

MapEditor をもちいた汎用 Pedestrian MAS

東京大学理科一類 1 年 富田寛

2011 年 2 月 5 日

1 開発動機

都市計画や建築物の設計、レイアウト作成の場面では、さまざまな状況で安全かつ快適に空間を利用できるように歩行シミュレーションがおこなわれている。2000 年からは建築基準法でもコンピュータシミュレーションによって安全性を立証する方法が認められた [1, pp.10–11]。しかしビルなどの建築物が十分に安全につくられているかをチェックするシミュレータは非常に高価で操作が難しい。一般人がイベントのレイアウトをするときなどに試行錯誤しながらつかえるシミュレータが必要とされていた。

artisoc はマルチエージェントシミュレーションの敷居を下げ、多様なモデルを手軽に再現できるようにした画期的なシミュレーション環境である。しかし既存の artisoc の歩行シミュレータではオブジェクトの位置などがハードコードされており、エージェントルールと空間情報が十分に分離されているものはすくなかった。このような実装方法は実行速度や厳密さの点ですぐれているが、あらたな空間の考察をはじめるときに毎回ルールを書く労力をはらわねばならない。そこで私は artisoc をベースに、より簡単につかえる歩行者シミュレーションアプリケーションを開発した。

2 “MapEditor for Pedestrian MAS” とは

MapEditor は artisoc 上で実行可能な model ファイルを生成するプログラムであり、HTML5 と JavaScript で書かれている。ブラウザだけで利用できるため、Java で実装されている artisoc とおなじく、実行環境をえらばない。ユーザはペイントソフトライクなインタフェースを通じてマップをデザインし、シミュレートできる。

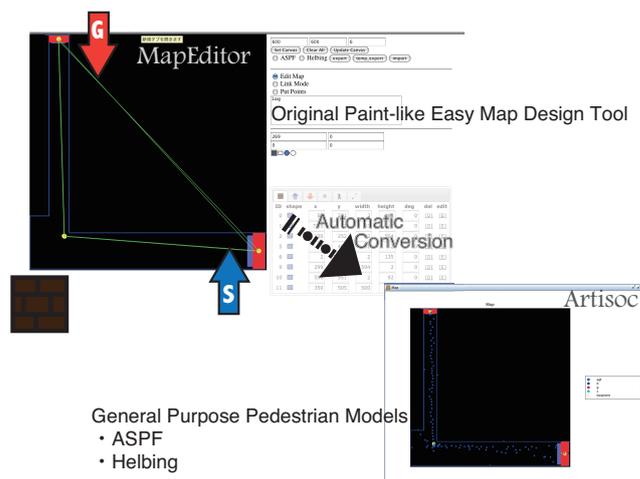


図 1 MapEditor システムの概念図

また MapEditor ではモデル作成、空間デザイン、シミュレーションの各ステップを完全に切り離せるので、グループで研究するにも最適である。MapEditor のすべてのコードを公開しているため、それぞれの目的にあわせてルールをかきなおしたり、あたらしいオブジェクトを導入したりできる。

2.1 マップにものを配置する

MapEditor でマップを作成する作業は、もの (オブジェクト) を配置するステップと、配置したものをシミュレータに解釈させるためにリンクするステップにわけられる。各ステップは往復しながら作業できるが、はじめに配置作業を説明する。

起動時の画面は図 2 のようになっている。黒い領域がマップを書くところで、右側が各種ツールパレットである。オブジェクトパネルのアイコンをクリックして、形状をえらんでペイントソフトのようにオブジェクトを配置していく。オブジェクトには障害物、スタート、ゴールと Waypoint (道標、いりくだマップでエージェントを誘導する) がある。また歩行者をあらかじめ配置することもできる。くわしい

