

特集 「エンターテインメントにおける AI」

ディジタルゲームにおける人工知能技術の応用の現在

Current Status of Applying Artificial Intelligence For Digital Games

三宅 陽一郎
Youichiro Miyake

株式会社スクウェア・エニックス
SQUARE ENIX CO., LTD.
miyakekey@square-enix.com, <http://www.jp.square-enix.com/>

Keywords: character AI, navigation AI, meta AI, agent architecture, behavior tree, goal-oriented action planning, evolutionary game design.

1. 概 論

ディジタルゲームの目標はユーザエクスペリエンス (User Experience, ユーザ体験) をつくり出し提供することにある。ゲーム技術はさまざまなユーザ体験をつくり出す科学であり、グラフィクス、サウンド、人工知能 (以下、AI と表記) などの複数の技術要素を組み合わせることで各ゲームタイトル固有の体験を生み出す。ユーザの体験には楽しい・怖い・感動する・興奮するなど言葉を越えたさまざまなバリエーションと振れ幅がある。ディジタルゲームにおける人工知能技術はその一つの要素であり「エンターテインメント AI」の一種である [三宅 08a]。

ディジタルゲームにおける人工知能技術の発展は、ディジタルゲーム自身の発展と共にある。1980 年代にディジタルゲーム自身が 1 個の小規模なソフトウェアであった時代には、人工知能技術の応用事例は極めて少数であり、人工知能技術やアルゴリズムを導入すると、ゲーム全体が AI のコンテンツのメインとなり「AI ゲーム」となった [森川 01]。しかし、ゲームが大規模化しソフトウェアとして複雑なアーキテクチャをもち、多数のモジュールへ分解されコンポーネント化が進められた現在、人工知能技術はそれぞれの機能モジュールに埋め込まれ、各モジュールを知性化している。ディジタルゲームの歴史は 1970 年代初頭から 40 年近くになるが、この 10 年でソフトウェアとしての構造が整備され、ハードウェアの急速な進歩が十分な計算・メモリリソースを提供し、ゲームと AI との融合を推進している。本稿では、急速に進歩するディジタルゲームにおける人工知能の全体像について概要説明を行う。

ロボット工学と同様にディジタルゲームの最大の特徴はリアルタイム性にあり、通常 30 分の 1 秒か 60 分の 1 秒のフレーム更新を行わなければならない。AI もまたこれに近い制約を受けるリアルタイムシステムであり、

この制限がディジタルゲームの AI に制限と特徴をもたらしている。

1.1 ディジタルゲームにおける三つの人工知能

ディジタルゲームにおける人工知能は 1980 ~ 90 年代にさまざまな応用が試みられたが、90 年代後半からその大規模化・構造化に伴い、しだいに役割が明晰化され、大きく三つの人工知能「キャラクタ AI」、「メタ AI」、「ナビゲーション AI」に分化することとなった [三宅 09]。

§1 キャラクタ AI

ディジタルゲームに登場する人物・動物はキャラクタと呼ばれる。プレーヤが操作するキャラクタはプレーヤキャラクタ、プレーヤが操作しない人工知能が駆動する仲間キャラクタ、敵キャラクタなどはノンプレーヤキャラクタ (NPC: Non-Player Character) と呼ばれる。NPC のことを単に AI ということもある (図 1)。

ディジタルゲームにおける最も大きな課題は、このキャラクタに搭載する人工知能「キャラクタ AI」の発展である。発展の方向は二つあり、一つは自律的な知性として高度化すること、一つはエンターテインメント AI としてセリフをしゃべりお芝居をしてプレーヤを楽しま



FINAL FANTASY XIV: A Realm Reborn © 2010-2015 SQUARE ENIX CO., LTD. All Rights Reserved.

図 1 『ファイナルファンタジー XIV: 新生エオルゼア』 (2013, SQUARE ENIX) プレーヤキャラクタ (手前) と木の形をした敵キャラクタ (正面奥)

せる演技性をもたせることである[三宅13].

§2 メタ AI

メタ AI はゲームシステムに宿る AI であり、ゲーム内に身体をもたない AI である。ここでゲームシステムとはゲームの進行を司るモジュールを指す。これまでゲームシステムはあらかじめ準備されたゲームコンテンツを逐次的に展開することが主な役割であったが、メタ AI が追加されることで、ユーザのスキルやゲーム進行を監視しながら、コンテンツを柔軟に生成・変化させていくことが可能となった。具体的にはプレーヤのスキルに応じてゲームの難易度を調整する、プレーヤの予測路上に敵を動的に配置する、ダンジョン・地形をユーザに合わせて生成する、ゲームに緩急をつけるために敵の出現率を動的にコントロールする、などである。

§3 ナビゲーション AI

デジタルゲームで「レベル」(level) といえ、地形とキャラクタなどを合わせたステージの全体構成を指す。ナビゲーション AI は、レベルとキャラクタ AI・メタ AI の間に立ち、レベル全体の静的な構造と動的な変化を AI が認識できる形のデータへ変換して提供する役割をもつ。例えば、地形を AI がパス検索できるナビゲーションデータとして保持し、ステージの動的変化を反映するのも、ナビゲーション AI の役割である。またレベルを解析する地形解析・状況解析の能力をもつ。ナビゲーション AI もまたゲーム内の身体をもたない AI であり、キャラクタ AI とメタ AI が、レベルの状態に沿って意思決定するためのサポートを行う。

この三つに分散した人工知能は協調してゲーム全体の AI システムを構成する(図2)。キャラクタ AI は各 NPC の知能として動作し、メタ AI はゲーム進行とともにキャラクタ配置・生成によってゲームデザイン・レベルデザインを調整する。ナビゲーション AI は、レベルを解析し、メタ AI、キャラクタ AI に地形情報とゲーム状態を提供する。これによって、メタ AI とキャラクタ AI がレベルデザインに応じた知能を発揮できるようにする。

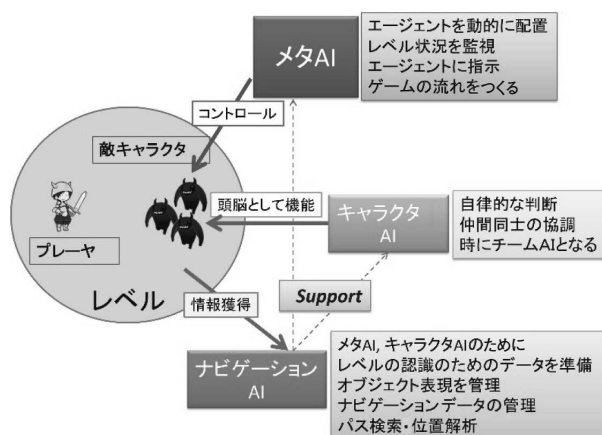


図2 三つの AI
「メタ AI」, 「キャラクタ AI」, 「ナビゲーション AI」

1.2 デジタルゲームの内部構成

ここでデジタルゲームの人工知能を説明していくために、デジタルゲームの現代的な内部構成を説明する。レベル内にあるもの(無生物)はオブジェクトと呼ばれる。例えば、武器や、岩、アイテム、家などである。レベルを構成する地面は地形(テレーン: terrain)と呼ばれる。

地形には森や草、壁や橋など移動できない対象も含まれる。キャラクタ、オブジェクト、地形など、ゲームを構成する要素をエンティティと呼び、これらが集まってレベルが構成される。したがって NPC の配置・動作、地形、オブジェクト配置全体をデザインすることをレベルデザインと呼ぶ。

エンティティ間にはインタラクションがおのおのの定義される。衝突や回転、重力や流れなどである。例えば河に箱を置くと流される。爆風で葉が揺れる。衝突でドアが陥没する、などである。エンティティはそれを動かすコードと、それを表現するデータからなる。このデータはアセットと呼ばれる。3D モデル、テクスチャ、シェーダ、衝突モデルなどである。アセットは主にアーティストによってつくられ、プログラムは主にエンジニアによって記述され、スクリプトはレベルデザイナーによってつくられる[三宅07b].

デジタルゲームは複数の独立した層(レイヤ)の積み重ねとして構成される[Gregory 09a]. キャラクタに関連する主なレイヤは描画、衝突、AI、アニメーションの四つのレイヤであり、各データはそれぞれのパイプライン(各レイヤの一連の処理)で独立に処理されるが、処理の順序には厳密なルールが存在する。描画パイプラインは1フレーム内で更新される必要があるが、AI パイプラインは必ずしも1フレームで動作させる必要はなく非同期で動作させることも可能である(図3)。

§1 描画レイヤ

描画レイヤはユーザに向かってゲームの空間を視覚的に見せるためのレイヤである。1980年代までは2Dでドットによるピクセルデータとその集合であるスプライトを単位として画面が構成されていたが、1990年代からは3Dグラフィックスに発展し、三次元モデルデータ、テクスチャ、シェーダを含む巨大なアセット群を擁するレイヤとなった。このレイヤに属するデータを描画データと呼ぶ。



図3 キャラクタ制御に関連する四つのレイヤ

§2 衝突レイヤ

衝突レイヤはエンティティ相互のインタラクションを司るレイヤである。このレイヤに属するデータを「衝突モデル」と呼ぶ。オブジェクトや地形の外形を直方体・球・平面などを組み合わせて覆ったデータであり、エンティティ相互の物理的インタラクションはすべて衝突モデルから計算される。例えばキャラクターとオブジェクトや地面、オブジェクト同士のインタラクションなどすべての衝突計算である。

§3 AI レイヤ

AI レイヤはAI が環境を認識するために用いるデータである。レベル内に存在する地形、オブジェクト、状況に対する知識表現データが属する。このデータを通じて、AI はレベルとその変化に対する認識を形成する。例えば一つの岩があったとして、その属性（固い・柔らかい、色など）、アフォードンス（動かせるか、壊せるか）、ルール（岩を動かすと何が起るか）などである。それぞれのエンティティは知識表現データを持ち、このデータをAI が受け取ることで、対象に対する認識が形成される。

§4 アニメーションレイヤ

キャラクター、オブジェクトなどの骨格や相互束縛などの内部構造を持ち、自律運動するエンティティのための運動を司るレイヤである。キャラクターには動物の構造を模した骨構造が用意され、あらかじめ用意されたアニメーションデータが再生されることで運動する、あるいは計算によって動的に運動が生成される。

各レイヤに属するデータはそのパイプライン内において独立に計算されるが、計算の順番はレイヤ間をまたいで厳密に決められている。

2. キャラクタ AI のつくり方

ゲーム開発では技術的には各レイヤのシステムを構築するが、コンテンツ開発として「キャラクターをつくる」という単位で製作が進行する場合が多い。キャラクターは描画・衝突・AI・アニメーションのモジュールを持ち、おのおのに対応して「描画データ」、「衝突データ」、「AI データ」、「アニメーションデータ」をもつ。それぞれのデータが各パイプラインの中で処理されるが、同時に一体のキャラクターは全体として自然な動きを構成する必要があり、特に後者三つの各レイヤ間の同期的な連携が必要とされる。AI は意思決定を行い、それによって必要なアニメーションを駆動するが、身体が駆動することで、他のオブジェクトや地面との衝突が発生し、身体は物理的フィードバックを受け、アニメーションは変形・変更される。アニメーションの変形・変更はアニメーションレイヤで行われる場合と、大きな変更の場合はAI レイヤまでフィードバックされる。このようにAI・衝突・アニメーションレイヤは互いに絡み合っており、この三者を組み合わせるとどのような全体のシステムを構築する

かがキャラクター製作の中心的課題である。

2.1 エージェントアーキテクチャ

キャラクターは内部構造として知能・身体をもち、一方で環境は自然構造をもつ。キャラクターの内部と外部を区別し、外部から情報を取得するセンサ（感覚器）、内部から外部へ影響を与えるエフェクタ（効果器）によって結び付けるモデルをエージェントアーキテクチャという[Russel 09]。内部と外部の境界には身体があり、センサもエフェクタも身体に含まれる。

エージェントアーキテクチャは「センサ」、「知能」、「エフェクタ」、「環境」の四つが結ばれる円環連結構造からなる。環境からセンサが得た情報と記憶をもとに、知能は知識と認識を形成し、意思決定を行い、エフェクタを通じて環境に影響を及ぼす。時間とともに変化する環境から情報を取得し、この運動を繰り返す（図4）。

ディジタルゲーム内部のエージェントアーキテクチャの特徴として、この円環の経路は情報の一方向の動的ループを形成する。この情報ループはインフォメーションフロー（情報回廊）と呼ばれる[Griesemer 02]。インフォメーションフローはいったん形成されると、世界と知能を動的に結び付け続けるヒステリシス機能をもっている。実際はインフォメーションフローは細い一本の流れではなく、さまざまな特徴をもつループが束になったものである[三宅13, 三宅14]。インフォメーションフローはまた知能内部をめぐる流れも存在する。これはさまざまな想起や内部思考に関連する情報の流れであり、内部循環インフォメーションフローと呼ぶ[三宅14]。

エージェントアーキテクチャのゲームキャラクターへの導入は90年代後半のMIT Media LabのSynthetic Character Group [MIT 03]のバーチャル空間のクリーチャの知能構造「C4アーキテクチャ」モデル（図5）[Burke 01, Isla 01]に端緒をもち、これが2001年のGDC（Game Developers Conference）で発表され、『Halo』（Bungie, 2001）のキャラクターの内部モデルとして導入された[Griesemer 02]。そこから、『F.E.A.R.』（Monolith Production, 2005）、『Killzone 2』（Guerrilla

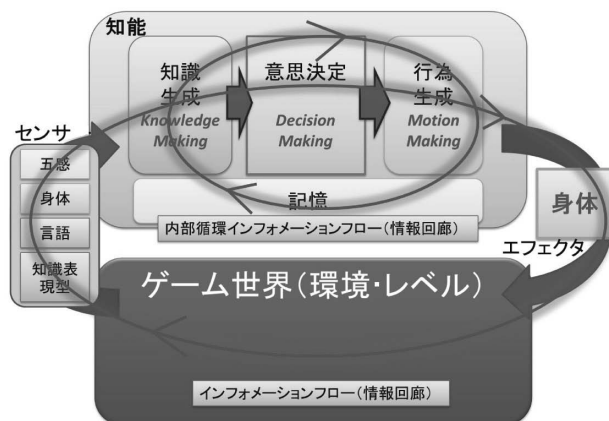


図4 エージェントアーキテクチャとインフォメーションフロー

Games, 2009) など大型の FPS (First Person Shooter) に導入され、ゲーム産業全体に広がっていった [三宅 08a, Orkin 05a, Orkin 05b, Straatman 09].

