

# 海外 Mod 文化の受容と理解

永江孝規

インタラクティブメディア学科

Acceptance and Understanding of the Foreign Mod Culture

NAGAE Takanori

*Department of Interactive Media*

(Received November 2, 2012; Accepted January 10, 2013)

## 1. はじめに

ビデオゲームの私的改変、より具体的には PC ゲームにおける Mod (modification の略) と呼ばれるゲームの楽しみ方は、日本では通常、単なる違法行為 (チート、クラッキング)、著作権侵害と見なされている。しかし、Mod を、ソーシャルメディアを通じて一般化しつつある CGM (消費者生成メディア; Consumer-Generated Media) において作られた UGC (ユーザ生成コンテンツ; user-generated content、または UCC User-Created Content) の一種と捉えれば、それは決して珍しいことでもなく、また必ずしも反社会的なことでもない、むしろゲーム産業として、積極的に「飼い慣らし」て、盛り上げていくべき方向であるということに、すぐ気がつくはずである。

UGC や CGM と言うと耳慣れないが、これらの用語は Web 2.0 世代のコンテンツ、特に SNS (Social Networking Service) が生み出すコンテンツを言い表すために、2005 年くらいから使われるようになり、現在でも盛んに使われている [Eto] [OECD]。今日完全に独立したメディアとして定着した Web は、UGC そのものである。ほんの二十年前、Web がなかった時代、個人が発信する「草の根メディア」や「同人メディア」などは、テレビや出版などのマスメディアに比べて、蟻にもミジンコにも等しい存在だった。出版・放送業界は、一般人が住むよりもはるかな高みに君臨しており、書籍や音楽、映像コンテンツなどは、その天界に上り詰めた一部のエリートの手元に寄せねば決して作れないし発信できないものとされていた。文化の創造はすべて特定少数のプロが担うものであった。

ところが、デジタルカメラや映像編集ソフトなどの普及、簡単にコンテンツをアップロードでき、管理や維持の手間もお任せの SNS やいわゆる Cloud サービスが普

及すると、世界中で膨大な規模のコンテンツがアップロードされ、その中には少なからぬ高品質かつ秀逸なコンテンツが含まれるようになってきた。地上派テレビ放送ですら Youtube のコンテンツを転載する時代なのだ。

1996 年頃すでに素人による「Web 日記」が現れ始めた [Nagae1] が、それが今日のようにブログメディアとしてマスメディアの脅威となるまで発達するとは誰も思わなかった。巨視的に見れば世の中のコンテンツ制作の趨勢は、閉じた特定少数によるものから開かれた不特定多数によるものへと変わりつつある。分野によってあるいは地域によって早い遅いのばらつきはあっても、例外なく、不可逆なもののように見える。貴族やブルジョワから大衆へ。1848 年のプロレタリアートによる赤色革命のようなものとは言えないか。

Mod もまた UGC の一種である。素人そのもの、素人に毛の生えたようなものもあれば、プロ同然の本格的なものもある。Mod を単なるホビーと見なせば本質を誤るだろう。初音ミクは日本の UGC の中で海外の Mod 現象に特に類似したものとして挙げてよい。初音ミクというキャラクターは、ファンやユーザの手に委ねられ、YouTube やニコニコ動画などに、いわゆる Mad ムービーとしてアップされることによって、そのコンテンツを爆発的に増殖させた。

UGC の媒介となる SNS には Youtube のような映像配信サイト以外にも、ソーシャルブックマークの del.icio.us、写真掲載サイトの Flickr や Instagram、他にも twitter や facebook、pixiv、Wikipedia、mixi などを挙げることもできる。ニッチ市場を狙って非モテ SNS、ハゲマスなど特定の属性を持つ参加者限定の SNS もある。それらは、すでに今日の多くの日本人にとって非常になじみ深いものである。一次コンテンツの二次創作的利用がある程度著作権者から容認されており、SNS などの一般ユーザが参加できるネット上のコミュニティがありさ

えすれば、ほとんどのコンテンツから UGC や Mod が自然発生し、コミュニティをより活性化させる。むしろユーザの活動を野放しにするのは危険だし、どんなコンテンツが出てくるか予測がつかない。法務部門を強化整備する必要もあろう。従って一次コンテンツの保持者してみると、自分の著作物の二次利用を許容することは、恐ろしくも煩雑にも感じるに違いない。しかし、初音ミクの成功を鑑みれば、二次創作コミュニティを萎縮させるようなことは、必ずしも得策ではないと気付く。最近、漫画家の佐藤秀峰が自身の作品の二次利用規約を発表したが、これもまた大きな反響を呼んだ。

海外の Mod 事情も急速に変化しつつある。Cry、Unity などのゲームエンジンがグラフィクスの美麗さと開発の容易さを強調し、インディーデベロッパーを中心に利用が広まりつつある。1990年代後半に Unreal と Quake が現れて、洋ゲーがようやく頭角を現し、世界に戦いを挑んできたときと同じ雰囲気だ。それらのゲームエンジンの最新動向についてサーベイすることは極めて興味深くまた緊急なことではあるが、今、本稿で扱う余裕がない。本稿では一世代前の、初代 Quake エンジンの子孫で、ある意味では「化石的な生き残り」である Source エンジンを主に扱う。Source の開発元の Valve 社は Source に代わる新しいオーサリングツールを開発中であるから、本稿に述べることは数年のうちに時代遅れになるだろう。

筆者が所属するインタラクティブメディア学科は、その前身のメディアアート表現学科のときから、3DCG 教育を学科のカリキュラムの柱の一つとしてきた。筆者もゲーム制作というよりも、ゲームの中で用いる 3DCG モデルやマップなどを制作できるようになることを目的としてゲームエンジンを教材として使ってきた。仮に卒業制作にゲームが作れなくても 3DCG かグラフィクスのスキルを学ぶことができる、というのがいわば保険だったのだが、だんだんにゲームエンジンの仕組みがわかってくると、できれば CG だけでなく、CG 教育をメインとしてゲーム開発を行いたいと思うようになり、そのために Mod をより積極的に活用したいと考えるようになった。

本稿ではまず PC ゲームと Mod について概観したのちに、筆者らがゼミの活動して行ってきたいいくつかの試みを紹介し、今後の Mod の方向性と可能性について考察し、Mod の展望を探ることにする。

## 2. PC ゲームと Mod

PC ゲームや Mod が日本で海外のように流行しなかつ

たのはなぜか、ということが筆者の長い間の疑問だった。難しい問題だが、暫定的に、筆者の現時点での考察を披露しておきたい。

PC はコンソール機と違ってゲームデータを改変しやすい。マップやテクスチャなどのゲームデータにプロテクトがかかっておらず、生のテキストデータや画像データとして提供される場合はなおさらである。もしゲームデータがアーケード機やコンソール機のように ROM にハードコーディングされたものであったなら、また、ソフトウェアだったとしても、実行ファイルの中にバイナリーデータとして組み込んであれば、ユーザの改変は困難だっただろう。しかし、プロテクトがしばしば破られたからでも、PC が技術的にコンテンツ保護困難なプラットフォームだったからでもなく、多くの場合、それらの生データはユーザが勝手に改変することを容易にすべく作者（経営者というよりはプログラマーやゲームデザイナー）が意図して提供したものである。少なくとも id Software 社の創設者の一人でテクニカルディレクターの John Carmack はそうであった。彼はバイナリーデータからわざわざマップやテクスチャなどのゲームデータを自由に改変可能な形式で分離し、レベルごとに実行ファイルからロードするようにしたのである。その上、リリースから数年を経過したゲームのソースコードまでも彼は無償公開した。当然経営陣はタダでソースを公開することに渋い顔をしたにちがいない。ソースコードは独占的（プロプライエタリ）な知的財産だ、という考え方は海外でも根強い。また、ソースを公開したら他社に模倣されてしまうかもしれない。

