彩りはゲームを素晴らしいものにするには間違いない。しかし、ゲームデザイナーは、その彩りを加えることでゲームがより複雑になっているという事実を知っていなければならない。その彩りは、度を越すと乱雑な印象へと変わる。デザイナーはできる限り彩りを増そうとするが、しばしば、加えた彩り以上に複雑なゲームを作ってしまう。ゲームデザイナーにとって、ゲームをどこまで複雑にするかというのは、常に考える必要のあるトレードオフであり、ちょっとした問題の解決のために安易に変更して良いものではない。

複雑な処理は往々にして、めったに起きないような例外処理の場合にゲームに入ってくる。たとえば、『東部戦線¹⁹⁴¹』にも、さまざまな例外処理があり、ゲームを複雑なものにしている。なかでも一番ひどいのは、特定のフィンランド軍ユニットは攻撃に参加できないというルールだ。このルールが適用されるのは、2個のフィンランド軍ユニットにすぎないため、ゲーム全体にはほとんど影響がないが、それでもプレイヤーを悩ませるには十分である。このルールは、ゲームとプレイヤーを混乱させるだけでメリットはほとんどない（あるデザイン上の問題をクリアするためにどうしてもこのルールを入れなくてはならなかったのだ。フィンランド軍が自力でレニングラードを手に入れてしまうのを阻止するにはそれしかなかった）。

枢軸国側（ルーマニア軍、ハンガリー軍、イタリア軍）のユニットに関しても、例外処理的なルールを採用した。それらのユニットは、ドイツ軍ユニットと比べると攻撃力が弱いのである。枢軸国ユニットは『東部戦線¹⁹⁴¹』中に6ユニット存在する。このルールは、2ユニットだけに関係するルールと比べれば使われることも多く、また、それほどゲームを複雑にしているわけでもない。また、機甲ユニットは歩兵ユニットよりも移動力が大きいというルールがあるが、このルールは、ゲーム中に登場する大量の機甲ユニットすべてに適用されるという点で、例外処理にはあたらない。これは、ゲームを無意味に複雑にするルールではないのだ。

以上の考察から、そのルールの適用範囲が狭ければ狭いほど、ゲームは乱雑になっていくとすることができる。すなわち、ゲームを乱雑にしないためには、例外処理のためのルールを使わずに、ゲーム上で再現すべきすべての状況を表現できるようなシンプルなルール体系を構築しなくてはならないのである。もし、完全なゲームデザインというものがあつたら、すべてのルールはあらゆる状況に普遍的に適用される。もちろん、完璧なゲームが作れるはずもないが、デザイナーはすべてのルールがなるべく広い適用範囲を持っているように努力する必要がある。プレイヤーは、すべてのルールの関わり合いを考慮して、意思決定を行うのだから。

中には、夢の島学派 ("humongous heap" school of game design) とでも呼んでやりたくなるようなゲームデザインをする連中がいる。彼らは、ゲームデザインというものは、単純な基本ルールの上に思いつく限りの枝葉末節（彼らはこれを称してディテールとか言っているらしいが）をめったやたらと積み重ねていくものだと考えている。ノミの代わりにシャベルを

使っているようなものだ。彼らは、デカさと壮大さ、ややこしさと巧妙さの区別がついていないのである。

鉄則5：少なく貯めて大きく使おう

我らがヒーローは街で大道芸を始めた。隣の男は、5、6個のボールをいとも簡単そうに操っている。おお、なんてことだ。我らがヒーローは、両手いっぱいボールをお手玉しようとして結局それにつぶされてしまった……。

コンピュータの情報記憶の役割は誤解されていることがある。コンピュータは単なる情報記憶装置ではなく、情報処理装置なのである。コンピュータの情報記憶は、あくまで情報処理のための前提条件に過ぎず、記憶するだけでは何にもならない。記憶容量がふえれば、処理すべき情報量も増加するが、コンピュータの処理能力はそれに追いつかないため結局は無駄になるだけである。理想的なプログラムでは、データの量と処理能力のバランスがとれている。それに対して、私が見てきた限りほとんどのゲームでは、扱う情報が多すぎて処理しきれない。データ量を増やすほうが、プログラムを改善するよりも簡単だからだ。どうしても、安易に流れるデザイナーが多いのである。

このように、膨大なデータ量を誇るゲームというのは、実は決してマシン性能を活かしきったゲームではないのである。コンピュータの性能をぎりぎりまで使い切ろうと思ったら、そのデータをいかに処理し、いかに動的に変化させていくかというところに力を注がなくてはならない。数値の変化しない数表などというものは、ルールブックに書いておけば良い。静的な情報を扱うことにおいては、紙とインクはいまだにコンピュータに勝るとも劣らないツールである。そこらに散らばっていてめったに使わないようなデータはさっさとゴミ箱に捨ててしまおう。コンピュータのメモリの中にはそんなものを置いておく余裕はないのだ。プログラムを見て考えよう。「この1バイトは金を貰うに値するだろうか？ちゃんと働いているのかな。たまにしか使われないような怠け者のデータじゃないだろうか」。プログラムの中には、怠け者のデータではなく、有用な働き者のコードを詰めこもう。

怠け者のデータは、例外処理の中に出てくることが多い（安っぽいビリヤード場でぶらぶらしている不良のようなものだ）。例外処理はめったに適用されない特別なルールである。当然、そのルーチンが呼ばれることも少なく、怠け者のデータとなりやすい。

この鉄則に従ったほうが良いもうひとつの理由は、ゲームをプレイする上での基本に関わっている。ゲームの醍醐味はやはりインタラクティブ性である。前にも述べたように、コンピュータは非常に柔軟で、その本質からしてインタラクティブである。しかし、コンピュータの持つインタラクティブ性は、プログラムしだいで容易に台無しになってしまう。静的な情報を重視したプログラムが動的になるはずはない。扱う情報が固定されてしまうと、当然ながら人間の入力に対する対応範囲が狭くなってしまい、インタラクティブ性を失う。一方、情報処理を重視したプログラムは、柔軟で入力に対する反応が良く、インタラクティブになる。そう、少なく貯めて大きく使うのである。

最後にコンピュータに限らず、すべてのゲームに関わることがらについて述べておこう。第一章で、ゲームは練り上げられた一本道の物語というよりも、細かなストーリーの分岐からなる網の目のようなものであると定義したのを覚えているだろうか。物語の善し悪しは、そこに含まれている情報の量によるところが大きい。一本道の物語というのは、実は情報こそ多いが、そこに含まれる情報そのものは決して変化しないのである。それに対してゲームの善し悪しは、そこに含まれる分岐の数によって決まる。そして分岐こそゲーム中の情報を処

理していくことに他ならないのである。データばかり多くて情報処理を怠ったゲームは、理想的なゲームというよりは、物語に近いものになってしまうのだ。

鉄則6：二足のわらじをはけ

我がヒーローは、壁を跳び越えて牢獄から脱出しようと、棒高跳びに挑んでいる。彼の左手の手錠は、共に投獄された走り高跳びの選手につながれている。腕がからまって地に墮ちるのは明らかなのに、彼らの顔は成功への期待に満ちている……。

ゲームを作るためにはデザインが必要であり、それをコンピュータ上に実現するためにはプログラミングの必要がある。ゲームデザインもプログラミングもなかなか習得できない技術であり、ましてや一人でその両方をこなせる人間など本当に数少ない。そのためたいていの場合、プログラムを知らない文系のゲームデザイナーと、芸術を知らない理系のプログラマーが組んで、開発チームを作ることになる。デザイナーとプログラマーがともに大人で、譲るべきところを譲りながら仕事を進めている間はうまくいくだろう。しかし、デザイナーとプログラマーの間には、ゲームを作る上で、血のにじむような選択をしなければならない場面があまりにも多い。デザインの専門家とプログラムの専門家にチームを組ませることは、棒高跳びの選手と走り高跳びの選手を手錠で繋いでおいて、あのバーを越えてみるというに等しい。悲惨な対立が起こるのは避けがたいのだ。

端的に言えば、デザイナーとプログラマー組んだチームが破綻しやすいのは、デザイナーがプログラミングについて無知なため、過度の期待をプログラマーにかけるからである。例をあげよう。私は『エナジー・ツァー』(ENERGY CZAR: エネルギー経済を扱ったシミュレーションゲーム)のデザインにおいて、あえてプレイヤーの行動記録を残さなかった。もちろん、記録があったほうが良いに決まっている。プロジェクトの最終段階で、ほとんどのスタッフがその点を指摘した。しかし、記録を残すためには、どうしても余分なメモリが必要になる。プログラミングを知らないデザイナーはそういった仕様にこだわって、プログラムを肥大化させ、必要メモリをあふれさせてしまうのだ。

『東部戦線1941』ではどうだったか。ゲームに使うカレンダーをコーディングしている間に、ここで何バイトか使えばカラーパレットを月ごとに変化させることができることに気付いた。この手を使って、画面上の木々の色を月ごとに変えることができた。ゲームを大きく進化させたとはいわないが、たった24バイトでこれだけのことができたのである。コストパフォーマンスは最高だといって良いだろう。プログラミングを知らないデザイナーには、こんなアイデアは浮かばないだろうし、もちろん、芸術を知らないプログラマーにだって無理だろう。

簡単に素晴らしいゲームを作る方法などというものはない。まずは、人間的な感覚と芸術的なセンスを持った、よきデザイナーを見つけなくてはならない。そして次に、その人にプログラミングの技術を教えるのだ。その逆(プログラマーからデザイナーへ)はうまくいかないと思われる。プログラマーを育てることはできるが、芸術家は生まれつきだからだ。プログラムを熟知したデザイナーさえいれば、あとは手足として働いてくれるプログラマーとデザイナーを個々に雇うことで、彼女の創造力を思う存分発揮させることができるだろう。雇われたプログラマーやデザイナーにも良い経験となるに違いない。いずれにせよ、本当に創造的な部分は独りで考えなくてはならない。いくら人を集めても、官僚的になり、意思決定が遅れて対応を悪くするだけで、創造的な作業には全く役に立たないのである。

結論

この章では、ゲーム環境としてのコンピュータについて述べてきた。コンピュータやそれが社会に与える影響についての議論は、肯定派と否定派の間の対立を生んできた。肯定派は、コンピュータにバラ色の未来を夢見て、コンピュータが生み出す無限の勝利について、あらゆる点で好ましいと語る。否定派は、コンピュータは人間性を奪う時間の無駄、いつかは人間を裏切る物としてすら見ている。この章では、私はコンピュータを単なるテクノロジーとして論じようとした。そう、とんかちと釘、土と石、紙とインクのような単なる道具、環境として。すべての道具や環境と同じく、コンピュータにはできることとできないことがある。芸術家がすべきことは、それが持つ欠点を巧みに回避して、利点を最大限に生かすために努力することなのである。

[目次に戻る](#) | [各章へ: 1-2-3-4-5-6-8-9](#) | [次章に進む](#)

[目標と題材の選択](#)[研究と準備](#)[デザイン段階](#)[I/O構造](#)[ゲーム構造](#)[プログラム構造](#)[デザインの査定](#)[プログラミング準備段階](#)[プログラミング段階](#)[テストプレイ段階](#)[事後検討](#)

ゲームデザインは、基本的には芸術的なものだが、同時に技術的なものでもある。ゲームデザイナーは、がりがりプログラムを書いているときでさえ、雄大で芸術的な目標を追い求めているものだ。ゲーム開発において、デザイナーは芸術と技術というふたつの非常に異なった世界で暮らしている。そのような異なった世界に、どのように折り合いをつけるべきだろう。どういう順序でコンピュータゲームのデザインを進めて行くのが良いのだろうか。前章では、このような手順に関係する若干の問題に触れ、いくつかの指針を定めた。この章では、実際にコンピュータゲームをデザインし、プログラムするまでの手順を提案しよう。

以下の手順は、私自身のゲームデザインの経験に基づいて、私がゲームデザインにおいて使っている習慣の多くが反映されている。しかしながら、私は、この手順を段階的に用いたことは一度もないし、他の人が厳密にこの手順に従うことも勧めない。だいいち、ゲームデザインというのは、形式的な手順にまとめてしまうには、あまりにも複雑な活動である。そして、ゲームをデザインしていく工程には、各デザイナーの個性が反映されるべきである。教条的に手順を守ることに依存してしまえば、ゲームデザインの芸術面が台無しになってしまう。最後に、私は主にパソコンゲームのデザインに関わってきたため、アーケード機や家庭用ゲーム機のデザイナーには、私の提案は完全には当てはめられない。したがって私は、公式としてではなく、将来のゲームデザイナーが仕事に取り入れることを望むかもしれない習慣集として、この手順を公開するのだ。このことを念頭において進もう。

目標と題材の選択

この段階が非常に重要だということは明白と思われるのだが、はっきりとした意志なしで出発したゲームデザイナーは、この段階を無視することがままある。ゲームデザイナーたちと話し合ってみると、ゲームの明確な目標の必要性に対する関心が欠けていると感じることが多い。ゲームデザイナーたちは良く、「楽しい」ゲームや「刺激的な」ゲームの創作を目指したなどと語るものだが、それ以上に明白な目標を持っていないことが多いのである。

ゲームは、目標が明確に定められている必要がある。プレイヤーに何を与えたいのかをはっきりさせる必要がある。ゲームが楽しいとか、面白いとか、刺激的であるとかいうだけでは不十分なのである。目標は、ゲームがどんな空想世界を作り出すのか、それがプレイヤーにどんな感情を呼び起こすのかを明示してなくてはならない。前にも述べたようにゲームの多くは何らかの意味で教育的であるから、目標はプレイヤーが何を学ぶのかを明示すべきである。ゲームがそのプレイヤーにどのような作用を与えるのか、ゲームデザイナーに尋ねてみる必要がある。

ゲームの目標をはっきりさせておくことの重要性が表面に現れてくるのは、ゲームデザイン

がかなり終盤になってからである。コンピュータゲームのデザインでは、何を捨てて何を残すかというトレードオフの問題が必ず出てくる。デザイナーが何かをゲームに盛り込みたいと思えばそれには必ず余分なメモリが必要になるし、コンピュータのメモリというものは常に不足しているものである。したがってデザイナーはトレードオフを考えなくてはならない。いくらかの要素は含めることができるが、いくらかは切り捨てなければならない。入れたい要素がふたつあり、どうしてもどちらかひとつを切り捨てなければならなくなったとき、どちらを残したら良いかを定める基準は、最初に明示した目標しかないだろう。明確な目標を持っていれば、辛いながらも毅然としてどちらかを切り捨てることができるだろう。目標がはっきりしていないと、間違っただけの選択をする可能性が高いし、どちらを選択したとしても、その決定が正しかったかどうかを判断することができないだろう。

どうすれば適切な目標が選択できるのだろうか。この質問に対する客観的な回答は存在しない。目標の選択は、コンピュータゲームデザインの芸術面における、最も主観的なプロセスである。これはデザイナー自身を表現する機会である。自分の信念や哲学、世界観を表現できるような目標を選択すれば良いのだ。正直であることが肝心である。もし、自分の好みではなく、観衆に媚びるような目標を選択するとしたら、生気のないゲームが出来上がることだろう。目標が、自分の興味や信念、感情と一致している限り、それが何であるかは重要ではない。自分に忠実に目標を選択しているかぎり、どんなゲームのであっても、他の人たちに対してそれなりの説得力を持つだろう。逆に、自分に対して正直でなければ、まわりくどく物まねのようなゲームができあがってしまうだろう。

