モノづくり・技術力について

東北大学
農学研究科
資源生物生産科学

奥野 雄貴

東北大学

https://github.com/YukiOkuno-2023/YukiOkuno-2023

発表の流れ

Ⅰ．研究制作物（生態系シミュレーション）

Ⅱ．研究に関連する領域の 自主制作物（人工生命）

Ⅲ．その他の自主制作物

研究制作物

研究の目的

生態系サービスを最大化したい

＝ 生態系と長く、上手く付き合っていく方法を知りたい。

私たちは絶滅と大量発生に注目

何故絶滅に注目したか？

水産資源の様に、絶滅を回避しながら利用し続けたい生物資源は沢山あります。

逆にどうすれば絶滅が起こせるのかを知ることが出来れば、資源と上手く付き合う方法が分かるかもしれません。

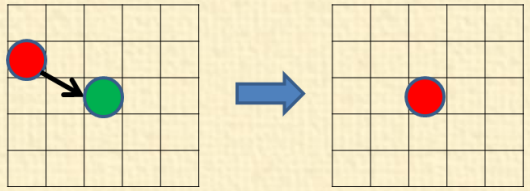
⇒ シミュレーション的なアプローチが有効

Stochastic Individual Based Model(S-IBM)

数学的に定義された従来の密度ベースのモデルとは異なり、個に注目した数理モデルをIBMと言います。
本モデルは確率過程を用いており、そのアルゴリズムは以下の通りです。

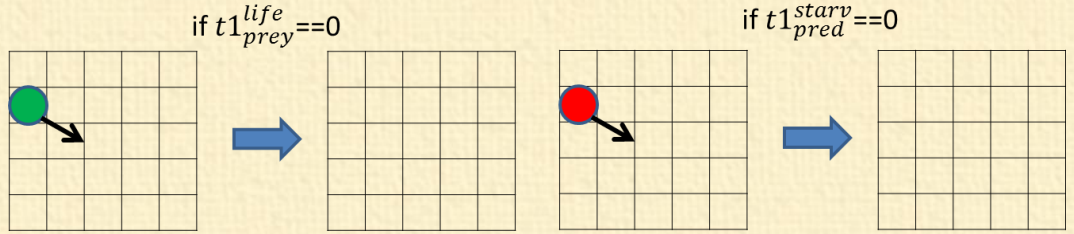
IBMでは、生物を定義するのではなく格子状の空間を定義し、空間にパラメーターを持たせ、それらを時間変化させることで疑似的に生物を再現します。

①移動の際、PredatorがPreyに追いつくと捕食します。

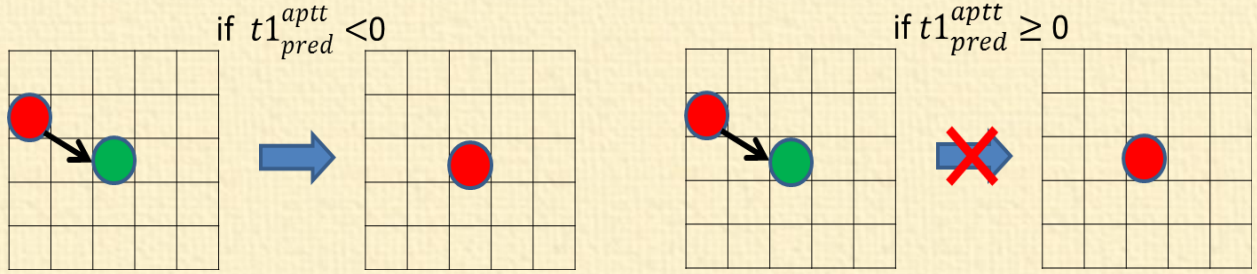


緑:prey , 赤:predator

②移動に伴って寿命と飢餓時間が減少し、0になると消滅します。



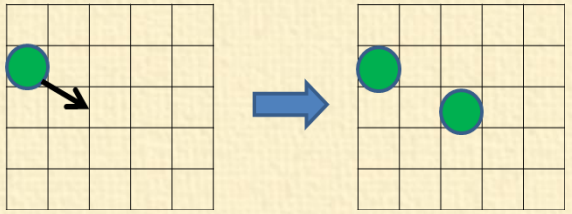
③捕食が発生すると飢餓時間は初期値に戻ります。
捕食後しばらくは捕食行動を取りません。



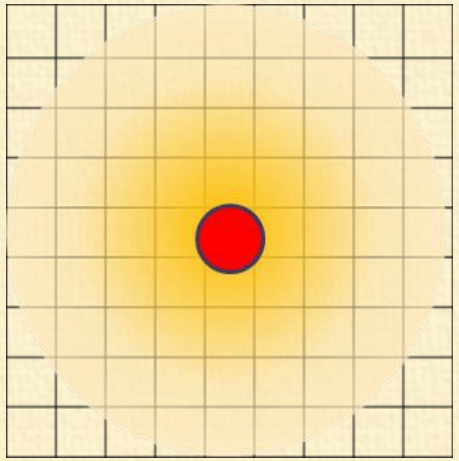
Stochastic Individual Based Model(S-IBM)

④prey, predatorは固有の増殖間隔を持ち、一定時間で増殖します。

if $t1_{prey}^{reprod}$==0



移動先の決定について



本モデルの最大の特徴は移動処理にあります。
Box-Muller法を用いることで、近い場所ほど移動しやすく、遠い場所ほど移動しにくいという生物的な振る舞いを再現しています。

