

ADVENTURE_Mates

Multi-Agent-based Traffic and Environment Simulator

Version 0.5b

プログラム使用マニュアル

November 2014

ADVENTURE Project

目次

1	はじめに	1
1.1	本プログラムの特徴	1
1.2	動作環境	1
1.3	コンパイルとインストール	1
1.4	実行方法	3
2	入力ファイル	5
2.1	概要	5
2.2	道路ネットワークに関する入力ファイル	5
2.2.1	mapPosition.txt	5
2.2.2	network.txt	6
2.2.3	speedLimit.txt	6
2.2.4	intersection/*****.txt	6
2.3	信号に関する入力ファイル	8
2.3.1	signals/default.txt	8
2.3.2	signals/defaultInter{n}.msa (n=2, 3, 4)	8
2.3.3	signals/*****.{msf, msa}	9
2.4	車両に関する入力ファイル	10
2.4.1	generateTable.txt	10
2.4.2	defaultGenerateTable.txt	10
2.4.3	vehicleRoutingParam.txt	11
2.4.4	vehicleFamily.txt	11
2.5	路側機に関する入力ファイル	11
2.5.1	detector.txt	11
2.5.2	genCounter.txt	12
2.5.3	init.txt	12
3	出力ファイル	17
3.1	概要	17
3.2	シミュレーション全般に関する出力ファイル	17
3.2.1	runInfo.txt	17
3.2.2	nodeShape.txt	17
3.2.3	linkShape.txt	18
3.3	信号に関する出力ファイル	19
3.3.1	signalCount.txt	19
3.3.2	timeline/signal/{000-999}/{000-999}/{0000-9900}.txt.gz	19

3.4	車両に関する出力ファイル	20
3.4.1	vehicleCount.txt	20
3.4.2	vehicleAttribute.txt	20
3.4.3	vehicleTrip.txt	20
3.4.4	timeline/vehicle/{000-999}/{000-999}/{0000-9900}.txt.gz	20
3.5	路側機による観測結果ファイル	21
3.5.1	inst/detD****.txt	21
3.5.2	inst/detS****.txt	21
3.5.3	inst/gen*****.txt	22
4	GUI 操作	23
4.1	メイン画面	23
4.2	View Control ウィンドウ	24
4.3	Visualizer Control ウィンドウ	25
4.4	Simulator Control ウィンドウ	26
4.5	Anget Generation ウィンドウ	27
4.6	Detail Information ウィンドウ	28
	参考文献	29

1 はじめに

本書は ADVENTURE Project[1] において開発中のマルチエージェント交通流シミュレータ ADVENTURE_Mates (略称 MATES [メイツ]) の使用マニュアルである。

本章ではまず、ADVENTURE_Mates の概要および実行までの操作手順を説明する。

1.1 本プログラムの特徴

ADVENTURE_Mates は知的マルチエージェントモデルにもとづく交通流シミュレータである。知的エージェントとしてモデル化された車両は道路環境中で他の車両エージェントと相互作用しながら自律的に意思決定(経路探索・加速度決定)する。複雑な交通現象は知的エージェント同士の相互作用の結果として創発するよう設計されている。

1.2 動作環境

本プログラムのコンパイルに必要なものは、GNU C コンパイラおよび C++ コンパイラ (gcc, g++)、CMake、make である。CMake がインストールされていない環境では、各ディレクトリに用意されている Makefile.alt を適宜編集して使用されたい。システムに zlib がインストールされていれば利用できる。可視化および GUI を伴うプログラム (advmates-sim) のコンパイルには OpenGL (GLX 拡張)、GLUT、GTK+2 の各開発用ファイルが必要である。

本プログラムは Linux にて動作確認を行っているが、多くの機能は Windows 上に構築された Cygwin (32bit) あるいは MinGW/MSYS でも実行可能である。Windows 上での可視化および GUI を伴うプログラム (advmates-sim) のコンパイルには OpenGL が必要である。

1.3 コンパイルとインストール

ADVENTURE_Mates をコンパイル、インストールするには以下に示す手順に従う。

(1) アーカイブファイルの展開

ダウンロードしたファイルのあるディレクトリで以下を実行する。

```
% tar xzf AdvMates-0.5b.tar.gz
```

ただし、“%” はコマンドプロンプトを表すため、実際には入力する必要はない。

アーカイブファイルの展開により、AdvMates-0.5b ディレクトリが作成される。AdvMates-0.5b ディレクトリは次のサブディレクトリを含んでいる。

- doc : ドキュメント
- examples : サンプルデータ
- include : ヘッダファイル群

- lib : ライブラリ群
- pre : プレ処理モジュールのソースファイル・ドキュメント等
- solver : メインモジュールのソースファイル

(2) Makefile の生成

AdvMates-0.5b ディレクトリにて以下を実行する .

```
% cmake -DINSTALL_PREFIX=install_dir .
```

ただし, MSYS 上でコンパイルする場合は以下を実行する

```
% cmake -G "MSYS Makefiles" -DINSTALL_PREFIX=install_dir .
```

“-DINSTALL_PREFIX=*install_dir*” はインストール先ディレクトリを *install_dir* に指定することを表す . 省略可能であり, 省略された場合には\$(HOME)/ADVENTURE/以下にインストールされる .

なお, 可視化・GUI ライブラリがインストールされていない環境で cmake に失敗する場合には以下を実行すると成功する場合がある .

```
% cmake -DPLATFORM=NOVIS .
```

(3) コンパイル

AdvMates-0.5b ディレクトリにて, 以下を実行する .

```
% make
```

(4) インストール

コンパイルに成功したら, 以下のコマンドによりインストールを行う .

```
% make install
```

インストール先ディレクトリに書き込み権限を持ったユーザによって行う必要がある . 以上の操作により, 以下のファイルが先に指定した *install_dir* (指定されていない場合は\$(HOME)/ADVENTURE/) 以下にインストールされる .