これはあくまで理想論であり、実現することが不可能な場合もある。たとえば、子どもたちのためのゲームは、未熟で子どもっぽい人がデザインすれば良いというわけではないし、優れたシューティングゲームは、射撃の名人しか作れないというわけでもない。現実には、市場の要請から、自分の趣味に反したゲームを作らなくてはならなくなる場合もある。そんな場合でも、熟練したプロのデザイナーがデザインするほうが、作り笑いしかできないような愚か者にデザインさせるよりはずっと良いだろう。しかしながら、そのような感情的に間接的なゲームは、決して、心からストレートに伝わってくるような、心理的なインパクトや芸術的なパワーを持つことはないだろう。

目標を決めたら、次には題材を選択しなくてはならない。題材は目標を表現する手段であり、ゲームがプレイされる環境である。それは、抽象的な目標へと到達するための、具体的な状況と出来事を集めたものである。たとえば、『スター・レイダーズ』(STAR RAIDERS)の目標は、プレイヤーの戦術眼と操縦テクニックに挑戦し、怒りの暴力的解決を通してプレイヤーのストレスを解消しようというものだ。題材は宇宙の戦闘である。『東部戦線1941』(EASTERN FRONT 1941)の目標は、近代的な戦争の性質、特に火力と効果を表現しようとした。題材は独逸間の戦争である。

最初に題材を選択したために、目標が題材に従属させられているゲームデザイナーが多い。というよりも、彼らはゲームを開発するとき、目標よりも題材によって記述するのである。私が、他のデザイナーに、リーダーシップについてのゲームに取り組んでいると言うと、不思議そうな顔をされる。それはいったい何だ、ウォーゲームなのかダンジョンゲームなのかと。私が、アーサー王についてのゲームだと言えば、彼らは納得するようだ。それは、目標を題材に従属させてしまうという、重大な間違いなのである。最初に思いついた時点では、目標よりも題材に焦点が合っているかもしれないが、題材の勢いに身を任せるのではなく、目標実現のために題材があるのだという意識を持っていないと、いけない。

良い題材を選択するには時間がかかる。なぜなら、それぞれの題材候補について、ゲーム

の目標をうまく実現する能力があるかどうか、慎重に吟味しなくてはならないからだ。多くの題材には、ゲームの目標を妨害しかねない余計なお荷物がついてくる。たとえば、私の最近のゲームは、アーサー王伝説をその題材として用いる。このゲームでの私の目標は、リーダーシップの性質を考察することである。私は、アーサー王伝説が、この目標のための説得力ある伝達手段だと考えた。しかし不幸にもこれらの伝説は、腕力で対抗者を征服するという強力な構成要素を含んでいるのだ。このテーマは、私がゲームで表現したいことの若干を否定し、したがって最終的に題材の有益性を弱めてしまう。しかし私は、不利な点を勘案しても、この伝説は、とても力強く融通が利くと判断したのである。

研究と準備

目標と題材がしっかりと決まったら、その次の段階は、題材を深く知ることである。その題材について、できる限りの資料を読むのだ。目標と題材を関連付ける努力をする前に、すべてを研究しなければいけない。どんな面が気に入り、どんな面が気に入らないのか。あなたがゲームにおいて「世界の再現」をしようと試みる環境を、確実に理解しておくことだ。あなたのゲームは、実世界のリアルな感触を与えなくてはならない。それは、あなた自身がゲーム環境をしっかりと理解していなければ達成できないことである。私は『エクスカリバー』(EXCALIBUR)の研究段階で、西暦400~700年間の英国の歴史を勉強した。リーダーシップの性質を描写するという私の目的に直結した記述は、歴史の本の中にはわずかしは見つけられなかった。しかし、アーサー王伝説の中に、私の目的により近いテーマを発見した。このように、研究を進めていくにつれて、自然と目的を再調整していることに気付くだろう。このような気まぐれは、目的がもともと不完全だったことを認めるようで決まりが悪いが、題材環境の要求に対応する、率直な自発的意志を反映している。これまで何度も自分を甘やかしてきた、罪深き理想主義からの脱却なのである。

この段階において重要なのは、資料にこだわりすぎないことと、コードを絶対に書かないことである。あなたのゲームを熟考するため、長い散歩に出かけると良い。考えて考え抜くのだ。目標、題材、そしてあなたが勉強したことを、心の中で一緒にして、ぐつぐつ煮込むようにしてほしい。それらを合わせて、全体の中に織り込むのである。この段階にはゆっくり時間をかけることだ。焦りは、ゲームそのものを台無しにしてしまうような失敗のもとになる。私は、少なくとも3週間をこの段階にあて、ゲームのアイデアを発展させるようにしている。『エクスカリバー』では数ヶ月をこの段階に費やした。私はこの間、最終的なゲームにほとんど関係のない、オープニングのグラフィックを描くことによって、早くコードを書きたくてうずうずしている自分の手をまぎらわせていた。

あなたはこの段階で、ゲームに実装したい多様なアイデアを生み出すはずである。しかし、それらをすべて寄せ集めたとしても、うまく組み合わせることはできないだろう。アイデアは、実際に使用する前に、整理することが必要だ。あなたは、どのアイデアにも執着すべきではない。実装のアイデアの山は、デザイン段階での有用な資源である。しかし、これは絶対必用だというアイデアのリストにすることは、足かせにしかならない。アイデアは自由に出すべきだが、実際のデザインのときには、それらを容赦なくふるいにかける用意をしておくことが大切である。

たとえば最近私は、もう一人の人と共同で、会社のシミュレーションゲームをデザインした。研究段階の間に、ぜひともゲームに取り入れたいアイデアをたくさん思いついた。この時点でゲームがフェミニストの見地を持ち、かつお説教っぽくならないようにしようということに同意していた。我々は、厳しい上司、難しいプロジェクト、納期、人間関係、男性優越主義者、中立の男性、中立の女性、家族と家庭義務、助言者、昇進のための競争・・・と

いった要素も取り入れようと考えた。我々は最終的なデザインに、これらのアイデアのほとんど全部を含めることに成功したが、家族の要素を統合することはできなかった。全てのアイデアに公平な待遇を与えることはさすがにできず、結局、我々は入れたかった家族の要素を切り捨てなければならなかったのだ。

デザイン段階

これまでの段階で、概念的なゲームのアイデアは明確になっているが、その具体的な形式については何も決まっていない。ここまできれば、実際のデザイン段階に入る準備はできている。デザイン段階での主な目標は、互いに相互依存する三つの構造のアウトラインを作成することである。I/O構造、ゲーム構造、そしてプログラム構造だ。I/O構造は、コンピュータとプレイヤーの間で情報を伝達するシステムである。ゲーム構造は、プレイヤーがゲームの中で制覇する障害を定義する因果関係の、内部的な構造である。プログラム構造は、メインルーチン、サブルーチン、割り込みルーチン、そしてデータから成る、プログラム全体を構成する機構である。これら三つの構造は協調して働かなければならないから、同時に制作する必要がある。ひとつの構造に対して何かを決定するとき、他の構造に影響がないかどうかチェックしなくてはならない。

I/O構造

私が好きなのは、三つの構造の中で最も制約が大きい、I/O構造から作り始めることだ。I/Oは、コンピュータとプレイヤーの間を取り持つ言葉であり、人間の言葉と同じように、我々が仲間と共有しようと望む思考や、理念や、感情の洪水といったものを集めてきて取り込む、じょうごのような存在である。I/Oによって、できることとできないことが決まってくるだろう。

I/Oは、入力と出力で構成される。人間の言葉と違って、そのふたつは対称的なものではない。コンピュータは、人間への出力の方法として、グラフィックとサウンドのふたつを持っている。将来、もっと変わった出力装置が現れるかもしれないが、差し当たり、このふたつが最も普遍的なものである。

人間は聴覚よりも視覚のほうを重視するので、サウンドよりもグラフィックのほうが重要だと思われる。このため、多くのゲームデザイナーが、そのエネルギーの大部分を、きれいな画面を作成することに費やす。それどころか、一部のデザイナーは、最初に表示部を設計してからゲームを開発するので、目標を見失ったゲームデザインになりやすい。

ただ見せびらかすためだけに格好良いグラフィックを作ろうとするのは、ありがちな間違いである。グラフィックは、プレイヤーに情報を伝えるという目的のために存在するのだ。プレイヤーとのコミュニケーションを、説得力と感動を伴ったものにするためだけに、グラフィックを使うべきである。他のどのような理由のためでもない。ゲームが再現する空想世界をサポートするために、重大な情報を伝達する、機能的で意味のあるグラフィックを設計することだ。ゲームデザインのお粗末さを、グラフィックでごまかそうとしてはいけない。もし、ゲームが単調で退屈であるなら、どんなに素晴らしいグラフィックも、それを包み隠すことはできない。最悪なのは、退屈なゲーム部の代わりに、格好良いけれど無意味なグラフィックをでっちあげることである。また、サウンドも同じルールに従い、ゲームで何が起きているのかをプレイヤーに伝えるために使うべきである。印象的だが無意味なグラフィックやサウンドが唯一役立つ場面は、ゲームのはじめに、雰囲気や気分を盛り上げるのを手伝う場合に限られる。

絵コンテというものは、映画産業で確立された技術であり、多くのゲームデザイナーを誘惑する、グラフィックのデザインツールである。それはしかし、ゲームにはふさわしくない。なぜかという、絵コンテは本質的にシーケンシャル（連続的）な技術だからである。ゲームはシーケンシャルではなく、ツリー構造である。本質的にシーケンシャルなツールを使うゲームデザイナーは、無意識のうちに、自分のデザインをシーケンシャルにしてしまう。道具は使用者の心に影響を与えるのだ。のこぎりを持てば木を切りたくなくなるし、フリーウェイに乗れば、目的地に行くより、どこまでもドライブしていたい気分になる。同じように、絵コンテを使うと、ゲームが自然とシーケンシャルな代物になってしまうのである。

ゲームの入力には、特に注意を払わなければいけない。入力、プレイヤーの触覚との接点である。人間は触覚を非常に重視する。あなたは、プログラマがキーボードの手触りにこだわることに気付いたことがあるだろうか。プレイヤーは、あなたのゲームに対してそれと同じように感じる。良い例が『ジョーブレイカーズ』（JAWBREAKERS）や『ムスクアタック』（MOUSKATTACK）だ。これらのゲームでは残念なことに、ジョイスティックを斜めに入れたときの挙動があいまいである。これは、プレイヤーに、ジョイスティックの操作性が悪いという印象を与える。私は、プレイヤーがいらついでジョイスティックをたたきつけ、もう二度とプレイするものかと言っているのを見たことがある。あなたが入力を設計するときには、それがプレイヤーをいらつかせ、怒らせるかもしれないということを忘れてはいけない。

入力は、すべてのゲームデザイナーが直面する、根本的なジレンマの中心に横たわっている。優秀なゲームというのは、プレイヤーの性格をゲームの中に十分に反映させるため、対抗者との頻繁な相互作用を可能にするものだ。それには、プレイヤーが彼の性格のニュアンスを表現できるだけの、多数の意味を持つ十分な選択肢が必要である。プレイヤーの決定は入力されなければならないから、プレイヤーを怖じけさせるような、大規模で複雑な入力が必要になってしまう。我々のジレンマというのは、優秀なゲームは、複雑な入力を必要とするように思われるということである。

多くの選択肢を可能とするシンプルな入力方法というジレンマを解決するためにはデザイナーの創造力が必用である。これは容易なことではない。満足のいく解決方法が見つかるまで、多くの案を出しては捨てるという試行錯誤が必要である。しかし、不可能ではない。私が『スクラム』（SCRAM: 原子力発電所シミュレーション）をデザインしたとき、私は次のような問題に直面した。プレイヤーはジョイスティックだけで、どうやって原子力発電所の全体をコントロールすれば良いのだろうか。一見、これは絶望的に思われる。だが、私はやがて大変うまく動作する入力法を発見した。プレイヤーは、発電所の表示の中でカーソルを動かす。コントロールできる装置にカーソルをあてて、オンにしたり力を強めたりするにはスティックを上、オフにしたり力を弱めたりするにはスティックを下に入れるのだ。このシステムはシンプルで簡単であり、プレイヤーはそれを見ただけで理解できる。

理論的なレベルにおいては、選択肢の豊富さと入力の綺麗さのジレンマに対して、普遍的な解が存在する。私はこの解のことを「網の目作用」（the webwork）と呼んでいる。網の目作用のゲームをデザインするためには、少数のコマから始める。そして、すべてのコマのペアに当てはまる関係を定義する。コマの関係が集まって、網の目作用を構成するのだ。網の目作用は容易に複雑にすることができるが、網の目作用を作るためには少数のコマで十分である。一般に、 N をコマの数としたとき、ペアの数は $N*(N-1)/2$ である。したがって、4コマで6組のペアができ、8コマで28組のペアができ、16コマでは120組のペアができるわけだ。コマの数が少なければ、ゲームにおいての関係の豊富さを犠牲にすることなく、プレイヤーが直面するI/Oの問題が少なくてすむことになる。

バックギャモンは、網の目作用によって、いかに小数のコマで複雑な関係を作ることができるかを示す好例である。バックギャモンには、たった³⁰のコマと、それらが置かれる²⁶のマスがあるだけだ。コマの間の関係はかなりシンプルで、動いたりぶつかったりする能力で表現される。それでもなお、どんな規定された動きでも、それぞれのコマが、ボード上の他の多くのコマと、攻めている、守っている、ふさいでいる、ふさがれている、などの関係を持っている。コマの関係はそれだけに限らない。なぜなら、コマの前にある、ほとんどすべてのマスは、サイコロの目によって届く可能性があるからだ。プレイングエリアの長さ（24マス）がちょうどサイコロの最大の目に等しいことは、偶然ではない。互いの射程範囲内にすべてのコマを位置させるため、そうでなくてはならなかったのである。それによって、重要なペア関係の数が最大になっているのだ。

たいていの網の目作用ゲームは、空間的に表現された網の目作用に頼っている。描写するのが簡単だし、プレイヤーが思い浮かべるのも簡単だからである。しかし少数のゲームは、空間的でない網の目作用を持っている。私の『ゴシップ』（GOSSIP）もそのようなゲームのひとつである。奇妙なことに、ゲーム自体の網の目作用は空間的ではないのに、内部計算では空間的な網の目作用を使うのだ。このことは、ゲームの網の目作用は本質的に空間的なものであることを意味するのかもしれないし、私が空間的な網の目作用の固定観念から抜け出せていないことを意味するのかもしれない。

入力装置に何を使うかという選択は、デザイン上非常に重要である。私は、良いゲームデザイナーは、入力のためにキーボードを使うことは避け、ひとつのシンプルな装置、つまりジョイスティック、パドル、マウスなどに限定すべきだと考えている。これらの装置の価値は、キーボードと比べての直接的な優越から生じるのではなく、デザイナーを抑制することにある。シンプルな装置を使用すれば、自然とシンプルな入力ができるし、複雑な装置は、複雑な入力を促進するのである。

I/O構造は、プレイヤーが見るゲームの顔であるから、三つの構造の中で最も重要である。それはゲームの相互作用のための媒介なのだ。それはまた、三つの構造の中で最も難しいものでもある。人間の感受性と、コンピュータの技術的な熟達の両方を要求するからだ。相応の注意を払わなくてはならない。

ゲーム構造

ゲーム構造の設計において中心的な問題は、目標と題材の空想を、どのように実行可能なシステムの中に蒸留すべきかを理解することである。ゲームデザイナーは、題材の中からキーとなる要素を見つけ、そのもとでゲームを構築しなければならない。このキー要素は、題材の中心であり、ゲームにおいて述べられる問題を象徴するものであって、かつ操作しやすく、そして理解しやすくなければならない。たとえば『東部戦線¹⁹⁴¹』では、私は近代戦争の巨大な複雑さの中から、ひとつのキー要素を抽出した。「移動」である。移動が、軍のユニットの性質を規定する。敵の位置に移動すれば戦闘が始まる。敵の背後に移動すれば、補給経路や逃走経路を絶つことができる。都市に移動すれば占領することができる。移動は、戦争のすべてを表現できるわけではないが、多くの局面が表現可能なキー要素である。それは簡単に操作できるし、すぐに理解することが可能である。

