

# MapEditor for Pedestrian MAS – Manual for Users and Developers

東京大学理科一類 1 年 富田寛

2011 年 2 月 11 日

## 1 はじめに

MapEditor for Pedestrian MAS にご興味をいただいております。MapEditor for Pedestrian MAS は歩行者シミュレーションをより容易に実行できるようにしようと開発された、artisoc(<http://mas.kke.co.jp/>) の補助プログラムです。

artisoc はマルチエージェントシミュレーションの門戸を計算機にくわしくない人々に開放した、画期的な汎用シミュレータです。モデルの構築につかう言語はわかりやすく、必要とされる実行環境も普通のパソコンで十分です。

いっぽう学園祭などのイベントの空間レイアウトや商品の配置などをおこなうときに、専門家ではない企画者が歩行者の動きを予測し、シミュレートするニーズがあります。しかし従来の歩行者シミュレータは建築の場面で利用されるおおがかりなものがほとんどでした。また、artisoc が簡単に利用できるといっても、歩行者シミュレーションに特化しているわけではないので、一定の労力をエージェントルール作成についやさねばなりません。いっぽう既存のモデルを再利用しようとおもっても、ほとんどのモデルではオブジェクトの情報がモデル内にうめこまれており、整合性をたもって自分のほしい空間をつくりだし、さらにその配置をかえてくりかえし試験することは困難でした。

ほんらいシミュレータとそこでシミュレーションされる対象 (空間) は完全に分離しているべきです。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用のマップ作成アプリケーションを作成しました。これが MapEditor for Pedestrian MAS です。

このマニュアルでは MapEditor を基本的な使い方を説明しています。ぜひ使用前にご一読ください。

## 2 セットアップ方法

MapEditor は出力部分に PHP(<http://www.php.net/>) を利用しているので、ふつうのソフトウェアとはちがい、インストールが少々面倒です。はじめにセットアップ方法を解説します。

MapEditor は HTML5(Canvas)、JavaScript(jQuery、jQueryUI をふくむ)、PHP を利用しています。基本的に MapEditor をどこかの HTTP サーバにセットアップし、そこにブラウザからアクセスしてつかうことになります。

このうち HTML5 は最近登場した技術なので、一部のブラウザでは対応していません。Ma-

pEditor は Firefox 3.6 と 4.0b7 および 4.0b8 での動作を確認しております。あたらしい Google Chrome、Opera などでは動作するはずですが、もし不具合があるようでしたら、ご連絡ください (付録 B 参照)。

PHP (“PHP: Hypertext Preprocessor”) はウェブページの構築によく利用される汎用スクリプト言語です。MapEditor が利用する PHP は HTTP サーバのうえに設定された CGI 版でなければいけません。PHP は編集したマップデータを POST メソッドをつかって送信し、zip ファイルのかたちにして出力するために利用しています。

以下の説明は個人で使用することを主としています。研究室などで共同で利用する場合は所属機関の提供しているサーバ (もしあれば) の利用をおすすめします。またセキュリティなどに関してはとくに考慮していません。以下の説明にしたがうことで発生したいかなるトラブル、不利益にも私は関与しません。

## 2.1 ダウンロード

<http://sites.google.com/site/tomykaira/> から最新版をダウンロードし、解凍します。index.html というファイルやいくつかのフォルダが得られます。

## 2.2 ためしにしてみるには

MapEditor が PHP を利用するのは出力部分だけなので、基本的なマップ作成の機能は面倒なインストールなしでも体験できます。2.1 節で得た index.html をブラウザで開きます。図 1 のような画面がでたら成功です\*<sup>1</sup>。

