

2019年度 卒業論文

生態系シミュレーションを用いた  
NPC 分布手法に関する研究

指導教員：渡辺 大地 准教授

メディア学部 ゲームサイエンス  
学籍番号 M0116295  
柳 晴樹

2020年2月

2019年度 卒業論文概要

論文題目

生態系シミュレーションを用いた  
NPC 分布手法に関する研究

メディア学部

学籍番号：M0116295

氏  
名

柳 晴樹

指導  
教員

渡辺 大地 准教授

キーワード

人工生命, 人工知能, 生態系シミュレーション, 自動生成, NPC 分布

近年のゲーム開発では技術向上によるグラフィックの向上やステージの広大化など、作業量の増加により開発が大規模化している。そのため、人工知能などを用いて一部の作業を自動化することで作業量を軽減するなどの試みが行われている。

開発技術の向上により現実と見間違ふ程のグラフィックを作成可能になったため、ゲーム内の世界にも現実感を持たせる必要がある。現在のゲーム開発において NPC の配置を自動で行う場合、地形などの環境を基に配置を行うことで NPC の生態を表現している。しかし、現実の生態系において生物同士の関係性も生態系を形成する重要な要素であり、ゲームにおいても生物の関係に着目した NPC の配置を生成する必要がある。

ゲームにおける NPC などの架空の生物は、現実の生物の生態や生態系を参考に作成している。そのことから、現実を基に作成した生態系シミュレーションを用いることで、ゲームにおける架空の生物に生物同士の関係性を持たせることが可能である。本研究ではゲームにおける架空の生物の生態系を生態系シミュレーションを用いて表現することで、生物同士の関係性からなる配置を生成することを目標とした。

本手法では植物、草食動物、肉食動物にパラメータを与え食物連鎖を基とした生態系シミュレーションを作成した。また、本手法ではオオカミがもつ他群の縄張りに侵入しないという特性を生態系シミュレーションに組み込むことで NPC の配置を生成した。検証では、既存のゲームにおける NPC 配置と自動生成手法としての有効性について考察を行った。その結果、特定の NPC 配置を再現可能であった。また、初期配置からの大きな逸脱無く、NPC 配置の自動生成を確認した。

# 目次

第 1 章	はじめに	1
1.1	研究背景	1
1.2	本研究における生態系について	3
第 2 章	提案手法	6
2.1	生態系シミュレーションについて	6
2.2	食物連鎖について	6
2.3	地形データについて	7
2.4	オブジェクトのパラメータについて	8
2.4.1	生物の大きさについて	9
2.4.2	速さについて	9
2.4.3	餓死と空腹について	10
2.4.4	寿命について	10
2.4.5	生物の生成について	11
2.5	縄張りについて	11
2.6	植物について	12
2.7	草食動物について	13
2.8	肉食動物について	16
第 3 章	シミュレーションの実行と考察	20
3.1	実行環境	20
3.2	シミュレーションの実行内容と結果	21
3.2.1	本シミュレーションの目標配置	21
3.2.2	オブジェクトについて	22
3.2.3	既存のゲームにおける NPC 配置再現の検証と結果	23
3.2.4	NPC 配置の自動生成手法としての有効性の検証と結果	26

3.3	考察 . . . . .	29
第4章	まとめ	31
	謝辞	33
	参考文献	34

# 第 1 章

## はじめに

### 1.1 研究背景

近年のゲーム開発では技術向上により、現実と見間違ふほど写実的な 3D キャラクターや広大なマップを用いたオープンワールド系のゲームを制作可能である。しかし、グラフィック技術の向上や広大なマップなど作業量の増加から開発は大規模化する。そのため作業量を減らすために開発を効率よく進める必要がある。現在では人工知能を利用する試み [1][2] など、マップやオブジェクトの自動生成などによる開発の効率化が行われている。三宅 [3] はゲーム開発における人工知能は開発を効率化以外にも、敵などの NonPlayerCharacter(以降 NPC と呼称する) にあらかじめ命令を設定するのではなく、現在の状況などの情報を基に意思決定を行い行動など人工知能の有効性を述べている。ゲーム開発における人工知能は開発の効率化や NPC の意思決定など、大規模化しているゲーム開発にとって必要である。

ゲームにおける現実感について、日本ではゲームに対して現実感をあまり求めない傾向にある。しかし、ゲーム開発の技術向上によるグラフィックの向上など現実と見間違ふ程の表現が可能になったことから、日本のゲームにも現実感を持たせる必要がある。現在の開発では、モンスターなどの NPC に生き物としての振る舞いや生態を人工知能を用いて表現することで現実感を与え

る事例がある。森川 [4] は人工知能を用いて NPC に生き物らしい振る舞いを試みることを述べている。三宅 [5] はゲーム内の世界を NPC が認識することで各 NPC ごとに異なる行動を行っていることを述べている。

人工知能を利用することで NPC の生態などを振る舞いにより現実感を与えることはできるが、その配置については地形などの環境を基に行っており生物同士の関係性は含まれてはいない。現在の NPC の自動配置はキャラクタと地形の関係性を重視しているが、ゲームにおける NPC はゲーム内の世界における生物的な役割があるため生物の関係性にも着目する必要がある。ゲームにおける NPC において肉食動物や草食動物としての役割がある場合、その生物が何を食べるのか、敵対している生物がいるのかなど生物としての設定がある。そして現在のゲーム開発において生物などの NPC について生態に注目する開発事例があることから、生物の生態に着目した NPC の配置は必要である。徳田 [6] はモンスターハンターワールドではモンスターの生態系を重視するため、モンスターの設定を基に生態系を再現したモンスター配置をしたことを述べている。スクエアエニックスが販売している FinalFantasyXV ではモンスターを制作する際に実際に動物の生態を参考に、NPC の動きや配置を設計しゲーム内に架空の生態系を形成している。

生物同士の関係性を表す手法として生態系シミュレーションを用いる人工生命という考え方がある。人工生命とは生物の行動や生態系などを疑似的にシミュレーションすることで、生物特有の結果を得る考え方である。星野 [7] は生命体特有の現象をシミュレーションを用いて生物特有の創発的な行動を得る人工生命について述べている。また、三宅ら [8] は人工生命をゲーム世界に用いることについて述べている。ゲームにおいてモンスターなどの架空の生物は役割があり生物同士の関係性によりゲームの世界を構築している。また、ゲームにおける架空の生物は現実における生物の生態や生態系を参考にしている。そのことから、現実の生態系を基に作成した生態系シミュレーションを用いることで、ゲームにおける架空の生物に生物同士の関係性を持たせることが可能である。本研究ではゲームにおける架空の生物の生態系を生態系シミュレーションを用い

て表現することで、生物同士の関係性からなる配置を生成することを目標とした。

本研究では現実の生態系を参考に、肉食動物や草食動物など仮想生物による生態系シミュレーションを作成し、本手法の実行結果から生物同士の関係性からなる配置の生成と自動生成手法としての有効性について検証を行った。その結果、特定の状況において生物同士の関係性からなる配置を生成可能であった。また、配置の自動生成手法として有効であり偶発的な得ることができることが分かった。

## 1.2 本研究における生態系について

本研究ではゲームにおける架空の生物の生態系を生態系シミュレーションを用いることで、生物同士の関係性からなる配置を生成することを目標している。ゲームにおける架空の生物は、現実の生物の生態や生態系などを基に配置をしているため、本研究における生態系シミュレーションも現実の生態系を基に作成する。本研究は生物の関係性による配置を生成することを目標としているが、地形によっても生物の関係性に変化があるため生態系シミュレーションに地形を取り入れる。

生態系シミュレーションには様々な種類のものが存在している。田中 [9] は内湾の生態系におけるプランクトンの動きを流動モデルを用いて生態系シミュレーションを行っていた。また、石川ら [10] はクロレラ、バクテリア、プロトゾアの3種類の微生物による生態系を形成し、その中にコロニーと呼ばれる固有の構造を自発的に発生させるシミュレーションを行っている。坂本ら [11] は3種類の生物が捕食対象の獲物とそれを捕食する天敵の関係を循環的にモデル化している。本研究では生態系の最も基本的な原理の、捕食者と非捕食者からなる食物連鎖とその関係性の生態系ピラミッドを用いる。本研究では坂本らの手法と森脇康介ら [12] の食物連鎖モデルを参考に、食物連鎖による疑似的な生態系を生成するうえで必要な植物、草食動物、肉食動物の3種類からなる生態系シミュレーションを作成する。

ゲームにおいて敵などの NPC は複数種類がマップに配置する。NPC には肉食動物や草食動物などの役割としての区別以外に、強さや個体数など NPC の種類ごとに設定が異なる。そのため本研究では複数の生物を肉食動物や草食動物で分けるのとは別に群れとして分けることにした。現実の生態系は様々な生物により形成されているが、生物の中には縄張りと呼ばれるその生物や群れが占有している地域を形成する種も存在する。縄張りを形成する生物には様々な種類がいるが、中でもオオカミは多くの調査 [13][14][15] が行われている。Kajetan ら [16] は生態系におけるオオカミには階級や集団で行う狩りなど知性的な行動をとると述べている。その中でもオオカミには他の群れのオオカミに襲われることを恐れ他の縄張りに侵入しないという性質があり、それによる無駄な争いを避けている。また、縄張り同士が干渉することで発生する緩衝地帯によって、草食動物が安全に行動できる領域が形成されると述べている。緩衝地帯によりオオカミが必要以上に草食動物を捕食することがなくなり、草食動物と肉食動物であるオオカミが少なすぎず多すぎない安定した生態系を形成することが確認されている。オオカミは縄張りにより各群れごとに生息域が明確に決まっていることから、本研究において架空の生物による生態系を形成しながら縄張りにより配置に必要な各群れの領域を指定する。本研究における縄張りは現実のオオカミと同様に肉食動物に適応するものとした。