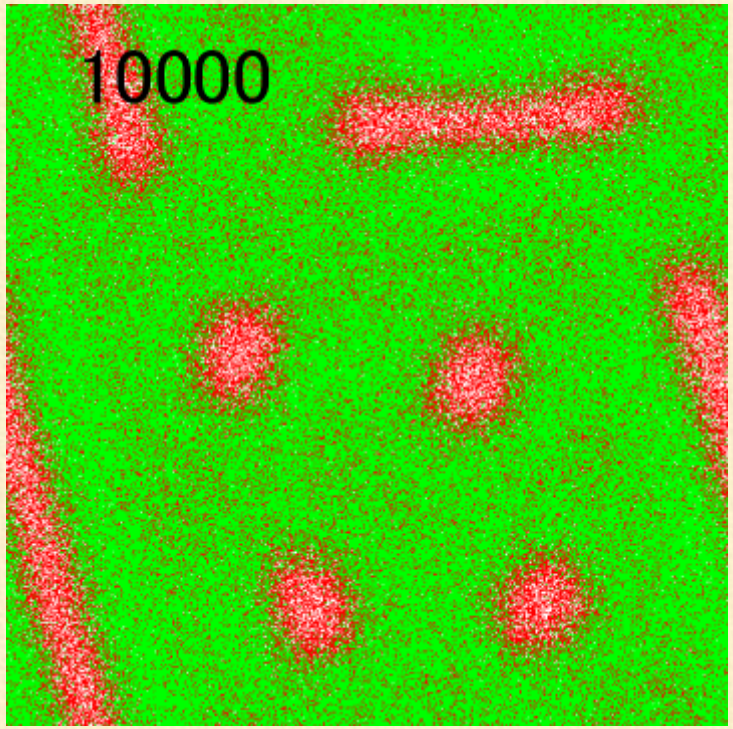
$$x_n = x_{n-1} + \sigma\sqrt{-2\log(a)} \times \cos(2\pi b)$$

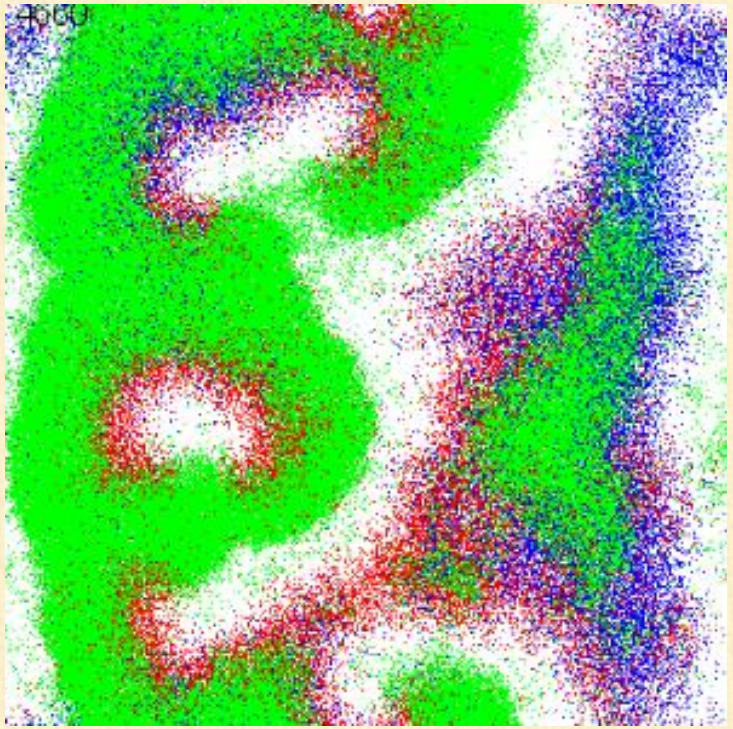
$$y_n = y_{n-1} + \sigma\sqrt{-2\log(a)} \times \sin(2\pi b)$$

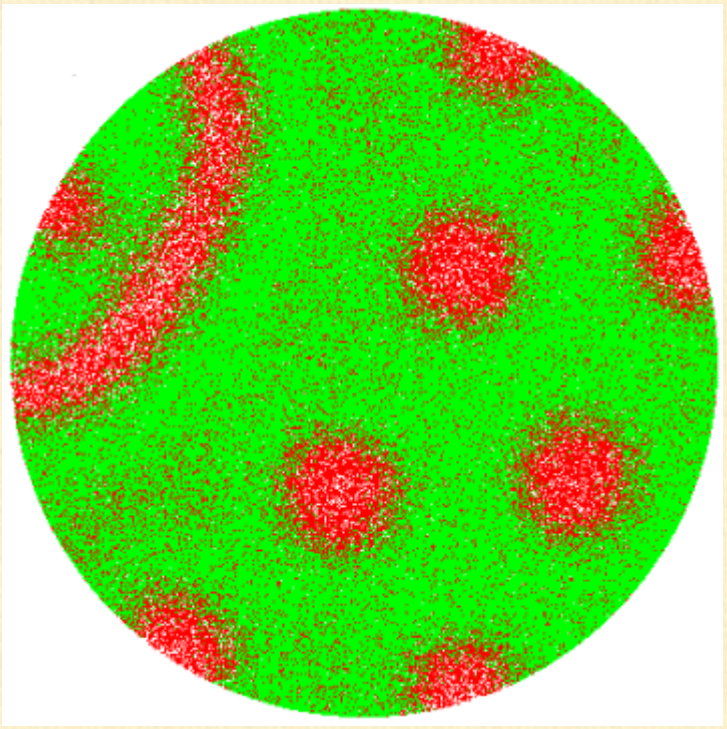
X_n: nステップにおけるx座標
X_{n-1}: (n − 1)step時の個体のx座標
a, b:0から1までの乱数
σ：**移動係数**(値が大きい程長距離を移動しやすい)

従来のIBMの移動処理はノイマン近傍（4方向の隣接するセル）のみを移動の対象としていました。

S-IBMの出力例







赤が捕食者、緑が被捕食者、青が寄生されている捕食者を示しています。

ソフトウェア開発

本モデルは確率過程を用いており、ある結果が本当に妥当と言えるかは繰り返し計算を行うことでしか確かめることが出来ません。

膨大な数の調査が必要な一方で、私が引き継いだプログラムはMATLABから直接計算結果を得るというもので、並行して計算を行おうとするとエディターを複数立ち上げる必要がありました。

そこで、私はモデルの概念を落とし込んだ動作の軽いソフトウェアを作り、再入可能性を持たせることで調査の効率化を図りました。

ソフトウェア設計

①条件の入力	テキスト形式のファイルを用いて外部からパラメーターの入力を行います。
②計算の実行	前のスライドで紹介したS-IBMのアルゴリズムに則り、計算を実行します。
③進捗の出力	研究計画に影響が出ないよう、計算の進捗をコンソールに出力します。 進捗は[現在のステップ数 / 全体のステップ数]で計算され、進捗が2%進むごとに出力されます。
④結果の出力	計算結果を独自のバイナリデータとしてファイルに出力します。 このファイルは0-nまでの数値が並んだもので、0は空間、1は被捕食者、2以降は各種の捕食者を表します。
⑤ファイル化	①～④の要件を満たす実行ファイルを生成し、アルゴリズムの変更をソフトウェアに容易に反映できるように設計します。

