

Program execution method:

MainProgram.java, right-click to run as Application, after the pop-up interface, execute the run button at the top of the interface, and the program will run automatically.

1. Directory structure

disease is the class of the main running program

The programs given by DiseaseSim's partners are mainly placed inside, and some programs given by partners have been modified

util: tool class, currently only the class TextExport.java that reads configuration files

conf.txt : Configuration file, parameters are configured in this file, each parameter name and parameter value are separated by =.

2. Procedure

MainProgram.java : The main program, the execution entry, this class is the core program, in which the message queue, the class that runs the UI, and some shared variables are defined. Among them, DiseaseGUI.java is the interface framework class. The AgentView class can be understood as the panel displayed by the interface. The AgentNodeView is the grids in the panel, and the properties of this class determine the content displayed by the grids and the style of the display. StimulationEvent and StimulationEventListener are classes for event association when the RUN button is clicked, so there is no need to study them in depth here. When RUN is clicked, the run method in RunListener is triggered and executed. The run method calls the startSimulate method in the MainProgram class and starts the GUI update interface process. Then start to initialize AGENT. After initializing AGENT, since each AGENT inherits runable, each AGENT is an automatic thread. When each AGENT thread starts, it will send messages to the message queue according to the running situation.

The GUI update interface process, the entry is at the startUI call in the startSimulate method in MainProgram.java, the call mainly starts the startUI() method in DiseaseGui.java, which instantiates an AgentDisplay thread, which loops from the message queue DiseaseGUI After the AGENT update message is obtained in .msg\_que, the agent on the synchronous update interface is displayed.

AgentManager follows the idea of ​​a partner, that is, randomly generates nodes, finds neighbors, and initializes node objects.

Note: Because the run method in the Agent class executes faster, in this method, the code of Thread.sleep(2000) is added, so that the node changes of the interface can be observed. If it is not added, the interface may not see the effect.