City Simulator V 2.0

Nội dung:

Tổng quan

Yêu cầu hệ thống

Có gì mới ?

Hướng dẫn cài đặt City Simulator (Thành phố giả lập)

Câu hỏi thường gặp

Chi tiết về cái máy giả lập này :

**Chế độ thực thi:**

**Bắt đầu**

**Tham số mặc định khi chạy**

**Tính chất của CitySimulator**

**City plan DTD**

**City Plan Interface**

**Xác xuất nội bộ (Có thể dịch là xác xuất cố định)**

**Hộp thoại nâng cao: Tùy chọn nâng cao**

**Hộp thoại nâng cao: Cài đặt sự tương tác**

**Output data format (Định dạng dữ liệu đầu ra)**

**Giao diện đồ họa thời gian chạy**

**Mô hình đối tượng**

**Java Class Files**

Author Bios

Tổng quan

***Thế nào là một City Simulator?***

**City Simulator là một mô hình thành phố 3 chiều, có thể mở rộng, nó cho phép tạo ra các dữ liệu không gian (ở đây là các tọa độ) một cách linh động, mô phỏng sự chuyển động của các phần tử (tối đa 1 triệu phẩn tử). Simulator được viết bằng** Java **và được thiết kế để tạo ra dữ liệu thực tế (realistic) để đánh giá các thuật toán cơ sở dữ liệu để lập chỉ mục và lưu trữ dữ liệu về vị trí động (dynamic location).**

**City Simulator được cung cấp trên alphaWorks để việc phát triển các hệ thống lập chỉ mục dữ liệu không gian có thể được kiểm tra về khả năng mở rộng vả hiệu suất dựa trên tập dữ liệu được tiêu chuẩn hóa.**

**Nó hoạt động như thế nào ?**

**Các thông số điều khiển (ở đây là mấy cái param lúc ae chạy chạy ứng dụng cần điền vào) cho phép tạo ra các sự kiện thực tế một cách dễ dàng, ví dụ như lộ trình đi làm hằng ngày. Một GUI (A graphical user interface ) cho phép quan sát sự mô phỏng (ở đây là cho phép quan sát các chuyển động của thực thể ở trên giao diện).**

**City Simulator cho phép tạo ra một mô hình giả lập với hơn 106 cá nhân (individuals) di chuyển dọc theo đường phố, tòa nhà, và giữa các tầng của tòa nhà trong không gian ba chiều. Cài đặt nâng cao cho phép khám phá (đánh giá) hiệu quả của các thuật toán lập chỉ mục qua các chu kì thay đổi (mỗi vòng di chuyển của các thực thể). Đầu ra của Simulation (việc mô phỏng) có thể được xem trong thời gian thực ; đầu ra cũng được tạo ra cách nhau bởi dấu phẩy (để đuôi csv xem trên microsoft excel rất trực quan), với duy nhất 1 ID , timestamp, và tọa độ (x, y, z). Dữ liệu này cso thể được import vào bất kì cơ sở dữ liệu nào để nghiên cứu lập chỉ mục không gian và công nghệ truy vấn không gian**

**Trademarks:**

Java **and all Java-based trademarks are trademarks of Sun Microsoftsytems, Inc, in the United State, other country, or both.**

Platform Requirements for City Simulator

**Platform**

**Windows® 95 or above**

**Java Tools**

**JRE**

**Computer**

**Pentium II or above**

**Minimum of 128MB of RAM**

Có gì mới?

