



AGENT BASED PROGRAMMING

PROJECT WORKSHEET

NAME: _____

INTRODUCTION:

NetLogo is a cross-platform multi-agent programmable modeling environment. NetLogo is free of charge and can be downloaded here: <http://ccl.northwestern.edu/netlogo/download.shtml>

We are using Netlogo to introduce the concepts behind computer agents, and to explore how simulations are created and how they can be used. The complete programming manual for NetLogo can be found here: <http://ccl.northwestern.edu/netlogo/docs/> ([alt link](#))

A shorter easier to user QuickStart Guide can be found here:
<http://ccl.northwestern.edu/netlogo/resources/NetLogo-4-0-QuickGuide.pdf> ([alt link](#))

This worksheet is designed to help you ask the questions necessary to begin work on your own project model. It can also guide you through the process of creating a model.

PART 1: UNDERSTAND THE PROBLEM

You will first need to pick a problem or situation that you are interested in modeling. Some ideas include:

1. Model a chemical reaction.
2. Model an ecosystem.
3. Model a planetary system.
4. Model a famous battle (fun with 3D settings).
5. Model a social simulation:
 - a. Spacing in an elevator (how people spread out as they get on an elevator).
 - b. Spread of a virus (Zombie Outbreak)
6. Create a simulation of any of the 3 "RoboCup junior challenges".
 - a. Have a group of robots (turtles) playing soccer
 - b. Have a robot follow a line or navigate a maze
 - c. Have a group of robots make music, draw a picture, etc...
7. Create a game (that uses lots of agents).