C4 アーキテクチャの特徴は黒板モデル (Blackboard Architecture) を基本構造としていているところにある。黒板モデルは 1970 ~ 80 年代に分散した小型 AI を協調するための仕組みであり使用例も多かった [石田 96, Nii 86a, Nii 86b]. 1990 年代ではいったん収束し、2000 年代に至ってエージェントアーキテクチャ内部の構造をモデル化するために用いられた。黒板モデルは三つの要素からなり、一つの機能に特化した人工知能 (KS : Knowledge Source) 群、KS 達が情報を読み書きする黒板、KS 達の動作を統御する調整モジュールであるアービタ (Arbiter) である。KS 達は間接的に黒板を通して接続されている (図 6)。C4 アーキテクチャも意思決定に用いる専門的なモジュール群 (図 5, 左列) を記憶領域 (図 5, 右列) を介してつながる形式となっている [Isla 02].

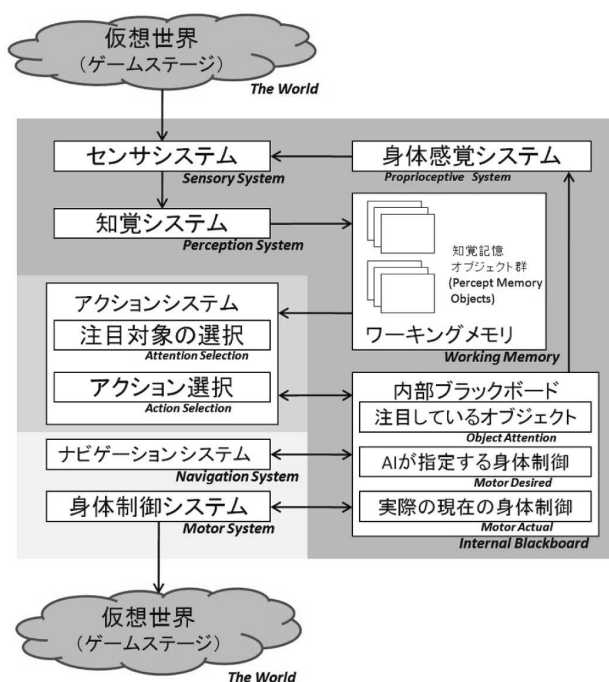


図 5 C4 アーキテクチャ [Burke 01]

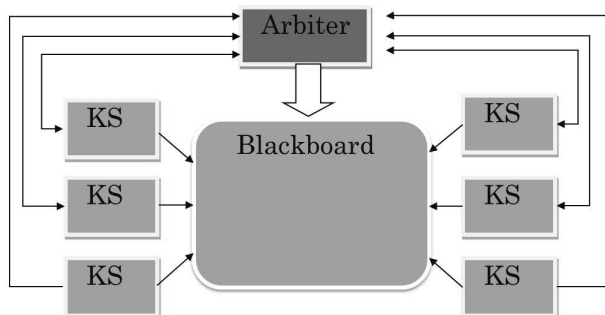


図 6 黒板モデル (ブラックボードアーキテクチャ) [Isla 02]

2.2 センサ

センサは本来的の意味は五感であるが、拡張して外界からの情報獲得全般を意味する。例えば身体に対する物理的衝撃や、言語コミュニケーション、抽象的な情報獲得全般を含む。世界と知能を結び付け具象から抽象までの情報を獲得できる太い情報のパイプを形成する。

§1 視覚

視覚の実装は、対象となるオブジェクトが自分から見えているか否か二値判定となる。通常この判定には 2 段階の過程を経る。最初にキャラクタの前方の扇型の視野領域に対象が含まれるかどうかを判定する [Walsh 14]. 次にキャラクタの位置からオブジェクトの位置まで直線を引いて、その間に障害物の有無を判定する。これを視線判定 (LOS : Line of Sight) といい、通常 1 本か位置をわずかにずらして複数本の視線判定を行う。視線の通ったオブジェクトからは、描画データではなく、知識表現データ (AI レイヤ) を取得することで認識を形成する。オブジェクトのもつ知識表現データは、そのオブジェクトが観測者からどのように認識されるべきかを定義するデータである。

視線判定の処理は認識対象が増えるほど重く、これを軽量化するために実装上はさまざまな工夫がされている。『Killzone 2』ではマップに敷き詰められた各ウェイポイントから全方位 384 の方向に対する LOS 距離が事前に計算し埋め込んでおくことで、ゲーム実行時の計算負荷を軽減している [Leeuw 09]. Saints Low IV ではマルチスレッドによる軽減を行っている [Canary 14].

§2 聴覚

聴覚の実装は、視線判定の代わりに音声到達判定によって行いが回折 (音の回り込み) の問題がある。あるオブジェクトが発生する音の伝搬範囲 (通常ある半径をもつ円) にキャラクタが入っているかどうか、あるいはより詳細に行う場合には、『Splinter Cell : Conviction』 (Ubisoft, 2010) のように音の伝搬経路検索をパス検索技術を用いて行う [Guay 12, Walsh 14]. 到達判定がなされたオブジェクトに対しては視覚同様に知識表現データを獲得する。このように五感の実装はリアルに感覚をシミュレーションするのではなく、オブジェクトのもつ情報を獲得する抽象的な手続きである。味覚・嗅覚・触覚に関してはほとんどシミュレーションしない。

§3 物理的な力

キャラクタは他のオブジェクトとの衝突から力を受け、位置、姿勢、アニメーションを変化させる。力を受けた事実をさまざまなレベルで認識する。身体がアクションによって反応する場合 (「倒れる」アニメーションを再生するなど)、自動的にバランスを取る場合 (外力に対して動的に歩行アニメーションを修正するなど)、意思決定が行動を変化させる場合 (行動を完全に变える) など、複数の知能・身体レイヤによる多重の対応が準備されている。また重力、爆風、河の流れ、人混み、群れ

制御のようにゲーム内に敷かれたベクトルフィールドから力を受ける場合もあり、同様に対処される [Anguelov 12, Fauerby 12].

§ 4 言語・コミュニケーション

ゲーム内のキャラクタ同士がコミュニケーションによって連携する、あるいは司令官からの命令に従って連携する場合、実際は言語というよりあらかじめ取り決めたプロトコルによってシンボルを伝達する場合が多い。集合、前進、退却などシンボル伝達は五感を通さず直接認識モジュールに伝達される [Isla 08, 三宅 10c].

2.3 エフェクタ

エフェクタは多くの場合、身体の衝突モデル、あるいは装備している道具・武器である。また司令官の AI の場合は直接身体はもたないが部隊や部下、大砲など間接的に世界に影響をもつ力を意味する。また物理的な影響力以外にも、言葉や命令のように抽象的なエフェクタである。行為をなすために必要な情報はセンサによって取得する、逆に行為をなした後、予想される結果はセンサを鋭敏に巡らせて感受する。このセンサとエフェクタの対応は協応と呼ばれる [Bernstein 03]. 例えば、敵に対して攻撃するには敵を目で追うことが必要である。攻撃を行えば実際に当たったという結果を見る必要がある。つまり意思決定された行為を中心に、センサリング、意思決定、エフェクタをつなぐ一連のインフォメーションフローが形成される。これが環境と知能を動的に結び付ける [三宅 14]. さらに行為と結果に対応は、学習アルゴリズムを機能させる基盤となる。

2.4 知 能

知能は認識、意思決定、行為生成、主に三つの役割をもつ。基本となるのが、自分（キャラクタ）が現環境下でどのような行動をなし得るかを認識することである。これは「アフォーダンス」(Affordance) と呼ばれる [Gibson 86]. 例えば「パンチ」、「キック」、「ガード」、「魔法を撃つ」というアクションができることを「認識」し、この中から最適な行為を選択することで「意思決定」を行い、決定された行為を状況に合わせてチューニングすることが「行為生成」である。例えば、対象の位置・高さに応じてモーションを修正する。これが「行為生成」の過程である。また「認識」(Knowledge Making) 「意思決定」(Decision Making) 「行為生成」(Action Making) の三つモジュールを流れていく情報の各瞬間のスナップショットを記憶 (Memory) へ蓄積する [三宅 13, 三宅 14].

§ 1 認識 (Knowledge Making)

認識の役割は、世界の姿を抽象的に自分（キャラクタ）を中心として再構成することである。センサによってレベル内の知識表現データを集め、その情報から自分を中心とした現在の世界を再構成し、意思決定に必要な情報

を集め推論によって補完する。この認識の精度が意思決定の精度を決定する。認識の形は意思決定においてどのような情報の形が必要とされるかに依存する。

認識が記憶に時系列的に積み重なることで、世界の状態の時系列変化データが形成される。この時系列データをデータマイニングすることで、未来の予測が可能になる。最も単純な例であれば、あるオブジェクトの座標を時系列で追うことで軌跡を認識し、その次の座標を曲線補完によって予測することができる [Burke 01]. 実際、これによって、待伏せや予測射撃を行うことができる。

認識で形成される知識を蓄積する記憶領域はワーキングメモリ (Working Memory) と呼ばれる。『F.E.A.R.』の場合、あらゆる記憶は統一事実形式 (WMF: Working Memory Fact) で表現され、ワーキングメモリ上に各対象オブジェクトごとに分類され時系列の順にスタックされる [Orkin 05a, Orkin 05b]. WMF は座標、向きなどの複数の各情報に情報の信頼度を付けて表現される [三宅 10a]. スタックされたデータをデータマイニングし、世界を 20 個のシンボルによって表現する。例えば、敵キャラクタが対象とすると、戦闘可能かどうか、どこに位置しているか、武装しているか、どんな乗り物に乗っているか、などである。このシンボル情報は 3.4 節で後述するプランニングのためのアクション表現に用いられる。

認識のもう一つの重要な役割は身体に対する認識を形成することである。生物が自身の身体状態をあらゆる瞬間に把握しているように、キャラクタも自身の身体状態を把握することで次の瞬間に可能な行動の広がりを予測する。ここで言う身体状態は、動的なパラメータとしては「速度、加速度、体勢、アフォーダンス、体力、魔力」、静的なパラメータとしては「最大速度、最大加速度、最大体力、攻撃力、もっている武器の種類と効果、もっている道具の種類と効果」などである。外界の情報と自己の情報の双方が意思決定には必要である。

§ 2 意思決定 (Decision Making)

意思決定は囲碁や将棋のようにターンベースかつ有限のグリッドからなるゲームの場合は、限られた手の中から次の一手を選択することである。この場合、意思決定は有限なゲームツリーの中から効率的に最善手を選択するアルゴリズムの問題に帰着させることができる。

一方、自然地形の中でリアルタイムに進行するアクションゲームの場合には、まず世界が有限の状態で表現することができない、さらに可能な行動も多様で無限にある。こういった連続空間、連続時間のゲームの場合には最善手によって次の状況を選択するのではなく、「即時的な状況に反射的に対応しつつ」、「自らの行動によって状況をつくり出す」方向で意思決定を行う。前者は反射型意思決定、後者は非反射型の意思決定と呼ばれる [三宅 12a].

キャラクタがもつ意思決定のアルゴリズムは主に 7 種

類あり,「〇〇ベース」と名称が付く。この「〇〇」は,ここに入る要素を思考の単位として用いるということである。例えば「状態ベース」(state-based)といえは状態を要素として思考するということであり,「ゴールベース」といえは,まずゴールを設定することから始める。「ルールベース」,「ステートベース」,「ビヘイビアベース」は反射型意思決定アルゴリズムである。次章で各アルゴリズムを詳説する。

§3 行為生成 (Motion Making)

意思決定は次の瞬間の行為のもととなる行為を決定する。もしゲームがボードゲームの場合は座標に駒を移動して終了である。しかし,アクションゲームでキャラクターが複雑な身体をもっている場合には,意思決定された行為の実行のために,環境に合わせて身体をアニメーションさせる複雑な過程を踏むことになる。

例えば「歩く」というステートを指定された場合に,レベル内でキャラクターが実際に歩くためには,経路に沿って地形のどこに合せて足を接地し膝を曲げるというモーションの作成が必要である。このために経路探索, IK (Inverse Kinematics), アニメーション再生, 衝突判定などを組み合わせる必要がある。このように意思決定の出力から実際のモーションを生成する過程があり,行為生成と呼ばれる。

3. 意思決定アルゴリズム

キャラクターの意思決定でよく使われる七つの意思決定アルゴリズム [三宅 12] を解説する。

3.1 ルールベース AI (Rule-based AI)

ルールを単位として思考する。通常はルールを複数もち,その状況に最も適したルールを選択することで行動を行う。状況によってルールを選択する思考を実装する場合もある (図 7)。複数のルールが発火した場合にどちらかを選択すればよいかを決めるために,ルール群には事前に優先度順位が付けられている場合が多い。

RPG (Role Playing Game) においてはユーザ自身に仲間キャラクターの知能を「ルール群とその順番を付ける」ことで作成する機能が与えられることもある。『Dragon Age』 (BioWare, 2009) などである。例えば「体

ルール = IF (行動条件文) then (動作命令文)