**Phiên bản 2.0 của City Simulator đi kèm theo nhiều tính năng mới:**

1. Kế hoạch thành phố hiện có thể được tạo từ tài liệu .xml. Một kế hoạch thành phố mẫu đi kèm với gói. Xem tệp mẫu city\_plan.xml để biết ví dụ về thành phố định nghĩa loại tài liệu kế hoạch (dtd). Để tạo kế hoạch thành phố của riêng bạn, bạn có thể viết bộ lọc của riêng bạn, ví dụ: nhập dữ liệu GML hoặc xác định các tòa nhà của riêng bạn và những con đường. Miễn là bạn tuân theo city\_plan dtd, chỉ cần đặt tên cho tệp xml của bạn là city\_plan.xml và City Simulator sẽ chạy cùng với kế hoạch thành phố của bạn. (Cái này dịch hơi tối nghĩa, hãy đọc đọc kĩ phần này và xem trong code những file mà nó nhắc đến để hiểu thêm)
2. CityPlan.java class bây giờ sẽ có interface CityPlanInterface.java và abstract class CityPlanAbstract.java. Nếu bạn muốn viết CityPlan cho riếng bạn để gennerate một plan (có lẽ là một cái generate ra một đống chấm nhỏ kia). Thì bạn hãy xem file XML\_CityPlan.java để biết làm như nào cho chuẩn nhé
3. Trình giả lập (simulator) bây giờ đã có file thuộc tính (CitySimulator.properties) dùng để cấu hình cài đặt cài đặt chương trình. CLI vẫn tồn tại giống như GUI (và đặt lên trên các giá trị trong tệp thuộc tính nếu được sử dụng). File thuộc tính cho phép điều khiển nhiều tham số khác nhau bao gồm tên của CityPlan class mà trương trình nên dùng (eg. XML\_CityPlan.java or your own class).
4. Các thông số ảnh hưởng đến quy tắc chuyển động hiện có thể được thay đổi trong thời gian thực nếu bạn chạy bằng cách sử dụng GUI. Chỉ cần nhấp vào một địa điểm (Đường, Tòa nhà, Giao lộ, v.v.) vàbạn có thể thay đổi xác suất giảm xuống đối với các tầng của tòa nhà, gây ra sự chậm trễ giao thông trên đường xá, gọi một mô hình phân luồng giao thông thực tế cho các con đường và giao lộ, v.v. Nhiều tham số là các biến lớp không tĩnh để người ta có thể làm trống một tòa nhà trong khi lấp đầy khác. (chức năng này hình như mình không cần làm )
5. A Boolean parameter conservePopulation has been added to the property file. When set to false, this parameter will cause the population do decrease (ie as people leave the city they are not replaced). Removal of people from the city after the relaxation is complete (determined by relax moves). This feature allows one to investigate the rate at which a region of a city can be evacuated. When set to false, population vs time will be displayed in a new graphical window. Poputation vs time data will also be output to a file “pop\_”+filename.txt where filename is the datafile name you specify.
6. Bây giờ đã có thêm màu sắc. Một phương pháp công khai có sẵn cho các tòa nhà(thang độ xám) dựa trên chiều cao của chúng
7. Địa điểm mở rộng đã được thêm vào
8. Thêm file runme.bat
9. CitySimulator bây giờ đã có thể chạy ngon trên Linux

**Installation Instruction for City Simulator**

1. Create a directory on your machine (such as City Simulator)
2. Download the zip file and extract it into that directory
3. From a command prompt, cd to your City Simulator directory
4. Type "java -classpath CitySimulator.jar CitySimulator.CitySimulator".

City Simulator requires that you specify a data directory for the output file. By default, this is C:\Temp. You can specify any data directory you wish, but it must already exist when you run the simulation.

**FAQ**

1. Ứng dụng nào có thể sử dụng cái data được generate ra bởi City Simulator

Đầu ra chỉ đơn giản là một tệp văn bản (với các biến được phân tách bằng dấu phẩy). Tệp văn bản chứa một ID người, dấu thời gian và tọa độ x, y, z. Dữ liệu này có thể được nhập vào bất kỳ cơ sở dữ liệu nào để nghiên cứu các công nghệ lập chỉ mục không gian và truy vấn không gian khác nhau.

1. Chạy cái City Simulator như nào ?

From a command line, type "java -classpath CitySimulator.jar CitySimulator.CitySimulator" or add the following command line options (see ReadMe.txt for details):

java -classpath CitySimulator.jar CitySimulator.CitySimulator true/false filename.ext drive:\datadirectory\ numpeople maxRelax\_Steps finalSteps

1. After I launch City Simulator and press Start, why do I get a message that reads "String entered is not a valid directory"?