- bin/advmates-calc : 実行モジュール
- bin/advmates-sim : 可視化および GUI を伴う実行モジュール
- doc/AdvMates/AdvMates-manual-jp.pdf : 日本語マニュアル

なお、プレ処理モジュールは一括でコンパイル・インストールされない。pre 以下の各ディレクトリに含まれるドキュメントを参照のこと。

1.4 実行方法

ADVENTURE_Mates には `advmates-calc` と `advmates-sim` という 2 種類の実行モジュールが含まれる。

(1) `advmates-calc`

画像による出力なしのシミュレーションを実行する。デフォルトで 1 ステップごとの車両状態をファイル出力する。1 時間程度のシミュレーションの時系列データを出力すると 1GB を超えることがあるので、ディスク容量には注意する必要がある。

`advmates-calc` は以下のコマンドで実行する。

```
% advmates-calc [options]
```

実行時には表 1 に示すオプションを指定できる。

(2) `advmates-sim`

画像による出力および GUI 付きのシミュレーションを実行する。GUI から指定された場合のみ出力ファイルを書き込む。GUI に関しては後述する。

`advmates-sim` は以下のコマンドで実行する。

```
% advmates-sim [options]
```

実行時には表 2 に示すオプションを指定できる。

表1 advmates-calc の実行時オプション

オプション	説明
-d <i>DataDir</i>	入出力ディレクトリのルートパスを <i>DataDir</i> に指定する．指定しない場合はカレントディレクトリをルートパスとする．
-r <i>Number</i>	乱数の種を <i>Number</i> に指定する．デフォルトでは時刻をもとに与えられる．
-q あるいは --no-verbose	詳細な画面表示を省略する．
--no-input-signal	signals ディレクトリ以下の信号入力ファイルを読み込まず，すべての信号が青であるとみなす．
--no-generate-random-vehicle	入力ファイル指定された車両以外を生成しない．
-t <i>MaxTime</i>	計算対象となるシミュレーション内の時間を <i>MaxTime</i> [msec] に指定する．指定しない場合は 3,600,000[msec]=1 時間である．グローバル変数が指定されていればそれを優先する．
-s あるいは --no-output-timeline	時系列データを出力しない．
-l あるいは --no-output-tripinfo	車両のトリップ情報を出力しない．
-m	車両感知器による計測データを出力しない．
--no-output-monitor-d	車両感知器による計測データのうち，詳細出力 (detD****.txt) を出力しない．
--no-output-monitor-s	車両感知器による計測データのうち，統計出力 (detS****.txt) を出力しない．
-g あるいは --no-output-generate	車両発生カウンタが計測するデータを出力しない．

表2 advmates-sim の実行時オプション

オプション	説明
-d <i>DataDir</i>	入出力ディレクトリのルートパスを <i>DataDir</i> に指定する．指定しない場合はカレントディレクトリをルートパスとする．
-r <i>Number</i>	乱数の種を <i>Number</i> に指定する．デフォルトでは時刻をもとに与えられる．
-q あるいは --no-verbose	詳細な画面表示を省略する．
--no-input-signal	signals ディレクトリ以下の信号入力ファイルを読み込まず，すべての信号が青であるとみなす．
--no-generate-random-vehicle	入力ファイル指定された車両以外を生成しない．

2 入力ファイル

本章では、ADVENTURE_Mates の入力ファイルを説明する。

2.1 概要

ADVENTURE_Mates を実行する際の入力ファイルは以下である。入力ファイルの場所は実行時に“-d” オプションで指定する。指定しない場合はカレントディレクトリを対象とする。†印は必須入力ファイルである。signals 以下を除く入力ファイルでは空白は無視され、“#”で始まる行はコメントとして読み飛ばされる。

- mapPosition.txt †
- network.txt †
- speedLimit.txt
- intersection/*****.txt
- signals/default.msf †
- signals/defaultInter{n}.msa (n=2, 3, 4) †
- signals/*****.{msf,msa}
- generateTable.txt
- defaultGenerateTable.txt
- vehicleRoutingParam.txt
- vehicleFamily.txt
- detector.txt
- genCounter.txt
- init.txt

2.2 道路ネットワークに関する入力ファイル

2.2.1 mapPosition.txt

必須の入力ファイルである。

交差点の座標を記述する。左から交差点 ID, x 座標 [m], y 座標 [m], z 座標 [m] である。z 座標はオプションであり、指定されていない場合は z=0 [m] とみなされる。

```
mapPosition.txt
0, 0, 0, 0
1, -100, 0, 0
2, 0, -100, 0
3, 100, 0, 0
4, 0, 100, 0
```


2.2.2 network.txt

必須の入力ファイルである。

交差点の流入・流出車線数と隣接情報を記述する。左から交差点 ID, 流入・流出車線数, 隣接交差点の ID (反時計回り) である。交差点 ID は 6 桁以下の自然数 (1 ~ 999999) である。

```
network.txt
0, 22322232, 1, 2, 3, 4
1, 22, 0
2, 22, 0
3, 22, 0
4, 22, 0
```

流入・流出車線数は、交差点の各境界における流入車線数, 流出車線数の順で記述する。1 つの境界につき 2 桁が必要なので, n 叉路の形状は $2n$ 桁になる (十字路なら $2*4=8$ 桁)。右折専用車線は向かい合う境界の流入車線数と流出車線数をみて自動的に作成される。流入車線数や流出車線数を 0 に指定することで, 一方通行を表現することもできる。

交差点を表す多角形の辺のうち, 単路部が接する辺を境界と呼ぶこととする。境界の順序は network.txt で指定した隣接交差点 ID の順に等しい。例えば交差点 ID:0 の接続交差点を順に 1, 2, 3, 4 と記述した場合, 交差点 ID:0 の境界 0 とは「交差点 ID:1 に接続する単路が接する境界」, 交差点 ID:0 の境界 3 とは「交差点 ID:4 に接続する単路部が接する境界」を指す。

2.2.3 speedLimit.txt

規制速度を記述する。

```
speedLimit.txt
0, 1, 30
1, 0, 40
```

左から, 始点交差点 ID, 終点交差点 ID, 規制速度 [km/h] である。始点と終点の指定順によって, 上りと下りで異なった規制速度を与えられる。

2.2.4 intersection/*****.txt

任意の交差点の属性を指定する。境界相対方向, レーン接続, 交差点頂点座標を指定できる。指定したい交差点と同じ名前のファイル名をつける。交差点 ID:5 の属性指定ファイルは”intersection/000005.txt”となる。GUI から交差点情報を出力すると, このファイルで使用する情報を取得できる。

境界相対方向, レーン接続, 交差点頂点座標の順に並べて指定する。これら 3 つのうち, 指定する必要のないものを全てコメントアウト, もしくは最初から書かないことで, いずれか 1 つのみを指定することもできる。途中のデータをコメントアウトする事も可能である。

(1) 境界方向指定

行列形式で入力する．行列の (i,j) 成分は，境界:i から流入して境界:j に流出する場合の相対方向である．

```

intersection/*****.txt (境界相対方向)
t, r, s, l
l, t, r, s
s, l, t, r
r, s, l, t

