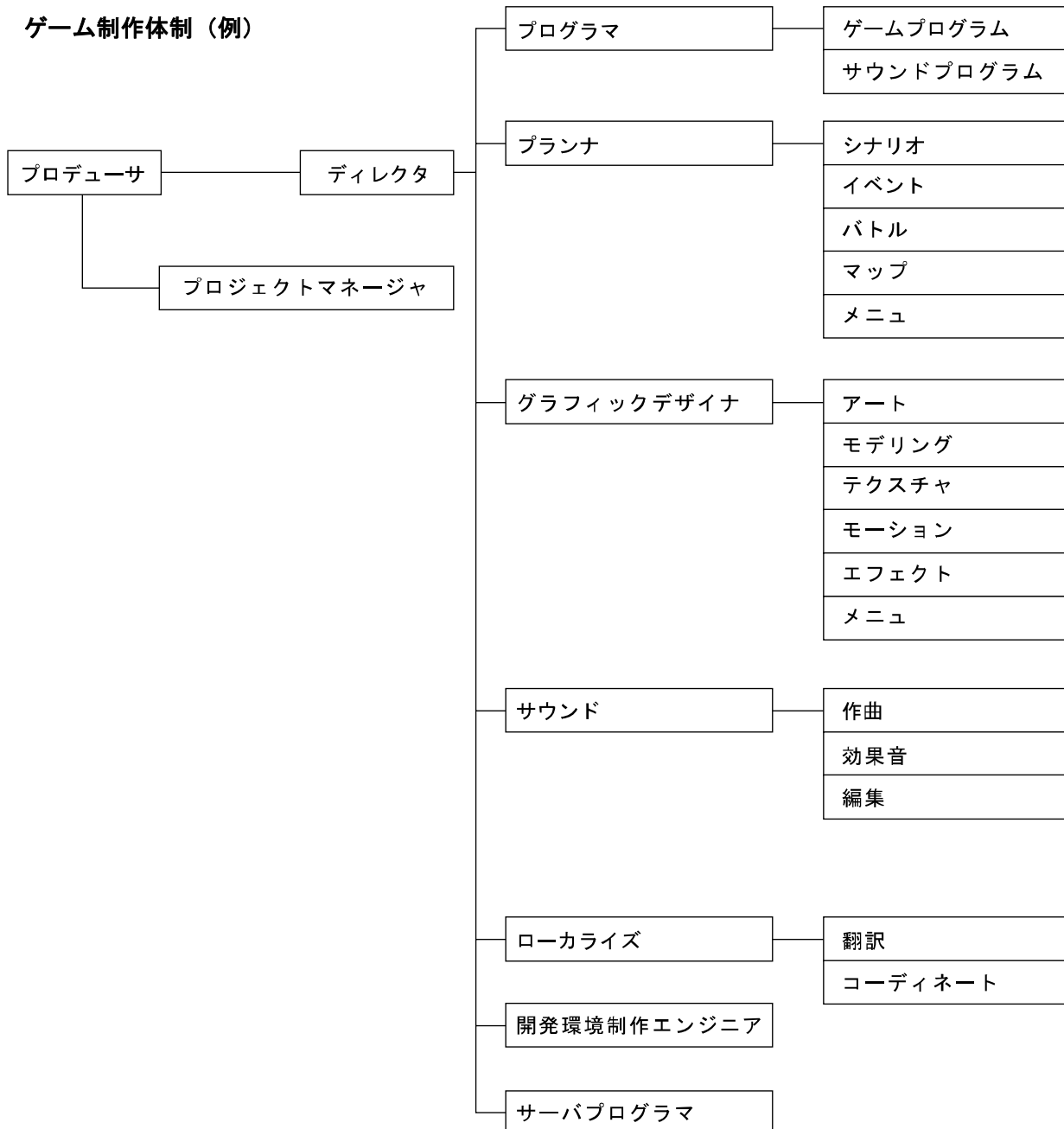


Ⅲ. 開発者のキャリアパスと教育カリキュラム

1. 開発部門の職種

ゲーム開発には、下記組織図例のような職種・職務が必要とされる。開発するゲームの種類、内容、開発チームの大きさによって、必ずしも添付の総ての職種・職務が明確にされない場合がある。これらの開発に必要な職種の中で、ゲームを動かす柱の役目を果たすプログラマと開発チームの構成員として最も多いグラフィックデザイナーについて具体的にスキルの分析をし、開発者のキャリアパス、教育カリキュラムへの道筋の整理を試みた。



注記：すべての職種が、同じ組織に属しているとは限らない

プログラマについては、ゲーム機の高度化とゲーム内容の大規模化を主な理由として、細分化・分業化が進んできている。

また、ゲーム本体を制作するプログラマ以外にも、現場に必要なプログラマは制作環境を構築するためのツールを作成するプログラマや、サウンド制御のためのプログラマなど多岐にわたる。今回はプログラマを以下の4つに大別し、それぞれに対するスキル表を作成した。

- ・ ゲームプログラマ
ゲーム本体を作るプログラマ
- ・ サーバプログラマ
オンラインゲームのサーバ部分を担当するプログラマ
- ・ 開発環境制作エンジニア
ゲームを開発するための開発環境を構築するプログラマ
- ・ サウンドプログラマ
サウンドドライバやツールなどを作成するプログラマ

人材の流動度を考えた場合に、必要なスキルが分かれており、それぞれ分けて考えやすいというのがその理由である。

勿論、ゲームプログラマに関しては、システムを担当するもの、グラフィックを担当するもの、AIを担当するもの、物理シミュレーションを担当するもの、エフェクトを担当するものと、その役割は多数の担当パートに分かれている。担当の分け方や呼び方は、各社、各チームにより異なることが多い。さらに、ゲームの制作をするプログラマは、制作が進むにつれ、別のパートの作業を引き受けること（たとえば敵 AI プログラムがエフェクトを作成したり、プレイヤープログラマがボス敵 AI を作業したりする等）も多く、その役割は流動的である。職種をそれごとに分ける意見もあったが、あまりにも専門的な部分が細分化しているため、それぞれは「知識」という項目に分け、それぞれの専門知識ごとに小項目で必要な知識を提示するようにした。実際、プログラマとしては専門知識以外の部分は共通したスキルが要求されることになり、2 つ以上の分野の専門知識を持ったプログラマも少なくない。スキル表としては、すべての知識項目を満たす必要はなく、担当分野の項目を満たしていればかまわないという見方を想定している。

ゲームプログラマ以外の3つの職種は、それとはかなり異なる知識を必要とする。

サウンドプログラマは多くの場合、ゲーム本体のプログラマとは別に、サウンドアーティスト達と共同で作業することが多い。また、複数のタイトルで共用されることを前提に作業することが一般的である。近年のハードウェアではサウンドに関してソフトウェア波形処理が一般的になってきたため、サウンドプログラマに関しても、リバーブや3次元音響など、音響工学の知識が必要になってきている。

サーバプログラマに関しては、昨今のオンラインゲームが多くなっていく中で必要度を増している職種である。ゲームのプログラマとは全く違う知識が必要となり、人材を育てるのが急務である。データベースに関する知識や、サーバハードウェアに関する知識等が必要となる。

開発環境制作エンジニアは、主にデザイナーやプログラマが使うツールを作成する。大規模開発の効率化には欠かせない職種である。ゲーム開発の作業は主として PC 上で行うため、Windows など PC 上での GUI プログラムの知識を多く必要とし、さらにゲーム中の処理に関しても幅広い知識が必要とされる。

1-2. グラフィックデザイナーの職種

ゲームのグラフィック制作は、ゲーム機の高度化により、表現力が向上する過程で、制作工程が細分化されてきている。勿論、開発するゲーム内容、開発チームの規模により工程の細分化度合も異なる。グラフィック制作という意味では、グラフィックデザイナーとして職種は一つとも言えるが、向上した表現力を追求するためには、工程を細分化し、それぞれの工程において深堀が求められてきた。本報告書においては、それらの深堀する専門範囲を職種として分けて分析している。開発チームにおいては、複数の職種（工程）を同一人物が担うことも多い。その意味では、複数の工程を担当する場合には、スキル表における該当職種（工程）の総合的なスキルが求められる。

まず、工程を踏まえ、職種を以下の7つに大別した。

- ・アートデザイナー
- ・モデルデザイナー（モデラと呼ばれることも多い）
- ・テクスチャデザイナー
- ・モーションデザイナー（アニメータと呼ばれることも多い）
- ・エフェクトデザイナー（VFX デザイナーと呼ばれることも多い）
- ・メニューデザイナー
- ・カットシーンデザイナー

また、デザインする対象として大きくキャラクタと背景に分かれる。キャラクタと背景の分けと前述した 7 つの職種を組み合わせ、キャラモデルデザイナー、背景モデルデザイナー等の名称が、一般的にゲーム会社が人材採用募集をする際に、使用されている。

・アートデザイナーは、世界観やキャラクタの方向性の設定（イメージボード制作）から、実際の登場人物やモンスター等のキャラクタの具体的なデッサン画作成までを担当する。外部の作家に依頼することもある。

・モデルデザイナーは、2D で描かれたデザイン画をもとに、人物や、モンスター、武器、背景等の立体化を担当する。モデルの出来が、モーションの自然な表現に影響する。

テクスチャデザイナーは、モデルに質感を付ける。最終的なビジュアルの品質を大きく左右する。業界全体においては、モデルデザイナーがテクスチャまで担当することが一般的のようである。しかし、ディスク容量の大きなゲーム機の登場の結果、映像の強化が進み、特にテクスチャ工程の増強が図られてきた。しかし、ゲーム機の更なる性能向上への対応は、マンパワーによる解決からシステムを活用した解決が重要となる。

・モーションデザイナーは、ダイナミックなアクションから、指先の微妙な表現まで豊かで、繊

細な動きを実現し、キャラクタに息を吹き込む。ゲーム機の進化の中で、モーシヨンの技術も向上した。例えば、映画制作にて活用されているモーシヨンキャプチャ技術の導入が顕著な例である。また、最近では、フェイス〔顔〕の仔細な動きを表現する特殊な技術も求められている。

- ・エフェクトデザイナーは、炎、雨、爆発、波打ち際など、多彩な現象を視覚的に表現し、それらに付随する特殊効果を制作する。実際に起こる表現を超えて、作品に魅力を加える表現効果を生み出すことも求められる。

- ・メニューデザイナーは、ゲームを快適にプレイするユーザとのインターフェースとなるフォント、アイコンのデザイン、メニュー画面のデザイン等を担当する

- ・カットシーンデザイナーは、ゲーム全体の流れの制作を担当する。シナリオから絵コンテやアニメティクスを作成し、各デザイン工程で制作されたものを載せていく（コンポジット、編集）。さらにライティングを施し全体の演出を施す。実際には、カットシーンに関わる業務は、デザイナーだけではなく、プランナも絡むことが多い。

また、ゲームの中にインタラクティブな部分と非インタラクティブ〔ムービー部分とも言われる〕がある。通常、主にインタラクティブな部分で構成されているが、非インタラクティブが物語の進行上の『つかみ』や説明部分あるいはフィナーレを飾るものとして多用されている。この部分は、インタラクティブ性がない分、より高い映像表現が可能である。映像制作の専門家をビジュアルデザイナーとして異なるチームを形成する場合もある。映像の専門部隊ということでは、外部の専門会社に委託することも多い。但し、ゲーム機の性能向上の流れの中で、インタラクティブな部分と非インタラクティブな部分の表現レベルが近づきシームレスに遊べるゲーム作りが基本となっており、インタラクティブな部分と非インタラクティブな部分が一体となって制作されることで、制作フロー、体制に変化が出てきている。

2. スキルの分析およびレベルの定義について

次章においてプログラマとグラフィックデザイナーのスキルについて、スキル表を参考に詳細を説明する。スキル表の構成は、必要とされるスキルを能力、技術、知識の3つに大別し、それぞれの職種において不可欠で重要なスキルを必須項目（スキル表において”◎”と表記）とし、習得することが望ましいスキルを必要項目（スキル表において”○”と表記）として、レベルを分けて表示した。これは各職種における重要なスキルを分かりやすく浮き彫りにするために、ある程度割り切った分け方をしている。

能力は成し遂げるために必要な力を、技術は成し遂げるために必要な技術あるいはノウハウを、知識は成し遂げるために必要な知識、行動を示している。知識については、各職種に特に必要なものと職種に関わらず開発者一般に必要とされるものとに分けている。

また、各スキルについて、その習得レベルを上長の指示に従い業務を進めるクラスをジュニアクラス、任された範囲については、自分で判断しながら完成させられるレベルをレギュラクラス、さらに他の担当者への指示を出したり、業務遂行のお手本として動ける人材をシニアクラスと呼ぶ。通常、ジュニアクラスからレギュラクラス、シニアクラスにレベルアップしていく過程では、必要とされる総ての項目が均等にアップすることは容易ではなく、実際には習得度合に差が生じ