もし Carmack が一介の雇われエンジニアに過ぎなかったら、このような壮挙は望めなかっただろう。彼は創設者の一人であり、会社の経営にコミットする権限を持っていた。Carmack は自分のコードを小説か何かのように広く人々に読んでもらいたかったのかもしれない。プログラマーにはソースをプログラマーどうして見せ合いたい、ソースをシェアしたいという強い欲求がある。もしソースを秘蔵してしまえば、会社が倒産したり解散したり売られたりしたときに、ソースもまた日の目を見ずに失われてしまう。しかし公開すれば自分の生きた証として、死後まで残るかもしれない。

ハッカーの自由な文化 [Levy] や、Richard Stallman が提唱してきた GNU（フリーソフトウェア運動） [RMS1] [RMS2] にもなんらかの形で影響されたかもしれない。いずれにせよ、自然発生的に、単にクラッキングの延長として Mod が生まれてきたのではない。Mod を生み出すべく働きかけた、影響力の大きな先駆者ら（いわば佐藤秀峰のような「確信犯」）が制作者側にいたのである。

Carmack は最初パブリックドメインに限りなく近い形でソースを公開したが、後にフリーソフトウェアとして（つまり GNU ライセンスで）公開しなおしている。

## 2.1 id Software と Doom、Quake

今日の PC ゲームは1991年に id Software が設立されてから加速した。Carmack とその他のスタッフは、まず Hovortank-3D という FPS（一人称視点シューティングゲーム；First Person Shooter）を作り、ほぼ同じシステムを踏襲して Catacomb-3D、Wolfenstein 3D を出した。これらは3D というより疑似3D、2.5D とでも言うべきゲームであった（たとえば、1985年に出たセガの「スペースハリアー」などが疑似3D ゲームの先駆けの一つと言える。Wolfenstein 3D はスペースハリアーを主観視点のダンジョンゲームにしたようなものと考えれば良い）。Wolfenstein 3D がヒットしたために1993年には Doom が、1995年には Quake が世に送り出された。ここまでは PC/AT 互換機と MS-DOS と Windows3.1 の時代の話である。

Microsoft は当初 DirectX をゲーム用の廉価な OpenGL として提供したが、性能と価格差が縮まるにつれ、DirectX と OpenGL は競合し始めた。しかし、DirectX または OpenGL という寡占的で適度に競争がある3D グラフィクス環境が生まれたこともあり、またおそらくは Microsoft 以外でも、Nvidia や ATI などが自社のグラフィックカードを普及させるために PC ゲームを後押しした。2000年代の Nvidia と ATI の間の熾烈な争いはまだ記憶に新しいが、のちには Intel が GPU 競争にくわわった。GPU は CPU よりも進化が加速し、PC のグラフィクス性能は急速に向上した。Dell や Gateway や HP などの PC/AT 互換機メーカーもやはり PC の普及のためにゲームのデベロッパーにコンソール機ではなく PC でもゲームを作らせるよう働きかけただろう。これらの要因があわさり、また Windows95 の登場を契機として、（たとえば Tomb Raider シリーズのように）PC ゲームは急速に発展し、2000年代半ばには Half-Life 2 や Call of Duty のように、コンソールゲームを凌駕するものが現れ始めた。

一方日本では、PC の規格は NEC、富士通、シャープなど各社でばらばらで、IBM-PC によってそれらがすべて駆逐されると、国産の PC ゲームもほぼ壊滅した。一方で同時期に日本ではソニーが PlayStation でコンソールゲーム業界に参入。PC ゲーム業界とはくっきりと明暗を分けた。家庭用ゲームとしてなじまない一部の成人ゲーム以外 PC ゲームは姿を消してしまった。そのため PC ゲームは「洋ゲー」または「エロゲー」「同人ゲー」とひとくくりされるようになる。これは私的見解だが、

日本で洋ゲーが流行らなかったのは単に日本に Microsoft のようなメジャーな（というより強引な）企業がなかったからであろう。日本人に PC ゲームを好まない何かの嗜好があるとは思えない。確かに洋ゲーは日本のテイストに合わないところがあるが、それと日本人が PC というゲームのプラットフォームになじまない、Mod が日本人の気質や日本の風土習慣になじまないこととは違うと思う。実際ソフトウェア産業や情報教育のすべての領域において、たとえば Ruby を開発したまつもとひろゆき氏のような、フリーソフトウェア開発に関わっている人を大勢見いだすことができる。

洋ゲーは、一昔前の日本では、ゲームバランスが雑で美味、繊細緻密な国産コンソールゲームに劣るものとされた。多くのゲームは大手のゲーム情報誌にレビューすら載らなかった。PC ゲームは日本人のごく一部の秋葉オタクのごとき連中のみ盛んなサブカルチャーの一種とみなされた。しかし、それも今は昔、良質な洋ゲーの中にはシリーズ化するものも現れ、ますます巨大な資本が投入されるようになると、洋ゲーのクオリティは飛躍的に改善された。Call of Duty 4: Modern Warfare (CoD4: MW) や Grand Theft Auto: Vice City (GTA: VC) などのように、世界市場において、1000万本以上を売り上げるような大ヒット作が生まれるようになり、日本のゲームユーザーにも漸く洋ゲーが広く受け入れられるようになりつつある（CoD4:MW の売り上げは1400万本、GTA:VC は1300万本。ちなみに世界で一番売れたのはスーパーマリオブラザーズで約4000万本である）。

## 2.2 Valve と Half-Life

Valve 社は最初、id Software から Quake エンジンライセンスされて出来た群小のデベロッパーの一つにすぎなかった。Quake はゲームデベロッパーの参入障壁を下げ、Mod というゲームの楽しみ方と、ゲームエンジン・ビジネスモデルを創始した。しかし、既存のゲームエンジンを利用しただけで誰にも面白いゲームが作れるようになったわけではなかった。言わば、同工異曲のつまらない FPS が粗製濫造されたわけである（現在の i アプリや Android アプリの氾濫に似ているといえる）。また多少売れたとしても id Software に支払わなくてはならないライセンス料が高くて、うまみがない。いわゆる利ざやがないので、次の作品に投資する資本も蓄積されない。あまりやる気のない Quake の亜種のほとんどはまたたくまに淘汰されていった。

Valve を興した Gabe Newell は1983年にハーバード大学を中退、Microsoft に入社、Microsoft Windows 1.0 (1985年) から3.1 (1993年) の開発に携わった。1993年

に id Software から Doom がシェアウェアとしてリリースされると、John Carmack に連絡を取り、これを Windows に移植した。

Newell は Doom に PC ゲームの可能性を見いだした。当時出たばかりの日本製の家庭用ゲーム機スーパーファミコンや、或いはセガのメガドライブなどと比較した上で [Newell1] [Newell2]。ゲームのプラットフォームとして、自らが手がけた Windows という OS にも未来を確信していただろう。彼は Windows95 と Direct X がリリースされるとほぼ同時期、Windows 開発者という将来を約束された地位を捨てて、1996年に Microsoft 社を去り、Valve 社を設立、Quake エンジン改造した GoldSrc というエンジンを開発し、1998年に初めてリリースしたのが Half-Life というゲームだった。Half-Life は確かによく出来たゲームであったが、とびきり優れた、画期的なゲームだったとは言えないと思う。リリースされて10年後で単体では930万本。Half-Life シリーズ全体で2000万本であるから、大ヒット作には違いないが、そのくらい売れたゲームは他にもいくらかある。ところが Half-Life の Mod として作られた Counter-Strike がオリジナルの Half-Life に負けぬほどの人気を獲得し、Valve は Counter-Strike の開発に社員をアサインするようになり、2000年にはとうとう Counter-Strike の権利を買って、自社のラインナップに加えた。