・開発環境
Visual Studio

・開発言語
C++

・コードの規模
約5,000行
（最新版）

・開発期間
6ヶ月

・開発人数
2人

中間ファイルの生成を行う必要性

計算結果を得るには計算工程と描画工程を経る必要があり、各プロセスは同時に処理できる数が異なっています。
これらを1つのソフトウェアにまとめた場合、並行して実行できる数はパフォーマンスの低い方に合わせることになり、効率が悪くなります。

加えて、実行時間の長期化は中段によるリスクが高くなります。
これらの問題に対処する方法として、計算工程で得た結果を一度独自のバイナリデータとして保存することにしました。

ex）計算工程は200通りを並行して実行できるが、描画は5通りずつ実行しなければならない場合

⇒ 1つのソフトウェアにしてしまうと並行出来る数は5通りに制限されてしまいます。

・エディター上での実行

・・・

リソースに余裕があり、処理が遅い

50分程度かけて1つの結果が得られる
(4日で110パターン)

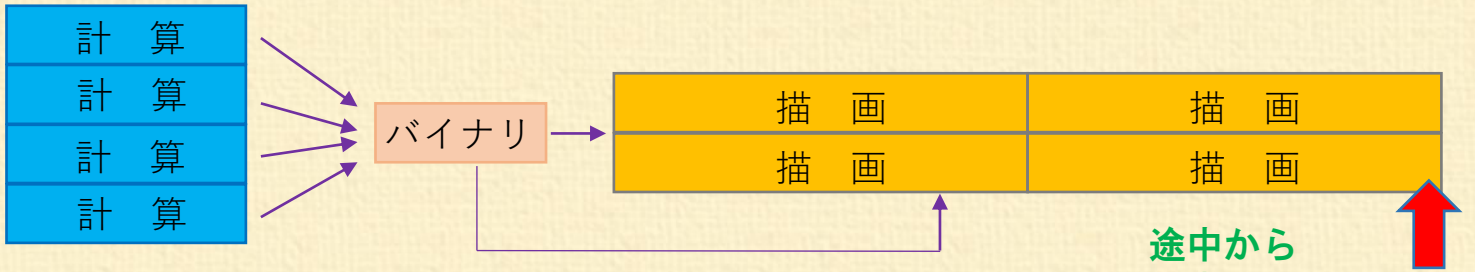
・計算と描画を1セットにしたソフトウェア

・・・

エディターを起動しない分速くはなったが、重い描画工程に合わせているため処理能力制限されている
中断のリスクが高い

約1.8倍

・計算と描画を切り離したソフトウェア



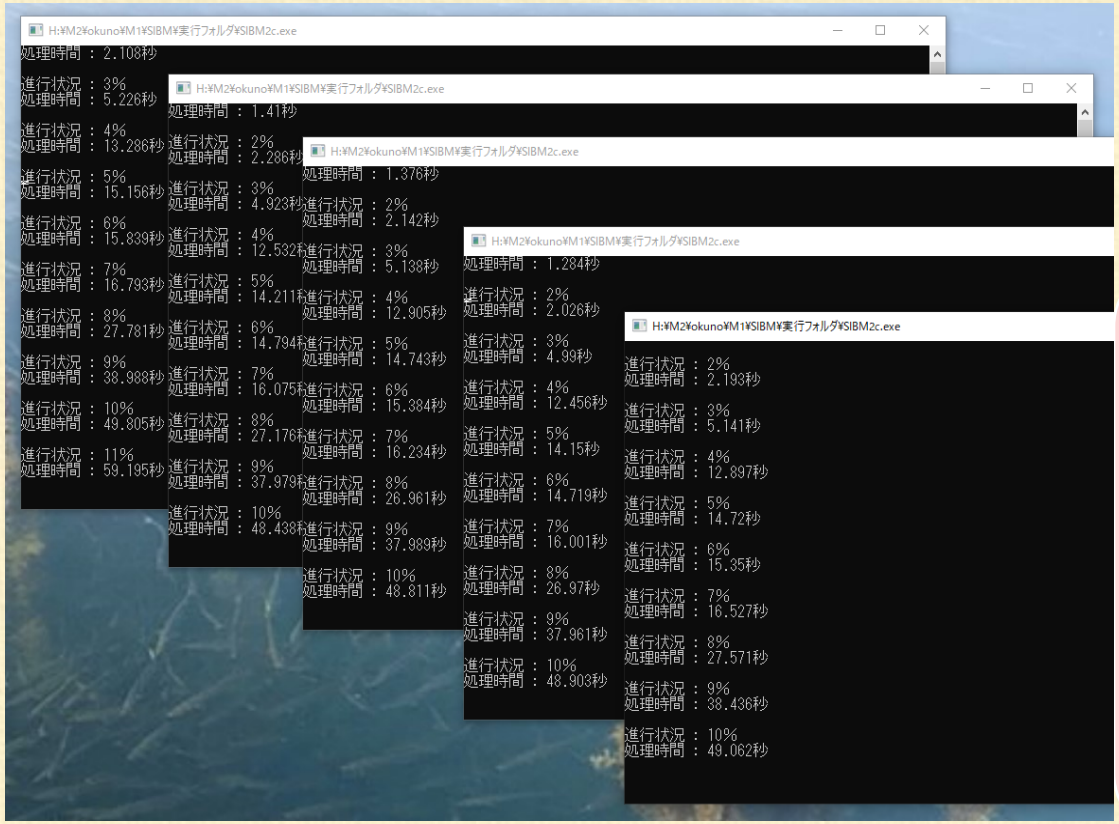
・・・

リソースを無駄なく使用しているため高速
(4日で約200パターン程度)
実行時間の長期化に伴うリスクを低減

ソフトウェア開発

下図はIPrey-IPredator系での計算の実行画面と結果の一部です。

例えば領域を500*500で定義している場合、計算結果のバイナリデータを500文字ずつ読み取り、番号に対応した色を割り当てつつ同じ行にタイリングします。
これを500行分繰り返すことでIstep分の計算結果を視覚的に理解できる形にします。
上記を数千ステップ繰り返し、動画ファイルとして出力することで系の時間変化を動画データから観測することが出来るようになります。