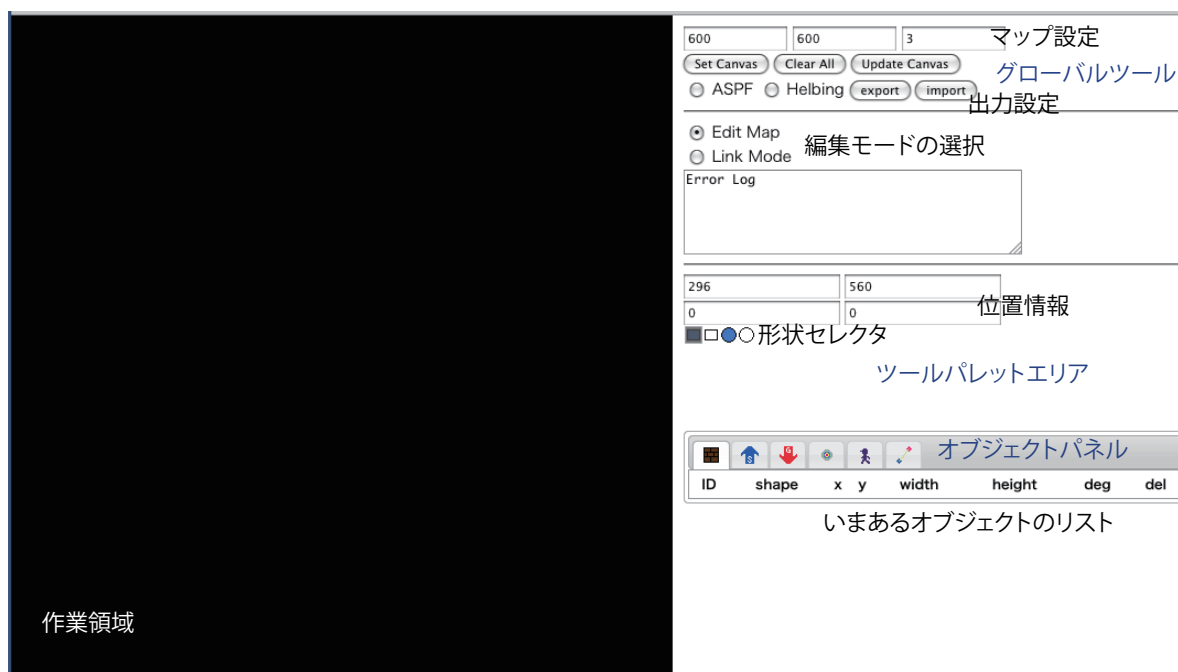


図 1 起動時の画面

この状態では PHP が利用できないので、zip ファイルの出力ができないことをご了承ください。

\*<sup>1</sup> スクリーンショットはすべて Mac OS X Leopard、Firefox 上でのものです。環境によってボタンや配色が多少異なる場合があります

## 2.3 Windows でのインストール方法

Microsoft Windows に MapEditor 開発環境を導入する方法を解説します。Windows Vista 以降をご利用の方は“Windows 7/Vista で自宅サーバー”(http://awoni.net/personal-site)を参照し、IIS をもちいてサーバを構築することをおすすめします。「2.IIS をインストール」と「4.PHP のインストール」の指示にしたがってください。今回の使用ではローカルからしかアクセスしないので、このふたつで十分です。Windows XP 以前をご利用の方は“04WebServer”(http://soft3304.net/04WebServer/) のうえに構築することをおすすめします。“Windows に PHP5 をインストールする | アマグラの日々”(http://ameblo.jp/amagra/entry-10018402853.html)などを参考にしてください。

2.1 節で得たファイルとフォルダをサーバのルートディレクトリ (デフォルトでは C:\inetpub\wwwroot) に mapeditor というフォルダをつかってコピーします。そのあとブラウザから http://localhost/mapeditor/ にアクセスしてください。2.2 節でテストしたのとおなじ画面がでてきて、かつ出力もうまくいけば成功です。うまく画面がでてこない場合は IIS がうごいているか、ディレクトリの階層設定が正しいかを確認してください。うまく出力できない場合は PHP が正常に動作しているかを確認してください。

## 2.4 Mac OS X でのインストール方法

Mac OS X に MapEditor 開発環境を導入する方法を解説します。はじめにウェブシェアリングを有効にします。Mac OS 10.5 では次のような操作でした。バージョンによって多少ことなるかもしれません。

1. メニューからシステム環境設定を開く。
2. 共有をひらき、Web 共有をひらく。
3. 左のチェックボックスにチェックをいれ、Web 共有:入になることを確認する。

つぎに“PHP: バンドルされている PHP の使用法 - Manual”(http://www.php.net/manual/ja/install.macosx.bundled.php)にしたがい、apache で PHP を有効にします。今回はローカルでのみ動作させるため、バージョンアップグレードはかならずしも必要ではありません。

設定がおわったら自分のユーザディレクトリ (デフォルトでは /Sites/) に 2.1 節で得たファイルとフォルダをコピーします。ブラウザから http://localhost/~あなたのユーザ名/mapeditor/ にアクセスし、2.2 節でテストしたのとおなじ画面がでてきて、かつ出力もうまくいけば成功です。うまく画面がでてこない場合は apache がうごいているか、ディレクトリの階層設定が正しいかを確認してください。うまく出力できない場合は PHP が正常に動作しているかを確認してください。

## 2.5 Unix などでのインストール方法

Unix 系の OS に MapEditor 開発環境を導入する方法を解説します。PHP が実行できる HTTP サーバがあることは前提とします。詳細についてはシステム管理者に相談するか、それぞれのディストリビューションの解説をさがすか、PHP マニュアルの該当箇所 (http://www.php.net/manual/ja/install.unix.php など) を参照してください。

いま仮に、HTTP サーバのルートディレクトリが/var/www であるとします。はじめに /var/www/mapeditor というディレクトリを作成します。ここに 2.1 節で得たファイルとフォルダをコピーします。ブラウザから <http://サーバのアドレス/mapeditor/> にアクセスしてください。2.2 節でテストしたのとおなじ画面がでてきて、かつ出力もうまくいけば成功です。うまく画面がでてこない場合はサーバがうごいているか、ディレクトリの階層設定がただしいかを確認してください。うまく出力できない場合は PHP が正常に動作しているかを確認してください。

## 3 マップをつくる

インストールが完了したら、実際にマップをつくってみます。マップ作成でつかうツールを説明します。

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関連付け (リンク) するステップにわけられます。

起動時の画面は図 1 のようになっています。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す。ツールパレット等の名称は以降の説明でつかいます。

### 3.1 全体の設定

はじめにマップサイズなどの設定方法を紹介します。グローバルツールをみてください。マップ設定の数値は左から順に、マップの幅、高さ、倍率です。倍率とは artisoc で大きいマップを処理するのは負荷がおおきいので、表示されているマップをこの倍率で縮小した空間をつかう、という意味です。これらの数値が 600,600,3 であれば、マップ編集では 600×600 のキャンバスで作業し、実行時にはそれが 200×200 になるということになります。

マップサイズはどんなモデルを構築したいかによって柔軟に変更してください。また、ASPF をつかう時に倍率をちいさくすると Wall オブジェクトの数がふえすぎて、実行速度がいちじるしく低下します。倍率は精度の要求がゆるすかぎり高く設定することを推奨します。