当初、Counter-Strike は Half-Life の単なる Mod に過ぎなかったから、Half-Life を買えば Counter-Strike も遊ぶことができた。中には Counter-Strike を遊ぶために Half-Life を買う人もいただろう。従って Counter-Strike と Half-Life のどちらがより売れたか、ということを厳密に知ることはできない。Half-Life の続編 Half-Life 2 が出ると、Counter-Strike は Counter-Strike: Source という独立したゲームになり、Valve のラインナップの一つとなった。2011年までに Counter-Strike シリーズは2700万本売れたそうである。

世界にはさまざまな Mod があるが、Counter-Strike は先駆的である上に、今日でもこれほどの成功事例を見いだすことは困難である。Counter-Strike の登場は Mod という意味さえも変え、Valve の原体験となり、以後 Valve は Mod コミュニティとともに成長する会社になった。

Half-Life の続編として作られたのが Half-Life 2だが、Quake にライセンス料を支払わなくてはならない GoldSrc エンジンをやめて、独自に Source エンジンを開発した。Half-Life 2は2004年当時極めて先進的なゲームであり、現在でもおそらく PC ゲームの中では最も多く売れたゲームである (CoD などは Xbox や PS3 などの

コンソール機用の売り上げが大きいということだろう)。他にも Source エンジンには Team Fortress 2、Left 4 Dead シリーズ、Portal シリーズなどを派生した。

### 2.3 Mod と Source エンジン

Source のゲームはゲーム画面のスクリーンショットを取る機能がもともと備わっており、動画を保存できる機能さえある。それらは Youtube や個人のブログだけでなく、ユーザであれば Steam の公式サイトに2GB までアップできるようになっている。非商用であれば Mod は自由に作ってよい。このような自由度の高さのために、Source は Mod コミュニティに絶大な人気があり、またそのような長年の実績の上に Valve という会社が成り立っているのである。

UGC とは何か、ということを確認に定義するのが困難のように、Mod コミュニティはあまりにも大きく、国際的で、多様なので、Mod とは何かを定義するのは難しい。素人の日曜クリエイターもいればクラッカーやチーター (犯罪者) もいる。二次創作に没頭するものもあれば、Mod という媒体から一次創作を試みるものもいる。本業から離れて無報酬で楽しむプロもいれば、同じプロでも生業は適当にこなして趣味を本業にするためにむしろ Mod 制作の方に本気を出す者もいる。単なる遊びのつもりの人もいれば Mod や UGC に新しいビジネスモデルを模索する人もいるだろう。Mod コミュニティがどんな作品を生み出すか予測することは困難である。著作権者から見ると明確な著作権侵害でない限り Mod コミュニティを制御するのは難しい。特に、非営利目的の (無報酬で制作し、無償配布されるような) Mod に関しては、Mod 制作者は営利や名声よりもむしろ自由な創作活動を選ぶ者が多い。

Mod コミュニティが無秩序で相互疎通がなければそれは経済的にも政治的にも創作的にも無価値だが、一つの目的に方向性がそろい、組織化されれば、それは巨大な金と力になり、文化創造の源泉ともなり得る。無秩序からの自己組織化、というネット上でしばしば見られる現象である。ある者はネット上で草の根的に連帯することのみで喜びを得られるが、またある者はネット社会を支配することを至福とするかもしれない。

Mod は基本的に製品版を正規購入していれば無料である。Mod の人気が高まることによって本編が売れることを著作権者は期待できる。しかし、Source にも完全に無料な Mod もある。

Alien Swarm というゲームは完全に無料である。もともと Alien Swarm は Unreal の Mod だったが、Valve が

そのModチーム Black Cat Games を丸ごと雇ってしまい、彼らを Left4Dead2 の開発チームに編入してしまった。つまり、強引に Unreal 陣営から引き抜いてしまったわけだ。さらに Source エンジンで Alien Swarm をリメイク。Valve はこれをソースコードごと公開してしまう。

Half-Life 2 はソースコードの一部を書き換えてダイナミックライブラリ client.dll と server.dll をビルドすることができる。これらダイナミックライブラリをリンクして動く hl2.exe という実行ファイルはソースコードが提供されていない。hl2.exe を自分でビルドすることは事実上不可能だ。hl2.exe はしかし、Half-Life 2 を買わなくても無料の Source SDK 2007 Base というものについてくるので、この無料の hl2.exe と、Mod として作られた client.dll と server.dll を合わせれば、まったく無料で配布できる Mod を作ることもできる。

つい最近、2012年9月にローンチされた Black Mesa という Mod がある (図1)。8年間をかけ40人のボランティアが作ったもので、初代の Half-Life を Half-Life 2 のクオリティでリメイクしたものだ (Half-Life 2 や Counter-Strike: Source に出てくるモデルを利用しているが、オリジナルなものも多い)。Source SDK 2007 Base の hl2.exe で動くので、無料でプレイすることができる。Black Mesa は製品版に遜色ない (初代 Half-Life を明らかに凌駕するクオリティの) Mod であるが、このようなものがまったく無料で手に入ることに、我々は驚嘆せざるを得ないし、Mod コミュニティや、またそのような Mod を作らせてしまう Valve にも感服する。おそらく他のゲームエンジンにはこのようなコミュニティは育たないだろう。ある特殊な条件がそろって初めて Mod 文化は生まれ育ち栄えるのである。Black Mesa の開発ボランティアたちがどのような意図で Black Mesa を制作し、また彼らが何を本業にしているのか、非常に興味深い。

Team Fortress 2 というゲームもまた2011年に無料になったがこれはアイテム課金制に移行したからである。Hammer エディタでマップを作ることもできるが、ソースをコンパイルするのはおそらく不可能だろう (Source SDK で Half-Life 2 の client.dll と server.dll をビルドするためのソースはついてくるが、Team Fortress 2 のものは提供されていない)。

通常、Mod というものは、匿名の二次創作者、三次創作者などが勝手にコラボして作り出すものである。筆者は法律家でもなく法学者でもなく法務部門の人間でもない。現場のプロデューサーでもディレクターでもない。

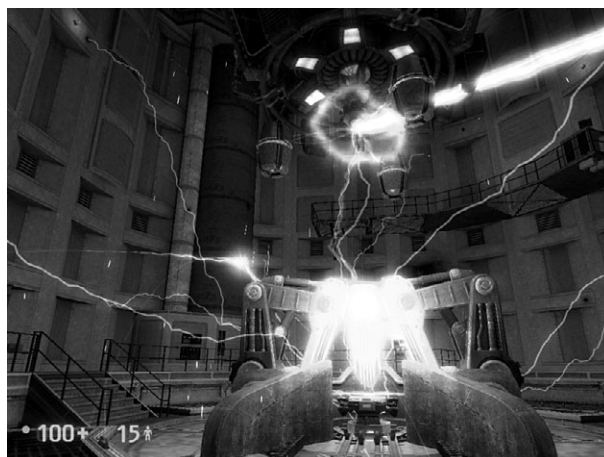


図1 Black Mesa スクリーンショット

単なる研究者である。あまり法律のことについてここに書きたくはないのだが、必要最小限の、一般論を述べておきたい。Mod が許容されているコンテンツ、たとえば初音ミクや Source SDK などについても、二次創作、三次創作と、不特定多数匿名の作者が関与した場合に著作権はどうなるのかなどについて明確に規定されているのではない。GPL (GNU ライセンス) のように、ライセンスが派生コンテンツに再帰的に伝播していくようにはできていないのである。明らかに現状の Mod の規約やガイドラインは GPL より不整備であり、法的にマネージメントするのが困難な状況にある。法整備は明らかに必要だが、現状ではすべて一次創作者の意向に委ねられている。

