

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN



GIAO THÔNG THÔNG MINH
BÁO CÁO BÀI TẬP NHÓM 6
ĐỀ TÀI: TÌM HIỂU VỀ SUMO-GUI

Giáo viên hướng dẫn: Đỗ Bảo Sơn

Nhóm thực hiện: Nhóm 6

Thành viên:

Nguyễn Trung Nam

Nguyễn Đoàn Đăng

Nguyễn Sỹ Đức

Phạm Trần Anh

Lớp: 70DCTT21

Hà Nội năm 2021

BẢNG PHÂN CHIA CÔNG VIỆC

Họ tên	Nhiệm vụ	Đánh Giá
Nguyễn Trung Nam (Nhóm trưởng)	Tìm hiểu nội dung và làm word.	Hoàn Thành
Nguyễn Đoàn Đăng	Tìm hiểu nội dung và làm word.	Hoàn Thành
Nguyễn Sỹ Đức	Tìm hiểu nội dung và làm word.	Hoàn Thành
Phạm Trần Anh	Tìm hiểu nội dung và làm word.	Hoàn Thành

MỤC LỤC

BẢNG PHÂN CHIA CÔNG VIỆC	1
1. Giới thiệu:.....	4
2. Mô tả sử dụng:.....	4
3. Tập cấu hình và ảnh chụp màn hình:	5
4. Đa cửa sổ xem:	6
5. Trực quan hóa edge-related data:	7
6. Ví dụ sử dụng:	7
6.1. Hiển thị văn bản tùy trong chế độ xem mô phỏng:.....	7
6.2. Hiển thị logo trong chế độ mô phỏng:	8
6.3. Hiển thị các tuyến đường vào các thông tin tuyến đường:.....	8
6.4. Điều tra các internal lane và internal edge:.....	8
TÀI LIỆU THAM KHẢO:.....	10

DANH MỤC HÌNH ẢNH

Hình 2.1 Giao diện sumo-gui	4
Hình 3.1: Ví dụ tệp cấu hình gui-setting	5
Hình 3.2: Ví dụ điểm ngắt	5
Hình 3.3: Ví dụ tham chiếu lược đồ bằng tên	6
Hình 4.1: Ví dụ đa cửa sổ xem	6

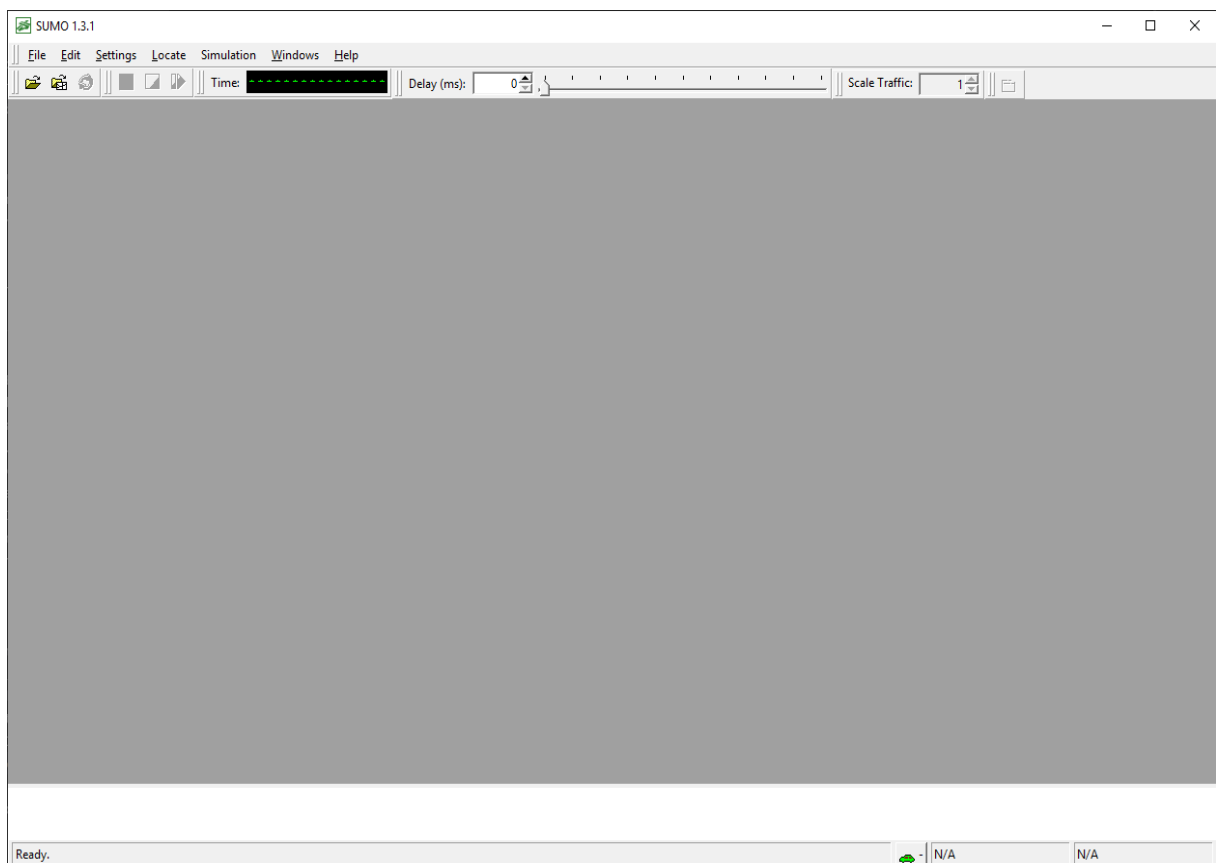
1. Giới thiệu:

Sumo-gui về cơ bản là ứng dụng tương tự như sumo, chỉ được mở rộng bởi giao diện người dùng đồ họa.

- Mục đích: Mô phỏng một kịch bản được xác định.
- Hệ thống: Portable (Linux/Windows được thử nghiệm); mở một cửa sổ.
- Đầu vào (bắt buộc): Tập cấu hình SUMO.
- Đầu ra: sumo-gui tạo ra đầu ra tương tự như sumo.
- Ngôn ngữ lập trình: C++

2. Mô tả sử dụng:

- Là một ứng dụng dựa trên cửa sổ, sumo-gui được bắt đầu bằng một cú nhấp chuột kép với nút chuột trái trên Windows, trên Linux có thể chỉ bằng một cú nhấp chuột. Sau đó, một cửa sổ trống sẽ hiển thị, tương tự như cửa sổ được hiển thị trong hình ảnh.



Hình 2.1: Giao diện sumo-gui

3. Tập cấu hình và ảnh chụp màn hình:

Sumo-gui sử dụng các tập cấu hình tương tự như SUMO. Các tùy chọn có thể nhận được bằng cách gọi `sumo --help` hoặc bạn có thể lưu tập cấu hình bằng cách gọi “**sumo –save-template <file> --save-commented**”. Lựa chọn **–gui-settings-file** là giành riêng cho **sumo-gui**. Nó cho phép bạn tải lên một tập **gui-settings** đã lưu trước đó. Cách dễ nhất để nhận tập **gui-settings** là thông qua View Settings-Dialog 🌈. Chỉ cần sửa đổi cài đặt và lưu.

example.sumocfg

```
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/sumoConfiguration.xsd">

  <input>
    <net-file value="C:\demoGTTM\ok1.net.xml"/>
    <route-files value="C:\demoGTTM\okok.rou.xml"/>
  </input>

</configuration>
```

Hình 3.1: Ví dụ tập cấu hình example.sum

gui-setting.cfg

```
<viewsettings>
  <scheme name="..."
  ...
  </scheme>

  <viewport zoom="200" x="100" y="-100"/>
  <delay value="42"/>
  <decalf file="background.gif" centerX="550.00" centerY="1530.00" width="64.00" height="64.00" rotation="0.00"/>
  <breakpoint value="42"/>
  <breakpoint value="1337"/>
</viewsettings>
```

Hình 3.2: Ví dụ tập cấu hình gui-setting

Có thể thêm tập điểm ngắt vào cài đặt mình theo cách thủ công.

```
<viewsettings>
  ...
  <breakpoints-file value="breakpoints.txt"/>
</viewsettings>
```

Hình 3.3: Ví dụ điểm ngắt

Một file phù hợp cho việc tải điểm ngắt có thể nhận bằng cách thiết lập các điểm ngắt trong gui và sử dụng menu-option để lưu lại (Edit – Edit Breakpoints – Save).

Tải *example.sumocfg* bằng cách mở *simulation-dialog* hoặc sử dụng command-line “**sumo-gui -example.sumcfg**”.