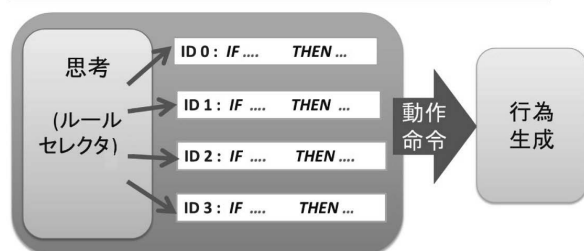


図 7 ルールベース AI

力が半分をきったら回復魔法を唱える」,「一番 HP の高い敵を攻撃する」などである。『The Sims 3』 (2009, Electronic Arts) では NPC の癖や特徴を出すために,各キャラクターにプロダクションルールを蓄積していく。例えば「夕方になったら TV を見る」,「朝起きたらジョギングをする」などである [Evans 10]。

3.2 ステートベース AI (State-based AI)

ステート (状態) はキャラクターの状態を指す。あるキャラクターが一つの状態にある場合には,キャラクターは状態で定義された行動を行う。例えば「歩く」という状態では歩き「攻撃」という状態では指定されたように攻撃する。キャラクターは複数のステートをもち,特定の条件で状態間を遷移する。この状態から状態へ移る条件を「遷移条件」という。複数の状態を遷移状態で結んだシステムを「ステートマシン」という。ステートマシンは,1990 年代から現在までキャラクター AI で最もよく使われてきた技術である。『Quake』 (id Software, 1993) や『UNCHARTED2』 (Naughty Dog, 2009) [Gregory 09b] など数千以上のタイトルで使用されている [三宅 10a]。また大規模なゲームでは一つのステートの中にステートマシンがあるような,階層型ステートマシンが使用されている (図 8)。

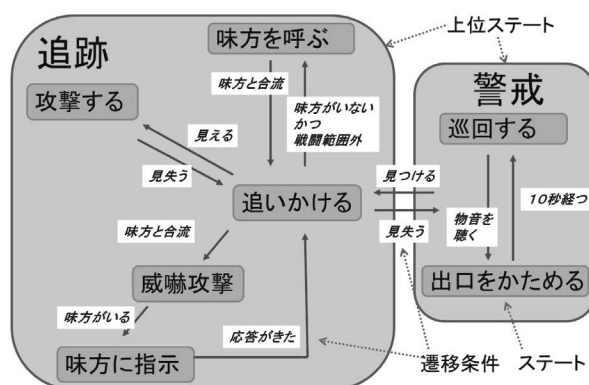


図 8 階層型ステートマシン

3.3 ビヘイビアベース AI (Behavior-based AI)

ビヘイビアとはキャラクターの振舞いのことである。ビヘイビアベースとは,キャラクターを動作を基本に思考を組み立てることである。特にこの数年で注目を集めている手法は,ビヘイビアツリー (BT: Behavior Tree) と呼ばれる手法であり,現在,ステートマシンに代わって最もよく使われる手法となっている。

ビヘイビアツリーは『Halo 2』 (Bungie, 2004) において Damian Isla 氏によって発案された手法であり,ステートマシンから環構造を排して,ルートから末端のビヘイビアに向かって行動を選択していく階層型ツリーの構造である (図 9) [Isla 05a]。末端のノードのみがビヘイビアであり,中間ノードはそれ以下のノードを含む中間階層ノードである。ルートから末端へ向かって選択が

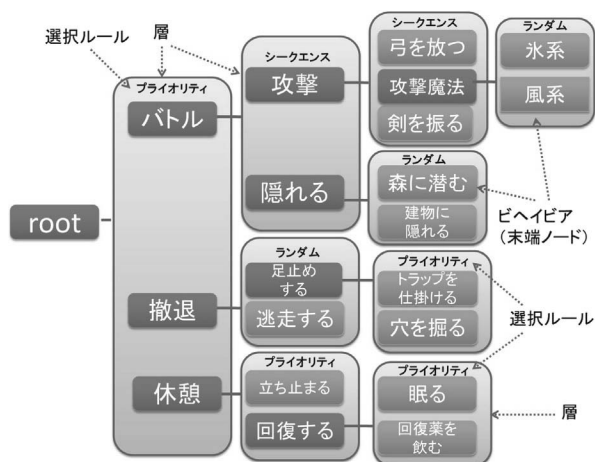


図9 ビヘイビアツリー

繰り返され、末端へたどり着いて初めてビヘイビアが実行される。末端で実行が終わるとルートに戻り、再び末端へ向かった選択が繰り返される [三宅 12a].

分岐ツリーのように親ノードによって分岐するのではなく、各層（ノードの集合）は子ノード同士が競合するモデルであり、各層に一つ競合のための選択ルール（Selection Rule）が指定される。最初に現在の状態で実行不可能なノードは選択から外される。現在実行可能なノードの中で「シークエンスルール」はあらかじめ定められた順番で実行し、「プライオリティルール」はあらかじめ定められた優先度に従って現在実行可能なノードの中で最も高い優先度をもつ実行可能なノードを実行する。「ランダムルール」はランダムにノードをピックアップする。

ビヘイビアツリーは制限の多い手法であるが、ツール上でゲームデザイナーだけで組み上げることができ、小規模から大規模にスケールすることが可能であり拡張性に優れている。またデバッグが比較的容易という利点があり、キャラクターにおけるデフォルトな方法になりつつある。『Spore』(Maxis, 2008) 『CRYSIS』(Crytek, 2012) など数百以上のタイトルで使用されている [Hecker 09, Pillosu 09].

3.4 ゴールベース AI (Goal-based AI)

「ルール・ステート・ビヘイビア」ベースの反射型のアルゴリズムの利点は即時的な対応が可能という点にある。例えば、「弓矢が飛んで来たら避ける」、「体力が少なくなったら回復する」という状況に対する適応性が俊敏につくれる。しかし長期的なゴール（目標）をもつことができない。逆に最初にゴールを決定し、それを達成するプロセスを考える、というアプローチがゴールベースである。

デジタルゲームで使用されるゴールベースの手法は二つある。ゴール指向型アクションプランニング (GOAP: Goal-Oriented Action Planning) と階層型ゴール指向プランニング (Hierarchical Goal-Oriented

Planning) である。

GOAP は、ゴールと初期条件の間をアクションの連鎖でつないでいく方法である『F.E.A.R.』では、STRIPS [Russel 09] を基本にした方法で、各アクションを「前提」、「実際の行為」、「効果」の三つを表現で記述する。「前提」、「効果」はシンボルで記述され、キャラクターのアクションをその形式で多数用意しておく（アクションプール）。「ゴール」、「前提」と同じ（あるいはそれを論理的に含む）「効果」をもつアクションをシンボルマッチングで探索し後ろ向きに連鎖させる。このようにして一連のアクションプランが作成される (図 10) [Orkin 03, Orkin 05a, Orkin 05b].

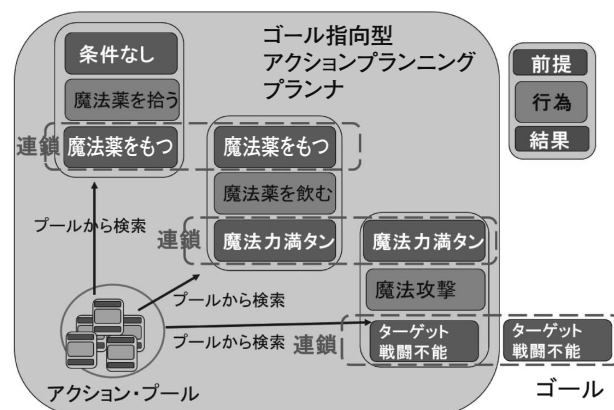


図10 ゴール指向型アクションプランニング (GOAP)

階層型ゴール指向プランニングは、まず最終的な大きなゴールがあり、スクリプトなどによってより小さなゴールへと分解を繰り返し、最終的に単純なコマンドまで分解する方法である [Buckland 04a]. 『Chrome Hounds』(FromSoftware, 2006) の NPC の意思決定で用いられ、NPC の性能・状態・ゲーム状況によってゴールの分解のされ方を変化せることで、戦況に応じた行動が生成された [三宅 06, 三宅 08b, 三宅 10a, 三宅 10c, 岡村 11a]. また『サカつく DS』(SEGA, 2008) では、ゲーム全体の組立に用いられている [安藤 10].

3.5 ユーティリティベース (Utility-based AI)

ユーティリティとは効用のことである。例えばキャラクターの「攻撃の効用」とは、敵に対して与えることができたダメージ、味方に対してかけた「回復魔法の効用」とは回復した割合である。例えば「戦う」、「防御する」、「魔法を唱える」、「回復剤を飲む」という四つの行為が可能である場合に、それぞれの行動の効用を評価式から評価し最高点を得た行動を選択する、これが効用による意思決定である。キャラクターによって評価式の形や式の定数を変化させることで意思決定に個性をもたせることができる [並木 11].

効用はキャラクターに用いる意思決定の中で最も古い手法であり、1980 年代から現在に至るまで継続的に用い

られている。行動選択のみならず、ゴール選択、武器選択、魔法選択、経路選択などにも用いられる [Buckland 04a]。例えば、複数の武器をもつ NPC がいる場合、現在戦っている相手に対して効用（この場合ダメージ）が最大になる武器が自動的に選ばれる仕組みを実装する。

『The Sims』（Maxis, 2000）は効用の概念を最大限に用いたタイトルである。キャラクターは8個の内部生理パラメータをもち（「人と話したい」、「眠りたい」など）、各パラメータの効用曲線が定義されている。効用曲線とは、そのパラメータの数値に対する効用のグラフであり、これは直線ではなく、パラメータが上がるほど効用度は上がりにくくなる。これを限界効用逓減の法則という。『The Sims』においてもキャラクター達はこの法則に従う。

例えば、「満腹度」（Hunger）パラメータが-80から10に変化したときの効用と、10から100へ変化したときの効用は

$$\begin{aligned} \Delta(-80 \rightarrow 10) \text{ の効用} \\ &= W_Hunger(10) * (10) \\ &\quad - W_Hunger(-80) * (-80) \\ \Delta(10 \rightarrow 100) \text{ の効用} \\ &= W_Hunger(100) * (100) \\ &\quad - W_Hunger(10) * (10) \end{aligned}$$

と計算される。ここで、 W_Hunger とは「満腹度」に関するウェイトであり、満腹度の関数である。『The Sims』では効用曲線ではなく、この「満腹度」を X 軸、「ウェイト」 W_Hunger を Y 軸に取った効用ウェイトグラフが準備される（図 11）[Forbus 02, 三宅 10a]。

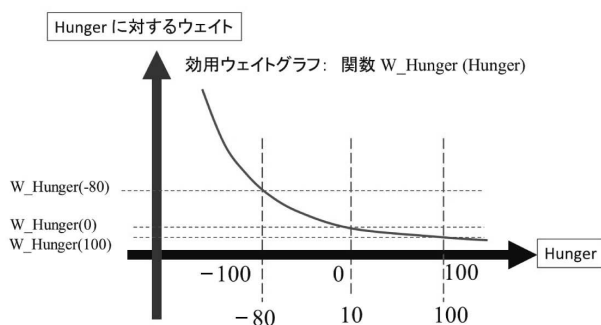


図 11 『The Sims』における Hunger パラメータに対する効用ウェイトグラフ（効用度関数 [Forbus 02]）

前者の効用は後者の効用より圧倒的に大きい。これは限界効用逓減の法則であり、不足している状態からそれなりに満足した状態に変化するほうが、ある程度満足した状態からさらに満足した状態よりも、ずっと大きな効用を感じる現象に対応させている。これによって『The Sims』の中の NPC は、空腹を満たすと次は別の欲求、例えば「他人と話したい」という欲求を満たそうとし、次はゲームをし、次はまた時間が経ってお腹が空くと空腹を満たす、といったように、複数の欲求の中で、その変化が最大の効用をもつ欲求に関する行動を自律的に選

択するように実装されている [Forbus 02, 三宅 12]。

3.6 タスクベース AI (Task-based AI)

キャラクターの行動の単位「タスク」を定めて、その単位によって行動を組み上げる仕組みであり、タスクはそれ自身の定義とタスク間の結合ルールが定められている。

階層型タスクネットワーク (HTN: Hierarchical Task Network) [Nau 03] は、『Killzone 2』のキャラクター・分隊長の意思決定モジュールに応用された手法である。階層型タスクネットワークはある目標があり、目標を達成するタスクに分解される（分解の仕方をメソッドという）。タスクはそれがそれ以上分解できないプリミティブな単純タスクになるまで階層的に分解される。タスク群には、タスクを実行する順序が定義されており、その制約を守れば順番を問わないため、結果として階層型タスクネットワークは一方向のタスクネットワークグラフを生み出す [三宅 12a]。『Killzone 2』では、この分解の仕方がスクリプトで記述され、状況に応じて変化し、状況固有のタスクネットワークが生成される [Straatman 09]。

3.7 シミュレーションベース AI (Simulation-based AI)

複雑な地形で自由度の高い運動性能をもつキャラクターを移動させる場合、そのモーションプランの作成が単純なロジックで解けない場合が多い。そこでキャラクターの運動を加速ベクトルの大きさとタイミングをある程度ランダムに組み合わせてシミュレーションし、その中で最良の軌道を見つける手法をシミュレーションベースという。実際に詳細な地形を使用する必要はなく、簡易的な当たりモデルの中、あるいはナビゲーションデータの中のシミュレーションを行えばよい。あるいは可能な軌道の組合せの中で動的計画法を用いて探索する場合もある [岡村 11a, 岡村 11b]。

3.8 階層構造・多層構造

大型ゲームのキャラクターは通常、意思決定の層が三層かそれ以上に多階層化されており、サブサンクション構造 [Brooks 86] を形成している。上位の階層が下位の意思決定をインタラプトする権限をもち、上位の階層はより抽象的な意思決定を行う。認識層では上位へ向かって情報が抽象化され、意思決定後は下位へ向かって行為が具体化・詳細化されていく [Buckland 04a]。この階層構造は多層構造 (multi-layered structure) と呼ばれる。WCCF (WORLD CLUB Champion Football, SEGA, 2004) シリーズで全体の AI の構成に用いられている [田邊 09]。

『Chrome Hounds』では、最上位層で「敵を叩く」というゴール意思決定がなされた場合、行為の詳細は、より下位の層のゴールへ分解を繰り返すことで詳細化・具

体化される(図12)[三宅06,三宅08b].「どの向きで」、「どこから」攻撃するか、そして攻撃ポイントへはどの経路をたどって近づくか、という具体的な問題を専用のモジュールによって解決していく。最終的に詳細な単純な行為のリストが生成され、実際にアニメーションデータが再生されることになる。この行為のリストを実行中に、それを阻害する行動が来た場合には、下位層で行為を修正するか、それでも防ぎきれない場合には最上位層の意思決定を変更することになる[三宅10a].

