



⑧ デジタルゲームにおける "AI" の役割

遠藤雅伸 ((株) モバイル&ゲームスタジオ)

ゲームの原点は「競技」であり、一定のルールに従ってプレイヤー同士が優劣を競うものでした。

しかしコンピュータの登場とともに、架空のルールを持った世界を提供し、与えられた課題を達成していくような「独り遊び」が生まれ、アルゴリズムを使って既存ルール競技の相手をさせる「コンピュータプレイヤー (CP)」が始まりました。CP も最初は人間の相手をするものでしたが、今ではそこにとどまらず、CP 同士が対戦することが競技として成り立っています。

一方バーチャルな世界を娯楽として楽しむゲームでも、ルール以外の設定の構築である「レベルデザイン」に、CP と同様のアルゴリズムが組み込まれることが当たり前になりました。このようなアルゴリズムは、ゲーム業界の技術者には "Game AI", 特に混乱の恐れがない場合は単に "AI" と呼ばれます。いくつかの例を挙げて、後者のようなゲームへの AI 応用について考えてみましょう。

AI が要りそうで要らない場合

まずはカードを使ったジャンケンゲームの例です。

- グー、チョキ、パーの3種類のカードがある
- 任意の種類で5枚を手札とする
- 手札決定後、お互いの手札を公開する
- 手札から同時に1枚ずつを選び、ジャンケンをする
- 手札は消費され5回戦の勝敗で競う

というルールになります。

相手の手札が分かっているのですから、単純な確率計算に従ってカードを出すアルゴリズムを考えることができます (図-1 参照)。



図-1 ジャンケンゲームの手札例

図のような手札を想定しましょう。理想的な展開になればコンピュータは全勝が可能ですが、相手が何を出すかは分かりません。そこで相手が手札をどれも同じ確率で出すと仮定し、1戦目に勝つ選択を探ると、相手はパーが少ないので、グーを出すのが最も分がよくなると分かります。

このように、毎戦ごとに最も分がよいカードを出す簡単なアルゴリズムができますが、毎回同じ状況で必ず同じ手を選択するため、実際にやってみるとすぐにプレイヤーに読まれ、図のように対応されてしまいます (図-2 参照)。

ゲーム本来のルールとは異なるゲーム性をメタゲームと言い、このジャンケンゲームではCPのアルゴリズムを看破する部分が、メタゲームとして成立しています。これに対抗するようにアルゴリズムを変更していくと、プレイヤーとの間にいたちごっこ的な別のメタゲームが始まり、純粋な独り遊びの相手としてはとても使えません。

本来のゲームルールに戻すためには、アルゴリズムの解を放棄してプレイヤーの予測を裏切ればいいわけですが、常に放棄してはやはり看破されてし

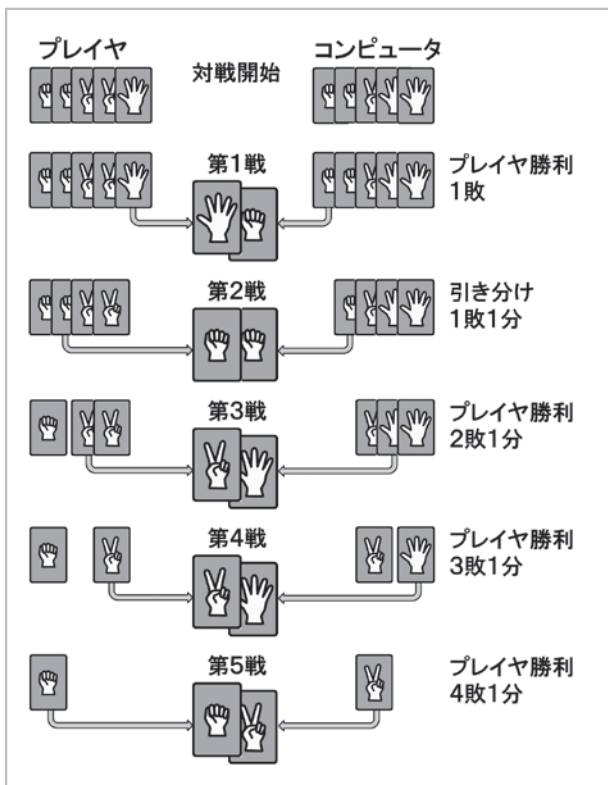


図-2 プレイヤにアルゴリズムを看破されたゲームの流れ

まうので、極論すると何を出しても良くなってしまいます。実際にやってみると、コンピュータが単にランダムに出すとまったく予想ができず、何と本来のゲームルールで楽しく遊べるようになるのです。