City Simulator requires that you specify a data directory for the output's text data file. You may specify any directory, but it must already exist when you press Start. The default data directory is C:\Temp.

**Simulator Details**

City Simulator là một thành phố Java, được tạo ra để generate ra dữ liệu thực để đánh giá các thuật toán cơ sở dữ liệu cho việc đánh chỉ mục và lưu trữ dữ liệu về dynamic location. We made this simulator available on alpha works so that individuals developing systems to index location data can test scalability and performance against a standardized dataset. The CitySimulator can generate a simulation with over 106 individuals moving along streets, buildings, and between building floors in three dimensions. Advanced settings allow one to explore efficiency of indexing algorithms over daily commute cycles. The output of the simulation can be watched in real time and/or output as a comma separated variable (csv) text file (đoạn này y hệt đoạn trên kia nên k cần dịch).

**Execution Modes :**

Cách chạy City Simulator dễ nhất là chạy file runme.bat đã được cung cấp sẵn trong source.

CitySimulator có thể được gọi ở chế độ hàng loạt bằng dòng lệnh giao diện (CLI), hoặc sử dụng giao diện người dùng đồ họa CitySimulator (GUI). Các giao diện dòng lệnh cho phép thực thi với khả năng sửa đổi nhiều thông số chạy mô phỏng. Giao diện GUI chứa các hộp thoại nâng cao để điều chỉnh đầy đủ các thông số mô phỏng. Trong cả hai trường hợp, người dùng có thể bật hoặc tắt đồ họa thời gian chạy để xem mô phỏng đang diễn ra hoặc để nhận ra hiệu suất tối đa cho các mô phỏng lớn.

**Getting Started:**

***java -classpath CitySimulator.jar CitySimulator.CitySimulator***

Execution with graphics, loading runtime defaults. Parameters may be changed with the “Input Run Parameters Dialogue”.

***java -classpath CitySimulator.jar CitySimulator.CitySimulator true/false***

Execution with runtime defaults and graphics enabled (true) or disabled (false).

***java -classpath CitySimulator.jar CitySimulator.CitySimulator true/false filename.ext***

Execution with runtime defaults and graphics enabled (true) or disabled (false). Replacing the default file name with filename.ext

***java -classpath CitySimulator.jar CitySimulator.CitySimulator true/false filename.ext drive:\datadirectory\***

Execution with runtime defaults and graphics enabled (true) or disabled (false).

Replacing the default file name with filename.ext

Replacing the default data directory with drive:\datadirectory\. This directory must exist.

***java -classpath CitySimulator.jar CitySimulator.CitySimulator true/false filename.ext drive:\datadirectory\ numpeople maxRelax\_Steps finalSteps***

Tương tự như trên với việc bổ sung các tham số sau:

NumPeople: Số người được giả lập(được thêm vào thành phố)

MaxRelax: Số bước thư giãn tối đa trước khi bắt đầu mô phỏng. Các trình mô phỏng lặp lại cho đến khi chu kỳ maxRelax hoặc cho đến khi ngưỡng start đạt (xem startThreshold bên dưới). Theo mặc định là 2000

finalSteps(bước cuối cùng): Số chu kỳ mô phỏng thực tế được lưu trữ trong datafile.

***java -classpath CitySimulator.jar CitySimulator.CitySimulator true/false filename.ext drive:\datadirectory\ numpeople max\_Relax\_Steps final\_steps startThreshold fillThreshold emptyThreshold***

Tương tự như trên với việc bổ sung các tham số sau:

startThreshhold: Mô phỏng bắt đầu với những người được thêm vào thành phố trên các con đường tại (mười) điểm vào hoặc "cầu" vào thành phố. Theo thời gian các tòa nhà đóng vai trò như bồn rửa cho người dân trên đường và trên đường hoặc dân số tầng trệt giảm khi các tòa nhà lấp đầy. Bắt đầu ngưỡng thể hiện phần số người nên ở trên mặt đất khi mô phỏng nên "bắt đầu" (tức là khi thư giãn là xong). Khi bắt đầu Ngưỡng 15% (0,15, giá trị mặc định) biểu đồ số người so với số sàn đối sánh biểu đồ số lượng địa điểm (tầng) so với số tầng do đó thành phố được coi là đang ở hoặc gần trạng thái ổn định.