『ゴシップ』では、いっそう難しいゲームデザインに挑戦している。このゲームは、社会的な人間関係を扱う。主題の複雑さと人間の相互作用の捻じ曲がり具合は、主題がゲームの振る舞いを超えたところにあることを示唆している。熟考の後、私はキー要素を抽出すること

ができた。それは「親近感の主張」である。社会の相互作用の多くは、余分なものを取り払えば、次のふたつのどちらかに要約することができる。「私はどちらかということサンドラの方が好きだ」という一人称の主張と、「トムはサンドラの方がちっとも好きではないと言っていたよ」という三人称の主張のふたつである。キー要素がなかなかうまく具合に、人間の相互作用を要約しているのがわかるだろう。それは簡単に操作できる。定量的ということだ。そして理解もしやすい。人間の相互作用から、「親近感の主張」をキー要素として分離したことが、『ゴシップ』というゲームを可能にしたのだ。

操作性は、ゲームの成功にはとても重要である。キー要素は操作しやすくしなければならない、しかし非常に限定された方法で。それは表情豊かに操作しやすくしなければならない。すなわち、プレイヤーが、ゲームの空想世界を体験する上で欲する、あるいは必要とすることを行ない、彼自身を表現することができなければならないのだ。たとえば、コンバットゲームは、ほとんど常に「射撃」がキー要素である。もし、撃つことに関してプレイヤーの自由がひどく制限されているとしたら、プレイヤーは空想を楽しむどころではない。同時に、操作性は簡潔でなくてはならない。ふたたびコンバットゲームの例を出すと、もし弾を撃つたびに火薬を消費するなどと言われたら、プレイヤーは、操作性に障害があると感じるだろう。操作性は、ゲームの空想世界をサポートする上で意味のあるものでなくてはならない。最後に、操作性は、プレイヤーの選択がキー要素を操ることと密接に関連することに、焦点が合わせられる必要がある。たとえば『ゴシップ』で、キー要素（親近感の主張）は、憎悪から愛に至るまでの値が、直線的に並んでいることを想定している。『エナジー・ツアー』(ENERGY CZAR: エネルギー経済学シミュレーション)は、この原則に違反している。プレイヤーに、巨大でまとまりがない選択肢から選ぶことを要求するからだ。メニュー構造やキーボードの使用は、キー要素に焦点が合っていないことから生じるのだ。

複数のキー要素を使用するゲームも多い。たとえば、たいていのコンバットゲームは「移動」と「射撃」の両方を含んでいる。これは必ずしも悪いことではない。両方のキー要素がシンプルさを保てるなら、あるいはひとつのキー要素が首位を保つなら、ゲームは成功する。しかし、これらの原則に違反しているキー要素があまりに多ければ、焦点の定まらない、ピンボケのゲームになってしまうだろう。

I/O構造を作る上での主な問題は、制約を克服することであり、ゲーム構造を作る上での主な問題は、可能性を実現することである。I/O構造におけるあなたの仕事は、ゲーム構造における限界を定義する。プレイヤーは直接それに遭遇しないだろうから、あなたは内部の構造でもっと多くの自由をとることができる。たとえば、私は『タクティクス』(TACTICS)というゲームで、鎧を突き刺す一撃の効果を現実的に計算する、非常に複雑な戦闘アルゴリズムを開発した。もし私がそれを説明しようとしていたら、このアルゴリズムの複雑さは、プレイヤーを混乱させることになっただろう。しかし、プレイヤーはアルゴリズムの内部の働きを理解する必要はなく、ただその効果を把握していれば良いのである。したがって私は、シンプルで直感的なアルゴリズムを設計することを強いられるようには感じなかった。

ゲームがリアルな感触を伝達することを保証する、十分な特色を提供することに集中しよう。ディテールを加える間、あなたのバランス感覚を保持することが大切である。ディテールの正確さを追求するあまりに、他の部分で基本的な要素を見落としてしまったら、それだけでゲームは台無しになってしまう。

あまりにも多くの要素を、ゲーム構造にどこどかと積み上げるというミスを犯すゲームデザイナーは多い。その結果、過度に複雑で例外だらけのゲームができてしまう。第四章で論じたように、例外処理は望ましくない。ゲームはシンプルさと表現力を兼ね備えていなくては

ならない。第四章では述べなかったことだが、例外処理には、第二の問題点がある。それは、ゲームのI/O構造を台無しにしてしまうということである。たとえば、『スター・レイダース』の長距離スキャンは、なかなか良い追加能力を提供するが、それは、プレイヤーが覚えなければならないキー操作がひとつ増えてしまうということでもある。これは望ましくない。幸い、この問題は『スター・レイダース』では無視できる。なぜなら空想世界の中ではプレイヤーは「宇宙船のコクピットに座っている」のだから、コントロールレイアウトの複雑さは、障害よりもむしろ、宇宙船を操縦しているという空想をサポートしてくれる要素となりうるからである。ゲームデザインを進めるうちに、I/O構造の品質を維持するために、ゲームに盛り込みたいアイデアを断念することを強いられることがあるだろう。逆に、どうしても捨てたくないゲーム要素を取り入れるために、I/O構造を変更することを強いられることがあるかもしれない。そのときには、ただ単に、新しいコマンドを追加するだけではない。I/O構造全体を見直して、新しいコマンドがうまく適合するように修正することが必要である。

ゲーム構造のデザインは、I/O構造のデザインとは感情的に非常に異なる。I/O構造をデザインするときには、ハードウェアの限界という名の嵐がデザインを前後左右に揺さぶる中で、表現能力という名のスキラの巨石と、表現の明快さという名のカリブディスの大渦に挟まれた、危険な航路を通り抜けなければならない。一方、ゲーム構造をデザインするときには、デザイナーは、水平線に向かって果てしなく広がった海の上にいる。いま突きつけられている課題は、「どちらへ行くのか」ということである。

プログラム構造

プログラム構造は、ゲームデザインの中で注目すべき第三の対象である。これは、I/O構造およびゲーム構造を、実際の製品に変換する媒介である。プログラム構造の最も重要な要素のひとつは、メモリマップである。重要なタスクに多くのメモリを割り当てなくてはならない。このような予防措置をとらないと、つまらない機能に大量のメモリを費やしてしまい、重要なタスクに使うためのメモリが十分に残っていないというはめになる。同じように、重要な変数とサブルーチンの定義も必要である。最後に、プログラムの流れに関する若干の文書も重要である。フローチャートとかWarnier-Orr図とか、何でも構わないが、あなたの好みに適したものを使うことだ。この本はプログラミングの本ではないから、あなたがプログラム開発においてガイダンスを必要とするなら、そういった専門書を読むと良いだろう。

デザインの査定

さて、ここまでで、三つの構造ができあがった。I/O構造、ゲーム構造、そしてプログラム構造だ。あなたは、三つの構造すべてが共存できることを確信しているはずである。次の工程は、ゲームを壊すようなデザイン上の欠陥がないかどうか、デザイン全体を査定することである。最初の質問は、このデザインは自分の目標を満足させるかどうかということである。ゲームは望みどおりのことをするだろうか。プレイヤーは本当に、こちらが意図したような経験をしてくれるだろうか。問題ないようなら、次に進もう。

ゲーム構造の安定性を調べよう。ゲームが動的なプロセスであることを思い出してほしい。ゲームが制御不能になるようなことはないだろうか。たとえば、そのゲームにお金の要素があるとしたら、プレイヤーが常識外れの大金持ちになってしまうような欠陥はないだろうか。つまり、ゲーム構造は、すべての値を妥当な範囲に収めることを保証しているかということである。万一そのような欠陥があるのなら、それを改善する構造的な変更を想定し、ゲーム構造を再検討しなければいけない。どうしてもやむを得ないときは、強引に解決する

必要に迫られるかもしれない（たとえば、IF MONEY > 10000 THEN MONEY = 10000 といった具合に）。

勝利への近道が予測できてしまわないかどうか検討しよう。プレイヤーがほとんど努力せずに確実に勝てる方法を見つけてしまったら、もう、そのゲームがプレイされることはないのである。プレイヤーが、あなたが望んだプロセスを必ず経験するように、すべての思いがけない近道が、うまく妨害できているかどうか確認することだ。妨害といっても、わざとらしくなく、妥当なものでなくてはならない。プレイヤーは決して、決められた道に誘導されていることに気付いてはならないのである。不自然な妨害の例が、『東部戦線』(WAR IN THE EAST)に見られる。このウォーゲームは、第二次世界大戦の東部戦線を扱う。ドイツはロシアに猛攻したが、彼らの前進はモスクワの手前で停止した。この状況を再現するため、デザイナーはドイツに圧倒的な優越を与え、補給の可能な距離を調節して、ちょうどモスクワの手前で足を引っ張られるようなルールを作った。効果は正しかったが、それを達成する手段はいかにもわざとらしくかった。

最後の、そして最も重大な決定は、ゲームの製作を中止するか続行するかという決定である。これは、プログラミングを始める前に決定すべきである。この段階で、製作を中止することを躊躇してはいけない。たとえ中止するとしても、いままでの努力に価値はあったといえるだろう。後々の段階になってギブアップする場合は、それこそ本当の損失である。致命的な損失なしで中止できるいまだからこそ、この選択を注意深く考えることが必要なのである。そのゲームにもう興奮を感じないなら、中止しよう。成功の可能性に疑問があるなら、中止しよう。首尾よく作り上げることに自信がないなら、中止しよう。私は自分のファイルに、100本近いゲームアイデアを持っていて、そのうち30~40本くらいのアイデアを詳細に検討した。しかし、デザイン段階で中止されなかったゲームはたったの8本しかないのだ。

プログラミング準備段階

ここまでに、アイデアを紙にまとめる準備ができたことになる。いままでに作った文書は概略だけであり、単なるメモや落書きのようなものだった。ようやく、ゲームの完全な仕様書を書く用意ができたのである。まず、これまでのすべてのデザイン結果を紙にまとめ、I/O構造とゲーム構造を定義しよう。この文書は、技術的な検討よりも、プレイヤーの経験することを強調したほうが良い。そして、この最初の文書を、事前に作成したプログラム構造のメモと比較し、必要に応じてプログラム構造を調整しよう。

プログラミング段階

この段階は、全段階の中で最も容易である。プログラミングは、他の何よりも多くを詳述するために注意を必要とする、単純で泥臭い仕事である。プログラマの技術が不足していたというだけの理由で失敗したゲームは、ほとんどない。プログラマが十分な努力をしなかったとか、仕事を急がせたとか、アセンブリ言語で書こうとしなかったとか、そういった理由でゲームの可能性を実現できないということはある。ゲームプログラミングにおいて決定的な要素は、才能があるかないかではなく、努力をしたかしなかったかというところが大きい。もし、プログラミングこそ、全てを決める一番重要な要素だと考えているのなら、ゲームデザインやシステム設計はやめておいたほうが良い。さあ、後はコードを書いて、デバッグするだけだ。

テストプレイ段階

理想的には、テストプレイは、ゲームデザインに磨きをかけ、洗練するための情報をもたらすプロセスである。しかし実際には、テストプレイは、修正しなくてはならない、ゲームデザイン上、プログラミング上の問題を明らかにする。そのため、テストプレイはしばしば、プログラムのデバッグ作業とごっちゃにされる。

テストプレイは、ときに、ゲームの致命的な欠陥を明らかにすることがある。致命的でない欠陥は普通、大した問題ではない。特色に欠けているとか、コマが多すぎるとか、アクションが足りないとか、プレイヤーに要求される計算が多すぎるとか、そういった問題は修正可能である。致命的な欠陥というのは、ゲームに絶対に必用な二つの要素が真っ向から対立してしまい、その不適合が予測できなかつたような場合に生ずる。あなたは、そういった致命的な欠陥を抱えたゲームを、思い切って捨てる勇気を持っていてはならない。プログラム済みのゲームにパッチを当てたとしても、せいぜい限定された利得を達成することができるだけだ。もしゲームがひどく奇形であるなら、手術よりも中絶のほうが望ましい。

テストプレイによって、重大だが致命的ではない問題が明らかになった場合、あなたは慎重に選択を検討しなくてはならない。手っ取り早いのが汚らしい、臭いものにふた式の仕事に陥る誘惑に屈してはいけないのだ。多くの場合、テストプレイで発見される問題は、真に根本的なデザインの欠陥の徴候にすぎない。詳細な分析を行って、問題の本質を見つけなければならぬ。いったん問題の本質を見いだしたなら、さまざまな解決方法を考えるのに、長い時間をかけるべきである。このプロセスを急いではいけない。ときに、理想的な解決は意外な角度から得られるものである。あなたのゲームの目標に、着実に近づけるような解決方法を選択しよう。あなたの目標に一番近づける解決方法の代わりに、最も簡単な解決方法を選択してはならない。

たとえば、『東部戦線1941』のデザイン中に、私は厳しい問題に遭遇した。プレイヤーが操作するには、あまりにもユニットの数が多すぎたのである。マップを縮小する方法を考えたり、単純にユニット数を減らせないか考えたりして時間を費やしたあと、結局、ゾーン単位のコントロールという方法に落ち着いた。ウォーゲームにおいてユニットの影響範囲を大きくする、標準的なテクニックである。この方法によって、ユニット数の問題を解決できただけでなく、兵站業務のルールをいっそう重要にし、ゲームの戦略性を豊かにすることができた。ユニット数を減らすというだけの目標で始めたわけだが、結果として、ずっと大きな改良点を見つけることができたのである。

あなたの最初のデザインがうまくいっていれば（または、あなたが単に幸運だったなら）、ゲームはこのような危機には直面しないだろう。その代わりに直面するのは、洗練の問題である。ゲームを構成するすべての要素は調子はずれで、あなたが思い描いていたような、しなやかなヒョウのように動いてくれず、酔っ払った恐竜のように動くことだろう。ゲームの調整には何週間もの時間を要する。あなたが他の問題に取り組んでいる間は、それほど調整はしなくて良い。ゲームの調整は、すべての要素にデリケートに依存するので、ゲームに何か変更を加えれば、それまでの調整がムダになってしまうからである。したがって、洗練段階の本当の終わりまで、最終調整の仕事は持ち越したほうが良い。

実は、テストプレイにはふたつの形式がある。ひとつめは、デバッグの最終段階でなされる、あなた自身のテストプレイで、ふたつめは、あなたがゲームを他のテストプレイヤーに引き渡してからの話である。このふたつの重要な違いは、発見されるバグの性質にある。あなた自身のテストプレイでは、すべてのプログラムバグ（プログラム構造の欠陥から生じるもの）と、多くのゲームバグ（ゲーム構造の欠陥から生じるもの）を明らかにして、排除すべきである。あなたがテストプレイヤーに見せるゲームには、プログラムバグがなく、彼ら

はゲームバグだけを発見するというのが望ましい。テストプレイヤーにわざわざ不完全なゲームを見せることはないし、それどころか、あまり早い段階から見せてしまうと、彼らの客観性を損なってしまう恐れもある。ゲームの完成に近づいて、あなた自身の改良案が減ってきていると感じるときがくるはずだ。そのときこそ、少数の選り抜きのテストプレイヤーに、ゲームを見せるときである。