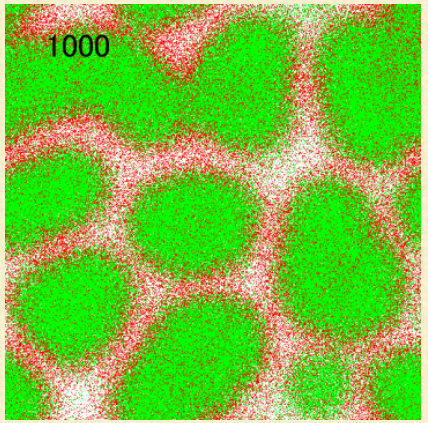
①

左図は計算中の画面です。
エディターを介さないことで複数の条件を同時に調べることが出来ています。

また計算の進捗をリアルタイムで更新しているため、終了予定時刻が予想しやすく、計画通りの進捗が行えるようにしています。

②

数字に対応した色で描画



③

動画ファイル化

開発にあたっての工夫

上図は前スライドで紹介した研究を引き継いだ時点でのプログラムのイメージ図です。
しかし実際は、

1step分を計算 ⇒ 計算した分を描画 ⇒ 画像ファイルとして出力

⇒ 繰り返し ⇒ 全工程終了後に画像ファイルを繋げて動画化

という流れで進行しています。
つまり正確な図で表すと、

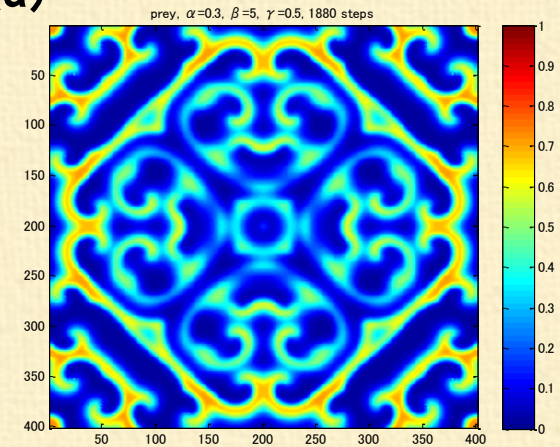
元々計算と描画は1つの塊として処理していました。

計算と描画を完全に分離する発想力と挑戦心、それを実装した技術力を評価して頂きたいと考えます。

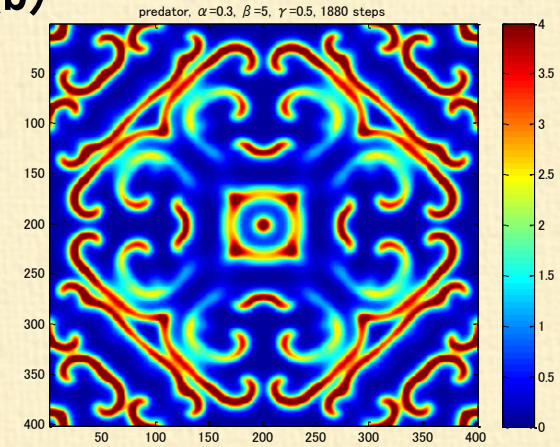
展望

本モデルは一定の規則に基づいてパラメーターを動かしています。
数学的に表現することが難しい相互作用でも本モデルであれば容易に実装することが出来ます。
他の数理モデルと比較するで互いに議論を高度化することが当面の目標となります。

(a)



(b)



(a), (b)
左図は次式で表される捕食-被食関係の数理モデル(RM-Diffusion model)による出力結果です。

$$\frac{\partial \mathrm{u}}{\partial \mathrm{t}} = au\left(1-\frac{u}{k}\right) - \frac{buv}{c+u} + D_U\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)U$$

$$\frac{\partial \mathrm{v}}{\partial \mathrm{t}} = \frac{puv}{c+u} - qv + D_V\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)V$$

左図の(a)は被捕食者の密度分布で、(b)はそのときの捕食者の密度分布です。
青い程密度が低く、赤いと密度が高い状態を表しています。

全く違うアルゴリズムでの計算結果ですが、S-IBMと同様の螺旋型のパターンが形成されていることが分かります。
異なる数理モデルと比較して類似した結果が得られた場合、互いに妥当性を評価し合うことが出来ます。
モデルに新たな相互作用を実装した際に比較し合うことで議論の高度化につなげることが出来ます。

関連分野の制作物

何故人工生命の領域に注目しているのか

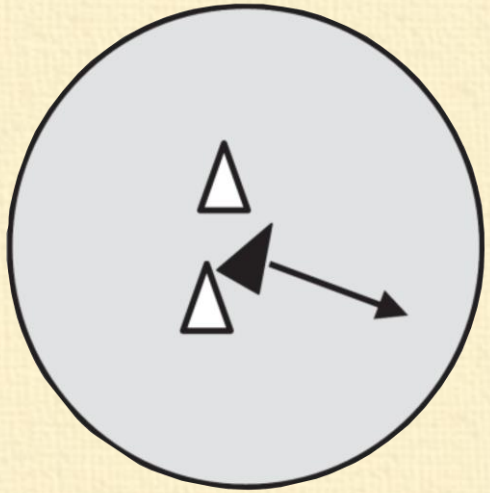
- 生き物についてより深く理解したかったから
- 幼い頃から生物を1から作ってみたいという欲求があったから
- 新たな理論を思い付いたから
  将来的にアイデアを形にするため

Boidモデル

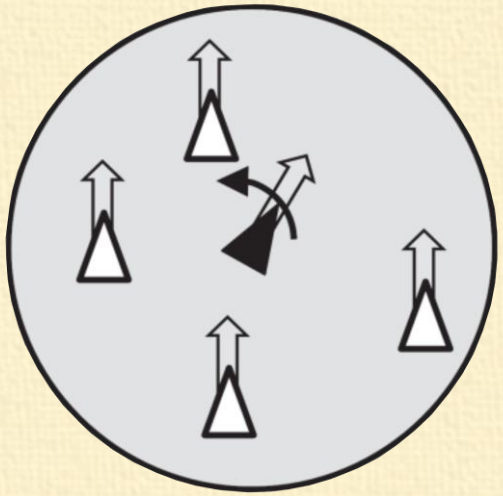
- boidとは「鳥」を表すbirdと「～モドキ」を表す「-oid」を組み合わせた造語で、Boidモデルとはコンピュータ上で鳥の様な仮想生物が群れを自然に形成するモデルです。