fillThreshold: Ngưỡng này đại diện cho tỷ lệ tối thiểu những người bao giờ nên được để lại trên mặt đất. Khi mặt bằng dân số giảm xuống dưới phần fillThreshold, tòa nhà nhập và xác suất thoát và xác suất lên và xuống của sàn là đã thay đổi để kích hoạt lộ trình đi làm ra khỏi các tòa nhà. Theo mặc định, fillThreshold là 9% (0,09). Phiên bản hiện tại của mã không cho phép điều chỉnh xác suất vào / ra hoặc lên / xuống - những đã được điều chỉnh để tạo ra sự vận chuyển hợp lý của mô phỏng Mọi người. Phiên bản trong tương lai có thể cho phép chúng được thay đổi.

emptyThreshold: Ngưỡng này đại diện cho tỷ lệ tối đa những người bao giờ nên được để lại trên mặt đất sau khi đi làm ra khỏi các tòa nhà đã được kích hoạt. Khi dân số mặt đất vượt quá emptyThreshold, tòa nhà đi vào và đi ra xác suất và xác suất lên xuống của tầng được thay đổi để kích hoạt tuyến đường đi làm trở lại các tòa nhà (tức là chúng được khôi phục thành các giá trị mặc định ở đầu mô phỏng). Theo mặc định Ngưỡng trống là 50% (0,50). Phiên bản hiện tại của mã không cho phép điều chỉnh nhập / xuất hoặc lên / xuống xác suất - những xác suất này đã được điều chỉnh để tạo ra vận chuyển của những người mô phỏng. Phiên bản trong tương lai có thể cho phép họ được thay đổi.

*Note: Nếu chế độ đồ họa được đặt thành “true”, người dùng vẫn có thể ghi đè hoặc sửa đổi bất kỳ và tất cả các thông số nhập trên dòng lệnh. Các thay đổi được thực hiện bằng hộp thoại Run Parameter trong GUI*

**Default run parameters:**

Nếu không có gì đặc biệt, các biến mặc định được chạy sẽ là :

FileName : test.txt

DataDirectory: c:\temp\ (the specified directory must be exist)

Numpeople: 100 (try 100 – 10^6)

MaxRelax : 2000 (need not be any larger for startThreshhold >= 0.15)

FinalSteps: 200

StartThreshold: 0.15

FillThreshold: 0.09

EmptyThreshold: 0.50

Xác suất cố định (nội bộ) (không thể thay đổi trong phiên bản hiện tại):

// in and out probabilities

static double exitHigh 0.9; // commute out

static double exitLow 0.1; // commute in

static double enterProb 0.1;

static double exitProb exitLow; // default at start

// up and down probabilities

static double downHigh 0.06; // commute out

static double downLow 0.03; // commute in

static double upProb 0.03;

static double downProb downLow; // default at start

**CitySimulator.properties**

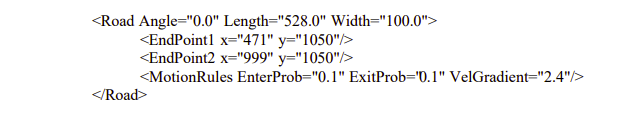
Tệp thuộc tính cho phép kiểm soát nhiều tham số khác bao gồm tên của Lớp CityPlan mà chương trình nên sử dụng (ví dụ: XML\_CityPlan.java hoặc lớp của riêng bạn). Bất kì thuộc tính có thể được chỉ định trong dòng lệnh hoặc GUI cũng có thể được đặt trong tệp thuộc tính. Nếu không có đối số dòng lệnh nào được sử dụng, các giá trị được chỉ định trong tệp thuộc tính được sử dụng theo mặc định và sẽ xuất hiện trong hộp thoại GUI (xem bên dưới). Chỉ cần chỉnh sửa CitySimulator.properties để tùy chỉnh mô phỏng.

