

Exercise 3 (Individual Exercise, Pass-Fail)
DUE: Tuesday, February 17th, 11:00am

Leigh Tesfatsion
Econ 308, Spring 2009

**** PLEASE NOTE: NO LATE ASSIGNMENTS WILL BE ACCEPTED - NO EXCEPTIONS!**

Hands-On Introduction to Agent-Based Modeling (ABM)

Basic References:

- 1 S. Railsback, S. Lytinen, and S. Jackson, **StupidModel: A Template Model for ABM Platforms**
 - 1.A * StupidModel Homepage
<http://condor.depaul.edu/~slytinen/abm/StupidModel/>
 - 1.B * **Agent-Based Simulation Platforms: Review and Development Recommendations**, *Simulation*, Vol. 82, No. 9, 2006.
<http://www.humboldt.edu/~ecomodel/documents/ABMPlatformReview.pdf>
 - 1.C ** **StupidModel Formulation**
<http://condor.depaul.edu/~slytinen/abm/StupidModelFormulation.pdf>
 - 1.D ** **StupidModel Downloads (NetLogo, Repast, Mason)**
<http://condor.depaul.edu/~slytinen/abm/StupidModel/>
- 2 * W. Rand, **Agent-Based Modeling Platforms: A Practical Introduction**
<http://www.econ.iastate.edu/classes/econ308/tesfatsion/ABMPlatDemo.BRand2007.pdf>
- 3 * M. Neugart, **Setting Up Repast J Projects in the Eclipse Integrated Development Environment**,
www.econ.iastate.edu/tesfatsi/SettingUpRepastJProjectsInEclipse.MNeugart2009.pdf.
(Note: This reference is for those choosing the Repast J ABM Toolkit for Ex 3)
- 4 * L. Tesfatsion, **Repast: A Software Toolkit for Agent-Based Social Science Modeling**. Topics covered included: Intro to CAS/ABM/ACE, Introduction to Java, Getting Acquainted with Repast J (Repast based on a Java development environment), and Programming with Repast J.
www.econ.iastate.edu/tesfatsi/repastsg.htm
(Note: This reference is for those choosing the Repast J ABM Toolkit for Ex 3)

EXERCISE PURPOSE:

The purpose of this pass-fail individual exercise is to give each of you a chance to acquire a hands-on introduction to one of the three major agent-based modeling (ABM) toolkits:

NetLogo; Repast J; or MASON. Subsequent exercises will build on this exercise. **On the due date please bring to class one copy of your exercise answer to hand in and one copy to keep for class discussion.**

EXERCISE OVERVIEW:

Railsback et al. [1] have developed a template model called “StupidModel” (very badly named!) to help people acquire facility with agent-based modeling. StupidModel includes many commonly used features of agent-based modeling (ABM) platforms.

The authors implement StupidModel using five different platforms. For the purposes of this exercise, however, attention will be focused on only three of these platforms: NetLogo (a self-sufficient ABM platform); Repast J (based on Java); and MASON (based on Java).

Sixteen successive versions of StupidModel are implemented for each of these platforms. These versions begin with a bare bones model (version 1) and progress in small incremental coding steps to a relatively sophisticated model (version 16) that involves two agent types, a full agent life cycle (birth, reproduction, predation, and death), and a habitat with data read from an input file.

In Ref.[1.B] the authors review and compare the different ABM platforms. In Ref.[1.C] the authors provide a concise description of the basic StupidModel formulation that takes the reader step by step through the multiple model versions. Links for downloading NetLogo, Repast J, and MASON can be found at site [1.D] along with source code (in zip files) and implementation notes for the implementation of StupidModel in each of these platforms.

Important Remark on NetLogo Implementation Notes:

The implementation notes for NetLogo are included in the NetLogo source code files. Also, the zipped source code files for NetLogo only contain versions 1-3 and 5-16; a “version 4” – introduction of probes for viewing characteristics of bugs and cells – is not included because the graphical user interface (GUI) for NetLogo automatically provides these capabilities.

TIPS REGARDING YOUR CHOICE OF ABM TOOLKIT:

NetLogo (<http://ccl.northwestern.edu/netlogo/>) is a stand-alone ABM toolkit that includes its own Integrated Development Environment (IDE) for developing, editing, compiling, and running programs. NetLogo is highly recommended for those with little or no prior experience with (or interest in) Java programming. Three special features of NetLogo are: (a) while based on Java, NetLogo uses its own special syntax; (b) NetLogo is not open source; and (c) a trade-off has been made between ease of use and programming power/flexibility (in favor of ease of use).

Repast J (latest version 3.1, http://repast.sourceforge.net/repast_3/download.html) is *not* a stand-alone ABM toolkit. It uses standard Java syntax and functions, and it requires a prior downloading and installation of a Java compiler, such as the freely available Java Development Kit (JDK) from Sun Microsystems ([Java.sun.com](http://java.sun.com)). Repast J is open source.

Repast J is recommended for those with at least some prior experience with Java who are interested in working within a Java programming environment. Using Repast J from within a Java IDE (e.g., the freely available NetBeans or Eclipse IDEs) is strongly recommended. If you do not have much (or any) prior exposure to Java programming, but you wish to choose Repast J as your ABM toolkit (e.g., as a way of learning Java), please see me for additional tips for setting up and running the Repast J version of StupidModel in an IDE and/or consult Refs.[3-4].

MASON (<http://cs.gmu.edu/~eclab/projects/mason/>) is the most recently developed and least documented of the three ABM toolkits. It is recommended for those who have a strong Java programming background and who are interested in acquiring skills with an ABM toolkit that stresses large-scale systems, efficient performance (over code readability), and 3D visualization capabilities. Like Repast J, MASON requires a prior downloading and installation of a Java compiler, and use of MASON within a Java IDE is strongly recommended. MASON is open source.

EXERCISE QUESTIONS:

Part A: Using references [1] and [2] for guidance, choose one of the three ABM toolkits NetLogo, Repast J, or MASON. Download and install your chosen toolkit from site [1.D] along with the corresponding source code for StupidModel.

Part B: Try to work through all sixteen versions of the StupidModel for your chosen toolkit, step by step. (As noted above, Version 4 is not present for NetLogo so only Versions 1-3 and 5-16 need to be worked through.) Here is a recommended approach:

1. Either print out the source code for each version or copy these files into editable text files.
2. Go step by step through the successive versions and highlight the CHANGES in code from one version to the next. Try to deduce from [1.C] and the platform-specific implementation notes exactly what the additional code is supposed to do in each successive StupidModel version.
3. Starting with a new file, type (or paste in) the new code for each successive version, run it, and try to verify that the code does what the implementation notes assert it will do. Play around with variations in the code to see what effects this has on the outcomes. (For example, try changing agent colors, numbers of agents, move rules, etc.)

Part C: Briefly but carefully discuss your experiences in carrying out Parts A and B above. In particular, was it straightforward? Did the code do what was claimed? Was it easy to understand? Were you able to master the code sufficiently to be able to introduce systematic changes in modeling assumptions? Was the documentation helpful? And so forth.