る。

3. プログラマのスキル

3-1. スキル表各項目の説明

(1) 能力

プログラム能力に関するスキル

① コンピュータ言語

プログラマにとってのコンピュータ言語は必須のスキルである。ゲームプログラムは初期にはアセンブリ言語などの低級言語で作成するのが普通であったが、近年ゲーム機の高性能化に伴い、ほとんど C 言語、C++言語などの高級言語で作成されている。しかしながら、プログラムチューニングやサブプロセッサプログラム開発におけるアセンブリ言語、携帯アプリケーション開発における JAVA 言語、開発環境を構成するスクリプト作成など、それ以外の様々な言語も状況に応じて使用されているため、C、C++に加えて複数の言語に関してスキルを持つことが望ましい。

また、言語がオブジェクト指向言語へとシフトしていくなかで、ゲームプログラムの設計段階からオブジェクト指向の考え方を取り入れるようになった。そのため、オブジェクト指向分析・設計の考え方を理解し、UML などを用いてそれを表現できるスキルが必要になる。

② アルゴリズム

アルゴリズムとは問題の解き方であり、コンピュータに対しアルゴリズムを示すのがプログラムである。既に過去のプログラム中で多くのアルゴリズムが示されてきているため、プログラマはプログラムを作成する際、まったく新しいアルゴリズムを開発するよりも過去のアルゴリズムを応用するほうが多い。また、新規アルゴリズムを必要とする場面においても、過去のアルゴリズムに関する知識を持つことで発想がより容易になる。そのため、アルゴリズムに付いてのスキルを持つことはプログラマにとって大きな強みとなる。

③ Windows プログラミング

Windows が大きなシェアを占める現在では、プログラマにとっても Windows プログラミングスキルの必要性が高くなっている。例えば Windows 環境で動作するターゲット（PC、ゲーム機）の場合そのプログラムスキルは必須である。また、開発環境が Windows 環境であれば、開発ツール作成といった開発環境構築にあたってスキルが必要になる。

Windows プログラムでは、DirectX をはじめとする WindowsAPI の利用が必要だが、低水準な API であるため、MFC を利用することも多い。Windows プログラミングスキルはこれらに関する知識と応用である。

④ ネットワークプログラミング

PC については勿論、家庭用ゲーム機にもネットワーク対応ゲームが増えている。当然開発者にもネットワークプログラムのスキルが要求される。またほとんどの場合、開発環境はネットワークを利用しているので環境構築・ツール作成の面でもネットワークプログラミングスキルは必要になっている。

サーバ・クライアントの機能やプロトコルに関する知識、インターネット・イントラネットの基礎技術などに関しての幅広い知識が含まれる。

⑤ 開発環境に関する知識と技能

開発環境はターゲットによって様々だが、いずれも基本機能は共通している部分が多い。また家庭用ゲーム機や PC で共通の統合開発環境を提供しているケースもあり、プログラミングするにあたってひとつの開発環境に精通すれば他の環境でも応用が利く。

開発環境を構成するのは、作成（エディタ）・実行形式変換（コンパイラ）・実行モニタ（デバッガ）の三機能で、統合開発環境では、これらを GUI 上でリンクさせて様々な機能を提供している。いずれの機能もプログラミングに重要な機能で、スキルを身につけることが生産性の向上につながる。

⑥ 各種フォーマットの知識

プログラムがデータを処理するものである以上、データのフォーマットに関する知識もプログラマにとって重要なスキルである。特にデータの一次加工をする開発ツール作成など、開発環境プログラミングにおいては、フォーマットに関する知識が必須のスキルである。

また、新たなフォーマットの策定を行う場合でも、既に存在するフォーマットに関する知識があれば、容易により良いフォーマットをデザインすることが出来る。

⑦ その他プログラミング手法

上記以外にも様々なプログラミングに関するスキルが存在する。

a. テスト駆動式開発（TDD）

小モジュールに対し最初にテスト・実装を繰り返し、テストにパスしてからソースの整理（リファクタリング）を行う開発手法。結合前に細かくテストを行うことで一つ一つの機能を確認でき、結合後の修正作業を減らし効率化が出来る。

b. セキュアプログラミング知識

セキュリティホールを埋め込まない安全なプログラムを作成するための知識。どのようなコードが悪用されるのか、安全性の高いコードを作成するノウハウなど。ネットワーク対応などのオープンな環境で使用されるゲームソフトが増えていく中で重要なスキルになりつつある。

c. Gnu-binutil の理解

オブジェクトファイルを扱う為のプログラミングツールで、アセンブル・リンク・オブジェクトファイルコピー・ダンプ・シンボル情報の操作など、汎用性が高い非常に強力なツール群。Linux 環境での開発では必須スキル。

d. デバッグ手法

プログラム内に埋め込まれたバグを発見し特定するスキル。ウォッチ・ブレーク・トレースなどデバッガの機能に対する理解とその応用に関する知識、過去のバグフィックス経験などによって構成される。

e. C/C++ABI の理解

コンパイラ・リンカの動作・機能に対する知識と理解。あるプログラムが、コンパイルしてもエラーが出力されないのにもかかわらず、期待した動作をしないことがある。そういったケースでは、プログラマがコンパイラ・リンカの動作・機能に関して正しく理解していない。また、こういったケースでは何が間違っていて動作しないのかが容易に発見できないことが多い。

f. インストラクションセットの理解

ターゲット CPU の持つ命令セットを理解することはそのターゲットの動作を理解する上でも重要である。またコンパイラの動作によって引き起こされるバグ（上記 e. コンパイラに関する知識不足から発生するものを含む）など、C 言語のソースレベルでは発見できないバグを特定する為にも必要になるので、インストラクションセットに関する知識も有用なスキルである。

g. データ圧縮

データ容量の増加が、ターゲットハードウェアのメモリ増加を上回る速度で進むゲームプログラムにおいて、メモリを有効利用する上で、またデータ転送速度を上げる上でもデータ圧縮技術が大きなポイントとなってきた。とくに高速な伸張が可能な効率の良い圧縮技術が求められている。必ずしもプログラマ全員が身につける必要はないが、ゲーム基礎技術として必須のスキルである。

(2) 技術

プログラマとして求められる技術知識のうち、主に専門的な知識の部分の項目について以下に述べる。

プログラマの作業は細分化が進んでおり、以下の項目すべてを網羅することは必要ではなく、担当する部分の項目（たとえばグラフィックプログラムであれば、グラフィックプログラムの項目）が必要になる。また、必要とされている項目内の小項目においても、すべてが必要という訳ではない。担当するゲームの内容によって必要とされる技術は様々であるため、この小項目ではショウケース的に技術を提示する方針とした。スキル表を見るときは小項目一つ一つを見るのではなく、関連する小項目を含む中項目全体でのスキルレベルの把握を前提としている。

① マルチコアプロセッサ

近年、PC 用のプロセッサ、家庭用ゲーム機のプロセッサにおいて、マルチコア化やハイパース

レッドィング対応が進んできている。マルチコアプロセッサで性能を出すためには、旧来の単一スレッドで記述していたプログラミングスタイルとは違う考え方が必要になる。プログラム面から見た場合、ほとんどはスレッドと呼ばれるテクニックを使って平行に動作するプログラムを記述する。マルチスレッドでは、同時に複数の仕事を実行するため、メモリなど共有リソースの排他制御や、同期処理、スレッド間通信など、旧来のシングルスレッドでは必要ではなかったテクニックが必須となる。

② 物理シミュレーション

主に3Dゲームにおいて、画面内のオブジェクトをより現実世界に近い挙動にさせるために、物理現象をシミュレーションする手法である。

工学分野でシミュレータとして研究されていた技術を転用していることが多いが、ゲーム特有の条件としてリアルタイム性を出すため、正確さより計算にかかる時間を少なくすることを優先する必要がある、ゲーム業界で特有の手法も出てきている。

シミュレーションの対象によって、以下のような分類が可能である。

a. 剛体運動シミュレーション

力学に乗っ取って、物体（剛体）同士の衝突や接触を取り扱う。一般的に物理シミュレーションといった場合はこれを指すことが多い。大学レベルの数学・力学の知識を必要とする。車の挙動等に用いられることが多かったが、近年ではゲームの中のオブジェクト全てに対して物理シミュレーションを行い、よりリアルな世界を再現しようとするゲームも多くなってきている。

b. 流体運動シミュレーション

液体や気体などをシミュレーションする。流体力学を基礎とし、一般に剛体シミュレーションよりも高い負荷を必要とする。

c. 弾性体（毛髪・布）シミュレーション

服や毛髪などをシミュレートする。方法としては、バネをメッシュ状に接続したものとして近似的に扱い、弾性体シミュレーションと呼ぶ。

d. 燃焼シミュレーション

炎や爆発時の煙などをシミュレートする。前述の流体運動シミュレーションで行う場合もあるが、より近似的な方法で行われることも多い。

e. 人体運動シミュレーション

人間の体の関節等をシミュレートする。手の位置から肘の関節の曲がり方を求める等、末端の位置から関節角度を決めるのはインバースキネマティクスとよばれ、様々な解法がある。最近では、攻撃や爆発で吹っ飛ばされた人体の挙動をゴム人形と近似して行うラグドールと呼ばれる手法も広く使われている。

③ エフェクト技術

ゲーム中で表示される火花や煙、破片等、映画で言うところの特殊効果に当たるものをエフェクトと呼び、専用のプログラマが割り当てられることが多い。また、2D のゲームにおいても、ゲームの見た目をよくするために、エフェクト技術が使用される。

代表的な処理に以下のようなものがある。

a. 高速パーティクル処理

火花や煙、破片等、小さな物体をパーティクルと呼び、その挙動を作成するには数が多くなりすぎて上述の物理シミュレーション法を使うと処理が重くなりすぎる。そこで、パーティクルの処理では、多少めり込むなど物理的におかしくても高速な処理が行われることが多い。どう処理を省き、それらしく見える物にするかということが必要になる。

④ AI シミュレーション

ゲームにおけるキャラクタの挙動について、それを制御する技術を AI シミュレーションと呼ぶ。たとえば、戦闘ゲームやスポーツゲームにおける、敵キャラクタや仲間キャラクタを、プレイヤーの動きに合わせてどう動かすか、などである。ゲームのおもしろさが、この AI に依っている割合は非常に大きい。AI に必要な処理としては、まず、経路探索・空間分割のような状況把握アルゴリズムが基本にあり、それを用いて、プレイヤーを攻撃したり、援護したりする行動を取るようになる。

主として NPC (Non Player Character) を制御することが多いが、それ以外の地形やゲームの流れ全体を制御することもある。

このカテゴリに必要な知識として大まかに以下のような物が考えられる。

a. 経路探索アルゴリズム

AI 処理に必要な基本的な処理のひとつ。ステージ上のある点からある点へどういう経路を取れば移動できるかを調べる。A*など、代表的なアルゴリズムがいくつかあるが、それぞれのゲームの性質によって使いわけたり改良したりすることが一般的である。

b. 空間分割アルゴリズム

ステージ上の空間を、小部屋や通路の連続として分解し、プレイヤーや他の NPC との関係を理解する処理を出来るだけ簡単にするためのアルゴリズム。BSP 法が主に用いられるが、これもゲームの性質によって、様々な方法が用いられる。

c. 戦闘アルゴリズム

プレイヤーをどう攻撃するかアルゴリズム。ゲームの基本部分のひとつである。

d. 集団アルゴリズム

敵や味方が複数出てくる場合、どう集団として協調するかというアルゴリズム。

e. NPC 制御アルゴリズム

プレイヤー以外の敵や味方などをどう制御するかというアルゴリズム。オンラインゲーム等ではいかに違和感なく人間の操作するプレイヤーに似せるかどうかも必要になってくる。

⑤ ユーザインターフェース

昨今、パッドなどの入力デバイスやディスプレイなどの出力デバイスに手を入れることで、ゲームのおもしろさを拡張する動きが出てきている。家庭用ゲーム機においても例外ではなく、今後の技術の動向には気を配らなくてはならない。