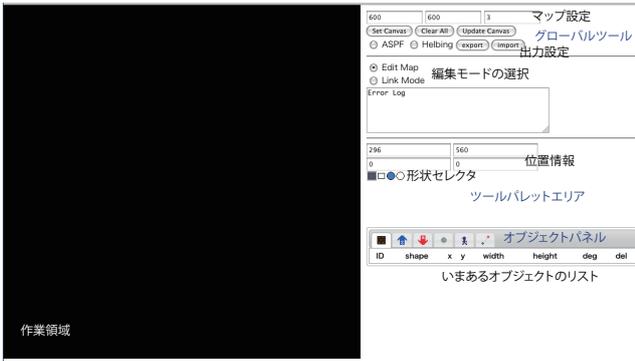


図 2 開始時の様子

操作方法は拙作 “MapEditor for Pedestrian MAS – Manual for Users and Developers” [2] を参照いただきたい。本アプリケーションの特徴として、マップのサイズを任意にかえられること、オブジェクトの位置を数値指定できることがあげられる。

2.2 オブジェクトをリンクする

さまざまなオブジェクトを配置した結果、画面は図 3 のようになった。次にリンクをおこない、オブジェクト関連付け、エージェントがこの空間を移動できるようにする。

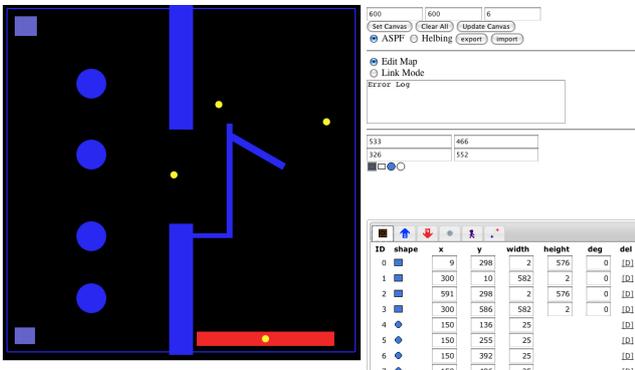


図 3 オブジェクトの配置がおわった状態

Pedestrian	Goal
Pedestrian	Waypoint
Waypoint	Waypoint
Start	Goal
Start	Waypoint
Goal	Start

リンクはリンクモードで、始点オブジェクトと終点オブジェクトを指定することでおこなう。リンクには上の組み合わせがある。

関連付けすると、リンクが黄色い線でえがかれて、図 4 のようになる。

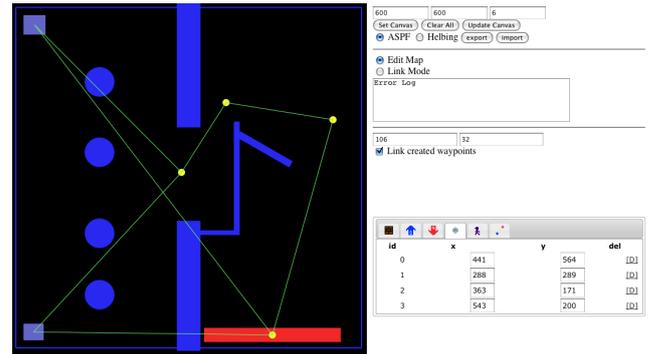


図 4 リンク処理がおわった状態

2.3 データの出力

編集結果を出力し、artisoc に読み込ませる方法を説明する。画面右上の入出力フォームで出力するモデルの形式 (第 3 章参照) を選択し、“export” ボタンをおすと、出力ダイアログが表示される。ASPF では設定項目はタイトルのみだが、Helbing モデルでは加えて時間分解、ひとマスの大きさ、目標設定ルーチンの指定ができる。

設定を入力して Export をおすと設定したタイトルで zip ファイルがダウンロードされる。内容は次のようになっている。

TITLE.model artisoc で実行できる model ファイル (TITLE は出力時に指定したもの)。

background.jpg artisoc で実行するとき読み込まれる背景画像。

temp.json 不要オブジェクトをとりのぞくまえのオブジェクトのデータ。

raw.json 不要オブジェクトをとりのぞいたあとのオブジェクトのデータ。import につかう。

3 モデルについて

MapEditor が利用する歩行者モデルは 2 種類ある。ASPF モデルと、Dirk Helbing 氏が中心となって開発されたモデル (Helbing モデル) である。