数値をかえたら Set Canvas ボタンで反映します。このとき既存のオブジェクトはすべて消えてしまいます。オブジェクトをたもったままマップサイズや倍率を変更したいときは、4.3 節の方法で JSON ファイルを編集してください。

Clear All ボタンはいまあるオブジェクトをすべて削除するボタンです。Update Canvas ボタンは編集結果がマップに反映されないときに反映させるボタンです。

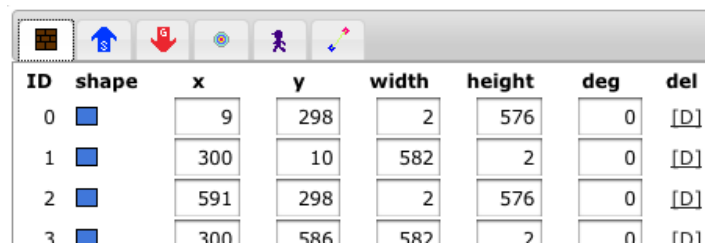
### 3.2 オブジェクト配置する

オブジェクトパネルに表示されているオブジェクトには障害物、スタート、ゴールと Waypoint(道標) の 4 種類と、事前に配置する歩行者エージェントがあります。エージェントをあらかじめ配置するのは、ある状況から避難するのにかかる時間を測定するなどの目的です。

#### 3.2.1 障害物 (青)

障害物オブジェクトはエージェントがぶつかるもの、壁や仕切りや机などをあらわします。パラメータは基本的な位置情報だけです。位置情報とはつぎの項目です。

shape 形状 (長方形、長方形の縁、円、円の縁)



ID	shape	x	y	width	height	deg	del
0	■	9	298	2	576	0	[D]
1	■	300	10	582	2	0	[D]
2	■	591	298	2	576	0	[D]
3	■	300	586	582	2	0	[D]

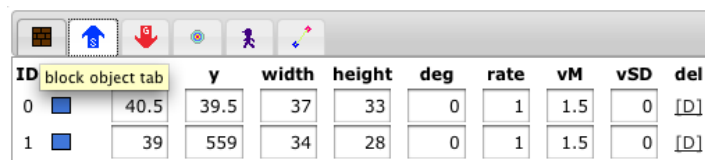
図 2 障害物オブジェクト

x,y 中心の x,y 座標

width,height 高さ、幅 (円のときは半径と中心角)

deg 角度 (度数法で x 軸から反時計まわり)

### 3.2.2 スタート (水色)



ID	block object tab	y	width	height	deg	rate	vM	vSD	del
0	■	40.5	39.5	37	33	0	1	1.5	0 [D]
1	■	39	559	34	28	0	1	1.5	0 [D]

図 3 スタートオブジェクト

スタートオブジェクトからはエージェントが発生します。スタートオブジェクトは多くのパラメータをもちます。これらを適切に設定しないと、出力時にオブジェクトがきえてしまいますので、注意してください (後述)。

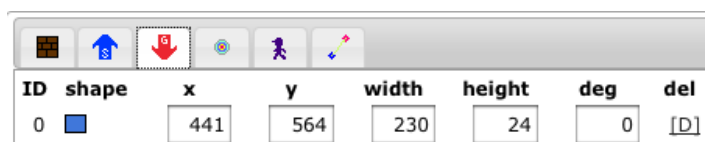
スタートからはゴールと waypoint にリンクしなければなりません。ゴールはそのスタートで発生したエージェントの最終目的地をしめし、waypoint はエージェントが最初にむかうべき方向をしめします。ただし、ゴールにはかならず waypoint があるので、とくに指定しない場合はゴールの waypoint が自動的にわりあてられます。

rate 一定時間 (秒) あたりにそのスタートから発生するエージェントの数。

vM そのスタートから発生するエージェントの平均速度。

vSD そのスタートから発生するエージェントの速度の分散。エージェントの速度はこの平均と分散をもつ正規分布にしたがう。

### 3.2.3 ゴール (赤)



ID	shape	x	y	width	height	deg	del
0	■	441	564	230	24	0	[D]

図 4 ゴールオブジェクト

ゴールはスタートとリンクされるオブジェクトです。あるスタートで発生したエージェントが

ゴールオブジェクトに接すると移動しおわったとみなして消滅します。

ゴールは位置情報以外のパラメータをもちませんが、スタートとリンクすることができます。スタートに向かってリンクしたゴールに到着したエージェントは、次のスタートのうえのランダムな点から再出発します。これは階段による階の移動などを実装するための機能です。障害物オブジェクトでくぎられたふたつの部屋を用意し、図 5 のようにリンクすれば擬似的に部屋をとって階段をおりる様子が再現できます。

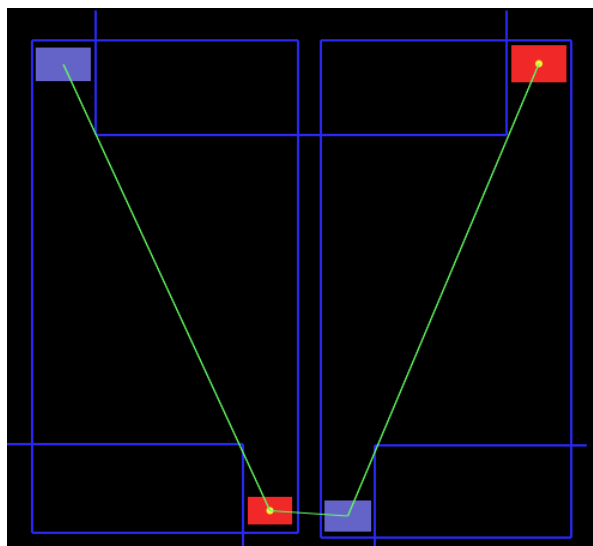


図 5 Goal-Start リンクの使い方。階段のあるマップの例

### 3.2.4 Waypoint(黄色)

id	x	y	del	
1	284.5	460.5	<input type="text" value="[ID]"/>	
2	509	87.5	<input type="text" value="[ID]"/>	