ここで大切なのは、学術的な AI における議論とは異なり、一般の方に楽しんでもらうゲームにおいては、プレイヤが常に理知的であるという前提を置けないことです。娯楽用ゲームにおけるプレイヤの目的は、それを「解く」ことではなく「楽しむ」ことにあります。「解く」楽しみは、ゲームを「楽しい」と感じた後のメタゲーム段階にあり、ゲームを「解く」ことがゲームの本質となっているのは、パズルゲームなどの一部に限定されます。

このように AI の不要な単なる確率ゲームは、現在暇つぶしとして広く遊ばれている、携帯電話のゲームなどでは人気があります。

キャラクタの個性を決める

アクションゲームでは、局面における最善の行動

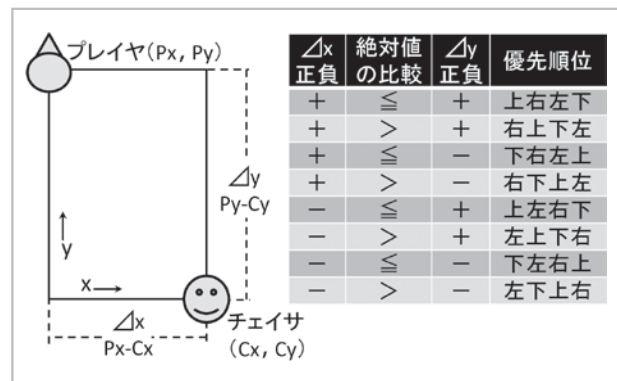


図-3 xy 座標上の位置関係だけから見た進路の優先順位

が分かっている、プレイヤーの操作技術が完璧ではないので最善の行動がとれないという、本質的な不確定要素が生まれます。次は迷路追跡ゲームの例です。

- 平面上に壁で仕切られた通路と分岐だけの迷路を設定する
 - プレイヤは上下左右に方向キーが押されている間進む
 - コンピュータにコントロールされた複数のチェイサがプレイヤーを追いかける
 - チェイサは一定速度で常に上下左右に動き、通路や分岐で逆戻りはしない
 - プレイヤがチェイサを振り切って全迷路を踏破したらクリア
 - チェイサがプレイヤーに追いついたらゲームオーバー
- というルールになります。

チェイサがプレイヤーを追い詰めるためには、平面上でプレイヤーと同じ座標を占めればよく、分岐点で簡単なアルゴリズムに従って進む向きを決定することで可能になります (図-3 参照)。

プレイヤーとチェイサの位置関係から座標上の差分 Δx と Δy を求め、その符号 (0 は正とする) と絶対値の大小に従って、チェイサから見たプレイヤーの方向にあたる、上下左右 4 方向の優先順位を決めるのが一般的です。図の場合は、 Δx が負、 Δy が正、絶対値が $\Delta x < \Delta y$ なので、表にあるように上右下の順になります。

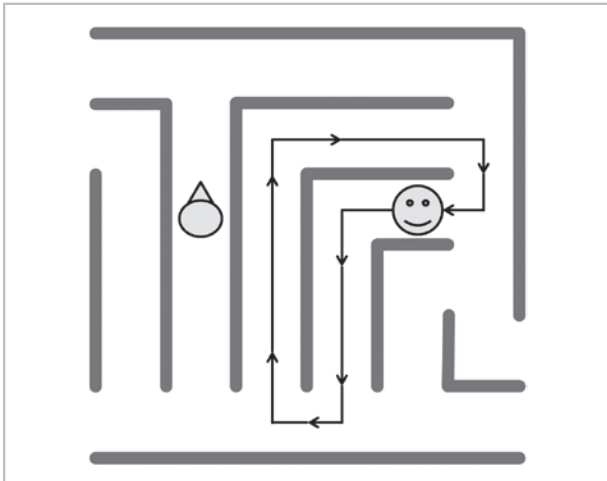


図-4 単純なアルゴリズムでは到達できない位置関係

通路を進んできたチェイサは、分岐のない曲がり角では優先順位にかかわらず、逆戻りしないというルールに従って道なりに進みますが、分岐がある場合は逆戻りを除く優先順位が上位の方向に進みます。こうすることによって、チェイサは確実にプレイヤーに近付くのですが、図のような特殊な位置関係の場合などはプレイヤーに到達することができません(図-4 参照)。

