

# FINAL FANTASY XVの 人工知能

— ゲームAIから見える未来 —

### ■ ご注意

本書は著作権上の保護を受けています。論評目的の抜粋や引用を除いて、著作権者および出版社の承諾なしに複写することはできません。本書やその一部の複写作成は個人使用目的以外のいかなる理由であれ、著作権法違反になります。

### ■ 責任と保証の制限

本書の著者、編集者および出版社は、本書を作成するにあたり最大限の努力をしました。但し、本書の内容に関して明示、非明示に関わらず、いかなる保証も致しません。本書の内容、それによって得られた成果の利用に関して、または、その結果として生じた偶発的、間接的損傷に関して一切の責任を負いません。

### ■ 著作権と商標

本書に記載されている製品名、会社名は、それぞれ各社の商標または登録商標です。本書では、商標を所有する会社や組織の一覧を明示すること、または商標名を記載するたびに商標記号を挿入することは、特別な場合を除き行っていません。本書は、商標名を編集上の目的だけで使用しています。商標所有者の利益は厳守されており、商標の権利を侵害する意図はありません。

*FINAL FANTASY XV*の制作に携わったすべての人々

未来でゲームを作る人々へ

そして、何より

このゲームを好きになってくれたすべての方々へ

本書を捧げます

2019.3.15

## 目次

はじめに	8	動的な障害物（常に動く障害物）	40
<b>基礎編</b>	<b>12</b>	2. ステアリングシステムとは?	40
		概要	40
		LoD（レベル・オブ・ディテール）	41
		パス追従	42
		衝突回避	43
		グループ制御	45
		シミュレーションAIの情報を引き出す	46
		3. 未来へ向けて	46
<b>1章 ゲームAI入門</b>	<b>14</b>	<b>4章 AIとアニメーション</b>	<b>50</b>
1. 人工知能を導入したらゲームはどう変わるか?	15	1. ゲームにおけるアニメーション	51
2. キャラクターを賢くする方法	18	キャラクター アニメーション	51
3. ゲームデザインと人工知能	19	2. AIの出力としてのアニメーション	52
自律型AI	19	3. AIの入力としてのアニメーション	54
4. 現代のゲームAIの3つの柱	20	4. 感情の表現先としてのアニメーション	56
5. ゲームAIの歴史	21	フェイシャルアニメーション	56
<b>2章 意思決定ツール</b>	<b>22</b>	感情とその分類	57
1. 概要	23	フェイシャルアニメーションの技術	58
意思決定処理とは?	23	リップシンク	60
意思決定ツールとは?	23	目は口ほどに物を言う	61
FFXVに登場するAIキャラクター	24	5. FFXVにおけるAIとアニメーションの協調動作	61
FFXVに求められた意思決定ツール	26	エージェントアーキテクチャ	61
2. ツールの基本システム	28	エージェント内の各層のやりとり	66
複数の意思決定アルゴリズムを組み合わせる仕組み	28	6. AIとアニメーションの未来	66
身体・敵・ゲームとAIをつなぐブラックボード	30	複雑さとの戦い	66
3. ツールの応用的な機能	31	アニメーション制作におけるAIの利用	68
高度なAI記述を実現する仕組み	31	7. まとめ	69
AI作成のワークフローを支える仕組み	33		
4. まとめ	34	<b>5章 位置検索システム</b>	<b>72</b>
意思決定ツールがFFXVにもたらしたもの	34	位置検索システムとは?	73
FFXVのその先へ	35	1. 概要	74
<b>3章 ナビゲーションシステム</b>	<b>36</b>	2. ポイントを生成する	75
1. ナビゲーションシステムとは?	37	隠れポイント用ジェネレータ	75
静的な障害物	38	最小サンプリング密度	76
パス検索	38		
擬似静的な障害物（移動可能な障害物）	39		

3. クエリ	76	“生きた” AIと話そう：会話システム	119
フォールバック・クエリ	76	5. 見合わない挑戦	119
グループベースのクエリ	77		
予約システム	77		
4. 例：NPC 生成における PQS の使用	78		
5. 目標点と経路	79		
6. エクステナナル・エディター（External Editor）	79		
7. FFXV における PQS の統計	80		
8. 今後の活動	81		
9. まとめ	81		

## AI 座談会～プログラマー編～ 82

## コンテンツ編 92

## 6章 仲間 AI 94

1. 時代は CX (Character Experience)	95
ゲームは複合エンターテインメント	95
ゲームとキャラクター	95
「設定」を「体験」に	97
AI に物語らせよう	98
2. FFXV と仲間 AI	99
プレイヤーは 5 人目の仲間	99
3. 魅力的な仲間 AI の四大基礎	100
FUNCTIONAL：ゲームの役割をちゃんと果たす仲間	101
BELIEVABLE：実在感のある仲間	101
COMFORTABLE：一緒にいたい仲間	105
EMOTIONAL：心を揺さぶる仲間	110
FFXV の仲間のアイデンティティ	110
4. “最も近く、最も心地よい”を作り出す技術たち	112
切れない絆：フレンドリンク	112
“友達”の距離感：フォームーションシステム	114
ゲーム世界の神：メタ AI	116
“人間らしさ”の源：意識 AI	118

## 7章 モンスター AI 120

1. モンスターの役割	121
プレイヤーに対する障害（ストレス）	121
プレイヤーへの報酬（カタルシス）	121
プレイの目標	121
世界観を伝えプレイヤーを誘う	122
2. モンスターとのバトル	122
パターン・アクション型	122
プロレス型	122
バーリトゥード型	122
3. モンスターの制作方法	122
資料を読み込む	122
実際の目で見る	123
試行錯誤のイテレーション	123
センサーシステムの設計	123
ナビゲーションシステムの設計	124
攻撃の命中予測	125
知識表現とアフォーダンス	125
PQS の活用	125
勢力マトリックス	126
感情、本能、個性	126
乱れと秩序	126
マスプロダクションへの対応	127
テリトリー	127
4. モンスターの課題と未来	128
自律した生命の表現	128
機械学習と AI	128

## 8章 兵士 AI 130

概要	131
この章について	131
純粹なる敵	131

1. 基本編：戦う AI とは？	132	10 章 写真 AI	158
勝利する思考／移動する思考	132		
最強の兵士 AI	134		
2. 準備編：AI を組む	135	1. 最強のフィーチャー「写真 AI」	159
AI を組むとは？	135	異常なプレイヤーアクティブ率の秘訣	159
3. 実装編：FFXV の兵士 AI はどのように組まれたか？	136	プロンプトの写真機能が生まれるまで	160
ニフルハイム帝国のキャラクターパリエーション	136	2. 理想的な写真への要望	161
レイヤー AI：並列の思考	137	写真はゲームプレイを表現しなければならない	162
チーム AI：ワイヤー連携攻撃	138	写真は旅体験を豊かにしなければならない	163
最適な座標の決定	139	写真は個性と遊び心にあふれなければならない	164
4. 実践編：最強の AI を組む	140	要望を仕様に	165
AI を組むエディター	140	3. 写真 AI は様々な AI で成り立っている	165
隙のない AI を組む	141	写真 AI の流れ	165
反応の良い AI を組む	142	シャッター AI	166
5. 概念編：演技する AI	145	でっちあげシステム	168
最強から迎合への移行	145	カメラ AI	169
6. 未来編：おもてなしする敵 AI	146	カラー効果	171
接待上手な対戦相手	146	コレクション AI	171
世界共通の敵	148	4. 写真はタイムマシン	173

## AI 座談会～デザイナー編～ 174

<b>9 章 アンビエント AI</b>	<b>150</b>	<b>未来編</b>	<b>184</b>
1. アンビエント AI の基本設計	151		
2. 人を生み出すアンビエント	152	1. ボイスは良好な投資	187
リージョン配置	153	2. 独り言からの脱却	188
人の設定	154	3. 会話の量産：組み立て会話の紹介	189
ゲーム進行に合わせた変化	154	セリフの組み立てと言語	190
3. 人々を行動させる SmartLocation	155	4. 組み立て会話とローカライズ	192
オブジェクトからキャラクターを制御する	155	5. 質の高い会話	192
プレイヤーに掛け合う	156	6. AIトークの仕組み「台本システム」	196
スクリプトの設計	156	会話の内容を自動コントロール	196
決まった絵を作るための対応	157	会話の頻度を自動コントロール	196
人々の会話	157	会話の演出を自動コントロール	198
スクリプト総数	157	7. まとめ	199
4. 見えない努力：最適化	157		

<b>12章 AI モード</b>	<b>200</b>	<b>4.まとめ</b>	<b>223</b>
1. AI 部外者のための AI	201	14章 これからのゲーム AI 技術	224
今日のキャラゲー、未来のキャラゲー	201	1. ゲーム AI の全体的な変化	225
「知恵の上書き」とは?	202	ゲーム AI 全体の進化	225
ドラマを徹底的に量産する仕組みの構築	203	学習・進化・ディープラーニング	226
AI を止めたいけれど止めたくない: AI モードの誕生	203	モーション生成	226
2. FFXV は AI モードでできている	204	メタ AI の進化	227
AI モードがレベル組みで大活躍	204	2. インタラクティブ・ストーリーテリング	227
カーバンクルは AI モードだけで動いている	205	インターラクティブ・ストーリーテリングとは?	227
実際、便利だった? 問題なかったの?	206	ストーリー構造による分類	228
3. AI モードの紹介	206	対話システムの試み	231
ウェイト／ターン: 「たたずんでいるだけでもドラマ!」	206	インターラクティブ・ストーリーテリングのまとめ	232
ゴートゥー: 「歩いているだけでもドラマ!」	206		
フォロー: 「ついて行くだけでもドラマ!」	207		
ルート／リード: 「巡る、導くだけでもドラマ!」	208		
その他のモード	209		
4. AI モードの作り方	209		
共通グラフ	209	<b>あとがき</b>	<b>244</b>
アセット化	211	<b>謝辞</b>	<b>245</b>
オーバーライド	211	<b>簡易用語集</b>	<b>246</b>
5. AI モードは完璧なシステムか?	212		
良かったところ	212		
悪かったところ	212		
6. 技術ではなく、人とワークフローとインターフェース	213		
<b>13章 ロギングと可視化</b>	<b>214</b>	<b>AI 座談会 ~チームリーダー編~</b>	<b>234</b>
1. アーキテクチャ	215		
ロギング	216		
ログ・アグリゲータ	217		
データフォーマット	217		
データ解析	218		
2. 統計解析	218		
3. 空間解析	220		
2 次元ナビゲーションメッシュマップ	220		
ヒートマップによる可視化	222		

# はじめに — FINAL FANTASY XV のAIを俯瞰する

三宅 陽一郎

FINAL FANTASY XV (以下 FFXV) には、たくさんの人工知能 (AI) 技術が散りばめられています。「キャラクターの頭脳」をはじめ、「キャラクターが複雑な森の中で道を見つけられるようにする」「ゲーム全体を面白くする」といった目的にもその技術が組まれています。ゲームAIはエンジニアだけで作り上げるわけではありません。ゲーム開発の仕事全般がそうであるように、アーティスト、ゲームデザイナー、アニメーターとの協調作業になります。それはゲームのキャラクターを「単なるアルゴリズム」(情報処理装置) として実装するのではなく、ゲームという世界の中の実体として、一個の身体と知性を持った「生命」として存在させようとしているのです。



図 仲間キャラクターと立つ主人公キャラクター(ノクト)

ゲームには様々なキャラクターがいます。仲間には「一緒に旅をするキャラクター(図)」「一時的に一緒に旅をするキャラクター」「街にいる人々」などがいて、敵にも「モンスター」「敵兵士(ニフル兵)」「大型アーマー」「ボスキャラクター」がいます。AIアーキテクト、AIエンジニアの役割は、それぞれの人工知能を、基本的に同じ仕組みで作成できるようにすることです。しかし、それぞれのキャラクターには特殊な条件があります。固有の案件はその分野のエンジニアが対応し、中身はゲームデザイナーと相談しながら作っていける仕組み、あるいはゲームデザイナー自身が思考を構築できるような仕組みが入っています。

## 本書の方針

本書には2つの方針があります。1つめは、「FFXVで培った人工知能技術、広い意味でキャラクターとゲームをインタラクティブに作る技術」をくまなく解説することです。これは、単に人工知能技術だけでなく、ゲームデザインやアニメーションデザインなど多面的な内容となります。2つめは、そこから見える「未来のゲームの人工知能」について語ることです。

ゲーム開発には、時間やリソースなど様々な制限があります。しかし、開発していると「今回はどうしてもできなかつたが、次はこれをやろう。次はこれができる」という次の一手が見えてきます。ゲーム開発者のキャリアはその連続で「今回できなかつたら次、次にできなかつたことはその次」というように、段階的に技術レベルが上がっていくします。それはゲーム開発の歴史そのもので、ゲームの歴史はそのような開発者の心によってでき上が

ってきたのです。本書でもそれぞれの開発者が成したこと、そしてそこから見えるゲームの未来について語ります。FFXVのAI技術のみならず、デジタルゲームやエンターテインメントの未来を俯瞰する内容となっています。

## 全体の構成と各章のダイジェスト解説

本書は3部構成となっており、その間には開発者を3つのグループに分けた「座談会」が収録されています。まず、それぞれの座談会から読み始めると、自然なかたちで中身に入っていけると思います。

**PART1「基礎編」**は技術的基礎を解説し、**PART2「コンテンツ編」**ではそれを使った具体的なコンテンツについて説明します。**PART3「未来編」**は、これから発展する余地の大きなAI技術を集めています。それぞれの章は独立しており、どこから読んでも大丈夫です。わからないことがあれば、PART1で解説されている多くの基礎技術を遡ってみてください。では、それぞれの章をダイジェストで紹介していきましょう。

PART1「基礎編」は1~5章となります。ここはいわば、FFXVのAIシステムの基礎部分です。ゲーム内の様々なコンテンツは、この基礎の上に積み重ねられていきます。

### 1章 ゲームAI入門

この章では「そもそもゲームAI（デジタルゲームの人工知能）とは何か」について、ポイントを最小限に絞り、なるべくシンプルに説明します。

### 2章 意思決定ツール

「キャラクターの意思決定」はゲームAIの中心的テーマです。あらゆる人工知能研究の中でも、ゲームAIが先進性を持っている分野と言えるでしょう。なぜなら、ゲームという箱庭の中で、複雑なマップ・状況における高度なミッションを人工知能に解かせるという実験を、極めて早い開発イテレーションの中で行えるからです。

### 3章 ナビゲーションシステム

仮想の地形空間の中で、仮想の身体を使って運動するのがゲームの特徴でもあります。ここではナビゲーション技術やパス検索に加え、身体を持つキャラクターがゲームの環境下で目的に沿って運動するための技術を説明します。ステアリング技術など、ロボット開発で培われた技術も活用されています。

### 4章 AIとアニメーション

「身体と知能をどのようにつなぐか」は、ゲーム産業の人工知能が直面している最も大きな問題の1つです。仮想の身体はモーションによって駆動しますが、そのタイミング・変形・次のモーションへの遷移などを決めるうえで、人工知能がどのように関わっているかを解説します。

### 5章 位置検索システム

位置解析は近年（2012年以降）、ゲームAIの基礎になった技術です。かつてゲームキャラクターの目的地は、あらかじめ決められた地点か、敵・味方キャラクター、あるいはその周辺などに限られていました。位置解析技術はそのような限定された、あるいはあらかじめ決定された位置ではなく、ゲーム内で現在そのキャラクターがいる周囲の地形を認識し、自分に最も相応しい立ち位置を見つけることを可能にします。仲間であれば味方との距離をどれくらい保つか、敵との関係性においてどこにいるべきか、モンスターであればどのような位置取りをすれば攻撃されにくいかを考えます。

