

User guide

Patrolling Game

Antoine Murat & Conor Ryan

5th June, 2017

Setup

In order to run the application, you will need to install several packages.

We will consider that you're running a Debian-based Linux distribution with aptitude.

Installing Java

Our software is running a Jason simulation. As Jason is built on top of Java, we need to install Java (≥ 5).

Run the following command in a terminal.

```
sudo apt-get install default-jre
```

Java path: Be sure that the java executable is in your path or add it if it's not.

Installing NodeJs

Although you can already find a NodeJs package in Debian or Ubuntu repositories, those versions are outdated. This application has been developed under node 7.10.0, but you shouldn't have any problem with any version that is $\geq 6.00.0$.

We first need to add NodeJs 7.x to the ppa along a needed python package.

```
sudo apt-get install curl python-software-properties  
curl -sL https://deb.nodesource.com/setup_7.x | sudo bash -
```


Then, install latest 7.x version by installing the nodejs package.

```
sudo apt-get install nodejs
```

Downloading the application

Clone the application directly from GitHub. You will need to install git first.

```
sudo apt-get install git
```



Then, go to the directory in which you want the server the application to be downloaded and clone the repository.

```
git clone https://github.com/IntelligentDistributedSystems/Antoine-Conor-patrolling
cd patrolling
```

Node packages: By cloning the Github repository, the server should be ready to use with all the packages installed. Nevertheless, you can update them or remove dev dependencies using npm.

Configure ports

By default, the application uses ports 8082 and 8083. You can change those ports by editing the *app.js* file.

```
cd web-server
vim app.js

# Edit those lines
const httpPort = 8082
const webSocketsPort = 8083
```

Ports & Internet: If you want to host the server for people outside of your network, be sure to setup a NAT redirection from your modem to the machine hosting the server on BOTH http and web-sockets ports.