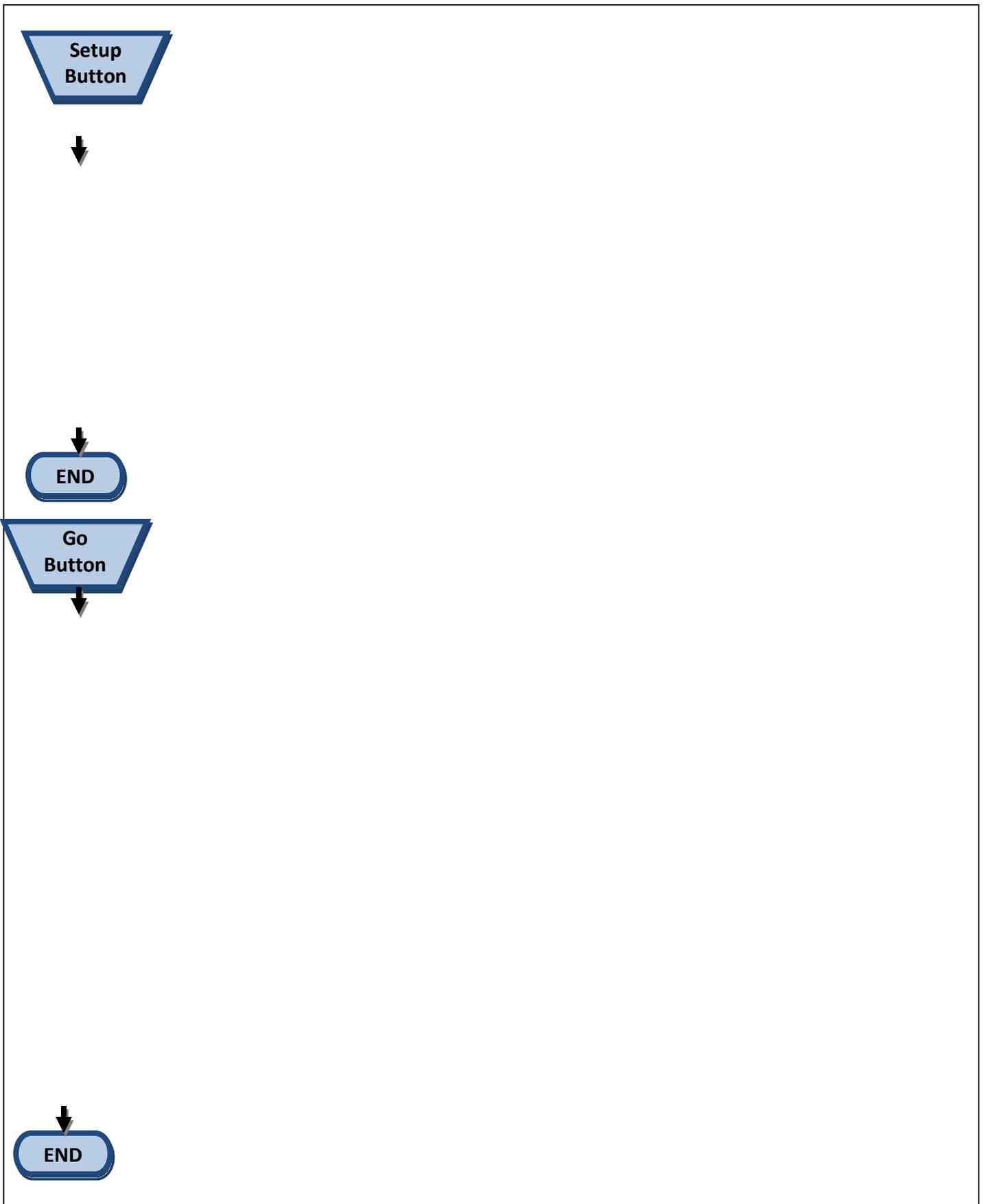


PART 2: PLANNING THE MODEL

Never start a NetLogo model by sitting down and coding. **Every** large programming project needs to be planned out, in as much detail as possible, BEFORE you start coding. The more time, effort and detail you put into your planning stage, the faster the coding will actually go.

Before you begin any model (including your project model) you should do the following:

- 1) Form hypothesis:
 - a) What is the question the model is designed to explore?
 - b) What do you think the model will demonstrate?
- 2) Identify model requirements:
 - a) What details from the system are essential; what details can be ignored?
 - b) What is the environment?
 - i) What will the "view" (the world) look like?
 - ii) What assumptions are built into the world? (Example: no friction, air resistance)
- 3) Who are the agents?
 - i) What variables will each agent require?
 - ii) What characteristics (behaviors) will each agent display?
 - iii) What are the limitations (rules) that the agents must obey?
 - iv) What are the "inter" and "intra" agent relationships?
- 4) Create Flowchart (pseudo-code model) of your program (next page):





PART 3: PROGRAMMING YOUR MODEL

Once you have detailed what your program will do, answered all the questions in the previous section and created a flow chart program for your model, you are ready to start coding.

1. Get a copy of the Project Template ([link](#))
2. Label your project (using comments) in the code at the top of the procedures tab.
 - a. You may also wish to fill in the fields on the information tab (but not for this lab).
3. Add any extensions you are going to need to the top of your code in the procedures tab:
 - a. Example: `extensions [sound] ;; Add the ability to use sounds`
4. Create breeds (if needed).
 - a. Breeds are turtle (or patch) sub-groups that have unique names.
5. Adjust world (screen) settings.
 - a. How many patches do you need for your simulation (size of world)?
 - b. What size should your patches be?
 - c. Where will location (0, 0) be?
 - d. Torus or rectangle (does the world "wrap")?
6. Create **setup** and **go** buttons.
 - a. All standard NetLogo simulations will have these two buttons.
 - b. The setup and go buttons will call the setup and go functions (respectively).
7. Create interface controlled variables (sliders, etc..)
 - a. Some variables you will want the users to be able to change, these variables should be implemented using sliders, or input forms.
8. Create global and local variables in the procedures tab (if needed).
9. Create and fill setup procedure with setup sub-procedures.
10. Create and fill setup sub-procedures.
11. Create and fill go procedure with go sub-procedures.
12. Create and fill go sub-procedures.
13. Create plot and plot pens (in the interface).
14. Create and fill do-plots procedure.



PART 4: TESTING, EVALUATING AND EXPANDING

Once you have detailed what your program will do, answered all the questions in the previous section and created a flow chart program for your model, you are ready to start coding.

Once you have completed your model you should test it:

1. Do the agents act as expected?
2. Do the rules behave as expected?
3. Are plots and monitors reporting what you expected?

Evaluate your hypothesis using your model:

1. How does the model change with different setups?
2. What is the effect of randomness on your model?
3. How many times do you think you need to run the model to test it properly?
4. Does the model you created validate or invalidate your hypothesis?

Expand your model:

1. What can you do to make your model more realistic/accurate?