John Carmack や Gabe Newell らもまた、必ずしも無条件に自社のライセンスをハッカーたちのコミュニティに「解放」しているのではない。たとえば Valve のオンライン対戦ゲームの Counter-Strike や Team Fortress 2 などの場合にはシングルプレイヤーゲームの Half-Life 2 よりもはるかにセキュリティが厳しく、クラッカーやチーターは厳しく排除される。Mod 制作上での制約も大きい。また、Half-Life 2 もクラッカーによってソースコードが流出するなどして、犯人は逮捕されたが、リリースが大幅に遅れたということがあった。Newell が犯罪者に対して同情的であると考えるのは困難だ。Newell は当然それら Mod コミュニティの自由奔放なユーザーたちを熟知しており、その上で彼らを飼い慣らそうとしているのである。それが Valve という会社の社となっている。

筆者の真意は、GNU 運動や Mod、初音ミク現象などを、商業コンテンツの生産者の立場ではなく、コンピュータサイエンスの研究者の立場でとらえ、分析したい、ということにある [Nagae3]。たとえばつい最近、2013年1月11日、ソフトウェア解放活動家の Aaron Swartz が司法取引を拒否して35年の求刑を受け、獄中で自殺し、フリーソフトウェアのコミュニティに衝撃を与えた。サイバー法学者 Lawrence Lessig も強く非難している [Lessig]。Swartz は MIT の電子図書館からオンライン論文を違法入手して無料で配布しようとしたのである。ソフトウェアの世界は広い。単なる窃盗犯もいれば、政治的確信犯もいれば、思想家もいれば解放者もいる。そのようなさまざまな事案について筆者は深い関心を持っているが、読者の方々には本稿をそのようなソフトウェア倫理、とでも言うべきコンテクストにおいて、読んでいただきたいのである。

## 2.4 ゲームエンジンの比較

Quake エンジンの系譜には総計 1 億本を売り上げた Call of Duty シリーズがあり、「洋ゲー」に疎い日本人にも、FPS を知らしめるのに貢献した。だが、最も成功したゲームエンジンは、比較的後発であるが、Epic Games 社の Unreal である、と言い切ってもよいと思う。Gears of War シリーズ、Rainbow Six シリーズ、Splinter Cell シリーズ、Medal of Honor シリーズ、などを出している。

Cry エンジンはもともと NVidia がグラフィックカードのベンチマークテスト用に開発したものだが、もし最先端のグラフィックスの美しさを求めるのであれば、Cry エンジンの方が良いかも知れない。F.E.A.R.などを生み出した LichTech エンジンはもともと DirectEngine と呼ばれ、Microsoft のサポートで生まれたものである。本家 Microsoft の Halo シリーズは独自の Halo エンジンを使っている。この他、ベルリンの壁崩壊後に低予算と高い基礎技術力を売りに勃興しつつある東欧のデベロッパーによる、クロアチアの Serious エンジン (Croteam 社、Serious Sam シリーズ)、ウクライナの X-Ray エンジン (GSC Game World 社、S.T.A.L.K.E.R シリーズ) などが注目に値する。

最近話題の Unity は、比較的簡単なピンボールゲームから、高度な FPS まで作ることができ、PC だけでなく iOS や Android、Flash にも対応しており、任天堂の Wii U も Unity をサポートすることになった。PC ゲームとして有名なものはまだ作られていないようだが、携帯でも動くマルチプラットフォームのゲームが多いように思われる。

最近のゲームエンジンは教育用にもインディー向けにも配慮されており、地形自動生成機能が付いていたり、物理エンジンが標準装備であったり、アセットの管理やモデリングやゲームの実行確認が一つのグラフィカルな開発環境の上でできるようになりつつある。

一方 Source エンジンは、未だに Quake 時代を引きずっているような、古めかしく難解なゲームエンジンだと見なされている。主なプラットフォームは未だに PC であり、Xbox360 と PS3、MacOSX には最近対応し、Linux は現在ベータテスト中だが、Wii や携帯端末、Flash (Web) にサポートしようというつもりはないようである。Gabe Newell は Windows 8 が従来の PC ゲームの世界を一変させる「カタストロフィ」だと言っている。その真意を推察してみる。Windows 8 は Xbox 360 で成功を収めた Xbox Live というオンライン配信システムを Windows にも持ち込もうとしている。一方で Valve は長期戦略的に、パッケージ販売に早々と見切りをつけ、

Steam というゲームコンテンツ販売兼クラウドサービスを独自に開発してきた。PC のオンライン販売環境はこれまで事実上 Steam しかなかったから、Epic も EA (Electronic Arts) も、オンライン販売はおおむね Steam を利用してきた。しかるに、Microsoft が Xbox Live を Windows の標準機能として搭載すれば、Steam と Windows は真っ向から競合する。Epic や EA のパッケージ版はもはや売れなくなり、Xbox Live 経由で売らなくてはならなくなるかもしれない。或いは Steam から Xbox Live に乗り換えるかもしれない。Xbox Live は Apple Store や Google Play に対抗できる Microsoft の切り札である。デスクトップの一番目立つところに Xbox Live のアイコンが飾られるだろう。仮に Steam によるオンライン販売が今後も Microsoft によって認められても、Steam は Windows 上では日陰の存在、脇役であって、売り上げを奪われるかもしれない。せっかく苦心して独自販路を開拓してきたのにそれを捨てて逆に Microsoft にショバ代を払わなくてはならないかもしれない。Valve には死活問題である。そこで Linux ベースの自社製コンソール機に活路を見いだすのではないかと噂されている [Newell3]。Windows の元開発者であり、Windows プラットフォームで成功した Gabe Newell が今や Windows から疎外されようとしているのは皮肉である。

Source エンジンを使ったサードパーティのゲームはほとんどなく、どれもヒットしていない。Unreal とは大きな違いである。Source エンジンの Mod コミュニティでの影響力を思うと不思議な気がする。Steam が有能な人材をいちやくヘッドハンティングしてしまうので、独立系のデベロッパーが育ちにくい、Mod 制作者も独立して苦勞するより Valve に雇われたい、ということもあるかもしれない。

Source SDK のマップエディタである Hammer Editor は GoldSrc 時代からほとんど進化しておらず、つまりマップの基本設計は14年前の Quake の頃と同じで、今では古色蒼然としてきた。Hammer で図面を引くように建物や地形を作り、マップをビルドした後ゲームを起動しなくてはならない。テキストチャやモデルのコンパイルにいちいち DOS 窓を開いてコマンドを実行しなくてはならない。サードパーティから GUI がいくつか提供されているが、キャラクターモデリングに Softimage を使わざるを得ないのも、時代の主流からはずれている。Newell 自身が、Hammer 他のツールは苦痛だ、と言っている [Newell4]。よく言えば「枯れている」が、悪く言えば「時代遅れ」なのである。

ただ、世間一般のゲームエンジンは、ひたすら FPS に最適化されているか、iOS や Android などのクロスプラットフォームを謳っているか、さもなくば課金システムのように有料のアセットを買わない限り、なかなか開発がはかどらないようなものばかりだ。Mod という入り口からゲーム開発に入ろうとする初心者には、まだ若干 Source エンジンにも利点はあるように思う。

Mod は基本的には素人が楽しみながら作るものである。良くある楽しみ方としては図2と3のように既存のモデルを自分の好きなキャラに差し替えて遊んでみるというものだ。図2は、左から、オリジナルの Metropolis (Combine というエイリアンが地球人を警官に人体改造したもの。手にはスタンバトンというスタンガンのような電気ショックを与える警棒を持っている)、その隣はスタンバトンをネギに置き換えたもの (筆者制作)、その隣はキャラを置き換えたもの (筆者制作) である。