オオカミは他の群れと争いを避けるため遠吠えやマーキングにより縄張りの境界を定期的に更新している。また、オオカミが縄張りを更新する頻度などについては、群れにおけるオオカミの個体数と環境に関係がある。本研究の基盤となる生態系シミュレーションでは食物連鎖を用いるため、消費者の捕食量により各生物の個体数が変化する。また、本研究のシミュレーションでは分解者の概念を入れるためフンや死骸といったものも取り入れる。そのため本研究では肉食動物が捕食時に発生するフンをオオカミのマーキングとし、複数のオブジェクト重なりにより肉食動物の縄張りを形成することとした。

生態系シミュレーションを用いた研究 [17][18] では食物連鎖など生物の動きだけではなく、検

証するその地域の正確なデータを用いて対象となる地域の生態系を表現しシミュレートする。しかし、本研究ではゲームにおける架空の生物の生態系を生成することが目的であるため、現実の生態系を再現するシミュレーションのような正確なデータによる生態系は表現しないこととした。マイクロソフト社が販売しているマインクラフトというゲームでは生物ごとの生息域をバイオームと呼ばれる地域データを基に設定し、世界観や生態系を表現している。本研究でもマインクラフトと同じく、現実のような正確なデータではなく山や湖など領域ごとに地形の名称を設定した地形データを基に生物の活動領域を決定する。

# 第 2 章

## 提案手法

### 2.1 生態系シミュレーションについて

本研究の生態系シミュレーションは 2 次元平面上のマップを用意し、その範囲内で植物、草食動物、肉食動物のオブジェクトを用いてシミュレートする。また、シミュレーションでは生物や地形などを事前に設定する。本研究は生態系シミュレーションによる NPC 分布を自動生成することを目的としているため、シミュレーション中は生物の追加や地形の変更などの操作は行わない。

本研究では植物、草食動物、肉食動物を生物オブジェクトとし、それ以外の生物ではないフンや死骸などを非生物オブジェクトと呼ぶこととする。

本研究の草食動物と肉食動物は坂本らのシミュレーションを参考に捕食によって個体数を増やす手法を取り入れる。

### 2.2 食物連鎖について

食物連鎖とは生物における捕食者と非捕食者の関係の繋がりであり生態系を形成する要素の一つである。食物連鎖は植物である生産者、草食動物と肉食動物などの消費者、微生物などの分解者で構成されている。一般的な食物連鎖の例として、植物が草食動物に捕食され、草食動物が肉食動物に捕食され、動物のフンや死骸が分解者によって植物の栄養となる関係がある。

本研究における生態系シミュレーションでは一般的に認知されている植物、草食動物、肉食動物からなる生物を用いて食物連鎖を再現する。また、本研究の食物連鎖では植物が草食動物に捕食され、草食動物が肉食動物に捕食され、肉食動物がしたフンや死骸が植物へ変化する構造を用いている。図 2.1 は本研究で用いる食物連鎖の流れを表している。

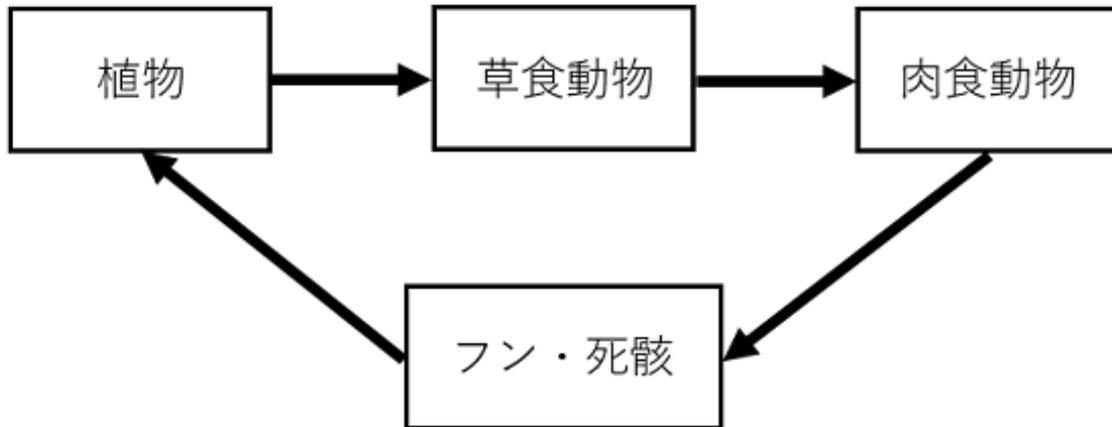


図 2.1 本研究における食物連鎖の流れ図

## 2.3 地形データについて

本研究では生態系シミュレーションにて各生物オブジェクトに生息可能地形を設定することで架空の生態系とゲームにおけるマップに関係性を持たせる。また、本研究における地形は地形ごとに森や山など名称を設定することで表現する。生物オブジェクトは地形ごとに活動可能かを可否を事前に設定する。生物オブジェクトがある地形で活動できる場合はその地形の上で移動できるが、活動できない場合はその地形に侵入することはできない。本研究のシミュレーションでは徳川らの講演で述べた、マップを参考に山、森、湖の3種類の地形データを用いることにした。本研究の地形データは長方形もしくは正方形の範囲を指定し3種類の内から種類を設定することができる。生態系シミュレーションではマップ全体に地形データを設定する。

本研究における生態系シミュレーションではマップにおいて地形データの可視化のため、地形

の種類ごとに色を設定する。図 2.2 は本研究の生態系シミュレーションにおける各地形データの色を表した。

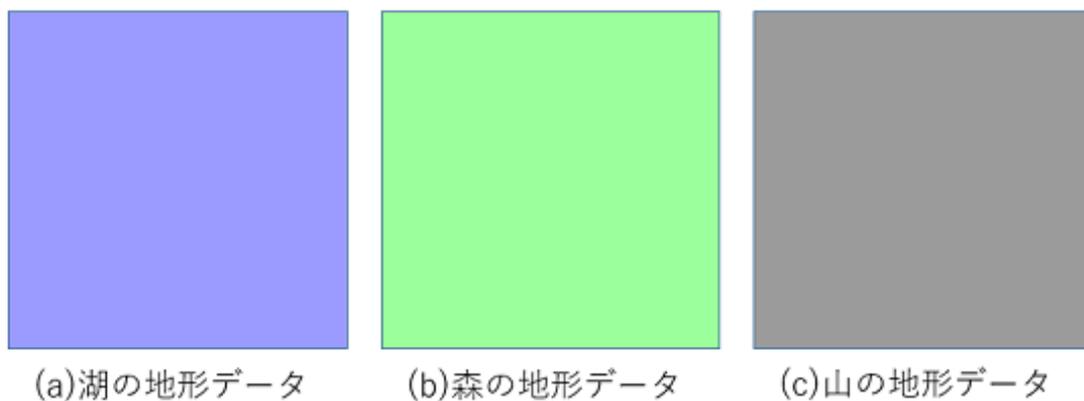


図 2.2 各地形データの色

## 2.4 オブジェクトのパラメータについて

本研究におけるシミュレーションでは実際の生態系を基にしているため、本研究における生物では餓死や寿命など生物として必要な要素を取り入れる。そのため生物オブジェクトや非生物オブジェクトに様々なパラメータを設定する必要がある。表 2.1 は本研究で使用するパラメータのまとめである。この節では本研究の生態系シミュレーションを行う上で生物オブジェクトに必要なパラメータについて述べる。

表 2.1 本研究で使用する生物のパラメータ

名称	説明
大きさ	生物の大きさを表す値
速さ	生物の速さを表す値
満腹値	空腹の度合いを表す値
捕食値	被捕食時に値を満腹値に加算する値
寿命値	その生物の寿命を表す値
老化値	寿命値から常に減少する値
生産値	生物が増殖までに必要な捕食回数

### 2.4.1 生物の大きさについて

本研究の生態系シミュレーションでは各オブジェクトとの当たり判定をとるため生物オブジェクトに大きさを設定する。生物オブジェクトは正円状のオブジェクトとし、半径は事前に設定する。本研究では各生物オブジェクトの大きさを半径の大きさとする。各生物の半径は植物は  $P_p$ 、草食動物は  $P_h$ 、肉食動物は  $P_c$  と呼称する。