**City Plan DTD**

Kế hoạch thành phố hiện có thể được tạo từ một tài liệu xml. Một kế hoạch thành phố mẫu đi kèm với gói hàng (package). Hãy xem tệp mẫu city\_plan.xml để làm ví dụ. Để tạo thành phố của riêng bạn kế hoạch, bạn có thể viết bộ lọc của riêng mình, ví dụ: nhập dữ liệu GML hoặc xác định bộ lọc của riêng bạn các tòa nhà và đường xá. Miễn là bạn tuân theo city\_plan dtd, chỉ cần đặt tên cho tệp xml của bạn city\_plan.xml và City Simulator sẽ chạy cùng với kế hoạch thành phố của bạn. Bạn cũng có thể chỉ định tên của tệp xml kế hoạch thành phố bằng cách thay đổi thuộc tính xmlPlanFile trong CitySimulator.properties.

Để đơn giản, trong phiên bản này, chúng tôi sử dụng một định nghĩa không gian chung cho tất cả các đối tượng. Trong phiên bản 2.0, tất cả các vị trí đều là hình chữ nhật hoặc hình chữ nhật xoay (đại diện cho mỗi người trong thành phố). Các phiên bản trong tương lai có thể hỗ trợ các giao diện khác để đặt các đối tượng và các hình dạng khác. Hiện tại, màu sắc được chỉ định trong lớp City.java riêng và không thể tùy chỉnh.

Để tạo bất kỳ đối tượng nào, phần tử xml phải chỉ định lớp đối tượng (ví dụ: Đường), hướng của đối tượng (góc tính bằng radian từ + PI đến –PI), chiều dài và chiều rộng của vật. Hai nút con phải chỉ định Điểm cuối của đường tâm đối tượng. Các khoảng cách giữa các điểm cuối này phải tương ứng với thuộc tính Độ dài được đặt trong cấp độ gốc thành phần. Thuộc tính Chiều dài chỉ được sử dụng để xác thực kế hoạch thành phố của bạn (nó được thử nghiệm dựa trên độ dài được tính từ hai điểm cuối). Trong phiên bản V2.0, tất cả các vị trí đều là hình chữ nhật và được chỉ định với dtd sau.



Trong quy hoạch thành phố mẫu, các con đường được chia thành các đoạn ngắn để tận dụng lợi thế của khả năng thời gian thực để tạo ra các vật cản cục bộ và khả năng chống lại luồng giao thông trên bộ phận đường.

Lưu ý rằng các giá trị x, y có thể là số nguyên hoặc số thực. Chiều dài và Chiều rộng phải là được chỉ định dưới dạng phao. Bạn có thể sử dụng bất kỳ đơn vị nào trong x và y (Vĩ độ, Dài, mét, v.v.). Các trình mô phỏng sẽ chia lại tỷ lệ tọa độ xy thành các đơn vị bên trong bằng cách sử dụng CityScale, Xmin và Các thuộc tính Ymin được xác định trong phần tử gốc.

<CityPlan CityScale="scale" Xmin="0.0" Ymin="0.0">

Bạn phải xác định ba thuộc tính này để khớp quy mô thành phố của bạn với quy mô nội bộ được sử dụng trong mô phỏng. Để thực hiện điều này, Xmin và Ymin nên được đặt thành X tối thiểu và Giá trị Y được xác định trong kế hoạch của bạn (trong bất kỳ đơn vị nào bạn chọn sử dụng). The CityScale tham số phải được chọn sao cho bất kỳ điểm x, y nào trong thành phố của bạn

x\*scale <= CITYSIZE

CITYSIZE is defined in the properties file and has a default value of 6300.