図 6 waypoint オブジェクト

waypoint オブジェクトはスタートからゴールまでエージェントが進むときに、障害物をうまくさけられるように道をアドバイスするためのものです。図 7 のようにいりくんだマップでは、ゴールに直接すすんでしまうと障害物にあたってたどりつけなくなります。エージェントがある waypoint に関連付けられた場合、エージェントはその点に向かって直進し、その点の付近にたどりついたら、waypoint のエージェントから次の点をおしえてもらってそちらに進みます。

waypoint はつぎに進むべき方向を指示するために、ほかの waypoint にリンクされます。例外はゴールとセットになっている waypoint で、次の waypoint を指定しなくてもよいです。

waypoint を連続的につくるときにリンクする手間がはぶけるように、自動リンク機能があります。ツールパレットの “Link created waypoints” にチェックをいれると、つぎに設置される waypoint から先が連続してリンクされます。

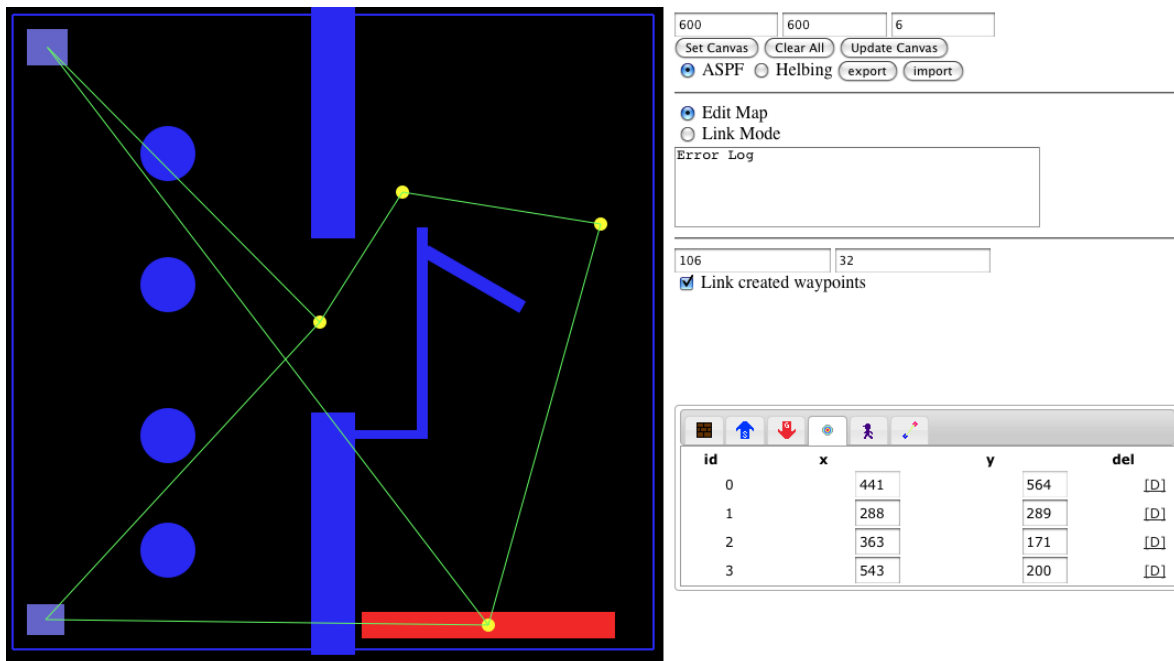


図 7 いりくんだマップの例

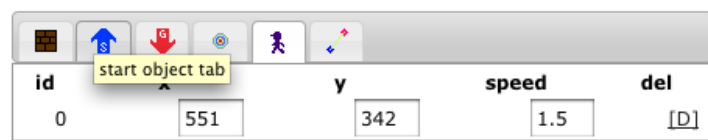


図 8 歩行者エージェント

goal ID:

first WP ID:

speed:

図 9 歩行者エージェントのツールパレット

### 3.2.5 歩行者エージェント

歩行者エージェントもマップに配置できます。歩行者エージェントには 3 つのパラメータがあります。

**goal ID** そのエージェントがめざすゴールの ID を記入します。ID はパネルでゴールをえらび、表の左端の項目で確認します。

**first WP ID** エージェントが最初にめざす waypoint の ID を指定します。ID の確認のしかたはゴールとおなじです。最初の waypoint についたあとは、waypoint の連鎖の誘導にしています。

**speed** 歩行速度 (m/s)。

また、整列している歩行者を簡単に設置するために、パターン配置機能があります。図 10 のようなダイアログでパターン配置ができます。

Put Pedestrian in Pattern

始点	10	Y	110
X			
間隔	30	Y	30
X			
個数	5	Y	4
X			
Goal	-1	firstWPID	-1
ID			
Speed	1.5		