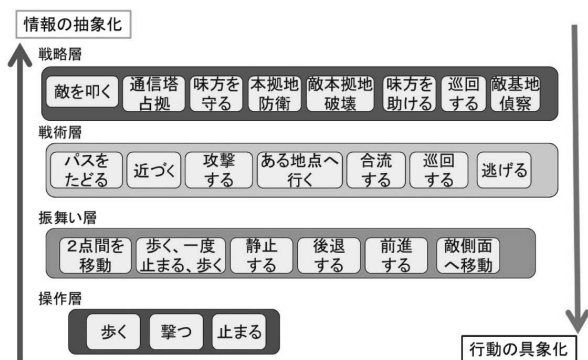


図12 『Chrome Hounds』のNPCの意思決定における階層構造・多層構造[三宅06]

例えば、攻撃ポイントへ近づく過程で、他の敵から攻撃された場合は反射的に避けるという行為は最下層のレイヤが行う。敵が位置を変えた場合は中間層の移動行動の目標地点を変更するだけでよい。しかし敵を見失うか、あるいは敵が味方によって先に倒されてしまえば、最上位層で改めて意思決定を行う。

3.9 意思決定とアニメーションの接続

上記で解説した意思決定アルゴリズムはシンボル操作による離散的な系である。一方、アニメーションシステムは数値的な連続的な系である。例えば意思決定は「投げる」、「走る」という二つのシンボル命令を出したとしても、アニメーションシステムはそれを環境の制限の中で実行し滑らかに二つの動作をブレンドで接続しながら実行しなければならない。また突発的な力に対して連続的に変形する必要がある。ここに意思決定層の離散系とアニメーションの連続系を融合する難しさがある。

もし意思決定と身体層の二層しかなければ、意思決定層は環境の制限と身体層の姿勢の限界を知っている必要があり、抽象的な意思決定と同時に進行するのは困難である。

意思決定と身体層をなるべく緩やかに融合するため、意思決定と身体層の間に複数の中間層が用意される。『Hitman Absolution』(IO Interactive, 2012)のNPCは意思決定と身体層の間に六階層に及ぶ階層構造があり、さらにポストプロセスで動作を修正する処理で動作を滑らかに見せつつ外力に対するさまざまな即応性をもつ[Angelov 13].

知能に階層構造をもたせるには、さらに別の二つの理

由がある。一つは多層化しておき、各層をできるだけコマンドやメッセージでつなぐことで、独立性を高めることで、各モジュールを独立して開発・発展させることが可能となる。もう一つの理由は演出的な目的でAIを外部から操作したい場合に、意思決定層を凍結して、別モジュールから制御コマンドを中間層に送ることで演出的な動きをさせることができる。この機能はチームAIやメタAIが直接各メンバを操作するときにも使用される[三宅10c].

現在、意思決定と身体層の間の階層構造のモデル化は探究されているところであり、未解決な問題として、タイトルごとにさまざまなモデルが提案されている。

3.10 キャラクタ同士の協調

キャラクタ同士を協調させる問題は、マルチエージェントの問題であるが、デジタルゲームには三つの方法がある。キャラクタ同士でプロトコルを決めてメッセージを送ることによってコミュニケーションさせる方法、チームAIと呼ばれる上位の司令官にあたるAIが命令を作成し上位から管理する方法、特別な協調システムによる方法である。

§1 メッセージングによる方法

プロトコルを決めてメッセージによってNPC同士を互いに協調させ、集団全体に統制をもたせる手法である。この手法は全体を一括管理できないのと、めまぐるしく状況が変わるゲーム状況においては不安定で調整が難しい面がある。よって、メッセージの内容は極めてシンプルで即時的な内容に限られる。例えば、ターゲットとする敵の共有や動作の命令や要請(「救援」、「護衛」など)である。また意思決定で七つのアルゴリズムを紹介したが、ステートベースであればメッセージによって相手の取るべきステートを指定する、ゴール指向であれば直接相手のゴールを指定する直接的な方法も取られる[三宅08].

§2 チームAI

チームAIは、キャラクタ達を協調させるためのAIである。キャラクタの1名が司令官となるか、あるいは、物理的実体のない司令官としてチームAIを製作する場合がある[Sterren 13, 田邊09]. キャラクタ同様にエージェントアーキテクチャによってつくられるが、センサとエフェクタが特別である。センサは個々のキャラクタからの報告による。チームAIはメンバから吸い上げた報告によってチームが直面している状況を組み立てる。エフェクタは各メンバへの命令である。つまりチームAIは、メンバからの報告とメンバへの命令によってコミュニケーションしながら全体を統制する。チームAIを導入した場合、メンバ間のコミュニケーションは煩雑さを避けるため構築しない場合が多い。

『Killzone 3』(Guerrilla Games, 2011)の場合、チームの階層構造は3層構造になっており、各分隊のメンバ、

分隊長、さらに分隊長を束ねる司令官が存在する。チーム AI に当たる分隊長、さらに各メンバは司令官に向かって現状を報告し、司令官は局所的な情報を集めて俯瞰的な状況を構築し、分隊長へ命令を下し、分隊長はその命令に従ってメンバへ命令を行う。各メンバはウェイポイントを用いて敵脅威の認識（敵が近づくとき近くのポイントの脅威度を上げる）とパス検索を行うが、各分隊はそれを階層化した戦術用大局グラフ上で敵勢力分布をインフルエンスマップ（Influence Map）を用いて計算し、分隊としてのパス検索を行う（図 13）[Champandard 08, Straatman13]。

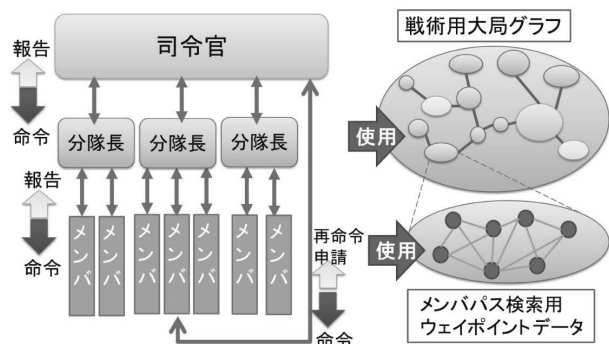


図 13 『Killzone 3』 NPC の組織図とナビゲーションデータ

§ 3 特別な協調システム

チーム全体を統御するのではなく、チームにおいて必要な部分的な協調機能を製作する場合がある。最も多いケースが複数の NPC と敵複数とが対戦する場合に、誰がどの敵を攻撃するか、という「ターゲティング」の問題である。チーム内の各メンバがランダムや一番近い敵を標的にしていたのでは効率が悪く、プレーヤから見た場合、明らかに手を抜いたように見える。そこで適度に戦力を分散させるために、ターゲットを振り分ける仕組みを導入する。

最も多く使われる方法はブラックボードを使った方法である。この場合、ブラックボードは伝統的な手法のしており、キャラクター間の協調のために用いる。まずアービタが敵の一体一体の強さを観測し、各敵の強さに応じてブラックボード上にスロットを用意する。強い敵 A は 3 スロット、弱い敵 B は 1 スロットのように、である。ここでスロットはその敵に対する攻撃権を意味する。各スロットに対して、メンバが攻撃の申請をし、アービタが許可することで初めてメンバはスロットを占有できる。その敵のスロットがすべて満たされれば、それ以上はエントリーできないので、メンバは他の敵のスロットにエントリーするしかない。このようにしてターゲットを振り分けることができる。

またプレーヤを複数の敵 NPC が囲んだ場合、複数の同時攻撃による袋叩きが発生するのを防ぐために、攻撃トークンを一つ用意し、そのトークンを順番に取り合うことで一度に一体の NPC しかプレーヤを攻撃できない

ようにする [Dawe 13]。

4. ナビゲーション AI のつくり方

意思決定はロボティクスと重なる部分が多いが、ナビゲーション AI はデジタルゲーム固有の部分が多い。ナビゲーション AI は本来の意味では、キャラクターの経路を導く AI のことであったが、近年複雑化するレベルデザインに対して、その構造と変化、行動に応じた情報を管理しキャラクター AI・メタ AI に提供する人工知能として、環境に宿る人工知能として構造化された。地形解析・状況解析などリアルタイムに大局情報を抽出する機能をもっている。

実空間において環境に任意の情報を埋め込むことはできないが、デジタル空間では可能である。あらゆるオブジェクトに対しキャラクターの認識に必要な情報を埋め込むことができる。逆に言えば、デジタルゲームではキャラクターのセンサをつくり込むことはなしに、本来、生物や動物として五感や認知を通じて得る情報を、あらかじめレベル上の地形やオブジェクトにもたせておくことができるのである。それによって環境を解析する重たい処理をナビゲーション AI に分担することができ、キャラクター AI・メタ AI はナビゲーション AI の用意する情報を受け取ることができる。

環境認識の基本構造となるのが、空間表現データとオブジェクトデータであり、それに付加的に認識に必要な情報を埋め込んでいく。静的な情報は開発中に準備される。

ナビゲーション AI の役割は世界の側からキャラクターに向かって「本来動物であれば認識すべき環境の情報」を伝えることである。当然、その情報はそのキャラクターがどのような生き物であるかに依存する。つまりナビゲーション AI の役割とは、環境世界に認識に必要な情報を用意し、それぞれのキャラクターの主観世界をつくることである。ナビゲーション AI が提供する情報は二つあり、感覚による対象の情報と、対象に作用しようとするときに必要な情報であり、前者は「感覚世界」、後者は「作用世界」と呼ばれる [ユクスキュル 05]。これらは生態学、認知科学の概念であるが、人工知能では知識表現がこの概念に対応する。オブジェクトや地形に対する知識表現は、いくつかの種類に分類される。

4.1 世界表現

レベルデザインの地形全体・空間全体に対する表現を特に世界表現（WR: World Representation）と呼ぶ [Straatman 05]。世界表現にはキャラクターのナビゲーション（位置移動）に必要なナビゲーションデータとそれに付随するデータの 2 種がある。ナビゲーションデータで最もよく使われる形式はウェイポイントとナビゲーションメッシュである [Buckland 04b, Buckland

04c].

1990 年代初頭までデジタルゲームの多くはナビゲーションデータをもっていなかった。キャラクタがその上を移動するポイントデータか、固定領域（例えば部屋の中）が指定されているだけであった。あるいはスクリプトの中で行ってはいけない場所に行かないよう制御していた。例えば「部屋の中心から 5 m 以上離れると引き返す」などである。

しかし、1990 年代半ばの 3D ゲームの出現のタイミングから、徐々にナビゲーションデータが導入され、その上の任意の 2 点に対して A* アルゴリズムによって動的にパス検索を行う手法 [Buckland 04b, Snook 01] が急速に広まり、現在では標準的な方法となっている。90 年代後半にはデータの作成のしやすさからポイント群を地形に沿って分布し、ポイントの近接関係をネットワークグラフに落としたウェイポイント法が最も多く使われていた。2000 年初頭頃からは、ナビゲーションメッシュ法が広く使われるようになった [三宅 07a]。パス検索によって、キャラクタはレベル上の任意の 2 点間を自由に移動する能力を得たのである。また『ARMORED CORE V』(FromSoftware, 2012) では三次元空間におけるパス検索が実装されている [岡村 11b]。

ナビゲーションメッシュは地形を凸多角形で覆ったデータであり、近接メッシュ同士がリンクをもつネットワーク型データである。ナビゲーションデータは地面衝突モデルより粗くてもよいが、一般に開発では広大なマップからナビゲーションデータを自動生成することが課題となっている。ナビゲーションデータを衝突モデルから自動生成するミドルウェアも多数存在する。

MMORPG『ファイナルファンタジー XIV：新生エオルゼア』においては広大なマップからナビゲーションメッシュを自動生成しており（図 14）、メッシュデータはサーバー上に置かれている。またパス検索の実行時の負荷を軽減するため A* アルゴリズムではなく重複領域をもつルックアップテーブルを参照する独自の方法でパスを検索している（図 15）[Gravot 12a, Gravot 13, 米田 12]。

ナビゲーションメッシュは面であるから、より地表の情報を反映しやすい。「雪」、「水」、「草」、「コンクリート」、



FINAL FANTASY XIV: A Realm Reborn © 2010-2015 SQUARE ENIX CO., LTD. All Rights Reserved.
図 14 『ファイナルファンタジー XIV：新生エオルゼア』のナビゲーションメッシュとリンク構造



FINAL FANTASY XIV: A Realm Reborn © 2010-2015 SQUARE ENIX CO., LTD. All Rights Reserved.