  boidモデルは次の3つの単純なルールによって定義されます。

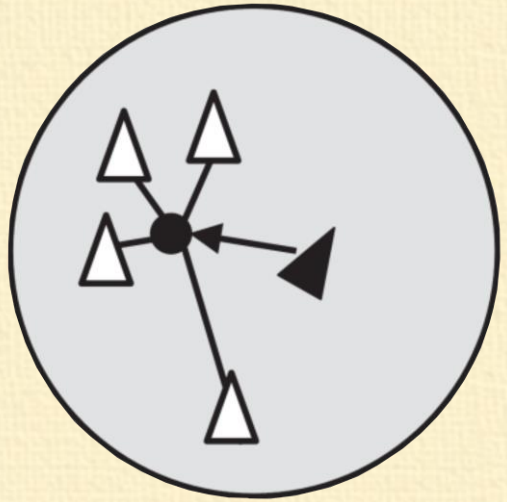
① 周囲の個体との衝突を回避する

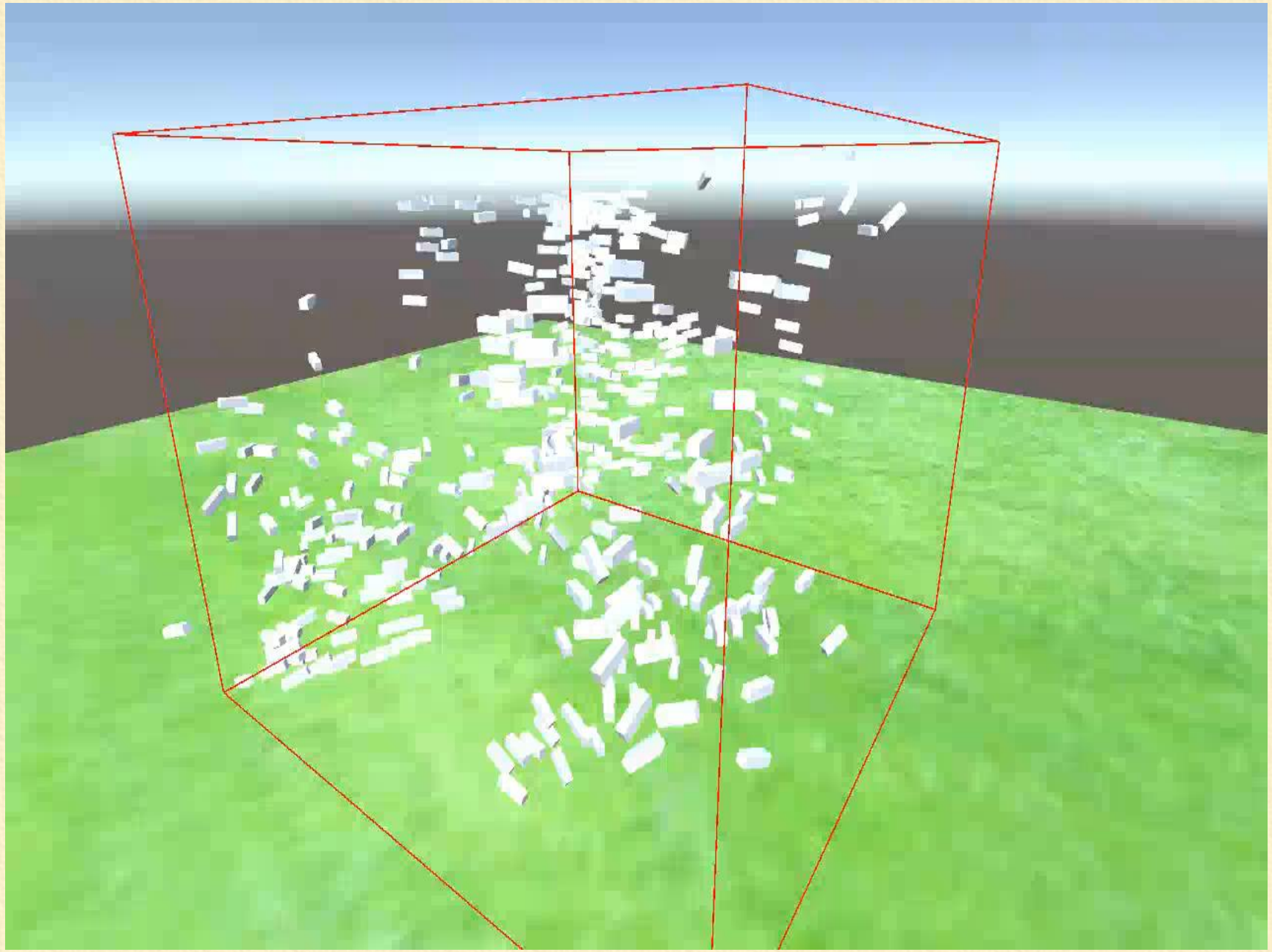


② 周囲の個体と速度と方向を合わせて移動する