```

t, s, r, l はそれぞれ転回，直進，右折，左折を意味する．対角項は必ず t とする．

(2) レーン接続

各レーンについて，始点コネクタ ID，終点コネクタ ID の 2 つを指定する．コネクタ ID は 4 桁であり，各桁は以下を意味する．

- 1 桁目 (0) : 現在のバージョンでは 0 で固定
- 2 桁目 (0, 1, ...) : コネクタが属する境界 ID (十字路ならば 0~3)
- 3, 4 桁目 (00, 01, ..., 99) : 単路部から交差点へ進入する向きに左から 00, 01, ...

```

intersection/*****.txt (レーン接続)
# 境界 ID: 0 からのレーン
0000, 0304
0000, 0203
0001, 0202
# 境界 ID: 1 からのレーン
0100, 0304
0101, 0303
# 境界 ID: 2 からのレーン
0200, 0104
0200, 0003
0201, 0002
# 境界 ID: 3 からのレーン
0300, 0203
0300, 0104
0301, 0103
0302, 0002

```

(3) 交差点頂点座標

必ず “vertex” と書いた行の下から定義を始める．交差点の形状を多角形で定義し，その頂点の x, y, z 座標 (単位は [m]) を与える．1 行が 1 つの頂点を表す．n 叉路の交差点は必ず 2n 角形でなければならない．座標は交差点の代表点 (mapPosition.txt で指定した交差点の座標) を基準としたずれとして与える．また頂点の記述の順にも注意する．交差点に流入する方向から

見て、境界 ID:0 の左側の点をスタートして左回りで記述する必要がある。

```

intersection/*****.txt (交差点頂点座標)
vertex
0, 15, 0
-15, 15, 0
-15, -15, 0
0, -15, 0
5, -13, 0
15, -3, 0
15, 3, 0
5, 13, 0

```

2.3 信号に関する入力ファイル

2.3.1 signals/default.txt

必須の入力ファイルである。

デフォルトの信号サイクル長とスプリットを記述する。サイクル長とは信号現示が1巡する時間であり、スプリットとは各現示に割り当てられた時間である。

```

signal/default.msf
0 86400000 140000 45000 10000 10000 5000 45000 10000 10000 5000

```

左から開始時刻、終了時刻、サイクル長、各現示パターンのスプリットを表す(単位は全て[msec])。サイクル長は全現示パターンのスプリットの総和であるため、3列目=4列目以降の和となる(上記では $140000=45000+10000+10000+5000+45000+10000+10000+5000$ である)。あるタイミングでサイクル長やスプリットを変更したい場合には、開始時刻、終了時刻を調整し複数行に分けて記述することもできる。

2.3.2 signals/defaultInter{n}.msa (n=2, 3, 4)

デフォルトの信号パターンを記述する。n は交差点の次数(隣接交差点の数)である。T字路は n=3、十字路は n=4 であり、それぞれ defaultInter3.msa、defaultInter4.msa が適用される。

```

signal/defaultInter4.msa
8
1 0 2 2 0 1 1 0 2 2 0 1
3 0 2 2 0 2 3 0 2 2 0 2
2 3 2 2 0 2 2 3 2 2 0 2
2 0 2 2 0 2 2 0 2 2 0 2
2 0 1 1 0 2 2 0 1 1 0 2
2 0 2 3 0 2 2 0 2 3 0 2
2 0 2 2 3 2 2 0 2 2 3 2
2 0 2 2 0 2 2 0 2 2 0 2

```

1 行目には信号パターンの総数を記述する。2 行目以降、左から { 境界 ID:0 の信号のメイン現示, サブ現示, 歩行者用現示 }, { 境界 ID:1 の信号のメイン現示, サブ現示, 歩行者用現示 }, { 境界 ID:2 の... }, ... を表す。ただし, 現バージョンでは歩行者用現示はシミュレーション中で使用されない。

メイン現示の数字と現示の対応は以下である。

- 1 : 青
- 2 : 赤
- 3 : 黄
- 4 : 赤点滅 (車両は交差点進入前に一旦停止する)
- 5 : 黄点滅 (車両は徐行しながら交差点に進入する)

サブ現示 (右折可の矢印等) の数字と現示の対応は以下である。左折と直進可の矢印を出す場合は“12”を指定する。

- 0 : 矢印なし
- 1 : 直進
- 2 : 左折
- 3 : 右折

先述の defaultInter4.msa の解釈すると以下のようになる。

```

signal/defaultInter4.msa
8
1 0 2 2 0 1 1 0 2 2 0 1 ...境界 ID:0, 2 が青, 境界 ID:1, 3 が赤
3 0 2 2 0 2 3 0 2 2 0 2 ...境界 ID:0, 2 が黄に
2 3 2 2 0 2 2 3 2 2 0 2 ...境界 ID:0, 2 が赤・右折矢印に
2 0 2 2 0 2 2 0 2 2 0 2 ...全境界が赤に
2 0 1 1 0 2 2 0 1 1 0 2 ...境界 ID:1, 3 が青に
2 0 2 3 0 2 2 0 2 3 0 2 ...境界 ID:1, 3 が黄に
2 0 2 2 3 2 2 0 2 2 3 2 ...境界 ID:1, 3 が赤・右折矢印に
2 0 2 2 0 2 2 0 2 2 0 2 ...全境界が赤 これで 1 サイクルの定義が終了

```

2.3.3 signals/*****.{msf, msa}

signals/{ 交差点 ID }.msf および signals/{ 交差点 ID }.msa を用意することで, 信号パラメータを個別に設定することができる。交差点 ID:5 の信号パラメータファイルは“signals/000005.{msf, msa}”となる。ファイルの書式は signals/default.msف および signals/defaultInter{n}.msa (n=2, 3, 4) の項を参照のこと。

2.4 車両に関する入力ファイル

2.4.1 generateTable.txt

自動車の発生する時間，発生地，目的地，発生交通量，車種 ID，経由地を記述する．

```
generateTable.txt
0, 86400000, 1, 3, 1000, 20, 0
0, 86400000, 2, 4, 600, 50, 3, 0, 6, 5
```

左から，開始時刻 [msec]，終了時刻 [msec]，発生交差点 ID，目的交差点 ID，発生交通量 [台/時]，車種 ID，経由交差点数，経由交差点 ID1，経由交差点 ID2...を表す．発生交差点および目的交差点は次数が 1 の交差点 (ネットワークの端点) でなければならない．発生交通量は整数値で記述する．経由交差点 ID は指定された経由交差点の数だけ指定する．

車種 ID は 2 桁の整数で 現在のところ 2*は普通車 5*は大型車を指し ,その他はすべて普通車のサイズおよび車両特性となる．ここで指定した車種 ID はそのまま result/vehicleAttribute.txt に出力される．

注意

車両発生には乱数を用い，発生のタイミングを厳密にスケジューリングしているわけではない．指定した交通量と一致することを保証しない (500 台と指定しても 520 台程度発生することがある)．

