連載

ゲーム開発現場における プログラマのあり方

市川 宗孝
（Creature's Ignition）

✔ ゲーム開発プログラマとは「クリエータ」である

「クリエータ」というのは，自ら発案と設計を行い，最終的にプログラミングまでできる技術者のことを指す．

ゲームの開発現場において，ゲームの仕様を作るのは企画者（あるいはディレクタ）であるが，彼らは技術者ではなく，彼らの設計はゲーム内容に特化している．したがってその仕様には，プログラム的な矛盾や具体的な動作指示がなされていない場合が多い．「クリエータ」はその仕様の隙間を埋め，実際の動作を考案・設計しプログラミングしていかなければならないのである．

こういった「クリエータ」となれる技術者には，単にプログラミングできるスキルだけでなく，SEとしての能力も要求される．

✔ クリエータとしてのプログラマの技術力

■ プログラマの環境

ゲーム開発現場で使用される言語は多種多様である．コンシューマ・PCゲームではC・C＋＋がメインであるが，昨今はダウンロード販売やネットゲームの浸透によってXMLやJavaなどを利用するシーンも多くなってきている．昨今はiPadやAndroidのような機器が普及してきているため，インターネットと親和性の高いObjective-CやJava（＋Eclipse）の環境が利用されるシーンが多くなるであろうと思われる．

また，ゲーム本体の開発だけではなくゲームのデータを作成・管理するためのソフトを作成するシーンも多い．たとえば，EXCELで用意されたデータをゲーム用データに変換したり，数百のファイルを一括コンバートするためのバッチプログラムなど，それぞれのケースに応じて多種多様な言語を併用し作成している．

■ プログラミング以外の必要技術

ゲームを開発するためには，プログラミング以外の技術が必要になることもある．これらは知識の分野であり，ゲームのプログラマがクリエータであることの最たる特徴と言える．

最近のゲーム開発で必要となる最たるものは，ゲーム開発の多くの部分を占める映像技術である．3D表現には透視変換・光学計算，物体の自然な動作に物理演算，2Dの表現には映像解析・映像加工，時には実際のカメラによる撮影技法に関する知識が必要とされるケースもある．

映像技術以外にも，さまざまな思考プログラム，またそれらを大量にこなす高速化，音楽･音の解析や，変わったところでは音声や顔の映像から人の感情を読み取る技術などがある．

もっと単純な例を挙げるならば，ゲームに犬が登場するならば犬の習性を知る必要があるし，車が登場するならば車の運動性を知っていなければならないということである．

このようにゲーム開発の現場では，プログラマだからといってプログラミングに傾倒するだけではなく多様な知識が必要で，知識獲得に常に興味を持ち探索する意識が必要であり，また知識を得たときには常に「プログラミングするのであればこうなるで

あろう」というロジックを頭の中で組み立てる癖が日頃からついていなければ，いざプロジェクトに必要となったとき，途方にくれてしまうことになる．

✔ ゲーム学校での教育とゲーム開発の現場教育

■ 新人の意識に必要とされること

1991年以降，いわゆるゲーム学校と呼ばれる専門学校が多く設立されるようになり，現在のゲーム業界の技術者にもゲーム学校出身者が多くなってきている．

ゲーム学校では実機教育が行われているため，ゲーム学校出身者は即戦力として期待される一方，社会人としてあるいは技術者としてのプロ意識に欠ける新人が多く，新人の著しい質の低下に現場は悩まされてきた．原因はさまざまあるが，専門学校では先端“技術”の教育に特化されている傾向があり，技術者としての“心”の教育やプログラマとしての基礎の“鍛錬教育”があまり進んでいないことが原因であると思われる．

■ 技術者としての“心”の教育

技術者としての心得としては，日頃からの技術の探求や与えられたパートをこなすだけではなくプロジェクトを高い視点から見た仕事の進行などが挙げられるが，ユーザ視点での製作意識というのが特に重要である．バグをなくすことやユーザフレンドリーに製作することはもちろんであるが，期日を守るということも完成を待っているユーザに対しての責任であることを忘れてはならない．