図 15 『ファイナルファンタジー XIV：新生エオルゼア』のナビゲーションメッシュ上のパス検索結果

「土」など地表の属性データを各メッシュに付属させ、この地表情報をパス検索のコストとして用いることで、各キャラクタに応じた個性的なパスを導くことができる [Buckland 04b, 三宅 08b]。例えば泳げないキャラクタに対しては水属性のメッシュは通過コストを大きくしておけば通らないし、雪上移動が得意なモンスターには雪のコストを低くしておけば雪に沿って移動することになる。また「一方向にしか通れないメッシュ」属性を用いることで、キャラクタに落下、回り込みを実現することができる [Gravot 12b, Gravot 13, 米田 12]。その他にもゲームデザインの要請に合わせて、隠れやすい場所、遠距離攻撃に適している場所などの情報を含ませていくことで、より深い環境認識を AI に伝えるほど、キャラクタはより環境を巧みに利用した移動が可能となる [三宅 06, Straatman 05]。また『Splinter Cell：Conviction』では、オブジェクトによって通れない場所ができる、あるいは逆に元あったオブジェクトが爆破されて通れるようになる、など動的なゲームの変化に対して、ナビゲーションメッシュの形状を変化させて対応する機能も実装されている [Walsh 10]。

4.2 オブジェクト表現

ゲーム世界の中の各オブジェクトには、キャラクタがそのオブジェクトを認識する知識表現のデータをもたせておく必要がある。通常、知識表現は知性内部の知識フォーマットを指す言葉であるが、デジタルゲームではオブジェクト自身のもつ情報として扱う。オブジェクトがもつべき情報は、キャラクタがそのオブジェクトに対してなすべき行動の種類だけある。最もよく使われるのが以下の表現である。

§1 シンボル表現

対象の分類のためのシンボルである。自然物か、人工物か、キャラクタか、敵か、味方か、敵であるとしたら、どんな種族か。『Halo 2』では、敵全体の分類ツリーがあり、ツリー構造でキャラクタ全般を管理している [Isla

05a].

§2 アフォーダンス表現

アフォーダンスはそのオブジェクトに対してキャラクターがなし得る行動である。例えば「岩」が軽ければ「特定の方向に動かすことができる」、「レバー」であれば「倒す」ことができる。さらに詳細なデータとして、オブジェクトのどの向きに押せば、その岩をどちらへ動かすことができるか、「車」に「乗り込む」のであれば、どの位置から乗ることができるか、など必要なデータとともに行動が指定されていることで、キャラクターはそのオブジェクトに対して正確な行動を取ることができる。『Halo 2』では車など特殊なオブジェクトに対してアフォーダンス情報が準備されている [Isla 05b]。『モンハン日記 ぽかぽかアイルー村』(発売元:カプコン, 開発:FromSoftware, 2010)では、環境上のオブジェクトに行動ごとのアフォーダンス値が設定されている [並木 11]。

§3 アニメーション補助データ

例えばキャラクターを椅子に座らせるときに、「座る」アニメーションが自然に見えるためには、椅子の座る側の方向、椅子の高さなどアニメーション再生のヒントとなる情報が必要である。ゲームの世界にはさまざまな椅子があり、そのすべてに対して固有のアニメーションをつくることはできないので、アニメーションを椅子に合わせて微調整するための情報がオブジェクト側に必要である。例えば樵が木を切るのであれば適切な位置を木がもち、露店で列をつくるなら適切な場所と手を伸ばして受け取る場所の情報をもつ。『Bioshock : Infinite』(Irrational Games, 2013)では各オブジェクトが座る場所、見上げる場所、立ち止まる場所など補助データをもち、さらに進んでオブジェクトの側に来るとオブジェクトがキャラクターを操作してオブジェクトの形状に沿った自然な振舞いをさせる。こういったキャラクターを制御する機能をもったオブジェクトはスマートオブジェクト、あるいはスマートテライン (Smart Object, Smart Terrain) と呼ばれる [Abercrombie 14]。

§4 敵の表現

キャラクターにとって敵は最も注意を向けるべき対象である。位置、種類、体力や魔力、性能(速度)、種族など、客観的な情報も必要であるが、キャラクターから主観的にその敵がどのように見えているか、という情報が意思決定のために必要である。具体的には、上記の客観的なパラメータを主観的なパラメータに変更することで行われる。敵の性能や位置、速度から自分にとって敵がどれだけ危険であるかを示す「脅威度」は、評価式をつくって算出する [Isla 09]。この脅威度の高さによって複数の敵からメインターゲットを選ぶことができる。また位置取り、速度・加速度ベクトル、履歴から敵の「意図」を推定する。「攻撃」、「撤退」、「待伏せ」などである。この推定された「意図」を意思決定で用いることで、相手を予測して行動する知能をつくる。このような敵に

対する認知はアウェアネス (Awareness) と呼ばれる [Rabin13]。

4.3 事実表現

ゲーム内で起こる事象についての表現である。表現がもつ情報は「誰が、いつ、どこで、何に対して、どのように、何をしたか」という 5W1H の事実、そして、その情報をどのような経路で入手したか (自分で目撃したか、伝聞か、など)、最後にその情報の信頼度である。

『Gunslinger』(未発売, Surreal, 2002)という西部劇をテーマにしたゲームでは、事実管理システムが街中で起こる事件を管理しており、目撃した NPC の間でその事実が共有され、そこから NPC 同士の会話によって街中に事実が流布していく「うわさシステム」が実装されている [Alt 02, 三宅 10c]。

ナビゲーション AI はこのようにレベルデザインの構造と性質、そこで起こっていることを常に表現として準備しておく。キャラクター AI、メタ AI はその表現を必要に応じて引き出し、認識を形成し、意思決定を行う。

5. メタ AI のつくり方

メタ AI はゲームシステムに宿る人工知能である。メタ AI はリアルタイムにゲーム状態を監視し、ゲームコンテンツに関する調整・変化を行う。1980 年代におけるメタ AI は『ゼビウス』(ナムコ, 1983)をはじめとするプレーヤのスキルに応じてゲームの難易度を調整するレベル自動調整システムであった [岩谷 05a, 岩谷 05b]。さらに近年では、より積極的にゲームコンテンツを動的につくり出す役割をもっている [三宅 10a, 三宅 10b, 三宅 10c]。

1980 年代においてメタ AI とキャラクター AI の立場は曖昧であった。キャラクターを動かす AI は俯瞰的にレベルを眺めてキャラクターのみならずゲームギミック (仕掛け) などゲームのすべてを動かしていた。ところが 90 年代を経て AI は、環境を認識する機能をナビゲーション AI として分化・独立し、自律的な AI としてキャラクター AI が独立し、メタ AI は、キャラクターを動かす役割から解放され、ゲーム全体を動かす役割に特化することとなった。例えば、メタ AI は NPC 達の配置をコントロールするが、いったん配置した後はキャラクター AI の自律的思考に任せることができる [三宅 11, 三宅 12a]。

メタ AI がゲーム状態を認識する方法は、キャラクター同様にナビゲーション AI からの情報による。メタ AI が認識の対象とするのは、ゲーム全体の大局状態と、プレーヤの現在の状態とプレイ履歴である。プレイ履歴からはプレーヤのスキルと進行の達成度を認識し、グローバルな状態からは現在のプレーヤの周囲の状況と地形を認識する。プレーヤの履歴はデータ収集と解析によるが、ゲーム全体のグローバル情報はナビゲーションメッシュなど

ナビゲーションデータを用いて各メッシュ上の敵・味方キャラクタ・砲台などのギミックを監視することで収集する。ここでナビゲーションメッシュはメタ AI によってセンサの網のような役割し、複雑な地形とその上に展開するゲームの状態をメッシュ単位で認識するのである。ここでもナビゲーション AI の機能のサポートが必要とされる。

5.1 メタ AI 事例

4 人がチームを組んで敵 AI 群と戦うオンラインゲーム『LEFT 4 DEAD』（Valve Software, 2008）では、ナビゲーションメッシュとプレイヤーの履歴（経路、戦闘ログ）によってプレイヤーの経路を予測し、さらにコントローラ入力と戦績から（どれぐらいの敵をどれぐらいの時間で倒したか）からプレイヤーの緊張度を割り出したうえで、プレイヤーの予測経路上の脇にある視線の通っていない領域に動的に敵をスポーニング（生成）する。通常、ゲームでは敵の配位、数はあらかじめゲームデザイナーによって決められておりデータ化されているが、本ゲームではユーザのスキルと移動パターンに合わせるため、メタ AI が動的に状況をつくり出す [Booth 09a, Booth 09b]。

『LEFT 4 DEAD』のメタ AI は「ゲームの面白さとは緊張とその緩和の繰返しである」というゲームデザインの方針のもと、メタ AI によって人工的にゲームに緊張と緩和の波をつくり出すことを目的としている。そこでユーザのコントローラ入力と戦績からユーザの緊張度を評価する式をつくり、ユーザの緊張度を監視している。モンスターをユーザの予測経路上の周囲にスポーニング（生成）してユーザ達を襲わせることでユーザの緊張度を高め（ビルドアップフェーズ）、目標の緊張度をクリアした後は現状を維持し（維持フェーズ）、その後スポーニングを中止してモンスターを少なくすると緊張度が下落していき（フェードフェーズ）、ほぼ 0 になった時点で何も起こさないユーザがリラックスする時間をつくる（リラックスフェーズ）。リラックスした時間がある程度続くと、モンスターをスポーニングし始める（ビルドアップフェーズ）。この四つのフェーズのサイクルによって、緊張と緩和をメタ AI によってつくり出す。

また『Warframe』（Digital Extremes, 2013）ではメタ AI は『LEFT 4 DEAD』のメタ AI の機能に加えて「ダンジョンを自動生成する」、「作成したダンジョンを解析して構造を認識する」、「アイテムを自動配置する」、「ヒートマップによってユーザの移動を監視する」というプロセスを追加し、よりゲーム全体を生成・管理システムへ発展している [Brewer 13, Brewer 14]。

5.2 メタ AI のエージェントアーキテクチャ

メタ AI の内部構造はキャラクタ AI・チーム AI 同様にエージェントアーキテクチャとして構築する。センサとしてはゲームのあらゆる情報にアクセスが可能であ

り、エフェクタとしてはプレイヤーを除くゲームのあらゆる要素と全域に対してコントロールする力をもっている。プレイヤーキャラクタは『LEFT 4 DEAD』の緊張度の計算のように、その状態を推定することで対応する。これが意思決定の過程となる。

5.3 メタ AI の将来

人工知能技術が他のゲーム技術と比較して顕著な特徴は、ゲームデザイン、レベルデザインと密に接している分野であるということである。メタ AI はさらにゲームデザイン、レベルデザインそのものをつくり出す人工知能として機能する。

これまではゲームデザイナーが開発中に作成した固定データを再生していたゲームの仕組みを、メタ AI はゲーム状況に応じて動的にゲームデザイン、レベルデザインをつくり出す仕掛けに変更する。メタ AI はいわばゲームデザイナーの知能を埋め込んだ人工知能である。

メタ AI はキャラクタ AI、ナビゲーション AI の仕組みを援用するので、製作の後期において、比較的低コストで実現可能であり、にもかかわらず、その効果は大きい。またメタ AI のみを変更することで、同じ要素を使った全く違ったゲームシステムを展開することができる。例えば、軽量のメタ AI のみを追加ダウンロード配布することで、同じ要素を使った全く違ったゲームを配信す

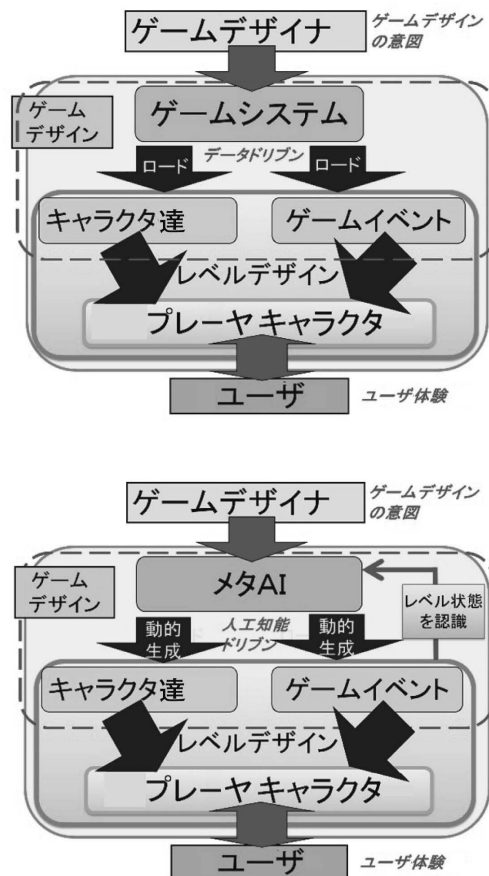


図 16 従来のゲームシステム（上）とメタ AI を用いたゲームシステム（下）の構造図

ることができる。

一般に、これまで固定化されていたコンテンツを動的に変化するコンテンツに変化させることをプロシージャル化(Procedural)という。この20年の間、ゲームが進化するに従い、知能のロジック、経路、地形、シナリオなど、これまで固定化されていたコンテンツがプロシージャル化してきた。メタAIは、これらのさまざまなゲーム要素のプロシージャル技術を使用し、ゲームそのものをプロシージャル化する。これはデジタルゲームの進化の自然な途上にある(図16)。

6. 三つのAIの組合せ方

デジタルゲームの目的はユーザにそのタイトル固有の体験を提供することである。まずキャラクタAIをつくることで、ユーザの周囲の至近距離の戦闘や直接的な短時間のインタラクションをつくっていく。次にナビゲーションAIによって、キャラクタがすべてのステージで地形に沿って動き、オブジェクトを利用した動きができるような仕組みをつくる。ナビゲーションAIによってキャラクタは長距離移動や回り込んで道具を使うなどができるので、ユーザ体験の空間スケールはずっと広がる。メタAIはプレーヤ、キャラクタ、レベルデザインの状態を監視し、微調整を加える、あるいはダイナミクスにゲームの進行を変動させる。これによって時間的な体験スケールがずっと広がる。