Mod の良いところは、他人のふんどしで商業レベルのゲームを作れるところ、或いは商業レベルのゲームの一部を自分自身の手でいじれるところである。もし Mod でなければ、商業レベルのクオリティを達成するところまでで疲れ果ててしまう。学部四年間の学生の場合には多くの場合途中で頓挫するか、適当なところであきらめてしまうだろう。学生レベルの作品を作って満足してしまうよりも、商業レベルの作品をいじってみてそのできばえに感動するとともに自分の無力さに気付くことの方が重要ではなからうか。商業レベルの作品を作るコツ、その内部構造、作業量の見積もり、学生レベルから一步抜け出すために必要なものにも、Mod 制作の過程で気付けるはずだと考えている。たとえばテクスチャを描くにしても、自分が作ったゲームでは適当に描いたテクスチャでも良し悪しの違いがわからないが、プロが作った作品の中に入れてみると、自分がどれだけ下手かがわかる。どのくらいまじめにテクスチャを描かねばならないか、自分で気付くことができるのである。

他の SF 的な FPS とは違い Half-Life 2 にはリアルな町並みや一般市民が登場する。たとえば Counter-Strike と比較すると Counter-Strike のキャラクターは必ず武器を持っている。目も口も動かない。戦うより他の動きは用意されていない。他の FPS も似たりよったりであり、ただ町の中をぶらぶらと歩き回り、声をかけると反応する、などというゲームを作るのは難しい。つまり、かつてのシェンムーのように RPG を3D 化したようなもの、村人に声をかけるとゲームが展開していくといったゲームを作るには Half-Life 2 をベースにするのが楽である。Gears of War や Halo のようなゲームを作るのであれば、



図2 既存の武器のモデルをネギに置き換え、キャラクターを差し替える



図3 敵キャラをゲームの中で差し替える

必ずしも Source エンジンを使う必要はないように思う。

ちなみに、Half-Life もまた普通の警備員や科学者などがたくさんゲームの中に登場する。彼らは目もまぶたも動かず、口はばくばくするだけだ。しかし話しかけると一緒についてきて鍵を開けてくれたり、一緒に敵と戦ってくれたりする。このようにゲーム内で AI (人工知能) を使うことに関して Valve は Half-Life の時代から積極的だ。ゲームの AI を学ぶ教材として Source エンジンは悪くない選択肢だ。

もう一つ Source を使うメリットとして、今後の Valve という会社に賭ける、という意味で現在のエンジンに習熟しておく、というのもありだと思う。Half-Life の GoldSrc エンジンを知っていた人が Half-Life 2 の Source エンジンにいち早く移行できたのは当然だろう。2005 年頃の Source の Mod 職人はみな GoldSrc から来た。Valve が今後優れた新しいゲームエンジンを出すための準備として今 Source エンジンになじんでおくのである。

### 3. ゲーム教育への導入

本稿を世に問うにあたって一つ申し上げておかねばならないことは、大学で Mod を教材にするということは、正規のゲーム教育のカリキュラムとは距離をおき、UGC 的作品制作や作品公開の可能性を探っていくというコンテキストに沿っているということである。Mod 制作はあくまでも筆者の研究室で実験的に取り組んでいるものである。

大学で、学生どうしの自主制作やサークル活動ならばともかくとして、カリキュラムの一貫として Mod を作って、ああ楽しかったでは済まされない。何らかのアウトプットが要求される。東京工芸大学芸術学部ではしばしば学生の同人活動と学科やゼミの制作活動が乖離している現象が見られる。学生の志向性とゼミの方向性をなんとかそろえられないものであろうか、学生の創作意欲、同人的な活動意欲をなんとか卒業制作なり、将来の職業にリンクさせられないものか。UGC や Mod が今後社会的に一般化していくのならば、積極的に大学教育もその潮流に対応すべきなのではなかろうか、ということを筆者は常々思案している。

Mod 制作は導入が容易で学生の動機付けも維持でき、初歩的な技能を身につけるのには向いている。では具体的に何をアウトプットとして Mod をカリキュラムに組み込むべきだろうか。

#### 3.1 マップ制作

Mod で一番簡単なのは、おそらくマルチプレイヤーのオンライン対戦ゲームのマップだけ自作する、というやり方だろう。たとえばプリレンダの CG のムービーを作ることを考えると、地形を作り、建物を作り、キャラクターを作って照明を当て、カメラを配置してアニメーションを撮り、それらを映像編集しなくてはならない。一通りのやり方を身につけるのには良いが一つ一つはどうしても浅くなる。これに対して、リアルタイムレンダリングのマルチプレイヤーゲームはマップさえ作ればあとは勝手にゲームになってくれる。実際筆者はすでに五年間にわたり、一年前期の初歩的な 3DCG 制作演習として、Counter-Strike のマップ制作を行わせている。使用ソフトは基本的には先述の Hammer エディタだけである。このエディタは三面図と透視図を見ながら壁や屋根と言った比較的「四角四面」な形状を作り、ガラスのような「割れる物体」(func\_breakable) や開く「ドア」(func\_door)、ハシゴ (func\_ladder) などといったものを作り、そこに既存のモデル (樹木や建物のように動かさない大きなものは func\_static, 椅子や机のように小さ



くて動かしたり壊したりできるものはfunc\_physbox)を配置し、適当に用意されたコンクリートや煉瓦、モルタルなどのテクスチャを貼れば、それなりに実在のマップらしきものが完成する。筆者は当初、まだ右も左もわからない入学したての学生が、3Dモデラーを操作して、1年次後期以降の3DCG制作のきっかけになればと始めたものであるが、次第に単なるマップ制作にとどめず、もっと複雑な、インタラクティブCGと言ってもよいようなものを教えた方がよいのではないかと思うようになってきた。そのため今年度から後述するように Softimage も若干教えるようにしており、来年度からは二年次学生に対して、より高度なキャラクターモデリングを教える予定である。

### 3.2 テクスチャの張り替え

モデルに手を加えずにそのテクスチャだけを書き換えるのは、やはりそれほど困難ではない。マップ作りは多少時間がかかるが、テクスチャ書き換えだけなら半日、馴れていれば一時間くらいでできるだろう。

図4は Counter-Strike の Lee というモデルのサングラスのテクスチャを透明にし、顔の髭などを消し、髪型を変え、眉間の皺などをなくするなどして、普通の人のような風貌にしたものである。形状はいじっていない。

### 3.3 物理属性を持つ剛体モデル制作

Source エンジンには Havok という物理エンジンがついているので、転がったり倒したり壊したりすることができる物体(prop\_physics)をマップの中に作ることができる。簡単な形状ならばマップエディタで作ることができるが、外部のモデラーで作ったモデルを prop\_physics として読み込むこともできる。

多くの場合、キャラクターモデルを作るときに難しいのは、形状のモデリングというよりも、テクスチャのUV展開であったり、テクスチャの描画であったりする。MudBox や3ds max の viewport canvas などを使えば、3D形状の上に直接描画することもできるのだが、筆者は一年前期に softimage を用いて、図5のダルマのような比較的単純な形状のモデルを作成させている。後々のため



図4 既存キャラクターモデルのテクスチャ張り替え

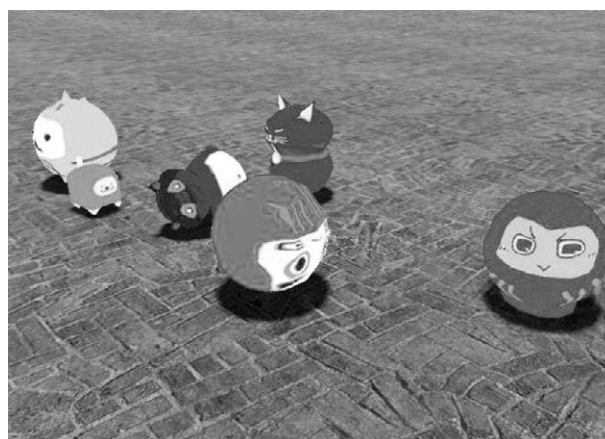
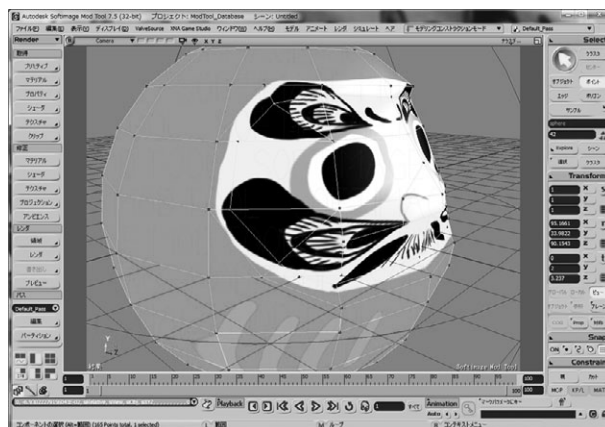