Apply

Cancel

図 10 エージェントのパターン配置用ダイアログ

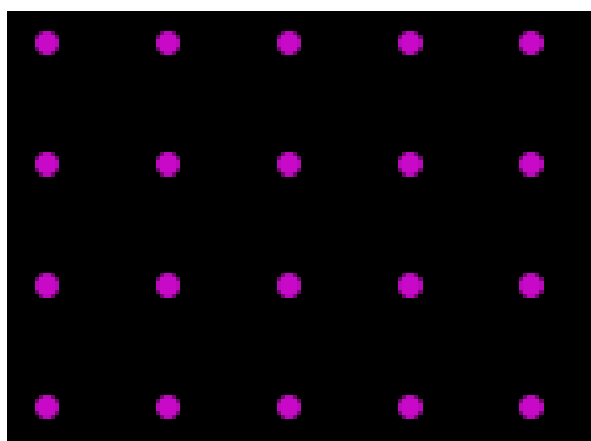


図 11 パターン配置されたエージェント

ではつぎに、以上のオブジェクトを実際に配置する方法を解説します。オブジェクトは配置方法の点から、大きさのある障害物、スタート、ゴールと、大きさのない waypoint、歩行者に分けられます。

大きさのあるオブジェクト（障害物、スタート、ゴール）はオブジェクトパネルでオブジェクトを選択し、ツールパレットエリアのオブジェクトの形状セレクトで形状を選択したあと、マップ上でマウスをドラッグして図形を書くことで配置します。ペイントソフトの操作感覚です。位置や大きさはあとから修正できるので、まずおおまかな形をかくといいでしょう。

点オブジェクト（waypoint、歩行者）を設置するには同様にオブジェクトを選択したあと、設置したい場所でクリックするだけです。waypoint と agent のツールパレットのつかいかたについては上の各項目を参照してください。



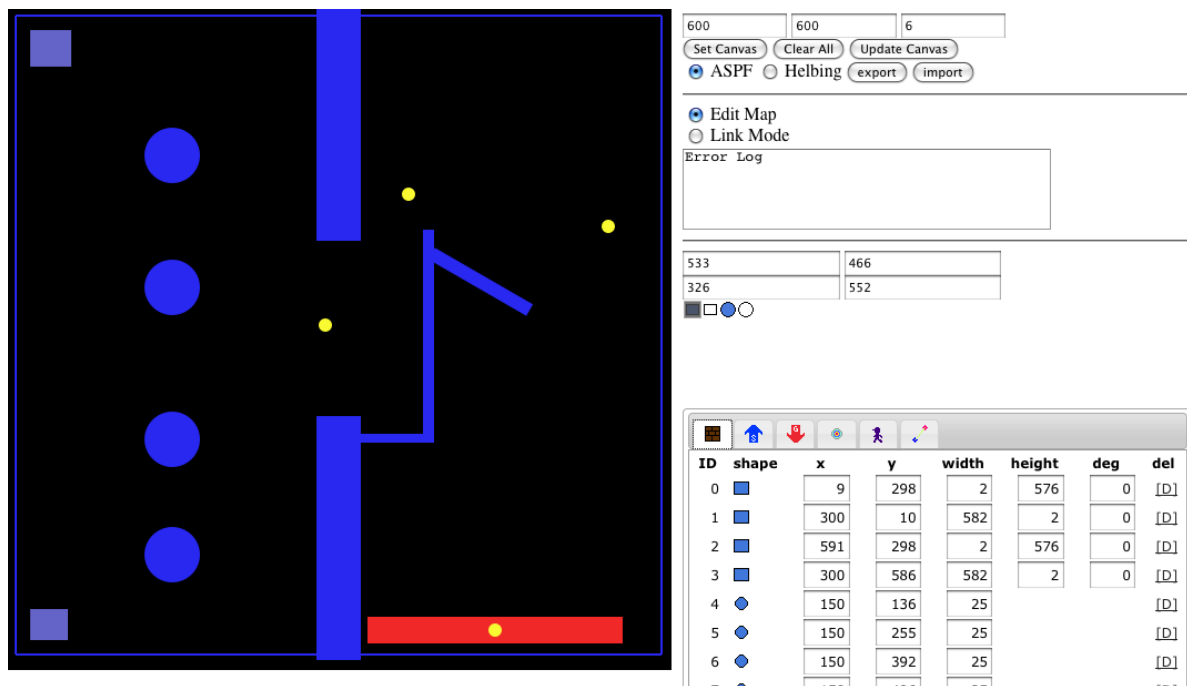


図 12 オブジェクト配置後の様子

### 3.3 オブジェクトの微修正

オブジェクトを配置するときにはマウスをつかいますが、厳密な位置を設定するには数値で指定したほうが楽です。また、すでに作ったオブジェクトの位置や大きさをかえたり、向きをかえたりしたい時もあります。そのときはオブジェクトパネルのオブジェクトのリストで、設定したいパラメータを変更します。

たとえば、3 つめの障害物の幅をかえたい場合は、障害物のタブを選択し、上から 3 つめのオブジェクトの width をクリックし、新しい幅を入力します。入力がおわったらエンターキーをおすとその値が反映されます。選択できない値は変更することができません。マップ上のどのオブジェクトが何番目のオブジェクトかわからないときは、オブジェクトのリストのうえでクリックをおしつづけている間、そのオブジェクトが黄色くマークアップされます。