テストプレイヤーは、注意深く選択しなくてはならない。そのあたりの友人たちをつかまえて、ゲームについてどう思うかと彼らに尋ねたりすることはできない。ゲームに精通し、何がしかの経験に基づいて、あなたのゲームを分析し批評することのできる人材が必要である。理想的には、テストプレイヤーは、彼ら自身ゲームデザイナーであるのが望ましい。彼らであれば、良いゲームデザインに欠かせない、トレードオフの評価を共有することが出来るからだ。あなたはまた、彼らの性格とゲームの嗜好を良く知っているべきである。テストプレイヤーの人数は5~6人とどめたほうが良い。それ以上だと、各テストプレイヤーの反応を慎重に吟味することができなくなる。

これまで試されたテストプレイの形式は、他にもいろいろある。一般の人たちを集めてきてゲームをプレイさせ、彼らの反応を評価することに頼っていることも多い。私は、このようなシステムは良いとは思わない。それは確かに、科学的で、客観的で、民主的であるけれども、消費者というのはお粗末な批評家であるゆえ、彼らはめったに有用なデザイン情報をもたらしてはくれないのだ。彼らの提案は、無意味で非実用的である。彼らは、実用的な提案ができるだけの、コンピュータやゲームについての知識を持っていない。こういう方法は、洗剤とかシェービング・クリームなどではうまくいくだろう。しかし私は、優れた映画や本や音楽が、この種の市場調査によって作られたということに対して非常に懐疑的である。このような方法が、才能のないデザイナーが安っぽいゲームの大量生産をする場合に役立つということは理解できる。それは認めるが、この本は、そういう考え方をする人には向けられない。

テストプレイヤーたちは、事前にゲームのマニュアルを必要とするだろう。それはゲーム本体ほど完成されたものである必要はなく、テストプレイヤーが、ゲームで何をすれば良いのかわかるだけの情報があれば十分である。マニュアルを見れば解決できるような問題を批判されて、時間を浪費しないようにしておこう。テストプレイヤーの隣に座って、ゲームをしながら教えたりしてはいけない。そんなことをすれば、彼の客観性を汚染してしまうだけである。テストプレイヤーの最初の反応は、マニュアルのわかりやすさを示すベストな情報なのである。実際にテストプレイヤーに会って話を聞く前に、一週間くらいゲームをしてもらうと良い。ただし、プレイの様子を、こと細かく記録するように頼んだりしてはいけない。彼はそんなことをしないだろう。その代わりに、心配な点に関して、マニュアルに若干のことを書いておけば良い。ゲームでどの選択肢を選んだかということと、それに続くスコアの単純な記録さえ書面で尋ねれば十分である。

テストプレイヤーに、ゲームを理解するのに十分な時間を与えたら、インタビューを行おう。そのとき、すべてのテストプレイヤーに尋ねる質問を、あらかじめ用意しておくといい。テストプレイヤーの回答を誘導してはいけないし、賞賛を求めてもいけない。あなたの仕事は欠陥を見つけることであって、賞賛を受けるのはもっと後の話である。インタビューに第三者を使うことはいっそう科学的であるが（それにより、いっそう公正な回答が期待できるから）、その場合は、テストプレイヤーの間に仲介役が必要となる。私は、テストプレイヤーから直接情報を受け取るほうが良いと思う。私はまた、インタビューの間、わざと自分から否定的な態度を取り、テストプレイヤーとともにゲームを批評して、その改善案が出されるように仕向けることを好む。

テストプレイヤーの批評は、評価することが困難である。たいていの批評は、さまざまな理由で否定されなくてはならない。いくつかは、あなたの目標と相容れないものだろうし、いくつかは、残っているメモリ容量では実現できない。いくつかは合理的であるが、利得と釣り合わないような、大掛かりなソフトウェアの改造が必要になるだろう。なされた提案の90%を否定することを躊躇してはいけない。残った10%は適切な提案なのだから、それを実装して、時間を浪費しないようにしよう。では、どうやってその10%を見分ければ良いのか。それは知恵や分別の問題であり、私には何ともいえない。

ゲームデザインの最終段階は、ゲームに磨きをかけることに尽きる。洗練段階は、実はテストプレイ段階の後半と同じであり、そのいくつかは、テストプレイヤーを伴っても良い。この段階は非常に重要である。デザイナーはいままで長い間ゲームに取り組んできて、新しいデザインの輝きは次第に薄れてきた。いまとなつては、数ヶ月前に終わられるべきだった大仕事にすぎない。テストプレイヤーはゲームに夢中になっているし、出版社もこのゲームを気に入り、いますぐ欲しがっているのだが、当のデザイナーはうんざりしているのだ。この厄介物を投げ出してしまいたいという衝動は強烈である。この衝動に抵抗して断固として維持しなくてはならない。そして、ひたすら磨き続けよう。ゲームをテストし、調整し、小さな装飾を加え続けよう。ドアから外に出したら最後、永久に戻ってはこないのである。私が手がけたゲームは、すべて同じパターンに従った。私は、これでもかというくらいにゲームを磨いた。うんざりして、もう二度と見たくもなくなるくらいにである。ついにゲームを送り出したときは大喜びだった。ついに、あの厄介物から解放された。私は自由だ。しかし、一ヶ月と経たないうちに後悔して、私が気付かなかった恥ずかしいバグをひとつ修正するチャンスがあれば良いのにと思っていた。三ヶ月のうちに、もっとたくさんのバグが発見され、私の後悔は羞恥心に変わっていた。そのころには、そのプログラムが広く知れ渡らないことを願っている始末だった。

ゲームをリリースする前に行わなくてはならない最後の仕事のひとつが、ゲームのマニュアルを準備することである。マニュアルは、コンピュータゲームに関係するすべての人たちから、しばしば軽視される。これは大間違いであつて、マニュアルは、ゲームのパッケージ全体の中で、極めて重要な要素である。コンピュータには多くの限界があるが、いくらかはマニュアルで補うことが可能だ。ゲームに関係する不変の情報の多くは、マニュアルで提供することができる。マニュアルはまた、絵やバックグラウンド・ストーリーのような、空想世界をサポートする要素を加える恰好の場所である。そして、うまく書かれたマニュアルは、ゲームでしばしば発生する誤解の多くを解くことができるだろう。

マニュアルは、デザイナー自身が執筆しなければならない。作文能力があるかどうかにかかわらず、また、プロのライターが最終的なマニュアルを用意することになっていたとしてもである。自分でマニュアルの執筆を試みることで、プロのライターの技能への敬意が増し、生産的な人間関係を持てる可能性が高くなるだろう。また、綺麗なゲームデザインのために有用なフィードバックが得られるだろう。不器用なデザインは記述が難しいのに対して、クリーンなデザインは記述が容易だからである。最終的にこのマニュアルは、プロのライターのための有用な情報源となるだろう。ライターがあなたの書いたマニュアルを捨てて、すべてを始めることができるようにしておこう。素人の努力にプロが磨きをかけるより、新たにマニュアルを書き起こしたほうがましだろうから。あなたは、ライターのすべての質問に、可能な限り完全に答えなくてはならない。デザイナーとライターがお互いに意思疎通を図って、助け合つてこそ、優れたマニュアルを作り出すことができるのである。

事後検討

プログラムが終わったら、批評家に備えて身を引き締めることだ。あなたの愛しいゲームを、彼らはその不潔な手でわしづかみにして恐ろしいことをしてくれるだろう。彼らは、ルールも読まずにプレイするだろう。ストラテジーゲームなら、彼らは、エキサイティングではないといって非難するだろうし、アクションゲームなら、知的な面で不十分だと判定するだろう。勝手に想像した技術的な欠陥を見つけて、あなたがなぜこのようなゲームを作ったのかという深層心理について、あれこれと的外れな憶測を始めることだろう。私の批評家の一人は、『タンクティクス』は明らかにきついスケジュールで急いで作られたと結論したが、実際には、開発を開始してから発売するまでの期間は5年と2ヶ月だった。また、『エナジー・ツアー』の場合は、彼の好きなアーケードゲームと比べてエキサイティングではなかったという理由で酷評された。こういった批評家の言うことを真に受けてはいけない。たいていの批評家は、プログラムを批評するだけの資格を持っていないのだから。コメントに耳を傾けるに値する思慮深い批評家は、ほんの一握りしか存在しない。3人以上の批評家の意見が一致している場合のみ、注意を払うようにするのが良いだろう。また、目標が批評家の好みでないときは、良い批評家でさえあなたを酷評するということを覚えていてほしい。

大衆はもうひとつの問題である。彼らがゲームを買ってくれない場合、あなたはふたつの意味で損をする。第一に、あなたとあなたの雇用者が儲からないということ。第二に、あなたのメッセージが人々に届かないということである。美しいメッセージを誰も聞いてくれなかったとしても、気に病むことはない。あなたは芸術家として失敗した。一回失敗したからといって心配するには及ばない。どんな芸術家だって、ときおり大失敗するものである。二回続けて失敗した場合はさすがに良くないので、三度目は、芸術的な価値を真剣に再考すべきである。さて、あなたは、気高いが貧乏な芸術家でありたいだろうか。それとも、いくぶん裕福な職人でありたいだろうか。自分の胸に聞いてみると良い。心の奥深くで、あなたが自分の目標を達成したことを知っているのならば、批評家と大衆を無視するのだ。

[目次に戻る](#) | [各章へ: 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9](#) | [次章に進む](#)

一人用ゲームのバランス物量作戦人工痴脳情報の制限スピードの調整バランスに関するまとめさまざまな対立関係対等なゲーム対等でないゲーム三角関係役者と間接的な相互作用スムーズな上達曲線もう少しで勝てるかもまとめ

芸術家なら誰でも、芸術を産み出すための自分だけのテクニックやアイデアというものを持っている。画家は、筆の使い方や絵の具の調合法、いかに対象物の質感を出すかということに気を配るし、作曲家はオーケストラ向きの編曲法、リズム、対位法といったものを勉強するだろう。一本のゲームを作り上げるためには、ゲームデザイナーにも、さまざまなスキル、テクニック、アイデアといったものが要求される。この章では、私がこれまでに使ってきたテクニックを紹介しよう。

一人用ゲームのバランス

プレイヤーが一人のゲームでは、おのずからプレイヤーとコンピュータが対戦することになる。しかし、コンピュータとプレイヤーの間には非常に大きな差が存在する。人間の思考方法が、多数の情報を頭の中にちりばめ、それを関連づけ、まとめていくという順序で進んでいくのに対し、コンピュータの思考方法は直接的で、わきみちを知らず、計算づくなのである。これがある問題を産み出す。コンピュータゲームがプレイヤーを楽しませるために作られている以上、コンピュータにとってではなく、人間にとって有利な土俵で戦わなくてはならないのだ。これは、コンピュータにとって大きなハンデとなる。計算やソートといったことが主体となるゲームでは、コンピュータは簡単に人間に勝てるだろうが、そんなゲームは少しも面白くはないだろう。コンピュータは、プレイヤー側のホームグラウンドで戦って、そんな不利な状況をなんとかしなくてはならない。どうすれば、プレイヤーにとってやりごたえがあるようなゲームをデザインできるだろうか。方法は四つある。物量作戦、人工痴脳、情報の制限、そしてスピードの調整である。

物量作戦

これはゲームバランスの調整法として最も良く使われている方法のひとつであろう。コンピュータは、不器用だが膨大な計算能力を持っている。この計算能力を使えば、思考パターンさえ単純なら大量の敵ユニットを同時に動かすことができる。この手を使っているゲームは数多い。『スペースインベーダー』(SPACE INVADERS)、『ミサイルコマンド』(MISSILE COMMAND)、『アステロイド』(ASTEROIDS)、『センチピード』(CENTIPEDE)、そして『テンペスト』(TEMPEST)、といった有名どころのゲームは皆そうだ。コンピュータ側のユニットそのものは数を少なくして、その代わりにプレイヤー側ユニットよりも強力なものにしておくという手もある。『バトルゾーン』(BATTLEZONE)におけるスーパータンクが良い例だろう。どちらにせよ、この手法によるバランス調整の首題は同じだ。プレイヤーの思考上の有利をコンピュータの物量で調整しようというのである。

この方法はふたつの利点をもっている。ひとつは、プレイヤーとコンピュータとの戦いの中にダビデ (David) とゴリアテ (Goliath) の戦いのような小が大を倒すというシチュエーションを入れられることだ。明らかな負け犬をいじめるよりは、対等以上の相手を倒したいと思うプレイヤーは多い。もうひとつは、この手は、コンピュータ上でのプログラミングの負担が一番少ないということだろう。コンピュータに人工知能を演じさせるのは難しいが、同じことを繰り返させるだけなら単にループを組むだけで良い。もちろん、楽なだけに欠点もある。そう、皆がこの手を使っているのだ。そんな安易なゲームは、膝まで埋まるほどあちこちに氾濫している。このテクニクの流行は、ゲームデザインを考慮した結果と言うより、デザイナーの決断力のなさや怠惰さの結果だろう。

人工痴脳

物量主義以外にコンピュータ側に人間に対抗できるだけの能力を持たそうと思ったら、人工知能を採用するのが良いに決まっている。しかしながら、残念ながら現状では、人工知能はゲームなどに簡単に使えるほどには進んでいない。樹形図を用いたしらみつぶし法は、チェス、チェッカー、オセロといったゲームにおいて何とか使えるレベルにはなっている。しらみつぶし法が使えるゲームに関してはこの手で処理が可能だ。しかし、残念ながらそういったゲームは極めて少ない。

そこで、それぞれのゲームにあわせて簡易的な人工知能ルーチンを組むというテクニクが使われる。そういったルーチンは、人工知能と呼ぶにはあまりに原始的で単純であり、私はそれを「人工痴脳」 (Artificial Smarts) と呼んでいる。このテクニクは、『タンクティクス』 (TANKTICS)、『東部戦線1941』 (EASTERN FRONT 1941)、『リージョネア』 (LEGIONNAIRE) といったゲームに採用され、それなりにうまく働いた。この手法はゲームデザイナーにとっては非常に負担が大きい。人工痴脳にしたがうコンピュータ側のユニットの動きは、合理的で、しかもプレイヤーに読み切られないものでなくてはならないからだ。

人工痴脳の設計においてまず考えなくてはならないのは、ユニットに合理的な動きをさせるということだ。コンピュータが、タンクに断崖絶壁を越えさせようとしたり、自軍の宇宙船同士を体当たりさせたり、あるいは、プレイヤーの構えた銃の目の前で一休みしていたりすれば、プレイヤーは興奮めしてしまうだろう。いくら人工痴脳といっても明らかにおバカな動きをしてはならないのである。そういうと、デザイナーたちは、まず考えられるすべてのおバカな行動をリスト化し、そのひとつひとつについてそれを禁止するルーチンを書かなくてはならないというように考えがちである。しかし、それは間違った考え方だ。なぜなら、コンピュータは一見間違ったような動きから、思ってもみなかったような手を産み出すこともあるからである。より良い (より困難ではあるが) 方法としては、おバカな行動のほとんどをカバーしてくれるような、より普遍的なアルゴリズムを考えることなのだ。

それに加えて、人工痴脳の動きが人間に読み切られるようではいけない。コンピュータが次に打ってくる手が丸わかりになってしまうようでは、敵が知性をもっているという幻想が台無しになってしまうし、あまりに簡単に勝ってしまうだろう。これは、一見、先ほど述べた「理にかなった動き」という条件とは矛盾しているように思えるかもしれない。パターン化された定跡というのは読みやすいものになりやすいのだから。しかし、この表面上の矛盾は、ゲームにおける相互作用というものを深く考察することで解消されるだろう。ここでは、三つの観点をあげる。第一に、ゲーム上で打たれる手というのは、かならず、ある程度は対戦者の影響を受けたものになるということである。理にかなった動きをするプレイヤーは、対戦者のくせや性格から次の行動を予想しようとするものだ。第二に、相互作用というものは、もちろん相互に対称なものだ。両プレイヤーは互いに相手の動きを予想する。第三