2.4.2 defaultGenerateTable.txt

自動車の発生する時間，発生地，発生交通量，車種 ID，経由地を記述する．

```
defaultGenerateTable.txt
0, 86400000, 1, *****, 1200, 20, 0
0, 86400000, 2, *****, 300, 50, 0
```

書式は上記の generateTable.txt と同一であるが，目的地交差点のカラムは “*****” でなければならない．generateTable は目的地指定，defaultGenerateTable は目的地ランダムという違いがある．

注意

generateTable.txt および defaultGenerateTable.txt で発生交通量を指定されていない交差点からは車両が発生しないわけではない．交差点の流出車線数に応じた割合でランダムに車両が発生する．車両を発生させたくない交差点がある場合には発生交通量 0 を明示的に指定するか，実行時に `--no-generate-random-vehicle` オプションを指定する必要がある．

2.4.3 vehicleRoutingParam.txt

車両の経路選択パラメータを設定する。1行が1つのパラメータセットを表し、経路の効用（効用が最小となる経路を選択する）を算出する際の各因子に乘じる重みを指定する。

defaultGenerateTable.txt

```
2, 1, 0, 0, 0, 0
1, 0, 0, 0, 0, 1
```

左から距離，時間，直進回数，左折回数，右折回数，道路幅（の逆数）に対する重み。例えば距離に関するコストが大きければ最短距離を選択する傾向が強くなり，道路幅（の逆数）に関するコストが大きければ広い道路を選択する傾向が強くなる。

このファイルが見つからない場合は，デフォルトのパラメータセット

```
{1,0,0,0,0,0}: (距離を重んじる)
{0,1,0,0,0,0}: (時間を重んじる)
{1,1,0,0,0,0}: (距離と時間を同程度に重んじる)
```

が利用される。各パラメータセットの選択確率は同等である。

2.4.4 vehicleFamily.txt

車種ごとの属性を指定する。

vehicleFamily.txt

```
21, 4.400, 1.830, 1.315, 0, 3.0, -5.0, 1, 0, 0.5
22, 4.400, 1.830, 1.315, 0, 3.0, -5.0, 1, 0.5, 0
51, 8.465, 2.230, 3.420, 0, 3.0, -5.0, 0, 0.1, 1
```

左から，車種 ID，車長 [m]，車幅 [m]，車高 [m]，車重（未使用），最大加速度 [m/s²]，最大減速度 [m/s²]，車体色の赤成分 (0~1)，車体色の緑成分 (0~1)，車体色の青成分 (0~1) を表す。

2.5 路側機に関する入力ファイル

2.5.1 detector.txt

車両感知器を設置する場所を記述する。

detector.txt

```
1, 1, 0, 10, 600000
2, 0, 1, -0.001, 300000
```

左から，感知器 ID，始点交差点 ID，終点交差点 ID，交差点からの距離 [m]，集計間隔 [msec] を表す。距離は非負であれば始点交差点からの距離，負であれば終点交差点からの距離を表す。時間間隔とは累積交通量を出力する間隔。300000[msec] なら 5 分おきの通過交通量を出力する。

2.5.2 genCounter.txt

車両発生カウンタを設置する交差点 ID を記述している。

genCounter.txt	
1	
3	
4	

車両の発生を計測するので、端点の交差点の ID を指定しなければならない。

2.5.3 init.txt

グローバル変数やフラグを“変数=値”の書式で指定する。表 3 から表 7 まで、変数名と初期値を示す。コマンドライン引数での指定、init.txt での指定、初期値の順で優先される。

表 3 グローバル変数名と初期値 (フラグ)

変数名	説明	初期値
FLAG_VERBOSE	詳細情報を出力するか	true
FLAG_INPUT_SIGNAL	信号情報を入力するか	true
FLAG_INPUT_VEHICLE	車両情報を入力するか	true
FLAG_GEN_RAND_VEHICLE	車両発生情報が定義されていない交差点から車両を発生させるか	true
FLAG_OUTPUT_TIMELINE	時系列データを出力するか	true (advmates-calc) false (advmates-sim)
FLAG_OUTPUT_MONITOR_D	計測機器の詳細データを出力するか	true (advmates-calc) false (advmates-sim)
FLAG_OUTPUT_MONITOR_S	計測機器の統計データを出力するか	true (advmates-calc) false (advmates-sim)
FLAG_OUTPUT_GEN_COUNTER	車両発生データを出力するか	true (advmates-calc) false (advmates-sim)
FLAG_OUTPUT_TRIP_INFO	車両のトリップ情報を出力するか	true (advmates-calc) false (advmates-sim)

表 4 グローバル変数名と初期値 (入カファイル)

変数名	説明	初期値
GV_INIT_FILE	グローバル変数設定ファイル	<i>data_dir/init.txt</i>
MAP_POSITION_FILE	交差点座標設定ファイル	<i>data_dir/mapPosition.txt</i>
MAP_NETWORK_FILE	隣接交差点設定ファイル	<i>data_dir/network.txt</i>
SPEED_LIMIT_FILE	制限速度設定ファイル	<i>data_dir/speedLimit.txt</i>
INTERSECTION_ATTRIBUTE_DIRECTORY	交差点属性ファイル格納ディレクトリ	<i>data_dir/intersection/</i>
GENERATE_TABLE	車両発生定義ファイル	<i>data_dir/generateTable.txt</i>
DEFAULT_GENERATE_TABLE	車両発生定義ファイル (目的地ランダム)	<i>data_dir/defaultGenerateTable.txt</i>
VEHICLE_FAMILY_FILE	車種の属性定義ファイル	<i>data_dir/vehicleFamily.txt</i>
VEHICLE_ROUTE_PARAM_FILE	経路選択パラメータ設定ファイル	<i>data_dir/vehicleRoutingParam.txt</i>
DETECTOR_FILE	車両感知器定義ファイル	<i>data_dir/detector.txt</i>
GENCOUNTER_FILE	車両発生カウンタ定義ファイル	<i>data_dir/genCounter.txt</i>
SIGNAL_CONTROL_DIRECTORY	信号パラメータ格納ディレクトリ	<i>data_dir/signals/</i>
SIGNAL_CONTROL_FILE_DEFAULT	デフォルトの信号現示時間定義ファイル (拡張子除く)	default
SIGNAL_ASPECT_FILE_DEFAULT_PREFIX	デフォルトの信号現示パターン定義ファイルの接頭辞	defaultInter
CONTROL_FILE_EXTENSION	信号現示時間定義ファイルの拡張子	.msf
ASPECT_FILE_EXTENSION	信号現示パターン定義ファイルの拡張子	.msa

data_dir は実行時に “-d” オプションで指定した場所 (指定しない場合はカレントディレクトリ) を表す。

表 5 グローバル変数名と初期値 (出力ファイル)