デジタルゲームのAI技術はこのようにキャラクタ、ゲーム全体をAI技術によって動かしながら、時間・空間のマルチスケールでユーザ体験をある程度自由にコントロールすることを可能にする(図17)。

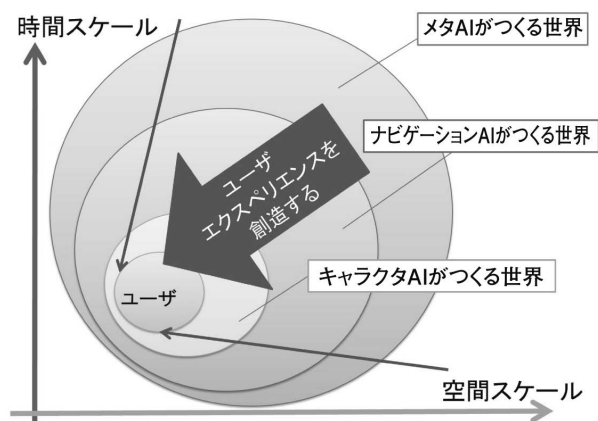


図17 三つのAIとユーザエクスペリエンス

1980～90年代には未成熟な一つのシステムであったものが、三つの分散人工知能「キャラクタAI」、「ナビゲーションAI」、「メタAI」に分化し、現代ではその三者の協調によって、デジタルゲームの人工知能はユーザの体験をある程度人工的にコントロールできる能力を得たのである。

7. 学習・進化・生成の手法

デジタルゲームはその黎明期から40年の間、基本的には、あらかじめ作成したデータを展開していくことでゲームを進行させてきた。しかし、そういった静的なゲームのつくり方は2000年代前半で一度ピークを迎え、これまでデータ固定化されていた部分を、計算によって置き換えるプロシージャル化が進んでいる。しかし、その中でも、学習・進化・生成の手法は、ソフトウェア構造としてのゲームの中心部分をデータから計算ダイナミクスで置き換えようとする試みであり、それ自身の導入の困難さに加えて、他のゲーム要素とのバランスを取ることが難しい。ここでは、少数ながら実例を踏まえた導入の現状を説明する。

7.1 キャラクタの学習・進化

キャラクタの学習・進化には、データのアプローチ、擬似学習、学習アルゴリズムの三つの方法がある。これを順次説明する。前者二つは通常技術的には「学習」と呼ばれないが、ゲーム演出の意味で「学習」に分類される。

§1 データ的アプローチ

データのアプローチとは、例えば、あるキャラクタのパラメータのセットを初期セット、第1段階セット、第2段階セット……最終セットのように用意しておき、ゲームの進行に合わせて入れ替えていくという方法である。これによって段階的ながらキャラクタの学習・進化を見せかけることができる。この方法はあらかじめゲームデザイナーが調整することができるので、最もよく使われる方法である。またデータセットではなくパラメータの成長曲線を用意しておく手法もよく使われる方法である。

§2 擬似学習

擬似学習は、あらかじめキャラクタにすべての能力を実装しておき、イベントが起こるたびに、一つ一つ解放して行く方法である。例えば、あるキャラクタに対しすべてのアクション技を実装しておくが、最初はすべて使用不可能にしておく。プレーヤと対戦して、プレーヤが技を使うたびに、キャラクタもその技を使えるようにする。すると、プレーヤから見るとあたかも自分の技をキャラクタが学習しているように見える。

§3 学習・進化アルゴリズム

学習・進化アルゴリズムのデジタルゲームへの応用は近年増加しつつあるが多くない。これまで学習・進化が応用された商業タイトルは通算でも200程度であり、現在でも年間20タイトル前後である。これまで、学習・進化アルゴリズムが応用されることが少なかった背景は四つあり、「計算リソースの不足」、「安定性への不安」、「ゲームデザインとのすり合わせの難しさ」、「学習環境の構築の限界」である。

計算リソースは、描画が重視された2000年以降の10

年間はAIに学習のための計算リソースを当てることは難しかったが、近年は多少解決されつつある。安定性の問題は、学習である以上、学習が失敗する可能性があり、その調整と不安が導入を困難にする。もし最初から学習が成功するのであれば、学習達成した後のパラメータを用意しておくデータのアプローチのほうが選択される。

そこで学習アルゴリズムが使用されるタイトルはあえて学習の過程を見せる必要があるタイトルに限られる。

ゲームデザインとの親和性はキャラクタを学習させる以上、繰返しのシミュレーションが前提であり、これをゲームデザインに組み込んでおかなければならないが、たいいていのゲームのキャラクタはそれぞれのステージで異なる役割が与えられており、同じ状況を繰返すことは少ない。

最後に学習環境の構築の問題であるが、この問題が最も大きい。学習・進化アルゴリズムは、それを機能させるために、パラメータセットを決め、学習アルゴリズムのチューニングをして、シミュレーション環境を整える必要がある。ディジタルゲームは開発しながら形を変えていくので、変化するゲームの中で学習・進化アルゴリズムが有効に機能するための環境構築は容易ではない。どのパラメータセットを選べばよいか、どうパラメータを規格化しておけばよいか、どの学習アルゴリズムを選択し、どのようにチューニングしておくか、試行錯誤を何度も繰返しても、ゲームデザインが変更されれば振出しに戻される。ゲームが完成してから導入すればよいかといえば、そのときにはもう入れる余地はなく、学習アルゴリズムはあらかじめゲームデザインとのすり合わせておく必要がある。

さらに、最も大きな問題は学習・進化アルゴリズムが適切に動作する条件領域は広くない点にある。適用条件が狭いと学習アルゴリズム自体がゲームデザインの自由を限定するという逆転現象が起こる。ディジタルゲームはゲーム自身をつくりながらAIをつくるので、この点は許容できないケースが多い。

このような問題をクリアできるタイトルはいくつかあり、またゲーム内でなく開発過程において応用するという事例も出てきている。以下、それらのタイトル事例を説明する。

7.2 1990年代の学習・進化アルゴリズムを使ったゲーム

1990年代に学習・進化アルゴリズムを使ったゲームは数例が知られているが、学習・進化アルゴリズムがゲームデザインの中心となるゲームが多い。これは計算リソースの多くが学習・進化アルゴリズムに消費されるために、人工知能をゲームデザインの中心にする必要があるからである。

例えば『Creatures』（Millennium Interactive, 1986）はモンスターを育成するゲームであり、一画面の中でモンスターにボールやさまざまな道具の使い方を教える。

モンスターはローブと呼ばれる数百個近いノードのパーセプトロン型ニューラルネットワークをもっており、ローブが数十組み合わせられる脳システムをもっている。このニューラルネットワークによってプレーヤからの教師信号を学習し、徐々に道具を正しく使えるようになっていく [Grand 97]。『Creatures Online』も同様のアプローチでつくられている [Hamaide 14]。

『アストロノーカ』（muumuu, 1998）は遺伝的アルゴリズムを応用してキャラクタを進化させるゲームである。プレーヤが畑でつくった野菜を食べにくるキャラクタを、プレーヤが途中の野原にさまざまな種類のトラップを仕掛けて撃退するというゲームであるが、プレーヤが新しい罠を仕掛けてもしばらくすると学習してトラップをかわすようになっていく。キャラクタの内部には遺伝子のパラメータセットがあり、そのパラメータに従ってプレーヤのトラップの中をどれぐらい畑の近くまでたどり着けたかによって評価点が決まる。上位の個体の遺伝子を掛け合わせていくことでキャラクタは学習・進化する。実際は画面に見えている一体のシミュレーション以外の20体のシミュレーションをバックグラウンドで動かして進化のスピードを高速化しており、プレーヤに体感できる進化スピードになるようにデザインされている [森川 99, 森川 08]。

7.3 2000年以降の学習・進化アルゴリズムを使ったゲーム

『Black&White』（Lionhead Studios, 2001）はキャラクタの学習のために簡単なパーセプトロン型ニューラルネットワークが使用している。1ノードだけの簡単な分岐ノードである [Evans 02]。

『Supreme Commander 2』（Gas Powered Games, 2010）では、敵NPCが入力層34ノード、中間層98ノード、出力層15ノードのパーセプトロンをもち、周囲のレベル状況を入力として出力として攻撃ターゲットの選択を行う。学習は開発中に行い、ゲームリリースするときはニューラルネットの結合率は固定される [Robbins12]。

『Forza Motorsport』（Microsoft Game Studios, Turn 10 Studios, 2004）ではプレーヤのドライビングの特性を機械学習によってAIが学習する。このAIはDrivatar [Microsoft 14] と呼ばれる。レースコースにはブロックごとにあらかじめ理想的なコースラインが設定されている。またハンドルとアクセル、ブレーキをタイヤの向きや回転と連動するリアルなドライブシミュレーションモデルが実装されている。Drivatarは理想的なコースラインと実際にプレーヤが走ったラインの差を検出し、これをハンドルさばきとアクセルとブレーキの踏みタイミングの特性まで還元して学習する。例えば、急カーブではどれぐらい手前からブレーキを踏み、どれぐらいハンドルを回すか、どのタイミングでアクセルを踏みか、などの数値を割り出しこれを記憶することで学習

する。Drivatar はコースのさまざまな場所で学習することでプレイヤーのドライビングの癖を蓄積する。プレイヤーは学習させた自分の Drivatar と対戦することができる。またサーバにアップロードすることができ、他のユーザーの Drivatar をダウンロードして対戦することもできる [Herbrich 08]。

『Hitman Absolution』ではキャラクタアニメーションの遷移分岐を強化学習によって決定する [Büttner 12]。『Hitman Absolution』のアニメーションシステムはモーショングラフという手法で各アニメーション同士の遷移ポイントが定義されている。ある速度と体勢から、別の速度と体勢へ遷移するときには、複数のアニメーション遷移ルートが考えられる。どのような遷移ルートを選択すれば自然な振舞いになるかは自明ではない。そこで速度と姿勢の変化分をコストとして、それが最少となる遷移ルートを強化学習しておく。

『NERO』(テキサス大学 Neural Networks Research Group, 2005) は大学の研究でつくられたゲームであるが、エージェントの意思決定部分にニューラルネットワークと遺伝的アルゴリズムが組み合わされた「Neuro Evolution」という技術が組み込まれている [Stanley 05]。遺伝子の部分にはニューラルネットワークの結合トポロジーが記述されており、二つの遺伝子を掛け合わせることで、新しいニューラルネットワークのトポロジーが出現する [Buckland 02]。『NERO』の中では勝ち残ったエージェント同士が遺伝的アルゴリズムによってニューラルネットワークを進化させるのである。この手法はアカデミックな研究においてさまざまな応用の広がりを見せている [NERO 05]。

7.4 プロシージャル技術

ゲームコンテンツは基本的にはデータを用意し、ゲーム実行中にそれらのデータを駆使してゲームを構築するが、プロシージャル技術とは、コンテンツをプログラムによって自動生成する技術である [三宅 10b, 三宅 10c]。

地形のプロシージャル技術とは、フラクタルやランダム関数を用いて計算によって地形を自動生成することである。キャラクタ配置のプロシージャル技術とは、計算によって状況に適切な場所にキャラクタを動的には位置することである。シナリオ自動生成とは、ドラマの分岐を計算によって動的に切り替えることである。

プロシージャル技術は、アセット作成のコストと分量を軽減し、データサイズを小型化する。しかしクオリティはデザイナーが作成したコンテンツに及ばないので、大量のアセットが必要な場合や、背景モデル、サイドストーリーなどの生成に応用される。また完全にアルゴリズムで生成するとアルゴリズムのパターンが顕著になるため、ある程度の元データから生成する手法をセミプロシージャル技術 (semi-procedural) という。例えば、プロシージャルアニメーションはほとんどの場合、元のアニメー

ションからリターゲットイングによって生成する [Hecker 08, 倉地 09a]。

プロシージャル技術はゲーム産業では AI 技術の一部に分類する場合が多く、実際、メタ AI と密接な関係をもつ。プロシージャル技術の起源は『Rogue』(1980) [Toy 80] の自動生成ダンジョンにあり、その後『ELITE』(BBC, 1984) 『風来のシレン』(チュンソフト, 1995) のステージ生成など多数の応用事例が見られる [Braben 11]。『Age of Empire』(Ensemble Studios, 1997) シリーズでは地形が対戦ごとに自動生成されている [Pottinger 00]。『FarCry2』(Ubisoft, 2008) では島全体の植物が自動生成される [Guay 08]。『Spore』(Maxis, 2008) は地形、アニメーション、テクスチャ、音楽など全般にわたるプロシージャル技術の集合である [三宅 08c, 倉地 09b]。近年では、『DUNIA ENGINE』(Ubisoft, 2008), 『Frostbite Engine』(EA DICE, 2008) のようにゲームエンジンの機能として、地形生成や植物生成・分布などが実装されるケースが増えている [Widmark 12]。

ゲーム開発後半へいけばいくほど、積み上げたデータの連関のために、修正や追加が難しくなっている。このゲーム開発後期の硬化現象を緩和する手段として大規模開発ほどプロシージャル技術が用いられるようになった。プロシージャル技術はプログラムによる生成のため開発後期においても比較的修正・変更の柔軟性を維持することができる。また次タイトルでも再利用が可能である。

7.5 ゲーム進化技術

プロシージャルな生成によって、ステージ、敵配置、音楽、などデジタルゲーム自身を生成することができる。コンテンツを自動生成することはプロシージャルコンテンツジェネレーション (PCG: Procedural Contents Generation) と呼ばれる。ゲームデザインを自動的に AI にプレイさせながら評価しつくり出すことは自動ゲームデザイン (Automatic (Automated) Game Design) と呼ばれる [Hom 07, Nelson 07, Nelson 12, Togelius 08]。