PART2「コンテンツ編」は6~10章までとなります。PART1で解説した基礎技術を組み合わせ、それぞれの分野へ応用します。

## 6章 仲間AI

主人公のノクトと共に旅をし、共に戦う仲間の人工知能を紹介します。プレイヤーと一緒にいる時間の長いキャラクターほど、高度な人工知能が必要になります。なぜなら、あらゆるケースに自然に対応させつつ、ほこりびが出ないようにしなければならないためです。FFXVでは、このように最も難易度の高い「仲間AI」に挑戦しました。「仲間AI」には、「メタAI」「キャラクターAI」「ナビゲーションAI」を含め、現代のゲームAI技術が集約されています。

## 7章 モンスターAI

モンスターAIでは「たくさんの種類のモンスターをどのように統一的な仕組みで扱うか」を探求しました。広大なマップのいたるところで群生するモンスターたちの人工知能を解説します。「生命を持つようなモンスター」を目指して、積み重ねられたいくつものノウハウをご覧ください。

## 8章 兵士AI

FFXVの兵士は「純粋な敵」であり「倒すべき相手」です。コンピュータゲームの敵キャラクターは、プレイヤーに立ちふさがる障害です。ただ邪魔な存在ということではなく、プレイヤーが敵と遭遇したときに「脅威」を感じ、倒したときに「障害を乗り越えた」という喜びを感じさせる存在でもあります。勝利のために「思考」「移動」「行動」する知能と、ゲーム開発では定番の「AIを組む」という工程について解説します。

## 9章 アンビエントAI

FFXVには多くの人で賑わう街が登場し、その特徴も様々です。街の活気を表現するのに必要なのは群衆（アンビエント）です。キャラクターの数が多いため、かなり特殊な作り方をしています。「物や位置にキャラクターを制御させる」という発想の逆転もその1つです。この章では技術のみならず、何より実際の街を観察して「街とは何か」「人の活気とは何か」という点を徹底的に追及したゲームデザイナーの工夫が、たくさん詰め込まれています。街の見方が変わるかもしれない、たいへん面白い内容となっています。

## 10章 写真AI

写真AIは、本書で最もユニークな章です。FFXVには、自動的に写真を撮る機能があります。人工知能が良いと思うタイミングで撮ってくれるので、では、写真を撮る「良いタイミング」とはいつでしょうか？ ゲームデザイナーとエンジニアがゲームの世界のキャラクターになりきって、シャッターチャンスを思考します。

PART3「未来編」は11~14章までとなります。ここでは、新しいAI技術が投入された分野を解説していきます。これらは完成された技術というより、これからより大きく高く発展していく分野です。

## 11章 会話AI

デジタルゲームには、キャラクターたちに「自然な会話」をさせたいという夢があります。自然な会話とは、その場・その状況だけでなく、それまでの物語としてのコンテキスト（会話の流れ、物語の流れ）が重要な役割を果たします。ところが、台詞がボイス付きであれば、テキストの会話のようにすぐに増やすことはできません。ここでは「いかにして効率よく会話にバリエーションを出すか」をテーマに、基幹技術を説明します。

## 12章 AIモード

デジタルゲームのキャラクターは自律的に活動すると共に、「演技」をする必要があります。ファンタジー世界の空間の中で自律的に動くのと同時に、物語の中で特定の役割を果たす必要があるのです。そこで、我々はキャラクターの特定の活動傾向を指定する「モード」という機能を開発しました。これにより、ある程度自律的に、特定の目的に沿った行動をキャラクターにとらせることができます。このような工夫の原理を説明します。

## 13章 ロギングと可視化

デジタルゲームはますます巨大なシステムになりつつあります。FFXVは、同世代の作品と比べても最も大きく複雑なシステムです。その全貌を把握することは、開発者にとっても容易ではありません。そこで「ゲームを客観的に分析してみよう」という発想が生まれます。本作ではゲームのログを取り、それを可視化（ビジュализーション）することを試みました。注目するゲームの特徴のデータを取り、グラフ形式などを用いれば、一目で状態を理解できるようになります。ちょっとした工夫が大きな効果を生む分野です。

## 14章 これからのゲームAI技術

本書の結びとなる実験的かつ挑戦的な章で、これからのゲームAIの展開について考察します。主題はゲームを「生成するシステム」です。つまり、アーティストやデザイナーがデータを作成するスタイルでなく、ある程度、ゲーム自身が基本データを元に生成的に形成・変化させていくシステムです。特に「物語生成」について、これまでの歴史的な経緯とともに大きなテーマとして取り上げます。

### 読み方ガイド

本書は複数の執筆陣ということもあり、傾向も難易度も様々です。読むときの目安として、各章を分類してみましょう。

#### ゲームの中身により添った章：6、7、8、9

☞ ゲームの裏側を知ることができます。

#### AIの基本的な仕組みを描いた章：1、2、4

☞ ゲームAIの基本的な仕組みを知ることができます。

#### AIの特徴的な技術を解説した章：10、11、12

☞ ゲームAIの専門的な独立性の高い技術を学ぶことができます。

#### やや専門的な章：3、5、13、14

☞ かなり詳細に踏み込んだ専門性の高い技術解説となっています。

自分の興味のあるところから、ぜひお読みください。また、巻末（P.246）の簡易用語集も合わせてご参照ください。





PART  
1

基礎編

# CHAPTER:1

## ゲームAI入門



三宅 陽一郎  
Yoichiro Miyake



横山 貴規  
Takanori Yokoyama

FINAL FANTASY XVではリードAIアーキテクトとして開発に従事。その他の代表作にFINAL FANTASY XIVなど。2011年入社、リードAIリサーチャーとして、デジタルゲームにおける人工知能の開発・研究に従事。国内・海外での講演も務める。毎週、社内AIセミナーを主催し250回に及ぶ。人工知能学会編集委員。

1978年生まれ。ゲームプログラマー。1児の父。AIを中心としたゲームキャラクター制御技術が専門。複数ゲームタイトルにてリードプログラマー経験があり、チームマネジメントも得意とする。在籍当時はAIエンジニアとして活躍し、FINAL FANTASY XVでは意思決定ツールの制作に携わった。

# 1. 人工知能を導入したらゲームはどう変わるか?

「人工知能(AI)をゲームに導入する」とはどういうことを簡単に解説するため、まず、最もシンプルなデジタルゲームの人工知能の作り方を紹介しましょう。図1のように、地面・岩・池のあるマップがあるとします。このステージにいるキャラクター(敵)の人工知能を作っていくします。

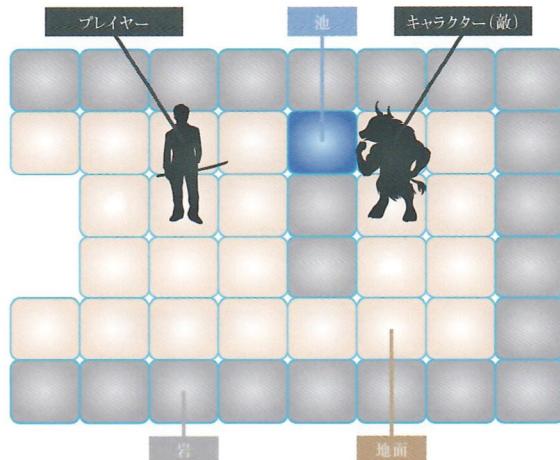


図1 ゲームの状況

このステージを見て、ゲームデザイナーは画面を2つに分類し、次のようなロジックを書きます(図2)。

- ・プレイヤーが左のエリアAにいるうちは何もしない
- ・右のエリアBに来たら、直進してプレイヤーにアタックする

このようにゲームを俯瞰的に見て、たくさんのキャラクターに対してルールを作っています。このルールはゲームデザイナーの頭の中にある考え方を表現したものであり、キャラクター自身が知識を持っているわけではありません。このタイプのAIは簡易プログラム言語で書かれることが多かったため、「スクリプティッドAI」と呼ばれます。現在でも、俯瞰的な画面の2Dゲームや小型の3Dゲームでは、スクリプティッドAIが主流です。しかし、この手法はある程度ゲームが大規模化・複雑化すると、とたんにルールが複雑になり、状況をすべて取りこぼすことなく列挙するのが難しくなります。

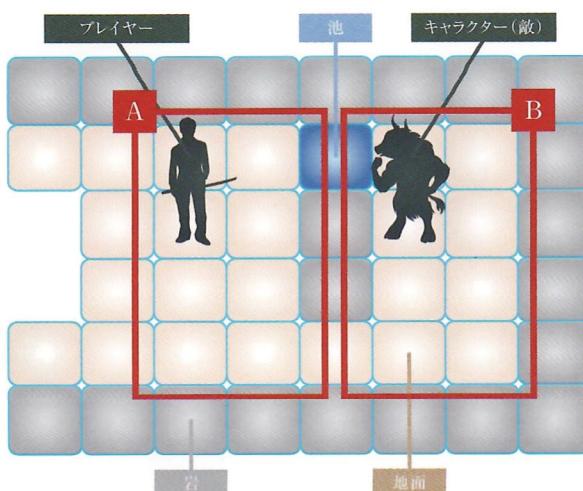
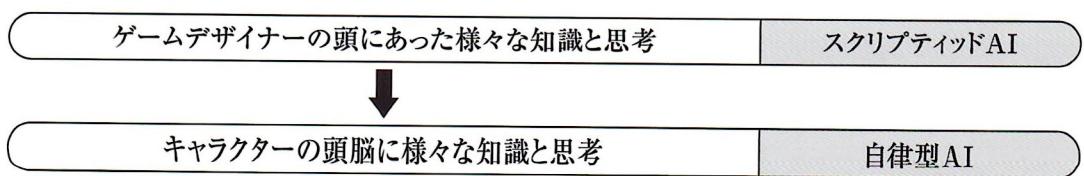


図2 エリアを2つに分割

例) キャラクターのルール  
 ・プレイヤーがAにいれば  
 Bをうろうろする  
 ・プレイヤーがBにいれば  
 近づいて攻撃する

俯瞰的にキャラクターを制御するロジックをスクリプティッドAI(Scripted AI)と言います。ゲームデザイナーがスクリプト言語で書く場合が多いのです。

そこで大きな転換点として、以下のように「スクリプティッドAI」から「自律型AI」に発展します。こうして、キャラクターが持つ知識を作るという手法により、大規模ゲームにも対応可能となりました。



ではキャラクターに「地形データ」を与えてみましょう。図3のように、キャラクターが歩ける場所を点で表して、それらをつなぎだデータを作ります。この点と点の連結データがあれば、キャラクターは自分のいる場所から到達地点までの経路（パス）を検索して、どのような場所にも自由に行くことができます。その結果、キャラクターは敵（＝プレイヤー）に近づけるようになります。

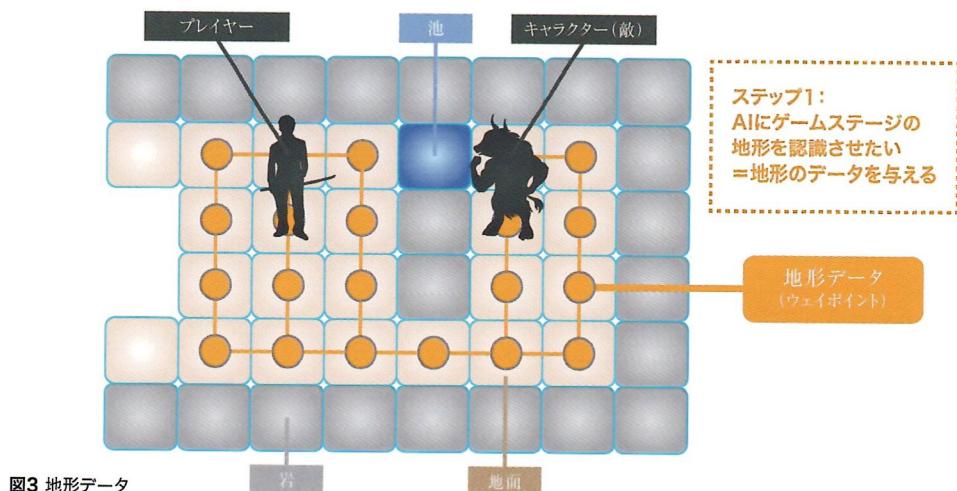


図3 地形データ

この地形データのことを「ウェイポイント」（Way Points）と言い、経路（パス）を発見することを「パス検索」と言います（図4）。キャラクターは「ウェイポイント」を使って「パス検索」し、「目的地に近づく」ことができます。これは現代のゲームAIの基本中の基本であり、そのおかげでキャラクターはマップ内を自由に動き回ることができるのです。「ウェイポイント」上で「パス検索」するAIを「ナビゲーションAI」と言います。

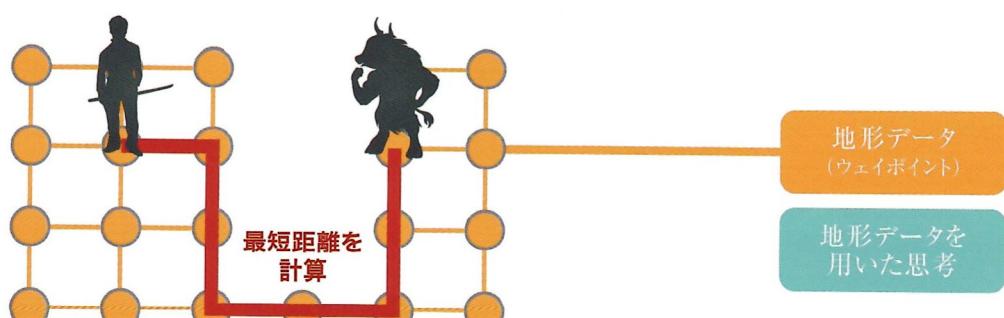


図4 パス検索で「近づく」

「ウェイポイント」のようにキャラクターが持つ知識のことを、「知識表現」と言います。ここで1つ大きな原理を提示しましょう。

キャラクターに与える知識表現が多いほど、キャラクターは賢い思考ができるようになる

この原理に従い、知識表現をどんどん精緻にしていきます。

ゲームAIでは「敵から見えている／見えていない」という判定が重要になります。そこで、このマップ上で「見えている点同士」をグルーピング・分類してみましょう。上の行の点は、どの点からも他の点が見えます。下の行も同様です。ところが、左中段の6つの点のグループは右のエリアから見えません。右中段の4つの点のグループも左のエリアから見えません（図5）。

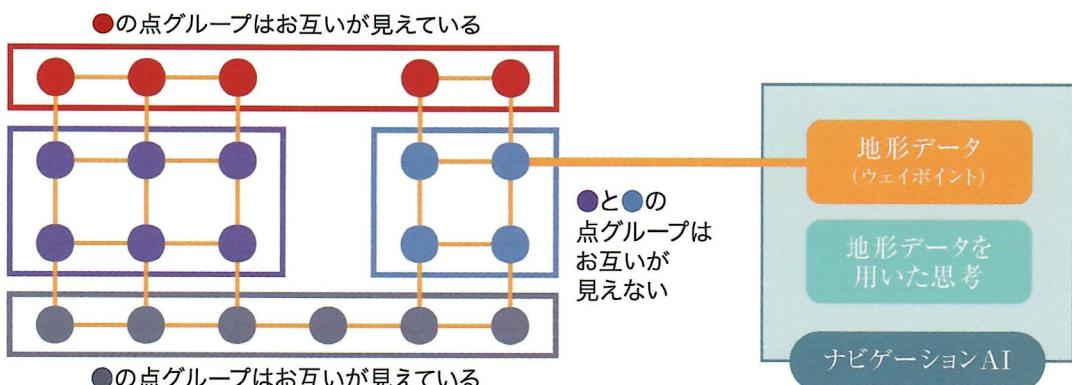


図5 可視情報(お互いが見えている点)によるウェイポイントの分類

この情報をウェイポイントに加えることで、次のような処理が可能となります。

- ・プレイヤーが左上の点にいるとき、敵は右のグループの4つの点に進み、「隠れる」（図6）
- ・プレイヤーが上段か下段にいるとき、敵は同じグループの点に進み、姿をさらして「威嚇する」（図7）