図5 平成24年度1年次授業課題。ダルマのモデリングとテクスチャ描画。

に、モデル制作になれておいてもらうためである。さすがに全員ができるようにはならなかったが、初心者でも半数はそれなりのものを作ることができた。

### 3.4 キャラクターモデリング

Half-Life 2のキャラクターはすべて Softimage で作られているのだが、2004年当時ならばともかく、Softimage でローポリモデリングし、ウェイトマップを塗り、UV展開して、外部の描画ソフトでテクスチャを塗り、それをまた貼り付けてモデルとテクスチャの位置合わせをする。フェイシャルアニメーションを作る。これらの一連の作業を2012年現在行うのはかなり時代遅れ



図6 平成20年度学生作品。まんだら屋の良太のキャラクター  
(左から久美子、月子、テツ、良太)

で苦痛でもある。ただし、今後もCGやゲーム制作の現場においてローポリモデリングの重要性は当分続くと考えられるので、今後の方向性を見守りつつ、カリキュラムを工夫していく予定である。図6は四年前に学生と共同で一から作ったキャラクターである。東京工芸大学のマンガ学科の故畑中純教授に、著書『まんだら屋の良太』のキャラクターの使用許可を得て作成したものである。これらのモデルの服のテクスチャは布地のパターンをGimp 2.6で自動生成しており、顔のテクスチャはマットなままである。マンガの雰囲気を出すためにあまり陰影が出るような描画をしなかったのだが、ただ単にのっぺりした顔になってしまったように思う。

図7は、『まんだら屋』のキャラクター作りの反省から出発したものであり、モデルを一から作るのではなく、既存のモデルを改変することによって目的のモデルを作成する見本として筆者が作ったものである。既存のキャラクターを流用すれば、フルスクラッチから作るよりずっと速く、かつ高品質なものを作ることが可能だ。Half-Life 2のキャラのテクスチャを観察し、マンガ的なキャラではなく、よりリアルなキャラを作ることを目指した。顔のテクスチャもわざとそばかすやほくろなどのキメを描き込んでいる。こうすることによって立体感や現実感を出すことができる。

図7では、それぞれ左側がオリジナルで右側が改変したモデルである。左上と右上の例ではまず実在の20代の日本人女性の皮膚と唇の色で顔を塗り直し、瞳の色を変え、額や鼻筋、まぶたやまつげなどの形を変更、眉、まつげ、髪の毛などを描き込んでいる。二つ目の例では顔の形はほとんど変えずテクスチャだけ描き換えて、髪の毛に房を加えた。三つ目と四つ目の例では顔の形もテクスチャも元のモデルがわからない程度に変えてみたつもりである。

Mod制作は一次作者の権利から逃れることはできないのだから、自分で作ることに固執せず、利用できる



図7 既存のモデルから日本人風モデルを作った例

ものは何でも利用して、真似できるものは真似て、できるだけ手早く作ることが肝心である。我々（特に素人）は、テクスチャは全部手描きで描こうとか、プログラムはDirectXかOpenGLで直接描こうとか、モデルも全部一から作ろうとしがちである。そしてあまりにも道が遙

かなので途中でやめてしまう。使えるものは SDK 付属の素材であれコードであれ、よそから取ってきたパブリックドメインのテクスチャであれ、なんでもかまわず使って、できるだけ速く人に見せられるものを作るべきだ。

図8は、顔のテクスチャに筆者自身の顔のテクスチャを使ったものである。我々はリアリティを増すためにはハイポリにすればするほどよく、キャラクターはポリゴンをたくさん使えば使うほど高精細で写実的なモデルが作れる、と考えがちである。しかしローポリ的手法を経験すれば、問題はポリゴンの多い少ない、ポリゴンメッシュの造形が細かいか荒いかということよりも、テクスチャに実際の写真を使ったかどうかの方が大きく関わることがわかる。目や鼻や口の周りなどには確かにたくさんのポリゴンを使ってこまかな造形をしなくてはならないが、それ以外の部分のポリゴンはさほど多くなくてもよく、品質はほとんどテクスチャによって決まる。

写真とテクスチャの合成は、まず元の円柱状に開かれたテクスチャに自分の顔写真を縦横に拡大縮小しながら位置合わせする。つなぎ目ができるだけわからなくなるように色をませたり髭を描いたりする。ほかしや色混ぜよりは、スタンプツールを使った方が「皮膚のシミや汚れまでコピーできる」ので、自然にシームレスに見せるのに有効である。

テクスチャを無理にポリゴンメッシュに合わせようとすると、テクスチャが歪んで別人になってしまう。むしろテクスチャの見た目が不自然にならないようにポリゴンメッシュを変形させたり、ポリゴンメッシュをさらに分割したりする必要がある。

肌も服もやや大げさなくらいに汚したくらいでちょうど良い。図7の学生服の場合も、あまりにテクスチャがのっぺりしていると平らな板のように見えてしまうので、皺や汚れなどを少しオーバーなくらいに描き込んである。またあごの下の陰影や、頬や唇のテカリなども手で描きこむ方が高度なシェーダーを使うより（敢えて理論的な照明モデルを無視したほうが）安価でかつ効果的にリアリティを出すことができるのである。つまり嘘も方便ということだ。このようなローポリ的手法はCGの教科書には載っていない。むしろ教科書に書かれていることに反していると言える。ドット絵職人の世界に近いものがあるかもしれない。

図5のダルマモデルの場合には、テクスチャはテクスチャで描き、モデルはモデルで形状を作って、あと適当にモデルにテクスチャを貼り付けるだけでも作ることができる。しかし図6、7、8など、通常のUV展開では、UVマップ上のメッシュ変形とモデルのポリゴンメッシュ



図8 実在の人物の顔のテクスチャを利用した例



図9 平成24年度3年次課題作品

の変形を同時並行に行ってうまくモデルとテクスチャが合うようにしなくてはならない。

図9は、今年度の三年次のゼミ生にキャラクター制作をさせた例である。図5のダルマを作り、キャラクターを作るまでおよそ半期を要した。図9一番左がオリジナルキャラクターであり、その右の四体は顔や服の形状、テクスチャなどを改変したものである。今回この紀要に掲載するのは間に合わなかったが、彼らは現在イラスト画によるキャラクター設定担当と、三次元のモデリング担当に分かれて、より個性的なキャラクター制作を行っている。今後2年次の演習科目と3、4年次のゼミ制作を有機的に組み合わせて、より効率的に3Dのアバターやキャラクターを制作できるカリキュラムにしていく予定である。

### 3.5 AI、振り付け、台詞

マップ制作とキャラクターのモデリングが終わってもまだ Mod 制作の半分も終わっていない。むしろその後がずっと大仕事である。マップの中には目に見えるキャラクターやモデルしかないわけではない。マップは、キャラクターを制御しストーリーを展開させるために、絶望するほど多くの、トリガーやノードなど目に見えないエンティティで満ちている。

キャラクターをマップに配置し、ある者は歩かせ、ある者は座らせ、話かけると返事をし、何か頼み事をするかもしれない。つまり村人に話かけることによってミッションが開始されるのである。ミッションを完了して戻ってくればトリガーが作動して（フラグが立って）、ドアが開いたり、村人が誘導してくれたりして、次のマップへ進めるようになるかもしれない。