そこで他のチェイサには別のアルゴリズムで追わせることで、これを解消するとともに「プレイヤーが止まっていると確実にチェイサに追いつかれる」というゲーム性を確保します。いくつかのバリエーションの例として、経路の決定には同じアルゴリズムを使い、目標地点をプレイヤーの位置よりプレイヤーの向きへ2ブロック先に設定してみると、プレイヤーの行く手に先回りするような挙動になります。分岐でプレイヤーの進行方向と同じ方向に向かうようにすると、迷路の端などに向かったときに追い打ちをかけるようになります。迷路の中でプレイヤーとは点対称の位置を目指すようにすると、中心に近づいたときにプレイヤーを待ち受けるようになります。

行動エリアを制限したり、巡回コースを決めれば、プレイヤーがチェイサに追われたときに動く障害物として逃げ道を減らします。このようなプレイヤーの挙動を考慮しないアルゴリズムの場合は、見通し直線上にプレイヤーがいたら追いかけるようにするなどで、

ガードマン的な動きとなります。

このようなアルゴリズムの違いは、チェイサに個性を付けることになり、プレイヤーは経験的にチェイサの性質をラベル化します。逆にゲームのテーマデザインとしてこれを利用し、制御しているアルゴリズム別にチェイサに色を付けるだけで、ゲームへの興味を深めたりプレイをしやすくもできます。このようにアルゴリズムの違いを性格付けに使い、登場するキャラクターの個性を立てていく方法はとても効果的で、多くのゲームで演出として利用されています。

また上級プレイヤーの場合は、チェイサのアルゴリズムに不確定要素がないので、理想的なプレイをした場合の効率的手順を構築するメタゲームも存在します。いわゆる「攻略パターンを作る」という遊び方で、アクションゲームでは理想的な操作でパターン通りにクリアすることでもカタルシスが得られ、最適解が得られてもゲームへの興味は失われません。

プレイヤーの手間を軽減する

昨今では複雑で手間がかかる操作を「面倒だ」と感じるプレイヤーが増えてきました。これに対応して、ゲーム自体の規模を小さくして操作を簡単にする流れがありますが、ゲームの規模はそのままに操作をアルゴリズムで自動化する方法も広く使われています。特にロールプレイングゲーム(RPG)では、プレイヤーが操作しないキャラクター(ノンプレイヤーキャラクター: NPC)をプレイヤーの味方として登場させ、プレイヤーがNPCと協力してパーティ(一緒に行動するグループ)となり、敵との戦闘における操作を軽減しています。

今度はRPGにおけるパーティ戦闘を考えてみましょう。

- キャラクターはヒットポイント(攻撃によるダメージを受けると減り、回復によって最大値まで増えるパラメータ。これが0になると行動不能となり回復が不可能となる: HP)を持っている
- 戦闘は敵味方を含めてキャラクターごとに操作の順

番（ターン）が繰り返される

- 戦闘における各キャラクターの操作は、相手を指定して攻撃、対象を指定して回復の2つ
 - 攻撃は攻撃力に等しいダメージを相手に与え、HPを減らす
 - 回復は一定のHPを増やすが、対象の最大HPを超えない
 - 敵味方どちらかのパーティキャラクター全員が行動不能になったら、相手の勝利で戦闘が終了
- 単なる叩き合いというルールなので、何もなければ攻撃し必要なら回復するというアルゴリズムになります。このような場合は、できる行動に対して優先順位を付け、それに従ってその行動をとるための条件判断をします。最大不利益はHPが0となることなので、回復を優先として判断基準を考えましょう。
- HPの減少分より回復量が多い場合は、最大HPを超えた回復が無駄になり効率が悪い
 - (敵の攻撃力) × (敵パーティの人数) が次のターンまでに与えられる最大ダメージとなり、HPがそれ以下であれば次のターンまでに行動不能になる可能性がある

ここまででパーティメンバーの誰かが、回復量を上回るダメージを受けている場合から、行動不能になる可能性がある場合までの範囲で、回復すると判断する閾値を設定できます。さらに

- 相手の攻撃が1人に集中するとは限らない
- 次のターンまでの間に自分のパーティの他のメンバーが回復できるチャンスがある

ことを考えると、リスクを考慮した上で意図的に閾値を下げるすることができます。また相手が最後の1人となり、今回自分の攻撃で相手を行動不能にすることができるなら、回復をしないという例外的状態もあります (図-5 参照)。