### 2.4.2 速さについて

草食動物や肉食動物は生きるために移動することから、本研究の生態系シミュレーションでは生物オブジェクトに速さを設定し常に等速で移動する。各生物オブジェクトは種類ごとに速さを保持しており植物は  $V_p$ 、草食動物は  $V_h$ 、肉食動物は  $V_c$  を保持している。また、生物オブジェクトは常に等速で移動するため草食動物の速さが肉食動物の速さより大きくなると、肉食動物が捕食できなくなってしまい食物連鎖が成り立たなくなる。そのため生物オブジェクトの速さは最上位の捕食者である肉食動物の速さ  $V_c$  を最大値とし式 (2.1) の大小関係とする。また、植物の速さ  $V_p$  は現実の植物が移動しないことから  $V_p = 0$  とし移動しない生物オブジェクトとする。

$$V_h \leq V_c. \quad (2.1)$$

### 2.4.3 餓死と空腹について

現実の生物が捕食しなければ餓死してしまうことから、本研究の生態系シミュレーションにおいても生物オブジェクトは捕食しなければ餓死するものとした。ただし、植物は光合成と地面から栄養を得るため餓死することはないとした。草食動物と肉食動物は満腹値  $L$  を保持しており、空腹の度合いを表している。植物と草食動物は被捕食時に捕食者に与えるエネルギー量を保持し、これを捕食値  $E$  と表す。本研究ではシミュレーションの実行からの時刻を  $t$  とし、各生物の満腹値  $L$  を式 (2.2) で計算し、 $L_t \leq 0$  となったとき餓死する。

$$L_t = \begin{cases} L_{t-1} + E - 1 & (if \text{捕食時}). \\ L_{t-1} - 1 & (else). \end{cases} \quad (2.2)$$

現実における草食動物や肉食動物は満腹時など捕食の必要がない場合には捕食行動を行わない。そのため本研究の生態系シミュレーションにおいても捕食が必要かどうかを判定する。本研究において草食動物と肉食動物は  $L_t \leq L_0/2$  のときに空腹とする。そして、本研究では草食動物や肉食動物は空腹のときに捕食するための行動を行う。

### 2.4.4 寿命について

本研究の生態系シミュレーションでは草食動物と肉食動物に寿命を設定する。本研究における草食動物と肉食動物はシミュレーション開始時の満腹値  $L_0$  と同じ値の寿命値  $Q$  と老化値  $G$  を  $G > 0$  の条件で保持している。現実の生物の老化の進み方と空腹で餓死になるまでの時間が異なることから、本研究では老化値  $G$  を設定することで違いを表現する。寿命値  $Q$  を式 (2.3) で計算し、 $Q_t \leq 0$  となったときその生物オブジェクトは寿命に達したとして消滅する。

$$Q_t = Q_0 - Gt. \quad (2.3)$$

## 2.4.5 生物の生成について

本研究の生態系シミュレーションでは生物オブジェクトは捕食することで繁殖を行う。そのため本研究の生物オブジェクトは捕食回数を記録し、指定した回数の捕食を行った際に個体数を増加する。草食動物と肉食動物は生産値  $B$  とシミュレーション中に捕食した回数  $C$  を保持している。生産値  $B$  はシミュレーション前に  $B > 0$  となるように値を設定し、 $C$  はシミュレーション中に捕食したとき 1 ずつ加算し捕食回数を記録する。シミュレーションでは  $B = C$  となったときに、その生物オブジェクトと同じ生物オブジェクトを同じ座標に 1 体生成し、 $C = 0$  とする。

## 2.5 縄張りについて

本研究における生態系シミュレーションでは食物連鎖だけではなくオオカミの縄張りを利用することで NPC 分布を行うことから、縄張りを形成する機能を生物オブジェクトである肉食動物に与える。また、本研究では複数種類の肉食動物による生態系シミュレーションを用いて NPC 分布を行うため、他の縄張りに侵入しないというオオカミの習性を利用する。Kajetan らは縄張り同士が重なることで形成される緩衝地帯には肉食動物であるオオカミが他種の縄張りに侵入できないことから草食動物の安全地帯となると述べている。また、Kajetan らは緩衝地帯によって肉食動物の捕食が制限されるため、草食動物が捕食されすぎないようにする機能もあると述べている。

本研究ではオオカミの縄張りの機能だけではなく、縄張りの生成方法についてもオオカミのマーキングを参考に行う。本研究では縄張りによる肉食動物の群れごとの生息域を決める必要があるため、シミュレーションでは肉食動物の群れごとによるそれぞれの縄張りを形成する。本研究における肉食動物のマーキングは、草食動物を捕食した際にフンを生成し同時に縄張りも生成する。縄張りは正円状のオブジェクト (以降縄張りオブジェクトと呼称する) で表現し、縄張りの半径  $R_z$  をシミュレーション前に事前に設定する。図 2.3 は捕食時の縄張りオブジェクトの生成

を表している。

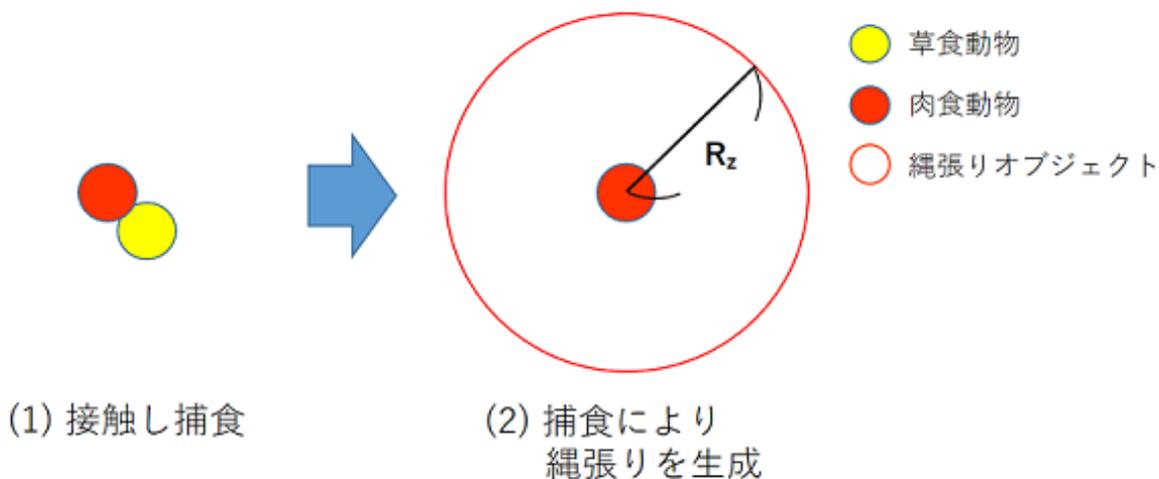


図 2.3 縄張りオブジェクトの生成

オオカミの縄張りは一時的な効果しかなく定期的に更新する必要があるため、本研究においても縄張りは時間で消滅するオブジェクトとし更新が必要なものとした。縄張りオブジェクトが消滅するまでの残りの数値を  $D$  とし、式 (2.4) で計算する。 $D_t \leq 0$  のときそのオブジェクトを消滅する。

$$D_t = D_{t-1} - 1. \quad (2.4)$$

## 2.6 植物について

本研究のシミュレーションにおいて植物は草食動物に捕食されるオブジェクトとして設定した。満腹値や老化値といった数値を設定せず、捕食値のみを保持したオブジェクトとする。

植物は生物のフンや死骸を分解者が分解することで栄養として取り入れ増殖する。そのため本研究においては他の生物オブジェクトとは違い捕食はせず、肉食動物のフンや死骸といった非生物オブジェクトを基に生成する。本研究では非生物オブジェクトはすぐに植物に変化するのでは

なく、消滅するまでの時間を設定し消滅したときに植物に変化する。非生物オブジェクトが消滅するまでの値を  $F$  とし、式 (2.5) で計算し、 $F_t \leq 0$  のとき消滅し植物を生成する。

$$F_t = F_{t-1} - 1. \quad (2.5)$$

また、本研究ではフンや死骸が生成する植物のオブジェクト数を 4 個とする。図 2.4 は実際にシミュレーションで肉食動物の死骸が植物に変化する様子を表しており、肉食動物のフンも同じ過程で植物に変化する。

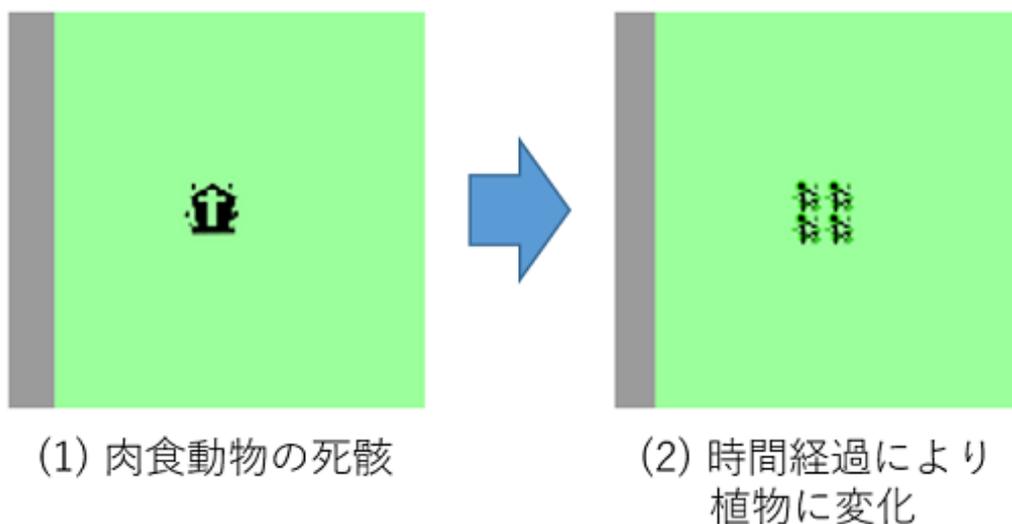


図 2.4 死骸の植物への変化

## 2.7 草食動物について

草食動物は肉食動物に捕食されるが植物とは違い移動する。草食動物は捕食値以外に速さや、満腹値、老化値、生産値を保持している。また、草食動物は肉食動物の縄張りの内側にあるときにどの肉食動物の縄張りにいるかを分布番号  $M_h$  として保持する。そのパラメータをまとめたも