### 3.4 リンクを設定する

マップの概形が完成したらリンクをはり、シミュレーションできる状態にととのえます。どんなリンクが可能かは上述しましたが、表のかたちで再掲します。

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

これ以外のリンクを設定しようとしてもできません。また、スタートとゴールは双方向でリンク可能なので、どちらからどちらにはっているかならず確認しましょう。

リンクをはるには、まずモードをきりかえます。モードセクタ (図 13) で “Link Mode” をえらんでください。ツールパレットの表示がかわったとおもいます。From と To のどちらがチェックされているかを確認しながら、リンクするオブジェクトを選択します (図 14)。リンクできたら、オブジェクトパネルでリンクのパネルを確認してください。リンクが追加されているはずです。



図 13 モードセクタ

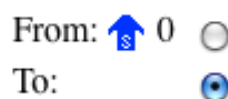


図 14 リンクするオブジェクトをえらんでいる様子

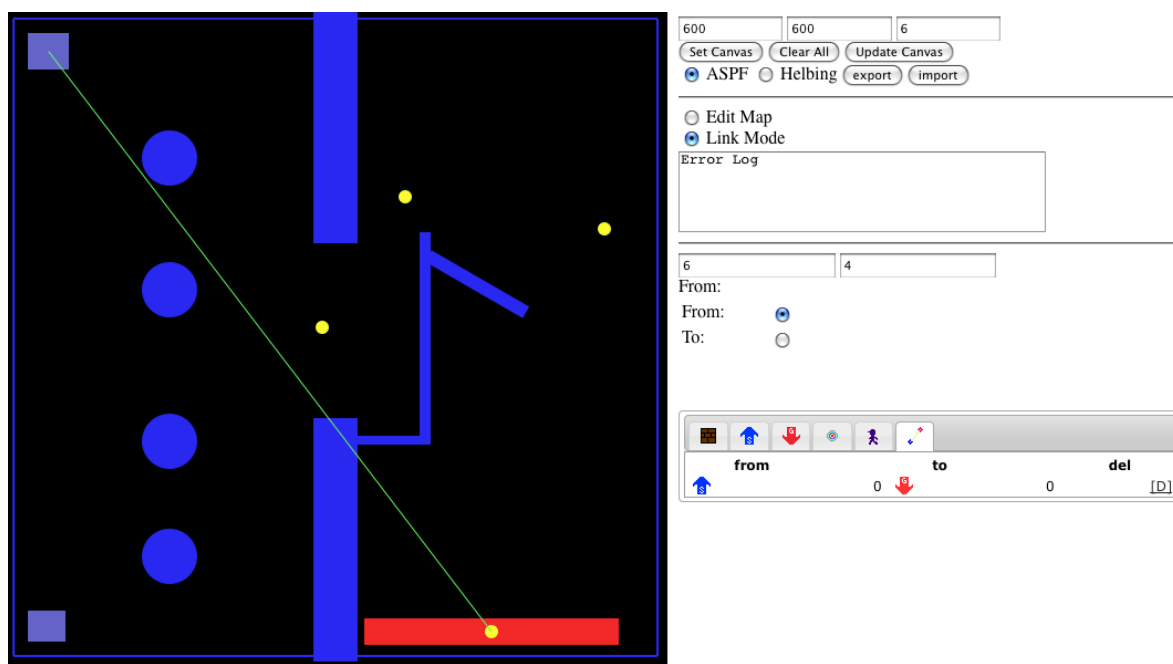


図 15 ひとつリンクしたところ

この操作をくりかえして図 16 のようにマップを完成させます。

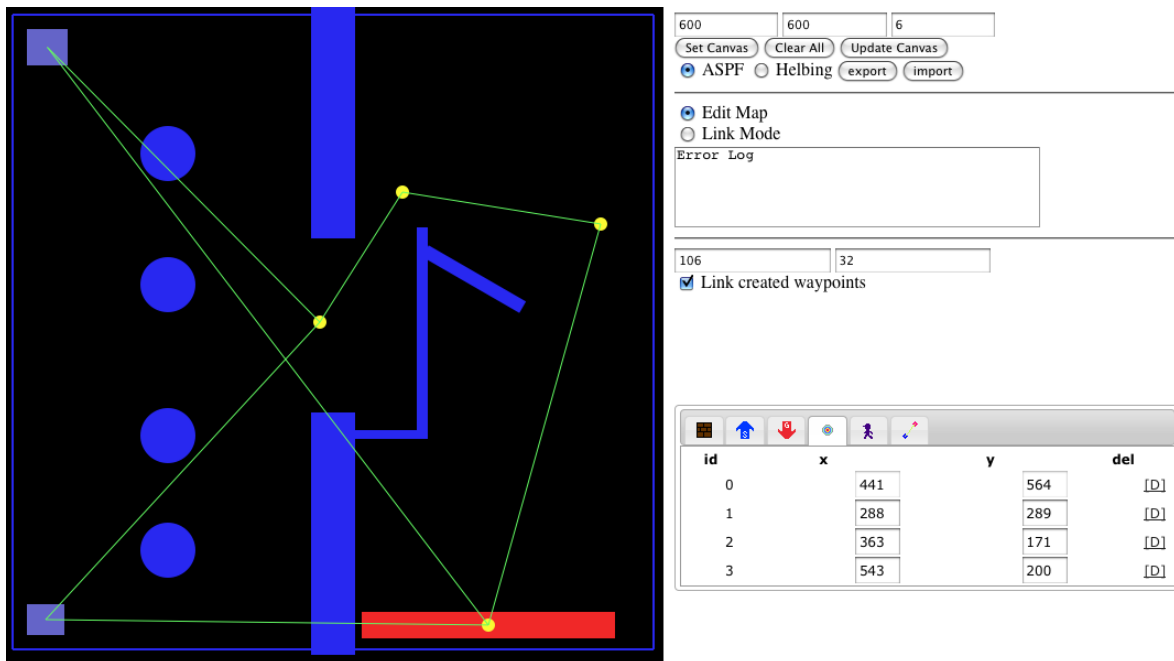


図 16 リンクがおわった状態



図 17 グローバルツールの入出力部分

### 3.5 出力する

出力するには入出力部分で出力につかう歩行者モデルを選択し、export ボタンをおすだけです。歩行者モデルは二種類ありますので、用途にふさわしいものを選んでください (モデルの詳細については拙作 “MapEditor をもちいた汎用 Pedestrian MAS”, <http://sites.google.com/site/tomykaira//presentation.pdf>)。出力の際に不要なオブジェクトがのこっていると、それらはすべて削除されてしまうことに注意してください。

不要なオブジェクトというのは、リンクされていない waypoint やエージェントを発生させないスタートなどです。すなわちそのオブジェクトのもつべき役割がはたせるようにリンクやパラメータが設定されていないものは削除されます。export した瞬間にオブジェクトがきえてしまったら、temp.json(4.3 節参照) から編集データを復元し、どこがわるかったかを検討して再度出力してください。

export ボタンをおすと出力設定のダイアログ (図 18) がひらきます。ASPF を選択した場合にはタイトルだけ入力してください。タイトルは zip ファイルの名前と、.model ファイルの名前に利用されます。Helbing を選択した場合にはタイムスライス、ひとマスの大きさ、ゴールをめざすか waypoint を目指すかを設定できます。設定をおえ、ダイアログの export ボタンをおすと PHP プログラムが実行され、zip ファイルをダウンロードするようにうながされますので、それをダウンロードして保存します。このファイルの使い方については 4 章で詳述します。

また、MapEditor には編集状態を保存する機能がありません。したがってブラウザを終了したり、まちがってページを再読み込みしてしまうと編集データがきえてしまいます。一段落したとき

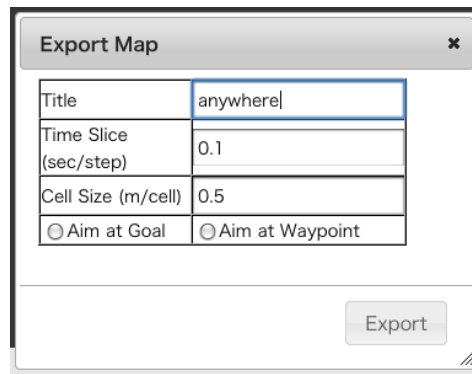


図 18 出力設定ダイアログ