パーティの人数を敵味方それぞれ3人ずつ、キャラクターのヒットポイントの最大値を100、回復量を40、攻撃力を15とすると、回復と判断する閾値を図のように31から60に設定すれば良いと分かります。このように幅を持った値を取ることができるの

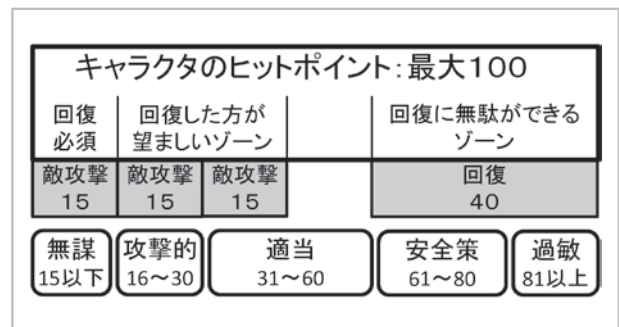


図-5 キャラクターのHPと回復閾値の関係

ですが、下限の31にするのと上限の60にするのでは、同じパーティとしてプレイヤーが行動した場合の印象がまったく異なります。31に設定したNPCは、イケイケで大胆な攻撃キャラに見え、60に設定すれば常に余裕を持った状態で戦いたい慎重派に見えるのです。

同じアルゴリズムを使っているのですが、パラメータの調整によって簡単に別の個性を作り出せるこの手法は、多くのNPCを設定する際の個性付けに大変役に立ちます。同じパラメータを適当の範囲から外れて10に設定すれば、自らの命を捨ててでも敵を執拗に倒そうとする狂戦士となり、90に設定すれば、味方の誰が傷ついても許せない病的な臆病者にもなります。

攻撃側についても、どの相手を攻撃するかによって同様の性格付けが可能です。敵パーティの中で最もHPが少ない相手を攻撃すれば、畳みかけるようなチームプレイ、あるいは弱った敵に追い打ちをかける陰険さを演出することができます。逆にHPが大きい相手を攻撃すれば、敵の攻撃を牽制しているようにも、空気が読めない間抜けにも見えます。この回復のタイミングと攻撃相手の決定を組み合わせるだけでも色々な性格付けが可能です。プレイヤーの好みに合ったNPCとパーティを組めるようになり、キャラクターに対する感情移入もしやすくなります (図-6 参照)。

| 回復閾値 | 攻撃相手 | 与える印象 |
|------|------|---------------|
| 10 | 最小HP | 執拗な攻撃手 |
| 10 | 最大HP | 迷惑なお調子者 |
| 10 | ランダム | 単なるバカ者 |
| 30 | 最小HP | チームプレイ重視の攻撃手 |
| 30 | 最大HP | 自分勝手 |
| 30 | ランダム | ちょっと間抜けなヤツ |
| 60 | 最小HP | 合理主義者 |
| 60 | 最大HP | 慎重なバックアップ |
| 60 | ランダム | 攻撃が苦手な弱気 |
| 80 | 最小HP | 人の陰に隠れた臆病者 |
| 80 | 最大HP | じっくり敵をいたぶるタイプ |
| 80 | ランダム | ビビってる初心者 |

図-6 キャラクタの設定がプレイヤーに与える印象

最強 CPU ではプレイヤーが楽しめない

複数のプレイヤーが遊ぶアクションゲームでは、プレイヤー同士の動きが干渉する同時操作のゲームが数多くあります。いわゆる対戦ゲームと呼ばれるジャンルですが、常に適正な対戦相手がいるわけではないので、1人でも遊べるようにNPCを用意します。レースゲームのドライバー、サッカーゲームの選手などは、現在自分が置かれている状況を判断するのに、多くの事象を考慮してその行動を決定するのですが、ここでは対戦格闘ゲームを例に考えてみましょう。

- 攻撃は小と大の2種類で、ガードもできる
- 攻撃には予備動作と、動作終了後に何もできない硬直時間（後硬直）がある
- 小攻撃は予備動作、後硬直ともに短く、大攻撃は長い
- ガードは予備動作、後硬直がない
- ガードにより小攻撃によるダメージは受けないが、大攻撃では軽減されたダメージを受ける
- 攻撃を受けるとダメージを追い、攻撃の強さに見合った硬直時間が発生する
- 予備動作中は攻撃を受ける
攻撃開始の入力から硬直が終わるまでを時間に沿ったチャートにすると、次のようなモデルになります