変数名	説明	初期値
RESULT_TIMELINE_DIRECTORY	時系列データ出力ディレクトリ	<i>result_dir/timeline</i>
RESULT_IMG_DIRECTORY	画像出力ディレクトリ	<i>result_dir/img</i>
RESULT_INSTRUMENT_DIRECTORY	路側機による観測結果出力ディレクトリ	<i>result_dir/inst</i>
RESULT_DETECTOR_PREFIX	感知器詳細出力データの接頭辞	detD
RESULT_DETECTORS_PREFIX	感知器統計出力データの接頭辞	detS
RESULT_GENCOUNTER_PREFIX	車両発生カウンタ出力データの接頭辞	gen
RESULT_RUN_INFO_FILE	シミュレーション実行情報出力ファイル	<i>result_dir/runInfo.txt</i>
RESULT_RUN_INFO_FILE	シミュレーション実行情報出力ファイル	<i>result_dir/runInfo.txt</i>
RESULT_NODE_SHAPE_FILE	交差点形状出力ファイル	<i>result_dir/nodeShape.txt</i>
RESULT_LINK_SHAPE_FILE	単路部形状出力ファイル	<i>result_dir/linkShape.txt</i>
RESULT_SIGNAL_COUNT_FILE	信号総数出力ファイル	<i>result_dir/signalCount.txt</i>
VEHICLE_ATTRIBUTE_FILE	車両属性データ出力ファイル	<i>result_dir/vehicleAttribute.txt</i>
VEHICLE_TRIPFILE	車両トリップ情報出力データ	<i>result_dir/vehicleTrip.txt</i>
VEHICLE_COUNT_FILE	各時刻における車両総数出力データ	<i>result_dir/vehicleCount.txt</i>

result_dir は *data_dir* 直下の *result* ディレクトリを指す。

表 6 グローバル変数名と初期値 (道路に関する定数)

変数名	説明	初期値
RIGHT_TURN_LANE_LENGTH	右折専用レーンの標準長さ [m]	30
DEFAULT_LANE_WIDTH	標準のレーン幅 [m]	3.5
SPEED_LIMIT_SECTION	標準的な単路部での制限速度 [km/h]	60
SPEED_LIMIT_INTERSECTION	標準的な交差点での制限速度 [km/h]	60
VELOCITY_CRAWL	徐行速度 [km/h]	10
VELOCITY_AT_TURNING_RIGHT	右折時制限速度 [km/h]	40
VELOCITY_AT_TURNING_LEFT	左折時制限速度 [km/h]	40
GENERATE_VELOCITY_LIMIT	車両発生時の制限速度 [km/h], 負なら制限なし	-1
MIN_HEADWAY_AT_TURNING	右左折時の最小ヘッドウェイ [sec]	1.7
DEFAULT_TRAFFIC_VOLUME_WIDE	3車線以上の車両発生点における発生交通量のデフォルト値 [台/h]	660
DEFAULT_TRAFFIC_VOLUME_NORMAL	2車線の車両発生点における発生交通量のデフォルト値 [台/h]	440
DEFAULT_TRAFFIC_VOLUME_NARROW	1車線の車両発生点における発生交通量のデフォルト値 [台/h]	125

表7 グローバル変数名と初期値 (車両に関する定数)

変数名	説明	初期値
REACTION_TIME_VEHICLE	加速度制御における緩和時間 [sec]	0.74
GAP_ACCEPTANCE_VEHICLE_CROSS	交錯車両とのギャップアクセプトランス [sec]	3.0
MAX_ACCELERATION_PASSENGER	普通車の最大加速度 [m/sec ²]	3.0
MAX_DECELERATION_PASSENGER	普通車の最大減速度 [m/sec ²]	-5.0
MAX_ACCELERATION_TRUCK	大型車の最大加速度 [m/sec ²]	3.0
MAX_DECELERATION_TRUCK	大型車の最大減速度 [m/sec ²]	-5.0
ERROR_VELOCITY	車線変更時に与える横向きの速度 [km/h]	7.5
NO_OUTPUT_LENGTH_FROM_ORIGIN_NODE	発生端点から L[m] 以内の車両は出力しない	0
VEHICLE_LENGTH_PASSENGER	普通車の車長 [m]	4.400
VEHICLE_WIDTH_PASSENGER	普通車の車幅 [m]	1.830
VEHICLE_HEIGHT_PASSENGER	普通車の車高 [m]	1.315
VEHICLE_LENGTH_TRUCK	大型車の車長 [m]	8.465
VEHICLE_WIDTH_TRUCK	大型車の車幅 [m]	2.230
VEHICLE_HEIGHT_TRUCK	大型車の車高 [m]	3.420
VEHICLE_VELOCITY_HISTORY_RECORD	速度履歴を保存するか	true
VEHICLE_VELOCITY_HISTORY_SIZE	速度履歴を保存するステップ数	180
VEHICLE_VELOCITY_HISTORY_INTERVAL	速度履歴を保存するステップ間隔	10

3 出力ファイル

本章では、ADVENTURE_Mates の出力ファイルを説明する。

3.1 概要

出力ファイルは、実行時に“-d” オプションで指定したディレクトリ (指定しない場合はカレントディレクトリ) 直下の result ディレクトリに格納される。result/img, result/inst, result/timeline の3つのサブディレクトリをファイル出力前に作成しておく必要がある。result/timeline 以下のディレクトリは自動で作成される。result/img がない状態で advmates-sim の WithCapturing にチェックを入れて動かすと異常終了するので注意すること。

ADVENTURE_Mates を実行して得られる出力ファイルは以下である。

- runInfo.txt
- nodeShape.txt
- linkShape.txt
- signalCount.txt
- timeline/signal/{000-999}/{000-999}/{0000-9900}.txt.gz
- vehicleCount.txt
- vehicleAttribute.txt
- vehicleTrip.txt
- timeline/vehicle/{000-999}/{000-999}/{0000-9900}.txt.gz
- inst/detD****.txt
- inst/detS****.txt
- inst/gen*****.txt

3.2 シミュレーション全般に関する出力ファイル

3.2.1 runInfo.txt

シミュレーション実行情報を出力する。

```
runInfo.txt
1000
100
```

1 行目は総ステップ数, 2 行目は 1 ステップあたりの時間 [msec] を表す。上記の例であれば, $1000[\text{step}] * 100[\text{msec}/\text{step}] = 100000[\text{msec}] = 100[\text{sec}]$ の計算であることを表す。

3.2.2 nodeShape.txt

交差点形状を出力する。以下に書式を示す。

nodeShape.txt の書式

```

交差点総数
# 以下交差点について繰り返し
# {
    交差点 ID
    代表点 x 座標, 代表点 y 座標, 代表点 z 座標
    頂点数
    # 以下各頂点について繰り返し
    # {
        頂点 x 座標, 頂点 y 座標, 頂点 z 座標
    # }
# }