簡単に例を挙げると、以下のような物がある。

a. 多機能コントローラ

今までの方向キー＋複数のボタンだけではなく、たとえば感圧センサーや傾きセンサー、ポインティングデバイス、タッチパネル等、様々な物が入力デバイスとして採用されている。入力だけではなく、コントローラ自体が振動したり、スピーカーがついたりと同時に出力機能が実装される場合も多くなってきた。これらに関してはそれぞれのセンサー等の理解と、各ゲーム機メーカーが出してくるライブラリの実装の仕方の理解が必要になる。また、各ゲーム機に標準的に USB のコネクタがつくようになってきたため、ゲーム個別にデバイスを作ることも多々あり、その USB のドライバをプログラマ側で作成することもある。この場合は USB の仕様の理解も必要になってくる。

b. 画像認識技術

USB 等を経由してゲーム機側にカメラからの映像を取り込み、それを使うゲームも出てきている。たとえば撮った人間の動きを検出してゲームの入力として扱ったり、カードを撮ってどの種類のカードなのかを認識する技術等がある。携帯など別の業界では、顔認識で誰かなのかを判定する技術や、表情認識等の技術もあり、これらがゲームに応用される可能性も大いにあるだろう。

c. 光学技術（立体映像・ドーム映像）

主に業務用の技術として表示系を一般の TV/ディスプレイに限らず、大きなドーム型に投影したり、立体映像として出力したりするゲームも出てきている。ハードウェアだけではなく、ソフトウェアについても、それぞれについて特性を理解し、それに合わせた形で画像を構成する技術が必要になる。

d. マルチチャンネルサラウンド

DVD や HDTV の普及に伴い、家庭にも 5.1ch や 7.1ch のアンプが普及し始めている。ゲーム機にもそれに対応した出力機能がつくようになってきており、徐々に対応することが当たり前の状態になりつつある。

e. 音声認識・合成

携帯型ゲーム機において顕著であるが、音声認識を使ったゲームが徐々に増えてきている。また、料理レシピ等実用ソフトも近年ゲームとして出てきており、そこで大量のデータをゲーム内で音声としてしゃべらせるための音声合成技術も必要となってきた。この分野では家電系メーカーが先行しており、そこからミドルウェアとして供給されることも多い。

⑥ システムプログラム

ファイル管理やメモリ管理、実行単位の管理等を総称してシステムプログラムと呼ぶ。ゲーム機メーカ提供のライブラリをよりそのタイトルに使いやすくするための処理を作成することが多い。また、デバッグのための機構なども含み、この部分の作りの善し悪しで開発効率も変わる、ハードウェアの深い知識と、高いノウハウが必要になる。

また、新しいデバイス等に対応するのもこの領域で行われることが多い。その場合は USB などのインターフェースの知識が必要になる。

あるいは、ゲームの流れの部分の処理や、AI 処理などを、C、C++などのプログラム言語ではなく、スクリプト言語で記述することも近年多くなってきている。この場合、それを解釈する部分をやはりシステムプログラマが作成することになる。

a. DVD,BD 等ディスクメディア操作

家庭用ゲーム機では、その媒体に DVD、BD といったディスクメディアが使用される。PC などによく使われる HDD などとは違い、シークが遅く、リードエラー処理が必要になるなど、気をつけなければならない点が多い。シークを出来るだけ少なくし、高速にファイルを読み込むために、読み込み順にファイルを並べて 1 ファイルにパッキング化したり、圧縮したりするなどのテクニックが必要になる。

また、近年では BGM などはずべてゲーム中にディスクから逐次少しずつデータを読み込んでいくストリーミング再生を行うことが多くなってきており、さらにそれと平行して同時にデータを読み込む処理を行うマルチストリーム処理などが必要になってきている。

b. USB 等デバイス操作

家庭用ゲーム機では、ユーザのデータが保存されるメモリーカード等の外部保存デバイスの扱いに関して、ゲーム機メーカのチェックが厳しく、毎回気をつけなければならないポイントになる。このあたりは各社のノウハウとしての蓄積が必要になる。また、それ以外でも、USB を使った独自デバイスを作成することもあり、その場合は USB 規格の知識が必要になる。

c. メモリ管理技術

通常メモリ管理関数はメーカ製ライブラリとして提供されるが、速度面やデバッグ効率の面で問題があり、独自にメモリ管理関数を書くことも多い。確保するメモリの粒度によって管理方法を変え、速度を改善したり、どのオブジェクトが確保したメモリなのかを記録するようしたりしてデバッグ効率を上げることがある。

d. 組み込み用スクリプト言語

ゲームが大型化するにつれ、ゲームの流れや敵などの配置処理の部分をインタプリタ型のスクリプト言語で記述することも多く行われるようになってきた。トライアンドエラーのサイクルを短くし、デバッグの手間を少なくすることが出来る。独自の言語を作成する場合もあれば、既存のオープンソースな言語を用いることもある。後者の場合は、lua や lisp、Javascript 等が考えられる。

⑦ グラフィックプログラム

ゲームの見栄えの部分司る、非常に重要なカテゴリである。近年、ハードウェアの進歩に伴って、シェーダと呼ばれるグラフィックスプロセッサ用のプログラミング言語が多く用いられるようになってきた。表現できることに自由度が増え、いろいろなことが出来るようになった反面、速度面での見積もりが難しくなっており、リアルタイムが必須とされるゲームのプログラムでは見栄えと速度のバランスがもっとも大きな問題になってきている。

a. シェーダを用いた CG 表現手法

頂点シェーダ(vertex shader)、ピクセルシェーダ(pixel shader。フラグメントシェーダ fragment shader と呼ばれる)を使って、頂点ごと、ピクセルごとに演算処理を行い、最終的にそのピクセルの色を設定する手法が一般的になってきている。自由度が高い反面、いろいろ処理を入れるとすぐに処理速度がかかるようになってしまう。すでにいろいろな CG 手法が開発されてきており、それらを参考に独自のシェーダを作成していくことが多い。

b. HLSL,Cg 等でのシェーダプログラミング

シェーダをより高レベルな言語で書けるように、C ライクな構文の、HLSL(High Level Shader Language) ,Cg といった言語が登場し、現在ではほとんどこれらを使って開発している。

c. グラフィック描画パスの構築

最近の CG 手法では、単純に順番にモデルを描画するだけではなく、影を書くために光源方向から見たモデルを別に描画したり、高速化のために手前から書いたり、半透明のために奥から書いたり、それぞれ違う処理を同じモデル群に対して順番に行って行く必要がある。これらひとつひとつの処理を描画パスと呼ぶ。描画プログラム中で、このパスをどういう順番でどう実行していくかが、問題になることが多い。

d. ターゲットゲーム機の処理特性の理解

家庭用ゲーム機ではそのグラフィックチップによって、全く特性が異なり、あるゲーム機で速度を出すための手法が、違うゲーム機では全く役に立たないこともよくある。それぞれのゲーム機の特性をよく知らないとは最適化は出来ない。

e. オブジェクトカリング手法

出来るだけ高速に描画するために、少しでも描くオブジェクトの量を減らす必要がある。その減らす処理をカリングと呼ぶ。基本的なのは、カメラから見えない物は表示しないという視錐体カリングだが、それ以外にも Z バッファを使う場合の Z カリングや、その場所から見えない物をカリングする PVS カリングなど、様々なカリング手法がある。

f. シーングラフ手法

各オブジェクト間を管理するデータ構造をシーングラフと呼び、それをどう管理するかということが重要になってくる。いくつかの代表的な構造を元に各タイトルに合わせてカスタマイズしていくことが多い。前述したグラフィックの描画パスやカリング手法とも絡み、どう効率よく描画、カリングできるかがこのデータ構造に関わってくる。

g. アニメーション (モーション) 処理

物体をデザインツールで指定したとおりに動かすことをアニメーション（モーション）処理と呼び、主に人体の場合に、各関節の角度を時間軸で変更していく処理として実装される。単純にデータを再生するだけではなく、よりなめらかに見せるために 2 つ以上のモーションを補完したり、上半身と下半身など、一部分だけ別のモーションを再生したりといろいろな処理が行われる。また、目的の位置に末端を持って行くための関節角度を計算する IK（Inverse Kinematics）等と組み合わせられることも多い。

⑧ オンラインゲームクライアント技術

オンラインゲームのクライアントにおいては、スタンドアローンのゲームとは違ったノウハウが必要になってくる。本当にそのサービスの会員であることを認証する必要があること、ネットワークのレイテンシ問題、WAN につないでいることによるチートやアタック対策などである。

a. ネットワークのレイテンシ、スループット制御

ネットワークの伝送速度は、現実には 10～100ms のオーダーであり、これは、60 フレームでのゲームでは数フレーム程度の遅れを意味する。これは相手の入力とそれに伴う挙動がそれだけ遅く伝わるということであり、完全に同期したゲームは作れないということである。主に操作情報そのものを送るのではなく、各オブジェクトの位置や動きを送ったり、操作によって起こる挙動イベントを別に送ったりする手法が用いられる。また、一度に送れるデータ量に関しても制限があり、それをどううまく制御するかがノウハウになる。

b. 認証技術

ゲームのユーザ ID の管理として、サーバ側と協調して行うことが多い。あるいは、サーバがなりすまされていないかどうかを確認するためにも必要となる。

c. チートアタック制御

クライアントのプログラムやデータそのものを書き換えられたりすると、それ以外の一般のユーザに対して圧倒的な優位でゲームを進めることができてしまい、ゲームの構造そのものになりたたなくなることもある。また、サーバやクライアントに対して DoS などのアタックがかけられることも多々あるので、防御機構は必須になる。

⑨ モバイルデバイスプログラム

近年の携帯電話のゲーム環境の拡大は著しく、一昔前の家庭用ゲーム機並みの画面が出るようになってきている。開発環境は Java や Brew といった独特の環境が使われ、また端末ごとの性能差をどう埋めていくかなど、家庭用ゲーム機とは全く違ったノウハウが必要になる。

a. Java, Brew プログラミング

携帯電話では Java が主に使われ、一部で Brew が使われている。いずれにしても使用できるメモリ量が限られているため、通常のプログラムと異なり、独特の工夫が求められる。

b. 各種クラスライブラリの知識

言語は同じ Java であるが、各キャリアから微妙に異なるクラスライブラリが提供され、そ

の対応が必要になる。ほとんどの場合、主なキャリアすべてに対応する必要が出てくるため移植性に気を遣ったプログラムを書かなければならない。

⑩ サーバプログラム

オンラインゲームのサーバプログラムでは、ゲーム本体のプログラミングと全く違うノウハウが必要になる。ゲームのタイプにより若干は異なるが、ある程度は Web サイトやショッピングサイト等に共通に必要な知識がある。以下にその主な領域をあげる。

a. ネットワークハードウェアの理解

サーバ側でどのようなハードウェア構成を取るかということ、まず一番重要な検討点となることが多く、そのあたりのハードウェア知識は必須となる。主には負荷分散のために冗長な構成を取り、それをコントロールする機器が必要となる。

b. スケーラブルなゲームサーバ、Web サーバ等の理解

オンラインゲームのサーバは、ゲームサーバ複数と Web サーバ等で構成されることが多い。耐障害性や、ユーザの増加に対応するために、簡単に数を変更できる冗長構成をとらなければならない。

c. サーバネットワークプログラミング

ゲームサーバは、C や Java, Perl 等で構築され、TCP や UDP を使ったネットワークアプリケーションとして構築される。当然、多人数をさばくためにマルチスレッドやマルチプロセスといった手法を使い、出来るだけ軽く多数のデータをさばいていかなければならない。

d. データベースプログラミング、(SQL 言語)

オンラインゲームの肝となる部分はユーザごとのデータを扱い、それを検索・更新する部分である。これには一般的に RDB (リレーショナルデータベース) が使われ、SQL 言語によって操作される。データベースの操作には特有のノウハウが必要になる。

e. Perl、PHP 等、CGI 言語プログラミング

ゲームサーバ以外にも、Web 系の知識は必要になる。ランキング情報やログイン処理、アカウント管理などは、Web サーバが使われることが多く、それは CGI 処理として実装される。一般的な CGI 言語の知識が必要とされる。

⑪ サウンドプログラミング

サウンドプログラミングは、ゲームプログラムとは別のチームとして扱われることが多い。作曲者などのサウンドアーティストと同じ部署で、どのゲームでも共通して使われるように作られることが普通である。BGM には昔は Midi シーケンス等が用いられていたが、近年ではディスクからストリーミング処理でならすことが主流になってきている。また、5.1ch などのマルチチャンネルサラウンドも普及し始めているため、その対応も必要になることがある。

a. 音響工学

近年 CPU の処理能力が上がってきたため、専用のサウンドハードウェアを使わず、すべて CPU で波形処理することが多くなってきている。リバーブやピッチ変換といった基本的な処理も、CPU で行うため、自由度が上がっている反面、作業量が増えている。

b. 3D 音響モデリング

3D ゲームで、さらに臨場感を持った音を出すため、単純にパンとボリュームだけではなく、壁の反射や材質等も考慮した音響モデリングを行うことも始まってきている。マルチチャンネルサラウンドと組み合わせて、よりリアルな音を目指す。

c. 音声圧縮フォーマット

BGM や音声をディスクからストリームできるようになったが、無圧縮の音声だと大きすぎるので、何らかの圧縮をかけることが多い。ADPCM 等が多いが、近年では MP3 や Ogg Vorbis、ATRAC のような処理は重いが高圧縮の圧縮形式も用いられるようになってきた。それぞれの特性を理解して使うことが重要である。