のが表 2.2 である。

表 2.2 草食動物のパラメータ

名称	説明
大きさ	生物の大きさ
速さ	生物の速さ
満腹値	常に減少し 0 になった場合は死亡し死骸になる
老化値	寿命値から常に減少する値
生産値	値分の捕食を行ったとき同じ種類の生物を 1 体生成する
捕食値	被捕食時に値を満腹値に加算する
分布番号	どの縄張りに属しているかを表す

本研究における生態系シミュレーションでは草食動物に自由行動、空腹行動、回避行動の 3 種類の状態を用意し、状態変化により草食動物の行動を表現する。

シミュレーション開始した直後は草食動物は自由行動を行っており、上下左右の 4 方向にランダムに移動する。本研究の生態系シミュレーションにおける自由行動は現実における生物が空腹ではないときの不規則な行動を表している。また、本研究では現実の生物がどのように動くのか表現する必要はないため上下左右の 4 方向ランダムに移動することとした。満腹値が空腹値の値以下になった場合は空腹状態となり、植物を捕食するために行動する。草食動物は自分の大きさである半径  $P_h$  を 10 倍とした索敵範囲  $S_h$  を保持しており、草食動物の位置を  $\mathbf{X}$ 、植物の位置を  $\mathbf{Y}$  としたとき式 (2.6) で索敵範囲の内側に植物いるかを判定する。

$$|\mathbf{X} - \mathbf{Y}| \leq S_h. \quad (2.6)$$

本研究の生態系シミュレーションでは空腹状態のときに索敵範囲内に植物がいる場合は、最も近い位置にいる植物を捕食する。索敵範囲  $S_h$  の内にある草食動物に最も近い植物の位置を  $\mathbf{Y}_m$  とする。草食動物の速さ  $V_h$  と、草食動物の位置  $\mathbf{X}$  から植物の位置  $\mathbf{Y}_m$  に向かう単位方向ベクト

ルを  $\mathbf{T}_h$  とし式 (2.7) で植物の方向に移動する。

$$\mathbf{X}_t = \mathbf{X}_{t-1} + V_h \mathbf{T}_h. \quad (2.7)$$

回避状態は肉食動物の位置を  $\mathbf{Z}$  とした場合に式 (2.8) を用いて索敵範囲  $S_h$  内にいる場合に、肉食動物から逃げる行動をとる状態である。

$$|\mathbf{X} - \mathbf{Z}| \leq S_h. \quad (2.8)$$

回避状態において索敵範囲  $S_h$  の内にある草食動物に最も近い肉食動物の位置を  $\mathbf{Z}_i$  とする。草食動物の位置  $\mathbf{X}$  から肉食動物の位置  $\mathbf{Z}_i$  の反対方向に向かう単位方向ベクトルを  $\mathbf{K}$  とし式 (2.9) を用いて肉食動物から逃げる行動をとる。図 2.5 は草食動物の空腹状態のときの植物へ向かう動きと回避状態の時の肉食動物から逃げる動きを表した。

$$\mathbf{X}_t = \mathbf{X}_{t-1} + V_h \mathbf{K}. \quad (2.9)$$

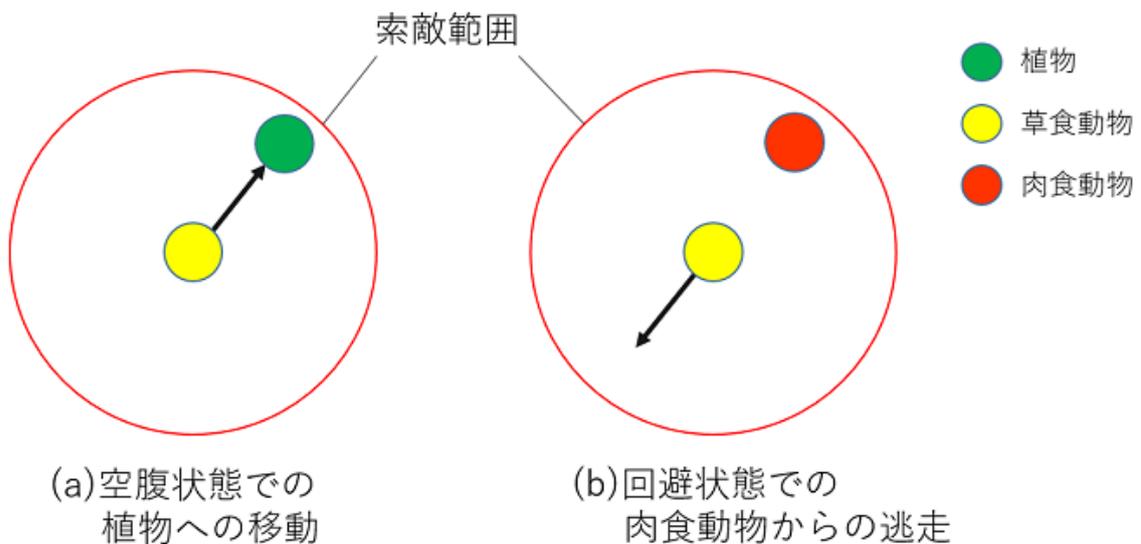


図 2.5 草食動物の空腹状態・回避状態の行動図

本研究における草食動物の回避行動は常に機能しているのではなく、自由行動時にのみ適応することとしている。理由として、現実の生物は生きるために捕食することが必要なため危険を冒しても捕食する必要があることから、本研究では空腹状態のときに回避行動させないことで表現している。図 2.6 は状態の遷移を表した図である。

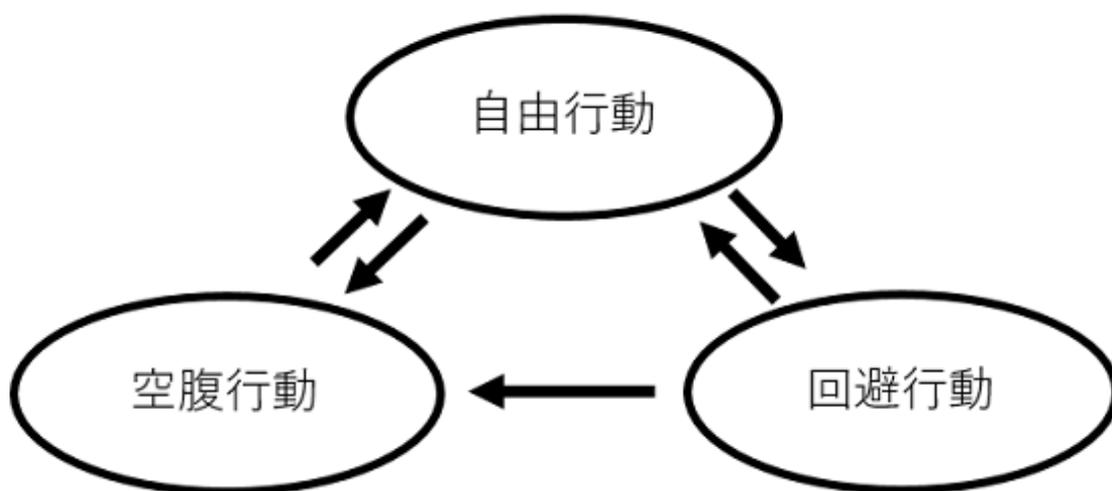


図 2.6 草食動物の行動アルゴリズム

## 2.8 肉食動物について

本研究におけるシミュレーションでは肉食動物のパラメータは草食動物のものと同じであるが、肉食動物を捕食する生物は本研究ではいないため捕食値は設定しない。また、肉食動物ごとの縄張りを識別するため縄張り番号  $M_c$  として保持する。表 2.3 は肉食動物に適応するパラメータをまとめたものである。

表 2.3 肉食動物のパラメータ

名称	説明
大きさ	生物の大きさ
速さ	生物の移動速さ
満腹値	常に減少し 0 になった場合は死亡し死骸になる
老化値	寿命値から常に減少する値
生産値	値分の捕食を行ったとき同じ種類の生物を 1 体生成する
縄張り番号	縄張りの識別番号を表す

肉食動物は基本的には草食動物と同じように自由行動、空腹行動、回避行動の 3 種類の状態を用意し、状態変化により行動を表現する。また、自由行動については草食動物と同じように上下左右の 4 方向にランダムに移動する。1.2 節でオオカミは他の群れのオオカミに襲われることを恐れ他の縄張りに侵入しないという性質を述べた。そのため本研究のシミュレーションでは肉食動物は縄張りに侵入したときのみ回避行動をとり他の縄張りに居続けないようにした。