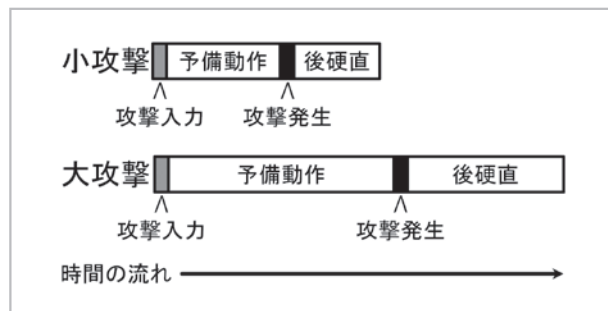


図-7 攻撃単体のタイムチャート

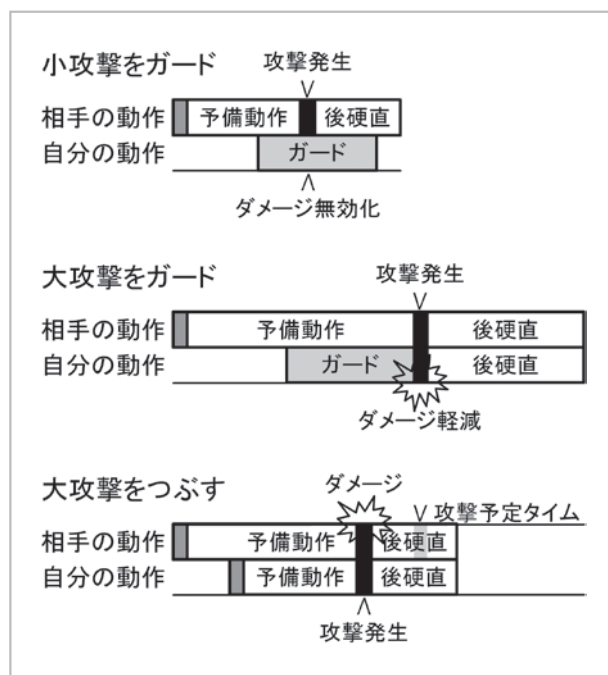


図-8 相手の攻撃に対応した動作のタイムチャート

す（図-7 参照）。

相手の攻撃に対し何もしなければ、攻撃が発生してダメージを受けますが、ガードをすれば、小攻撃のダメージを受けません。ガードは予備動作がないので、慣れてくれば相手が攻撃予備動作に入ったのをキッカケにガードできるようになります。ここで問題になるのは、大攻撃はガードしてもダメージを受けてしまうということです（図-8 参照）。

これを避けるために、相手が大攻撃の予備動作に入ったらこちらが小攻撃を出して、相手の動きを止めることができます。ただし単に予備動作だけを見て反応してしまうと、相手の攻撃が小攻撃だった場合はノーガードでダメージを受けたり、反応が遅れ

て相手の攻撃が先に発生してしまっても、同じくノーガードでダメージを受けるリスクがあります。大攻撃には小攻撃が有効、小攻撃にはガードが有効、ガードには大攻撃が有効というジャンケン構造がそこにはあり、常に相手の行動に対して正確に反応できれば、後出しジャンケンのごとく確実に勝てるのですが、失敗すると大きくダメージを受ける部分に對戦格闘ゲームの面白さがあるのです。

この後出しジャンケンというのが、コンピュータゲームにおける對戦アルゴリズムの使用では、大きな問題となっています。実際のジャンケンでも、動態視力が良く、反射神経、運動神経ともに発達していれば、相手が出そうとしているものを見て、それに合わせて自分が勝つ手をほぼ同時に出すことができるかもしれません。ところがコンピュータは人間とは問題にならないくらい判断速度が速いので、100%勝てる後出しジャンケンができるのです。

この對戦格闘でも、NPCはプレイヤーが行動を起こしたのと同時に何を仕掛けてきたのか正確に分かります。人間は予備動作を見て判断するしかできないのに対し、コンピュータは押したボタンの信号を見ているのですから間違いもありません。プレイヤーが小攻撃ならガード、大攻撃なら小攻撃をさせるだけで、100%引き分け以上のアルゴリズムになります。問題は、こんな相手と對戦してプレイヤーが楽しいのか？ということです。