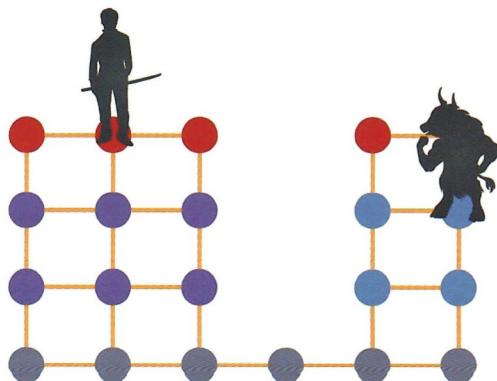


図6「隠れる」

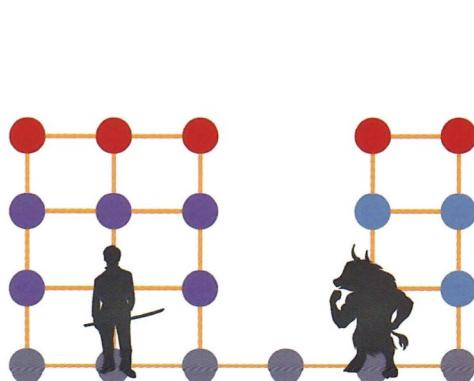


図7「威嚇する」

とても簡単ですね。とはいっても、もしあなたが開発者で「隠れる」「威嚇する」を実装してほしいと依頼されたら、途方に困ることもあるかもしれません。そんなときに人工知能の基礎として「ちょっとした知識をキャラクターに持たせる」という知識表現のアイデアを知っていると、一見難しいことでもさっと作れてしまうわけです。

さて、この時点での「近づく」「隠れる」「威嚇する」という3つの行動が定義されました。次は3つの行動可能性から1つを選ばねばなりません。これを「意思決定」と言います。その選び方は、様々な「意思決定アルゴリズム」によって決まります。意思決定については、2章「意思決定ツール」の解説をご覧ください。また3章「ナビゲーションシステム」の解説で、理解がさらに深まるところでしょう。

## 2. キャラクターを賢くする方法

ここで、前節を振り返ってみましょう。まずキャラクター制御について触れ、それを「スクリプティッドAI」と呼びました。つまり、ゲームデザイナーが外側からキャラクターを操るため、「操り人形の操り方」を記述したわけです。

次の大きな方向転換として、キャラクター自身に知能を持たせることになります。これを「自律型AI」と呼び、知識と思考を与えました。先ほどは「地形の情報」という知識を与え、さらにその情報を段階的に詳細にしていきました。そしてその情報を用いることで、「近づく」「隠れる」「威嚇する」など、3通りの知的な行動を実現することができました。

つまり、キャラクターの賢さ（知能）とは

知識 × 思考

ということができます（図8）。

ここでいう「知識」とは、知識表現によって型付けられたデータのことです。

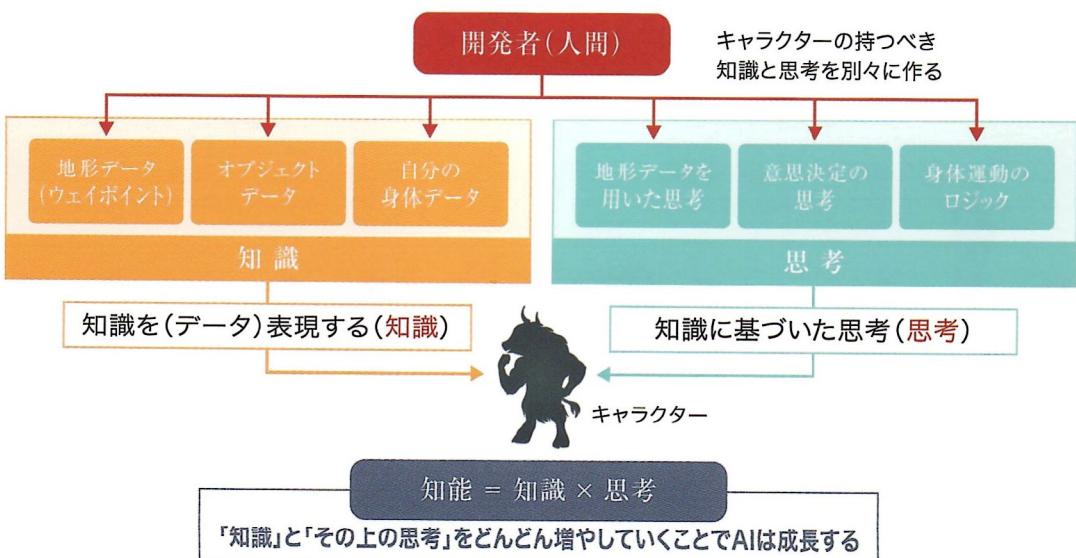


図8 知能とは知識と思考

思考を築くためには確かな知識の土台が必要であり、その土台が強固であるほど高い思考を作ることができ、知識が精密であるほど精緻な思考を作ることができます。これはあらゆる知能に言えることで、「メタAI」「ナビゲーションAI」についても同様です。メタAIの知識とは、ゲームの大局的な情報のことであり、思考とはそれを用いたゲームの変化に当たります。ナビゲーションAIについても、知識は「地形情報」であり、思考とはそれを用いた地形情報の解析となります。

したがって「知識」をより大きく詳細にしていけば、キャラクターは賢くなります。周囲のものや自分自身について知れば知るほど、キャラクターは賢くなっていくのです。本書を読み解く鍵は「知能 = 知識 × 思考」です。どの章を読むときも、「何が知識として固定」され、「どのようにそれを使っているか」(=思考)を読み解いていきましょう。様々な「知的機能」を実現するために、知識と思考をどのように作っているかがわかれれば、人工知能を理解したことになります。

本書では、

- (1) どのような知的機能を実現する必要があるか、それはなぜか
- (2) その知的機能をどのように実現するか(知識と思考)
- (3) 結果としてどのような知能が実現できたか

という3点について解説していきます。

### 3. ゲームデザインと人工知能

ゲーム開発では、まずゲームデザインが決められます。プレイヤーの皆さんにどのような体験を与えたいかによって、そのゲームデザインが決まり、それを実現できるようにキャラクターに対して必要な役割をそれぞれ割り当てます。

敵／味方など大きな分類もありますが、「ここで30秒間足止めして欲しい」「なんなく障害になって最後はあっさりやられて欲しい」「プレイヤーの傍にいて守って欲しい」といった、細かな役割も設定されています。そのため、それぞれの場所やキャラクターには、その使命に沿った動きをさせる必要があります。たとえモンスターであっても、本能的な行動ばかりではなく、きっちり果たすべき役割があるので。このように、自分自身の思考によって役割を果たせる人工知能のことを「自律型AI」「自律型エージェント」といいます。

#### ❖ 自律型AI

FFXVのキャラクターは学術的に「自律型AI」と「自律型エージェント」に分類されます。これは「1つの役割を持ち、自分の感覚で感じ、自分の頭脳で意思を決定し、自分自身で身体を動かし、役割を果たそうとする人工知能」という意味です。

簡単に言えば、世界にぽんと放り込むと、目的に向かって自分で思考し、行動する人工知能です。そのために、我々はそれぞれのキャラクターに感覚、頭脳を組み込み、身体を動かす手段を与えます。それも単に身体を動かすだけではなく、目的に沿い、周囲の環境に適応した行動をとるようにします。味方を守るために、敵と味方の間の位置に走り込んだり、遠くへ向かうたびに梯子を上ったり、段差を飛び降りたり、最も適切な敵を最初に攻撃したりします。このような技術的達成もまた、知識と思考の掛け算によるものです。

## 4. 現代のゲームAIの3つの柱

現代のゲームAIには3つの柱があります(図9)。まずキャラクターの知能である「キャラクターAI」は、キャラクターを自律型知能にするために必要な頭脳のことです。前述のように、知識と思考がその基礎となります。

次は、環境を認識する「ナビゲーションAI」です。このAIは、キャラクターが直面する多様な地形に関する思考を抽象化する役割があります。その基礎となるのが、地形に関する知識表現です。これは「世界表現」とも呼ばれます。「ウェイポイント」が世界表現に基づくデータとなり、さらにその上に載せられたデータを用いて思考するのがナビゲーションAIになります。とりあえず、ウェイポイントやナビゲーションメッシュといった世界表現のデータを作つておけば、どのような地形であれ、そのデータ上で思考すればよいのです。

最後に、ゲームを神様の視点から眺めて調整を行う「メタAI」があります。このAIはゲームを俯瞰的な視点から観察し、ゲームが面白くなるようにコントロールします。これについては、6章「仲間AI」で詳しく説明されています。この3者の連携によって、ゲームAIの様々な効果が生み出されています。書かれている内容がわからなくなったら、この3つのどれについて語っているかを振り返ってみましょう(図10)。

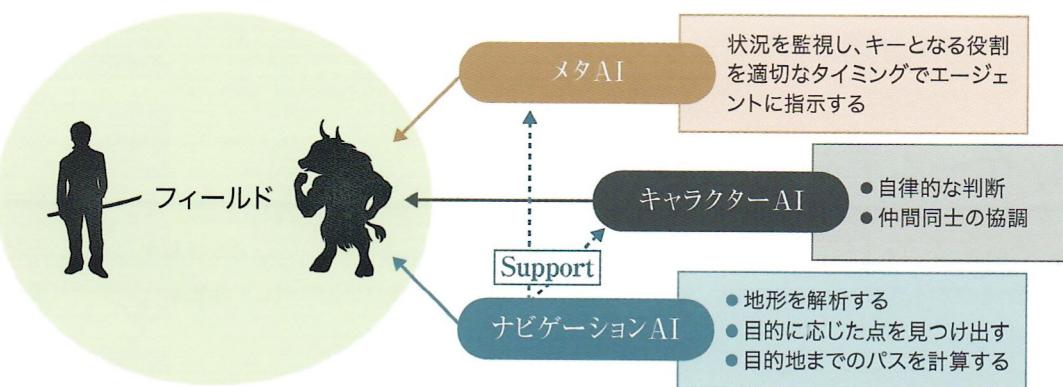


図9 現代のゲームを形作る3つの人工知能

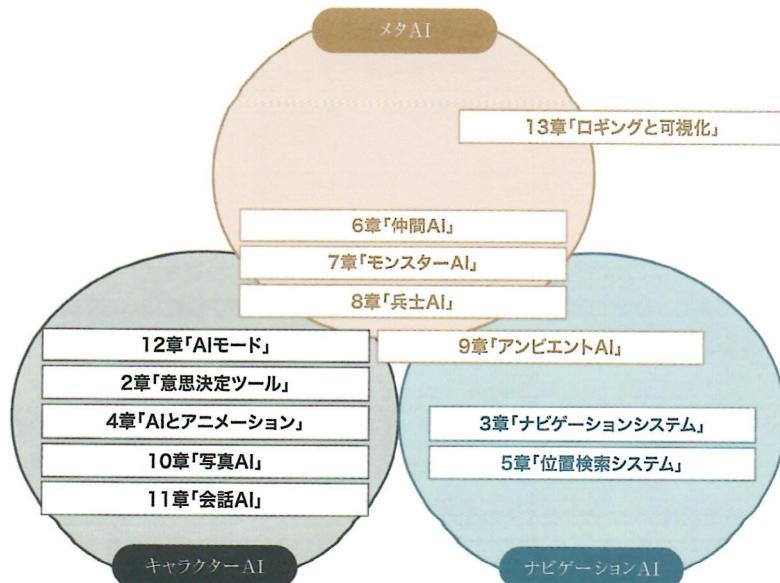


図10 本書の各章の分類

## 5. ゲームAIの歴史

ゲームの人工知能の分野は、1970年代以来のゲーム開発の歴史の中でも、特に1995年以降に発展しました。ゲームが小型の頃は、俯瞰的にゲームを見渡してキャラクターを操ることができました。その後、ゲームにドラマが導入され、3Dになり、ネットワークゲームになり、ソーシャルゲームになり…と、ゲームが変化するごとにゲームAIも発展してきました。つまり、**ゲームAIの発展はゲームの発展と共にあったのです。**

あるときは、ゲームが要求する仕様に応えようと人工知能が発展し、あるときは人工知能の発展がゲームの発展を支えてきました。このように、ゲームの中核部分と深く相互的に関わりながら、ゲームの人工知能は発展してきたのです。

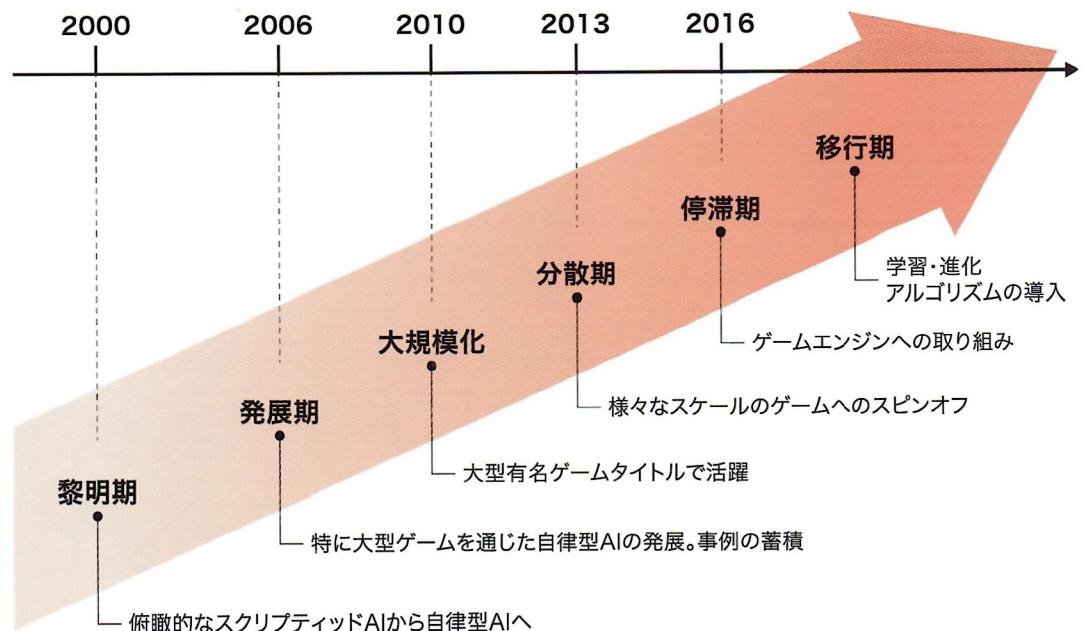


図11 ゲームAI発展の歴史

FINAL FANTASY XVは現世代（2019年）の段階に位置し、AAAと呼ばれるもっとも大きなサイズのゲームに分類されます。大型ゲームと呼ばれるこの分野では、小型ゲームにはない、大きく多様でダイナミックな世界が再現されています。また、そこに住むキャラクターたちも、必然的に、高度な知能が実装されています。それらによって、プレイヤーもまた広大で多様な世界の中で、様々なスケールの大きい体験を味わい、物語の世界に没頭できるのです。

# CHAPTER:2

## 意思決定ツール



白神 陽嗣  
Yoji Shirakami



下川 和也  
Kazuya Shimokawa

FINAL FANTASY XVではプログラマーとして開発に従事。主にキャラクターのAIや身体制御の基幹システムを作成。その他代表作として、FINAL FANTASY 霊式など。

FINAL FANTASY XVではプログラマーとして開発に従事。主に意思決定ツールの開発、仲間キャラクターのバトルを担当。

# 1. 概要

## 【意】意思決定処理とは？

意思決定処理を簡単に言うと「環境から得た情報を基に、自分の目的に応じた取るべき行動を決定すること」です。言葉にするとたったこれだけですが、実現するには、非常に高度な作業が必要となります。

「認識の形成」フェーズでは、センサーから得た事実を記憶に格納し、その情報を解釈して意味付けを行い、認識を形成します。例えば「遠くに何か動く物体が見えている」という事実から、「15m先に敵がいる」と認識するなどです。