Trong quy hoạch thành phố mẫu, Hệ số quy mô được đặt là 1,0 và thành phố được xác định theo các đơn vị đầu cuối. Để xác định hệ số tỷ lệ của bạn, hãy tính (xmax-xmin) và (ymax-ymin) cho thành phố của bạn theo đơn vị của bạn. Cái lớn hơn trong hai cái này là của bạn (phạm vi thành phố). Đặt thuộc tính CityScale thành:

Scale = CITYSIZE/(city extent) for your city

Lưu ý rằng độ cao (của các tầng) không bị ảnh hưởng bởi hệ số tỷ lệ (Note that the altitude (of floors) is not affected by the scale factor)

Có một yếu tố tỷ lệ thứ hai trong tệp thuộc tính được gọi là MAPSCALE. Điều này quyết định kích thước màn hình hoặc MAPSIZE được hiển thị bằng pixel. Với CITYSIZE = 6300 và MAPSCALE = 10 kích thước màn hình MAPSIZE = 630 pixel. Đây là những khuyến nghị các giá trị. Nếu bạn thay đổi MAPSCALE hoặc CITYSIZE trong tệp thuộc tính, bạn cũng PHẢI thay đổi MAPSIZE.

MAPSCALE=10

CITYSIZE=6300

MAPSIZE=630 #MAPSIZE=CITYSIZE/MAPSCALE

Quy hoạch thành phố xml cũng phải chỉ định các quy tắc chuyển động. Chúng tôi đề nghị ban đầu bạn nên sử dụng quy tắc được cung cấp trong tệp mẫu ít nhất là ban đầu. Các thuộc tính tham số chuyển động là nơi phụ thuộc (xem mẫu). Chúng có thể được thay đổi theo thời gian thực trong quá trình mô phỏng bằng cách nhấp vào một địa điểm trên bản đồ.

**City Plan Interface**

The CityPlan.java class now has an interface CityPlanInterface.java and abstract clas CityPlanAbstract.java. If you wish you may write your own CityPlan class to generate a plan.

Any class to create a city plan must extend CityPlanAbstract. For example:

public class XML\_CityPlan extends CityPlanAbstract

Your class must also define a buildPlan method as specified in the public CityPlanInterface.

public void buildPlan(City acity);

See the sample code XML\_CityPlan.java for an example of how this is done.

**Đối thoại nâng cao: “Tùy chọn nâng cao” (cài đặt):**

*Với các tham số chạy mặc định, khoảng thời gian của chu kỳ đi lại là khoảng XXX*

*các chu kỳ chương trình.*

**Gif File Name (not used)**

**Start Threshold .15**

Quá trình cân bằng hoàn thành khi 15% số người ở trên mặt đất và phần còn lại đang xây dựng các tầng. Ở giá trị này, biểu đồ của người so với tầng xây dựng xấp xỉ mật độ tầng so với chiều cao tầng). Xem thảo luận của các tòa nhà và các tầng.

**Fill Theshold .09**

Khi còn lại 9% người trên mặt đất, hãy chuyển sang commute-to-work parameters.

**Empty Threshold .5**

Khi 50% người trên đường chuyển sang commute-to-work parameters.

**Đối thoại nâng cao: “Đặt tương tác”**

Với các tham số chạy mặc định, khoảng thời gian của chu kỳ đi lại là khoảng XXX chu kỳ chương trình.

**Prob thoát tối đa 0.9**

Xác suất thoát khỏi một tòa nhà nếu ở trong một tòa nhà và gần cửa ra vào ở trạng thái đi làm.

**Xác suất thoát tối thiểu 0,1**

Xác suất thoát khỏi một tòa nhà nếu ở trong một tòa nhà gần cửa ra vào ở trạng thái đi làm.

**MaxDown Prob 0.06**

Xác suất chuyển xuống tầng thấp hơn nếu ở tầng trong trạng thái đi làm từ nơi làm việc.

**Xác suất giảm tối thiểu 0,03**

Xác suất chuyển xuống tầng thấp hơn nếu ở tầng trong trạng thái đi làm.

**Nhập Prob 0,1**

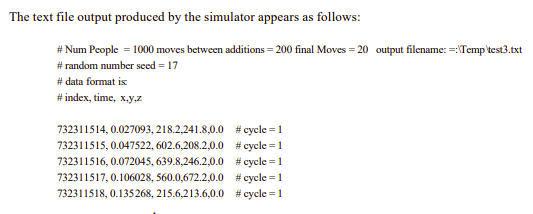
Xác suất vào một tòa nhà nếu chỉ ở bên ngoài một tòa nhà gần cửa ra vào.