Có thể sử dụng tệp định nghĩa lược đồ XML để thiết lập cấu hình sumo-gui. Cũng có thể tham chiếu một lược đồ được xác định trước chỉ bằng tên:

```
<viewsettings>
  <scheme name="real world"/>
</viewsettings>
```

Hình 3.4: Ví dụ tham chiếu lược đồ bằng tên


- Ảnh chụp màn hình:

Có thể chụp ảnh màn hình vào các thời điểm xác định trước bằng cách thêm các phần tử vào cấu hình:

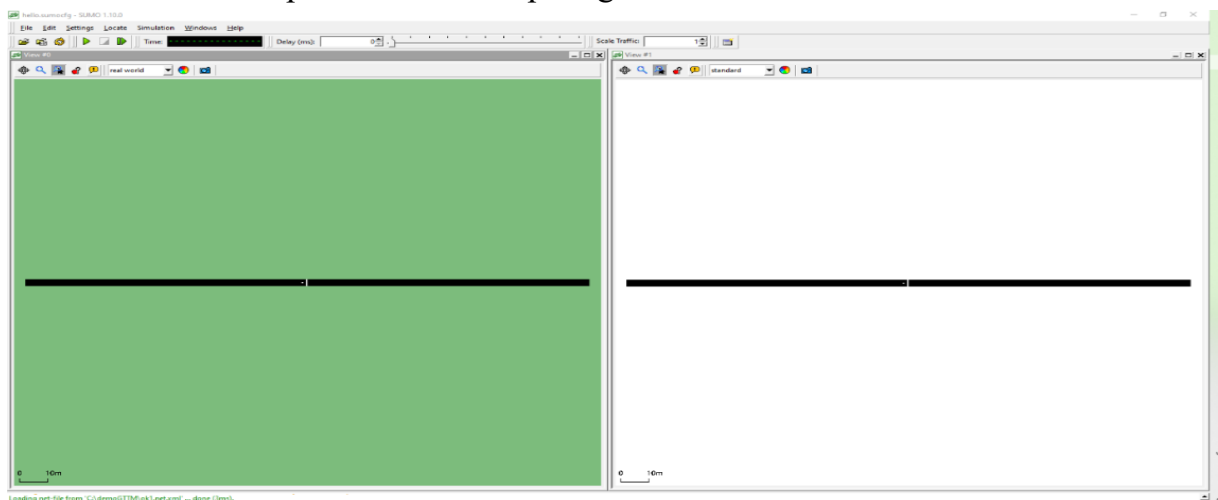
```
<viewsettings>
  <snapshot file="myScreenshot.png" time="42"/>
</viewsettings>
```

Hình 3.5: Ví dụ ảnh chụp màn hình

4. Đa cửa sổ xem:

Sử dụng nút  để mở nhiều cửa sổ xem trên cùng một mô phỏng. Các cài đặt hiển thị có thể được đặt độc lập cho mỗi chế độ xem. Có thể quản lý các cửa sổ xem bằng menu Windows trên thanh menu chính.

Khi chuyển nhiều tệp sang tệp sumo –option ==**gui settings-**, một cửa sổ xem sẽ được mở cho mỗi tệp khi bắt đầu mô phỏng.



Hình 4.1: Ví dụ đa cửa sổ xem

5. Trực quan hóa edge-related data:

Một số ứng dụng đã tạo ra các biện pháp đo edge-related cho một hoặc nhiều khoảng thời gian.

- Các tệp đầu ra edge data
- Tệp edge-probability tạo ra bởi randomTrips.py
- Marouter netload-output
- Làm mượt thời gian di chuyển từ device.rerouting khi chạy **sumo** với option – **device.rerouting.output**
- **Netedit** có thể được dùng để tạo và chỉnh sửa các tệp edgeData.
- Các tệp này có thể sử dụng với **duarouter** và **sumo** để chỉnh việc định tuyến xe. Khi chạy **sumo-gui**, các tệp này có thể được hiển thị bằng cách thiết lập màu cạnh theo trọng lượng được tải.
- **Sumo-gui** cũng có thể được sử dụng để trực quan hóa dữ liệu được chứa để xem các biện pháp lưu lượng khác nhau đã thay đổi như thế nào theo thời gian.
- Các tệp dữ liệu biên tập để trực quan hóa có thể được tải bằng tùy chọn cài đặt – **edgedata-files**. Tất cả các thuộc tính sẽ được tải và có thể được chọn trong cài đặt hiển thị cạnh. Chúng cũng có thể được tải trong **sumo-gui** từ menu bằng cách sử dụng Files-> Open EdgeData. Thời gian kết thúc mô phỏng sẽ được tự động điều chỉnh đến cuối khoảng dữ liệu.
- Để sử dụng dữ liệu đã tải, màu của edge phải được đặt thành màu của edgeData. Nút *Recalibrate Rainbow* có thể được sử dụng để tạo bảng màu để hiển thị toàn bộ khoảng dữ liệu.
- Khi xác định bảng phối màu, luôn có thể chỉnh cấu hình màu dành riêng cho dữ liệu bị thiếu ('No Data').