「意思の決定」フェーズでは、解釈された情報を基に行動を決定していきます。例えば「敵に近づくため、一旦安全な場所を検索して、そこに移動する」といった行動を決定します。その際、優先すべき事項、「メタAI」(6章「仲間AI」参照)などからのシグナル、身体の状態などの情報も加味します。通常の状態では「敵を攻撃するために近づく」といった行動を取りますが、回復対象の仲間がいる場合は「回復を優先するために仲間に近づく」といった行動を取ることもあります。

「運動の構成」フェーズでは決定した行動を基に、身体に行動生成を依頼します。ここは、4章「AIとアニメーション」で詳しく説明します。

これら一連の流れのうち、意思の決定フェーズで行う処理を、「意思決定処理」と言います(図1の赤枠)。

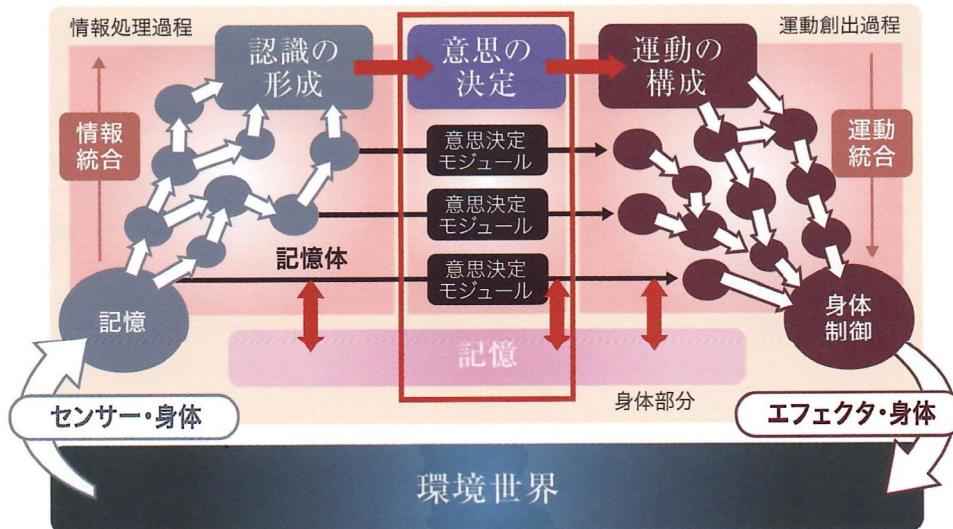


図1 意思決定処理の範囲 <出典1>

## 【意】意思決定ツールとは？

意思決定処理は、簡単な条件分岐だけであればプログラムやスクリプトで記述することも可能ですが、判断すべき情報や状況が増えると、すぐにメンテナンス不能なほどにコード量が膨れ上がってしまいます。そのため、近年では、それぞれのゲームに最適な意思決定アルゴリズムを選んで、意思決定処理を記述することが多くなっています。

意思決定アルゴリズムは、主に以下の7つの型に区別されます。

- ① ルールベース：ルールを基本に組み立てる
- ② ステートベース：世界や人間の行動を状態で表す
- ③ ビヘイビアベース：振る舞いや行動レベルでキャラクターを動かす
- ④ タスクベース：タスクに分解して実行順序を選択する
- ⑤ ゴールベース：目標の達成から行動を組み立てる
- ⑥ ユーティリティベース：効用・見返りから行動を決定する
- ⑦ シミュレーションベース：事態を予測して想像する

<参考:出典2>

「ゲームデザイナーがこれらのアルゴリズムを視覚的に作成できるようにすること」を目指して作られたのが「意思決定ツール」です。

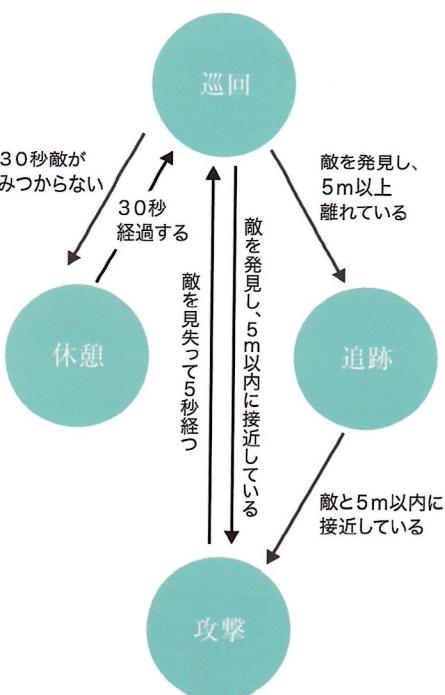


図2 ステートベースAI <出典2>

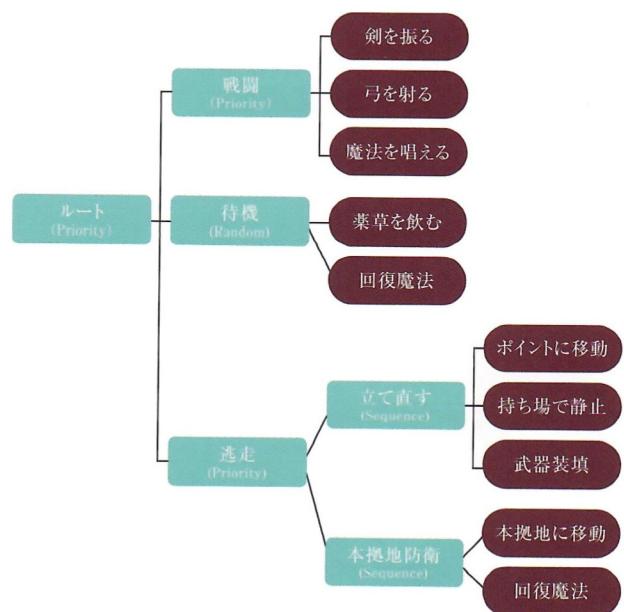


図3 ビヘイビアベースAI <出典2>

## FINAL FANTASY XVに登場するAIキャラクター

FINAL FANTASY XV (以下 FFXV) はオープンワールドのゲームであり、たくさんの種類のキャラクターが世界の中で生息しています。これらのキャラクターが持つ目的は、その性質によって様々です。NPC（ノンプレイヤーキャラクター）は日常生活を営み、モンスターは生態を形成し、仲間は常に主人公と行動を共にします。ニフル兵は主人公たちを倒すことだけを目的に活動しています（図4）。

それぞれの目的から意思を決定するには、必要な入力情報も、意思決定の時間粒度も様々です。例えば、日常生活を営むAIの場合、意思決定してからの行動時間はやや長めですが、バトルをしているときのAIの場合、意思決定を非常に短い時間で繰り返します。



(a)



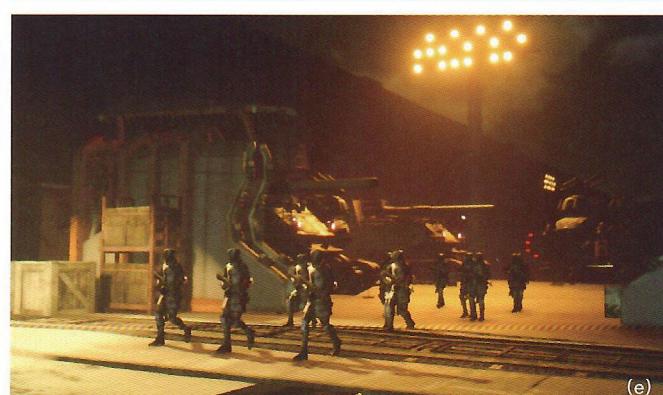
(b)



(c)



(d)



(e)

図4 (a,b)街で生活する人々、(c)主人公と旅を共にする仲間たち、(d)世界に生息するモンスター、(e)主人公たちを狙うニフル軍

## FFXVに求められた意思決定ツール

オープンワールドで自律的に活動するためには、まず自分の置かれている状況を認識しなければなりません。キャラクターは、ナビゲーションAI（3章「ナビゲーションシステム」で詳しく説明）やセンサーシステムで適切に情報を受け取り、自分以外のキャラクターや地形の情報を認識していきます。また「ブラックボード」を介して、自身やゲームの情報（HPや毒の状態など）を受け取り、自分の目的に応じた行動を決定します。例えば、バトル時に「攻撃するべきか隠れるべきか」「どのように攻撃するか」などの行動を決めます。隠れるときは、最適なポイントを「PQS（Point Query System、位置検索システム）」に問い合わせ、パス検索とステアリングシステムにそのポイントまでの移動を委ねます。また、ゲームの進行状況や、バトル時のチーム行動においては、「メタAI」からの指示を優先させる必要があります。

FFXVではこのような要件を満たすため、それぞれのAI技術を利用できるようなツールが求められました。こうして、意思決定ツールが各AI技術を結ぶハブとなり、これを通して、ゲームデザイナーはAIを作成していました（図5）。

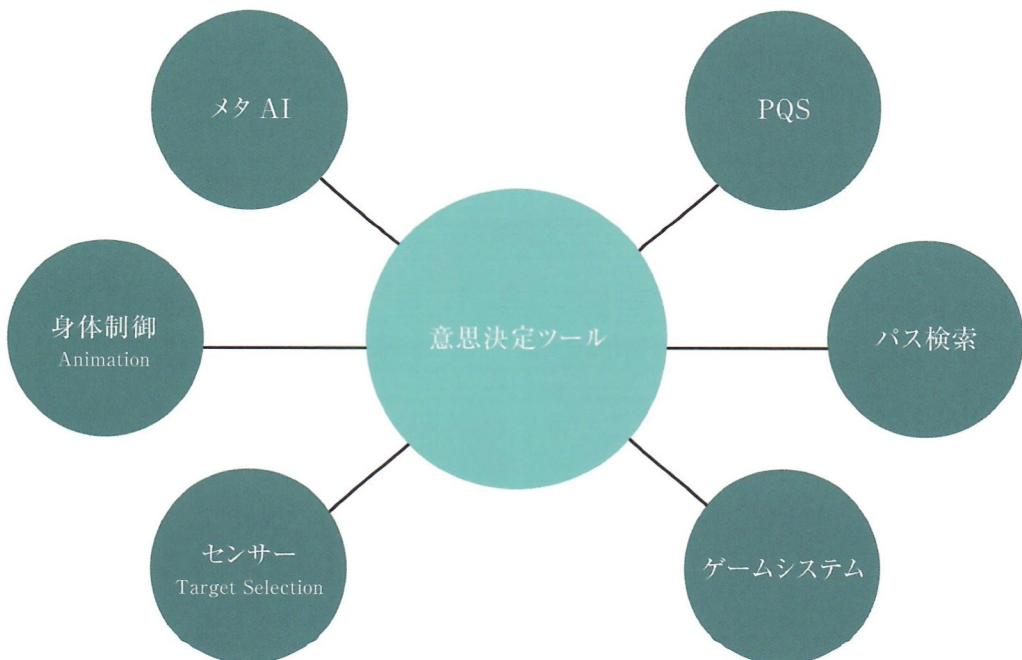


図5 意思決定ツールを通じてAI技術を結ぶ

FFXVに登場するキャラクターは様々で、意思決定のアルゴリズムにも向き不向きがあります。そのため、単一の意思決定アルゴリズムで全てのAIを表現するのは簡単ではありません。例えば、リアクティブな処理が得意なビヘイビアベースAIは、バトル表現には向いていますが、状態を保持することができないため、長い時間をかけて目的を達成するような行動には向きません（図7）。逆に、ステートベースAIは厳密に状態を保持できますが、細かい状態が多くなると、状態遷移が多くなり、メンテナンス性が著しく低下することがあります（図6）。そのため、FFXVの意思決定ツールには、複数のアルゴリズムを組み合わせて記述できる仕組みが求めされました。

本作で実装されているのは、「ステートベースAI」と「ビヘイビアベースAI」の2つのアルゴリズムですが、他のアルゴリズムも順次追加していくる仕組みとなっています。

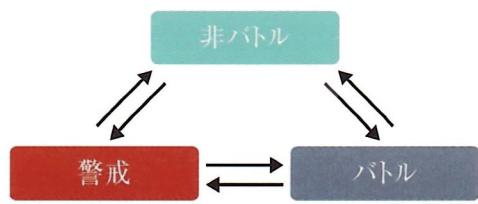


図6 ニフル兵のAIを記述するとき、バトル中／非バトル中などの状態で管理するため、上位層ではステートベースでAIを記述したい

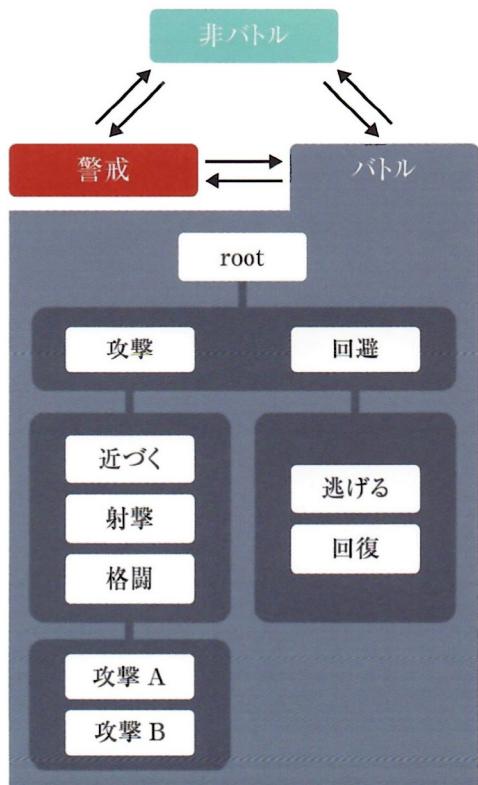


図7 バトルの状態で取る行動については、リアクティブな思考に向くビヘイビアベースでAIを記述したい

## 2. ツールの基本システム

### 複数の意思決定アルゴリズムを組み合わせる仕組み

先述のとおり、FFXVではステートベースAIとビヘイビアベースAIの2種類の意思決定アルゴリズムを組み合わせて、意思決定を記述できるシステムが構築されています。ステートベースAIは「ステートマシン」という状態遷移グラフで記述します。これは、状態と遷移条件を設定して行動を記述する方法で、一つひとつの行動を厳密に実行したい場合に向いています。

一方、ビヘイビアベースAIは、「ビヘイビアツリー」というツリー型グラフで記述します。これは、連続動作やランダム性をシンプルなツリーで制御するもので、一連の動作を指定する場合に向いています。それぞれ異なる特徴を持つこれら2種類のグラフを組み合わせて使うことで、柔軟なデータ作成を可能にしています。ただ、一言で複数のアルゴリズムを組み合わせると言っても、そう簡単にはいきません。FFXVの意思決定ツールでは、異なるグラフ構造のアルゴリズムを組み合わせて使えるようにするために、2つの仕組みを導入しています。

### ハイブリッドノード フォーマット

1つめは、どんなグラフ構造でも処理を共有できるようにするための「ハイブリッドノード フォーマット」です。これは、グラフ構造の役割を果たす「グラフノード」と、処理の中身を実行する役割を果たす「処理ノード」を分離した形式です（図8）。グラフノードはグラフの制御に特化した機能であり、システムで提供されています。

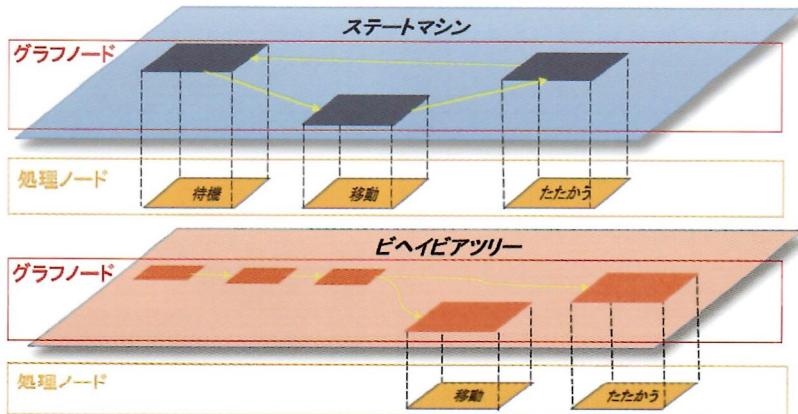


図8 グラフノードと処理ノードを分離

処理の中身となる「処理ノード」（AIロジックを記述するために必要となる行動のノード。「移動」「たたかう」など）は、ゲームプログラマーが作成します。これは、初期化時／起動時／実行時／終了時に呼び出される処理をプログラミングするようなフォーマットになっています。用意したノードはシステムに登録され、ステートマシンとビヘイビアツリーで同じように使用できるようになります。