には export して状態をセーブしておくことをおすすめします。このとき上述の機能で一部のオブジェクトが消えてしまうかもしれませんが、データは全部 temp.json に保存されるので安心してください。

## 4 出力ファイルについて

マップがぶじ出力されたら、artisoc 上で実行したり、グループのメンバに配布したりしなければなりません。本章では出力された zip ファイルのとりあつかいについて説明します。

### 4.1 zip ファイルの使用方法

zip ファイルにはつぎのファイルがふくまれます。

TITLE.model artisoc で実行できる model ファイル。TITLE は指定したタイトル。

background.jpg artisoc で実行するときに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。

### 4.2 model の実行方法

artisoc の academic か professional を起動し、model ファイルを開きます。パラメータやルールの変更がなければ、そのまま実行するだけです。変更する場合は 5.1 を参考にしてください。

### 4.3 JSON ファイルの使用方法

zip ファイルにふくまれる、temp.json と raw.json は MapEditor にインポートできるデータです。以前の編集内容を復元したい時は、json ファイルをテキストエディタで開き、中身をすべてコピーして、インポートダイアログにペーストしてください。インポートダイアログは MapEditor の import ボタンをおすと開きます。

また、この内容を直接編集することもできます。コピーペーストを活用して編集することで、パターン状のオブジェクトをらくに配置できるでしょう。オブジェクトのパラメータについては 5.2 節を参照してください。

マップの設定を変える場合は mapinfo 項目を変更します。

```
"mapinfo": {"x": 600, "y": 600, "ratio": 6}
```

それぞれ幅、高さ、倍率をあらわしますので、任意の値にかえてロードすればオブジェクトはそのままでサイズを変更できます。

## 5 改造する

MapEditor はすべて GPL ライセンスのもとで配布されています。したがってライセンスの要求事項をまもるかぎり、自由にスクリプトに手をくわえることができ、その成果物を共有することができます。ここでは MapEditor の仕組みを簡単に紹介し、改造する際の手引きをします。

### 5.1 モデルを改造する

標準のモデルでは歩行者が移動するだけで、パラメータをかえて実行したり、統計的データをとったりといったシミュレーションらしい機能は実装していません。これはどんな歩行者シミュレーションにも使えるようにというコンセプトによるものです。

くりかえし実行や条件分岐、さらには実装されていない機能 (たとえばイベント開催など動的な条件の変更) をつかうためにはモデルを改造しなければなりません。MapEditor のつよみは同じモデルのマップをかきかえてつかえることですから、ユーザが自分でテンプレートをかけることが重要だと考えています。本節ではエージェントルール以外の部分の構成をかんたんに紹介します。

テンプレートファイルは ASPF\_template.model と Helbing\_template.model です。テキストエディタで開くと、通常の model ファイルを開いたときとおなじようなコードが表示されます。違いは JavaScript がマップデータをうめこむための置換文字列があることです。置換文字列は \$ または \$\$ でかこまれています。置換の詳細については 5.2 節を参照してください。