**Tăng Prob 0.03**

Xác suất di chuyển lên tầng cao hơn trong tầng của tòa nhà (bên dưới tầng trên cùng).

**Output Data Format:**

The text file output produced by the simulator appears as follows:



Bốn dòng đầu tiên chứa thông tin tiêu đề.

Dòng tiêu đề 1,2 tóm tắt một số tham số chạy.

Dòng 3 và 4 mô tả định dạng dữ liệu.

Theo sau tiêu đề là dữ liệu chạy. Trong mỗi hàng theo sau tiêu đề là 5 cột dữ liệu.

Cột 1, một số nguyên, là khóa chính duy nhất cho mỗi "người" mô phỏng đang được theo dõi. Để tăng tốc độ mô phỏng, chúng tôi không tạo khóa ngẫu nhiên mới cho mỗi người. Thay vì chúng tôi chọn khóa ngẫu nhiên và sau đó để tạo ra một khóa duy nhất mới cho mỗi người. Không có ý nghĩa khác đối với con số.

Cột 2-5 chứa số chính xác gấp đôi. Cột thứ hai là thời gian (tính bằng giây) tại đó báo cáo vị trí mô phỏng đã được "nhận".

Các cột 3-5 đại diện cho vị trí (x, y, z) trong METERS cho báo cáo vị trí đó. Ban đầu, giá trị z cho tất cả số người là 0,0 khi mọi người được giới thiệu về thành phố tại các điểm vào trên những con đường mà tất cả đều có độ cao bằng 0 (chúng tôi có thể thêm các biến thể chiều cao cho các con đường và tầng trệt của các tòa nhà nhưng vẫn chưa thực hiện được). Như tiến độ thời gian và mọi người di chuyển đến và lên các tầng của tòa nhà, giá trị z được báo cáo sẽ tương ứng với độ cao của sàn nhà mà người đó đang ở.

Cuối cùng, sau dữ liệu, mỗi hàng chứa một chú thích cho biết số chu kỳ. Một chu kỳ hoàn chỉnh mất đặt mỗi khi mỗi báo cáo vị trí của người được mô phỏng được cập nhật. Vì vậy, nếu dân số là 10.000 chu kỳ sẽ chứa 10.000 bản cập nhật hoặc báo cáo. Con số này chỉ có thể thay đổi một chút khi mọi người rời thành phố. Bất cứ khi nào một người rời đi, một người mới được giới thiệu đến thành phố để tổng dân số được giữ không đổi. Chúng tôi xem dữ liệu vị trí hết hạn (một người rời khỏi khu vực) như một thực tế cần được xử lý bởi bất kỳ lập chỉ mục nào kế hoạch.

**Run Time GUI**

Khi bắt đầu mô phỏng, dữ liệu có thể được nhập qua hộp thoại “Tham số chạy đầu vào”. Sau khi bắt đầu được nhấn, ba cửa sổ mới sẽ xuất hiện: (At the start of the simulation, data may be input via the “Input Run Parameters” dialogue. After the start button is pressed, three new windows will appear:)

1) A console window containing run time messages and information

2) A map of the city

3) A “Floor Population Histogram” window.

Đồ họa sẽ tĩnh trong một thời gian trong khi trình mô phỏng thực hiện giai đoạn "thư giãn" được mô tả ở trên. Sau khi thư giãn, những người đang di chuyển sẽ hiển thị và được biểu thị dưới dạng các điểm được hiển thị ở một trong ba màu sắc. Điểm trắng là những người trên đường hoặc ở tầng trệt của một tòa nhà. Các điểm màu đỏ nằm trên tầng trệt nhưng không ở tầng trên cùng. Điểm vàng là những người ở tầng cao nhất của một tòa nhà. Tổng số số tầng khác nhau từ tòa nhà này sang tòa nhà khác.