⑫ 開発環境制作

直接ゲームのプログラムには含まれないが、ゲームを開発する環境としてのツール群を作る必要はある。特に大きなチームになると日々のルーチンワークをいかに効率よくこなす環境を構築するかで、チーム全体の作業効率向上に大きく寄与することが出来る。主としてデザイナーが使うモデル・アニメーションなどのプレビューツールや、ゲームデザイナーが使うイベント設定ツールなどが作られることが多い。そのほか、3D ツールが出力するデータを実機用のデータにコンバートするツールや、膨大な作成データを管理するためのツール、開発チームの情報共有用の Web ページ、掲示板の制作など、その業務は多岐にわたる。そのため、その担当者も広い知識が求められることが多い。

a. Windows プログラミング

デザイナーやゲームデザイナーが使うツールは主として Windows アプリケーションとして作成されることが多く、Windows 上でのプログラムノウハウが必要となる。

b. CGI プログラミング

昨今では、チームの情報共有に Web がよく使われ、簡単な CGI を使って掲示板等を作ることも多い。

c. PHP,Perl 等を使った Web アプリケーション

CGI プログラミングを一步進めて、たとえばデータ管理や進捗管理などで、PHP,Perl を使った本格的な Web アプリケーションを作ることもある。

d. Perl、Ruby 等 LightWeight 言語でのツール制作

テキストコンバータや一連の作業の自動化などで、perl や ruby といった言語を使って簡単なツールを作ることは多い。一連のルーチンワークをこういう言語を使って簡単にツール化すれば、かなりの時間の短縮が期待できる。

e. Excel 等の Office アプリケーション

ゲーム内の各種パラメータや配置データなど、数値情報は Excel を使って管理されるケースも多く見られる。マクロ等を駆使して、ひとつのアプリケーションのようなデータシートを仕上げているケースもある。

f. 外部のツール、フリーソフトウェアの知識、情報

開発環境制作においては、外販するわけではないので、外部のツールやフリーソフト等を使ってより効率よく優れた物を作ることを求められる。ネットから幅広く情報を集めて、それを生かすことが求められる。

g. 一般的な画像形式、データフォーマットの理解

テクスチャやモデルデータ等、一般的な形式でオーサリングツールから出力されたデータを、独自形式のデータに変換することは良く行われる。よく使われるフォーマットについては、どうすればプログラムに取り込めるのかを知っておく必要がある。

h. XML などによるデータ交換手法

昨今では、XML によってデータが出力されることが多くなっている。独自フォーマットもバイナリ形式でなく、XML 等を使うことで、相互運用性や、未来に渡って再利用しやすくなることが期待できる。

i. XSI, Maya, 3D Studio Max 等の 3D ツール知識

デザイナーがモデルやアニメーションを作る環境である、3D ツールに関する知識は必要である。ツールによっては、スクリプト等を使って一連の作業を自動化できる物もあり、これらスクリプトをうまく整備することで開発効率を上げることが出来る。

(3) 知識

プログラムに関連する知識に関するスキル

① OS・基本ソフトウェアの学習

プログラミングが PC 上で行われ、入力・コンパイル・デバッグをアプリケーションによって行う以上、プログラマにとって OS・基本ソフトウェアに関する知識は前提である。

一般に開発環境には Windows 環境と Unix/Linux 環境があり、ターゲットに応じて、また処理内容に応じて使い分けるので、それぞれについて最低限の知識が必要とされる。

② コンピュータグラフィックス

プログラマが直接グラフィックを作成することはないが、グラフィックデザイナーの作成した素材をターゲット上で処理するために、グラフィックに関するスキルが必要とされる。ハードウェアの進化とともに必要とされるレベルが上がり続けていて、単純なフレームバッファの仕組みから質感などの高度な 3D 表現を可能にする為の処理まで様々な段階がある。シェーダプログラムなどを作成する為には、当然高レベルのスキルが必要とされる。

③ ゲーム物理

ゲーム内で物理現象をシミュレートしたり、より高度な画像表現を実現する為に、数学や物理に関する知識が必要とされる。かつては単純なものしか扱えなかったが、ゲーム機の高性能化によって、本格的なシミュレートを行うことが可能になるなどスキルの幅がどんどん広がっている。また、ゲームでは良く採られる簡易モデル化（処理的に正確なシミュレートが不可能なため、近い結果が得られるような簡素化したモデルを考案する）といった場合でも、当然対象に対する知識が必要とされる。

④ ゲームハードウェア

ハードウェアに関する知識は、担当業務に応じて必要なレベルが異なる。内部処理プログラムを作成する為には最低レベルでも可能であるが、デバイスプログラムとなれば当然高いレベルが要求される。理想はデバイスプログラムやライブラリレベルでハードウェアの相違を完全に吸収してしまうことである。デバイス・ライブラリ部分を含め、ゲームプログラムにおいてはしばしばハードウェアの特性を考慮したプログラムが要求されるので、重要性の高いスキルのひとつといえる。

⑤ データベースプログラミング

データベースに関するスキルは特にサーバプログラマに必要とされる。様々なデータを集積し、時々刻々移りかわる状態を反映し外部の要求に応じて送信するデータベースの処理はゲーム上でも必要になる。多人数ネットワークゲームのデータなどはその典型といえる。ネットワークに関する知識・情報セキュリティに関する知識と密接に関係しているので、同時のそれらのスキルも身につけているのが望ましい。

⑥ 情報セキュリティ

ネット環境での開発が当たり前になった現在では、環境構築や運用に当ってセキュリティの基礎的な考え方が必要とされる。また、サーバプログラム・ネットワークゲームプログラムなどでは情報セキュリティスキルはより高いレベルが要求される。サーバプログラマには特に必要なスキルである。

⑦ ネットワーク基礎

現在では開発環境がネットワーク上にあるのが前提であるため、ネットワークに関する基礎的な知識は環境構築・運用に必須である。小規模なLANを構成したり、簡単なネットワークエラーは自力で解決できることが望ましい。また、ネットワークに対応したゲームの場合ではさらに重要性は高くなる。

⑧ 基礎知識

プログラマに必要とされる基礎知識としては、代数幾何（ベクトル・行列、2次元・3次元空間）、微分積分、確率統計などの数学、計算論、アルゴリズム、言語、アーキテクチャなどの計算機科学、視覚、聴覚、触覚についての心理・生理学、計測・制御学、があげられる。これらの基礎知識は、個別の開発業務に対応するものではないが、複数のスキル・知識にまたがって必要となる前提知識を含んでいる。これらを一から学ぶことは時間的にも業務への効果を考えても効率的ではないが、各スキルの習得に難しいと感じたときには、これらの基礎知識に関する本などを調べ、解決できることが望ましい。たとえば、スキル表の各スキルに対応する知識を挙げると次のようになる。

アルゴリズム：計算機科学(計算論・アルゴリズム・言語・アーキテクチャ)

ゲーム物理：数学（代数幾何，微分積分，計算幾何学），力学，ロボット工学，制御学

エフェクト技術：数学（微分積分）

AI シミュレーション：数学（確率統計，代数幾何），制御学，計算機科学（計算論・アルゴリズム・言語）

ユーザインターフェース：数学（代数幾何，微分積分），力学，計測・制御学，心理・生理学）

グラフィックプログラム：3次元空間，線形代数，微分積分，計算幾何学，計算機科学

計算機科学はアルゴリズムに対して処理時間の見当をつけたり，高速化の見当をつけたりする際に必要な知識であり，実際のアルゴリズムを開発しながら習得するとよい。

心理・生理学は，心・体についての知識である。たとえば，人間の反応速度や，視力の限界など，プログラムの性能への要求について，見通しを立てるために役立つ。また，ゲーム機に用いられるユーザインターフェースの進化に伴い，人間側の基礎知識として，心理・生理学の重要性が増している。また，計測側の基礎知識としては，各種センサ(加速度・光・音・画像処理など)や入力情報の処理といった用途から計測・制御学の重要性も増している。

3-2. キャリアパスの現状

入社後、開発者各自が自らのキャリアパスを認識し、業務に従事しているかを今回実施の『ゲーム産業における経営戦略と人材マネジメントに関する総合調査（以下アンケート調査結果と称す）』より考査した。(Q10)『将来的なキャリアパスに関する情報』の提示は6.7%と、ほとんどの企業で実施されていない状況にあるように見受けられ、開発者に対するキャリア・パスが作られていない状況にあると思われる。またそれを裏付ける他の要因として、(Q12)『評価基準を開発者に公開している』に対して『どちらともいえない』～『公開していない』の占める割合が（62.1%）半数以上の企業であり、この状況からも開発者に対する評価目標を認識させることが困難な状況であると思われる。

また、上記以外の要素として、ゲーム業界はまだ若い産業であり、社員平均年齢も30代前半で

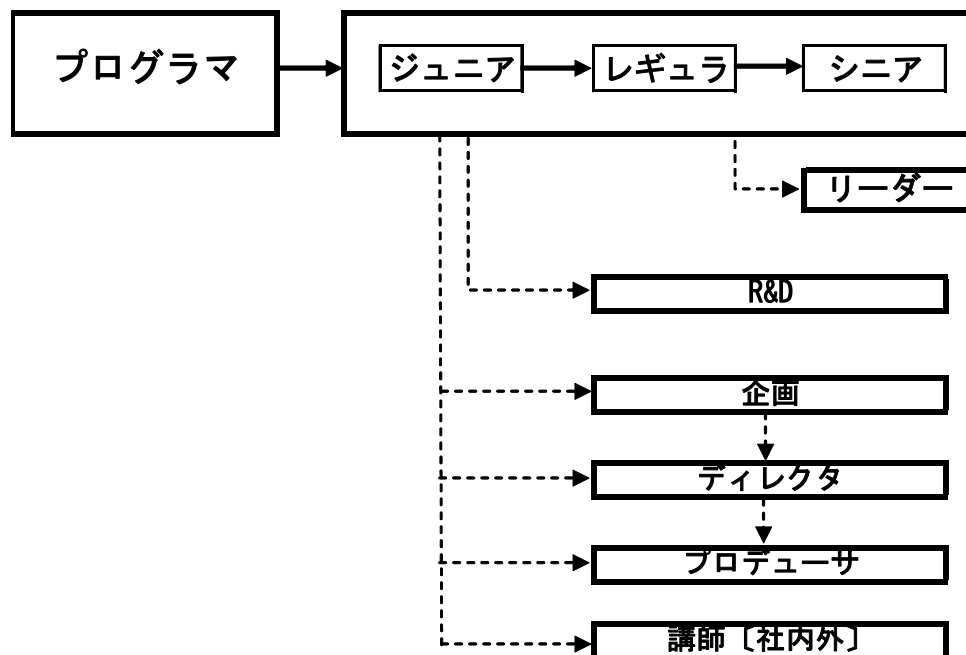
あり、キャリアパスの実例を開発者が見るケースが少なく、中々認識しづらい状況にあると推測される。

ただし、現状のキャリアパスとしては、以下記載の通りのパターンがあると考えられる。

キャリアの積み方として、以下のようなケースが考えられる。

- ・ある特定分野の専門家として、ジュニアクラスからシニアクラスにスキルアップすること（例えば物理シミュレーションの専門家という意味）。
- ・ある特定分野の専門家であってもシニアクラスになるとチーム運営の一端を担うケースもあり、専門分野のセクションのリーダーとしてのマネジメントスキルが期待される
- ・プログラム全般の専門家として、幅広い能力発揮をする。チーム全体のシステムの方向性を決める
- ・他分野におけるプログラマ（サーバプログラマから開発環境制作プログラマ）への転換
- ・プログラマからゲームデザイナー（プランナ）への職種転換し、ゲーム企画をする
- ・チーム全体のディレクタとして、プロジェクトを制作面から引っ張る
- ・プロデューサとして、総合的にチームを仕切る
- ・経験・知識を生かし、プログラム技術研究を行う（例えば、最新ゲーム機向けに共通技術の開発・研究）
- ・経験・知識を生かし、学校の講師等

【ゲームプログラマキャリアパス例】



*各プログラマの職種間の異動も当然キャリアパスとして行われる

*実線は標準的なキャリアパスであり、破線はまれなケースでのキャリアパスである。

3-3. 在職者の教育カリキュラムの現状

アンケート調査結果（Q6）より、現状の育成活動において『OJT』中心（96.6%）の指導がほとんどであり、Off-JT（44.8%）、キャリア開発支援（34.5%）を実施している企業は半数を大きく割る状況であった。しかし今回のアンケートの効果とも考えられるが、今後の育成活動に関しては、Off-JT（79.3%）、キャリア開発支援（75.9%）と8割近くの企業にて『OJT』以外の育成活動も積極的に実施していく考えを表明している。