```

なお、実際には“#”で始まる行は出力されず、インデントも行われない。

3.2.3 linkShape.txt

単路部形状を出力する。以下に書式を示す。

linkShape.txt の書式

```

単路部総数
# 以下単路部について繰り返し
# {
    単路部 ID
    右レーン数, 左レーン数
    始点交差点 ID, 終点交差点 ID
    始点側端点の信号の有 (=1) 無 (=0), 終点側端点の信号の有 (=1) 無 (=0)
    単路部の中心点個数 (現状では 2 に固定)
    # 以下各中心点を始点交差点から順に
    # {
        x 座標, y 座標, z 座標
    # }
    単路部の右境界点個数 (現状では 2 に固定)
    # 以下各右境界点を始点交差点から順に
    # {
        x 座標, y 座標, z 座標
    # }
    単路部の左境界点個数 (現状では 2 に固定)
    # 以下各左境界点を始点交差点から順に
    # {
        x 座標, y 座標, z 座標
    # }
# }

```

なお、実際には“#”で始まる行は出力されず、インデントも行われない。

3.3 信号に関する出力ファイル

3.3.1 signalCount.txt

信号総数を出力する。

```

signalCount.txt
5
4

```

1 行目は交差点総数, 2 行目は信号機の総数である。n 叉路の交差点には n 個の信号機が設置されているとして総数をカウントする。

3.3.2 timeline/signal/{000-999}/{000-999}/{0000-9900}.txt.gz

各時刻ステップにおける信号の状態を出力する。ディレクトリ名とファイル名に含まれる数値を連結した 10 桁の数値が時刻 [msec] を表す。zlib がインストールされていない環境では拡張子.gz は省略される。

```

timeline/signal/{000-999}/{000-999}/{0000-9900}.txt.gz
00000000000001, 32
00000000000002, 8
00000000000003, 32
00000000000004, 8

```

左から、信号機 ID、現示の状態 (10 進数) を表す。

このファイルにおける信号機 ID は交差点 ID ではなく、“(dst 交差点 ID)+(src 交差点 ID)” で与えられる。例えば交差点 ID:000011 から交差点 ID:000022 へ向かう車両が見るのは信号機 ID:000022000011 である。

信号機状態は、青、黄、赤、左折矢印 (←)、直進矢印 (↑)、右折矢印 (→) の 6 種について以下のように 2 進数で扱う。出力時に状態を足し合わせて 10 進数に変換する。

```

100000 : 青
010000 : 黄
001000 : 赤
000100 : 左折矢印 (←)
000010 : 直進矢印 (↑)
000001 : 右折矢印 (→)

```

例えば、青信号なら 32、赤信号で右折矢印が出ていれば 8+1=9 となる。

3.4 車両に関する出力ファイル

3.4.1 vehicleCount.txt

各時刻ステップにおける車両総数を出力する。

```
vehicleAttribute.txt
184400, 8, 8
184500, 9, 9
184600, 9, 9
184700, 9, 9
184800, 9, 9
184900, 10, 10
```

左から、時刻 [msec], 車両総数, 走行中車両総数を表す。車両が発生しても発生点が渋滞しているなどして、すぐにシミュレーション上に登場できず発生点で待機状態となる場合がある。車両総数には待機車両が含まれ、走行車両総数には待機車両が含まれない。

3.4.2 vehicleAttribute.txt

各車両の属性データを出力する。

```
vehicleAttribute.txt
000000, 20, 4.4, 1.83, 1.315
000001, 51, 4.4, 1.83, 1.315
```

左から、車両 ID, 車種 ID, 車長 [m], 車幅 [m], 車高 [m] を表す。車種 ID については入力ファイルの vehicleFamily.txt および generateTable.txt を参照のこと。

3.4.3 vehicleTrip.txt

各車両のトリップ情報を出力する。

```
vehicleTrip.txt
000001, 195, 11700
000002, 195.173, 12800
000006, 180.353, 12800
```

左から、車両 ID, 走行距離 [m], 旅行時間 [msec] を表す。目的地に到着した車両は目的地までの走行距離および旅行時間を、シミュレーション終了時に目的地に到着できていない車両は、シミュレーション終了時までの走行距離および旅行時間を出力する。

3.4.4 timeline/vehicle/{000-999}/{000-999}/{0000-9900}.txt.gz

各時刻ステップにおける各車両の走行状態を出力する。ディレクトリ名とファイル名に含まれる数値を連結した 10 桁の数値が時刻 [msec] を表す。zlib がインストールされていない環境では拡張子.gz は省略される。

— timeline/vehicle/{000-999}/{000-999}/{0000-9900}.txt.gz

```
000568,20,-65.4477,-1.75,0,90,0,15.2169,0.61694,NULL,000000000001,
000569,21,1.46923,-8.53077,0,315,0,6.17724,0.696618,000000,NULL,
000577,22,-15.3472,5.25,0,270,0,0,0.0209177,NULL,000000000001,
```

左から, 車両 ID, 車種 ID, x 座標 [m], y 座標 [m], z 座標 [m], XY 回転角, Z 回転角, 速度 [m/s], 加速度 [m/(s²)], 交差点 ID, 単路部 ID を表す. 単路部を走行中は交差点 ID が NULL, 交差点を走行中は単路部 ID が NULL となる. XY 回転角は速度ベクトルを XY 平面上に投影した時に y 軸正の方向となす角 (0° ~ 360°: 左回り正) で表す. Z 回転角で XY 平面となす角 (-90° ~ 90°: 上向き正) で表す.

3.5 路側機による観測結果ファイル

3.5.1 inst/detD****.txt

****には実際には感知器の ID が入る. 交通量の詳細データとして, 観測点を通過した個々の車両の情報を出力する.

— inst/detD****.txt

```
#section:000000000001, isUP:0, length:-0.002, detectors:2
#time, lane, vehicle_id, vehicle_type, origin, destination, velocity
000017400,01020001,000006,22,000001,000004,0.0166667
000054400,01020001,000019,22,000001,000004,0.0166667
000176800,01030000,000051,22,000001,000004,0.0155114
```

1 行目の “isUp:0” は, 単路部 ID:000000000001 の下り方向を観測していることを表す (上り方向であれば “isUp:1”). “length:-0.002” は単路の下流端から 0.002[m] だけ上流の地点を計測していることを表す. “detectors:2” とは観測位置を横切るレーンの数を表す.

3 行目以降は, 左から通過時刻 [msec], レーン ID, 車両 ID, 出発地交差点 ID, 目的地交差点 ID, 速度 [m/msec] を表す.