例えば「ターゲットが近い場合は【近接格闘攻撃】をして、ターゲットが遠い場合は【ターゲットまで移動】するが、近づけない場合は【射撃攻撃】をする」というAIを作るとします。ゲームプログラマーは、【近接格闘攻撃】【ターゲットまで移動】【射撃攻撃】を実行する処理をフォーマットに従い作成します。すると、これらのノードは図9、10のように、ステートマシン／ビヘイビアツリーのどちらにも配置できるようになります。

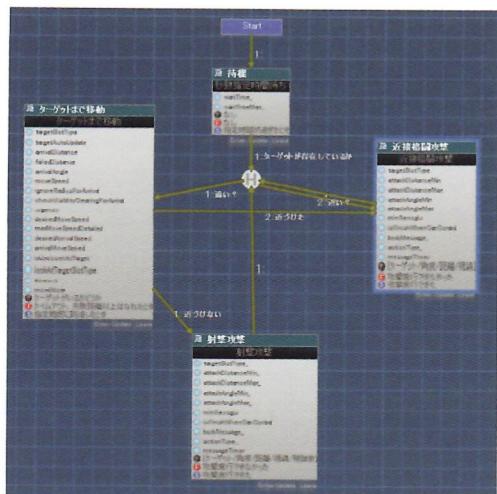


図9 前述のAIをステートマシンで表現した場合

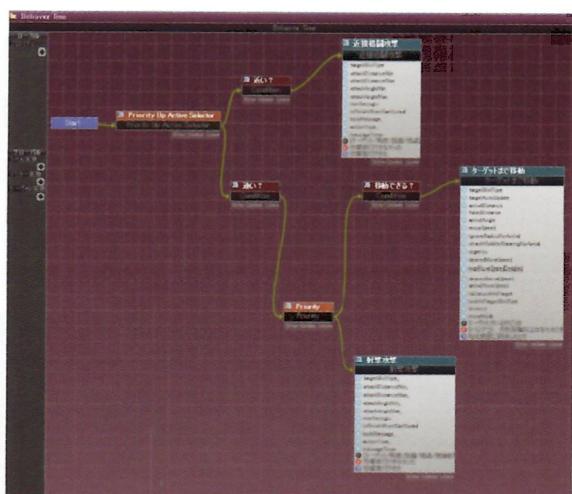


図10 前述のAIをビヘイビアツリーで表現した場合

この仕組みによって、どんなグラフ構造のアルゴリズムでも、処理の記述を共有できるようにしています。

## 階層型グラフシステム

もう1つの仕組みが「階層型グラフシステム」です(図11)。これは、処理の中身である「処理ノード」をグラフにすることにより、あるグラフのノードの中に別のグラフを構築できるようにするシステムです。言い換えると、複数のノードで形成されるグラフを1つにまとめることができるというものです。ここでは、ひとまとめにしたグラフの単位を「トレイ」と呼びます。このトレイ単位で意思決定アルゴリズムを自由に選択できるため、状況に応じて最適なアルゴリズムでAIを記述できます。

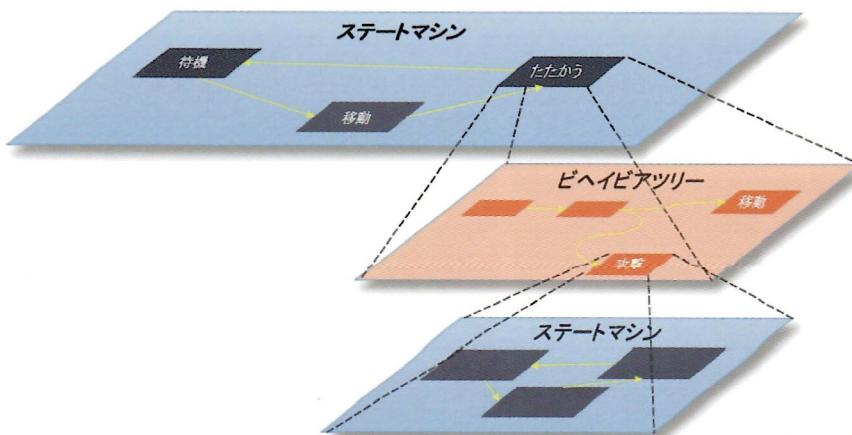


図11 階層型グラフシステム

トレイ単位でグラフを階層化し、データを適切な粒度で分ければ、いわゆる“組み合わせ爆発”を減らせます。その結果、不要な遷移の少ない単純なグラフになり、手軽に新しい処理を追加できるようになります。このように、階層型グラフシステムは「拡張性の担保」という意味でも大きく貢献します。

ゲームの開発が進み、グラフが大きく複雑になってくると、メンテナンス性が著しく落ちてきます。特にステートマシンにはそれが顕著に現れます。しかし、グラフを階層型にすることで、こうした問題にも柔軟に対応できるようになります。

## 身体・敵・ゲームとAIをつなぐブラックボード

キャラクターが身体の情報やゲームの情報などを取得するのに、「ブラックボード」という仕組みを利用します(図12)。これは、その名のとおり、自由に読み書きできる黒板のようなものです。ブラックボードに直接書き込まれている情報もありますが、ブラックボードに参照先だけが書き込まれている場合もあります。いずれにせよ、AIはブラックボードというインターフェースを通して、記憶や身体、ターゲットの情報などにアクセスできるのです。

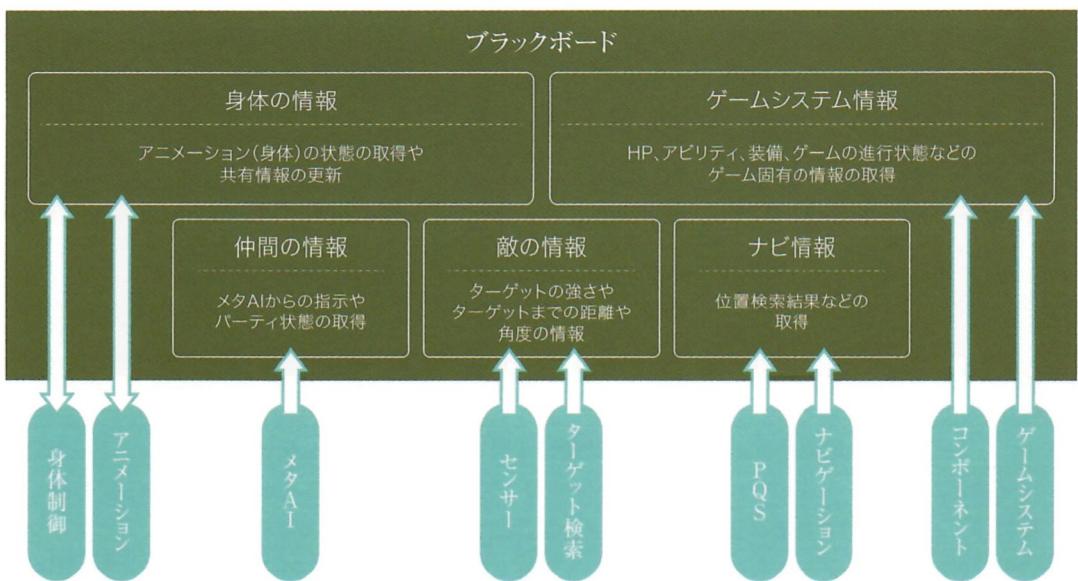


図12 ブラックボード概念図

FFXVの意思決定ツールにおけるブラックボードは、ノードとリンクできます。ノードに対して、外部パラメータとして情報を与えたり、条件判定の式の中で使われたりします(図13、14)。情報は自由に追加できるので、意思決定に必要な情報はすべてブラックボード経由で取得できます。

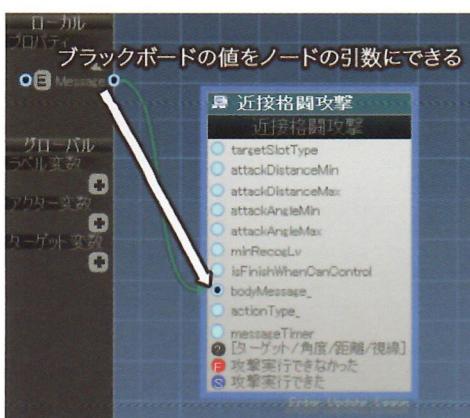


図13 ブラックボードの値をノードの外部パラメータとして使用

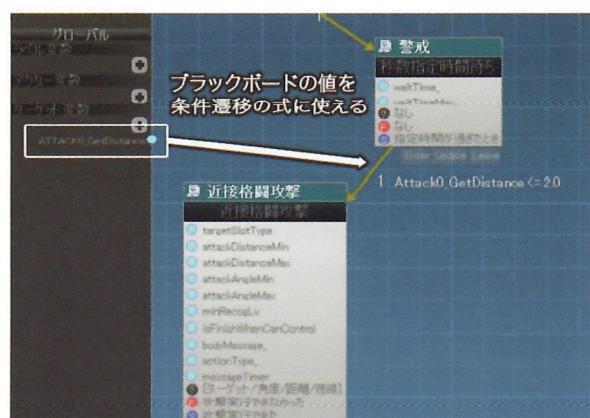


図14 ブラックボードの値をステートマシンの条件遷移に使用

### 3. ツールの応用的な機能

#### ☞ 高度なAI記述を実現する仕組み

##### 並列思考

AIキャラクターが行動している最中に別の行動をさせたり、常に別のことを考え続けさせたい場合があります。これは一般的に「並列思考」と呼ばれるものですが、FFXVの意思決定ツールでは、ステートマシンでもビヘイビアツリーでも、この並列思考を実行する仕組みを導入しています。ステートマシンの場合は、グラフのrootとなる [Start] ノードを複数配置して、トレイの中で複数のステートマシンを実行できます（図15）。



図15 ステートマシンによる並列思考

ビヘイビアツリーの場合は、[Parallel] ノードで並列思考を実現します（図16）。いずれかの配下ノードが終了したときに、[Parallel] ノードも終了します。

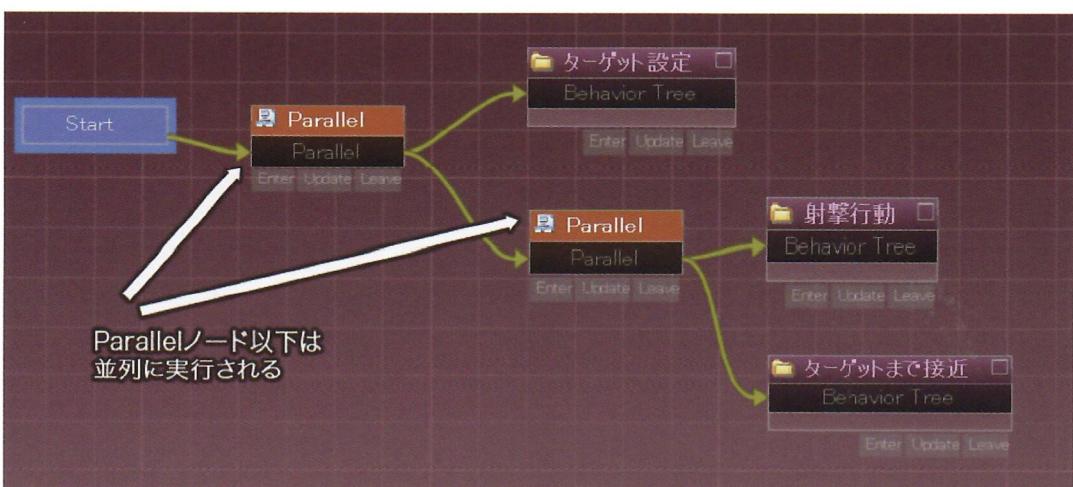


図16 ビヘイビアツリーによる並列思考

## 割り込み制御

割り込み処理は、主に身体制御システムからのコールバック（ダメージを受けたなど）や、メタAIからの指示に反応するための仕組みです。ステートマシンではリアクティブな条件遷移を実現できるように、Interrupt（割り込み）という仕組みを導入しました（図17）。[Start] ノードに紐づける形で [Interrupt] ノードを配置でき、その [Start] ノード配下のステートマシンが実行している間、常に Interrupt 条件を監視し続けます。これにより、各ステートからの条件遷移を書くことなく、スマートに割り込み処理を実現しています。



図17 Interrupt

ビヘイビアツリーでは [ActiveSelector] ノードにより、リアクティブに実行ツリーを切り替えられるようにしています（図18）。このノードの配下のノードは、常に条件を監視しており、条件を満たす場合に、実行ツリーを切り替えるものです。

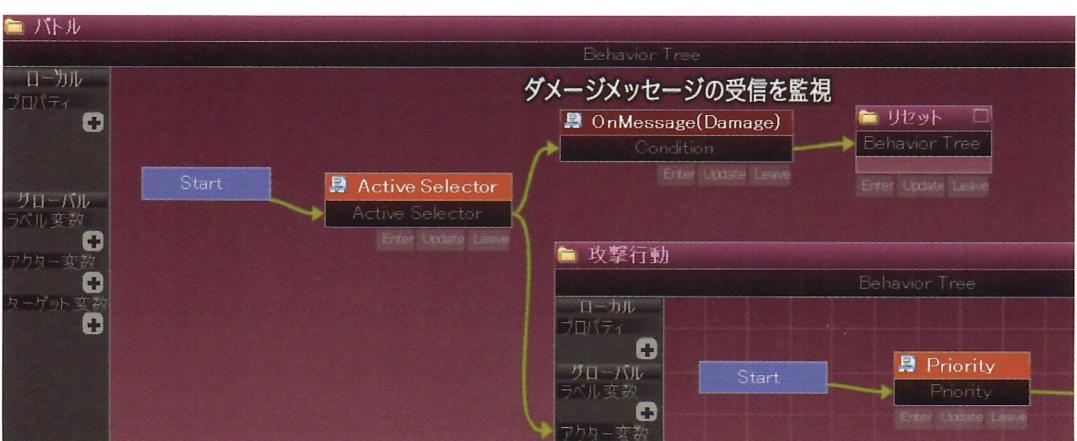


図18 ActiveSelector

# AI作成のワークフローを支える仕組み

## トレイのアセット化とオーバーライド

AIの作成はとても時間がかかります。ゲーム開発の中盤以降は、キャラクターを量産していくますが、ゲームデザイナーは「似ているが少し違うAI」を作ることが多くなります。このような場面で、コピー＆ペーストをせず、同じような構造のAIを複数作成できるように、共通のデータを使い回す機能が用意されています。

1つの機能は、トレイの「アセット化」です。アセット化は処理を共有するための仕組みで、汎用性の高い処理を1つの「部品」として保存します。アセット化されたトレイは、同グラフ内の複数箇所に配置したり、別のグラフで再利用することも可能です（図19）。またアセット化されたトレイに変更を加えると、そのトレイを利用しているすべてのデータが更新されるので、共通処理に変更があった場合にも、一つひとつすべてのデータを修正する手間が省けます。