肉食動物は 2.7 節で記した草食動物と同様に自分の大きさである半径  $P_c$  を 10 倍とした索敵範囲  $S_c$  を保持しており、式 (2.10) で索敵範囲の内側に草食動物がいるか判定する。

$$|\mathbf{Z} - \mathbf{X}| \leq S_c. \quad (2.10)$$

ただし、本研究の生態系シミュレーションにおいて肉食動物はオオカミと同じように縄張りを持っており、他の肉食動物の縄張りに侵入しないという性質がある。そのため肉食動物は自分の縄張りか他の肉食動物の縄張りの外にいる草食動物のみを索敵し捕食する。索敵範囲  $S_c$  の内にある  $M_h = M_c$  か  $M_h = 0$  を満たす草食動物の位置を  $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \dots, \mathbf{X}_n$  とし、肉食動物に最も近い草食動物の位置を  $\mathbf{X}_u$  とする。捕食のための移動については、肉食動物の速さ  $V_c$  と、肉食動物の位置  $\mathbf{Z}$  から草食動物の位置  $\mathbf{X}_u$  に向かう単位方向ベクトルを  $\mathbf{T}_c$  とし式 (2.11) で草食動物の

方向に移動する。図 2.7 は肉食動物の捕食行動を表した。

$$\mathbf{Z}_t = \mathbf{Z}_{t-1} + V_c \mathbf{T}_c. \quad (2.11)$$

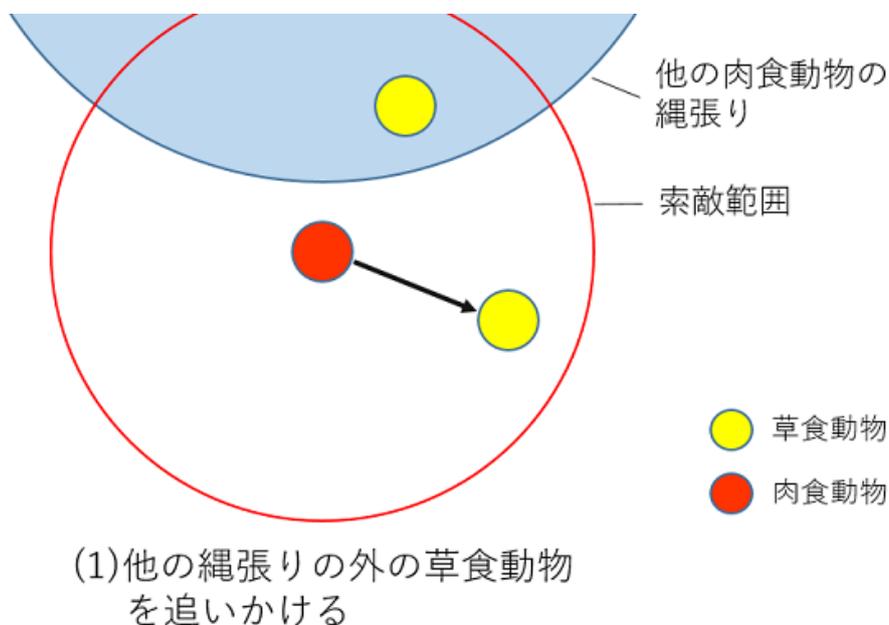


図 2.7 肉食動物の捕食行動

実際のオオカミが形成する縄張りは定期的にマーキングや遠吠えなどによって維持が行われている。本研究のシミュレーションでは 2.5 節にて肉食動物が捕食した際にフンをするをマーキングとし、捕食時に縄張りオブジェクトを生成しその重なりにより縄張り と NPC 分布を生成する。

本研究の生態系シミュレーションにおける肉食動物は縄張りに侵入しないという性質を持たせるため、自由行動や空腹行動のときに回避行動の状態に移れるようにしている。図 2.6 は肉食動物の自由行動、空腹行動、回避行動の遷移を表した図である。

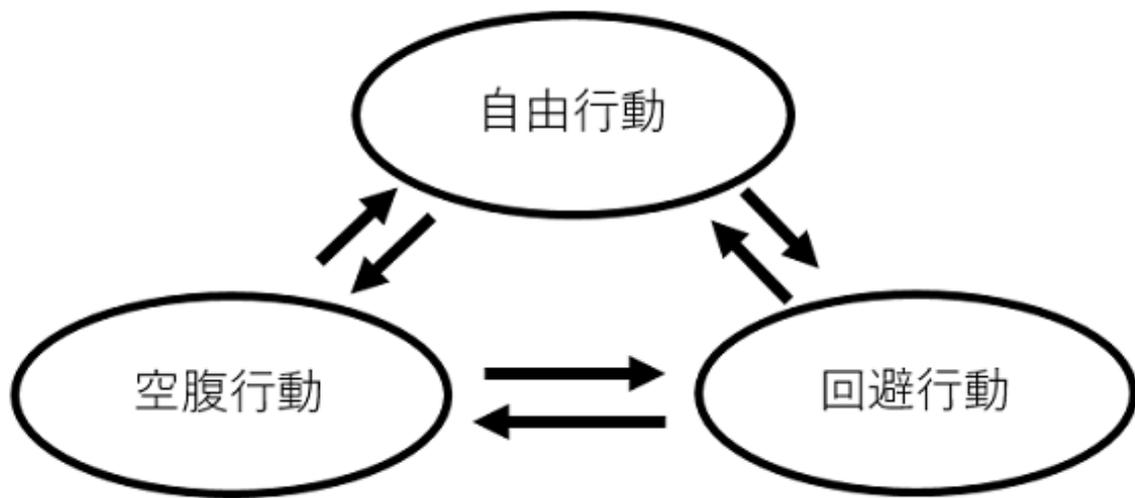


図 2.8 肉食動物の行動アルゴリズム

## 第 3 章

# シミュレーションの実行と考察

### 3.1 実行環境

本研究では VisualStudio2019 と DX ライブラリというゲームライブラリを使用し、生態系シミュレーションを作成した。本研究の生態系シミュレーションではマップを一辺の長さが 1000 の正方形とする。また、マップに適応する地形データは一辺の長さが 200 の正方形として縦横に各 5 個ずつ計 25 個を設定する。図 3.1 はシミュレーションのマップに地形データを設定した例である。

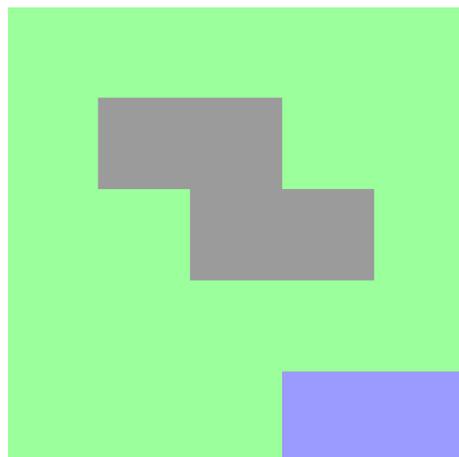


図 3.1 生態系シミュレーションのマップ設定例

## 3.2 シミュレーションの実行内容と結果

### 3.2.1 本シミュレーションの目標配置

本研究はゲームにおける架空の生物の生態系を生態系シミュレーションを用いて表現することで、生物同士の関係性からなる配置を生成することを目標にしている。本研究では生態系を表現するために、植物、草食動物、肉食動物からなる生物に食物連鎖の関係性を持たせた。本研究では肉食動物が生成する縄張りオブジェクトをその生物が活動する領域とした。そして各生物が保持している活動できる領域をその種の生物を配置する領域とした。本研究では2項目について検証を行う。

1つ目は、本研究では食物連鎖や縄張りなど特定の要素が必要なゲームにおいて、生物同士の関係性からなる配置を生成することができるかを検証する。本研究では既存のゲームにおいて生態系や縄張りなどによる配置をしているものと比較することで検証することとした。徳田のモンスターハンターワールドにおけるモンスターの配置は生態系や生物の生態などに着目しており、ゲーム内でも縄張り争いなど生物に縄張りがある。そのため、徳田のモンスターハンターワールドの配置は、本手法の食物連鎖や縄張りと同様な考えで配置を決定しているといえる。本研究では、徳田のモンスターハンターワールドのモンスターの配置を本手法で再現し比較することで配置を生成できているかを検証する。

2つ目に自動生成の手法として有効性を示す。本手法において自動生成を行う前に生物の初期位置やパラメータを設定する必要がある。本手法の初期配置は配置を生成する際に、生物の大まかな配置領域を決める指標である。そのため、生物の初期配置から離れた位置にその生物の領域が生成されることは、ゲームにおいてNPCの配置が予定していた位置から離れていることになりゲームのデザインを損なってしまう。そのため、生物が初期配置を離れて生物の領域を生成しないか検証することで自動生成として有効性があるかを判断する。