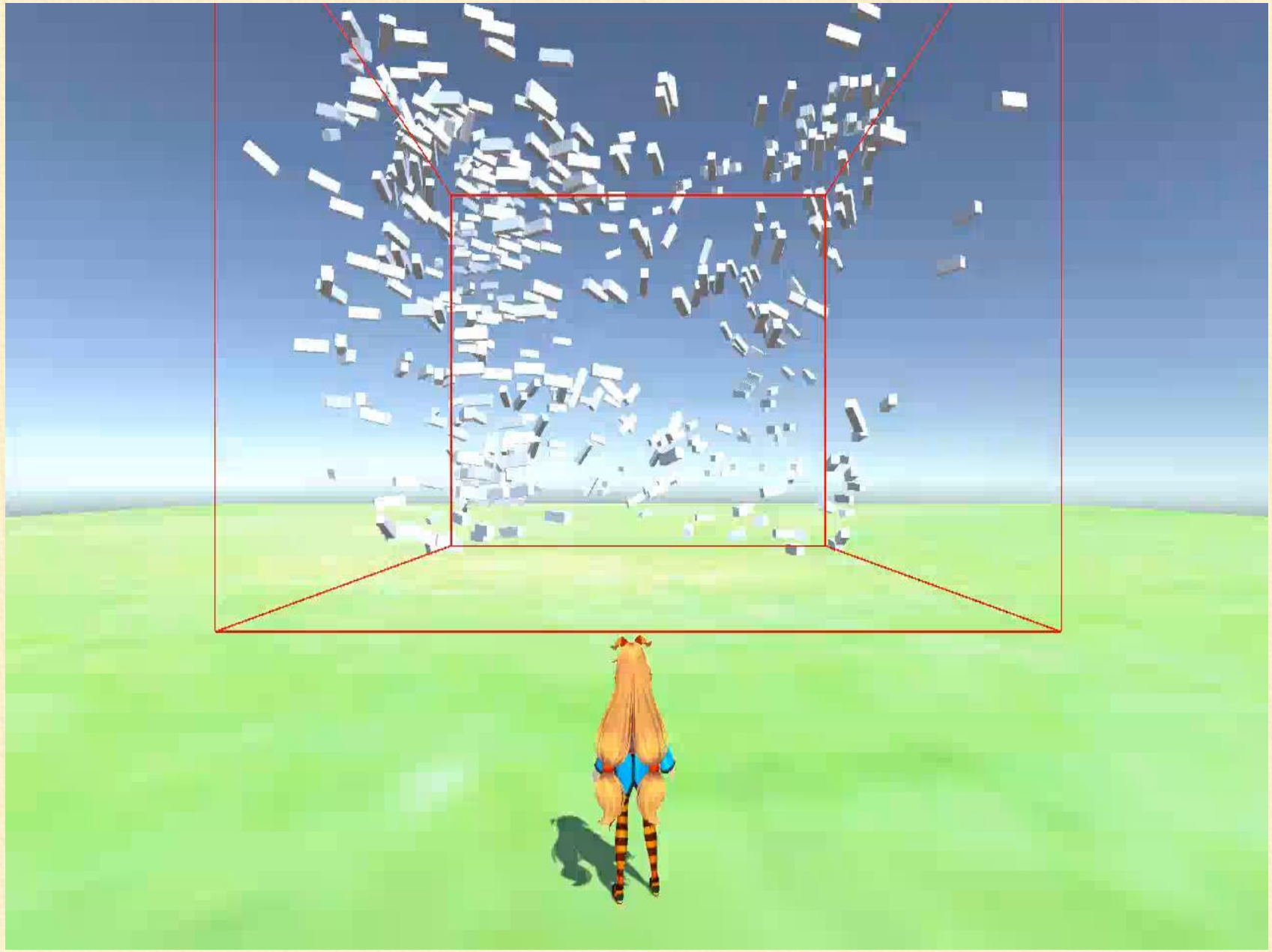
③ 群れの中心に移動する





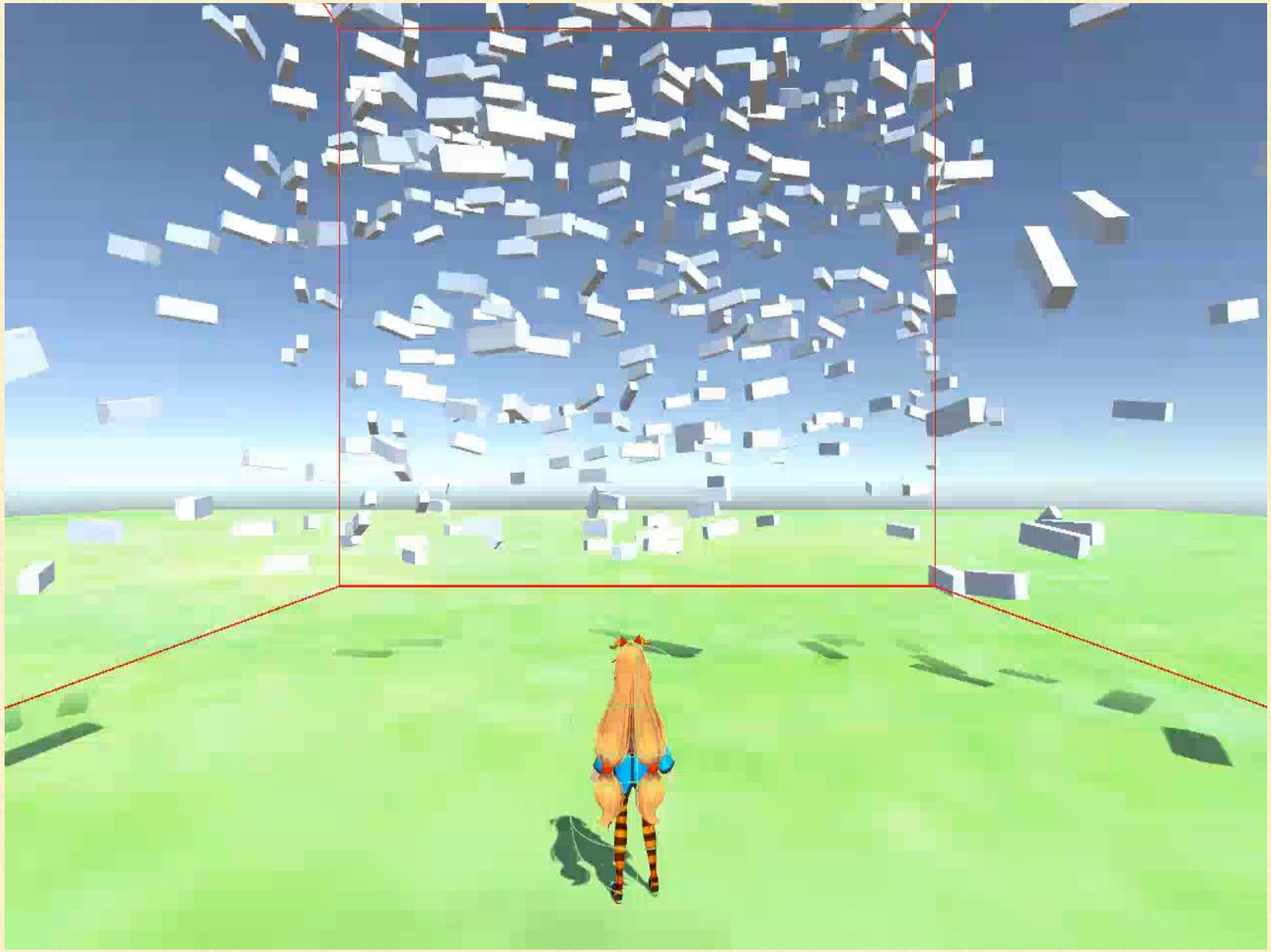
立方体の中でシミュレーションを行うことで魚群の様な挙動を再現することができます。