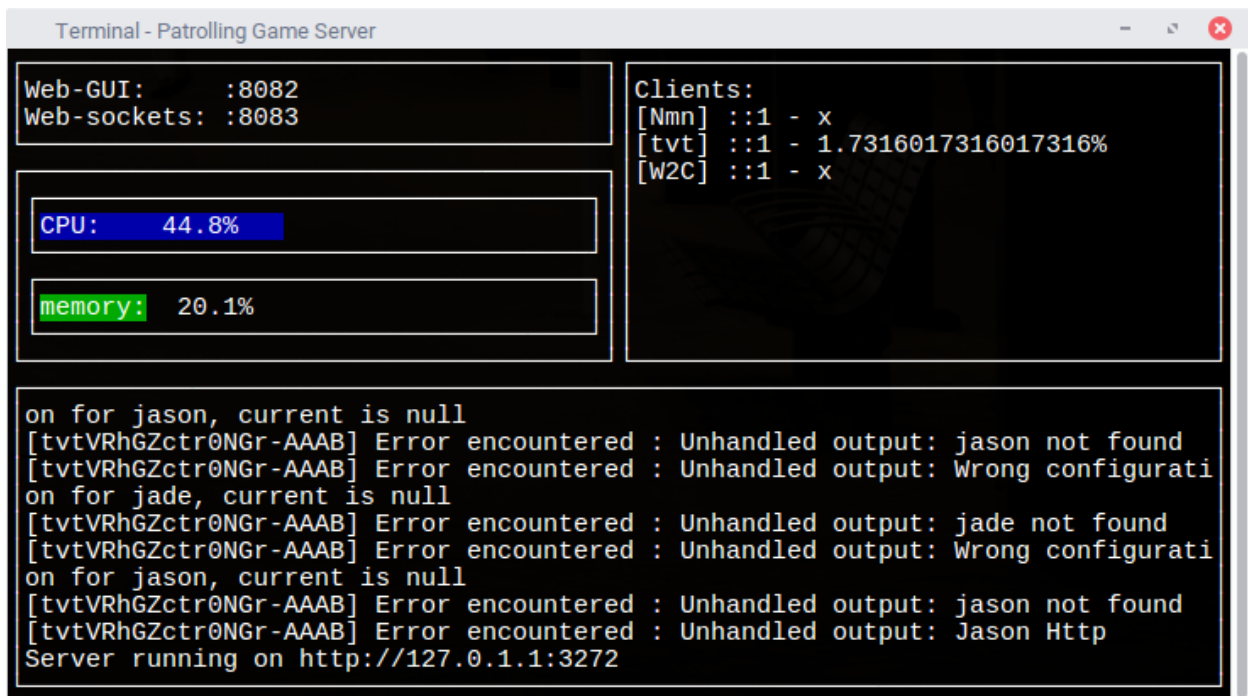
Server-side usage

Starting the server

Assuming you are on the cloned repository directory (patrolling), enter the following commands.

```
cd web-server
node app
```

Reading the server terminal



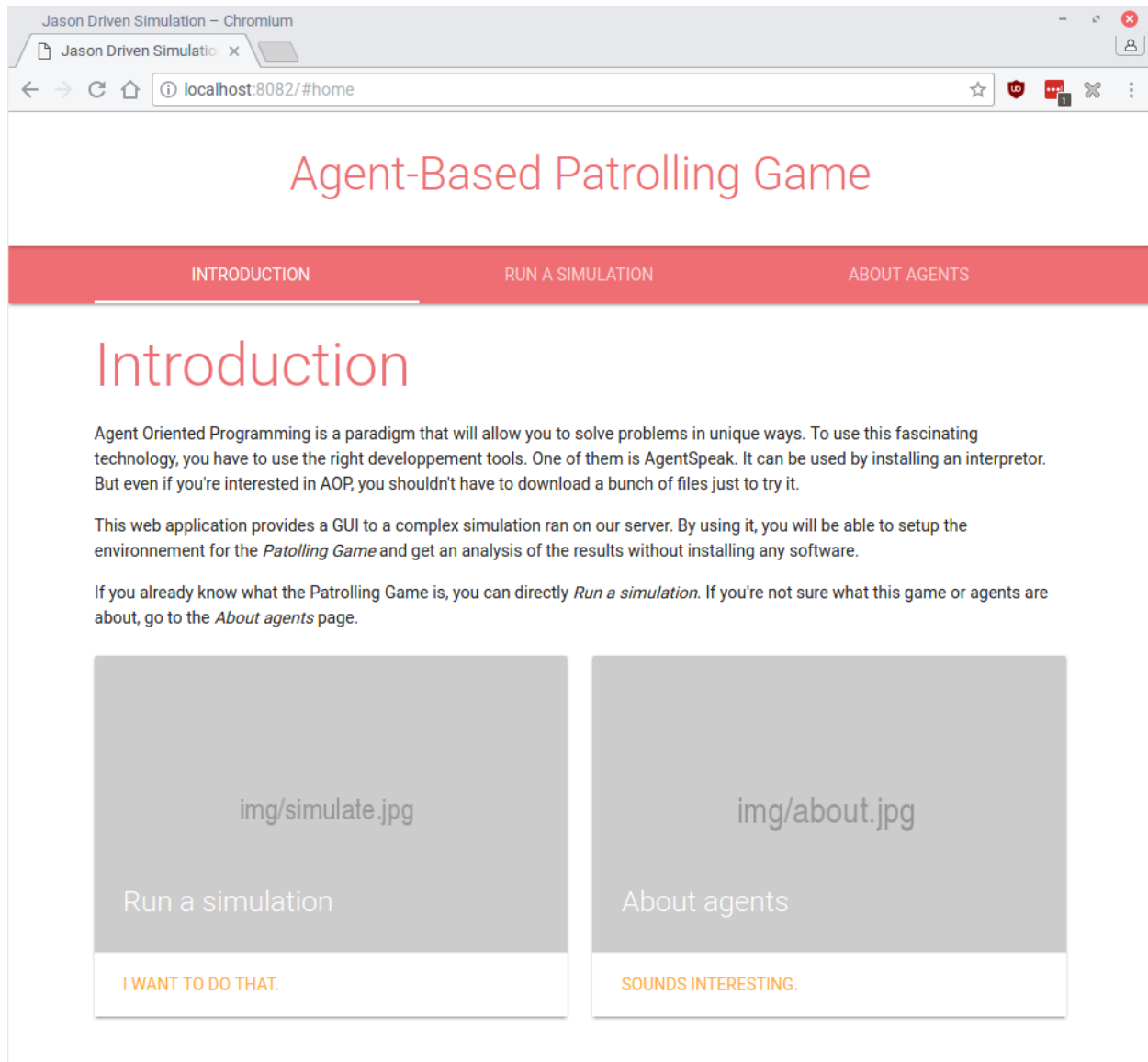
Console with 3 connexions and 1 simulation running.