### 3.2.2 オブジェクトについて

本研究の生態系シミュレーションではオブジェクトの大きさを固定し、シミュレーションを行った。各生物オブジェクトは正円状のオブジェクトであり、植物の半径  $P_p$  を 5、草食動物の半径  $P_h$  を 8、肉食動物の半径  $P_c$  を 10 としシミュレーションを行った。

本研究の生態系シミュレーションではオブジェクトを可視化する為、各オブジェクトごとに画像を適応しシミュレーションを行った。また、肉食動物は複数種類でシミュレーションを行う為、適応画像は肉食種の物を同一で使用し、色を種別によって変更した。各肉食動物が形成する縄張りオブジェクトは、自身の色と同一とする。図 3.2 は生態系シミュレーションの各オブジェクトをまとめた画像である。各オブジェクトに適応する画像の大きさは、それぞれオブジェクトの大きさに合わせることにした。また、死骸は基となった生物の大きさ、フンは植物に大きさを合わせて画像を適応した。



図 3.2 各オブジェクトの画像

本研究のシミュレーションにおいて肉食動物の縄張りオブジェクトとフン、死骸は、全てのシミュレーションで同じパラメータを使用する。肉食動物の縄張りの半径  $R_c$  を 100、消滅するまでの残りの数値  $D$  を 2400 とした。また、フンや死骸が消滅するまでの数値  $F$  を 300 とし本研究ではシミュレーションを行った。

### 3.2.3 既存のゲームにおける NPC 配置再現の検証と結果

既存のゲームにおける NPC 配置再現の検証では徳田のモンスターハンターワールドにおけるモンスターの配置を参考に生物やマップの設定を行い、本研究の生態系シミュレーションを実行した。本手法におけるマップや生物の設定は徳田のモンスターハンターワールドにできる限り近づけることで、比較した際に条件が異なることにより結果に変化が起きないようにした。

徳田のモンスターを配置したマップは、中心に巨大な樹木が立っておりそれを囲う形で森が広がっている。また、マップの右側には海も含まれている。既存のゲームにおける NPC 配置再現の検証では中心に山の地形データを設定し、周りを森で囲い右側に湖の設定しマップを再現した。図 3.3 は既存のゲームにおける NPC 配置再現の検証で使用したマップの画像である。

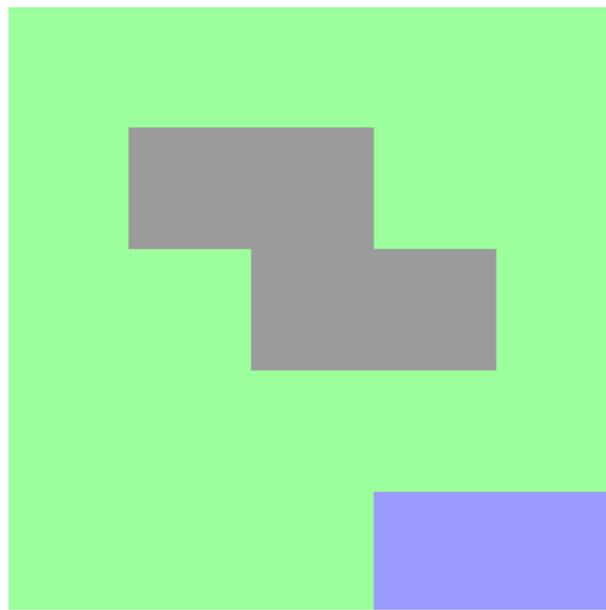


図 3.3 既存のゲームにおける本手法の検証のマップ

生物オブジェクトのパラメータについては、植物は捕食値を 90 とし、草食動物は生命値を 2000、老化値を 20、生産値を 3 とし、捕食値を植物と同じ 90 とした。また、肉食動物は生命値を 1600、老化値を 20、生産値を 4 とした。草食動物の速さ  $V_h$  は 0.5、肉食動物の速さ  $V_c$  は 0.8

とした。表 3.1 は各生物のパラメータを表している。

表 3.1 既存のゲームにおける NPC 配置再現の検証における生物のパラメータ

パラメータ	植物	草食動物	肉食動物
速さ	0	0.5	0.8
生命値	-	2000	1600
老化値	-	20	20
捕食値	90	90	-
生産値	-	3	4

本研究の生態系シミュレーションにおける、各生物の活動できる地形は徳田の用いたモンスターの配置図を参考に設定を行った。徳田のモンスターの配置図では肉食動物であるモンスターは、地上で生活する 2 種と空中と地上の両方で活動できる 1 種の計 3 種類で構成されていた。そのため本研究の生態系シミュレーションにおいて地上は森、空中は山で活動できるものとした。森で生活する 2 種と山と森の両方で活動できる 1 種の 3 種類の肉食動物を設定した。また、植物と草食動物は活動できる地形を山と森に設定した。表 3.2 は各生物の活動できる地形をまとめたものである。

表 3.2 既存のゲームにおける本手法の検証の活動可能地形

生物名称	山	森	湖
植物	可能	可能	不可
草食動物	可能	可能	不可
肉食動物 A	可能	可能	不可
肉食動物 B	不可	可能	不可
肉食動物 C	不可	可能	不可

徳田のモンスターハンターワールドにおけるモンスターの配置図では、マップの中心に空中で活動できるモンスターを配置し他の 2 種がそれを上下で挟むような形で配置している。そのため既存のゲームにおける NPC 配置再現の検証では地形が山でも活動できる肉食動物 A を上下で挟むような形で残りの 2 種を配置する。また、肉食動物は各種 5 体ずつ横に広く配置し植物 400 個

と草食動物 200 体をランダムに配置しシミュレーションを行った。図 3.4 は既存のゲームにおける NPC 配置再現の検証の各生物を配置した画像である。

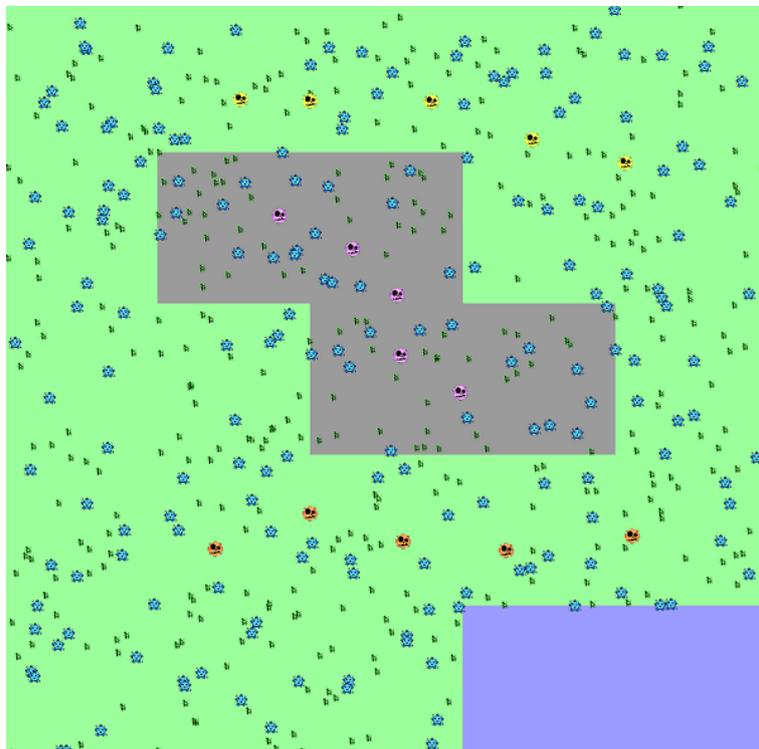


図 3.4 既存のゲームにおける本手法の検証の初期配置

既存のゲームにおける NPC 配置再現の検証ではシミュレーションを同じ条件で 2 回行った。その結果、既存のゲームにおける NPC 配置再現の検証の初期配置では肉食動物 A を肉食動物 B と肉食動物 C で囲んでいたが、シミュレーションの結果も同じように肉食動物 A が囲われる形で 2 回とも生物の領域が生成されていた。また、1 回目と 2 回目の NPC 分布の結果は大きな変化もなく、多少の違いはあるが基本的な形は同じであった。図 3.5 は既存のゲームにおける NPC 配置再現の検証のシミュレーション結果をまとめたものである。