前のスライドで紹介した加速度の前に係数を設けることで結束の強い群れやその逆を作ることが可能です。



ある物体との間にベクトル場を形成させることで個体同士以外にも回避行動を取ることが出来るようになります。

これをキャラクターに適用することで群れをわって入る様な光景を再現できます。

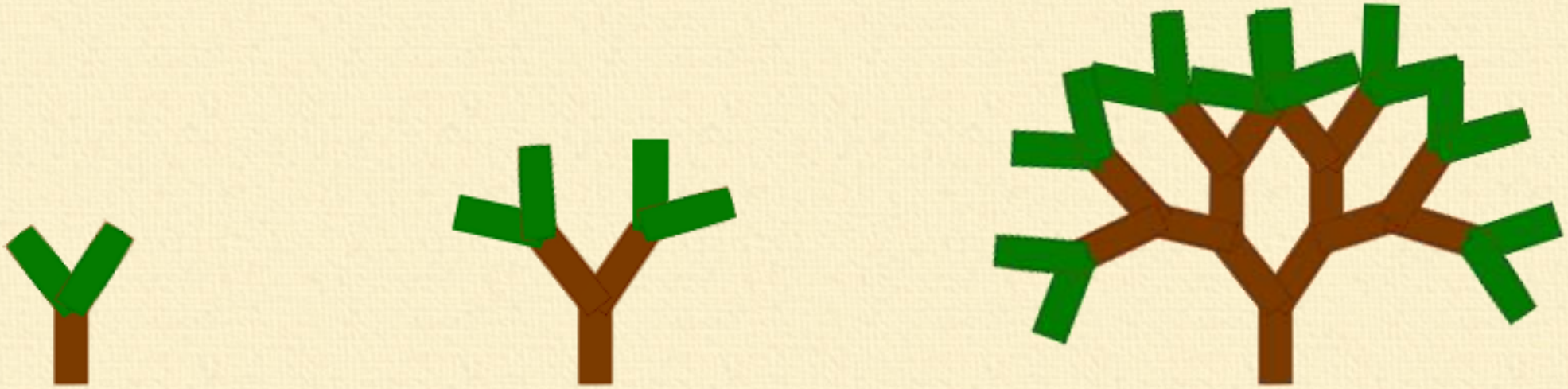


また、特定のアニメーションの最中だけオブジェクトの間に発生するベクトル場を大きくすることで驚いて逃げるような挙動を自然に盛り込むことも可能です。

L-System

提唱者であるA. Lindenmayerに因んで名付けられた手法で、バクテリアの細胞の並びは自己相同性を持ち、その成長過程は指令表に変換できるという理論です。

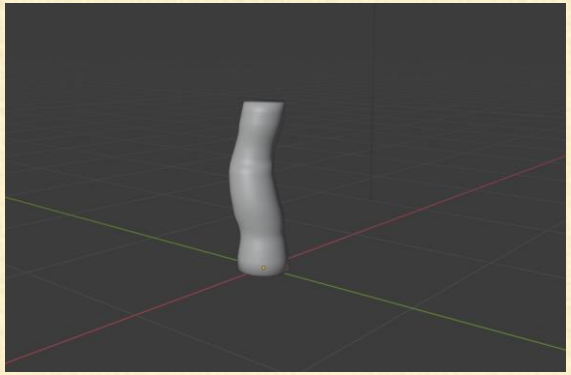
自己相同性を持つ物体の自動生成が可能と考えられています。



緑の部分をフラクタル図形で置き換える操作を繰り返すことで樹木の様なものが形成されます。

L-Systemの出力結果

左図の様な基本的な図形を用意し、フラクタルの最小単位となる図形を生成します。



ランダムで生成した基本図形を再帰的に置換することで上記の様な樹木を自動で作ることが出来ます

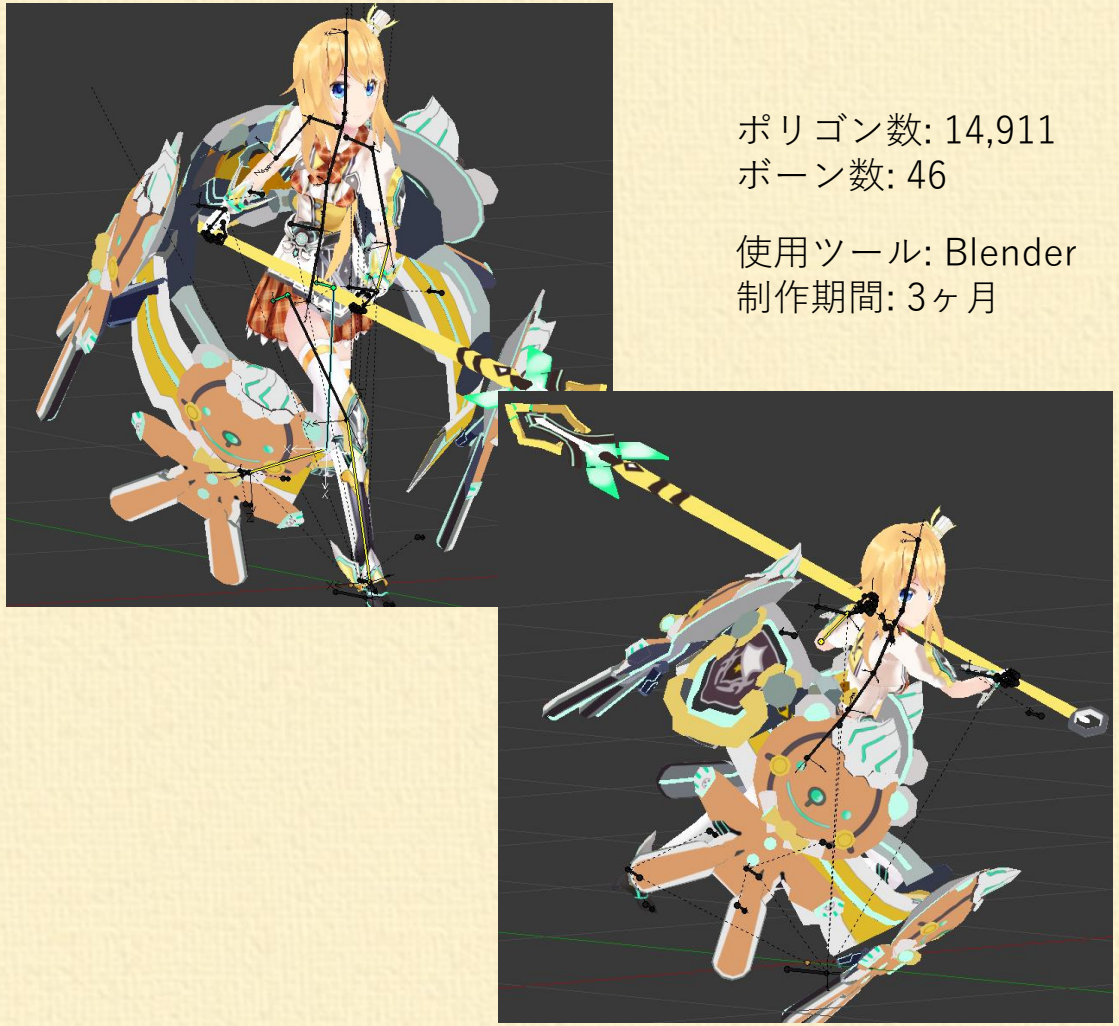
その他の制作物

ゲーム制作

サービスが終了した作品をもう一度遊びたいと思ったため、
鋭意制作中です。

出典：株式会社gumi

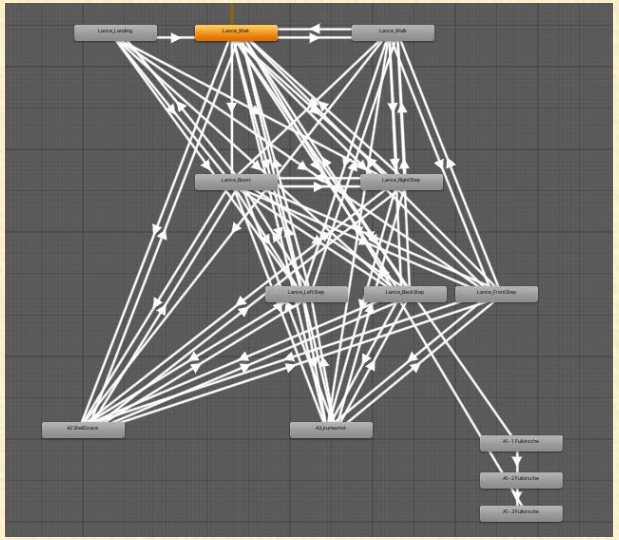
ドールズオーダー

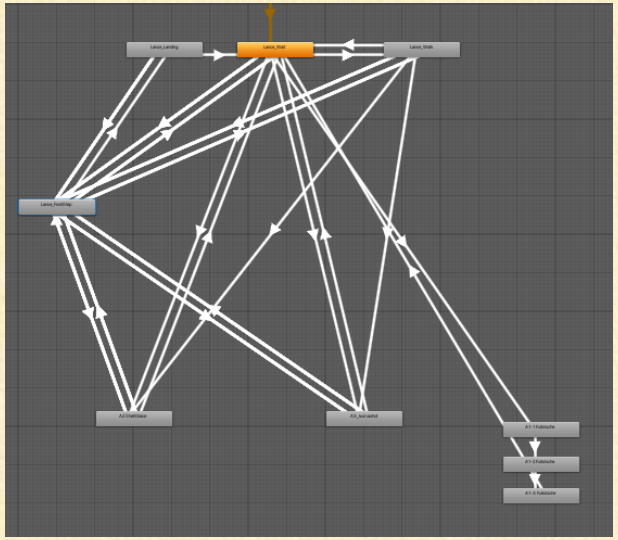


ポリゴン数: 14,911
ボーン数: 46

使用ツール: Blender