1. In the top-left corner you will find the ports on which the servers are running.
2. Under it, you will find resources usage (for the whole OS, not only the Patrolling Game).
3. In the top-right corner you will find the list of the connected clients.
Each client line follow the following syntax : [ID] IP address - progress of the simulation.
A progress of "x" means that no simulation is running for this given client.
4. In the bottom, you will find regular logs. If the log has to do with a specific client, you will find its full ID at the beginning of the line.

Using the web app

Open your favorite web browser and browse `http://localhost:8082`.

You should obviously replace `localhost` by whatever the IP address of the server is.

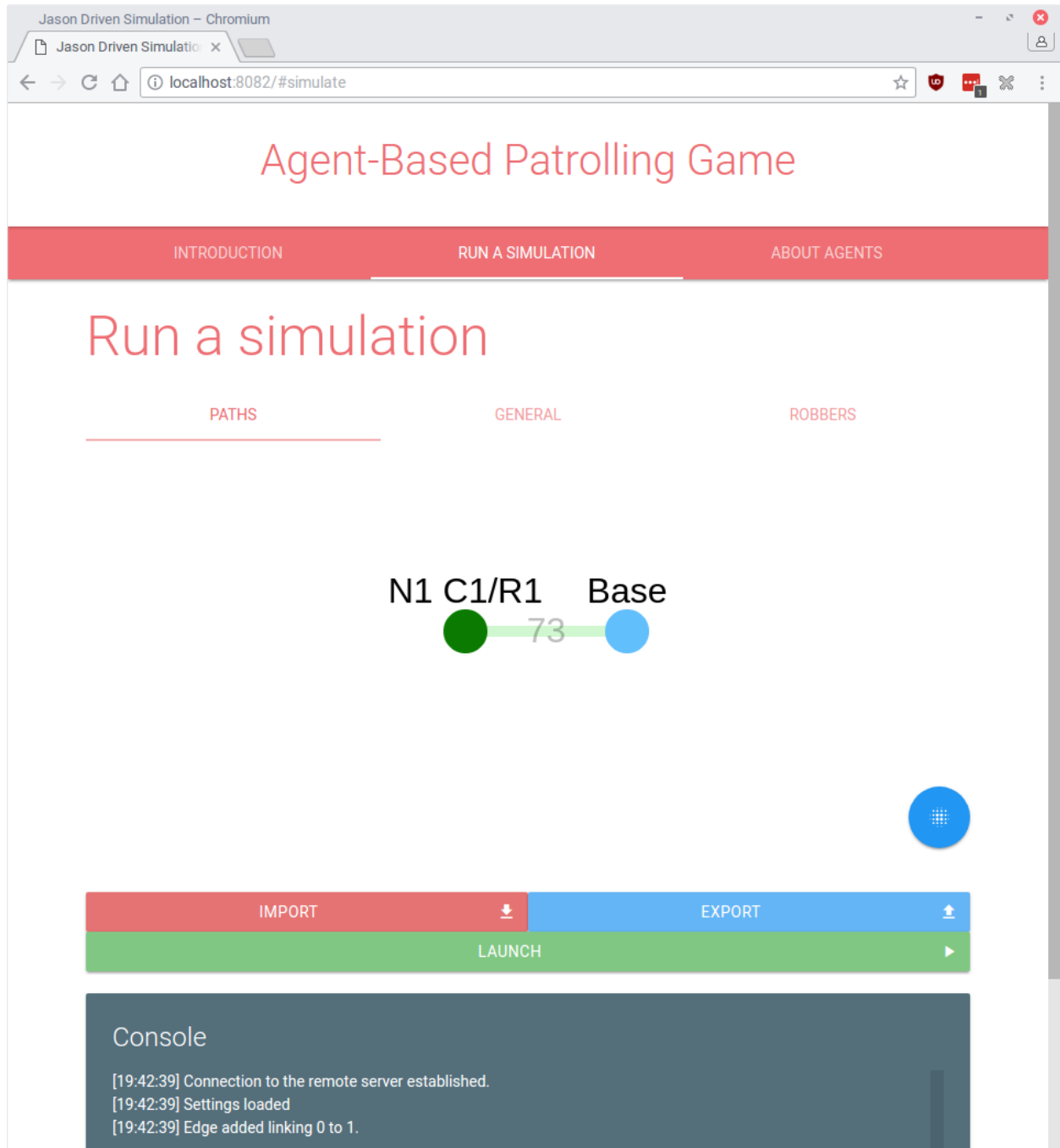


Home page.

Configuring the environment

Open the “Run a simulation” tab.

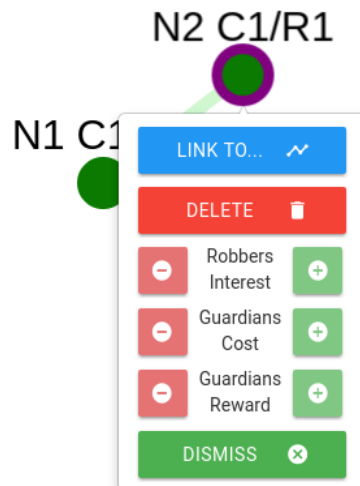
Paths settings



“Run a simulation” tab.

You can add new targets by clicking on the graph. By clicking on an existing target, you will open a menu that will let you link it to another target or set the target parameters that are:

1. Robbers interest, linked to the color of the target (from green to red),
2. Guardians cost (C),
3. Guardians reward (R).



Target menu open.

You can also drag and drop the targets to move them and modifying the distance between each of them.

Preventing the mess: You can sort the targets by clicking the blue button in the bottom right. Be aware that all distances will be lost.

General settings

PATHS	GENERAL	ROBBERS
<hr/>		
Number of iterations		
20		
<hr/>		
Guardian speed		
200		
<hr/>		