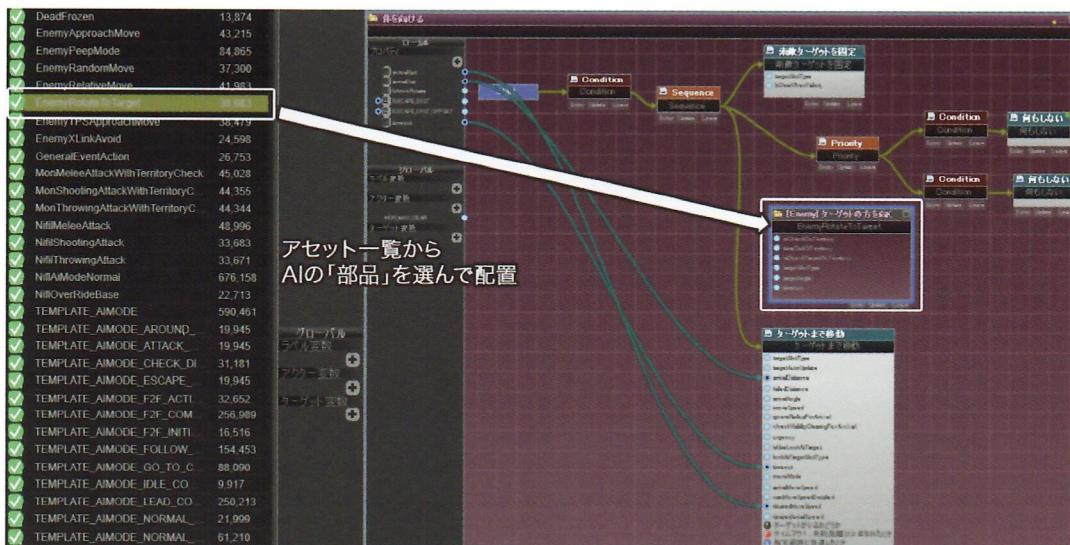


図19 アセット化したAIの部品を自由に配置できる

もう1つの機能は、トレイの「オーバーライド」です。これは構造を共有するための仕組みで、アセット化されたトレイ内的一部分を、別のアセット化されたトレイに置き換えます（図20）。同じような構造を持つAIで、一部だけを変更したいときに利用できます。



図20 オーバーライド

## デバッグシステム

もう1つ、AIの作成ワークフローを支える重要な要素が「デバッグ」です。ゲームデザイナーの願望として「実装したAIの確認や修正のトライアンドエラーのサイクルタイムを短くし、AIを思い描いた通りに堅実に動作させること」があります。しかし、キャラクターの挙動を画面から見ているだけでは、AIがゲームデザイナーの意図通りに動作しているかわかりません。そのため、意思決定ツールは、ゲームプログラムとノード情報を通信することにより、AIの思考情報をダイレクトにツール上へ反映させています(図21)。また、ツールを立ち上げていない状態でも、簡易的なウィンドウで実行中の状態を確認できるようにしています(図22)。



図21 AIの実行状態をリアルタイムにツールで確認できる

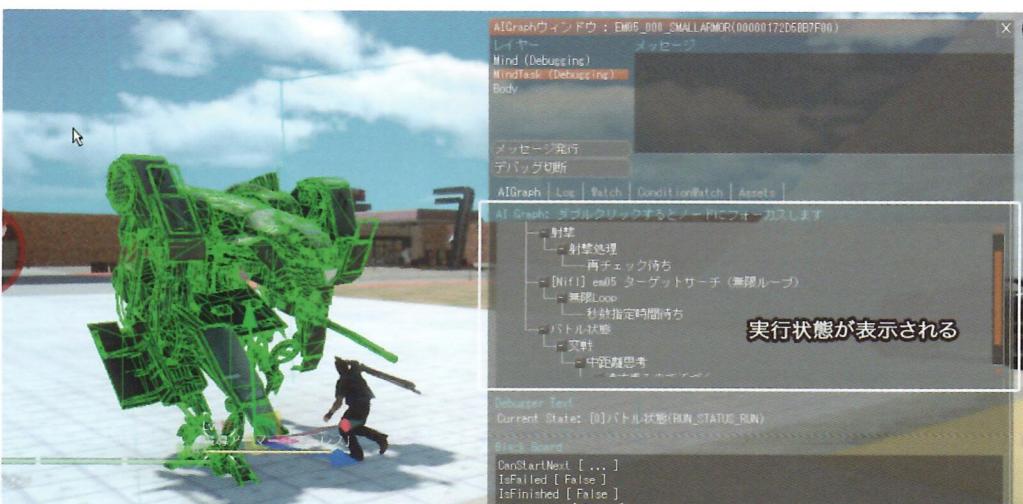


図22 インゲームウィンドウでも簡易的に確認できる

## 4. まとめ

### ☞ 意思決定ツールがFFXVにもたらしたもの

意思決定ツールがFFXVにもたらしたもの、それは様々な要件を持つ多種多様なキャラクターAIを開発・量産できたという点に尽きます。例えば、常にプレイヤーキャラクターと行動を共にする仲間キャラクターたちには、様々な状況に対応するための柔軟な制御が必要でした。そこで、ステートマシンを中心とした意思決定ツール上でメタAIなど他のシステムと連携させ、大規模かつ精密な意思決定が構築されています。

一方、仲間キャラクターほど広範囲に渡る複雑な制御は必要ないものの、自然なリアクションやプレイヤーを楽しませる行動が求められるモンスター、ニフル兵などの敵キャラクターは、ビヘイビアツリーを中心に構築されています。なお、敵キャラクターは種類が多いので、テンプレートトレイを活用し、量産を意識したデータ構造にしています。

意思決定ツールの最大の利点は、仕様やデータ作成者の思想が可視化されていることです。大規模なゲーム開発では、大量のデータを複数人で効率良く作成し、メンテナンスし続けることが大切です。大量のデータを作成すること自体は、人数と時間をかけねば可能ですが、それをメンテナンスし続けることは非常に困難です。データのメンテナンスは、データ作成者が行なっていくのが効率的だと思われますが、開発初期から末期まで1人でメンテナンスし続けるのは、ほぼ不可能です。つまり、他の人が作成したデータでも、容易に編集が可能でなければなりません。

他の人が作成したデータを編集する場合、まず既存データの内容を理解するところから始まります。そのようなとき、意思決定ツールが最大限の威力を発揮します。データが視覚的にわかりやすい形になっているので、全体像を掴みやすく、理解の手助けとなってくれます。また、過去に自分が作成したデータでも、月日が経つと忘れてしまうこともあります。そのようなときにも、わずかな時間で思い出せることでしょう。

## FFXVのその先へ

本作で実現したかった要素の1つとして、「ゲームの自動攻略」が挙げられます。本来ユーザーの操作によって進行するゲームプレイのすべて、またはその一部をAIでやってしまおうという試みです。これができれば、開発中の実装状況の確認やバグの早期発見など、QA的にも大いに役立ちます。これを実現するには、プレイヤーキャラクターを自動で目的地に向かわせて、ゲーム進行に必要なインタラクションをさせたり、バトルをさせたり、様々な状況に応じて幅広く制御することが必要になります。バトルなどの限定された環境下での制御は、現状のシステムでも十分可能ですが、ストーリーを進めていく上で、必要な処理のすべてを記述するのは非常に難しく、作業コストがかかります。

そこで、ゴールベースAIを導入することで、より効率的にこれを実現できるのではないかと考えています。このAIでは、設定された目標（ゴール）の達成を目指して意思決定を行います。ステートマシンやビヘイビアツリーと異なり、反射性は薄いですが、中長期的な意思決定に向いているアルゴリズムです。先述のとおり、ゲームAIで使われる代表的な意思決定アルゴリズムは、7つに分類されます。それぞれのアルゴリズムには、長所と短所があり、得意とするゲームジャンルが異なります。つまり、汎用的に使える万能なアルゴリズムはないのです。こうしたことから、開発者はそのゲームにあわせて、適切なアルゴリズムを取捨選択することが重要になります。

意思決定ツールの理想形は、開発者に多くの選択肢を提供するものだと考えています。我々が目指しているのは、7つのアルゴリズムをゲームの特性に応じて使い分けられるゲームAI作成ツールです。ただ、汎用性が高まるに従い、ユーザーへの敷居が高くなってしまう問題もあるため、ノードを実装するプログラマーや、ノードを組み合わせてAIを作成するゲームデザイナーが実装時に迷わないように、ヘルプの充実やユーザインターフェースの向上も図りたいと考えています。

### ● 参考文献

- 出典1:FINAL FANTASY XV -EPISODE DUSCAE- におけるキャラクターAIの意思決定システム(CEDEC2015)
- 出典2:三宅 陽一郎『人工知能の作り方』技術評論社

# CHAPTER:3

## ナビゲーションシステム



ファビアン グラヴォ

Fabien Gravot

2004年、フランス トゥールーズのPaul Sabatier大学でコンピュータ・サイエンスの博士号を取得。2011年入社。FINAL FANTASY XIV: A Realm RebornとFINAL FANTASY XVでナビゲーションを担当。現在リードAIエンジニア。『Game AI Pro』『Game AI Pro 3』のチャプター執筆なども担当。

ナビゲーションシステムは、アニメーションより1つ上の階層の「移動に関与するシステム」です。これがきちんと動作すれば、環境をスムーズに移動できます。しかし、問題があると、NPCは地形に引っかかるか、到達できない地点の周りを延々と廻ることになります。したがって、ナビゲーションシステムは完璧に動作する必要があるのです。古典的なシステムですが、同時に最も成熟したシステムもあります。

ナビゲーションシステムは、出発地点から別の点に移動するためにあります、キャラクターは歩く以外にも「ジャンプする」「飛び越える」「飛ぶ」といった移動も可能です。また、アーティストやアニメーターと協力して群衆の動きを作ったり、モーションの質を高めたりすることも重要です。こうした仕事の大半は、ゲームを開発しても気づかないことが多いかもしれません。しかしこの分野では、多くの人工知能技術がゲーム開発を高速化するために使われています。

ナビゲーションシステムは環境の様々な情報から作られ、アニメーションシステムを学習し、より良い意思決定を行えるようにします。そして、QA(品質保証)やゲーム開発過程にも活用されています。また、開発者が創作に集中できるように、自動化を促す技術もあります。まずいくつかのわかりやすい事例を説明し、次に見えにくい事例について説明していきます。

## 1. ナビゲーションシステムとは？

最初に「ナビゲーションシステムとは何か」という質問にお答えしましょう。これは、障害物を回避して目的地に辿り着かせるためのシステムです。障害物は次の3種類に分けることができます（図1）

- **静的な障害物 (Static Obstacles)**：変化することのないもの
- **擬似静的な障害物 (Pseudo-static obstacles)**：大抵の場合、動くことはないが壊れるもの。駐車されている車など
- **動的な障害物 (Dynamic Obstacles)**：常に動いているもの。長期に渡ってバスを阻害しないもの

次は、この3種類の障害物をどのように扱うか見てていきましょう。

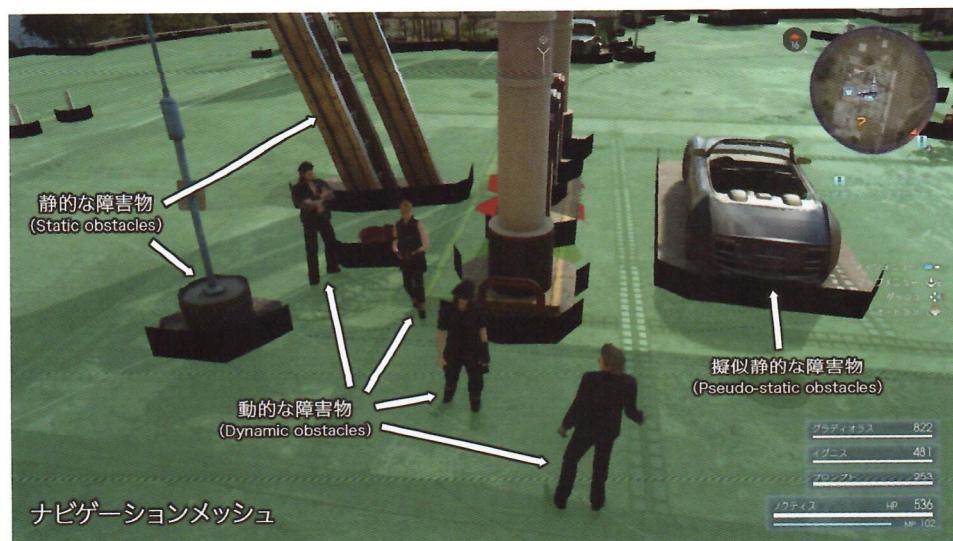


図1 3つの障害物のタイプ

## 【静的な障害物】

静的な障害物とは、すなわち動かないオブジェクトです。NPCがこれを避けて通れるようにするには、ゲーム内でNPCに認識させるのではなく、あらかじめ計算によってナビゲーションデータを作成しておきます。大抵のゲームでは、ポリゴンとその接続情報からなる「ナビゲーションメッシュ」が使用されます。FINAL FANTASY XV(以下FFXV)のすべてのステージで、高品質のナビゲーションメッシュを作ることは大きな挑戦でした。しかし、いったん作成てしまえば、接続されたグラフを用いて最短経路を計算できるようになります。図2のように、イグニスはノクトに向かってこの最短パスを辿ることで、噴水を避けて通れるのです。

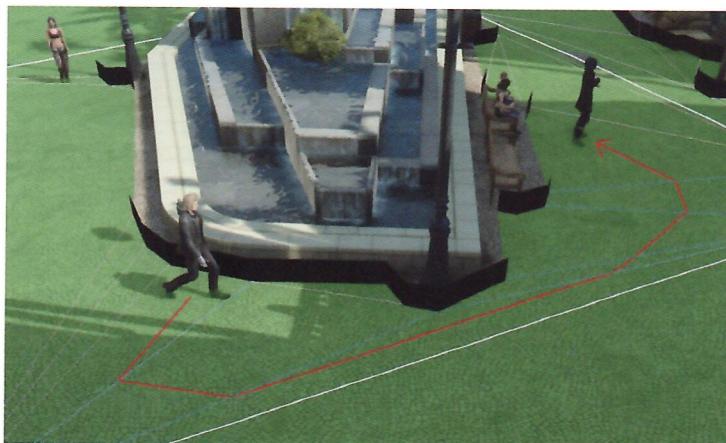
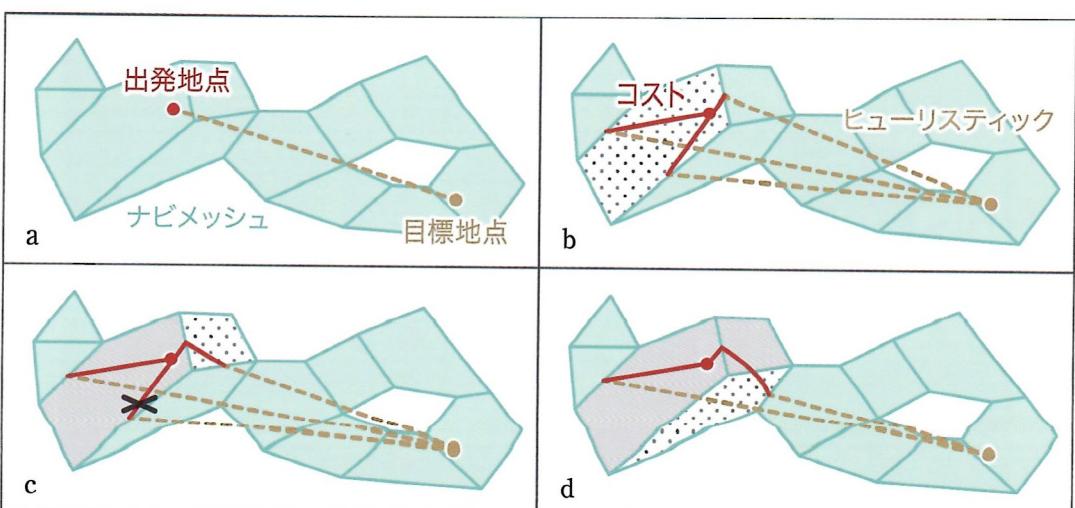