3.5.2 inst/detS****.txt

****には実際には感知器の ID が入る. 交通量の統計データとして, 指定された時間間隔ごとに通過車両台数を出力する.

— inst/detS****.txt

```
#section:000000000002, isUP:1, length:20
#interval:60000, detectors:2
#lane0:00039000
#lane1:00040100
#begintime, endtime, total(conv), total(simple), total_p,total_t
#volume_p, volume_t, volume0p, volume0t, volume1p, volume1t, ...
000000100, 000060000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
000060100, 000120000, 4, 4, 4, 0, 4, 0, 3, 0, 1, 0
000120100, 000180000, 5, 5, 5, 0, 1, 0, 0, 0, 1, 0
```


書式は以下に示す。(1), (2), ... は第 1 カラム, 第 2 カラム, ... を指す。

- (1): 計測時間帯の開始時刻
 - (2): 計測時間帯の終了時刻
 - (3): 解析開始からの総交通量 (換算)=(5)+(6)*1.7
 - (4): 解析開始からの総交通量 (台数)=(5)+(6)
 - (5): 解析開始からの小型車の総交通量 (台数)=(7) を縦に集計したもの
 - (6): 解析開始からの大型車の総交通量 (台数)=(8) を縦に集計したもの
 - (7): 計測時間帯 ((1) から (2) まで) の小型車の交通量 (台数)=(9), (11), ... の和
 - (8): 計測時間帯 ((1) から (2) まで) の大型車の交通量 (台数)=(10), (12), ... の和
 - (9): 観測リンクの 0 番目のレーン (lane0) を計測時間帯に通過した小型車の交通量 (台数)
 - (10): 観測リンクの 0 番目のレーン (lane0) を計測時間帯に通過した大型車の交通量 (台数)
- 以降, 奇数カラムに lane1, lane2, ... の小型車交通量 (台数),
偶数カラムに lane1, lane2, ... の大型車交通量 (台数)

3.5.3 inst/gen*****.txt

*****には実際には交差点 ID が入る。genCounter.txt によって指定された交差点から発生した車両の情報を出力する。

```

inst/gen*****.txt
#incident section:000000000001, lanes:2
#inflow_time, time-headway, generation_time, generation_time_interval,
#lane, vehicle_id, vehicle_type, origin, destination
000019500,19500,000019500,19500,01020001,000005,22,000001,000004
000127000,107500,000127000,107500,01020001,000038,22,000001,000004
000231800,104800,000231800,104800,01030000,000077,22,000001,000004

```

左から, 流入時刻 [msec], 前の車両流入との時間間隔 [msec], 発生時刻 [msec], 前の車両発生との時間間隔 [msec], レーン ID, 車両 ID, 車種 ID, 出発地交差点 ID, 目的地交差点 ID を表す。

注意

車両の発生と流入は別時刻となることがある。発生時刻は乱数によって決定されるが, 前の車両流入との間隔が短い場合には単路部に十分な空間がなく, 新しく発生した車両を単路部に流入させることができないためである。このとき, 発生した車両は流入可能となるまで発生交差点において待機する。

4 GUI 操作

本章では、ADVENTURE_Mates の GUI 操作について説明する。

4.1 メイン画面

advmates-sim を実行すると、図 1 のような画面が表示される。灰色が車道、車道上の白線がレーン (車線) である。長方形が車両を表す。車体の表示色は vehicleFamily.txt で指定されたものであり、指定がなければ赤色が普通車、水色が大型車である。交差点と単路部の境界にある正方形が信号現示を表す。

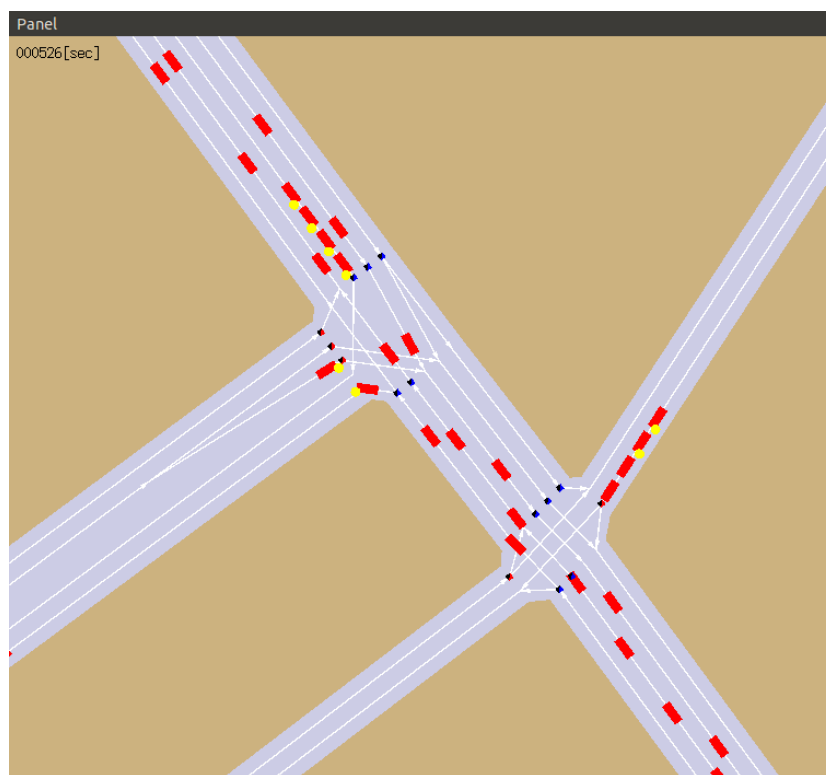


図 1 View ウィンドウ

注意

advmates-sim の終了はコンソールに Ctrl-c を入力するか、サブウィンドウの “Quit” ボタンを押す。

サブウィンドウの開閉は必ず View ウィンドウの “Panel” から行う。サブウィンドウ上部の “× (閉じる)” ボタンは使用しない。

4.2 View Control ウィンドウ

View ウィンドウの“Panel” から“View Control”を指定すると、図2のような View Control ウィンドウが表示される。ここで画面移動や拡大縮小を実行する。操作方法を表8に示す。