General settings tab.

You can modify the number of iterations ran for each strategy. The bigger it is, the more accurate the result will be but also the longer the simulation will take.

You can also modify the speed of the guardian. The bigger it is, the less the distances between nodes is taken into account.

About speed: You are advised to run the simulation multiple times until you find the best speed. Setting a “near c” speed will ruin the simulation making every strategy that goes through all the targets as efficient (having the same utility).

Robbers settings

The interface shows two tabs: PATHS and GENERAL. The ROBBER tab is active, displaying two robber configurations. Robber 1 has a discretion of 0,5. Robber 2 has a discretion of 0,65. Each robber has buttons for DISCRETION, REWARDS, and DELETE. A blue plus button is located at the bottom right.

Robber	Discretion	Buttons
Robber 1	0,5	DISCRETION, REWARDS, DELETE
Robber 2	0,65	DISCRETION, REWARDS, DELETE

Robber settings tab.

You can add a new robber type by clicking the blue “plus” button.

You can set each robber discretion between 0 (will always get caught) and 1 (will never get caught).

You can set cost and rewards for each robber type by clicking on the “rewards” button.

The modal titled "Robber 2 configuration" displays a table with three columns: Target ID, Cost, and Reward. The table has two rows of data. A "CLOSE" button is located at the bottom right.

Target ID	Cost	Reward
1	1	1
2	1	1

Robber configuration modal.

Exporting and import

You can save the current settings as a *.json* file that you will download. To do so, click on the “export” button.

