

REBELLION MODEL DESCRIPTION

GRACE I. LIN - GSOC 2011

CONTENTS

1. Model Description	1
2. Files and GUI	2
3. Model Parameters	3
4. References	4

1. MODEL DESCRIPTION

The rebellion model (Figure 1) simulates how a group of people rebels against the authority (cops). Each person is in one of the three states:

- **active:** red color; person decides to rebel (see calculation below for becoming an active person)
- **jailed:** black color; active person arrested by cops; remain in jail until jail term expires
- **quiet:** green color; neither active nor jailed

The cops (blue color) will randomly arrest active people within the visibility area. To become an active (rebel), the person calculates his/her level of grievance and arrest probability:

$$\begin{aligned}grievance &= perceived_hardship \times (1 - government_legitimacy) \\ arrest_probability &= 1 - \exp \left(-k \left[\frac{num_of_cops}{num_of_active_people} \right] \right) \\ rebel \text{ if } (grievance - risk_aversion \times arrest_probability) &> threshold\end{aligned}$$

where

$perceived_hardship = 1.0$ (fixed for lifetime)
 $government_legitimacy =$ between 0 and 1; user specified
 $k =$ constant for one cop and one person within vision = 2.3
 $risk_aversion = 1.0$ (fixed for lifetime)
 $threshold = 0.1$