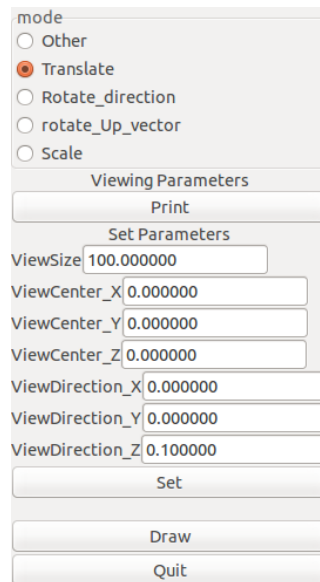


図2 View Control ウィンドウ

表8 View Control ウィンドウの操作

カテゴリ	項目	意味
mode	Other	(現在は使用しない)
	Translate	マウスドラッグで画面を平行移動する
	Rotate_direction	マウスドラッグで画面を回転する
	rotate_Up_vector	マウスドラッグで画面を回転する
Viewing Parameters	Print	ビューイングパラメータをコンソールに出力する
Set Parameters	ViewSize	ビューのサイズを入力する
	ViewCenter_X	ビューの中心の座標を入力する
	ViewCenter_Y	
	ViewCenter_Z	
	ViewDirection_X	ビューの視線方向ベクトルの成分を入力する
	ViewDirection_Y	
	ViewDirection_Z	
	Set	上記で設定したパラメータを反映してビューを再描画する
-	Draw	ビューを再描画する
	Quit	プログラムを終了する

4.3 Visualizer Control ウィンドウ

View ウィンドウの“Panel” から“Visualizer Control” を指定すると、図 3 のような Visualizer Control ウィンドウが表示される。ここで詳細な描画方法を指定する。操作方法を表 9 に示す。

The screenshot shows a control panel with the following sections and options:

- General**
 - Show Simple Map
- Vehicle**
 - Show ID
 - Vehicle Color Mode
 - vehicle family
 - average velocity
- Intersection**
 - Show ID
- Lane**
 - Show ID
 - Show Intersection Lane
 - Show Section Lane
- Signal**
 - Show
- Roadside Unit**
 - Show
- Connector**
 - Show Connector ID
 - Disable
 - Global ID
 - Local ID

At the bottom of the panel is a "Draw" button.

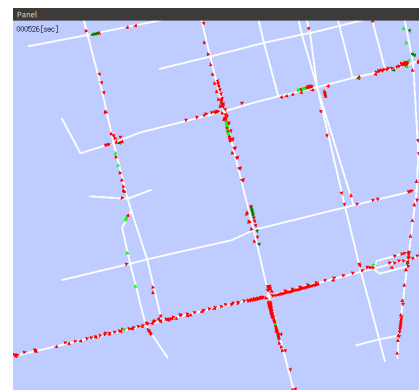


図 4 シンプル描画モード

図 3 Visualizer Control ウィンドウ

表 9 Visualizer Control ウィンドウの操作

カテゴリ	項目	意味
General	Show Simple Map	図 4 のようなシンプル描画モードに切り替える
Vehicle	Show ID	車両 ID を表示する
	Vehicle Color Mode	車両の描画色を選択する
Intersection	Show ID	交差点 ID を表示する
Lane	Show ID	レーン ID を表示する
	Show Intersection Lane	交差点内のレーンを表示する
	Show Section Lane	単路部内のレーンを表示する
Signal	Show	信号現示を表示する
Roadside Unit	Show	感知器の設置箇所を表示する
Connector	Show Connector ID	コネクタ (レーンの結節点) の ID を表示する
-	Draw	上記設定を反映してビューを再描画する

4.4 Simulator Control ウィンドウ

View ウィンドウの“Panel” から“Simulator Control” を指定すると、図 5 のような Simulator Control ウィンドウが表示される。ここでシミュレーションの実行を制御する。操作方法を表 10 に示す。

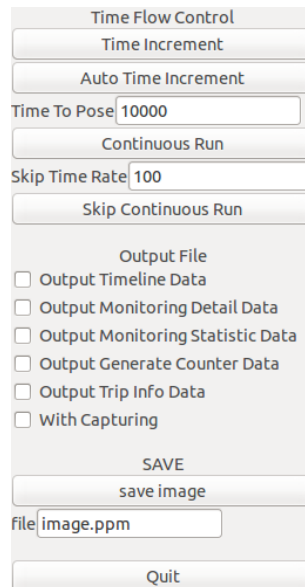


図 5 Simulator Control ウィンドウ

表 10 Simulator Control ウィンドウの操作

カテゴリ	項目	意味
Time Flow Control	Time Increment	シミュレーションを 1 ステップだけ進める
	Auto Time Increment	連続してシミュレーションを進める (再クリックで停止する)
	Time To Pose	Time To Pose で指定した時刻 [msec] までシミュレーションを進める
	Continuous Run	
	Skip Time Rate	Time To Pose で指定した時刻 [msec] までシミュレーションを進めるが、描画は Skip Time Rate で指定した間隔 [msec] で行う
	Skip Continuous Run	
Output File	Output *** Data	結果をテキストファイルに出力する
	With Capturing	result/img/以下にキャプチャ画像を出力する
SAVE	save image	カレントディレクトリに file で指定した名前の画像を出力する
	file	
-	Quit	プログラムを終了する

4.5 Agent Generation ウィンドウ

View ウィンドウの“Panel” から“Agent Generation”を指定すると，図 6 のような Agent Generation ウィンドウが表示される．ここでは任意のタイミングで任意の属性を持った車両を発生させられる．操作方法を表 11 に示す．

図 6 Agent Generation ウィンドウ

表 11 Agent Generation ウィンドウの操作

カテゴリ	項目	意味
Vehicle	Type ID	車種 ID を入力する
	Origin	出発地交差点 ID を入力する
	Destination	目的地交差点 ID を入力する (省略時はランダムに与えられる)
	Routing Parameters	経路探索パラメータを入力する (省略時はランダムに与えられる)
	Stop Points	経由地交差点 ID をカンマ区切りで入力する
	Manually Generate	上記の属性を持った車両を生成し，次のステップで登場させる

4.6 Detail Information ウィンドウ

View ウィンドウの“Panel” から“Detail Information”を指定すると、図7のような Detail Information ウィンドウが表示される。ここで車両や交差点の詳細情報を確認する。操作方法を表12に示す。

The image shows a software interface for searching and displaying information. It is divided into two main sections: 'Vehicle' and 'Intersection'. Each section contains a text input field for an ID, a 'Search' button, and a 'Show Info' button.

図7 Detail Information ウィンドウ

表12 Detail Information ウィンドウの操作

カテゴリ	項目	意味
Vehicle	Vehicle ID	車両 ID を入力する
	Search Vehicle	指定した車両を中心にして再描画する
	Show Vehicle Info	指定した車両の詳細情報をコンソールに出力する
Intersection	Intersection ID	交差点 ID を入力する
	Search Intersection	指定した交差点を中心にして再描画する
	Show Intersection Info	指定した交差点の詳細情報をコンソールに出力する

参考文献

- [1] ADVENTURE Project ウェブサイト. <http://adventure.sys.t.u-tokyo.ac.jp/>.