Then, you will be able to import the settings contained in a file by clicking on the “import” button.



Files related buttons.

Interpreting the results

Launch the simulation by clicking on the “launch” button.



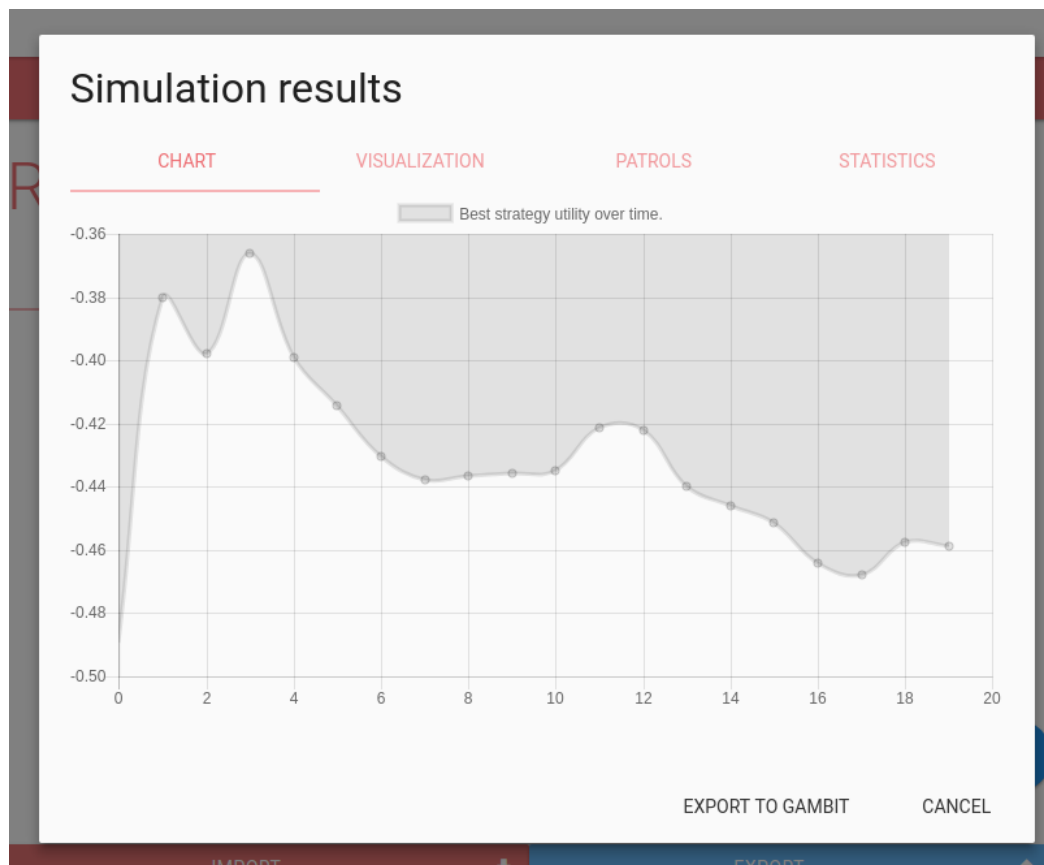
The launch button.

A modal will then open asking you to wait for the simulation to be calculated on the server-side.

Utility evolution overview

On this first tab, you can see how accurately the utility of the best strategy was calculated, iteration after iteration.