制作期間: 3ヶ月

ゲーム制作

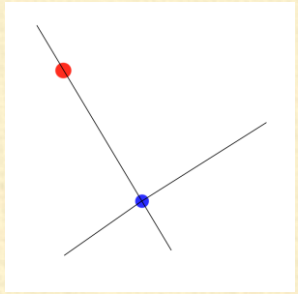




作成したアニメーションの一部をアニメーターに登録した状態です。相互にキャンセルし合える仕様は非常に煩雑な構造を伴います。
キャンセル出来るという共通の特性を持つステートを重ねることで意図しない挙動を防ぐことに役立つと考えています。。

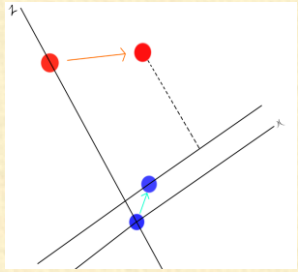
誘導に関する手法の開発

物体の誘導に関して、既存の方法に不満を感じたため、アルゴリズムを自分で考えて実装しています。



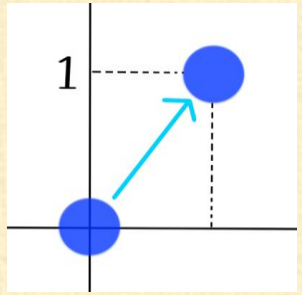
①発射時に弾頭を原点とした左図のようなローカル軸を定義します。

各軸は2点を通るベクトルと、直交するベクトルです。
弾頭が直進する方向をZ軸、Z軸に直交する誘導方向の軸をX軸とします。
青：弾頭　赤：ターゲット



②ターゲットの現在位置を用い、ターゲットの横方向の移動をローカルX軸に投射します。

ローカルX軸を、Z方向に進んだ分だけ位置ずらすことでを投射を1成分で表すことが出来、演算にベクトルを用いないことで高速化しています。
ターゲットの投射と弾頭の位置の差を求めることで、着弾に必要な誘導の強さが分かります。



③Z方向への移動量を1と定義し、②で定義した値と共にベクトルとしてTranslateで弾を移動させます。

ベクトルを正規化することで、弾速は常に一定になります。
正規化しているため、直進成分と誘導成分は和が1となる相対量です。
誘導成分を定数倍し、誘導が支配的になりやすくした場合は強誘導となります。
逆に、誘導成分に掛ける定数を小さくすると直進成分が支配的になりやすく、低誘導の弾丸を作ることが出来ます。

誘導に関する手法の開発