§1 進化型ゲームデザイン

自動ゲームデザインの問題はそのクオリティであって、現在のところ人間が一つ一つのコンテンツを作成したほうが高い。それでもクオリティを引き上げる方法が存在する。その方法は「自動生成したゲームを AI によってプレイし評価させ進化アルゴリズムによって自動的にゲームデザインを進化させる」という手法である。これは「進化型ゲームデザイン」(Evolutionary Game Design) と呼ばれる [Nelson 12]。進化アルゴリズムとしては主に遺伝的アルゴリズムか遺伝的プログラミングが用いられる。

プロシージャルに地形を生成した場合『Age of Empire』では、コネクティビティチェックを行う。これは生成したマップ上でスタートからゴールまでキャラクタが

たどり着けるか保証を経路検索によってチェックするものである。これは最低限のゲーム成立の保証である [Pottinger 00]。

さらに進んで、プロシージャルに生成したレベルデザインを AI にプレイさせることで評価し、その評価値をもとに進化アルゴリズムによってゲームを進化させるという方法が「進化型ゲームデザイン」である。例えば、横スクロールのアクションゲームのレベルをプロシージャル生成し AI によってプレイさせることで、クリア可能であるか、得点、クリアタイム、ジャンプ回数、スピード変化、倒した敵の数などによって、そのステージがどれくらいよくできているかを評価し、総合的な評価値を導く。さまざまな生成アルゴリズムや調整パラメータを変化させながら、その評価値が高くなるほうへ進化させていく [Dahlskog 14, Shaker 12, Shaker 13]。

§2 自動進化するボードゲーム

進化型ゲームデザインを実装する場合に、重要なのは「ゲームの表現」である。その表現ができれば、それぞれの要素を変化させることで、ゲームデザインを変化させることができる。例えば、ボードゲームの進化型ゲームデザインは、ボード表現、ゲームルール（駒の動かし方、終了条件）に分類される。それぞれの表現を列挙したデータとしてもち、遺伝的アルゴリズム、遺伝的プログラミングでクロスオーバーさせながら進化させ、AI によるプレイによって評価値をフィードバックすることで、ボードゲームデザインを自動進化させる。結果として新しいボードの形の上に、新しいゲームのルールが生み出される [Browe 10, Pell 92]。

8. ま と め

ユーザはゲームプレイ時間のほとんどの時間をキャラクタとのインタラクションに費やしている。本稿で述べたようにユーザエクスペリエンスの微妙な機微を調整できるのは人工知能であり、その進化はユーザに新しい体験をもたらすのである。よって、ディジタルゲームの人工知能技術のレベルが上がるということは、ゲームデザインに進化を促すことなのである。

ゲームにおける人工知能技術は、ゲームデザインとともに説明される必要がある。本稿ではディジタルゲームの技術のみならず、それがどのようにゲームデザインと関わっているかを述べた。

またゲーム産業における人工知能技術は開発工程の一部である。将棋や囲碁のように初めから完成したゲームがあるわけではなく、ゲーム自身をつくりながら、そのタイトルに必要な、そのタイトルをより良くする、あるいは、そのタイトルの中核となる人工知能技術を開発し組み込まなければならない。人工知能技術によってゲームデザインが変化することもあれば、ゲームデザインの要求に応えようとして人工知能技術も進化する。

また人工知能の開発をいかにゲーム全体の開発パイプラインに組み込むか、という課題もゲーム産業では大きな課題である。例えば、ナビゲーショndataはキャラクタ AI 製作の中で最大容量のデータであるが、これを製作するためにはツールパイプラインを整備しどのタイミングで、どのデータを組み合わせてビルドするかを決定し、開発フローを整備しなければならない。

ゲーム産業における人工知能技術はこの 20 年で進歩を遂げ、ようやく大きな概形が決まろうとしている。その進化はある程度、アカデミックな人工知能研究の流れと独立した流れであったが、ようやくアカデミックな研究の成果を取り込む土台が出来上がった。また同時に新しい研究テーマをアカデミック側に投げる体制が整いつつある。今後はゲーム産業とアカデミックな研究の協調体制がお互いの進化を加速していくと期待している。

9. ゲーム産業における AI の情報源

ゲーム産業における人工知能の開発事例・技術情報は、論文になっているケースが少なく、ゲーム産業カンファレンス (GDC (米)、CEDEC (日本)、Game AI Conference (欧州) など) の講演資料が公開ライブラリとして公開されているか、各企業の技術公開サイト、ゲーム産業 AI 専用サイト (AiGameDev) で公開されている。書籍としてはゲーム開発者の技術解説記事の寄稿を集めた形で刊行されている。情報は収集し難い状況にあるため、以下にゲーム産業における人工知能技術の主な情報源を列挙する。

●GDC (Game Developers Conference) Vault

<http://www.gdcvault.com/>

●CEDEC Library

<http://cedil.cesa.or.jp/>

●Game AI Conference 資料公開サイト

<http://gameaiconf.com/recordings/>

●AAAI AIIDE (Artificial Intelligence and Interactive Digital Entertainment)

<http://www.aiide.org>

●AiGameDev

<http://aigamedev.com/>

●Valve Software

<http://www.valvesoftware.com/company/publications.html>

●Guerrilla Games

<http://www.guerrilla-games.com/publications.html>

●Naimad Games

<http://naimaãames.com/publications.html>

●Bungie

<http://halo.bungie.net/inside/publications.aspx>

- AI Game Programming Wisdom 1-4
Steve Rabin 編, 1-2,4 Charles River Media
3 Cengage Learning (2002 ~ 08)

- Game AI PRO
Steve Rabin 編, A K Peters/CRC Press (2013)

- Game Programming Gems 1-8
ボーンデジタル (2001 ~ 11)

またテーマごとに参考文献をまとめた資料 [三宅 12b]
も参照されたい。

◇ 参 考 文 献 ◇

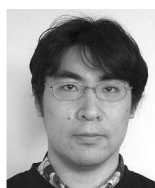
- [Abercrombie 14] Abercrombie, J.: *Bringing BioShock Infinite's Elizabeth to Life: An AI Development Postmortem*, GDC (2014). <http://www.gdcvault.com/play/1015387/Animation-Driven-Locomotion-for-Smoother>
- [Alt 02] Alt, G. and King, K.: A dynamic reputation system based on event knowledge, *AI Game Programming Wisdom*, Vol. 1, No. 8.6, pp. 426-435 (2002)
- [安藤 10] 安藤 毅:「サカつく」のサッカー試合 AI システム, CEDEC (2010). <http://cedil.cesa.or.jp/session/detail/379>
- [Anguelov 12] Anguelov, B., Leblanc, G. and Harris, S.: *Animation-Driven Locomotion For Smoother Navigation*, GDC (2012). <http://www.gdcvault.com/play/1015387/Animation-Driven-Locomotion-for-Smoother>
- [Anguelov 13] Anguelov, B.: *Managing the Movement*, GDC (2013). <http://www.gdcvault.com/play/1018145/Managing-the-Movement-Getting-Your>
- [Bernstein 03] Bernstein, N. A. 著, 工藤和俊 翻訳: デクステリティ 巧みさとその発達, 金子書房 (2003)
- [Booth 09a] Booth, M.: *Replayable Cooperative Game Design: Left 4 Dead*, GDC (2009). <http://www.valvesoftware.com/company/publications.html>
- [Booth 09b] Booth, M.: The AI Systems of Left 4 Dead, *AIIDE* (2009). <http://www.valvesoftware.com/company/publications.html>
- [Braben 11] Braben, D.: *Classic Game Postmortem — ELITE*, GDC (2011). <http://www.gdcvault.com/play/1014807/Classic-Game-Postmortem>
- [Brewer 13] Brewer, D.: *AI Postmortems: Assassin's Creed III, XCOM: Enemy Unknown, and Warframe*, GDC (2013). <http://www.gdcvault.com/play/1018223/AI-Postmortems-Assassin-s-Creed>
- [Brewer 14] Brewer, D.: The living AI in warframe's procedural space ships, *Game AI Conference* (2014). <http://gameaiconf.com/recording/living-ai-of-warframe/>
- [Brooks 86] Brooks, R.: A robust layered control system for a mobile robot, *IEEE J. Robotics and Automation*, Vol. 2, No. 1, pp. 14-23 (1986)
- [Browe 10] Browne, C. and Maire, F.: Evolutionary game design, *IEEE Trans. on Computational Intelligence and AI in Games*, Vol. 2, No. 1, pp. 1-16 (2010)
- [Buckland 02] Buckland, M.: *Evolving Neural Net Topologies, AI Techniques for Game Programming*, Cengage Learning PTR (2002)
- [Buckland 04a] Buckland, M.: *Goal-Driven Agent Behavior*, Chapter 9, *Programming Game AI by Example*, Wordware (2004)
- [Buckland 04b] Buckland, M.: *Practical Path Planning*, Chapter 8, *Programming Game AI by Example*, Wordware (2004)
- [Buckland 04c] Buckland, M.: *The Secret Life of Graphs*, Chapter 5, *Programming Game AI by Example*, Wordware (2004)
- [Burke 01] Burke, R., Isla, D., Downie, M., Ivanov, Y. and Blumberg, B.: Creature smarts: The art and architecture of a virtual brain, *Proc. Game Developers Conf.*, pp.147-166 (2001)
- [Canary 14] Canary, A.: *Free-Range AI*, GDC (2014). <http://www.gdcvault.com/play/1020110/Free-Range-AI-Creating-Compelling>
- [Champandard 08] Champandard, A. J.: *On the AI Strategy for KILLZONE 2's Multiplayer Bots*, AiGameDev (2008). <http://aigamedev.com/open/coverage/killzone2/>
- [Dahlskog 14] Dahlskog, S. and Togelius, J.: A multi-level level generator, *Proc. IEEE Conf. on Computational Intelligence and Games (CIG)* (2014)
- [Dawe 13] Dawe, M.: Beyond the Kung-Fu circle, *Game AI Pro*, Chapter 28, pp. 369-375 (2013)
- [Evans 02] Evans, R.: Varieties of learning, *AI Wisdom*, Vol. 1, No. 11.2, pp. 567-578 (2002)
- [Evans 10] Evans, R.: *Modeling Individual Personalities in The Sims 3*, GDC (2010). <http://www.gdcvault.com/play/1012450/Modeling-Individual-Personalities-in-The>
- [Gravot 12a] Gravot, F., Yokoyama, T. and Miyake, Y.: バス移動 (動画) (2012). https://www.youtube.com/watch?v=9_vdpndn4jI
- [Gravot 12b] Gravot, F., Yokoyama, T. and Miyake, Y.: 落下移動 (動画) (2012). <https://www.youtube.com/watch?v=o3ihN2kSzMs>
- [Gravot 13] Gravot, F., Yokoyama, T. and Miyake, Y.: Precomputed pathfinding for large and detailed worlds on MMO servers, *Game AI Pro*, Chapter 20, pp. 269-287 (2013)
- [Fauerby 12] Fauerby, K.: *Crowds in Hitman: Absolution*, GDC (2012). <http://www.gdcvault.com/play/1015315/Crowds-in-Hitman>
- [Forbus 02] Forbus, K.: *Simulation and Modeling: Under the hood of the Sims*, Northwestern University, Lecture note (2002). http://www.cs.northwestern.edu/%7Efforbus/c95-gd/lectures/The_Sims_Under_the_Hood_files/frame.htm
- [Gibson 86] Gibson, J. J. 著, 古崎 敬 訳: 生態学的視覚論, サイエンス社 (1986)
- [Grand 97] Grand, S. and Cliff, D.: Creatures: Entertainment software agents with artificial life, *Autonomous Agents and Multi-Agent Systems*, Vol. 1, pp. 39-57 (1997)
- [Gregory 09a] Gregory, J.: *Game Engine Architecture*, A K Peters/CRC Press (2009)
- [Gregory 09b] Gregory, J.: *State-Based Scripting in Uncharted 2: Among Thieves*, GDC (2009). <http://www.gdcvault.com/play/1730/State-Based-Scripting-in-UNCHARTED>
- [Griesemer 02] Griesemer, J.: The illusion of intelligence: The integration of AI and level design in Halo, *Proc. Archive, GDC 2002* (2002). <http://halo.bungie.net/inside/publications.aspx>
- [Guay 08] Guay, D.: *Procedural Data Generation in FAR CRY 2*, GDC (2008)
- [Guay 12] Guay, J.-F.: *Real-time Sound Propagation in Video Games*, GDC (2012). <http://gdcvault.com/play/1015492/Real-time-Sound-Propagation-in>
- [Hamaide 14] Hamaide, J.: Creatures 3 Online, *Game AI Conf.* (2014). <http://gameaiconf.com/recording/2014-creatures/>
- [Hecker 08] Hecker, C., et al.: Real-time motion retargeting to highly varied user-created morphologies, *Proc. ACM SIGGRAPH 2008*, Vol. 27, Issue 3 (2008)
- [Hecker 09] Hecker, C.: *Spore Behavior Tree Docs* (2009). http://chrishecker.com/My_liner_notes_for_spore/Spore_Behavior_Tree_Docs
- [Herbrich 08] Herbrich, R., Graepel, T. and Candela, J. Q.: *Halo, Xbox Live The Magic of Research in Microsoft Products*, Microsoft Research (2008). <http://research.microsoft.com/en-us/projects/drivatar/ukstudentday.pptx>
- [Hom 07] Hom, V. and Marks, J.: Automatic design of balanced board games, *Proc. 3rd Artificial Intelligence and Interactive Digital Entertainment Conf.*, pp. 25-30 (2007)