Hộp thoại có nhãn Biểu đồ dân số tầng cũng hiển thị biểu đồ của các tầng của tòa nhà (không đổi) như sự phân bố của con người (động) như một chức năng của tầng xây dựng. (The dialogue labeled Floor Population Histogram shows the histogram of building floors (constant) as well as the distribution of people (dynamic) as a function of building floor.)

**Object Model**

The city is constructed with collections of various objects. The high level objects in the simulation are:

Place

Person

The following object extend place:

Building

Road

Intersection

Floor

Places have several attributes. Coordinates, extents, altitude or floor number, and pointers to neighboring places. Places also contain enter and exit probabilities, up down probabilities, drift probabilities (on roads), scatter probabilities (on intersections), etc. People move according to rules based on the place they are in. A person on a building floor does a random walk. At specific points (stairways) they may, move up or down (if not at the top floor or ground floor respectively) , leave a building (if near a door) etc. A person on a road moves with a linear combination of: Velocity = (random walk component) + (Drift Velocity). Magnitude of the drift velocity increases as a person moves closer to the center of a road. Direction of the drift velocity changes sign at the center (so on one side of a road people move North or East, on the other side they move South or West). Road objects also have orientations that determine if the drift velocity is North/South or East West.

The implementation of this object model is quite efficient and simulations of up to one million people may be fully relaxed and iterated for several hundred cycles in a run time of just a few hours on a PC.

The package CitySimulator.jar contains the following files

**Java class files:**

Building.class

ChartPanel.class

City.class

CityPlan.class

CityPlanAbstract.class

CityPlanInterface.class

CitySimulator.class

CitySimulator.properties

CityWindowAdapter.class

Console.class

Constants.class

DocumentOutputStream$1.class

DocumentOutputStream.class

ExitPoint.class

Floor.class

GrassyField.class

GridBagHelper.class

Input\_data.class

Intersection.class

LocationChartFrame.class

matrixDisplay.class

MyOutput.class

Person.class

Place.class

PointVelocity.class

Road.class

RunSimulation.class

TransPoint.class

VFlowLayout.class

XML\_CityPlan.class

XML\_CityWriter.class

**Gif Icon File:** city2.gif

**Author Bios:**

**James Kaufman**

IBM Almaden Research Center

Dr. James H. Kaufman is a Research Staff Member in the Web Technologies Department at the Almaden Research Center, IBM Research Division. Dr. Kaufman received his B.A. in Physics from Cornell University and his PhD in Physics from U.C.S.B.. Dr. Kaufman has worked in several areas of research at IBM including micro-magnetic modeling, magneto-electronic devices, dynamical systems, pattern formation, critical phenomena, and diamond-like carbon for protective thin film disk overcoats. He has also served as a technical manager, and led a department in Science and Technology responsible for development new materials for thin film recording heads and non-volatile magnetic memory (MRAM). He recently moved to computer science where his current research interests included management of dynamic location data and technology for privacy protection. James can be reached through email at [kaufman@almaden.ibm.com](mailto:kaufman@almaden.ibm.com).

**Jussi Myllymaki**

IBM Almaden Research Center

Dr. Jussi Myllymaki is a Research Staff Member in the Web Technologies Department at the Almaden Research Center, IBM Research Division. He received his MS degree in Industrial Management from Helsinki University of Technology, Finland, and his MS and PhD degrees in Computer Science from the University of Wisconsin at Madison. His early work focused on performance evaluation of tertiary storage devices and database systems. Later, he worked in Web search engine technology and Web data extraction (see the IBM developerWorks article at http://www-106.ibm.com/developerworks/library/wa-wbdm/). Dr. Myllymaki’s current interests include location-based services and management of dynamic location data. His IBM Research web page is at <http://www.research.ibm.com/people/j/jussi/>.

Jussi can be reached through email at [jussi@almaden.ibm.com](mailto:jussi@almaden.ibm.com)

**Jared Jackson**

IBM Almaden Research Center

Jared Jackson is a Research Associate at IBM's Almaden Research Center and a recent graduate from Harvey Mudd College with a degree in Computer Science. Jared can be reached through email at jjared@almaden.ibm.com.