

Boidを用いたクラウド内のファイル検索システム

木下研究室

鈴木 健人 (200902854)

1 はじめに

近年、クラウドコンピューティングが普及しクラウド内のデータ量が大量になってきており、これによって大量にあるファイルの中から重要なファイルを見つけづらくなっている。これを新たなファイルシステムを提案することで、目的のファイルを視覚的に見やすく検索しやすいファイルシステムを行えるようにしたい。視覚的に見やすくするためにファイルに群れの行動を取らせ重要なファイルは群れの中心に来るようにする。群れをつくるものとして Boid を使い群れの行動を取らせる。

2 提案

重要性や関連性があるファイルほど群れの中心に集まり、低いファイルほど群れの外側に来るようにする。重要としているファイルはユーザーによって違うので何を重要としているのかを選択できるようにし、目的のファイルを群れの中心へ集める。

2.1 Boid

Boid とは 3つのルールを規定するだけで鳥の群れをシミュレーションできるといふもの。

分離：近くにいる仲間と衝突しないようにする。

整列：近くの仲間と衝突しないようにする。

結合：近くにいる仲間を囲まれた状態になろうとする。3つのルールを図1に示す。

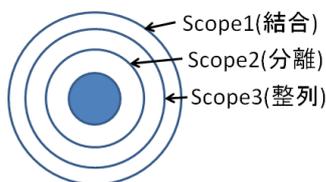


図 1: Boid を構成する 3つのルール

2.2 重要度の定義

ファイルに重要度を定義する。

- 使用頻度
使用頻度が多く、かつ閲覧日が新しいファイルほど重要度が高いとする。
- ノルマ
ノルマがある仕事に対して、ノルマが達成されていない場合、重要度が高いとする。

2.3 重要度の選択とフィルタ

- ファイル一つ一つに重要度を幾つか持たせそれを選択できるようにすることで何を重要なファイルとするかを定める。
- File クラスのファイルの最終更新日を取得するメソッドを使用して更新が新しいものしか表示しないようにするフィルタを作成し、ファイルの表示を限定する。

3 実装

Boid をプログラムして群れをなし、餌を撒いたら餌へ向かうようにした。Boid のアプリケーションを図2に示す。

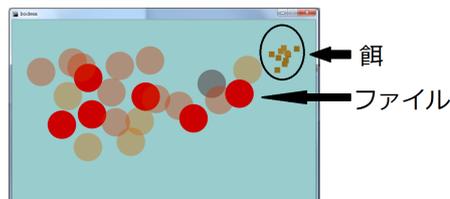


図 2: Boid のアプリケーション

Boid が餌を追うアルゴリズムを図3に示した。これはまず Boid と餌の間の距離を出し、scope1 の 2 倍の範囲内に餌があれば、Boid と餌の距離が 0 になるまで処理を繰り返すことによって餌を追っている。これだけでは群れるだけなので餌を追うアルゴリズムと重要度を組み込んで、餌を集合地点とし重要度が高いものはそのまま餌に向かっていき、重要度が低いものは餌を検知する範囲を 0 にすることで餌を検知させないようにするが、Boid によって群れからは離れないので群れが分裂することはない。よって、使用頻度を重要度とした場合、使用頻度が高いファイルはそのまま餌を追うアルゴリズムに従い集合地点である餌の元へ行き、使用頻度の低いファイルの検知範囲を 0 にして餌を認識できなくして重要としているファイルが餌に集まってくると考える。

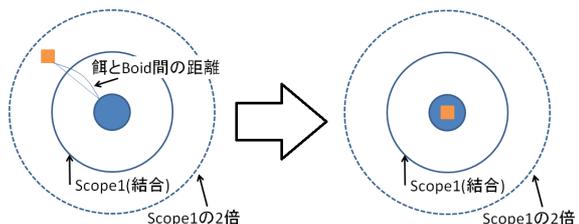


図 3: Boid の餌を追うアルゴリズム