次の動作は両者に共通である。

- エージェントはルールにしたがって Waypoint(WP) をめざす。

- WP 付近に到着したら WP のもつ、次の WP の情報を取得し、それをめざす。
- エージェントはマップ上にあらかじめ配置されるか、スタートで生成される。
- スタートの枠内に一定速度でエージェントが生成される。エージェントはスタートからゴールと最初にめざす waypoint を指定される。
- ゴールについてエージェントは消滅する。ただしゴールがつぎのスタートにリンクされていれば、そのスタート上にエージェントがうまれる。

3.1 Helbing モデルの詳細

Helbing モデルはエージェントが感じる力をまわりのエージェントや静物との関係で記述し、その微分方程式を近似的に解くことで速度、位置を決定する。 i 番目のエージェントが感じる力はつぎのようにあらわされる [3]。

$$\begin{aligned} \vec{f}_i(t) &= \frac{v_i^0(t)\vec{e}_i(t) - \vec{v}_i(t)}{\tau_i} + \sum_{i \neq j} (\vec{f}_{ij}^{\text{soc}} + \vec{f}_{ij}^{\text{ph}}) + \sum_b \vec{f}_{ib} \\ \vec{f}_{ij}^{\text{soc}} &= A_i \exp\left(\frac{D}{B_i}\right) \vec{n}_{ij} \left(\lambda_i + (1 - \lambda_i) \frac{1 + \cos(\phi_{ij})}{2} \right) \\ \vec{f}_{ij}^{\text{ph}} &= k\Theta(D)\vec{n}_{ij} + \kappa\Theta(D)((\vec{v}_j - \vec{v}_i) \cdot \vec{t}_{ij})\vec{t}_{ij} \\ \vec{f}_{ib} &= (A'_i \exp\left(\frac{r - d_{ib}}{B_i}\right) + k\Theta(r_i - d_{ib}))\vec{n}_{ib} - \\ &\quad \kappa\Theta(r_i - d_{ib})(\vec{v}_{ib} \cdot \vec{t}_{ib})\vec{t}_{ib} \\ D &= 2r - |\vec{x}_i - \vec{x}_j| \end{aligned}$$

この方程式には定数が多い。定数はエージェントルールを直接編集することで変更できるので、マップにあわせて定数を調節し、リアルな動きを再現させたい。また微分方程式を解く過程で力や速度の情報がえられるので、どれぐらいの圧力が発生するかといった分析を行うのに適している。だが、Helbing モデルは ASPF モデルにくらべて計算量が多いため、実行速度が遅い。そのためエージェントが多く、空間がひろい場合には ASPF をつかうべきである。

3.2 ASPF モデルの詳細

ASPF (Agent-based Simulator of Pedestrian Flows) は、名古屋工業大学 兼田研究室で開発された広い空間に適した歩行者モデルである。今回の Map Editor の開発にあたり、利用を快諾くださった兼田敏之教授、ならびにモデルの開発に尽力された研究室のメンバーの方々にはこの場を借りて御礼もうし