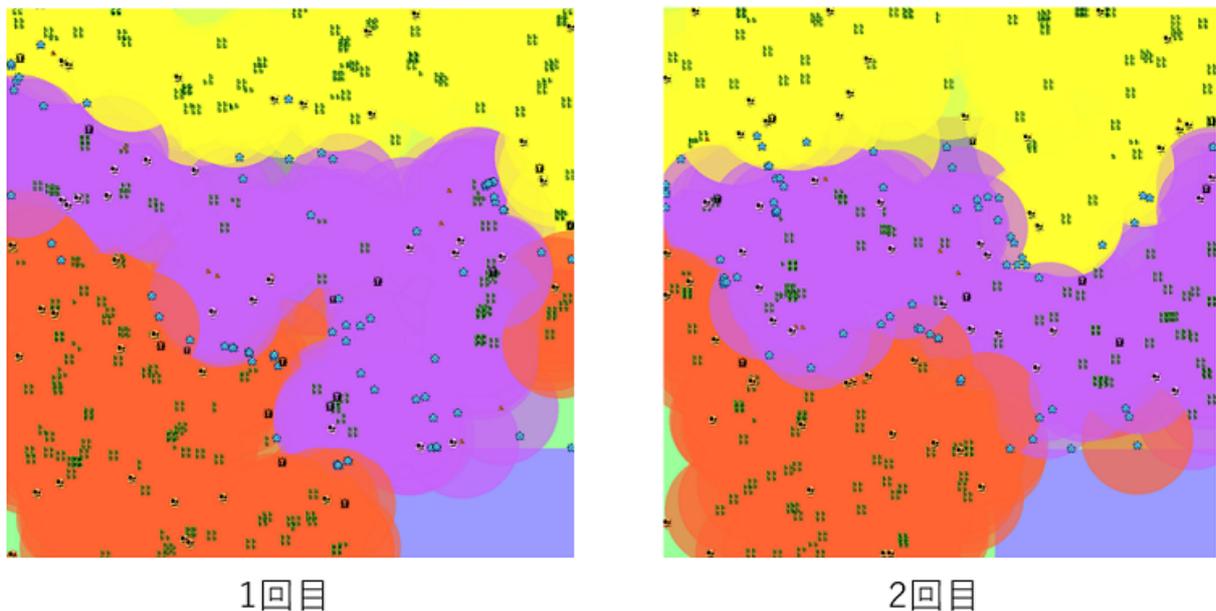


図 3.5 既存のゲームにおける本手法の検証の結果

### 3.2.4 NPC 配置の自動生成手法としての有効性の検証と結果

ゲームにおける NPC 配置では、一定領域内に存在する NPC 種類は複数である事が一般的である。そのため、NPC 配置の自動生成手法としての有効性の検証では複数種の肉食動物を設定し検証を行った。また、種類ごとに間隔を空けて初期配置をすることで生物が自由に活動できるようにした。初期位置から離れた場所に領域を生成しやすくすることで、自由に活動できたとしても初期位置に近い位置で領域を生成することを確かめた。

NPC 配置の自動生成手法としての有効性の検証におけるマップは、森を広く取り山や湖を無作為に設定した。図 3.6 は NPC 配置の自動生成手法としての有効性の検証で使用したマップの画像である。

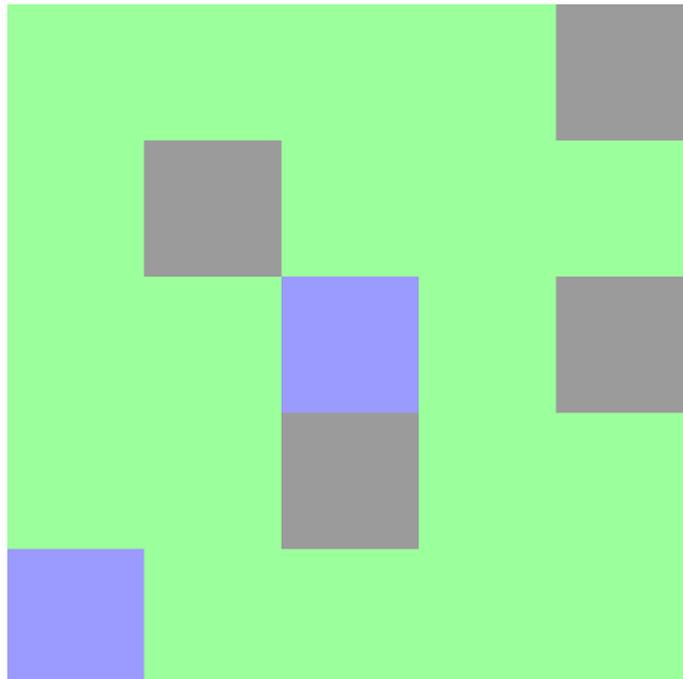


図 3.6 自動生成手法としての有効性の検証のマップ

NPC 配置の自動生成手法としての有効性の検証ではパラメータに違いを出さず複数の種類の肉食動物による NPC 分布を行う。生物オブジェクトのパラメータは既存のゲームにおける NPC 配置再現の検証で使用した設定をそのまま使用する。

生物のパラメータは既存のゲームにおける NPC 配置再現の検証と同じものを使用するが、生物ごとの活動できる地形について肉食動物は全て同じ地域を活動することとした。表 3.3 は各生物の活動できる地形をまとめたものである。

表 3.3 自動生成手法としての有効性の検証の活動可能地形

生物名称	山	森	湖
植物	可能	可能	不可
草食動物	可能	可能	不可
肉食動物	不可	可能	不可

肉食動物について NPC 配置の自動生成手法としての有効性の検証では 5 種類を設定し、各肉食動物ごとに 3 体ずつ配置した。また、植物と草食動物は既存のゲームにおける NPC 配置再現

の検証同様に植物 400 個と草食動物 200 体をランダムに配置した。肉食動物の配置については間隔をあけ、中心には肉食動物を配置しないようにした。また、肉食動物の配置の間隔については無作為に行った。図 3.7 は NPC 配置の自動生成手法としての有効性の検証の各生物を配置した画像である。

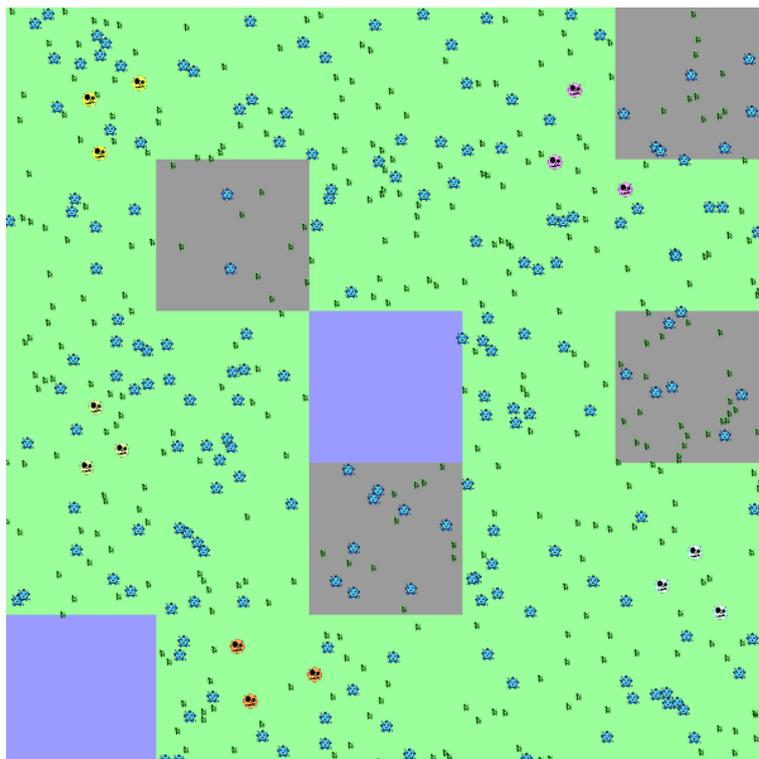


図 3.7 自動生成手法としての有効性の検証の初期配置

NPC 配置の自動生成手法としての有効性の検証も既存のゲームにおける NPC 配置再現の検証と同様に 2 回シミュレーションを行った。その結果、既存のゲームにおける NPC 配置再現の検証の結果と比べて 1 回目と 2 回目の結果に変化があることを確認した。シミュレーションの 1 回目の結果は肉食動物 A と肉食動物 C が肉食動物 E をマップ左端に追いやる形で縄張りを形成していた。しかし、2 回目では逆に肉食動物 E が肉食動物 A を縄張りを右端に追いやる形で縄張りを形成していた。1 回目と 2 回目の結果の違いから、肉食動物を間隔をあけて配置することで NPC 分布の結果が大きく変わってくることが確認できる。しかし、2 回の検証で領域の大きさが