図10は一般市民（村人）たちをいろんな姿勢で立たせたり座らせたり壁に寄りかからせたりした例である。通常、プレイヤーがある場所（トリガー）を通過すると彼らは立っている姿勢から座ったり、座っている姿勢から立ち上がったたり、歩き始めたりするのだが、これらのややこしい動作をすべて仕込まなくてはならない。プレイヤーの視点（カメラ）を制御してカットシーンを入れることもできる。すべては Source エンジンに備わっている機能だが、それらを使いこなすのは極めて難しい。

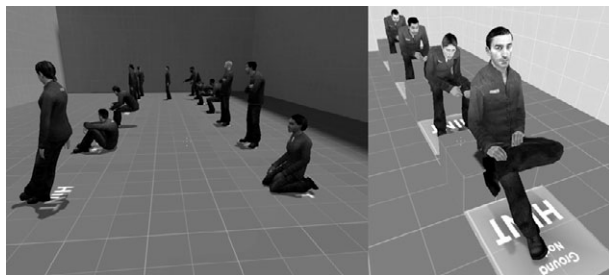


図10 キャラクターの制御

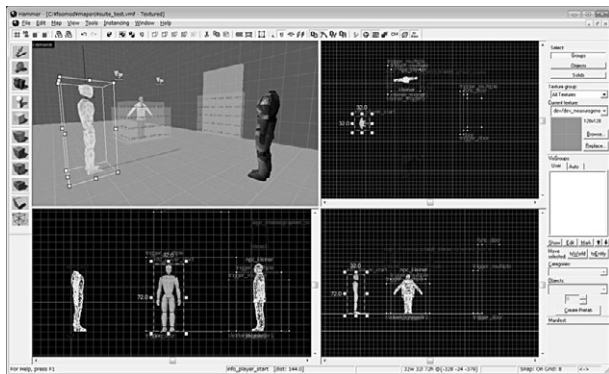


図11 トリガーとストーリー進行

図11には、プレイヤー（左）とスーツ（右）があり、奥に白衣の男が立っていて、その右奥にドアがある。最初ドアはロックされているがプレイヤーがスーツを着るとロックが解除される仕掛けになっている。男に話かけると、スーツを着る前には「スーツを着なさい」と返事をする。スーツを着た後には「よろしい。行きなさい」と返事をする。つまり、A、B という二つのトリガーがあり、A が「スーツを着なさい」、B が「よろしい。行きなさい」とそれぞれ返事をさせる。スーツを着る前は A だけが有効で B は無効になっている。スーツを着るというトリガー C があって、C によって A は削除され、B は有効にされ、ドアのロックが解除されるのである。

図12は、プレイヤー（主観視点ゲームの場合はカメラ視点）がベッドの上で目を覚まし、ベッドから起き上がるシーンを仕込んだものだが、たったこれだけのことにも、アニメーションやカメラの複雑な制御を行わなくてはならない。

図13のように、特にシングルプレイヤーの主要なキャラクターには、動作や挙動だけでなく、セリフを吹き込んだり、目線や眉の動きや変形、口パクなどの表情を付けなくてはならない。その表情も、やり方次第では、喜怒哀楽などの感情や心理をかなり複雑に表現できる。

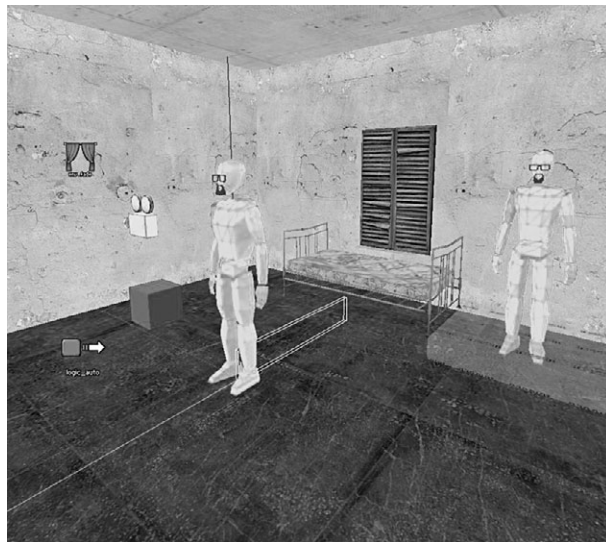


図12 プレイヤー視点の制御



図13 表情

Source エンジンにはキャラクターの制御が非常に複雑かつ緻密であり、比較的古いゲームエンジンには皆無であった、最近のゲームエンジンでもあまり見られない仕様になっている。思うに Source エンジンをうまく使えば、役者も照明も大道具もスタジオも使わずに、全自動でドラマを作ることさえできるだろう。そのためにはそのような根気が必要だ。果たして学部生にそこまで使いこなせるかどうか疑問である。一つのゲームを仕上げるまでにどれほどこのような作業をしなくてはならないかと思うと気が遠くなるではないか。

近年は Half-Life 2 のような、シリアスなシングルプレイヤーゲームが発達し始めてきており、(プリレンダのカットシーンではない、通常のプレイ中のリアルタイムレンダリングでも) キャラクターに繊細な表情の表現が用いられるようになってきた。非常に面白いが、手間のかかる部分でもある。筆者のゼミの学生に関して言えば、男女を問わず、ストーリー性のあるシングルプレイヤーゲームよりは、どちらかと言えばシューティングがメインな、特にマルチプレイヤーの対戦ゲームを好む傾向にある。マップもしくはキャラクター(つまり、ゲームのアセットのほんの一部)を制作すれば、そのままゲームとして成立してくれるので、楽なのだろう。しかし、それでは、Mod 環境という箱庭の中で遊んでいるだけではないのか、という批判を招くだろう。

Mod の世界は良くも悪くも新しいことが面白い。カリキュラムはそれに対応して毎年作り直さなくてはならない。こうしなきゃならないという方法論もない。学生の反応も見定めつつ方向性を手直ししなくてはならない。教育者としてそこにおもしろさを見いだすのは危険だということは自覚しているつもりである。大学の教育や研究には限界があり、企業には企業の利点がある。大学が企業の真似をするのは多くの場合非効率である。学部四年間という長いようで短い在学期間で、彼らが Mod のどの部分を習得すべきか、カリキュラムとしてどこまで無理なく到達可能か、何を彼らのアウトプットとするか、教員はマネジメントという形でどこまで彼らの制作に介入すべきか、ということに、日々悩まされているのが現状である。

#### 4. 今後の展望とまとめ

Youtube などの動画サイトを見れば Mod がどれほど熱狂的にゲームファンに支持されているかわかるが、日本のマスメディアがそのような海外の状況をほとんど一般人に伝えてくれないのは残念である。もちろんインターネットや Youtube に国境がないように Mod にもない。日本国内にも Mod のファンはいくらでもいる。表面的

に知られていないだけで。

マップの制作、テクスチャ描画、キャラクターのモデリング、スクリプトの制御、台詞の吹き替え、身振り手振りの仕込み、トリガーの仕込みなどは、それぞれはさほど難しくない。それらの個々の要素だけでも Mod にはなり得る。しかしそれらを総合して一つの新しいゲームを作るのは大変な作業量となる。単なる個人レベルの Mod と、Mod チームを組んで作る本格的なゲームとしての Mod には天と地の差がある。当然のことではあるが、ゲームを作るにはプロジェクトのマネジメントが必要だ。大学の卒業制作や大学院の修了制作に作る Mod は少なくともインディーズのレベルに到達してはならない。そのためには制作チームをインディーズと同程度までに組織化する必要があると考えている。