図2  
イグニスがノクトへのパスを見つける様子

## 【パス検索】

ナビゲーションメッシュを用いて最短パスを見つけるため、1968年以来よく知られている「A\*アルゴリズム」を用います。図3はそのステップを示したものです。出発地点(スタート)と目標地点(ゴール)を指定すると、ナビゲーションメッシュによって、出発地点のポリゴンから目標地点に至るまでのエッジが出力されます(a)。パフォーマンスの観点から、パスはエッジの中心をつないで形成されます。これらの中心点と出発地点との距離がパスのコストとなります。A\*アルゴリズムは、探索したポイントからゴールまでの推定距離「ヒューリスティック距離」を計算します(bの点線)。そして「スタートからのコスト + ヒューリスティックコスト」の合計が最小となるポリゴンを選択します(c)。



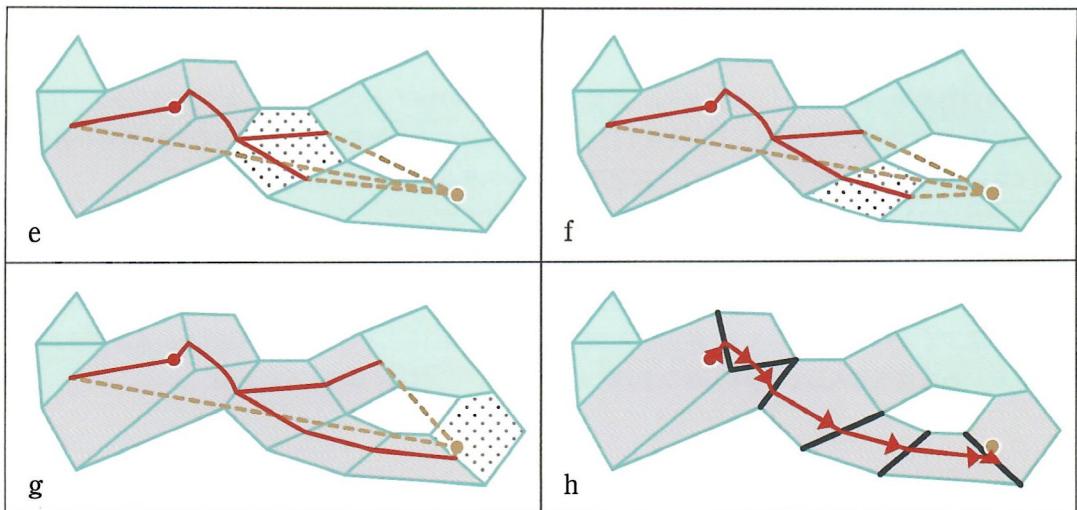


図3 小さなナビメッシュ上のパス検索アルゴリズム

**d ~ g**は、パス検索アルゴリズムの原理を示しています。この例のように、最短パスを見つけるためにチェックするポリゴンが、ヒューリスティックのコスト計算によって上手く選ばれていきます。**g**の時点では、少し上のパスも探索していますが、A\*アルゴリズムは処理を高速化するために、探索対象のポリゴンの数を減少させます。しかし、特に長い距離の探索、本来到達しない場所への探索は時間がかかるため、タイムスライスを行います。つまり、パス検索システムが1フレームで行える処理を実行します（応答は遅れるものの、人間が認知できない程度に過ぎません）。**h**は、最終的に発見されたパスです。NPCはこのパスを辿ってゴールに到ります。最適化と品質向上のため、実際はA\*による探索の前に直線を引き、パスが通るかチェックしています。

## 擬似静的な障害物（移動可能な障害物）

移動可能な障害物とは、一時的に移動するものの大抵の時間は停止しているオブジェクトのことです。キャラクターは、それらを避けて移動しなければなりません。この処理負荷は小さくなく、数フレームをまたぐこともあります。そのため、FFXVでは次の2つの手法を採用しています。

- ・ **リメッシュ**: その障害物をシンプルな形状（ボックス、シリンドラなど）として扱い、ナビゲーションデータを再計算する手法
- ・ **コンテキストスイッチ**: あらかじめ変更用のデータを計算しておき、オブジェクトの位置が変わると、そのデータをロードする手法

リメッシュはよくあるようなシンプルな地形にとても有用です。一方、コンテキストスイッチはより精緻なナビメッシュを使えるため、新しい橋や道などにも対応できます。なお、リメッシュは追加された障害物を避けるようにメッシュを再分割します。その障害物の上は歩けず、障害物が増え続けると歩ける道は段々なくなります。

コンテキストスイッチの場合は、その障害物の上にも道を作ることができます。ただ、どう変化するかは、あらかじめ想定されていなければなりません。具体的には「大きな岩が破壊される」「新しい道が開く」といったシナリオ上の変化があった際に用いられます。ナビゲーションメッシュの更新は、静的なコリジョン（衝突）の更新以上に処理負荷がかかり、わずかに時間が必要となります。そのため、リメッシュはまず、LoD（レベル・オブ・ディテール）にしたがって、プレイヤーの周囲のメッシュから更新していきます。

## 【動的な障害物（常に動く障害物）】

動的な障害物とは、基本的に「モンスター」「移動する車」「他のNPC」などを指します。実際のゲームプレイにおいて、特定の場所で回避したり、通過を待ったりすることになります。局所的な回避や短期的な待機については、長い回避路を作ることはなく、「ステアリングシステム」の範疇で対応可能です。

ステアリングシステムとは、その障害物の速度と位置から運動を予測して回避するシステムです。前述のパス検索プラナーを用いて生成されたパスエッジを使い、エージェント（指示を受けたキャラクター = NPC）が互いに衝突するのを防ぐ仕組みです。最短パスでなくエッジを用いることで、単にNPC同士が避け合うよりも遙かに高い自由度を実現しています。このステアリングについて、詳しく解説しましょう。

## 2. ステアリングシステムとは？

### 【概要】

ステアリングは動的な障害物を避けるためのシステムです。各エージェントは、現在の速度・位置・センサーの向きなどの情報を持っています。最も重要なのは、エージェントの動作が3つの過程（入力・処理・出力）の連携で成り立っているという点です。これらの過程で、どのようなステアリングをするつもりなのか、どのようにその結果を扱うのかを表現します。例えば「回避」という動作は、他のエージェントや障害物を回避することで達成されます。3つの過程は、それぞれ図4のような役割を担っています。

### 入力（インプットビヘイビア）

入力はエージェントの運動の指針となります。具体的なインプットデータは、パス・力・ポイントなどです。運動の指針は、これらの中からどれを取るかということに強く依存します。いずれにせよ、フレームごとに各インプットに基づいて目標となる表現（動き）を計算します。例えば、プレイヤーと共に移動する仲間キャラクターは、決まった目標位置を持たず、常にプレイヤーの現在位置に依存しながら動きます。

### 処理（プロセッシングビヘイビア）

二次元、あるいは三次元における衝突回避の処理を行います。入力の過程で計算された目標に基づいて、エージェントの移動を変更する力を算出します。

### 出力（アウトプットビヘイビア）

処理の過程で算出した力を、制御ベクトルとしてアニメーションへアウトプットします。さらにスムージングやデッドバンド（不感帯）を適用し、キャラクターの特性を考慮しながら、最終的な変更を促します。処理の過程では、速度0から最高速度まで計算するものの、デットバンドで最低速度が設定されているため、0～最低の速度は使えません。そのため、0になると止まるアニメーションをアウトプットする特殊なシステムが必要となります。すなわち、ヒステリシス（Hysteresis、履歴現象）と似た、停止と最低速度の繰り返しを起こさないようにするシステムです。

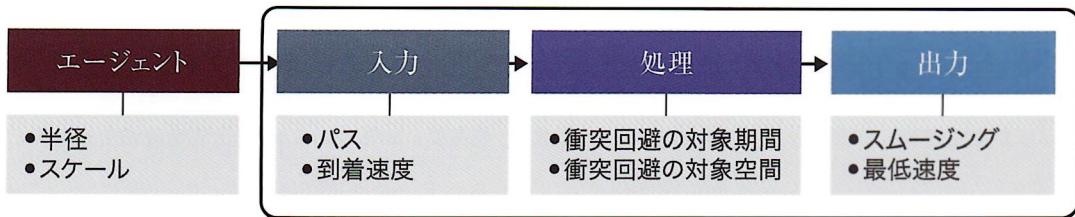


図4 エージェントと各ビヘイビアのデータレイアウト

## LoD (レベル・オブ・ディテール)

ビヘイビアはエージェントの意思（避けようとする意思、フォーメーションを組もうとする意思など）を表現します。そして、各ビヘイビアにアルゴリズムを具体的に指定せずとも、抽象的なレイヤーを加えることが可能です。この仕組みによって、アルゴリズムを自由に入れ替えて、多段階のLoD（レベル・オブ・ディテール、いくつかの制御の段階）を持つことができます。実装された具体的なアルゴリズムは「プロセッサ」（Processor）と呼ばれ、各ビヘイビアは複数のプロセッサを保持できます。

例えば、FFXVのパス追従において、高い段階のLoDではターン時のスピードを予測し、制限をかけています。衝突物を回避させる際には、3つのLoDが用いられます。最も低いレベルでは回避も衝突計算もありませんが、より高いレベルでは様々な方向に対して計算が行われます。

LoDの意思決定を簡素化するため、3つのビヘイビアに対して個々のオプションが設定されています。ランタイムでは、ステアリングライブラリが新しいビヘイビアの組み合わせに応じて、すべてのプロセッサの組み合わせを生成します。このような組み合わせを「パイプ」（Pipe）と呼びます。1つのビヘイビアに対応するすべてのパイプのセットを「パイプグループ」（Pipe Group）と呼びます。

パイプグループのパイプは、緊急性の高い順に並べられます。図5はその一例で、LoDの決定は単純化されています。エージェントは、パイプのインデックスを高い順や低い順に並び替えたり、あるいはそのままの形で維持したりできます。

いくつかのLoDの組み合わせは、不安定な結果を招きます。例えば、群衆制御と回避行動を行うエージェントは、回避行動を完全に止めることができません。なぜなら、障害物を回避しないとナビゲーションメッシュの外に出てしまうからです。一方で、パスをフォローしている間は、エージェントがメッシュ上にいることが保証されるので、回避行動を完全に削除できます。このように、問題のある組み合わせは、シンプルなルールで禁止される仕組みになっています。

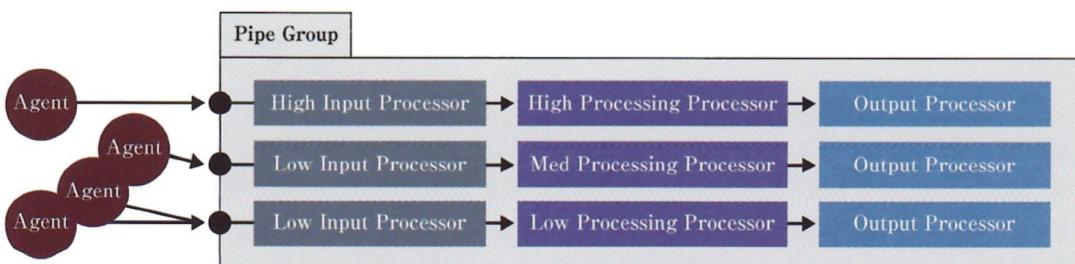


図5 LoDにおけるプロセッシングパイプ

LoDの決定はCPUリソースとの兼ね合いで行われます。これには、Sunshine-Hill [5] のモデルが採用されています。それぞれのプロセッサには、コストと緊急性のタグが付けられ、コストの総和である「C」と緊急性の平均値である「F」で管理しています。それぞれのエージェントには、プレイヤーとの相対距離に応じ、動的な重要度「I」が設定されています。この3つの値（コスト・緊急性・重要度）を用いて、LoDの決定を最適化します。

LoDを変更する際には、2つのステップを踏みます。

### ① 最も効果が高いパイプの選択（パイプのインデックスを減少）：

最も高い  $(F_{\text{new}} - F_{\text{old}})/(C_{\text{new}} - C_{\text{old}})$  を選び、パイプを変更します。これをコストの限界をやや過ぎるまで行います。

### ② コストの限界を守るため、最も早く計算できるパイプの選択（パイプのインデックスを上昇）：

最も高い  $(F_{\text{new}} - F_{\text{old}})/(C_{\text{new}} - C_{\text{old}})$  を選び、パイプを変更します。これをコストの限界ぎりぎりまで繰り返します。

このような数式によって状況が変わり続ける中でも、LoDのトレーディングアルゴリズムが上手くLoDの割り当てをコントロールします。ただし、CPUのリソースの消費には気をつけねばなりません。先に述べたように、コストの値は相対的なものであり、実際のコストを反映しているわけではありません。つまり、状況が変わるたびにアルゴリズムのコストが再調整されるのを避けるため、自動で対応するシステムが求められます。次に、エージェントの数に応じて、それぞれ一定のコストがかかる点にも気をつける必要があります。

また、それぞれのエージェントと他のエージェントに対するコンスタントなファクターを結びつけてはなりません。例えば、衝突回避のコストは、シーンの中のエージェントの密度とエージェントの数の二乗に比例します。なお、相対的なコストと1/100秒単位で計測される負荷のギャップを関係づけるため、「PID制御」を使用しています。このシステムによって、各フレームにおける割り当て時間と実際の実行時間の差が、誤差として計測されます。PID制御は、この値を動的にパイプの抽象的な単位に変換し、基本的にパイプごとのコストを更新します。

コストは特に時間単位でなくてもよいですが、基本的に2倍のコストは2倍の時間として反映されなければなりません。実行時間を測ると、コストと時間の関係も計算でき、さらに各パイプのコストを更新できます。これにより、デザイナーが設定したコストが実際にどのくらいかかるかを計算できます。もともとパイプは他のエージェントに影響し、そのコストは環境によっても変わるので、動的なコスト計算が必要です。何より、LoDの変更時においては、デザイナーが設定したコストという意味で  $C_{\text{new}}$  が一層重要となります。

## パス追従

個々のエージェントのパス追従は、もっとも多い運動の1つです。ここでは、Sunshine-HillやAnguelovが用いたものと似た、パスのエッジによる表現を用います [6]。これはウェイポイントにおけるキャラクターたちの混雑を避ける上でも有用です。途中の目標点についても、この仕組みを応用します。

図6のように、目標地点は向かうべき方向にあるエッジの端点となります。パスをフォローする間、進むべき方向の途上にあるエッジを選ぶことで、適切な方向に向かいます。もし回避する必要が生じた場合、逸れるか、エッジを用いてどの方向に進むべきかを決定します。パス追従では、辿ったパスのエッジのインデックスを記憶しており、そこから次に向かうべき最近接のエッジ（部分的に隠れていても視野に入っているもの）を決定します。向かうべき方向はそれから決定します（図7）。

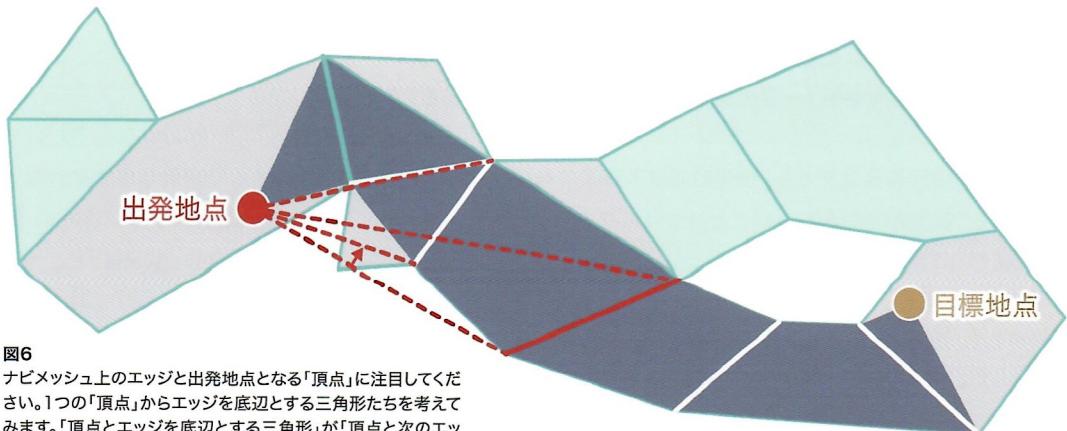


図6

ナビメッシュ上のエッジと出発地点となる「頂点」に注目してください。1つの「頂点」からエッジを底辺とする三角形たちを考えてみます。「頂点とエッジを底辺とする三角形」が「頂点と次のエッジに作る三角形」に完全に含まれてしまわなければ、その「エッジ」が次の目標エッジとして選ばれることになります