に、相互作用は、それぞれのものが「ゲーム度」を量る基準となっている。この三つの観点は以下の喩えで理解できるだろう。ゲームは、二枚の合わせ鏡に喩えることができる。互いのプレイヤーは、それぞれの鏡を覗き込んでいるのだ。パズルは2枚の鏡が全く互いを映していない状態である。プレイヤーは静止した自分の顔を見ることだろう。多少なりともインタラクティブ性があれば、鏡は何度か反射して何人もの自分と対戦相手を映すに違いない。最高にインタラクティブな（つまり最高に「ゲーム度」の高い）ゲームでは、合わせ鏡は無限の像を映す。どちらのプレイヤーがちょっと違う動きをすれば、鏡に映る像は全く違うものとなるだろう。どんなに定跡だけを打っていても、無限に複雑な予測と予測の重なり合いが起きてその結果は全くわからないものとなるのだ。これこそ、合理的でかつ予測不能ということなのである。

残念ながら、完全にインタラクティブなゲームは、現行のパソコンレベルの処理能力では手が届きそうにない。コンピュータ側に完全なインタラクティブ性を持たせるためには、人間が次にどう対応するのかをコンピュータに読ませる必要があるが、それには、プレイしている人間の人格や性格といったものまでもコンピュータに分析させる必要があるからだ。さすがに、それは望み薄といった所だろう。現状では、もっと単純な方法に頼るしかない。私がこれまでに使ってきた手を例にあげよう。私の経験では、「限定」ルーチンのアルゴリズムほど読みやすいものはない。ここでは、ゲーム上のすべてのパターンの中で何かに特化したという意味で「限定」という言葉を使っている。「ウォーゲーム」を例にあげると、「最近接のユニットを探してそれを攻撃せよ」というアルゴリズムは、「限定」的であり読まれやすい。

私の経験では、最高のアルゴリズムとは、限りなく広がったさまざまな条件の中で、可能な限り多くの条件を検討しているアルゴリズムである。すなわち、少数の「限定」的な条件によるのではなく、できる限り多くの条件を加味するように作られたアルゴリズムを採用すべきなのだ。先ほどの例でいうなら「相対距離、移動力、砲塔の向き、友軍との距離、地形効果、射線などを分析して、こちらが破壊される確率と相手を撃破できる確率を計算し、撃破率と被害率の差が最も大きなユニットを探してそれを攻撃せよ」ということになるだろう。

では、そのような人工痴脳アルゴリズムを実現するにはどうしたら良いのだろうか。私にも一般的な方法があるとはとても思えない。少なくともまだ見つかってはいないだろう。私にできるのは、まずユニットの動きを得点化するような評価関数を作成し、実際にテストしながら、それを少しずつ変えていくという方法しかないだろうということだ。

評価関数は、まず、すべての可能な動きについて、そのメリット、デメリットを数値化できるように作成する必要がある。これはこれまでに作成されてきたすべての人工痴脳システムで有効だった方法である。しかし、評価関数の計算には膨大な計算が必要になる。第一の問題は桁数の問題である。デザイナーは、評価関数の計算の中で、1バイト（8ビット）で表現できる最大の数値を超えてしまって桁あふれを起こすことのないようにしておかなければならない。評価点が 2^{55} 点を越えるような手が複数あった場合には、どちらが最善手であるかを判断することができなくなってしまうのだ。まあ、この問題に関しては 16 ビットシステムがもっと一般的になれば問題ですらなくなってしまうだろうが。

評価関数を決める上でのふたつめの問題は、互いに矛盾するような要因にどのように評価点を割り振っていったら良いのかということである。戦車戦ゲームをデザインしたときには、次のようなことがあった。丘の上に登るのは良い行動であるというのは確かだった。しかし、道の上を移動するのもまた良い行動なのである。問題はどっちがより良いかだ。丘の頂上で行動を終えるのを 15 点にしたとして、道の上にいるのは何点にしたら良いのだろうか。こ

れは非常に答えにくい。正解を出すためには、そのゲームに関して本当に良く知っていなければいけないのだが、残念なことに、まだゲームが完成していない以上、最善手などわかりようもないのである。似たようなゲームに関する広い知識、再現すべき状況に関する深い理解、紙の上での分析、そして試行錯誤、そういったものに頼るしかない。

私が人工痴脳を設計するときを使うふたつ目の手法は、パターン解析である。この手は、各ユニット間に空間的な関係があるときにしか使えない。そのようなゲームでは、プレイヤーは、コマとコマの間の位置関係を把握し、次の手を決めるために一種のパターン認識を行っている。もちろん人間レベルのパターン認識をやるのはパソコンのレベルをはるかに越えてしまうが、それでも盤上のパターンを分析することで、パターン認識に近いことは可能になる。キーとなるのは、コンピュータにそれらしい動きをさせるために必要十分なだけの評価点を、コンピュータが計算できる形でそれぞれの場所に与えてやることである。たとえば、いくつかのウォーゲームでは、そこはどれくらい危険か、どれくらい安全かということをおぼろげに危険ポイント、安全ポイントというものを導入した。危険ポイントは、近くにいる敵ユニットすべてについて、各々の攻撃力をそのユニットまでの距離で割り、それを合計したものである。この危険ポイントは、近くの強力なユニットは危険で、遠くにいる弱いユニットは脅威にならないということを再現している。同じことを、友軍に関してやれば安全ポイントが定義できる。危険ポイントと安全ポイントを比べてやることで、あるユニットに行くか逃げるかを決定させることができる。一旦、この判断をさせておいた上で、可能な移動範囲の中でどこに行ったらより危険で、どこに行ったらより安全かを計算させる。行く気ならより危険な動きをすれば良いし、逃げるならより安全な動きをすれば良いのだ。パターン解析の手法を使うことで、ユニットに盤上での連携といったものを演じさせることができるようになる。

人工痴脳をうまく働かせるためのテクニックの中には、あまりに単純過ぎて詐欺っぽい手法もある。そう、ゲームのほうを手直ししてしまうのだ。評価関数に組み込みにくいような要因に関しては、ゲームから外してしまうのである。どうしても手に負えないような要因は、使うのをあきらめるしかない。たとえば、私は『タンクティクス』のデザイン中に湖の取り扱いでひっかかった。湖の岸にへこんだ部分があると、タンクが岸にぶつかっては戻り、岸にぶつかっては戻りという堂々巡りをしてしまうのである。そのとき使っていたアルゴリズムにとって、へこみのある湖はトラップになってしまっていたのだ。私は、へこんだ湖のまわりで妙な動きをしないようなアルゴリズムを作るために本当に長い時間を費やした。そして、ついにうまい方法に気付いたのである。問題の湖をマップから取り去ってしまえば良いということに……。

理想的には、熟練したゲームデザイナーはゲームのデザイン段階で、こいつは手に負えないから避けよう、といったような勘を働かせるべきなのだろう。しかし、実際には、実装段階でひどいことになっていることに気付いて、デザイン段階へと後戻りすることがままある。そういった経験が、デザイナーの勘を鍛えるのだ。

どんなに優秀なアルゴリズムであっても、それには適用範囲というものが存在する。特定のアルゴリズムは、特定の狭い条件下でのみうまく働くことが期待できる。しかし、面白いゲームをデザインしようと思えば、さまざまな状況を産み出すようにゲームをデザインしなくてはならない。すなわち、デザイナーは多くの場合いくつものアルゴリズムを作成してそれを状況の変化にあわせて切り替えていく必要があるのである。しかし、アルゴリズムの切り替えにはやはり危険が伴う。切り替えの前後でコンピュータ側の動きに連続性がなくてはならないからである。このアルゴリズムの切り替えに関しては、『リージョネア』をデザインしていたころの失敗がいまでも思い出される。コンピュータが操るバーバリアンは三種類

のアルゴリズムを与えられていた。「安全地帯へ撤退」、「接敵せよ」、そして「攻撃せよ」の三種類である。とある条件下ではこんなことが起きた。「接敵せよ」アルゴリズムにしたがって、プレイヤー軍に向かってひたすら突撃してきたバーバリアンが、「攻撃せよ」アルゴリズムに切り替わったとたんに、その攻撃が無謀だったことに気付いて、攻撃をすることなく「安全地帯へ撤退」アルゴリズムにしたがって尻尾を巻いて敗走を始めるのである。プレイヤーの眼前には、獰猛に突撃しては、きびすを返して狂ったように逃げていくバーバリアンの乱舞が繰り広げられる。誰も実際の戦闘には入れない。私は結局アルゴリズム切り替え方式をあきらめ、デザインをやり直した。アルゴリズムをひとつにまとめて「前進して攻撃せよ」にしてしまったのである。

数多くのユニットを一度に、コンピュータに移動させようとする、また別の問題が出てくる。いくつものユニットを整然と渋滞することなく移動させるにはどうしたら良いのかという問題である。ひとつの方法は、ユニットの行動順を決めてしまって他のユニットとぶつからないように動かしていく方法である。まず、ユニット¹番を動かして、次に²番、³番と空いている所に移動させていくのだ。もっとも私の経験では、この方法ではイライラするような大渋滞が、せいぜいのろのろ運転に取って代わるだけだ。もっと良い手としては、仮想的な移動を採用する方法がある。すべてのユニットを、仮想的な位置関係にしたがって一旦移動させてみるのだ。具体的にはこういうことになる。始めは、現状のユニットの位置関係を記憶する。次に、現在のユニットの位置関係を架空のボードにコピーする。そして、個々のユニットに関して、互いにぶつからないような架空の移動プランを探させるのだ。あるユニットに関して移動プランが決まったら、架空のボード上でそこに移動させてみる。次に別のユニットに関して同じように移動先を決める。すべてのユニットが動いたら、初めのユニットをもとの場所に戻して同じプロセスを繰り返す。こうやって、他のユニットの行き先を暫定的に決めながらベストな移動法を探していくのである。この膨大な計算を必要とするループはある程度の繰返し回数で打ち切って収束させる必要があるが、これによってコンピュータに基本的な移動条件を与えるだけで、各ユニットの行き先を決めることができる。この手法は、多数のユニットを渋滞させることなく移動させる必要があるときに役立つだろう。

ここまで述べてきたような人工痴脳のテクニックは、ユニット間に空間的な相互作用があるようなゲームにおいて有効なものである。しかし、またユニット間にそういった相互作用が存在しないようなゲームも多数存在する。そういったゲームでは、また違った人工痴脳が必要になる。空間的な相互作用が重要な意味を持たないようなゲームでは、複雑系のモデルに従った動作をさせることが多い。多くの場合、複雑系をモデル化するために、対になった差分方程式が導入される。『ルナ・ランダー』(LUNAR LANDER)、『ハンムラビ』(HAMMURABI)、『エナジー・ツァー』(ENERGY CZAR)、そして『スクラム』(SCRAM)といったゲームがそれらの代表格である。そういったゲームをデザインする上で一番気をつけなくてはならないのは、ゲームシステムの複雑さによってプレイヤーを打倒するのではないということである。大きなゲームに対になった差分方程式を持ちこむときには十分気をつけるようにしたい。『ハンムラビ』では、³セットの対になった差分方程式が使われた。まあ、これくらいなら何とかできるというプログラマーも多いだろう。しかし、システムは、差分方程式の数が増えるにしたがって急激に複雑化する。『エナジー・ツァー』では、なんと⁴⁸個もの差分方程式が使われた。こんな離れ技は、よほど制約が多いとき以外には使いたくはないものだ。普通は、⁴セット程度に止めて置くようにしたほうが良い。もし、どうしても多数の差分方程式を使わなくてはならなくなったときには、連立差分方程式の形で書けないかどうかを検討してみたほうが良いだろう。パラメータを行列表示にして、簡単な同一の方程式に適用するのだ。

私の経験からいうと、バランス調整のためには、差分方程式には前に進めようとするドライ

ピングファクター Drv と押し止めようとするダンピングファクター Dmp のふたつの因数が必要で、

$$f(n+1) = f(n) + Drv/Dmp$$

といった形で表現される必要がある。

ダンピングファクターを小さくとると、あちこち跳ね回るような、変化の激しいシステムになる。一方、ダンピングファクターを大きくとると変化が乏しく停滞したイメージになる。気をつけないといけないのは、ダンピングファクターを安易に決めて安心してしまうと、とんでもないことになる場合があるということである。 $f(n+1)$ とドライビングフォースの間でネガティブフィードバックがかかっている場合には、ダンピングフォースを大きくとりすぎると、ネガティブフィードバックが抑制されてしまい、パラメーターが発散する可能性がある。差分方程式の動作はけっこうややこしい。興味があるデザイナー志望者は、数学の本を読んで、ダンピングファクターが大きな場合、小さな場合、丁度中間的な場合について、どのような解が得られるのかを調べてみると良いだろう。もっと一般的な差分方程式の解法に関しても、数値解析の教科書を読むことをお勧めする。

こういったテクニックを用いることで、コンピューター側がある程度の知能を持っているかのように見せかけたゲームを作ることは可能である。しかし、あまり高望みはしないほうが良い。どんなに苦労したところで、本当の意味での知性をコンピューターに与えることはできない。私の示した三つの手法のいずれも、それだけでは、プレイヤーに歯ごたえがあると思わせるのには足りないのだ。実際には、プレイヤーの実力の半分にも満たないといったところだろう。

情報の制限

コンピューター側の思考力の不足を埋め合わせるには、プレイヤー側に与える情報を制限してしまうという方法もある。さすがに十分な情報なしでは、プレイヤー側も問題を簡単に解いてしまうというわけにはいかない。このテクニックもさじ加減が重要である。あまりやりすぎると、ゲームが運だけで勝敗が決まってしまうようなものになってしまう。しかしながらこの手を使えば、コンピューター側の勝率を五分五分にまで持っていくこともできる。もし、情報操作に適切な理由がつけられれば（たとえば、プレイヤー側が斥候を出すまで敵軍が見えないとか）、情報が制限されていても不自然にはならない。

情報を制限することには、コンピューターを強くする以上のメリットもある。未確認の情報を小出しにすることで、プレイヤーの想像力をかきたてることができるのだ。これをうまくやるためには、プレイヤーに与える情報は意識して吟味して置く必要がある。情報を細切れかつでたらめに与えられたのでは、プレイヤーは、じらさせるのではなく混乱していらついでしまうのだ。いきなり裸の女性を見せつければ男の目をひくことは間違いないが、着飾った女性が少しずつ服を脱いでいったほうがずっと色っぽく見えるだろう。初めからあちこちに穴の空いてる服を着てる女性？ それは、ただみっともないっていうんじゃないかい。

スピードの調整

スピード調整によってバランスをとることもできる。確かにプレイヤーの思考能力はコンピューターに比べて優れているが、単純計算を大量かつ高速に行うことにかけては、コンピューターのほうがはるかに上である。ゲームのスピードが速ければ、プレイヤーも落ちつい

て考えることができず、あわてふためくことになるだろう。この手法は実現が容易なため、非常に多くのアクションゲームのデザイナーに採用されている。

しかし、私はやたらとスピードをあげることで難易度を調整するのはまずいと考えている。展開の早いゲームは、どうしてもクリアまでの時間が短くなり、プレイヤーにとってのやり込みの要素がなくなってしまう。やり込みの要素のないゲームは、やりがいのないゲームなのである。コンピュータゲームにおけるスピード感というのは、一夜限りの恋のようなものだ。そのときは楽しいが、めったに新婚旅行まで発展することはない。繰り返して言おう。スピードによる難易度調整は安易に行うべきではない。