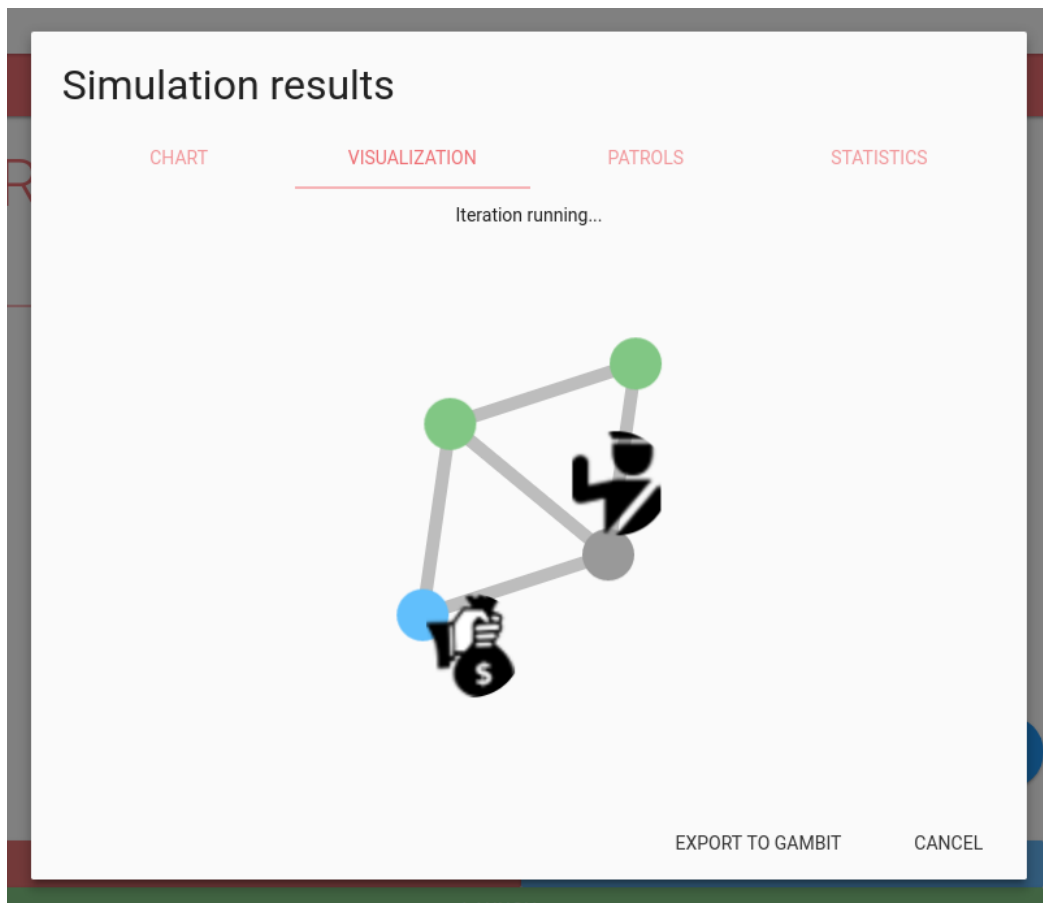
If the curb is still oscillating at the end of the simulation, you should increase the number of iterations and launch the simulation again to get better results.



We definitely need more iterations.

Live visualization

You can imagine what would happen “in real life” by watching the scene in the “visualization” tab. The guardian will use its best strategy while the robber will follow its interests based on the targets settings.