大きく変化する生物はいたが初期位置から離れた場所に活動領域を生成した生物はいなかった。

図 3.8 は NPC 配置の自動生成手法としての有効性の検証のシミュレーション結果をまとめたものである。

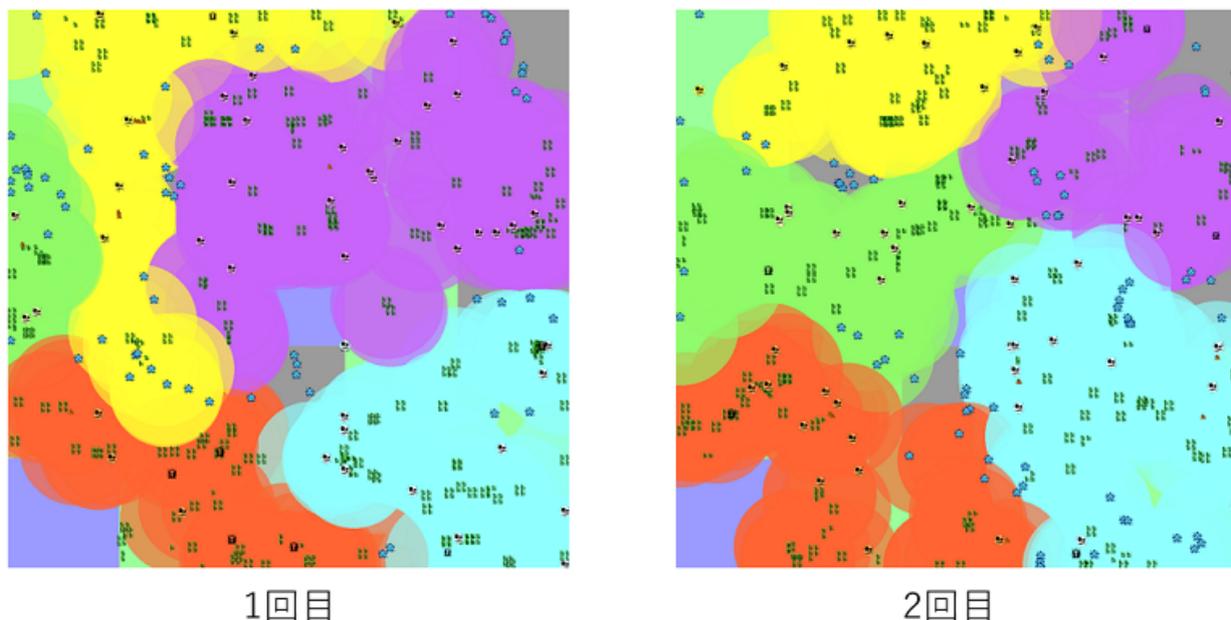


図 3.8 自動生成手法としての有効性の検証の結果

### 3.3 考察

本研究では食物連鎖による生態系シミュレーションに縄張りの機能を加えたシミュレーションを行い生物の配置を生成した。その結果、本手法により食物連鎖や縄張りなど特定の要素が必要なゲームにおいて、生物同士の関係性からなる配置を生成することができることが分かった。また、NPC の配置を自動生成する手法として有効性があることが分かった。

既存のゲームにおける NPC 配置再現の検証では、徳田のモンスターハンターワールドのモンスターの配置を参考に本手法で生物の配置を生成を行った。同条件で 2 回の検証を行った結果、2 回とも基となったモンスターの配置図と同様の形に生物の領域を生成した。このことから、食物

連鎖や縄張りによりゲームにおける架空の生物の関係から生物の配置を生成できることが分かった。また、徳田のモンスターハンターワールドでは肉食動物のモンスターは互いに敵対関係にあることから、本手法は敵対する生物同士の配置を生成することもできると考える。

NPC 配置の自動生成手法としての有効性の検証では、5種類の肉食動物を種類ごとに間隔を空けて配置し初期配置から離れずに生物の配置を生成できるかを検証した。結果として、初期配置から離れた位置に生物の領域を生成する生物はいなかったが、領域の形状は実行ごとに大きく変化していた。本手法において初期位置から離れた位置に領域を生成しなかったことから NPC の配置における自動生成の手法として有効であることが分かった。

本手法では生態系を再現するために最低限要素による食物連鎖と疑似的に縄張りの機能を追加することによって NPC 分布を生成した。本手法における生物の行動や食物連鎖などは現実の生態系の一部を参考にしたものであり、他に様々な生物的な要素を加えることで本手法とは違う結果を得ることができると考える。村上ら [19] の羽虫の群衆についての行動制御の研究や、西村ら [20] の捕食関係によって生物の行動を決定する手法など生物の行動をデジタルで表現する研究は多い。群衆などの様々な生物の行動制御や生態を表現する研究と組み合わせることで、本研究ではできなかったような生物の関係性や行動による NPC を生成することができると考える。

## 第 4 章

# まとめ

本研究は生態系シミュレーションを用いることで、ゲームにおける架空の生物による生態系を再現した NPC 分布を自動生成することを目的としその手法を提案した。生態系を再現した NPC 分布を生成するため、本研究では現実の生態系における縄張りを NPC 分布と見なした。縄張りを形成する生物において、オオカミは他の群れのオオカミに襲われることを恐れ他の縄張りに侵入しない習性があることから、本研究における NPC 分布に利用した。

本手法では植物、草食動物、肉食動物の 3 種類の生物による食物連鎖を基とした生態系シミュレーションを作成し、肉食動物に縄張りを形成する機能を与えることで NPC 分布を生成した。また、生物には寿命や空腹など様々なパラメータを与え、捕食や回避などを与えることで現実の生物らしい振る舞いを再現した。

本手法を 2 種類のパターンでシミュレーションを行いその考察を行った。その結果、本手法による NPC 分布では設定により創発的な分布や、詳細に設定することで特定の生態系を再現した分布を生成することも可能である分かった。そのため本手法は NPC 分布の手法としては、様々な分布を生成できることから有用であることを確認した。

今後の展望としては、本手法における生物に群衆行動や縄張りの情報などを取り入れることで、本研究の目的である架空の生物による生態系の NPC 分布をより現実に近い形で再現できるので

はないかと推測した。

# 謝辞

本論文を執筆するために、夜遅くまで対応していただいた渡辺先生、阿部先生に深く感謝いたします。そして、研究に協力していただいた研究室の皆様、中間発表で意見を下さった方々にも深く感謝いたします。

## 参考文献

- [1] Yinxuan Shi and Roger Crawfis. Optimal cover placement against static enemy positions. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.383.1384>. 参照:2020.02.28.
- [2] 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, 第 22 巻, pp. 745–756, 2010.
- [3] 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 (<特集>ゲーム AI). 人工知能学会誌, Vol. 23, No. 1, pp. 44–51, 2008.
- [4] 森川幸人. テレビゲームへの人工知能技術の利用 (AI 技術の産業応用 [第 7 回]). 人工知能学会誌, Vol. 14, No. 2, pp. 214–218, 1999.
- [5] 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の現在 (<特集>エンターテインメントにおける AI). 人工知能, Vol. 30, No. 1, pp. 45–64, 2015.
- [6] 徳田優也. フィールドとモンスターの制作工程から読み解く「モンスターハンター:ワールド」のゲームデザイン. [https://cedil.cesa.or.jp/cedil\\_sessions/view/1942](https://cedil.cesa.or.jp/cedil_sessions/view/1942). 参照:2020.02.28.
- [7] 星野力. 人工生命の現状と将来への期待. 計測と制御, Vol. 32, No. 8, pp. 677–683, 1993.
- [8] 三宅陽一郎, 池上高志. ゲーム AI と人工生命が交わるとき、革命的な物語が生まれる。

<http://boundbaw.com/inter-scope/articles/9>. 参照:2020.02.28.

- [9] 田中昌宏. 内湾の生態系シミュレーション. 日本流体力学会誌「ながれ」, Vol. 20, No. 5, pp. 354–364, 2001.
- [10] 石川芳男, 東海林直哉, 杉浦桂. 人工生命理論による微生物生態系でのコロニー発生のシミュレーション. *CELLS JOURNAL*, Vol. 9, No. 1, pp. 5–12, 1996.
- [11] 坂本恋, 和田勇気, 大島登志一. MitsuDomoe: バーチャル生態系の捕食連鎖シミュレーション体験システム (第2報). エンタテインメントコンピューティングシンポジウム 2017 論文集, 第 2017 巻, pp. 435–438, 2017.
- [12] 森脇康介, 横井大祐, 犬塚信博, 伊藤英則. 遺伝的プログラミング技法を用いた多出力二分決定グラフの進化: 食物連鎖におけるマルチエージェントの進化シミュレーション. 人工知能学会誌, Vol. 14, No. 3, pp. 477–484, 1999.
- [13] 常田邦彦. シカ個体群の制限要因としてのオオカミ. 哺乳類化学, Vol. 35, No. 1, pp. 45–48, 1995.
- [14] William J. Ripple and Robert L. Beschta. Wolf reintroduction, predation risk, and cottonwood recovery in Yellowstone National Park. *Forest Ecology and Management*, Vol. 184, pp. 299–313, 2003.
- [15] K.Zub, J. Theuerkauf, W. Jedrzejewski, B. Jedrzejska, K. Schmidt, and R. Kowalczyk. Wolf Pack Territory Marking in the Bialowieza Primeval Forest (Poland). *Behaviour*, Vol. 140, No. 5, pp. 635–648, 2003.
- [16] Perzanowski Kajetan. オオカミ-その生態と他種との関係. 哺乳類化学, Vol. 30, No. 1, pp. 29–33, 1995.
- [17] 北澤大輔, 熊谷道夫. 流動場—生態系結合モデルによる琵琶湖生態系シミュレーション. 哺乳類化学, Vol. 59, No. 1, pp. 21–26, 2007.

- [18] J.P. Kimmins, Daniel Maily, and Brad Seely. Modelling forest ecosystem net primary production: the hybrid simulation approach used in forecast. *Ecological Modelling*, Vol. 122, No. 3, pp. 195–224, 1999.
- [19] 葉村上, 宜之土橋, 強山本. 羽虫の群れの動きのコントロール. Technical Report 7, 北海道大学大学院情報科学研究科, 北海道大学大学院情報科学研究科, 北海道大学大学院情報科学研究科, 2018.
- [20] 西村信一郎, 池上高志. 捕食関係によって誘引される空間パターン戦略の解析 (ポスターセッション-離散多体系、生物、粉体、交通流など-, 複合系 II 要素と全体-現象論の視座-, 研究会報告). <http://hdl.handle.net/2433/95665>, 1996. 参照:2020.02.28.