これはアンケート調査結果（Q5）の開発者（プログラマ）の活用に対する企業の考え方として、『ゲーム開発には、組織に蓄積したノウハウが重要であるため、当該人材を長期間雇用する』との回答企業が圧倒的に主流を占めていることから納得のできる結果であると考えられる。

しかしながら、別のアンケート調査結果（Q10）では、『社内の教育訓練、キャリア開発に関する情報』、『社外の教育訓練、キャリア開発に関する情報』の提供が今後においても全体の5割弱の企業であり、上記育成活動の改善伸び率と比較すると、まだまだ具体的に考えている状況であると言いがたい状況にあると推測する。

社外での『開発者の Off-JT』の手法としては、『CEDEC』（57.1%）が活用状況としては最も高いが、それ以外の国内・海外での研修機関・研究会の活用はほとんどされていない状況であり、なかなか外部研修の活用もうまく出来ていないのではと推測する。

3-4. プログラマを育成するためには

プログラマの仕事は、企画が作成したゲーム仕様をもとに、グラフィックデザイナーが制作した映像素材やサウンドデザイナーが制作した音素材を組み合わせ、ゲーム機上でインタラクティブに動くゲームとしてまとめていくのが仕事であるが、ゲーム仕様を具体的な設計仕様に落とし込み、次に設計仕様にもとづいたプログラミングを行い、最後にデバッグをしてゲームの完成までの一連の業務をこなすのが役目である。そのような業務のために、「3Dシェーダ」、「共有ライブラリ」、「サーバプログラム」、「アプリケーションプログラム」、「ビジュアル作業支援ツール」、「サウンド支援ツール」、「データ作成用ツール」、「データベース」、「開発環境の整備（ネットワーク）」などの関連プログラムの開発も行うため、その仕事内容は非常に多岐にわたっている。

また、企画からあがってきたゲーム仕様に従ってプログラミングするだけでなく、ゲームユーザの立場で遊びやすさや使いやすさを考えてゲーム開発をすることが要求される。そのため、プログラマの資質として、開発業務をスムーズに進めるコミュニケーション能力、ゲームユーザのニーズを捉える感性や柔軟な頭、より良い製品にしていけるプロ意識が求められている。

以上の多岐にわたる仕事を立派に果たすプログラマを育成する方法は、単純な方法ではなく種々の方法の組み合わせが考えられるが、ここでは、2つの視点（個人育成または組織育成、企業内育成または業界内育成）からプログラマを育成する方法を検討した。次表には、検討結果のプログラマ育成マップを示す。

プログラマ育成マップ

	企業内育成	業界内育成
個人育成	① キャリアパスの明確化： 職務についてどのような技術をマスターすればよいのか明示することが重要であり、さらに、上位の仕事内容を見えるようにすることも必要である。 ② 評価育成の導入： 業績連動型賞与や発明考案の実績褒賞制度を導入し、技術者のモチベーションをアップさせることが積極的に行われつつあるが、技術能力の育成として、ゲーム技術の社内論文制度や社内コンペティションを導入し、ゲーム技術者が喜んで自らの技術を開示し、お互いに技術を切磋琢磨する環境作りが考えられる。また、開示された技術を評価し人事制度に反映させるで、モチベーションを高めることも方法のひとつとして考えられる。	① キャリアパスの明確化： 企業の垣根を越えて、同一職種の技術者がどのような能力を有するのかの基準作りを行い、また、資格制度を検討することも考えられる。 ② 評価育成の導入： 従来、ゲームの社会的評価は、作品としての評価であるため、プロデューサやグラフィックデザイナーなどが主な評価対象であり、プログラマが直接評価されることがなかった。ゲーム作品の裏方としてどのような新しい技術やアルゴリズムを考案したのかを競う国内や国際的コンペティションを導入することにより、ゲーム技術者が喜んで自らの技術を開示し、お互いに技術を切磋琢磨する環境作りが図られる。技術者が応募するに当たっては技術開示を条件にすれば、個人の育成ばかりでなく、業界全体の技術者育成にもつながる。
組織育成	① 技術情報の共有と伝承： 従来のゲーム制作工程においては、開発した技術情報の共有、伝承にあまり時間をかける余裕がなかった歴史がある。しかし、ゲーム開発期間の長期化と開発費用のアップを考えると、同じ失敗を繰り返さない効率的な開発体制を築く必要がある。そのためには、ゲーム制作工程の中に業務として、技術データやドキュメントを整理させ、共有と伝承する仕組みの確立が考えられる。	① 技術情報の共有と伝承： CESAのような業界団体において、技術情報の共有と伝承の仕組みを構築する。 ・ ハンドブックや技術便覧を業界と大学研究所で共同整備 ・ 各企業からの技術者と大学研究者による特定プロジェクト（最新シェーダ、ゲームエンジン、物理エンジンの共同開発など）の共同推進。 ・ 若手技術者のための交流の場（手軽にミーティングができ、必要な技術図書の閲覧もできる場）

	② 社内技術者による技術者教育： 日進月歩の発展を遂げる先端ゲーム技術を目指す組織を充実することは重要であるが、それだけでなく、技術を底辺からアップすることにより組織としての技術能力アップすることが重要である。そのためには、OJTだけでなく、先端技術内容を咀嚼できる優秀な技術者による社内技術者に対する教育が有効である。さらに、指導役の技術者に対しては人事上の評価があるなど、モチベーションアップも期待できる。	② 業界、大学研究所関係者による技術者教育： 業界の指導的な技術者や大学研究者による各技術分野の基礎から先端までを開設する定期的な技術シンポジウムの開催 （例：IGDA主催ゲーム開発者セミナー、SIG-GTやACM SIGGRAPH Course、LA SIGGRAPHなど）
--	---	---

4. グラフィックデザイナーのスキル

4-1. スキル表各項目の説明

(1) 能力

グラフィックデザイナーに必要な能力を一言で言えば、画力である。本項目において、画力は、単純に紙に筆記具を使用して描くデッサン能力を指すのではなく、平面上（紙の上）でデザインされたものを3次元化し、息を吹き込み、作品として表現していくすべての能力として定義する。この画力をスキル表において、デッサン、色彩、デザイン、アニメーション、演出に分けて説明している。また、スキル表の観察力、表現力、描写力は以下の定義で用いることにする。

観察力 理解して忠実に再現することができる力

表現力 理解したものを応用表現できる力

描写力 描くことができる能力（ツールの使用も可）

(1-1) 能力項目説明

① デッサン

3DCGベースのゲーム作りにおけるデッサンは、立体（形状、陰影）・空間を把握することと描写する力を指す。

a. 立体

立体観察力

立体の全体の形状に留まらず、立体表面の形状や陰影を把握する力を指す。

立体描写力

アートワークにおいては、立体を平面上で描くことができる能力を指すが、モデリング、テクスチャにおいては、3DCGにて表現できる能力を指す。

構造観察力

立体は、一つの部分で成り立っているものだけではなく、幾つかの部分が繋がって成り立っている。あるいは、一つの部分で成り立っているものでも動きに変化が生じる構造のものもある。これらの部分が繋がっている構造、あるいは、変化が生じる構造つまり、関節やギミックを含んだ立体を把握する能力を指す。

構造描写力

アートワークにおいては、立体を平面上で描くことができる能力を指すが、モデリング、テクスチャにおいては、3DCGにて表現できる能力を指す。

b. 空間

空間観察力

街並や空など、空間を把握する能力に加え、空間を含んだ立体の形状を把握する能力を指す。

空間把握、構合力

基本的には、街並や空など、空間を描く能力を指すが、空間を含んだ立体の形状を描く能力あるいは、巨大な物体を把握し、描く能力を指す。

② 色彩

色彩学、質感（光と色）に関する力を指す。色の属性（色相（色味）、明度（明るさの度合）、彩度（あざやかさの度合））をベースに人間の目で見たものを平面上に表現する力とコンピュータ上で、基本色光三原色（RGB, Red, Green, Blue）の組み合わせにより、多彩な色を表現する力がある。質感は、素材を表現することを指す。例えば、同じ金色の物体でも、金属なのか、プラスチックなのかによって表現が異なる。

a. 色彩

色に対する観察力

色を正しく認識し、把握する能力を指す。

色に対する表現力

アートワークにおいては、平面上で表現できる能力を指すが、モデリング、テクスチャにおいては、3DCG にて表現できる能力を指す。

色に対する構成力

ものを表現する場合には、色々な色が使われている。これらの色を組み合わせ、構成することができる能力を指す。

b. 質感（光と色）： モデルデータの作成において重要である。

質感に対する観察力

立体表面の性質の違いから受ける感じを把握する能力を指す。

質感に対する表現力

3DCG にて表現できる能力を指す。

③ デザイン

ここでのデザインの意味は、図案や模様のレイアウトではなく、広義の意味として日本語の「設計」を指す。ゲームデザインとは、ゲーム全体の設計を指すが、グラフィックデザイナーにおけるデザインとは、画に関わる部分のデザイン（設計）を指す。

a. デザイン

既存のテイストに合わせてデザインできる能力

あるがままを設計（構成）できる能力を指す。

オリジナリティを出せる能力

独自性を出し、設計（良い構成）を工夫できる能力を指す。

④ アニメーション

時系列を伴うもの（動き）に関する力を指す。

a. モーション

生物の動作の観察力

生物の動作を理解し、忠実に再現できる能力を指す。

生物の動作の表現力

生物の動作を応用力を発揮して描写することができる能力を指す。

物理現象の観察力

物理現象を理解し、忠実に再現できる能力を指す。

物理現象の表現力

物理現象を応用力を発揮して描写することができる能力を指す。

自然現象の観察力

自然現象を理解し、忠実に再現できる能力を指す。

自然現象の表現力

自然現象を応用力を発揮して描写することができる能力を指す。

想像上の事象の表現力

魔法などの本来現実には存在しない事象を描写することができる能力を指す。

⑤ 演出

物体や事象を表現するときに、それを効果的に見せる力を指す。

(1-2)職種毎に必須、必要とされる能力の説明

それぞれの職種においても、扱う対象により重点の置かれる能力が異なることを分析し、スキル表は示している。

① アートデザイナー(キャラクタデザイン、背景デザイン、イメージボード制作)

キャラクタや世界観のデザイン、アートワーク作成のため、対象に関わらず、アート共通必須能力として色彩、デザイン能力は不可欠である。また、新しいものを生み出すという意味では、色々な物事に対しての興味知識は欠かせない。ゲーム機の性能が向上したことで、表現できる範囲が広がり、仔細度合いが増していることから、今まで以上に最初のアートデザインの段階で求められる能力が高度化している。

a. キャラクタ

ゲームに登場する一切のキャラクタ（モンスター等を含む）を具体的なデザインに落とし込むため、アート共通必須能力に加え、立体デッサンは必須。また、世の中に無いキャラクタ（モンスター等）の制作においては、生み出されたキャラクタの特性を示すためにも生物の動作への理解があることが望まれる。

b. 背景

ゲーム全体の世界観を具体的なデザインに落とし込むため、アート共通必須能力に加え、立体デッサン、空間認識は必須である。但し、立体構造についてのスキルも身につけているのが望ましい。

c. イメージボード

ゲームの方向性（世界観やキャラクタの持つ基本人物像イメージ作り。コンセプトワーク）のためスキル表のほぼ全項目に関して広く能力を必要とする。不特定多数のシーンや情景を描く事が必要で、その為の観察力や表現力（演出力）が求められる。その際に自然現象や物理現象をベースに想像上の事象なども含めて世界観やシーン、キービジュアルなどを構築していくケースも多々ある。ゲームの方向性を決めるためのきっかけ、イメージ作りということで定義しているため、詳細な動きや色、仕様に落とし込むのはアートデザイナー、エフェクトデザイナーや背景デザイナーなどが行うこととして整理している。

② モデルデザイナ(人、フェイシャル、モンスター、メカ、背景)

モデルを作成する上でまず、デザイン画から立体物としての最終イメージを明確に持つことが大切である。立体物をどの角度から見ても違和感のないようにする必要がある。粘土細工、彫刻などの造形物を作成することと基本同じであり、道具がコンピュータになっただけである。違うのは、ゲーム機の制約があるところである。そのために、基礎能力は非常に重要である。人、モンスターなどは、置物ではないので、ポーズをとらせた状態での形状も非常に重要となる。骨格、筋肉がどのようにになっているかを考え、理解し、再現する必要がある。その意味でも資料の観察は非常に重要である。

初心者が陥りやすいのは、デザイン画に描かれた角度は良く見えるが、他の角度から見た場合におかしい、動かしてみると関節が破綻を起こす、筋肉の流れや骨格がおかしい、細かい所に目がいき過ぎて基本のプロポーションがおかしい、ポリゴンの配分バランスが悪いなどである。