先日、筆者は町田市立国際版画美術館でインディーゲームに関するパネル・ディスカッションを行った。私以外のパネラーの一人箕輪氏は版画美術館の学芸員であり、もう一人の牧氏は、東京工芸大学芸術学部メディアアート表現学科卒で現在フリーランスのサウンドプロデューサーでインディーゲームのプランナーである。牧氏は日本のゲーム産業でもすでに Mod は利用されていると指摘した。Wii U が Unity をサポートするというような話 [Unity1] ではない。牧氏は自分の企画プレゼンの企画書の図版や、ゲームのプロトタイプ開発に Mod を使っているという。以前企画書の図版は、プレイ画面の例を Photoshop でそれらしく描いていたが、今ではゲームエンジンでさっとユーザインタフェースを作り、ゲームプレイ画面を作る。そうするとただの文字だけ、ただの挿絵だけのプレゼンよりも企画が通りやすいとのことである。

多くのスタッフをアサインした本番のゲーム開発ではともかく、その前の段階の、ゲーム企画やプロトタイプ開発においては、工程がずっと少なく、試行錯誤しやすく、少人数低コストで制作可能な Mod が有効なのである。このような Mod を利用したプリプロダクション的、或いはラピッドプロトタイプ的な手法は、開発途中にもゲームバランスのチューニングやリリース直前の最終調整にも(必ずしも好ましいことではないかもしれないが)用いられる可能性がある。つまり一連の工程の全ディレクションに影響を与え得る。或いは映像作品のプリプロダクション段階に Mod が使われることも十分あり得る。Mod はすなわちリアルタイムレンダリングであり、CG のプリレンダリングの前にざっと流れを見て修正を加えるのに有効だ。実写映像のコンテ代わりにもなるだろう。

これらが則ち大学教育における Mod 制作が現状で直

接活かせる場面である。

海外では Mod チームが丸ごと企業に買われたり、チームのメンバーがヘッドハンティングされたりする。特に Valve が積極的にそういうことをしている。日本にはそのようなキャリアパスはほとんどまったく存在していない。しかし、ゲーム会社の内部では、すでに Mod が制作の現場でプリプロダクションや企画に使われ始めていて、学生時代に Mod を作っていたということが社会に出て無駄になるのではない、ということを感じた。

Unity のサイトにいけば「簡単にリッチな演出が実現できる」ことを [Unity2]、Unreal のサイトでも「開発費や技術を持たないアーティストやインディーズでも使いこなせる」ことを謳っている [Unreal]。なるほど、しかし我々はそんなうたい文句に何度もだまされてきた。ゲームエンジンの選択は重要なことであり、その日進月歩の進化に常についていく必要がある。我々教育現場にいる者としては、多すぎる情報に振り回されるより、Mod で何ができるか、Mod で何かをするためにはどういうマネジメントをするかについて、より深く配慮すべきかと思う。

## 参考文献

- [Eto] 江渡浩一郎：ユーザー参加型の価値を追究する新しい学会ニニコ学会βの試み（情報管理 Vol. 55 (2012) No. 7 P 489-501)
- [Lessig] Lawrence Lessig: Prosecutor as bully, <http://lessig.tumblr.com/post/40347463044/prosecutor-as-bully>
- [Levy] Steven Levy: Hackers: Heroes of the Computer Revolution, 1984
- [Nagae1] 永江：インターネットにおける自発的コミュニティの形成、特に Web 日記に関して（尚美学園短期大学研究紀要 14, 59-80, 2000-03-31)
- [Nagae2] 永江、牧周作、箕輪裕：デジタル・インターコネクション2012 第一部『インディーズ・ゲームの世界によろこそ！』パネル・ディスカッション インディーズ・ゲームとは何か？』
- [Nagae3] 永江：グラフィカルインタラクティブシステム前史、東京工芸大学芸術学部紀要11, 53-62, 2005-03-31
- [Newell1] Creative Minds: Gabe Newell – Feature: From Half-Life to the stars – <http://www.computerandvideogames.com/172835/interviews/creative-minds-gabe-newell/>
- [Newell2] Gabe Newell [http://half-life.wikia.com/wiki/Gabe\\_Newell](http://half-life.wikia.com/wiki/Gabe_Newell)
- [Newell3] Valve’s Gabe Newell discusses the frontiers of gaming, from wearable technology to “tongue controllers” <http://venturebeat.com/2012/07/25/valves-gabe-newell-talks/>  
One, we’re trying to make sure that Linux thrives. Our perception is that one of the big problems holding Linux back is the absence of games. I think that a lot of people — in their thinking about platforms — don’t realize how critical games are as a consumer driver of purchases and usage. So we’re going to continue working with the Linux distribution guys, shipping Steam, shipping our games, and making it as easy as possible for anybody
- who’s engaged with us — putting their games on Steam and getting those running on Linux, as well. It’s a hedging strategy. 我々は Linux が今後発展するかどうか見極めようとしている。我々の認識では Linux が抱える大きな問題はゲームコンテンツがないということだ。ゲームのプラットフォームにとって、ゲームコンテンツというものがどれほど大きな意味を持っているか多くの人は気付いていない。それで我々はますます Linux プログラマと協働し、Steam やゲームを売り、ゲームを Steam にのせ、Linux で走るようにし、それらをできるだけ容易に購入できるようにしたい。これはいざというときの保険だ。
- I think that Windows 8 is kind of a catastrophe for everybody in the PC space. I think that we’re going to lose some of the top-tier PC [original equipment manufacturers]. They’ll exit the market. I think margins are going to be destroyed for a bunch of people. If that’s true, it’s going to be a good idea to have alternatives to hedge against that eventuality. But when you start thinking about a platform, you have to address it. You have to address mobile. You have to look at what’s going to happen post-tablet. If you look at the mouse and keyboard, it was stable for about 25 years. I think touch will be stable for about 10 years. I think post-touch, and we’ll be stable for a really long time ? for another 25 years.
- 私は、Windows 8は PC 世界の破滅をもたらすものだと考えている。PC ゲーム業界の有力者を失うかもしれない。市場から撤退しようとしている。多くの人にとって利ざやが破壊される。そのような不測の事態にそなえて予防線を張っておくのは良い考えだ。しかしプラットフォームといえば、携帯についても言及しなきゃならないな。タブレットの後には何が起きるだろうか。マウスとキーボードは25年間も使われてきたがタブレットは10年もつかない。タブレットの後にくる、もう後25年もつものとは。
- [Newell4] SteamCast: Transcript for Steamcast Episode #47: “Gabe Newell 2011 Interview, Part 1” <http://www.thesteamcast.com/episodes/47/episode47transcript.pdf>  
So, our current tools are... very painful, so we probably are spending more time on tools development now than anything else and when we’re ready to ship those I think everybody’s life will get a lot better.  
我々の現在の開発ツールを使うのは非常に苦痛であり、我々は他の何より今はツールの開発に時間を割いているかもしれない。そしてそのツールを公開できれば、みんなの暮らしはもっと良くなるだろう。
- [OECD] PARTICIPATIVE WEB: USER-CREATED CONTENT, Working Party on the Information Economy, DSTI/ICCP/IE (2006) 7/FINAL, OECD 2007 <http://www.oecd.org/internet/interneteconomy/38393115.pdf>
- [RMS1] GNU General Public License, version 1 February 1989, <http://www.gnu.org/licenses/gpl-1.0.html>
- [RMS2] GNU 一般公衆利用許諾契約書 GNU GENERAL PUBLIC LICENSE Version 2, June 1991 <http://www.gnu.org/licenses/gpl-2.0.html>
- [Unity1] Unity Technologies Enters Extensive Agreement With Nintendo to Support Wii U (Unity Press Release Sep. 20, 2012) <http://unity3d.com/company/public-relations/press-releases>
- [Unity2] あなたに最適な Unity は？ <http://japan.unity3d.com/unity/licenses/which-license-is-best-for-me>
- [Unreal] Free Game Engine for Indie Game Development: <http://www.unrealengine.com/udk/>