バランスに関するまとめ

これら四つのバランス調整のためのテクニックは、決してバラバラに使われることはないだろう。多かれ少なかれ、すべてのゲームの中でこれらのテクニックは組み合わせて使われている。普通は主に物量作戦とスピードで難易度を調整していて、人工痴脳や情報の制限をうまく使っているものはほとんどない。だからといってこれらすべてのテクニックを一度に使ってはいけないという理由はどこにもない。むしろ、それはもっとゲームを素晴らしくするはずである。それぞれのテクニックをうまく組み合わせることで、それぞれの弱点を補いあうことができるのだから。デザイナーは、目的のゲームを完成させるために、どのテクニックをどう使うかということを考えていかななくてはならない。

さまざまな対立関係

どんなゲームであれ、各プレイヤーが自分に最大の利益をもたらそうとしている限り、そこにプレイヤー間の関係が生じる。これは、ゲームにおいて主要な役割を演じている。このプレイヤー間の関係がどんなものであるかによって、ゲームの基調が決まるのである。コンピュータゲームでは、プレイヤー同士の関係は非常に単純な、質、量ともに限られたものであることが多い。ここでは、より面白いゲームを作るためにプレイヤー間の関係についてより詳しく考えてみよう。

対等なゲーム

一番単純な関係は、プレイヤー同士が対等な場合だろう。どちらも同じパラメーター、長所、欠点を持っている。プレイヤー同士が対等に作られたゲームは、自動的にゲームバランスが取れるという点で明らかに優れている。また、プログラミングも楽だ。両者に同じルーチンが使えるのだから。おまけに、ゲームのルールも理解しやすく、慣れるのも早いだろう。この手のゲームの代表としては、アタリ2600 (ATARI 2600) の『コンバット』 (COMBAT) や、バスケットボール、そして、グレイ・チャン (Gray Chang) がデザインした『ドッグ・デイズ』 (DOG DAZE) などがある。

プレイヤー同士が対等なゲームにもさまざまな弱点が存在する。最大の欠点のひとつは、すべてが全く対称的であるということだ。もし、本当に有効な必勝法というべきものが存在したときには、どちらのプレイヤーも同時にそれを実行することができるのである。そうになると、もはや戦略をたてることに意味はなくなってしまい、どちらが先に必勝法に沿ってプレイしたかだけの問題になってしまうのだ。

その代わりと言ってはなんだが、この手のゲームでは勝利は非常に僅差になりやすいともいえる。たとえばチェスでは、ポーン一個の差で勝敗が決まるようなことも良くあるではない

か。

対等でないゲーム

以上で述べたように、プレイヤー同士が全く対等なゲームにはいくつかの弱点があるため、あえてプレイヤー間に不平等を持ちこんでいるゲームも多い。そういったゲームでは、個々のプレイヤーはそれぞれ全く異なる弱点と利点を持っている。このようなゲームをデザインするときには、デザイナーは同レベルのプレイヤー同士のゲームで勝率がほぼ拮抗するようにゲームバランスをとる必要がある。一番簡単なのは、初期条件をある程度変更できるようにすることだろう。基本的には対等なゲームにしておいて、初期条件をいくつかの制限のもとで選べるようにするのである。アヴァロンヒル (Avalon-Hill) のボードゲーム、『ウィザーズ・クエスト』 (WIZARD'S QUEST) を例に上げよう。このゲームではプレイヤーはゲーム開始時に、互いに接してはならず、飛び地を作ってはならないという条件のもとでいくつかの領地を選ぶことができる。こうすることにより、(プレイヤーは、ゲーム開始時にN個の領地を持っているという) 対等な条件が、(プレイヤーAは、領地を一直線に並べ、Bは中央に固めたというように) 対等でない条件に変わったのである。この不平等は、プレイヤー自身が作り出したものであり、たとえその結果がどんなに不利になろうとも、プレイヤーは自分自身を責めるしかない。

もっとあからさまに不平等なゲームもある。コンピュータゲームでは、プレイヤーとコンピュータの間にもものすごいハンデがあることが多い。コンピュータの人工痴脳では人間にはとてもではないが太刀打ちできないからである。そのため、プレイヤーにはその卓越した戦略眼を発揮するのにふさわしいだけの限定された資源が、コンピュータにはその知性の欠如をごまかせるだけの豊富な資源がそれぞれ与えられることになるのである。

三角関係

プレイヤー同士を不平等にしておくことの利点のひとつに、ゲームに非推移的な三角関係を導入できるということがある。推移律という言葉は数学的には厳密に定義されているが、ゲームに限った話ならじゃんけんを例にあげると理解しやすいだろう。二人でじゃんけんをする場合を考えよう。それぞれが心の中でグー、チョキ、パーのどれを出すかを考え、ついで同時に手を出して結果を比較する。もし、同じものを出していたらあいこでやり直した。そうでなければ、石ははさみを刃こぼれさせ、はさみは紙を切り、紙は石を包み込むという、いわゆる三すくみのルールで勝敗が決まる。このように、あるものには勝つが他のものには負けるというような関係があるとき、これを非推移的な関係というのである。石がはさみより強く、はさみは紙より強いからといって、石で紙には勝てない。ここで気をつけておきたいのは、このように明白な非推移関係が成立するのは、要素が三つの場合に限られるということである。すなわち、ある要素は他のふたつの要素とのみ関係があり、ひとつには勝ちもうひとつには負けるという場合である。たとえばじゃんけんの手が三種類でなかったら、結果は勝ち負けの二種類だけでは収まらなくなってしまう。たとえば、ポイント制などを持ち込むことで、さまざまなレベルの勝敗というものを表現する必要があるだろう。

推移律は、非常に興味深い数学上の問題ではあるが、これ以上、数学的に厳密な理論を展開したところで、あまりゲームの役に立つとは思えない。より一般化して論じたほうが良いだろう。そこで、このような非推移的な状況を表現する上でより一般的な「三角関係」という言葉を使うことにしよう。

ゲームにおける三角関係の例として、『バトルゾーン』 (BATTLEZONE) を例にあげよう。

ゲーム中にUFOが出現した場合、プレイヤーは敵戦車ではなくUFOを狙う。ここにも、プレイヤー、UFO、敵戦車という三つの因子が存在する。プレイヤーはUFOに狙いをさだめることで（三角形の一边）、敵戦車が一方的にプレイヤーを攻撃するのを許す（別の辺）。残った一边であるUFOと敵戦車の関係は、とりあえずプレイヤーにはあまり問題にならない。コンピュータはUFOを使って、プレイヤーを不利な場所におびき出そうとしてはいるのだが。私の言うゲームにおける三角関係という言葉は、この例で理解できるものと思う。

このような三角関係は、攻守が入り交じったような状況を表現するのに用いられることが多い。多くの対戦ゲームでは、攻めと守りの中間状態を考えることは少ない。攻撃的な行動と守備的な行動ははっきりと分かれている。中には、守備と攻撃を完全に分けてしまい、片方が攻撃権を持っているときには一方は守備しかできないようなゲームもある。実はこれは個々のプレイヤーの取れる行動オプションを大きく制限してしまうという点で、かなりリスクの大きいゲームデザインである。取れる行動が制限されてしまうと、互いに影響を与えあうのが難しくなっていくのだ。個々のプレイヤーが、戦略的に攻めと守りのバランスを配分できるようなゲームのほうがもっとずっと面白いものになるだろう。そういったゲームでは、どちらのプレイヤーも攻めようが守ろうが自由だ。しかし、攻めにリソースを費やすと守りが甘くなってしまうという点を忘れてはならない。このような状況下では三角関係が自然に成立する。

三角関係で一番重要なのは、それが間接的な関係を含んでいるということである。一対一では、直接的な殴り合いが避けられない。敵対者は互いにまっすぐ相手陣に向かい戦闘を行う。最初から殴り合いが起きることは明白に予想が付き、いつしか退屈でありふれたゲームになってしまう。三角関係を持ち込むことで、プレイヤー同士に間接的な相互作用の機会が与えられる。それによりゲームはより複雑で面白いものとなるのである。

役者と間接的な相互作用

ゲームデザインにおいて、間接的な相互作用を実現できるというのが三角関係のエッセンスであることはすでに述べた。しかし、この間接的な相互作用とは何かということ自体、もっと掘り下げて考えてみる必要がある。三角関係は、間接的な相互作用の中で最も基礎的なものに過ぎないのである。多くのゲームでは、敵との関係は直接的なものとして描かれる。

たいていの場合、敵はプレイヤーにとって単一のものであり、その間に生まれる最も単純で最もありふれた関係とは、敵をせん滅することになる。その結果として多くのゲームが非常に暴力的になっているのはご存じの通りだ。一方で、三角関係を採用したゲームは何らかの間接的な相互作用を持っている。

三角関係のもとでは、敵対する二勢力は、それぞれにもうひとつの勢力、第三者と関わっていかなくてはならない。その第三者は、交渉を受け付けるだけの代理人かもしれないし、ある程度の戦力を持っているかもしれない。プレイヤーと全く同等の一人前の戦力をもっていることもあるだろう。しかし、ゲーム機の前にプレイヤーを二人呼んでくるのも大変なのに、三人となったらなおさらである。そんなわけで、第三勢力はコンピュータが操る役者が演じることが多くなる。ここでいう役者は、プレイヤーに対する敵とはまた別の意味である。役者は単純なシナリオにそって動いていく。行動を決めるような人工痴脳やそれ自身の目的は持っていない。たとえば、『バトルゾーン』におけるUFOが役者の例である。UFOは、戦場をある軌道で移動し直接戦闘には参加しない。その目的はプレイヤーを悩ませることであり、それ以上の役割は持っていない。

役者という概念を導入することで、間接的関係という概念をより深く理解することができるだろう。模式的に説明していこう。

間接的な関係においては、プレイヤー同士は直接戦闘することはなく、実際に戦闘するのはプレイヤーと契約した役者たちである。ミューズソフトウェア (Muse Software) の『ロボットウォー』 (ROBOTWAR) を例にあげよう。このゲームでは、プレイヤーは各自の戦闘ロボットを受け持つ。プレイヤーがすることは、ロボットの行動原理に関して短い詳細なプログラムを組むことである。ロボットはこのプログラムに従って古代ローマの剣闘士たちのように競技場で戦うのだ。このゲームでは、プレイヤーは実際の戦場から排除され、その代わりに役者、ロボットが戦う。自分のロボットがどれかといのは明白だ。これは本当の意味では間接的な関係というわけにはいかないだろう。実際の戦闘そのものは直接的なものであるのだから。しかし、プレイヤーを戦場から放り出し、観客席に座らせておくことには成功している。私がこの種の間接的な関係を、不完全で過渡的なものと考えているのはそのためである。

より完全な間接的関係の例として、ジム・デュニガン (Jim Dunnigan) の『ドイツ戦線』 (BATTLE FOR GERMANY) をあげよう。このゲームは、第二次世界大戦末期 (1945年頃) のドイツ戦線を扱っている。これが明らかに対等な戦いではないのは判るだろう。当時、ドイツは東ではロシアを、西では米欧連合軍を、同時に相手にしなくてはならないのだ。こんな不利な状況でドイツを担当させられたらたまったものではない。デュニガンはこれを戦線をふたつに分けることで解決した。プレイヤーの一人は、ロシア軍とドイツ西部方面軍を、もう一人が米欧連合とドイツ東部方面軍を担当することにしたのである。どちらのプレイヤーも、侵攻側となり防御側となる。逆にいえばどちらのプレイヤーも侵攻軍でもドイツ軍でもない。対立する双方は明確な定義を失い、役者としてのみ機能することになる。

やはりデュニガンがデザインした『ロシア革命』 (RUSSIAN CIVIL WAR) こそ、間接的関係の極限というべきものだろう。このボードゲームは、赤軍と白軍の間で戦われたロシア革命戦争を扱っている。デュニガンがこのゲームに採用した素晴らしいアイデアは、プレイヤーと戦闘に使うトークンの関係を完全に切り離してしまったことだ。それぞれのプレイヤーは、赤軍ユニットと白軍ユニットを同時に保持し、手元の赤軍ユニットで他プレイヤーの白軍ユニットを、また手元の白軍ユニットで他プレイヤーの赤軍ユニットを攻撃する。ボード上から赤軍、白軍のどちらかが排除されたときがゲームの終了で、その軍の勝利に最も貢献した (戦争に敗けた側のユニットを最も多く倒し、勝った側のユニットを最も多く生き残らせた) プレイヤーが勝者となる。

このゲームは、間接的対立関係というものを非常に良く表している。実際に戦争をしている両軍は、ゲームの勝ち負けには本当の終盤になるまで関わってこない。両軍は役者なのである。プレイヤーが操作しているにもかかわらず、両軍の戦闘にはプレイヤーの意志が現れてこないのだ。このゲームのデザインにおいて重要なことがひとつある。このゲームシステムが有効に働くためには、少なくとも二人以上のある程度熟達したプレイヤーが必用とされるのだ。ともあれ、ここまでプレイヤー間の関係を間接的にしたデザインというのは非常に魅力的なものであることは確かである。直接的な戦闘は暴力的、破壊的であり、本来人間社会からは疎まれるものである。しかしまだに現実世界の中にも、対立はより間接的でややこしい形となって残っている。我々はこの現実世界の中で互いに微笑みをかわし、第三者を味方に引き入れ、あるいは圧力をかけ、共同歩調をとることによって戦っている。プレイヤー同士が直接的に対立するようなゲームは、そういう意味では現実の人間関係を再現するものとはいえない。間接的な対立関係を再現できるようなゲームだけが、現実社会を表現できるのである。

スムーズな上達曲線

プレイヤーが着実、かつスムーズにうまくなっていけるようなゲームが望ましい。初心者であってもある程度は楽しめ、慣れてくればそれなりのスコアが得られ、熟練すればハイスコアが出るような難易度が必用なのだ。プレイヤーがゲームに費やす時間とスコアの関係グラフにすれば、スムーズに上昇するような曲線がえられるだろう。難易度はこのように調整すべきだ。

もちろん、ゲームによってはさまざまな上達曲線が得られるだろう。傾きの小さなゲーム、上達の難しいゲームもあるだろうし、傾きの大きな簡単にエキスパートになれるようなゲームもある。中には、上達曲線に不連続な飛びが現れるものもある。何か特別なトリックが仕掛けられていて、それをマスターすることがゲームの進展に大きく寄与するようなゲームだ。トリックの数が多ければ、それだけ、不連続な飛びの数も多くなるだろう。上達曲線がその中央部で平らになったり、むしろ下降するようだと、それはまずいゲームだと言えるだろう。ゲームに慣れていくにつれて、プレイヤーが混乱するような互いに相反する要素が見えてくるようなゲームに、この傾向がある。いずれにしろ、スムーズに上昇していくような上達曲線が理想なのは間違いない。

上達曲線が不適切なゲームは、なかなかスコアがアップしないためにプレイヤーをいらつかせることになる。プレイヤーは、ゲームが難しすぎるとか簡単過ぎてつまらないとか、あるいは単にひどいゲームだとかの印象を持つだろう。適切な上達曲線が得られるようにデザインされたゲームは、すべてのレベルのプレイヤーに適切な手応えを与え、うまくなってもっと先が見たいという気持ちを持たせることになる。

適切な上達曲線を持ったゲームでは、初心者が自然と熟練者へとレベルアップしていくことができる。そのためには、デザイナーはゲームを単一の障害からなるものではなく、互いに関連した複数の障害が連続したものとしてデザインする必用がある。それぞれの障害はもちろん面白いものでなくてはならず、対象となるプレイヤーのレベルに合ったものである必用がある。理想的には、障害の選択は自動的であるのが良いだろう。初心者レベルから始まって、プレイヤーのレベルに従ってコンピュータが用意する障害を判断して変更していくのだ。なかなかそれも難しいので、一般的には、プレイヤー自身が自分のプレイしたいレベルを選ぶことが多いだろうが。