スキル表での整理は、対象によって細かく分けてはいるが、基本的に、立体デッサンに関わる能力、質感に対する能力は必須である。デザイン能力は、アートデザインにおいて基本的に表現されたテイストを3次元に表現できる能力が必須である。色彩については、テクスチャ作業側能力としてここでは切り分けて考える。

a. 人

日常的に見慣れている対象のため、リアル系であれば、ちょっとした違いが非常に違和感を感じさせる。デフォルメ系であれば、そのデフォルメのポイントをよく理解して再現する必要がある。

b. フェイシャル

人からフェイシャルを分けた理由は、ゲーム機の高度化、受像機（テレビ）の解像度の向上により、より仔細な表現が求められるためである。輪郭、目、鼻、口、耳、眉毛の大きさ、位置が少し変わるだけで印象が変わる。特に、目、鼻、輪郭は、実際に顔面整形でも手を加える箇所でもあり、非常に重要である。髪型も重要であり、実際の人でも髪型で印象が変わるのと同様である。

c. モンスター

モンスターの元になる生物の観察が非常に重要である。元になる生物が見つからないデザイン画の場合は、幅広い生物観察力が必要である。実在しないものをあたかも実在しているかのように感じさせる存在感、説得力が必要である。巨大なものは、スケール感を損なわないポリゴン配分が必要である。単にモンスターだけを表現する能力ではなく、例えば、街レベルの大きなモンスターであれば空間部分を含めた表現ができる能力が求められる。

d. メカ

実在するものであれば、資料をよく観察することが重要である。キャラ系のメカであれば、生物のバランスやギミック、綺麗な曲線と直線との融合などの再現が重要である。巨大なものは、スケール感を損なわないポリゴン配分が重要である。単にメカだけを表現する能力ではなく、例えば、街レベルの大きなメカであれば空間部分を含めた表現ができる能力が求められる。

e. 背景

モデル共通能力に加えて、空間が必須である。モーション（『動きのある表現』、や『動きの一部を切り取った表現』など）について、物理現象、自然現象、想像上の事象、いずれの項目もモデリング段階からしっかり詰めることが求められる。例えば、物理現象の観察力については

物が壊れた時の表現において、自然現象の観察力は水もの（波打ち際、滝など）の表現において、さらに、想像上の事象の表現力は異空間の背景において、それぞれ必要となる。また、全体に関わるため、連携が重要であることから、コミュニケーション能力は求められる

③ テクスチャデザイナー(人、有機物、無機物、背景、マットペイント)

デザイナー職の中で、テクスチャデザイナーが、質感能力を最も必要とする。勿論、立体デッサン（立体、構造共に）、色彩も必須である。但し、モデリングされる対象により必須項目を区分けている。また、デザイン能力については、指定されたテイストに合わせる能力が必須である。ものが何によってできているのかを要素分解して理解し、表現する際にはその要素がどのように組み合わせられるのかを考える。そのため、対象を見る観察力は重要となる。表現する際には対象を具体的にイメージ出来る力が必要になる。

a. 人

立体デザイン、色彩・質感における能力について、人間は、実存する（見本がある）事から、観察力の方を重視する。

b. 有機物

立体デザイン、色彩・質感における能力について、有機物（モンスター等を想定）は、実在ではないもの（見本が無い）ことを前提に、描写力を重視する。

c. 無機物

立体デザイン、色彩・質感における能力について、無機物（メカ、武器等を想定）は、実存する（見本がある）事から、観察力の方を重視する。

d. 背景

テクスチャ共通能力に加えて、空間認識が必須である。物理現象に関する能力については、モデルの方が重要であるが、テクスチャでも必要である。

e. マットペイント

テクスチャ共通能力に加えて、町の全景や壮大な風景、巨大な城などが主な対象のため、空間認識が必須である。3D データと区別のつかない見た目の一枚絵を作成する。ニュアンスのオーダーがあるので演出力があることが望ましい。

④ モーションデザイナー（ボディ、フェイシャル、無機物の動き）

モデルに動きを与える作業のため、立体の構造については必須である。アニメーション能力は、対象がボディ（人とモンスターを同じ括りと考える）、フェイシャル、無機物の動きかどうかによって必須項目を分けている。

a. ボディ

デザインの既存のテイストに合わせる能力は必須だが、オリジナリティについては、スキル表の意味合いを分かりやすくするために必須としていない。

b. フェイシャル

立体描写力、構造描写力を考えるとき、ボディとは異なり、フェイシャルの場合は、モデルとモーションは、一体となって制作される部分がある。双方が意識しながら制作することが必要である。

c. 無機物の動き

物理シミュレーションによって動きを表現することが基本だが、立体に関する項目に加えて、ビル等の大規模な対象物が壊れる状態をよりリアルに表現するために空間に関する能力が望まれる。シミュレーションして終わりではなく、ある程度の演出を入れることがあるので演出項目についても重要である。

⑤ エフェクトデザイナー

炎・雨・爆発など（現象）を視覚的に表現し、付随する特殊効果を制作することから、共通能力として、色彩、デザイン、物理現象、想像上の事象が求められる。特殊効果は、キャラクターの動きに関連した表現場面として（バトル）、主に背景に関連した表現場面として（マップ）、非インタラクティブ部分（ムービー）を合わせた3つの場面を主な特殊効果の表現場面として抽出した。

a. バトル

エフェクト共通能力に加え、デザインでオリジナリティが求められる。又、演出も不可欠である。

b. マップ

エフェクト共通能力に加え、空間に関する能力が不可欠である。自然現象についての能力も欠かせない能力である。

c. ムービー

エフェクト共通能力に加え、インタラクティブ部分に比べて、全体的に映像の質の高さ（見た目の美しさが前提）が問われることから、質感に関する能力が不可欠である。

⑥ メニューデザイナー

フォント・アイコンのデザイン、ステータス画面などのレイアウトや仕様、セーブデータのアイコンなどを制作ことから色彩関連の能力が必須である。ユーザインターフェース部分を担当するので、開発チーム以外とのコミュニケーションも求められる。ゲーム上とウェブ上があるが、特に能力を区分する必要はない。

⑦ カットシーンデザイナー

シナリオに書かれているドラマの内容を、一連の映像にするための「演出力」である。

a. 絵コンテ

3D化作業のための「設計図」として、平面上（紙と鉛筆）でドラマの流れを描く。ストーリーを映像化する際の最初の工程なので、演出センスがもっとも重要だが、加えて画面内に起こる主な事象の動きを、絵で的確に伝える能力が大事である。

b. アニマティクス

絵コンテ（場合によってはシナリオ）の内容を3Dツールによって、動きのある映像に再構築する。演出意図を伝えるための最低限のアニメーションスキルは必要だが、映像としてのクオリティは低くても構わない。また、絵コンテはあくまで2Dなので、それを3D化するための演出センスも重要である。

c. 編集

ショット単位で尺の長さやつなぎの順番、そして取捨選択（不要なカットの破棄）を行う。こ

これらの操作で演出の方向性やドラマのテンポがかなり変わる上、映像作業においては演出操作の最終工程にあたるので、演出能力、特に「映像のリズム感」といったものが、他の工程よりも重要である。

d. コンポジット

プリレンダー（ムービー）において、1ショットは大抵複数の素材（レイヤー）に分けて制作される。このレイヤーをひとつに合成する作業がコンポジットである。コンポジットする時に、各素材を馴染ませるために色味や濃淡の調整を行ったり、演出意図に沿って各種エフェクト等かけるので、写真や映像等に対するアカデミックな知識が重要である。

e. ライティング

レンダリングを行うために、シーンの中の物体（キャラも含む）に光をあてる作業である。演出的には写真撮影や実写映像撮影のライティングと基本的に一緒なので、撮影機材に関する知識が必要ないという点を除けば、基本的に求められる能力も一緒である。光と色を使ってストーリーを演出するセンスが重要である。

(2) 技術

① アートデザイナー

魅力あるキャラクタや背景を描く能力は基本ではあるが、キャラクタデザイン、背景デザインにおいては、種々の制約（ゲームのスペックやコスト制限等）を踏まえて、最良の結果を出すことが求められる。平面（紙）で見栄えのあるデザインができることではなく、3D（ゲーム内）で魅力的に見えるデザインかが重要である。プランナや他のグラフィックデザイナーと制約条件を確認しながら、魅力を失うことなくゲームにあったデザインをすることが求められる。例えば、あるバトルシーンでモンスターを1万ポリゴンでという発注で、モンスターは中ボス3体となると、1体約3000ポリゴンで表現できるデザインを考えるようなことが必要となる。

② モデルデザイナー

現在は、ポリゴンで造形することが主流である。

実際のポリゴンデータの制作方法としては、既存の基本的な形から分割を増やしたり、減らしたり、頂点を移動させたり、変形させる方法が主流である。難しい曲面生成に優れたナーブスから面を生成後ポリゴンに変換する方法等もある。

映像の質の高さを追求するあまり、ひとつのオブジェクトに対してのポリゴン数が多過ぎると表現できる内容が限定されるため、できる限り少ないポリゴン数で、高い品質の映像を得る技術が重要である。

UV マッピングを考慮したポリゴン分割、関節部分に破綻を来さないようポリゴンを分割する技術も重要である。

③ テクスチャデザイナー

テクスチャをモデル面に貼り付ける主な方法は、モデルに2次元のビットマップ（UV座標に展開したマップ）を貼り付ける方法（UVマッピング）がある。さらに、陰影によりモデルの形状を凹凸状に見せるバンプマッピングやバンプマップの欠点であるモデルのエッジ処理を施し

たディスプレイメントマッピング、反射の強度を設定する反射マッピング等を総合的に活用することで全体の質感を向上させる。また、種々のブレンドモードを活用することで質感表現がさらに増す。

シェーディング技術（表面に陰影を加える技術）も質感表現では重要な技術である。本来の色に加え、表面の性質（ガラス質、金属質、ゴム質等を反射光、光沢、鏡面反射率等で表す）、光源の位置やその色、周囲からの映り込み等を考慮し表現する。

④ モーションデザイナー

モーションのデザイン手法は手付けモーション、モーションキャプチャ、シミュレーションに大別できる。また、基本的な技術としてリギングも大切である。

a. 手付けモーション

ボディ、フェイス、無機物に大別できる。モーションデザイナーの思い通りにコントロールしながら制作できるので融通が利く反面、「長い尺」を「大量」に制作するのには向かない。

- ・ ボディモーション・フェイスモーション

クリーチャー等、人間と大きく異なる構造を有しているキャラクタをアニメーションさせる場合にボディモーションを手付けする必要がある。人間のモーションであっても、モーションキャプチャでは不可能な場合、ボディや表情にそれらしい動きをつけるための技術が必要である。ゲーム自体のテイストがアニメ的で動きのタイミングや表情そのものがデフォルメされている場合には、カートゥーン様のモーションを付ける技術も重要である。

- ・ 瓦礫や布、機械などの無機物

動きの情報量が少なく、すぐに結果を得たい場合に必要とされる。物理法則を無視したような、シミュレーションで再現不可能な動きを表現するための技術である。

b. モーションキャプチャ

モーションキャプチャ（以下MC）が採用されるケースとしては

- ・ 人間の動きで再現可能なもの
- ・ 長い尺を必要としているとき
- ・ 大量のデータを必要としているとき
- ・ 複数人が絡む動き

などが挙げられる。必ず複数の人間が関わるので小回りが利きにくく、少量しか使用しない場合はコスト的にも高くなりがちである反面、効率よく大量に撮影する事ができれば、安定した品質のアニメーションを大量に得る事ができる。

MCの場合、必要な技術は大まかに分けて「撮影」と「ポスト処理」に分かれる。

- ・ 撮影時

対象の動きを考慮に入れ、計測に必要なエリアを適切に設定する必要がある。また対象の運動によるマーカーの遮蔽が少なくなるようにカメラを配置し、正確にキャリブレーションを行うための技術が良いMCデータを得るためには重要である。他にも撮影の前に、スタジオ、演技者、小物の手配等が必要であり、全工程を通したノウハウの有無が効率に大きく影響する。

- ・ ポスト処理

MC データをコンピュータに取り込んだ後のデータのエラー処理を指す。得られた複数の MC データの編集し、加工するための技術が必要となる。

c. シミュレーション

シミュレーションが採用されるケースは、衣服が揺れたり、壁が崩れ落ちたり、といった複雑で多量の情報を含んだアニメーションが必要な場合がほとんどである。固形物だけでなく、煙や水等の流体をアニメーションさせるのにも使われる。

直接物体にアニメーションを入れていくのではなく、物体に対して重力や摩擦係数等の様々なパラメータを与えて、それを計算させることによってアニメーションを得る。最近ではゲーム機上でリアルタイムにシミュレーションを行うケースも増えている。