そこで「適当に相手に花を持たせる」AIが必要になります。チェスや将棋などのコンピュータプレイヤーは「勝つ」ためのAIが頂点なのですが、それらも對戦ソフトに搭載される場合には、初級者プレイヤーの相手ができるよう手を読む深さなどを調整して「強さ」を変えます。すでに解の分かっているこの對戦ゲームのような場合は、プレイヤーの技量に合わせるためにデチューンするのではなく、プレイヤーが楽しめるようなデチューンをしなければなりません。

具体的にはプレイヤーの行動に対して反応する時間を遅らせたり、ガードを怠ったり、タイミングが分かりやすいように大攻撃を出したりするわけですが、

楽しさを考えるとキャラクターの個性なども想定する必要があります。実際の對戦格闘では技の種類や動きの速さ、体格や力の強さなどによってあらかじめいくつかのキャラクターが設定され、プレイヤーは好みのキャラクターを使って遊びます。同じキャラクターでもプレイスタイルによって個性が違い、また相手の戦い方を解析するメタゲームも発生するため、時流に従ってプレイスタイル自体も変化します。

これに対応して、時流に沿って変化する個性あるAIを最初から組み込んでおくのは難しいため、オンラインでバージョンアップをして組込みのアルゴリズムを変えていく方法もありますが、別のAIでNPC操作のアルゴリズムを調整する手法などもあります。その1つはプレイヤーの操作情報を収集分析し、プレイスタイルそのものを学習させる方法です。特定のプレイヤーが特定のキャラクターを操作したデータごとにまとめ、古いデータを参照の対象から外していけば、それだけで1つの時流に則した個性あるNPCを生成することができます。初級者のデータを使えば、上手いクセにわざと手を抜いているような不自然な「NPCに遊ばれている感」「NPCに遊ばせてもらっている感」もなくなります。

このように色々なゲームにAIはさまざまな使われ方をしますが、実際にはゲームルールはもっと複雑で、AIも状況判断のために必要なデータや設定するパラメータが多く、計算にも時間がかかります。作られたAIの最初の仕事は、必勝パターンが存在しないことの検証であることもしばしば。もしAIが必勝のパターンを作ってしまったら、ルールに立ち返って必勝パターンを消していくことが必要だからです。

そして一番大切な役割は、適正な難易度の設定です。ルールに従って作られたフィールドにも難易度はありますが、プレイヤーの障害としてダイナミックに設定されるNPCは、複合的に難易度を形成するため調整も難しく、机上の計算や想定と実際にプレイした感触が大きく異なることもあります。規模の大きいゲームであれば、この調整だけに数カ月をかけたりますが、その根底に流れるのは、規模にか

かわらずプレイヤーを「もてなす」気持ちです。

理想的には2回目のトライで突破できる程度の課題を用意し、プレイヤーの心を折らない注意と達成感を与えること。プレイヤーの混乱を防ぐために、新たな課題では、今までやったことのない操作を複数同時に要求しないことなど、アルゴリズムの違いや設置で対応します。もちろん商業的にはアルゴリズムを複数用意するにはコストがかかるので、パラメータの設定だけで対応できるような柔軟性と振れ幅のあるAIが、より優秀であり必要とされています。

別の視点から見ると、演出的効果としてのキャラクタ付けや場面設定のしやすさも要求されます。単に個性や特徴があればいいというだけでなく、プレイヤーの印象に残るものが適切にあるのが理想です。倒すのに苦労するボス敵など、設定自体が濃いものは当然ですが、広く薄く印象に残ることも繰り返し遊ぶゲームでは大切になります。たとえばゲームの快感の1つに、圧倒的な強さで相手を蹂躪する自在感の楽しさがありますが、これを演出するために意図的に「カモ」を作ったりもします。逆に苦手な

ものを作ることで、スパイスのようなピリッと感を出すこともしますが、スパイスと同じく入れ過ぎは元々の味を壊してしまうので注意が必要です。

今やデジタルゲームに使われるAIがデータとして扱いコントロールするのは、プレイヤーのモチベーションや満足感です。ある意味、いつの時代もAIが戦っているのは常に人間なのかもしれません。

(2011年10月31日受付)

遠藤雅伸
endoh.m@mgst.co.jp

1959年東京生まれ。ゲーム作家。1980年代初頭よりアーケードゲーム、家庭用ゲーム、PCオンラインゲーム、カードゲーム、携帯電話ゲームなどに、ジャンルにこだわらず多くの作品を創出し続けている。宮城大学客員教授。