もう少しで勝てるかも

いつでもプレイヤーにもう少しで勝てるかもという印象を持たせることも重要だ。プレイヤーに何度もゲームに挑戦してほしいのなら、プレイヤーのやる気を損なってはいけない。そのためには、初心者にも熟練者にも、もう少しで勝てるかもと思わせる必用があるのである。もちろん、簡単そうに見えて本当に簡単だったり、こんなのクリアできるわけないと思わせて本当にクリア不能だったりしてはいけない。誰にでももう少しで勝てるかもと思わせるような難易度調整は本当に難しい。中には、初心者お断り、熟練者オンリーというようなゲームもある。特定のプレイヤー以外には、とてもやっつけられないというような印象を与えるゲームである。『テンペスト』(TEMPEST)が良い例だ。ほとんどのプレイヤーにはクリア不可能な難易度を誇っている。難易度調整の点で、最もうまくいっているゲームとしては『パックマン』(PAC-MAN)があげられるだろう。誰だって1、2面はクリアできるように見えるし、本当の意味で最後まで行けるプレイヤーはほとんどいない。

プレイヤーにもう少しで勝てるかもと思わせるためには、見通しの良いゲームシステムを採

用することが必用である。ごちゃごちゃしたゲームは、あまりに細かいデータのせいで初心者には敬遠されがちだ。初心者は、いつまでもこのゲームには何か汚い仕掛けがあって絶対に勝てないようにできているんだという疑いを捨てないだろう。いっぽう、ゲームシステムさえすっきりしていれば、どんなプレイヤーだって、もう少しで勝てるかもという気持ちを持ち続けることができるのである。

プレイヤーに、「もう少しで勝てそうだからもっと続けよう」という意欲を持続させる上でもうひとつ重要なのが、プレイヤーがどこでつまずきやすいのかを子細に分析しておくということである。どんなゲームだって、クリアするまでにはプレイヤーが何度も失敗するようにできている。では、プレイヤーが引っかかっているのはどんなトラップだろう。ゲーム自体やユーザーインターフェイスの欠陥が原因でミスをしているのなら、プレイヤーは「ひでー、こんなのやってられない」といってすぐにゲームを投げってしまうだろう。難易度が高すぎて超人的なテクニックを必用とするようなら、やはり、「きったねー、こんなの勝てない」という声が聞こえてくる。しかし、プレイヤーに自分のささいな失敗、しかも次には何とかかなりそうなミスでつまずいていると思わせることができれば、プレイヤーは、それを越えようと努力してゲームに向かうだろう。ミスのたびに、「ありゃ、またこんなつまらないところでミスっちゃった」とかぶつぶつ言いながら。

まとめ

この章では、これまでゲームデザインをする上で使ってきたさまざまな小技、ノウハウといったものを紹介した。こういったものは、とにかくやたらと使えば良いというものではなく、うまく組み合わせて、いわゆる「テクニック」というものとして使っていくものだ。

「テクニック」は何をゲームの題材として選ぶかという「テーマ」と同じくらい重要で、デザイナーの個性が現れる。偉大なベートーヴェン (Beethoven) の数々の交響曲、荘厳な第五、熱狂的な第六、忘我の域へと誘う第九……これだけ印象の違うどの曲を聞いてもベートーヴェンの曲だと判るのは、そのすべてにベートーヴェンの比類なき「テクニック」がふんだんに活かされているからである。ゲームデザイナーを目指すのなら、あなたの、あなただけの「テクニック」を作り出し洗練させていくことだ。

[目次に戻る](#) | [各章へ: 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9](#) | [次章に進む](#)

始まり

初期の仕事：1982年1～4月

長く退屈な日々：1982年5～12月

仕切り直し：1983年1～4月

仕上げ：1983年5～6月

第五章では、私が理想と考えるデザインの各工程を、順を追って示した。これまでに述べてきた概念を取り入れた一般的な手法を紹介したのである。しかし、それはあくまで理想論であって、実際に使う上では現実に合っていないのが実情である。私自身、第五章に述べた順序どおりにゲームをデザインしたことは一度もない。実際にはデザインは、もっと紆余曲折している。この章では、実例として、私がどのように『エクスカリバー』(EXCALIBUR)をデザインしたかを紹介しよう。現実のデザインがどれだけ愚かでミスだらけであるかを、第五章で述べたような理想論と比べることで、理論と実践の間のギャップを埋める手助けとなるだろう。

始まり

1981年12月、私はアタリ社(ATARI)のアラン・ケイ(Alan Kay)が新設したコーポレート・リサーチ・グループ(Corporate Research unit)での仕事を始めた。ケイ博士は、「すべてを任せるから、自由にゲームをデザインしていい」といつてくれた。私も信頼に応えるべく、それに値するだけのゲームを作ろうと決意した。私はこのゲームを、雄大で輝かしく、他のすべてのゲームを恥じ入らせるほどの、最高のゲームにしたかった。マーケティングの面は目をつぶることにして、このゲームは、メモリ48KB、フロッピーディスク専用というぜいたくな環境で動作させることにした。これにより、コンピュータ資源はふんだんに使えるようになった。

私はもともとウォーゲームをデザインしてきたので、自然と、今度のゲームも戦争を扱おうと考えた。戦争は、人間同士の対立の極限を表現するものであり、最大の邪悪であり、最高の悲劇である。深刻なテーマを好む芸術家がまずこれを主題に選びたがるのも当然であろう。私は、ウォーゲームにおける戦争の扱いに新しい切り口をみせたかった。従来のウォーゲームは、戦争について深く考えもせず、英雄的行為の表現として賛美したり、魅惑的な知的パズルとして扱ったりしている。私はウォーゲームに、さらなる何かを欲したのだ。戦争という言葉が意味する本当の何かを。私が目指すゲームでは、生き残るための選択肢として戦争が行われることもあるだろう。ただし戦争を軽薄に扱うことはしたくなかった。ただ戦うことだけを考える戦争屋は、どうしても負けてしまうようなことになるゲームが欲しかった。なぜなら、平和的な戦略こそ最善であることが多いということを、私は深く信じているからだ。このゲームでは、人間の政治的手腕を扱い、それは必ず、リーダーシップに焦点を合わせることになるだろう。もうひとつの基本的な目標は、実際には多くのゲームを連結させて、ひとつのゲームを成り立たせるということだった。これによって、さまざまなことを描写できるはずだ。政策、政治的手腕、そして、戦略上の間接的なレベルから戦術上の直接的なレベルまでの、さまざまなスケールの戦争を。

次に、ゲームで再現すべき空想の世界観を決めた。私は、ふたつの候補のどちらを採用するかで悩んだ。大きな核戦争後の米国のゲームか、それとも、ローマの権威崩壊後の、暗黒時

代の英国のゲームか。どちらも、惨事の後に自らを立て直そうとしている社会である。前者は、私の目標とするゲームで扱うにはあまりにも悲惨すぎると思えた。さらに、後者には、アーサー王伝説 (the legends of King Arthur) という、本質的に興味深い主題が重なっていた。よって私はアーサー王のほうを選択した。

このゲームにおいて、プレイヤーはアーサー王ということになるだろう。そして目標は、英国を統一して平和をもたらすことである。他の王たちのアーサー王の権限に対する反発がプレイヤーの前に立ちふさがる課題となるだろう。権限を確立するため、プレイヤーは、さまざまな行動を行う必要があるだろう。軍事行動は、さまざまな選択肢のうちのひとつに過ぎない。むしろ、軍事の過剰利用は民族を非情にして、果てしない反乱と無政府状態をもたらすようにしようと、私は決意した。これらの崇高な目標を立てるとともに、私は本格的にゲームデザインの仕事を始めた。

初期の仕事：1982年1～4月

私はまず、「リーダーシップとは何か」という難問に取り掛かった。ゲームのテーマであるこの問題に答えることはゲームを作る上で重要なことだ。国家レベルのリーダーシップの本質を明確にし、それをゲームで処理しやすい形に煮詰めることは不可欠だった。リーダーシップの核となるものを抽出し、それを表現するための形式をデザインする必要があった。軍事的な面でのリーダーシップは最も明白で、扱うのも容易である。プレイヤーが軍事的な決定をするゲームをデザインするのに、私にとって何も難しいことはなかった。だが、それだけでは十分ではない。もっと広い範囲を扱いたかった。私のゲームは、リーダーシップの、社会的、外交的、対人関係的な側面を扱わなければならなかった。こういった要素を、どうやってゲームで表現し、操作すれば良いのだろうか。私は何ヶ月も悩みつづけた。

このような基本的な問題で、いらいらが募るのは早かった。早く先に進みたいと願う私の中の子供の部分は、当面の満足感を欲した。そこで、気を紛らすため、ゲームのタイトルとエンディングシーンを書いた。これらはゲームの構造上それほど重要ではなかったので、私のデザインの完全性を危うくせずに、面白いグラフィック技術を考える機会を得ることができた。エンディングシーンでは、若干の面白い問題があった。エクスカリバーが湖上の空中でくるくると回転して落ち、水の中から突然現れた手がそれを捉え、そして波の下に隠れていくシーンである。私は、剣の刃が風を切る寂しげな音を入れようとして多くの時間を費やしたが、満足いく結果は得られなかった。そこで私は発想を転換し、タイトルとエンディングシーンに適切な音楽を添えることにした。私は次の2曲を最有力候補として選んだ。ワーグナー (Wagner) の『ジークフリートの死と葬送行進曲』 (Siegfried's death and funeral) と、ドヴォルザーク (Dvorak) の『交響曲第七番』 (Dvorak's Seventh Symphony) の一部である。

私はまた、ゲームの基本的な構造を決めた。このゲームを4段階に入れ子になったシステムとして表現することにした。第1段階のシステム、『キャメロット』 (CAMELOT) が、アーサーの城内での活動を扱う。これらは、王国の内政、外交の処理、軍の準備などを含むだろう。第2段階の『ブリテン』 (BRITAIN) は、アーサーが軍を率いて国中を周り、戦略上の軍事活動に携わることを可能にする。第3段階の『バトル』 (BATTLE) では、敵軍との戦闘を行う。もしアーサーが戦場で敵国王に出会うことができれば、第4段階『ジャウスト』 (JOUST: 馬上槍試合) に入る。これは、アーサーが敵を馬から振り落とす、シンプルなアクションゲームにしようと考えた。完全な一人称視点で、前進する騎手を描き、画面を上下に揺らすことで、アーサーの馬が疾走する様子を表現する。私は、3Dグラフィックを生成するうまいアルゴリズムを考えたりして楽しんだ。けれども、多くの努力を費やした後、ゲーム全体の中で『ジャウスト』をプレイするのはほんの数秒だけで、ほとんどプレイヤーへの障害にならな

いということに気付いた。それで私は、新しいアイデアで始めからやり直すことにした。今度は剣での切りあいをゲームに入れたらどうだろう。最初の問題は、ジョイスティック操作で、どうやって剣の動きをシミュレートするかということだった。私はものさしを持ち出して、それをリビングで振ってみたりして、何時間も過ごした。ジョイスティックでうまく表現することのできる何らかのパターンを、直感的に見極めようとしたのだ。これは困難だった。剣術における剣の振り方は非常に複雑な動きであり、ジョイスティックで十分に表現できるようなものではないからだ。だが結局、私は合理的なシステムを見出した。ジョイスティックの端から端への動きを、剣の攻撃の角度に対応させるのだ。面、胴、袈裟切り、ジョイスティックを倒した方向に剣を振り、逆方向に倒せば剣先をかえすことになる。

この問題が解決し、私は3Dグラフィックで敵の剣士を描写する、新しいグラフィックルーチンを作ろうとした。これは非常に難しい仕事であることがわかった。私は結局、馬上槍試合ゲームを捨てたのと同じような多くの理由により、ちゃんばらを断念することにした。そもそも私は、剣のみでアーサーが勝利を掴めるようにはしなかったのだ。剣による戦いが勝利に結びつかないとすれば、それがゲームの中に存在している理由は何だというのだ。

3月になり、私は『ブリテン』モジュールの仕事を始めた。これは、たくさんの飾り付けがされたスクロールマップだった。私は以前に『東部戦線1941』(EASTERN FRONT 1941)や『リージョネア』(LEGIONNAIRE)で同じような経験をしていたので、このモジュールの実装は容易だった。このゲームではより多くのメモリが使えたので、奮発して、巨大なスクロールマップを作ることに決めた。私は、非常に大きい、6Kのマップを仕上げた。

デザインは私の頭の中で、徐々に形になっていたが、ある基本的な問題に答えを出せないまままでいた。これは歴史のゲームなのか、フィクションのゲームなのかということだ。すなわち、6世紀の英国についてのゲームなのか、アーサー王についてのゲームなのか。私は、両方の題材について可能な限りの本を読んだ。この研究により私は、6世紀の英国というのが、無秩序で憂鬱な場所だったと結論した。先住民のケルト人(Celts)は、東海岸からのアングロ・サクソン人(Anglo-Saxons)の侵入から故国を守っていた。2世紀をかけて、アングロ・サクソン人は、ケルト人を徐々に西方へと押しやっていった。実在のアーサー王は、アングロ・サクソン人に対する反攻を指揮した、ケルト人の将官である。バドンの丘(Mount Badon)の戦闘に勝利し、およそ50年の間、アングロ・サクソン人の攻撃を止めたのだ。けれど、アーサーの成功は短い休息にすぎず、最終的にはケルト人は敗北した。したがって、歴史の記録は、それ自身を再構築しようとする社会という、私が当初再現しようとしたテーマを満たしてはくれない。それどころか、暗黒時代の英国の物語は、侵入者によって容赦なく追い払われる一民族の物語なのだ。

だが、征服された者たちの夢が、それに打ち勝つアーサー王の伝説を生み出し、伝説は時代を経て、どんな語り手のニーズにも適するように都合よく形作られていった。これらさまざまに変化しながら語りつがれてきた伝説を読んだとき、私は、彼らの並外れた柔軟性に衝撃を受けた。それぞれの芸術家が、アーサー王伝説に異なった性格付けをした。宗教的暗示や、下品な物語や、騎士道といったご都合主義である。かのマーク・トウェイン(Mark Twain)さえ、それを彼特有の痛烈な社会のコメントに転用した。

映画『エクスカリバー』(EXCALIBUR)を観たときに、大きな転機が訪れた。それは素晴らしい映画で、アーサー王伝説のベストな要素を美しく捉え、なお独自の表現をしていた。私は何度もそれを観て、物語の深さを大いに楽しんだ。この映画があまりにも素晴らしかったので、私は恥ずかしくなった。私は、美しいグラフィックに固執して、重要な芸術的問題を妥協してしまっていたことを悟った。私は、静かな丘に登り、熟考して一日を過ごした。私

は、以前に決めた崇高な芸術的目標に、再び専念することにした。私はまた、自分ひとりではそれらを実現できないということに気付いていた。助けが必要だった。私は、ラリー・サマーズ (Larry Summers) の助けを借り、また、ヴァレリー・アトキンソン (Valerie Atkinson) を雇った。決意を新たにして、我々は仕事に取り掛かった。

長く退屈な日々：1982年5～12月

1982年5月の時点で、私はすでに大まかなデザインを確立させていたが、細部のほとんどは未完成のままだった。コードはすでにいろいろ書いていたが、それらは別々に動作していて、まとまってはいなかった。総合的なデザイン仕様書は一切なかった。この期に及んで、私は愚かにも、明らかに重要でない仕事を選択した。私は、画面にゴシック文字を描画する、『カリグ』モジュール (CALIG) を書いた。ヴァレリーは、そのルーチンのためのビットマップテーブルの準備に着手した。ラリーは、タイトル画面に音楽を加える仕事の仕上げと、フェードイン・アウトのルーチンを担当した。この仕事で、あろうことか2ヶ月近くを費やしてしまった。誇張ではない。