物理法則に基づいた動きをきっちり再現できるので、非常に説得力のある絵になりやすい。反面、現実世界の物理的要因を全てシミュレートするわけではないため、必ずしも意図通りの動きにはならない。よって頭で思い描いたような結果を得るためのノウハウの蓄積が非常に重要である。

d. リギング

アニメーションさせる対象物の実体とは別に構造体を設けて、その構造体を動かす事で実体をコントロールする手法を指す。現在、ほとんどのゲーム制作現場で用いられているが、対象物が非常にシンプルな構造の場合等、使用しない方がメリットが大きいケースもある。作業効率、確実性、品質等非常に重要な要素の一部に大きな影響を与えるので、様々な工夫、改良を重ねて固めていくケースが多くノウハウの蓄積が非常に重要である。作業工程の中ではモデリングの後、アニメーション作業の前に位置する。

使用する場合は以下のようなメリットがある。

- ・アニメーション作業の効率化
- ・アニメーションの品質の均一化への貢献
- ・対象物の構造に関わらず、ある程度モーションデザイナーに優しいカスタマイズが可能
- ・多人数で作業する際に混乱が起き難い
- ・アニメーション作業終了後でも対象物を差し替える等、柔軟な対応が可能

反面、以下のようなデメリットがあるためそのトレードオフを考慮して利用する必要がある。

- ・想定外の動きへの対処が困難
- ・アニメーション制作の手法が有る程度限定されるため、個人の手法的な好みが犠牲になる
- ・動作が重いためハンドリングが悪い

⑤ エフェクトデザイナー

エフェクト制作は、特有の技術はあまりないが、グラフィックデザイン全体（モデリング、テクスチャ、アニメーション等）の知識を広く浅く習得する必要がある。また、リアルタイムエフェクトでは処理時間（CPU,GPU）の扱いが非常にタイトに要求されるため、できる限り品質を保ちつつ、限られた処理内で構築していく技術（ポリゴン数やパーティクルの数、シェーダの選定、不必要なライティングの削減）が必要である。

- ・アイデア（デザイン、コンテ、演出）

主にバトルエフェクトで必要とされる。ポイントは、数秒のエフェクトではあっても起承転結は存在すること、使い古された演出でないこと、である。「攻撃力を2倍にする魔法」といった抽象的な発注に対する見せ方、サウンドのタイミング等（データ作業は行わない）も考える。

- ・エフェクトのモデリング

基本的にモデルデザイナーの項目に準じるが、複雑なモデリング技術は要求されない。

- ・エフェクトのテクスチャ

ドローイングソフトで作成するが、モデル面に貼り付ける作業はテクスチャデザイナーに準じる。パーティクルシステムを使ってエフェクトを作成し、レンダリングしてテクスチャ素材として使用する場合も多い。

- ・エフェクトのアニメーション

モーショントレーサーと異なり、ボーンを入れて動かしたり、モーショントラッキングは行わない。基本的にモデルアニメーションとシェイプアニメーションを組み合わせで動きを付ける。

モデルアニメーション

移動、拡大、縮小、回転といったエレメントの数値でのアニメーションや UV スクロール、モーフィング技術を利用して表現する。シミュレーションの知識もあると有利である。

シェイプアニメーション

爆発等の表現で1枚絵の連番素材を使い、アニメのセルアニメーションのように見せる技術を指す。シェイプアニメーションをパーティクルとして扱い多数表示して、画を構築することもある。例えば、煙突の煙などは、一枚絵の煙テクスチャを多数発生させて表現する。また、物理、流体シミュレーションの知識があるとより深くリアルな表現が可能である。

- ・センス

動きの緩急、発光物の発光感、色使い、適度な演出時間、わかりやすかつこいいといった発想力等が非常に重要である。

⑥ メニューデザイナー

ゲームの中でのルール・カテゴリ・階層構造を的確にイメージ化し、表示するための技術が求められる。メニューはゲームの内容により最終的に決まるものではなく、プランナーやプログラマーとのやりとりの中でゲームの持つ構造をユーザにわかりやすいレイアウトに落とし込む調整能力も必要となる。また、ゲームの世界観とマッチしたアイコンやフォントをデザインする技術も必要である。物体、人物の抽象化及び操作手法や動作、現象などを視覚的に表現するための技術も必要である。

⑦ カットシーンデザイナー

a. 絵コンテ

映像の流れや演出を絵で伝えるために、効果的な手法が数多くあり、それらを実践するための知識が必要である。また、絵で相手に様々なことを伝えられる程度には描く力が必要である。

b. アニマティクス

ほとんどの作業がアニメーションなので、3D ツールの隅々まで習熟する必要はない。

c. 編集

膨大なショット・データをハンドリングするので、編集ツールを充分使いこなせることが必要である。

d. コンポジット

編集同様、膨大なショット・データをハンドリングするので、編集ツールを充分使いこなせることが必要である。

e. ライティング

技術というよりは、光と色に関する卓越したセンスと知識を持っていて、それを忍耐強くトライ&エラーできる精神が重要である。

⑧ その他

3DCGに関するソフトウェアには、色々な種類があるが、いずれのソフトを使用する場合でも重要なことは、それぞれのソフトにあったスクリプトを覚えて使いこなすことである。スクリプトを使いこなすことで単純作業を軽減し、効率をアップさせることができる。よりクリエイティブな仕事に時間を使うことが可能となる。

(3) 知識

グラフィックデザイナーとして、各キャリアレベルで知っておくべき事。

① ゲーム制作全般

グラフィックデザイナーとしてのゲーム制作プロジェクトへの関わり方、関係のある他職種についての知識などが必要である。キャリアが上がるにつれ、知っておくべき視点も上がり範囲も広がる。最初は自身の業務に直接関わる範囲で十分であったものが、段々に間接的な範囲に広がっていく。例えば、プロジェクト立ち上げや試作に対しての知識、デバッグやマスターアップに関すること、さらに、画像素材の制作に密接に関係する知的財産権、肖像権や倫理規定に関することなどである。

② コンピュータリテラシー

コンピュータを使用して画像素材を制作するという業務なので、必要となるコンピュータリテラシーも特殊なものがある。

画像素材に関するフォーマットや処理できるアプリケーション、グラフィックボードについての知識などがそれに当たる。

③ コミュニケーション能力

グラフィックデザイナーの情報認知の順序や傾向について、例えば、論理的な説明より観念的な説明から導入する方が有効である、効率性よりも優位性に重点を置いた説得などである。

④ 一般知識

上記3項目以外の知識を指す。

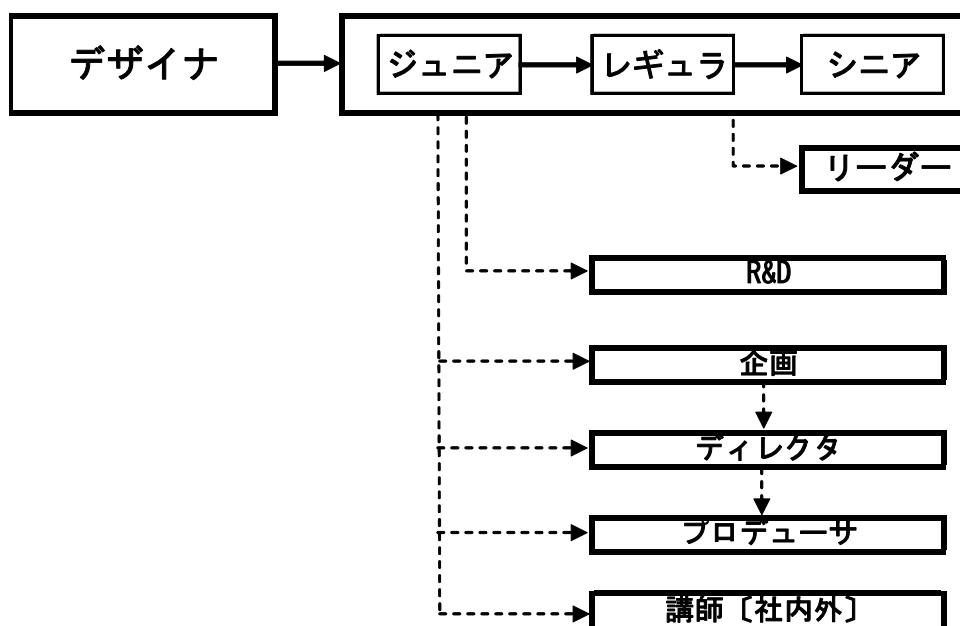
例えば、サブカルチャー関連の知識、美術や美術用語に関する知識、CGに関しての最新知識、ゲーム業界以外でのCG技術の使用例・トレンドなどが挙げられる。

4-2. キャリアパスの現状

キャリアの積み方として、特定の制作分野の専門家となるか、セクションあるいはチーム全体を率いる立場となる。さらに職種転換をするなど以下のような例が挙げられる。

- ・ある特定の分野の専門家として、ジュニアクラスからシニアクラスにスキルアップすること（例えばモーシヨンの専門家という意味）。
- ・ある特定分野の専門家であってもシニアクラスになってくるとチーム運営の一端を担うケースもあり、専門分野のセクションのリーダーとしてのマネジメントスキルが期待される
- ・デザイナー全般の専門家として、幅広い能力発揮をする。チーム全体のグラフィック〔アート〕の方向性を決める
- ・デザイナーからゲームデザイナー（プランナ）への職種転換し、ゲーム企画をする
- ・チーム全体のディレクタとして、プロジェクトを制作面から引っ張る
- ・プロデューサとして、総合的にチームを仕切る
- ・経験・知識を生かし、デザイナー技術研究を行う（例えば、共通技術の開発・研究）
- ・経験・知識を生かし、学校の講師等。

【グラフィックデザイナーキャリアパス例】



*各デザイナーの職種間の異動も当然キャリアパスとして行われる

*実線は標準的なキャリアパスであり、破線はまれなケースでのキャリアパスである。

4-3. 在職者の教育カリキュラムの現状

他のゲーム開発者の職種と同じ様に、明文化された教育カリキュラムや教育のためのプロジェクトがあるわけではない。したがって、育成・成長は OJT によって行われる。とはいえ、Off-JT が全く行われていない訳ではなく、SIGGRAPH や CEDEC といった技術カンファレン

スへの参加も行なわれている。但し、この様なカンファレンスへ」の出席は当然コストが必要とされるため、企業規模によっても参加人数や頻度にバラつきが見られる。

ここで、グラフィックデザイナーの場合、技術や知識だけでなく、クリエイティブな感性が制作物に大きな効果をもたらすキャリアアップの重要な要素となる。よって、上職者からの経験などからくる指導に加え、外部からの情報（各種メディアや自身の体験）により感性を高める事が重要となる。これを Off-JT と捉えれば、ロケハンと呼ばれる取材のための出張からアーティスト的な催しへの参加、あるいは映画の視聴さえも、指導側の目的を持った誘導さえあれば Off-JT と言える。

多くの場合、教育対象者に対し明確な教育者が存在する訳ではなく、上職者が教育者を兼任する。キャリアアップに関しても同じ様に昇格のための目標が提示される事より、より良い業務を行うための目標が提示されるケースが多く見られる。

しかし、この様な明文化されたカリキュラムがなく、教育プロジェクトが明確でない状況の中でスキルレベルをキャリアレベルに当てはめる価値観は各社ともに大きなズレは無い。この事は、現状を整理しデータ化することで今後の教育カリキュラムの整備に有効であると考えられる。

以下に、現状キャリアアップを誘導するために取られる方法をまとめる。

(1) ジュニアからレギュラ

上職者の監督下での業務を行っている状況からのキャリアアップであるので、「いかに上職者からの指示（リテイクを含む）が無くても業務を行えるか。」がハードルとなる。

上職者からの指示を受け、要求を満たす業務成果物を提出する事が十分に行われる様になると徐々に上職者からの指示回数や指示の細かさが減っていく。結果、上職者からの指示がない状態でも業務成果物を完成させる事が出来るようになる。

最初は、可能な限り多くの情報（手法や参考素材）を与え、成果物の完成への方法を習得させる。徐々に情報を減らし、単独で処理できるように誘導する。

(2) レギュラからシニア

このレベルでは上職者がアートディレクタにより近くなり、指示や要求される事が観念的な物になる。趣旨を導き出し理解して要求を満足させる必要が出てくる。加えて、ジュニアレベルの上職者となる場合も多くなり、新たに「指導」が求められる能力に加わる。

対象の上下に人が配置される（上職者と部下）。上職者からの指示を理解する事は以前と同じだが、さらに「伝える」という項目が追加され、その品質がそのまま成果物の品質となって現れる。

(3) シニア以降

アートディレクタかマネジメントスタッフか、キャリアアップの分岐点となる。任されているセクションスタッフへの指示・指導に加え、横並びの別セクションとの協調や折衝が多く発生する。しばしば、品質保持のため妥協が許されない状況に陥り、解決の手法を学ぶ様になる。