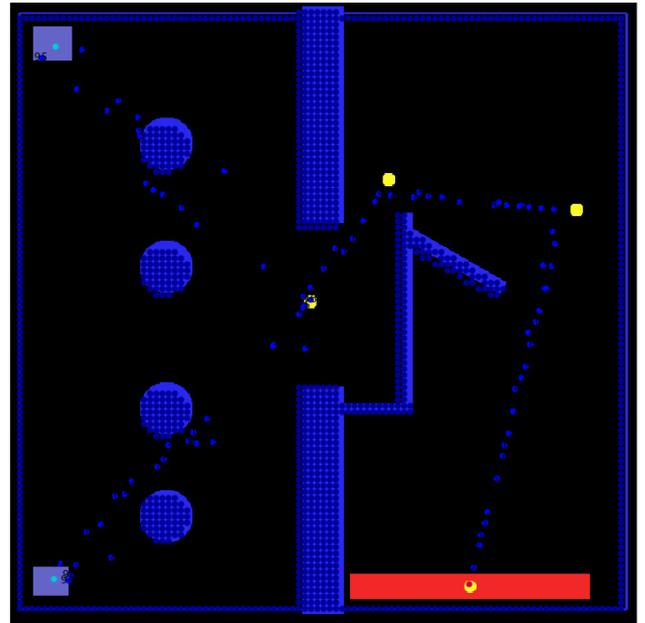


図5 ASPF 実行の様子

あげたい。本シミュレーションでは ASPFver4[4] をベースに、細部を変更して実装した。

ASPF の特徴はエージェントがセルオートマトンのようなルールにしたがうので実行速度が速い。連続空間で使用することができる点にある。また、Waypoint も最初からとりこまれているため、大規模な施設などでのシミュレーションに適する。

ASPF モデルの 1 ステップの時間間隔は 0.5sec/step、ひとマスの大きさは 0.4m/cell である。これらは universe.dt と universe.cellsize で指定している。

4 外部プログラムを用いたシミュレーションの今後

MapEditor を開発するうえでは、artisoc の方針をうけつぎ、「だれでも簡単に」という点を重視した。マウスとアイコンの直感的な操作でマップ作成をおこない、それをダイレクトに artisoc 上でシミュレーションできるようにした。これによって artisoc ユーザの裾野がさらにひろがり、ライトウェイトシミュレーションが世間に普及することを願っている。

同時に外部プログラムである MapEditor と model ファイルテンプレートを GPL で公開し、このシステムのさらなる拡張や改造が簡単にできるようにも配慮した。MapEditor に搭載したテンプレートは歩

行者を移動させる以外の機能をもたず、シミュレーションして得たいデータを収集することはできない。これはそれぞれのニーズにあわせて実装すべき部分であると考えたからである。さらに現実装では、Helbing の提唱する相互作用の要素がまったく入っていないかったり、ASPFver4 で研究されたイベントの実施という状況変化を再現する方法がないなど、改善すべき点は数々ある。

ある目的に特化すればするほど、その範囲内で利用することが容易になる。本プロジェクトで提案した、テンプレートとなるモデルとそれを設定するツールを組み合わせることで簡単なシミュレータを実現するという方法は歩行者シミュレーションにかぎらず、どんな分野のシミュレーションにも応用可能である。本研究がさきがけとなって、さまざまな目的に特化した簡易シミュレーション環境が開発されることを期待する。

参考文献

- [1] 兼田敏之代表編者、構造計画研究所創造工学部、名古屋工業大学兼田研究室 (2010) 『artisoc で始める歩行者エージェントシミュレーション 原理・方法論から安全・賑わい空間のデザイン・マネジメントまで (人工社会の可能性 3)』 発行構造計画研究所 発売書籍工房早山.
- [2] 富田寛 (2011) “MapEditor for Pedestrian MAS – Manual for Users and Developers” <https://sites.google.com/site/tomykaira/>.
- [3] Dirk Helbing, Illes J. Farkas, Péter Molnár, and Tamás Vicsek(2002) “Simulation of Pedestrian Crowds in Normal and Evacuation Situations” pp.21-58 in: M. Schreckenberg and S. D. Sharma (eds.) Pedestrian and Evacuation Dynamics (Springer, Berlin).
- [4] 何 雁峰, 指導教員 兼田 敏之 (2007) “目的地移動機能を有する自律的歩行者エージェントによる群集シミュレーションの研究” 第7回 MAS コンペティション参加作品.