ふたつの歩行者モデルを採用していますが、この二者で異なるのは歩行者モデルの中核と Wall エージェントだけで、のこりのコードはほとんどおなじです。Wall エージェントは ASPF では壁を探索するために、壁がエージェント化されていなければいけないために追加しました。

#### 5.1.1 Universe ルール

標準ではなにもありません。オブジェクトの位置情報などは変数の初期化コードをうめこむことでおこなっています。統計をとるときなどに利用してください。

#### 5.1.2 Pedestrian エージェントルール

歩行者モデルの核となる歩行者のルールです。

Init では my.init 変数の初期化のみおこないます。これは歩行者がほかのエージェント (スタートやほかの歩行者) によって生成されたとき、それらが変数をあたえるまえに初期化がおこなわれてしまうため、Init に初期化処理をかけないためです。通常のプログラミング言語ではコンストラクタなどがもちいられますが、そのかわりに my.init というフラグがたっていないければ (第一回のエージェントルールの実行。エージェントルールはかならず生成側の処理がおわってから実行される) 初期化をするというルールをかきました。常に一定の値をとる場合は Init で初期化してもいいですが、そうでない場合はつぎの init part に書かなければいけません。

Step でははじめに変数を宣言します。つぎに init part があり、ここでは変数の単位の変換 (メートル、秒からマス、ステップへ)、ゴールの確認、はじめの waypoint の ID 取得をおこないます。内部の数値のあつかいは基本的にすべてマス、ステップ、度数法でおこないます。初期速度と方向がきまります。

つぎに normal part、ふつうの Step にあたる部分があります。はじめにゴールに到着したかと waypoint に到着したかがチェックされます。その後移動ルールがつづき、最後にエージェントが画面内にいるかを確認する関数でおわります。

arriveGoal 関数 (ゴールに到着したときに呼ばれる) を説明しておきます。ゴールが nextStartID 属性をもっているかをチェックし、もっていればその ID をもつスタートをさがしだして、そのうえに移動します。そしてあたらしいゴール ID と WPID をスタートからもらいます。nextStartID 属性をもっていない (-1) ときはそのまま自殺します。

### 5.1.3 Start エージェントルール

スタートがエージェントを生成するとき (generateAgt 関数) も arriveGoal 関数とほぼおなじです。スタートは経過ステップ数をかぞえ、それと generateSpeed の積が 1 以上ならエージェントを生成します。エージェントの生成・消滅について統計をとる場合はこのへんでカウント処理をおこなえばいいでしょう。

## 5.2 MapEditor を改造する

MapEditor は基本的に JavaScript でかかれています。プログラミングになれたしんだ人ならばソースを直接読んで編集できるとおもいますが、全体の構造などよみとく上でのヒントになる情報を掲載します。

index.html ブラウザでよみこむ際に表示される画面の構造を記述。HTML はこれのみ。

init.js 起動時によみこまれるスクリプトが中心。

object.js 歩行者をのぞくオブジェクトの管理をおこなうクラスを記述。

agent.js 歩行者の管理をおこなうクラスを記述。

paint.js 描画関数が中心。

ui.js HTML との相互作用のためのコードなど。ダイアログなどを管理。

save.php zip ファイル書き出し用。

### 5.2.1 オブジェクトの構造体

CObject クラスには block、start、goal、waypoint の配列がふくまれ、それぞれの要素は構造体 (あるいは連想配列) です。これを直接 json フォーマットにかきかえ、出力したのが temp.json にふくまれるコードです。それぞれのパラメタの意味は上述のオブジェクトの説明のものと同じですが、リンク関係のパラメタが直接ふくまれている点が異なります。

### 5.2.2 歩行者の構造体

歩行者では各エージェントのパラメタは構造体ではなくただの配列に記述されています。順に、x、y、goalID、nextWPID をさします。goalID と nextWPID はリンクにつかう値です。

### 5.2.3 リンクのしかた

リンクはリンク先の ID にあたるパラメタ (ゴールなら pairID、waypoint なら WPID)などをリンク元のパラメタに設定することでなされます。エージェントルールで対象エージェント集合から検索しているのはこれです。詳細は makeLink 関数を参照してください。

その他疑問な点やプログラムミスがありましたら、お気軽に富田 寛 <tomykaira@gmail.com>に連絡してください。問題点の報告については附録の「バグ・不具合について」もお読みください。

## 付録 A 著作権

プログラムの著作権は、富田 寛が制作した部分 (ソースコードに記載) をすべて GPL ライセンスとします。GPL ライセンスの詳細については付属の COPYING を参照してください。

## 付録 B バグ・不具合について

重大なバグが修正されましたらあたらしい版を公開することも予想されます。バグ報告のまえには <http://sites.google.com/site/tomykaira/> をチェックして最新版でふたたび試行してください。類似の不具合についてなにかコメントがあるかもしれないので、それも確認してください。

MapEditor や MapEditor が出力する model ファイルにおいて不具合をみつけた際は、問題修正のためぜひ富田 寛 <tomykaira@gmail.com> に連絡してください。タイトルには「MapEditor」をいれてください。問題が発生した状況とご利用の環境をできるだけ詳しく記述してください。問題が出力に関係したことでしたら、出力した zip ファイルも添付してください。ご協力よろしくおねがいします。機能やユーザエクスペリエンスに関する要望もお待ちしております。ソフトウェア改善のためにご協力よろしくおねがいします。