その際、指針となるのが自身の価値観とアートディレクタが示す方向性である。この二つを自身の中できちんと理解できているかどうかが解決に対しての大きな要素となる。スケジュー

ル管理に関しても、不確定要素が多くなり「担当者の力量から予測する難易度」や「問題発生
の確率」などを見極める能力が必要となってくる。

キャリアアップのための指示が行われる事は、ほとんど無くなる。逆に上職者への報告の項
目が増える。進捗や状況報告だけではなく、自身の考えや導き出した結論を報告する。これに
より今後のキャリアアップの方向を決める材料を蓄える。

4-4. グラフィックデザイナーを育成するためには

高性能ゲーム機向けソフトの開発において、新しい表現手法を作り出すグラフィックデザイナ
の育成は重要な課題である。

現在企業内で行われている OJT を中心にしたデザイナー教育は、今後も各企業の開発タイト
ルの特徴や人材マネジメント政策に則した形で継続されていくのものと思われるが、ゲーム開
発の環境が大きく変化したことによりグラフィックデザイナーに求められるスキルも高度化、多
様化し続けていることを鑑みて、企業内 OJT 中心の教育から体系的な教育プログラムへと改善
を進めていく際にポイントとなる事項を整理した。

(1) レベル別の到達目標

① ジュニアデザイナーの育成

a. 基本能力の再教育

近年、新卒デザイナーの採用試験の結果から、新卒者のデザイン能力（スキル表の能力項目：
デッサン、色彩、空間）の低下が指摘されている。モノは作るものではなく買うものであると
いう風潮が一般的になり、このことが少なからずデザイナーを志望する学生の能力低下にも影響
を与えていると思われる。

新卒者の基本能力の低下は、採用した企業のみならず業界全体のグラフィック制作能力の低
下をも招きかねないが、現状では各企業の社内研修で補わざるを得ないのが状況である。

また、この新卒者向け社内研修を通じて各自の適性を把握し、デザイナーの各職種への振り分
けを行っている企業もある。

② レギュラデザイナーの育成

a. 専門技術の教育

専門分野（職種）に関しては技術レベルが一人前で、上位者（シニア）の設計図をもとに商
品レベルの CG 制作を標準レベルの制作スピードで行うことができることを目標に教育を行う。
（制作物の最終的な品質の確認は上位者の指示を受ける。）

各職種が習得すべきスキルは、スキル表に整理されたとおりであるが、到達すべき品質を作
り込むために必要な技術の教育については、やはり OJT の活用が現実的である。

b. 制作管理技術の教育

上位者の指示により、知識・技術が必要な作業を下位者（ジュニア）に指示して行わせるこ
とが出来る程度の制作管理技術を習得させる。

③ シニアデザイナーの育成

a. 専門分野のリーダー教育

専門分野（職種）のリーダレベルで、デザイン設定、品質管理、効率化、新技術導入ができることを目標に教育を行う。

b.品質管理技術の教育

専門分野（職種）において、上位者からのCGコンセプトを受け、スケジュール（納期）と品質のバランスを考慮しながら、デザイン設定、仕様決定ができること目標に教育を行う。

c.制作管理技術の教育

制作プロジェクト全体のスケジュール策定、コスト見積もり、作業割振りが行え、マネジメントサイクルを通して適切な対応が行えること目標とする。

(2) 職種別デザイナーの育成

① アートデザイナーの育成

アートデザイナーは、ゲームに登場するキャラクタや世界観をゲーム内容や人物設定を参考に書き起こしデザイン画として作成する。最も重要なことは、一定の制約条件の中で、ゲームの登場人物として、魅力あるキャラクタが描けるか、世界観として魅力ある世界が描けるかである。そのため、この職種について、ゲーム内容に合った著名なキャラクタデザイナーを社外から起用したりするケースも多く、また、社内制作の場合には、社内のデザインコンペなどで人材を発掘し選任するケースもある。

ゲーム制作において、この職種は特殊な才能を必要とすることから、可能性を秘めた人材を少数人数選び、直接アートデザイナーの下で育成する。

② モデルデザイナーの育成

モデルデザイナーは、ゲーム機の性能向上に対応したモデル制作が行えるように新技術を習得させる必要がある。

具体的には、従来の絵画的なテクスチャ表現に加えて、法線マッピングを利用した精細な表面凹凸の表現技術の習得である。法線マッピングによる顔の皺、モンスターの皮膚、武器の彫り物、建物のレリーフなど、その利用範囲を広げる研究も必要である。特にリアリティを追求するタイトルではその重要性は高い。

また、ポリゴン数が数千万レベルのモデル制作は、ディテールの作り込みを要するデザイン・モデリング時間が爆発的に増大する傾向にある。モデルデザイナーは高精細なモデル制作技術を習得すると共に、効率的な制作方法も習得する必要がある。テクスチャと同じで、たしかな基礎力が最も重要である。これは従来のゲーム機と最新のゲーム機でも変わらない部分である。

最新のゲーム機で表現の幅が広がった事により高精度の表現を求められる。今までポリゴン数の制約上、テクスチャで処理していた個所を造型として再現する必要がある。その為、基本のプロポーションの取り方を重要視する。取り方を誤った状態で、細かな作りこみをした場合の修正は、時間を大幅にロスする。

対象を見る観察力には、要素をきっちり整理した上で理解して「観る」必要がある。

人の骨格とは何か、人の筋肉とは何かが、正確にわかっていないと最終的に何をすれば良いかが分からなくなる。また正確さを欠くと中途半端なものしか出来ない。

しっかりと観た上で表現（モデリング）する必要がある。架空のモノであっても、実際のモノを応用してより具体的なイメージを持たなければ表現は出来ない。

③ テクスチャデザイナーの育成

最新のゲーム機の表現を担うテクスチャデザイナーの育成においては、従来から行ってきた「絵画的なテクスチャ」の制作技術に加えて、光学系の質感情報を表現する各種マッピングデータ制作技術を習得する必要がある。

また、これらのマッピングデータはシェーダプログラムと対になって描画されるため、シェーダプログラムとの共同作業が容易に行える制作環境の構築も重要である。

一方、以上のような技術的側面については、重要ではあるが、あまり大きなハードルではなく、タイトルやゲーム機によって変化するもののため、対応力の問題となる。いつの時代でも最も重要なのは、確かな基礎力である。確かな基礎力というのは、光、形状、質感の状態を正確に理解し表現する力を指す。この力を確実にコントロールすることが大切であり、その訓練を徹底的に行うことが求められる。加えて、タイトルの持つタッチ、表現を理解できればどのようなデータでもこなせるようになる。従来でも光学的な要素を含んだ絵画的データを作っているが、要素を別々に描くことが今後求められる。理屈を整理して、最後に実機に表示される結果を予測しながら個別要素の作業を行うことになる。

④ モーションデザイナーの育成

モデルやテクスチャのリアリティが向上するにつれ、それらとバランスがとれたリアリティのあるモーションが要求されるようになってきている。

特に顔は見た目がリアルになるほど、ちょっとした動きのぎこちなさもモデルがリアルであるが故に違和感を増幅してしまう顕著な例と言える。

モーションデザイナーの育成においては、従来から行ってきた手付けやモーションキャプチャなどのモーション制作技術の向上を図っていくと同時に、リアルタイムシミュレーションによるモーション補助・補完の研究と導入を進めていく必要があると思われる。

⑤ エフェクトデザイナーの育成

一言でエフェクトといっても様々な分類に分かれる。炎・噴水・街灯の灯り・波打ち際等、背景に必要とされるマップエフェクト、雨や雪、霧といった環境に関するもの、魔法・ヒットマークといった遊びに関わる部分のエフェクト、イベントシーンに合わせて必要とされるものなどがある。これら特殊効果に加えて、映像面での演出や、ゲームを体験した時の感触といった「味付け」の部分も担っている。既成概念にとらわれず自由に発想しクオリティを追求する人材を育成することが必要である。

⑥ メニューデザイナーの育成

技術的な側面はあまり重要ではなく、世の中にあふれている様々なコンテンツ（特に Web 上）を常に意識し、ユーザにとって分かりやすい、ゲームの面白みを向上させるフォント、画面レイアウト等の研究を怠らないことが重要である。

⑦ カットシーンデザイナーの育成

この工程で最も重要な演出力を養うために、シナリオ全体の把握力、いかに全体を魅力的に見

せるかの研究が重要である。映画、テレビドラマ等、内容、尺の異なるコンテンツの見せ場作り、見せ方を研究することも大切である。必要とされるセンスが、写真や映像のカメラマンと共通するものが多いという点は、トレーニングのアプローチを考える上で、ぜひおさえておくべきである。

5. 共通して必要なスキル

職種に関係なく必要なスキルとして、ゲーム制作工程全体の知識、コンピュータリテラシー〔基礎知識〕、コミュニケーション力、種々のポップカルチャーの知識が上げられる。特にコミュニケーション力については、開発チーム規模が大きくなっていることや制作工程が細分化されていることから、セクション内あるいは、前工程や後工程との情報の伝達、共有が各開発チームの品質維持、アップ、スケジュールの維持において、最も重要なスキルとして認識されている。以下、各スキルについて詳説する。

(1) ゲーム制作全般についての知識

- ・ゲーム開発をチームで進めるためには、チーム内で自己が求められる仕事内容を把握する必要がある。そのため、ゲーム開発者各職種の業務内容及び相関関係。
- ・ゲーム制作の順序と流れ。(プロジェクト立ち上げからデバッグ、マスターアップまで)プロジェクトの組織構成内容とその意味
- ・制作の際に注意すべき事。(知的財産、倫理規定)についての知識が求められる。

(2) コンピュータリテラシー

ゲーム開発ではコンピュータを用いる。このため、職種を問わず、コンピュータについての基礎知識、コンピュータリテラシーが必要となる。ゲーム開発のために必要なコンピュータリテラシーとしては、

- ・コンピュータ用語 (CPU、GPU、メモリ、HDD、など)
- ・コンピュータの動作 (ビット→バイト、メインメモリ→キャッシュなど)
- ・OS (Windows、Mac、ディレクトリ、ファイル、など)
- ・ネットワーク基礎 (インターネット、メール、Web、サーバ、FTP、など)
- ・コンピュータ画像処理 (レンダリング、MPG、JPEG、WMV など)

などが挙げられる。これらについては、言葉の意味を理解できる程度の基礎知識は職種によらず必要となる。

また、業務用ゲーム機のプロジェクトや家庭用でも音楽ゲームのように専用インタフェースを用いるプロジェクトに参画する場合には、機械設計・製造の手法や工程についての基礎知識が必要となる。

(3) コミュニケーション能力

コミュニケーション能力については、開発チーム規模が大きくなっていることや制作工程が細分化されていることから、セクション内あるいは、前工程や後工程との情報の伝達、共有が

各開発チームの品質維持・アップ、スケジュールの維持において、最も重要なスキルとなっている。コミュニケーション能力を分類すると、

- ・意思の表現力（会話、分かり易い説明）
- ・意思疎通の方法（認識を合わせる為の素材などの提示、映画やコミック、ラフ案）
- ・フィードバックの引き出し方（情報発信後の理解度や認識の確認）
- ・対複数人数に対しての意思疎通の方法（説明資料など）
- ・文書作成能力

が挙げられる。業務では、会議資料、企画書、仕様書、説明書、報告書、引継ぎ書など、文書を用いたコミュニケーションも日常的に行われており、文書作成能力も重要なスキルとなっている。

(4) 一般知識

ゲームは、エンタテインメント作品であることから、ゲームの制作において表現内容等についての理解や意思疎通のために、他のエンタテインメントや現代文化についての基礎知識が必要となる。

- ・ポップカルチャー（共通言語として知っておくべき文化）たとえば、映画、アニメ、マンガなど
- ・コンピュータグラフィックス（用語、隣接する映画やアニメ業界でのトレンド）
- ・美術（用語や手法。古典から近代美術・ポップアートまでの簡単な歴史）
- ・その他雑学（知っている事で役立つもの）たとえば、歴史（戦国時代、三国志、中世ヨーロッパ）、メカもの（兵器から乗り物まで）など

また、簡単な数学の知識が必要となることも多い。

(5) 英語力

英語については、世界中の技術者・デザイナー・研究者がドキュメント・企画書・論文などを英語で作成していることから、日本語版のない資料を扱うことは避けられない。最新情報を得たり、必要な資料を集めたりするために英語を読解することがどうしても必要となる。また、GDC や SIGGRAPH のような国際会議に参加して情報収集をしたり、海外の企業と協業、外注したりする機会もあり、リスニング力・会話力が必要となる場面もある。近年ではアジア地域との協業・外注をする企業が増えているが、アジア地域でも、意思疎通に英語が用いられる場合が多い。