6. Ví dụ sử dụng:

6.1. Hiển thị văn bản tùy trong chế độ xem mô phỏng:

- Để hiển thị văn bản tùy ý trong chế độ xem mô phỏng, cách dễ nhất là đặt một phần tử poi với kênh alpha 0 (ẩn), đặt thuộc tính loại của nó thành văn bản bạn muốn hiển thị và tải cài đặt gui hiển thị các loại poi theo màu và kích thước mong muốn.

Để sử dụng kích thước hoặc màu văn bản khác, sử dụng tham số poi và chọn tham số đó bên cạnh hộp "show poi text param".

Thủ thuật tương tự có thể dùng đối với Polygons.

- Bạn cũng có thể sử dụng **traci.simulation.writeMessage** để đặt các tin nhắn tùy chỉnh ở dưới cùng của cửa sổ tin nhắn.

6.2. Hiển thị logo trong chế độ mô phỏng:

- Các hình ảnh nền (decal) hỗ trợ thuộc tính **screenRelative** để đặt một đối tượng liên quan đến màn hình chứ không phải mạng. Điều này cho phép đặt logo ở một vị trí cố định.

6.3. Hiển thị các tuyến đường vào các thông tin tuyến đường:

Để hiển thị các tuyến đường của một chiếc xe trong mô phỏng, nhấp chuột phải và chọn *Show Current Route or show all Routes*.

- Để hiển thị duy nhất phần còn lại của tuyến đường hiện tại, chọn *Show Future Route*.
- Các điểm dừng sắp tới và thời gian dự kiến (hoặc điều kiện kích hoạt) sẽ tự động hiển thị dọc theo tuyến đường. Để hiển thị duy nhất vòng tiếp theo của tuyến đường tắt phần *show looped route*
- Việc lùi hướng của các phương tiện đường sắt sẽ được hiển thị dọc theo tuyến đường với dòng text 'reverse' cùng một chỉ số
- Để hiển thị chỉ số của từng edge dọc theo tuyến đường, có thể kích hoạt *show route index*
- Để hiển thị tuyến đường của một người, nhấp chuột phải và chọn *Show Current Route*.
- Để hiển thị quỹ đạo trên đường đi bộ, hãy chọn *Show Walkingarea Path*
- Để đánh dấu một tập hợp các edge trong mô phỏng, tạo một tập *selection* và *color edges by selection*
- Để trực quan hóa tất cả các tuyến đường trong tập tuyến đường, sử dụng công cụ **Tools/Routes#route2poly.py** hoặc **Tools/Routes#route2sel.py**
- Để trực quan hóa số lượng xe đến hoặc khởi hành, sử dụng công cụ **Tools/Routes\countEdgeUsage**.

6.4. Điều tra các internal lane và internal edge:

- Internal Edge xác định các chuyển động qua giao lộ. Chúng được mặc định ẩn sau hình dạng của đường giao nhau để mang lại hình ảnh đẹp hơn. Cần phải hiểu internal lane nào tương ứng với chuyển động nào vì ID của chúng có thể được tham chiếu trong đầu ra hoặc trong thông báo lỗi. Các cài đặt trực quan hóa:

- Các cài đặt mối nối
 - . *disable draw junction shape* (vô hiệu hóa hình dạng đường giao nhau)
 - . *show internal edge name* (hiển thị tên cạnh bên trong)
 - . *show internal junction name* (hiển thị tên đường giao nhau bên trong)
 - . *show link junction index*

TÀI LIỆU THAM KHẢO:

- <https://sumo.dlr.de/docs/sumo-gui.html>