6月に、我々は『キャメロット』モジュールの仕事を始めた。これは、ヴァレリーがメインとしてプログラミングすることになった。このモジュールは、実際には図案化されたメニューだった。それぞれの部屋 (メニュー) に、一単語で記述された四つの選択肢があった。プレイヤーは、縦方向に王冠カーソルを動かして、メニューを選択できるようにした。その右に、重要な情報を表示するためのグラフィックウィンドウを配置した。たとえば、「円卓の間」 (Round Table Room) には、円卓を表す円を描き、そのまわりに、騎士たちを表す盾を描画した。部屋における彼らの空間的位置は、そのまま、社会的な関係を示していた。「宝物庫」 (Treasury Room) には積み重ねられたコインの絵を表示するつもりだったが、後に、より詳細な経済データを表示するために、このフィーチャーは削除しなければならなかった。また、スクリーンに多くの色を表示できるディスプレイを使うつもりだったが、実行時間がかかりすぎたため、このアイデアも断念した。

ヴァレリーがこの大仕事を始めたころ、私は、円卓における社交のゲームに取り掛かった。やろうとしていることの規模は実感していなかった。私は、アーサーに社会集団の管理をさせるための、小さなゲームを作りたかった。すぐに私は、このようなシチュエーションの面白いところは、放射状の関係 (アーサーと他の騎士たちの関係) ではなく、騎士たち同士の横の関係であるということに気付いた。アーサーは必然的に騎士たちを放射状に扱わなくてはならないけれど、横の関係は決定的要因となるだろう。このシステムが魅力的であることがわかったので、それを集中的にやってみることにした。私は、グループの行動を一番面白くモデル化するアルゴリズムを開発した。このアルゴリズムには非常に満足していたので、急いで、それを単独のゲームにした短いBASICプログラムを作成した。このゲームは非常に有望に思われた。特に印象的だったのは、私の妻の反応だった。彼女はつまらないゲームに手厳しいのだが、このゲームはすぐに気に入ってくれた。彼女が楽しめるものを作れたということは驚きであり、嬉しかった。『エクスカリバー』のデザイン中に生まれた、この新しいゲームを、完全に新しいプロジェクトとして研究を続けることに決めた。アリック・ウィルマンダー (Aric Wilmunder) が、『ゴシップ』 (GOSSIP) と名付けられたこの新ゲームのデザインのために雇われた。

7月からは、遅々として進行せず、挫折気味の長い日々が続いた。私はこの本を書くことに時間を割き始め、さらに他のことでも注意がそらされた。私が毎日アクティブに参加できなくなり、プロジェクトはもたつき始めた。ラリーとヴァレリーは、弱気な状態の中でも最善を尽くして、彼らの仕事をこつこつやっていた。彼らは、数ヶ月をかけてゆっくりと、私が手

短に記述していた骨格に肉付けをして、システムを作り上げた。私は、プロジェクトに割ける時間がわずかしかなく、勘を頼りに多くのデザインを決定することになった。週に一回の打ち合わせで、彼らは私に最近の問題を報告する。もう先週のことほとんど忘れてしまっているため、一緒に当面の解決策を考えるのだ。わたしの勘もそこそこのものではあったらしい。こんな嘆かわしい状態が繰り返された。結局、私のその場しのぎの決定の多くが、全体的なデザインの中で、微妙な問題を引き起こすことになった。気の毒なヴァレリーは、『キャメロット』モジュールに機能を追加しては、それを取り上げられたり、後になって再導入したりするはめになった。

この期間の我々の記録は、多くの無駄な努力を示している。我々は、宝物の間のコインの積み重ねで、資金量を表現することを意図していた。コインの描画ルーチンを作成するため、多くの時間が費やされた。しかし結局、このコインの積み重ねを表示する十分なスペースがないことに気付いて、単なる数字を使うことになってしまった。それだけではない。宣戦布告、同盟、包囲作戦、貢物の要求、動き回る軍隊など、これらは皆、我々がデザインし、プログラムして、結局捨てることになった要素であり、私自身の事前計画が失敗だったことをあらわにしている。

6ヶ月が、この混乱状態の中で過ぎていった。しかし完全に無駄だったというわけではない。いや、多くの進歩があった。ラリーは、経済の処理や『ブリテン』モジュール、ディスク交換、外交上のニュースの表現、そして、肥大化していくコードの統合の多くを完了した。ヴァレリーは、『キャメロット』モジュールにさらに新しい機能を盛り込み、ゲーム全体で最も大きく複雑なモジュールに仕立てた。とはいえ、私がもっと計画的で、プロジェクトにもっと多くのエネルギーを捧げていたら、半分の時間で同じことができたはずだ。クリスマスには、皆もう疲れていて、士気が下がっており、プロジェクトの完了を絶望視していた。暗い日々だった、本当に。

仕切り直し：1983年1～4月

1983年1月、私は私の中で『エクスカリバー』を最優先のプロジェクトに戻すことができた。私は、再びプロジェクトに取り掛かったが、その心は冷めていた。早いところこのプロジェクトをやっつけてしまおうと。1982年に思い描いていた、雄大で高尚な感覚、真に壮大なゲームといった深遠なビジョンは、どこかへ行ってしまっていた。

何が何でもゲームを完成させるという悲壮な決意がそこにあった。私は頻繁に、ラリーやヴァレリーと話し合った。デザインを容赦なく切り裂き、あいまいな定義や必須でない部分を剥ぎ取っていった。私自身の能力が足りなかったために、膨大なコンピュータ資源を消費することで逃げようとしていたつけが結局全部、自分自身に返ってくることになった。私は自宅で、円卓の間の騎士たちのための人工知能ルーチンに取り組んだ。これには数週間を要した。それから、戦闘シーンに取り掛かった。2月から3月の間に私はこのモジュールを書き、デバッグし、テストプレイをした。私は、期限を4月1日と決め、それまでにゲームを完成させるように自分を追い込んだ。私の記録によれば、一日平均300バイトのデバッグ済みコードを書いたことになる。ちなみに業界の平均は一日75～100バイトである。ラリーとヴァレリーは狂乱に巻き込まれた。彼らは猛然とプログラムの断片を統合し、それによって生じた無数の矛盾を解決した。マーリンの部屋 (Merlin's room)、経済、臣下、税金、交換といった要素を処理するすべてのモジュールがデザインされ、コーディングされ、そしてデバッグされた。

それにもかかわらず、4月1日の期限には間に合わず、4月15日に延期することになった。しか

し、それさえ間に合わせるのは不可能になった。それでも我々は、4月15日に重要な目標を設定した。すべてのコーディングを、その日までに完了させよう。

4月の最初の2週間は、熱狂的な努力で費やされた。毎日、ときには4時間ぶっ通しで打ち合わせをして、我々は確かに、デザイン上で最も難しかった部分、人工知能アルゴリズムを苦労して完成させた。

AIルーチンはデザインのすべての面を反映しなくてはならないため、この仕事は最後まで残しておいた。AIルーチンを設計する前に、デザインが完成していて、すべての変数が定義されていなければならない。いったんAIルーチンを組んでしまうと、デザインの変更は利かなくなる。あとから重要なデザイン変更をすれば、全部のAIルーチンが台無しになってしまうからだ。

『エクスカリバー』のAIは明らかに、私がいままでに試みた中で最も難解なものである。異なった国王の性格、経済の要素、軍の要素、幾何学的な要素などを考慮しなくてはならない。我々が開発したシステムは、軍の威信、経済の信頼度、国王自身の好感度、などといった媒介変数を使用する。また、野心、愚かさ、守備的態度などといった性格も、アルゴリズムに取り入れられている。

仕上げ：1983年5～6月

4月15日までにすべてのコーディングを終わらせるという目標の達成は、ほぼ成功した。残っているコードは非常に些細だった。我々は皆、2週間の休暇をとった。5月に我々は、『エクスカリバー』の仕上げを開始した。ラリーとヴァレリーはバグ出しを始めた。私がいまこれを書いているとき、彼らはまだ仕事に取り組んでいる。6月には、バランス調整と洗練作業に入るつもりだ。私は、このゲームを洗練するのにもっと多くの時間をかけたいけれど、予定は大幅に遅れてしまっている。18ヶ月もの開発期間をかけてしまったし、3プログラマー一年を費やしてしまった。早ければ6週間の開発期間でゲームが作られる昨今において、『エクスカリバー』はこれまでで最も丹念に作られたゲームのひとつだろう。いままで試みられた中で、最も意欲的なデザインのひとつであることは確かである。成功するという保証はないが、もし失敗したとしても、それは努力が欠けていたためではないだろう。

クロフォードの1998年のメモ：『エクスカリバー』は7月に完成した。

[目次に戻る](#) | [各章へ: 1 - 2 - 3 - 4 - 5 - 6 - 8 - 9](#) | [次章に進む](#)

日付：1997年6月17日(火)

インタビュアー：ピーボディ教授 (*Prof. Peabody*)

クロフォードさんがこの本を執筆なさってからの10年間の変化について、どのようにお考えですか？何が予想通りで、何が違っていたのでしょうか。また、執筆当时には書かなかった、あるいは書けなかったようなことで、いま現在、何か追加したいことはありますか？

そうですね……。私がこれを書いたのは1982年ですから、実際には15年も経っています。ですから私は、本の中で予想した10年後（1992年）と現在（1997年）の間に差があったとしても、それからさらに時間がたっているぶん、恥をかかなくてすみますね。特定の記述については、どうこういうつもりはありません。代わりに、全体的な考えについてお答えしましょう。エンターテイメント・ソフトウェアの市場の成長率については、おおかた私が予想したとおりでした。ゲーム産業は、80年代前半よりも、ずっと大きくなりました。私の見当違いだったのは、市場そのものの拡大という楽観主義的な部分です。私は当時、エンターテイメント・ソフトウェアはもっと多様化し、さらに多くの人々を楽しませるようになるに違いないと信じていたのです。でも、それは間違っていました。コンピュータゲームは未だ、本質的には全く変化していないと言っても良いでしょう。私たちが15年前に非難していた、せわしないシューティングゲームや、オタク的なストラテジーゲームと、何ら変わってはいないのです。

『*Excaliber*』はいかがでしたか？（大成功だったのでしょうか？）

いえいえ、とんでもない。あれが出たのは、ちょうどアタリが崩壊したころで、崩れ落ちる瓦礫にまみれて消えてしまいました。ただ、数こそ少ないとは思いますが、あのゲームをやってくれた人は感動してくれたのではないのでしょうか。程度はともかく、あのゲームを革新的だと評価してくださるデザイナーも多いようですし。

コンピュータゲームは、歴史の研究に役立つとお考えでしょうか？

役に立ちますとも。その長所とともに、短所すら、歴史の研究に役立つと思います。史実を検証しようとした場合、どうしても偏った見方をすることになります。たとえば、歴史の資料には、歴史上の大物のことばかり偏って書いてあるものです。私たちは、カール大帝については細かいところまで良く知っていますが、彼の支配の下に生きた数百万人の小作農については、ほとんど知りません。また、私たちは鉄器時代よりも青銅器時代のことを良く知っています。鉄と比べて青銅は腐食しにくいからです。

もちろん、コンピュータゲームの中でそうなったからといって歴史的な証拠にするわけにはいきませんが、それを通して手がかりを覗き込むためのプリズムにはなりますし、なにより私たちの見方に方向性を与えるものでもあります。このことは、私たちが物事をそのまま信じ込んでしまうような「神話的」な見方よりも、実際にこの手でやってみるような「実証主義的」な見方をしたいときに、長所となるのです。つまり、歴史は「驚くべき物語」でありうるし、「自然に起きたこと」でもありうるわけです。神話的な見方をすれば、ナポレオンがあれほど多くの戦いに勝利したのは、「ナポレオンは才知に長けた戦略家だ。ナポレオンばんざい！」ということになるでしょう。けれども私たちは、ウォーゲームを通して、彼の

行動を追体験することができます。それによって彼がなぜそういう戦略・戦術を取ったのかをより深く理解することができるのです。同じく、「実証主義的」な見方をすると、ナポレオンが「庶民」に関しては、彼の対抗者よりもずっと無慈悲であったということがわかります（ナポレオンはすべての小作農から食物を取り上げているのです）。

文章にしてしまうと、「実証主義的」な考え方は、「神話的」な考え方よりもインパクトは弱くなってしまいます。たとえば、ミッドウェイ海戦 (Battle of Midway) に関して、戦いが終わったあの劇的な瞬間（日本の海軍提督が空を見上げ、頭上にアメリカの急降下爆撃機を見たその一瞬）を好んで描写したがる人は多いでしょう。しかし、彼らはどのようなプロセスを経て、その瞬間に到達したのでしょうか？ もちろん、細かな事象を時系列で文章にしておくことで、何がこのような幸運な大逆転を導いたのかを伝えることは可能でしょう。しかし、さながら太平洋で実演される雄大なギリシャのドラマであるかのような神話的印象を保つためには、それだけでは不可能です。一方、ゲームで実際にそれをプレイすれば、微視的な出来事の積み重ねが、最終的、巨視的には驚異的な結果につながるのだということを、肌で感じ取ることができるでしょう。

コンピュータゲームは、歴史のいくつかの側面を強調するために使うことができます。たとえば、私は数年前、「経済成長から自動的に技術開発が生ずる」というテーマで、『ガンズ・アンド・バター』 (Guns & Butter) というゲームをデザインしました。たいていの技術史は、誰か偉い人がこれこれを発見・発明したという「偉人伝」的な考え方を持っています。そこで私は、経済が新しい技術を利用するに値するほど十分大きければ、技術は自動的に発生するのだという代替意見を提示したのです（ついでながら、これは技術史のマルクス主義的見方とでもいえるでしょうか）。私はこの命題が必ずしも正しいとは主張しませんが、歴史のプロセスの異なった見方を提供したことは間違いないでしょう。社会におけるグループは常により大きいグループに吸収されやすいという傾向は、このゲームの中で明快に再現されています。

ゲームにも、乱用される恐れのある多くの余地があります。出版界の作家や編集者たちと比べると、ゲームデザイナーは不遜で愚かだったために、コンピュータゲームは啓蒙よりも害悪の源となってしまっています。ゲッベルス (Goebbels) が恐ろしかったのは、宣伝のために近代的なマスメディアを使う方法について、彼が非常に的確に把握していたからです。いまのところ、我々ゲームデザイナーはそのようなノウハウを持っていません。しかしいつかは、今度はインタラクティブなメディアを悪用する、ゲッベルスのような人間が現れないとも限らないでしょうね。

歴史のゲームと歴史の本や論文では「命題」の表記法に大きな違いがありますよね。論文では命題は過去時制で表記されますが、ゲーム中では現在のものとして表現されます（たとえば「フランス革命は、政府の財政危機、経済の非常事態、君主制の権威の低下の結果として生じた」というように）。

ちょうど、最近いくつかのボックス版ウォーゲームを整理していたところなのです。それは皆、箱の表に「時は1476年5月21日木曜日、午前6時」などと書いてあります。それが現在であるかのような雰囲気は、シミュレーションには極めて重要です。最も強力な魅力です。歴史学では、現在は過去にアクセスできないことになっていますよね。

さて、ソフトウェアの納期が迫っているのです。もう行かなくては・・・。

[目次に戻る](#) | 各章へ: [1](#) - [2](#) - [3](#) - [4](#) - [5](#) - [6](#) - [8](#) - [9](#)