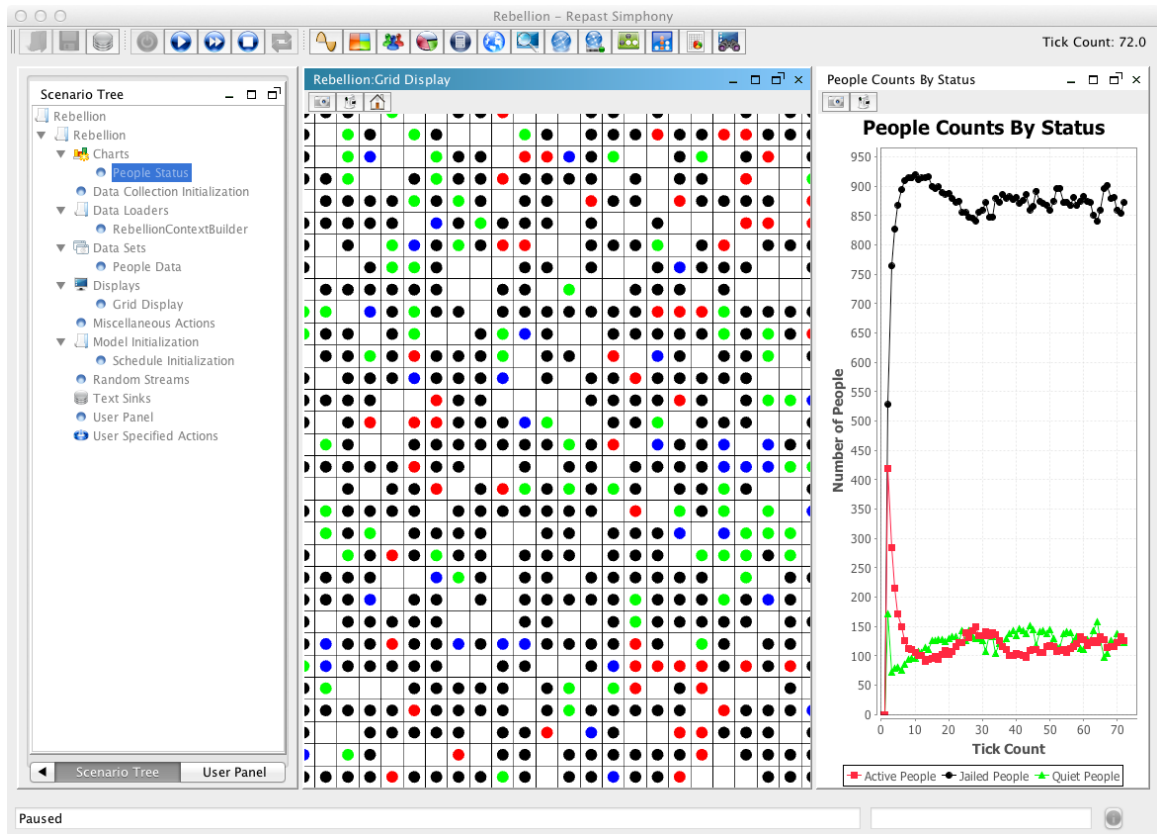


FIGURE 1. Rebellion Model

2. FILES AND GUI

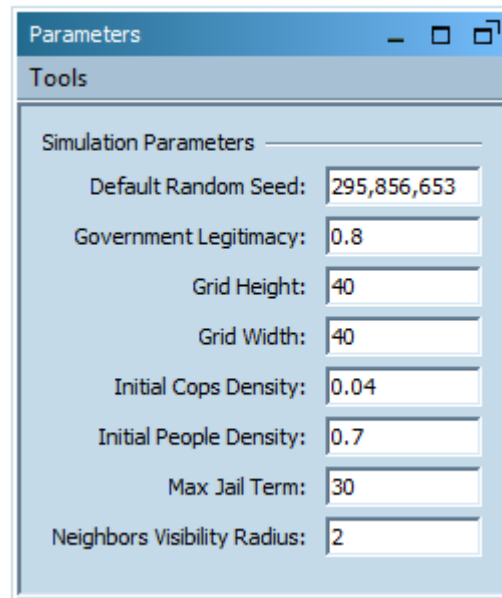
Source files and description:

src/rebellion/**Constants.java** : list of constants
 src/rebellion/**Cop.java** : class for cop
 src/rebellion/**CoverageCounter.java** : counts number of people in different states;
 used for data sets and charts in GUI Scenario Tree
 src/rebellion/**Person.java** : class for person
 src/rebellion/**RebellionContextBuilder.java** : set up environment initially
 src/rebellion/**SMUtils.java** : miscellaneous functions
 src/rebellion.observer/**PersonStyleOLG2D.java** : color assignment for cop/per-
 son

Remember to setup the grid and space projections in the resource folder "Rebellion.rs": in context node, add projection for type=continuous space and id=space, and add projection for type=grid and id=grid. The id's can be found in the file Constants.java. Once we run the Rebellion model, set up the Scenario Tree as follow:

- **Data Loaders** : Use RebellionContextBuilder
- **Display**: Projection with grid, agents are Person and Cop, and agent style is rebellion.observer.PersonStyleOLG2D for both Person and Cop
- **Data Sets**: A single aggregate data set with data sources for active, jailed and quiet person counts. The CoverageCounter class provides these values.
- **Charts**: One that displays the person counts using the single data set.

3. MODEL PARAMETERS



Simulation Parameters	
Default Random Seed:	295,856,653
Government Legitimacy:	0.8
Grid Height:	40
Grid Width:	40
Initial Cops Density:	0.04
Initial People Density:	0.7
Max Jail Term:	30
Neighbors Visibility Radius:	2

FIGURE 2. Rebellion Model - Parameters

User-specified parameters are shown in Figure 2 with descriptions as follow:

- **Government Legitimacy**: Number of rows.
- **Grid Height**: Number of columns.
- **Grid Width**: Number of rows.
- **Initial Cops Density**: Percentage of grid size populated by cops.
- **Initial People Density**: Percentage of grid size populated by people.
- **Max Jail Term**: Maximum turns for people who are jailed.

- **Neighbors Visibility Radius:** Radius of area that the agent can see.

4. REFERENCES

- NetLogo's termites model: <http://ccl.northwestern.edu/netlogo/models/Rebellion>
- Modeling civil violence: An agent-based computational approach (<http://www.pnas.org/content/99>,