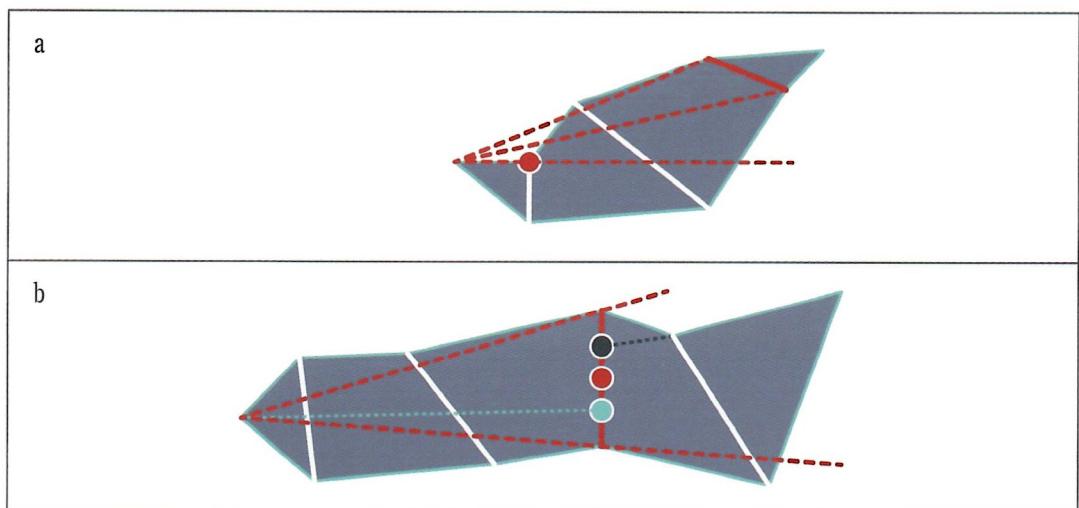


図7 次に向かうべきエッジが隠れているか(a)、隠れていないか(b)によって、次の方向を選択する

- a) もし隠れているエッジが完全に左(右)にある場合、エッジ上にある一番左のポイント(一番右のポイント)を選択します
- b) エージェントからエッジ上の最近接点(水色)と、次のエッジの最近接点(青)の中点(赤)

衝突回避のために、処理(プロセッシングピヘイピア)がエッジと方向をさらに解析します。エージェントが通過するエッジは、次のエッジへの距離が一定の閾値以下であれば(例えばエージェントの半径の10%以下であれば)スキップし、エージェントがエッジを超えてしまったら(エッジの外に行ってしまったなら)そのエッジ上か、最も近いエッジに戻されます。急カーブなどで、こういったケースが起こる可能性があります。

## ☒ 衝突回避

衝突回避は、ステアリングにおいて最も難易度の高い問題の1つです。四足歩行の動物を含め、どのようなエージェントも他のすべてのエージェントを回避する必要がありますが、様々なモーションモデルが途中で止まる可能性も考慮しなければなりません。そこで、いくつかの仮定を導入して問題を簡単にし、解決することになります。

- すべてのエージェントをシリンドと見なす
- 速度ベクトルの急変も可能とする
- エージェントは拘束系（一定の条件で動きが拘束されているもの）とする

もちろんこのように仮定すると、物理的には不正確になります。そして、この仮定のもとで完全な操作を行なったとしても、衝突が避けられないことはまれにあるでしょう。しかし大抵の場合、良い結果につながります。最もよく使われている衝突回避テクニックは、「確率的衝突回避」(ORCA)です[1]。しかし、我々は「ヒューマンライク衝突回避」(HL)という、Jérôme Guzzi氏によって開発された方法を採用しました[2][3]。これを分析して、計算コストと失敗の確率を比較したところ、平均速度と最短パスの発見において最も良い成績となりました。HLは最近接の対象との回避方法を予測して、対象との距離を最小化する速度と向きの候補を試し(図8)、安全に避ける速度を選択します。さらに、我々はオリジナルのHLを以下のように変更しています。

- センサーのモジュールの中だけではなく、全方位にサンプリングを行う
- ターゲットに向かって必ず1つのサンプリングを行う
- ゼロベクトルとなるサンプルを最優先（最小値）にする
- 向きを決めるときは距離を考慮に入れる（目標をポイントでなくエッジにしているのが主な理由）

我々のバージョンのHLではn個のサンプリングを行い、そのうち1つはビヘイビアに決定された方向へ向けられます。サンプリング数nは8~20個で、LoDレベルごとに異なります。LoDレベルが高い場合は、まず速度の最小値 $v_i^{\min}$ と最小移動距離をサンプリングごとに計算します。もしエージェントが動かなかった場合(つまり速度がゼロの場合)、衝突する可能性が出てきますが、それなりの対処方法があります。エージェントが動くときは、今回的方法で回避できます。この計算は状況に応じて切り替えることになります。

2つ目のステップとして、向きごとに最小速度とエージェントの理想的な速度 $v^d$ が与えられたとき、サンプルiごとにその方向に衝突するまでの距離 $d_i$ を計算します。ここでは、動的なオブジェクトもナビゲーションメッシュの境界も考慮に入れないものとします。もし何の衝突も起きなければ、デフォルトの最大値を利用します。衝突までの推定距離を用いて、エージェントはこの距離 $d_i$ の中で安全に止まることができる速度 $v_i$ を計算しますが、この速度の上限は $v^d$ に制限されます。

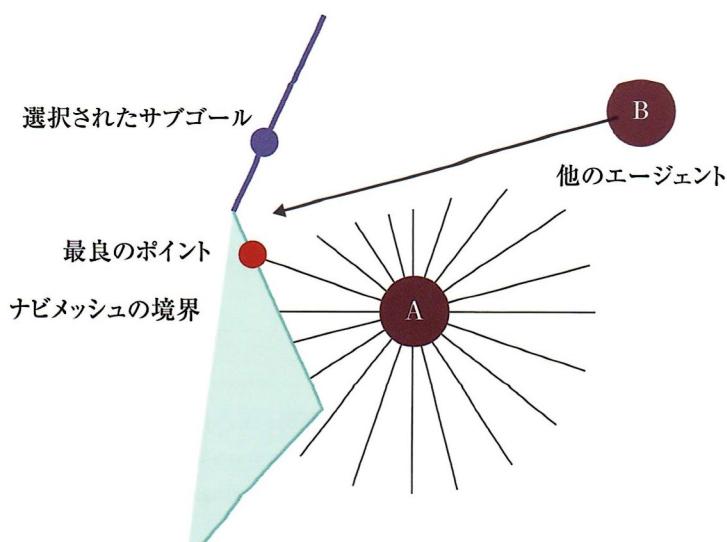


図8 衝突回避

この計算に少し違和感があるとすれば、1つの距離を計算するのに1つの速度を用いており、同じ距離を異なる速度を求めるために使っている点でしょう。しかし最初のステップで、エージェントの要求する速度に基づいて方向を算出しているので、2つめのステップではこの方向が速度を決定します。実際、これらはとても有効な方法です。サンプルを1つ選ぶため、まず選択に条件をつけます。

- 最小のスピードは安全なスピードより小さい  $v_i^{\min} <= v_i$
- 速度  $v_i$  はエージェントが取り得る速度の最小値より大きい
- 距離  $d_i$  はエージェントが移動可能な最小の距離より大きい

この3つの条件を満たす方向のサンプルのうち、サンプルの線分からゴールまでの距離が最小となるものをエージェントが選択します。ゴールはエッジなので、この線分同士の距離が最小となるように決定します。サンプルの中から最近接点までの移動距離によって、ベストな方向を1つ選び出すというわけです。最後に、結果を安定させるアルゴリズムを追加します。

## ☞ グループ制御

移動するものは単体に限りません。「整列した軍隊の行進」から「家族のショッピング」「サバンナの動物の群れ」「みこしを担ぐ人たち」まで、集団固有の運動があります。ステアリングライブラリはグループにも適応し、こうした集団の運動を表現できます。それぞれのエージェントは1つのグループに属しており、各グループのビヘイビアは、取り換え可能なコンポーネントのセットとして定義されます。グループには、いくつでもコンポーネントを加えることができますが、少なくとも1つは所持しなくてはなりません。

図9は、群衆制御の模範的な手法を示したもので、最初のフェーズですべてのグループが同時にアップデートされ、第2フェーズで、すべてのエージェントが同時にアップデートされます。これは、グループがそのメンバーに安全かつ自由にアクセスできることを意味しています。各グループでは、個々の入力（インプットビヘイビア）とグループビヘイビアがペアを組むことになります。「PathPlanningFlockMover」はパスを保持し、インプットビヘイビアによるブロッキングの方程式（Reynolds [4]）を守るような個々のエージェントへの力を計算します。

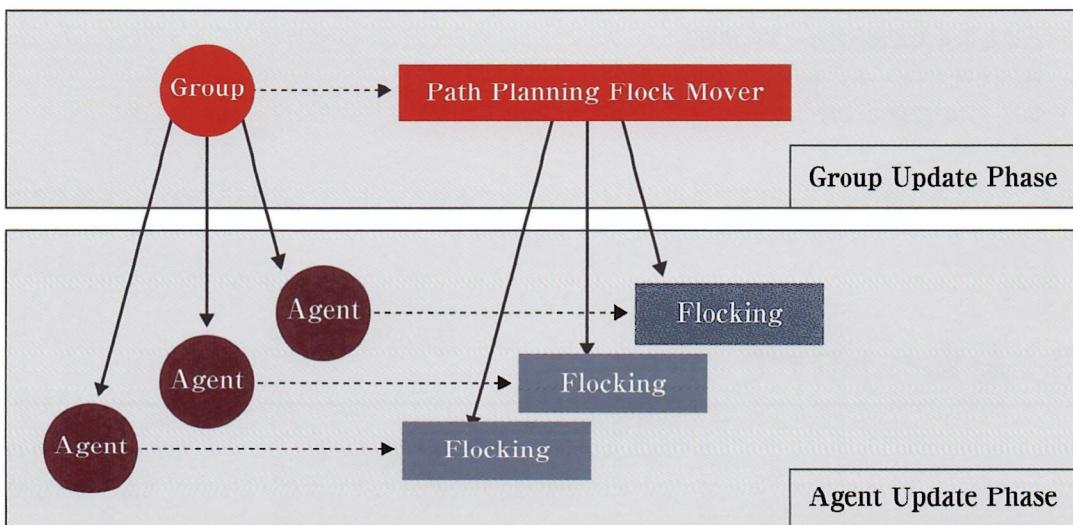


図9 グループのプロセッシングの例

以下のグループの例は、2人のNPCの婦人が水辺でステアリングしながら歩いている状況です（図10）。



図10

オルティシェにおいて散策しているグループ

## ■ シミュレーションAIの情報を引き出す

FFXVにおいて運動はアニメーションドリブン（主導）で決められている一方、移動の仕方はAIが決定します。これらの決定に必要なデータはシミュレーションによって取得されます。つまり、それぞれのキャラクタータイプが、インプットの速度に対してどのように反応するかをシミュレートして記録するのです。このデータは解析され、ランタイムで使用するモデルの生成に用いられます。この詳細は13章「ログインと可視化」で説明します。ランタイムにおけるモデルは、以下のような情報を持っています。

- ・エージェントの最高・最低速度
- ・止まるまでの距離（スピードに依存）
- ・旋回半径（スピードに依存）
- ・スピードの減衰の仕方

ただ、このような情報は、あらゆるアニメーションドリブンの運動アプローチで必要というわけではありません。制御する上で、どの程度の精度が必要かによって変わります。

## 3. 未来へ向けて

衝突回避のようなステアリングの課題を解決する上で、計算速度の限界が壁となります。これが改善されれば、アニメーションとAIの間の溝を埋めることができるはずです。それぞれの身体部位まで考慮すると山積みの問題を抱えてしまうので、ロコモーション（移動）の観点から、全身の身体運動をプランニングすることが最も有望な解決方法と言えるでしょう。

現在のプランニング技術では速度が遅く、ロボット的な制御しかできないため、自然には見えません。しかし、AIが人間の動作を学ぶことができたらどうでしょうか？近年の機械学習を用いれば、完全に環境にマッチするアニメーションを生成することも可能です。モーションフィールドの技術は、アニメーションに新しい可能性を開くと予想されます。

本章で説明したように、人工知能技術はより一層、開発プロセスの中に浸透していくことになります。FFXVの人工知能技術は「キャラクターたちのモーションを解析し、それに基づいたモデルを構築する」という限定的な使用法にとどまっていました。しかし将来的には、デザイナーが本質的な仕事に集中できるよう、ワークフローの大部分にAIが用いられるようになるでしょう。我々が夢見てきた息吹を感じられる世界を実現するため、オプションとしてではなく、全面的な適用が必要なのです。そのような未来を想像するとワクワクしますね。

**我々の目標は、単にデザイナーの仕事をAIに置き換えることではなく、よりクリエイティブな仕事に集中してもらうことです。**例えば、ナビゲーションシステムはキャラクターが動き回ることを可能にするだけでなく、群衆の流れを個々のキャラクターの最短パスの集合以上のものになるように表現します。そのため「スマートウェイポイント」という、ナビゲーションメッシュよりも高い優先度を持つ点列を定義し、各キャラクターの移動に割り当てています。これにナビゲーションメッシュを用いると、環境に沿ったパスになります。環境の変化に合わせて自動的に対応できるので、デザイナーの作業工程の効率化につながります。また、ポイントと幅は群衆のコントロールにも役立つことでしょう。

パス検索は、内部的には有効半径と、より発展したスムージング技術が実装されています。このようなクエリですべてのNPCを動かすには負荷が高いますが、パスを共有するシステムであれば使用可能です。さらに、人々が混雑を避けてバラバラと方向に散開させることもできます。

図11はスマートウェイポイントの活用事例です。黄色で示しているのは双方向のパスです。障害物を迂回するように計算されたパスには、一定の幅を持たせてあります。



図11 スマートウェイポイントとナビメッシュ上パス(ピンク色)

## ● 参考文献

- [1] Jur van den Berg, Stephen J. Guy, Ming C. Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In International Symposium on Robotics Research, 2009.
- [2] J. Guzzi, A. Giusti, L. Gambardella, and G. A. Di Caro. Human-friendly robot navigation in dynamic environments. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 423\_430, Karlsruhe, Germany, May 6\_10, 2013.
- [3] J. Guzzi, A. Giusti, L. Gambardella, and G. A. Di Caro. Local reactive robot navigation: a comparison between reciprocal velocity obstacle variants and human-like behavior. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), pages 2622\_2629, Tokyo, Japan, November, 3\_7, 2013.
- [4] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In Computer Graphics, pages 25\_34, 1987.
- [5] Ben Sunshine-Hill. Perceptually Driven Simulation. PhD thesis, University of Pennsylvania, December 2011. Publicly accessible Penn Dissertation.
- [6] Ben Sunshine-Hill and Bobby Anguelov. Managing the movement: Getting your animation behaviors to behave better. In GDC, 2013. URL <http://www.gdcvault.com/play/1018060/Managing-the-Movement-Getting-Your>.

PART 1

基礎編  
ナビゲーション  
システム編  
【3章】

TALK 1

A I 座談会  
プログラマー編

PART 2

コンテンツ編

TALK 2

A I 座談会  
デザイナー編

PART 3

未来編

TALK 3

A I 座談会  
チームリーダー編

# CHAPTER:4

## AIとアニメーション



今村 紀之

Noriyuki Imamura

FINAL FANTASY XVではリード アニメーション エンジニアとして開発に従事。アニメーター向けのツールやワークフローの整備、ゲームにおけるアニメーション関連の仕様策定・実装など全般的に担当。