現場においては，仕様通りに完成したとしてもレスポンスが悪かったり見栄えに問題があれば「ユーザがこれで納得するのか？」ということを再検討させるなどして教育を行っている．

■ 基礎鍛錬

基礎の鍛錬教育というのは特に重要である．基礎鍛錬というのは，仕様からプログラムのロジックを組み上げる技術のことであり，言語には縛られない技術といえ，何ごともロジック化して論理的に考えられる技術が必要になってくる．

昨今はプログラミング環境が充実していたり，インターネット等情報が豊富なために，よく考えずに行き当たりばったりでプログラミングしようとする新人がいる．繰り返し自分の頭で考え，ロジックを組み立てる癖をつけさせる必要がある．

✔ 考えるという技術の開発

■「考える」技術とは

話がそれるが，「掃除のできない人」は「掃除の仕方が分からない人」であるという説がある．掃除の仕方を教えられていない人は，物を移動させるだけで掃除した気になり，実際にはできていないという説である．これは「考える」ということができない人と同じ論理である．

学生時代の経験において，試験や受験に向け常に「正解」を教えられてきた人にとって，この「考える」という“技術”を習得できていないため，どこから何をどうやって考えていったらいいのかというロードマップが見えていないのである．

物事を順序立てて考えたり，プログラムを俯瞰した視点から整理することができる「思考の仕方」を身につけなければプログラミングは上達しない．

■ フローチャートを描かせる教育

私は後輩がプログラミングに悩んだ際，いつも「フローチャートを作成してみよ」とアドバイスしてきている．これは「考える」教育の一環である．フローチャートとは，プログラムの骨格を視覚化したものだと思われがちだが，それだけではない．目標とされる仕様を実現するために，プログラマがプログラムを構築する上でどう組み立てていったらいいのかの考えを視覚化し，考えをまとめていくためのツールでもある．

プログラムというのは一見して奇怪な英数文の羅列であり，プロであってもその流れを追うというの

はとても大変な作業である．ゼロの状態からその奇怪な英数文を目標に向かってプログラミングするのはなおさら困難な作業になる．新人にプログラムを作らせて「よく考えよ」と言ったところで，プログラムの中からどこが問題なのかを洗い出すことは「思考の仕方」ができていない新人には分からないのである．

昨今は，実務的なフローチャートを書類として作成する機会が減っている．が，幾多の現場を経験してきているプログラマは大抵のことはフローチャートをビジョンとして頭の中に描くことができ，必要に応じて描き出して有効活用している．新人に「思考の仕方」を根付かせるためにはフローチャートを描かせ，仕様を分析することとプログラムがどう組み立てられていくのかという様を視覚化して考える癖をつけさせることが言葉で説明するよりも手っ取り早い．

✔ 教育機関に望むこと

昨今の新人には特に「意識」の問題が多く，数回再提出を指示されただけで会社を辞めてしまう者までいる．与えられた仕様だけでなく，周辺の欠落した仕様も考え，プレーヤの視点にも配慮して「これでどうだ」と仕様の上を行く成果を何度でも突き出せる気骨が「クリエータ」には必要なのである．

技術面においては，フローチャートを描いてロジックを考える基礎鍛錬を徹底させてほしい．基礎鍛錬ができていれば，現場の技術をすぐに吸収して上達することができ，新しいことを任せていくことができる．

大学・専門学校を含め，プログラマ教育関係の皆様には，個別の技術よりも，そういった伸びの見込める新人を輩出していってもらいたいものである．

（平成22年7月20日受付）

市川宗孝
平成4年「コンピュータ総合学園HAL」を卒業後，ゲーム業界にて，企画・プログラミング・CG作成などを業務．代表作品：ライトブリンガー，ガメラ2000，モノポリーめざせ大富豪人生，RAYSTORM HDなど．