- [石田 96] 石田 亨, 桑原和宏, 片桐恭弘: 黑板モデル, 分散人工知能, 第 4 章, コロナ社 (1996)
- [Isla 01] Isla, D., Burke, R., Downie, M. and Blumberg, B.: A layered brain architecture for synthetic creatures, *Proc. IJCAI* (2001)
- [Isla 02] Isla, D. and Blumberg, B.: Blackboard architectures, *AI Game Programming Wisdom*, Vol. 1, No. 7.1, pp. 333-344 (2002)
- [Isla 05a] Isla, D.: Managing complexity in the Halo2 AI, *Game Developer's Conf. Proceedings* (2005), http://www.gamasutra.com/view/feature/130663/gdc_2005_proceeding_handling_.php
- [Isla 05b] Isla, D.: Dude, where's my Warthog? From pathfinding to general spatial competence, *AIIDE* (2005). <http://naimaames.com/publications.html>
- [Isla 08] Isla, D.: Halo 3 objective trees: A declarative approach to multi-agent coordination, *Artificial Intelligence and Interactive Digital Entertainment (AIIDE)* (2008). <http://naimaames.com/publications.html>
- [Isla 09] Isla, D. and Gorniak, P.: Beyond Behavior: An Introduction to Knowledge Representation, *AI Summit*, GDC (2009). <http://naimaames.com/publications.html>
- [岩谷 05a] 岩谷 徹: バックマンのゲーム学入門, p. 101, エンターブレイン (2005)
- [岩谷 05b] 岩谷 徹: セルフゲームコントロールシステム, 「バックマン」岩谷氏, 「Rez」水口氏ら 4 人のクリエイターが語る世界のゲームデザイン論「International Game Designers Panel」(講演記事) (2005). http://game.watch.impress.co.jp/docs/20050312/gdc_int.htm
- [倉地 09a] 倉地紀子: Real-Time Motion Retargeting 『Spore』のモーションリターゲット・パイプライン, *CGWORLD*, Vol. 125, pp. 80-85 (2009)
- [倉地 09b] 倉地紀子: プロシージャル技術の先にあるもの, *CGWORLD*, Vol. 125, pp. 44-47 (2009)
- [Leeuw 09] Leeuw, M. van der: *The PlayStation®3's SPUs in the Real World: A KILLZONE 2 Case Study*, GDC (2009). <http://www.guerrilla-games.com/publications.html>
- [Microsoft 14] Microsoft Research: Drivatar™, *Forza Motorsport* (2014). <http://research.microsoft.com/en-us/projects/drivatar/forza.aspx>
- [MIT 03] MIT Synthetic Character Group (2003). <http://characters.media.mit.edu/>
- [三宅 06] 三宅陽一郎: クロムハウズにおける人工知能開発から見るゲーム AI の展望, *CEDEC* (2006). <http://cedil.cesa.or.jp/session/detail/50>
- [三宅 07a] 三宅陽一郎: ゲーム AI 分野, デジタルコンテンツ制作の先端技術応用に関する調査研究報告書, 第 3 章, デジタルコンテンツ協会 (2007). http://www.dcaj.or.jp/project/report/pdf/2007/dc08_07.pdf
- [三宅 07b] 三宅陽一郎: コンテンツ管理技術, デジタルコンテンツ制作の先端技術応用に関する調査研究報告書, 第 5 章, デジタルコンテンツ協会 (2007)
- [三宅 08a] 三宅陽一郎: デジタルゲームにおける人工知能技術の応用, 人工知能学会誌, Vol. 23, No. 1, pp. 44-51 (2008)
- [三宅 08b] 三宅陽一郎, 横山 貴規, 北崎 雄之: エージェント・アーキテクチャに基づくキャラクター AI の実装, デジタルコンテンツシンポジウム第 4 回, 予稿集, 2-2 (2008)
- [三宅 08c] 三宅陽一郎: Spore におけるゲーム AI 技術とプロシージャル, 日本デジタルゲーム学会, 第 14 回月例研究会 (講演録) (2008). <http://digrajapan.org/?wpdmact=process&id=Ni5ob3RsaW5r>
- [三宅 10a] 三宅陽一郎: デジタルゲーム AI, デジタルゲームの教科書 (ソフトバンク パブリッシング), 第 23 章, pp. 214-218 (2010)
- [三宅 10b] 三宅陽一郎: プロシージャル技術, デジタルゲームの教科書 (ソフトバンク パブリッシング), 第 22 章, pp. 395-420 (2010)
- [三宅 10c] 三宅陽一郎: オンラインゲームにおける人工知能・プロシージャル技術の応用, 知能と情報: 日本知能情報フェイジ学会誌, Vol. 22, No. 6, pp. 745-756 (2010)
- [三宅 11] 三宅陽一郎: これからデジタルゲームの AI の進む道を
知るために知っておきたいこと, デジタルゲームの技術, 第 5 章, pp. 127-214, ソフトバンク クリエイティブ (2011)
- [三宅 12a] 三宅陽一郎: はじめてのゲーム AI ~意思を持つかのように行動するしくみ~, *WEB+DB PRESS*, Vol. 68, pp. 87-120, 技術評論社 (2012)
- [三宅 12b] 三宅陽一郎: 次世代デジタルゲームにおける人工知能の研究課題について, ゲームプログラミングワークショップ 2012 論文集, Vol. 2012, No. 6, pp. 108-113 (2012)
- [三宅 13] 三宅陽一郎: デジタルゲームのための人工知能の基礎理論, 日本バーチャルリアリティ学会誌, Vol. 18, No. 3, pp. 28-33 (2013)
- [三宅 14] 三宅陽一郎: デジタルゲームにおける人工知能エンジン, 映像情報メディア学会誌, Vol. 68, No. 2, pp. 125-130 (2014)
- [森川 99] 森川幸人: テレビゲームへの人工知能技術の利用, 人工知能学会誌, Vol. 14, No. 2, pp. 214-218 (1999)
- [森川 08] 森川幸人: ゲームと AI はホントに相性がいいのか?, *CEDEC* (2008). <http://cedil.cesa.or.jp/session/detail/156>
- [並木 11] 並木幸介: ぽかぽかアイルー村における, アフォーダンス指向の AI 事例, AI に多様な振る舞いをさせる手法, *CEDEC* (2011). <http://cedil.cesa.or.jp/session/detail/697>
- [Nau 03] Nau, D., et al.: SHOP2: An HTN planning system, *J. Artificial Intelligence Research*, Vol. 20, pp. 379-404 (2003)
- [Nelson 07] Nelson, M. J. and Mateas, M.: Towards automated game design, In *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, pp. 626-637, Lecture Notes in Computer Science 4733 (2007)
- [Nelson 12] Nelson, M. J.: Encoding and generating videogame mechanics, *Computational Intelligence and Games 2012 tutorial* (2012). http://www.kmjn.org/notes/generating_mechanics_bibliography.html
- [NERO 05] Neuro-Evolving Robotic Operatives (2005). <http://nerogame.org>
- [Nii 86a] Nii, H. P.: The blackboard model of problem solving and the evolution of blackboard architectures, *AI Magazine*, Vol. 7, No. 2, pp. 38-53 (1986). <http://www.aaai.org/ojs/index.php/aimagazine/article/view/537>
- [Nii 86b] Nii, H. P.: Blackboard application systems, blackboard systems and a knowledge engineering perspective, *AI Magazine*, Vol. 7, No. 3, pp. 82-107 (1986). <http://www.aaai.org/ojs/index.php/aimagazine/article/view/550>
- [岡村 11a] 岡村信幸: ARMORED CORE V の階層型ゴール指向プランニングと機体制御, *CEDEC* (2011). <http://cedil.cesa.or.jp/session/detail/591>
- [岡村 11b] 岡村信幸: ARMORED CORE V のパス検索, *CEDEC* (2011). <http://cedil.cesa.or.jp/session/detail/593>
- [Orkin 03] Orkin, J.: Applying goal-oriented planning for games, *AI Game Programming Wisdom*, Vol. 2, No. 3.4, pp. 217-227 (2003)
- [Orkin 05a] Orkin, J.: 3 States & a Plan: The AI of F.E.A.R., *Game Developer's Conf. Proc.* (2006). <http://web.media.mit.edu/~jorkin/>
- [Orkin 05b] Orkin, J.: Agent architecture considerations for real-time planning in games, *AIIDE 2005* (2005). <http://web.media.mit.edu/~jorkin/>
- [Pell 92] Pell, B.: METAGAME in symmetric chess-like games, In *Heuristic Programming in Artificial Intelligence 3 — The Third Computer Olympiad*, Ellis Horwood (1992)
- [Pillosu 09] Pillosu, R.: Coordinating agents with behavior trees, *Paris Game Conference* (2009). <http://aigamedev.com/open/coverage/paris09-report/#session3>
- [Pottinger 00] Pottinger, D.: *Terrain Analysis for Realtime Strategy Games*, GDC (2000)
- [Rabin 13] Rabin, S.: Agent awareness and knowledge representation, *Game AI Pro*, CRC Press (2013)
- [Robbins 12] Robbins, M.: Using neural networks to control agent threat response, *Game AI Pro*, Chapter 30, pp. 391-399 (2013)

- [Russel 09] Russell, S. and Norvig, P.: *Artificial Intelligence: A Modern Approach*, Third Edition, Prentice Hall (2009)
- [Shaker 12] Shaker, N., Yannakakis, G. N. and Togelius, J.: Digging deeper into platform game level design, *Proc. European Conf. on Applications of Evolutionary Computation (EvoGames)* (2012)
- [Shaker 13] Shaker, M., Sarhan, M. H., Naameh, O. A., Shaker, N. and Togelius, J.: Automatic generation and analysis of physics-based puzzle games, *Proc. IEEE Conf. on Computational Intelligence and Games* (2013)
- [Snook 01] Snook, G.: ナビゲーション メッシュによる 3D 移動とパス発見の単純化, *Game Programming Gems*, Vol. 1, No. 3.6, pp. 279-294 (2001)
- [Stanley 05] Stanley, K. O., Bryant, B. D. and Miikkulainen, R.: Evolving neural network agents in the nero video game, *Proc. IEEE 2005 Symposium on Computational Intelligence and Games* (2005)
- [Sterren 13] Sterren, W. van der: Hierarchical plan-space planning for multi-unit combat maneuvers, *Game AI Pro*, Vol. 1, Chapter 13, pp. 169-183 (2013)
- [Straatman 05] Straatman, R., Beij, A. and Sterren, W.V.D.: *Killzone's AI: Dynamic Procedural Combat Tactics* (2005). http://www.cgf-ai.com/docs/straatman_remco_killzone_ai.pdf
- [Straatman 09] Straatman, R., Verweij, T. and Champandard, A.: Killzone 2 multiplayer bots, *Paris Game/AI Conference* (2009). <http://www.guerrilla-games.com/publications.html>
- [Straatman 13] Straatman, R., Verweij, T., Champandard, A., Morcus, R. and Kleve, H.: Hierarchical AI for multiplayer bots in Killzone 3, *Game AI Pro*, Chapter 29, pp. 377-390 (2013)
- [田邊 09] 田邊雅彦: WCCF AI エンジニアリング サッカーゲーム AI の設計と実装, 日本デジタルゲーム学会, 公開講座 2009 年 5 月期 (2009). <http://digraJapan.org/?wpdmact=process&did=OS5ob3RsaW5r>
- [Togelius 08] Togelius, J. and Schmidhuber, J.: An experiment in automatic game design, *Proc. 2008 IEEE Conf. on Computational Intelligence and Games*, pp. 111-118 (2008)
- [Toy 80] *UNIX Free Software Developed by Michael Toy*, Glenn Wichman, Ken Arnold (1980)
- [ユクスキュル 05] ヤコブ・フォン・ユクスキュル: 生物から見た世界, 岩波文庫 (2005)
- [米田 12] 米田 聡: [SQEXOC 2012] FF XIV で使われている AI 技術~敵 NPC はどうやって経路を探索しているのか?, 4gamers (講演記事) (2012). <http://www.4gamer.net/games/032/G003263/20121205079/>
- [Walsh 10] Walsh, M.: *Dynamic Navmesh - AI in the Dynamic Environment of Splinter Cell: Conviction*, GDC (2010). <http://www.gdcvault.com/play/1012651/Dynamic-Navmesh-AI-in-the>
- [Walsh 14] Walsh, M.: *Modeling AI Perception and Awareness in Splinter Cell: Blacklist*, GDC (2014). <http://www.gdcvault.com/play/1020195/Modeling-AI-Perception-and-Awareness>
- [Widmark 12] Widmark, M.: *Terrain in Battlefield 3: Modern, Complete and Scalable System*, GDC (2012). <http://www.gdcvault.com/play/1015414/Terrain-in-Battlefield-3-A>

2014 年 11 月 17 日 受理

著者紹介



三宅 陽一郎 (正会員)

株式会社スクウェア・エニックス テクノロジー推進部リード AI リサーチャー。1999 年京都大学総合人間学部基礎科学科卒業。2001 年大阪大学大学院理学研究科修士課程物理学専攻修了。2004 年東京大学大学院工学系研究科博士課程 (単位取得満期退学)。同年、株式会社フロム・ソフトウェア入社。2011 年退職。同年、株式会社スクウェア・エニックス入社、

現職。日本デジタルゲーム学会理事、国際ゲーム開発者協会日本ゲーム AI 専門部会代表、CEDEC アドバイザリーボード。共著『デジタルゲームの教科書』(ソフトバンククリエイティブ, 2010), 『デジタルゲームの技術』(ソフトバンククリエイティブ, 2011), 翻訳監修『ゲームプログラマのための C++』(マイケル・ディックハイザー 著, ソフトバンククリエイティブ, 2011), 『C++のための API デザイン』(マーティン・レディ 著, ソフトバンククリエイティブ, 2012), インタビュー「人工知能は数学を理解できるのか」(『考える人』2013 年夏号, 新潮社), 特集記事『はじめてのゲーム AI』(WEB + DB PRESS, Vol. 68, 技術評論社, 2012), デジタルコンテンツシンポジウム 第 4 回 船井賞受賞 (2008), CEDEC AWARDS 2010 プログラミング・開発環境部門優秀賞。日本デジタルゲーム学会 2011 年若手奨励賞受賞。デジタルゲームにおける人工知能技術の理論的確立と実際のゲームタイトルへの具体的導入に従事。