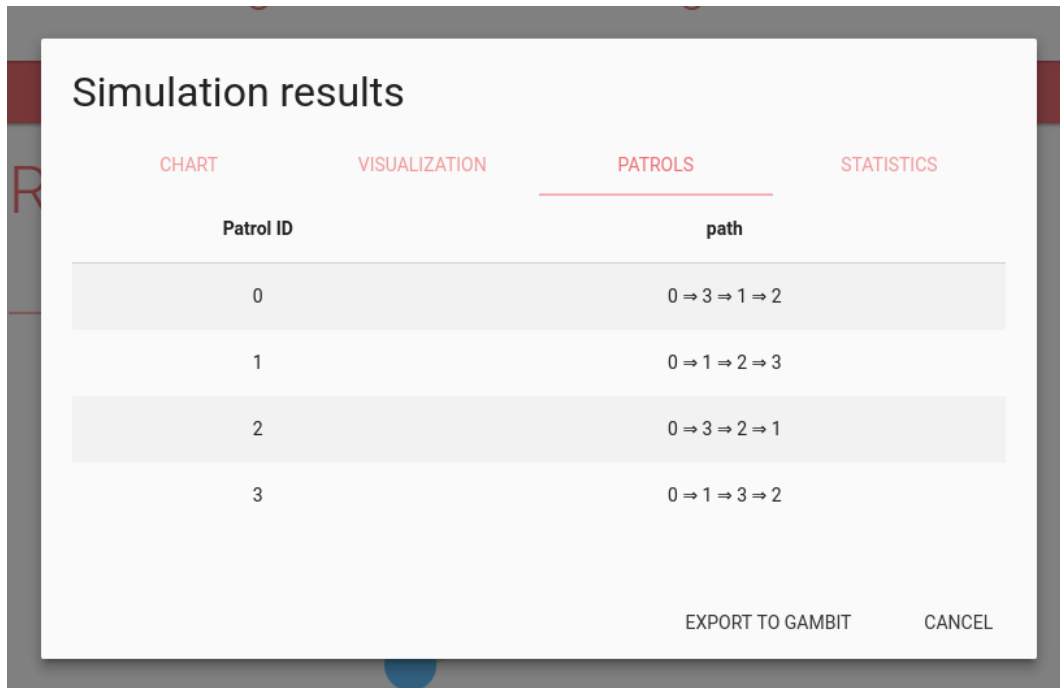


The guardian already secured 2 targets and is about to secure the last one... the robber will probably get caught.

About precision: You are not supposed to use live simulations to make any calculations as they are nowhere as precise as the simulations ran on our server.

Patrols list

A guardian strategy is a probabilistic distribution over the different patrols he can do. Our server doesn't do the calculations for every single paths but only for the ones that make sense. You can find their list in the "patrols" tab.



The screenshot shows a 'Simulation results' dialog box with four tabs: CHART, VISUALIZATION, PATROLS, and STATISTICS. The PATROLS tab is selected and contains a table with two columns: 'Patrol ID' and 'path'. The table lists four patrol paths. At the bottom right of the dialog are two buttons: 'EXPORT TO GAMBIT' and 'CANCEL'.

Patrol ID	path
0	0 ⇒ 3 ⇒ 1 ⇒ 2
1	0 ⇒ 1 ⇒ 2 ⇒ 3
2	0 ⇒ 3 ⇒ 2 ⇒ 1
3	0 ⇒ 1 ⇒ 3 ⇒ 2

Here we have 4 different patrols.

Strategies list

You will find in the last tab (“statistics”) the comparison of each guardian strategy. Each strategy is defined by a probabilistic distribution over the patrols you could find in the “patrols” tab. They are sorted by guardian utility.



CHART	VISUALIZATION	PATROLS	STATISTICS
Probabilities		Guardian utility	Robber utility
0.00 0.50 0.00 0.50		-0.4590	0.3125
0.25 0.00 0.75 0.00		-0.4688	0.2696
0.50 0.25 0.00 0.25		-0.4707	0.2970
0.25 0.75 0.00 0.00		-0.4731	0.3211
0.50 0.00 0.25 0.25		-0.4784	0.3581
0.75 0.25 0.00 0.00		-0.4798	0.3378
0.00 1.00 0.00 0.00		-0.4863	0.2987
0.00 0.75 0.25 0.00		-0.4892	0.3768
0.50 0.25 0.25 0.00		-0.4922	0.3393
0.00 0.50 0.50 0.00		-0.4933	0.3584

Although we only have 3 target, we have numerous strategies.

Export to gambit

Our application does a good job at finding the optimal guardian strategy assuming the robbers won't change their strategy. But this might not be really accurate and we might need to use formal mathematics instead of simulations. To do so, we can calculate the Nash equilibrium. Although our app doesn't do those calculations, it is really easy to export the data to the gambit file format.

Simply click the "Export to gambit" button in the bottom of the results modal to obtain the *.nfg* game file. Assuming you installed both gambit and its python interface, run python and then copy and paste the following lines.

```
python
```

```
import gambit
g = gambit.Game.read_game("file_name.nfg")
solver = gambit.nash.ExternalEnumMixedSolver()
solver.solve(g)
```

Not only for python: You can also import the file using the gambit GUI.