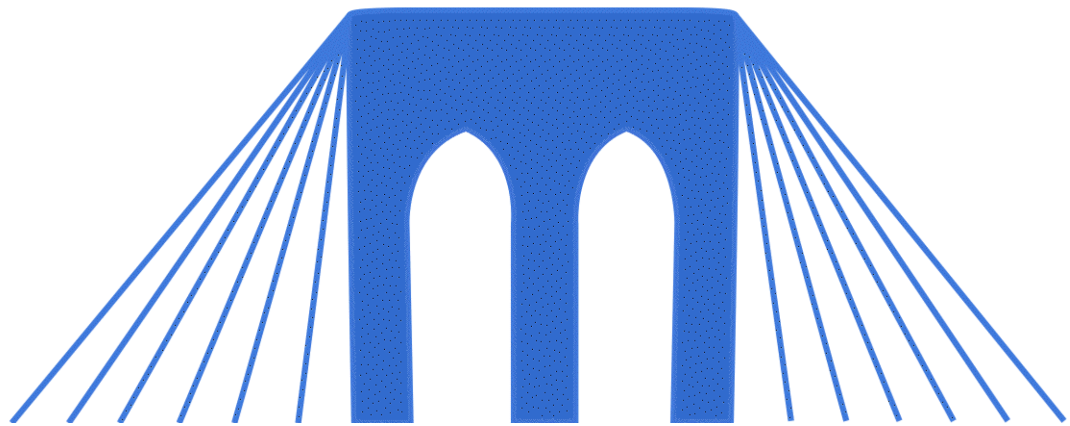
Bridges To Computing



General Information:

- This document was created for use in the "Bridges to Computing" project of Brooklyn College.
- You are invited and encouraged to use this presentation to promote computer science education in the U.S. and around the world.
- For more information about the Bridges Program, please visit our website at: http://bridges.brooklyn.cuny.edu/

Disclaimers:

- **IMAGES**: All images in this presentation were created by our Bridges to Computing staff or were found online through open access media sites and are used under the Creative Commons Attribution-Share Alike 3.0 License. If you believe an image in this presentation is in fact copyrighted material, never intended for creative commons use, please contact us at http://bridges.brooklyn.cuny.edu/ so that we can remove it from this presentation.
- **LINKS**: This document may include links to sites and documents outside of the "Bridges to Computing" domain. The Bridges Program cannot be held responsible for the content of 3rd party sources and sites.

Agent Based Programming & Simulations

Lecture 3: Models and Simulations

References

- Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, http://aima.cs.berkeley.edu/ , chpt. 2
- Stan Franklin and Art Graesser (1996); Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents; Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996
- An Introduction to Multiagent Systems, by Michael Wooldridge, John Wiley & Sons, Ltd (2002).

Content

1. Agent Based Programming and Simulations
   - Cost Savings
   - Danger Prevention
2. Agent Input and Output
   - Discrete vs. Continuous
   - The Signal to Symbol Problem
3. Creating a simulation
   1. Understanding the problem
   2. Planning the model
   3. Programming the model
   4. Testing, evaluating and expanding the simulation

Agent Based Programming and Simulations

- We discussed the two main reasons for agent based programming in our last lecture:
  1. Managing Complexity
  2. Emergent Behavior (phenomena)

- Another SIGNIFIGANT reason for using agent based programming is that it allows us to run "virtual simulations" of real world scenarios.
- Why would we want to do this?

Simulations – Cost Savings

- Biosphere2 (Biosphere 1 is Earth) was built to help explore the use of an enclosed biosphere as part of future space colonization.
- It cost $200 million to build and operate (1985-2007).
- During the 1st Biosphere experiment oxygen levels fell below safe levels.
- The reason for the oxygen drop was that the exposed concrete of the structure was absorbing gases (CO2).

Biosphere 2: Planners failed to model in all the possible variables when designing the enclosed ecosystem.

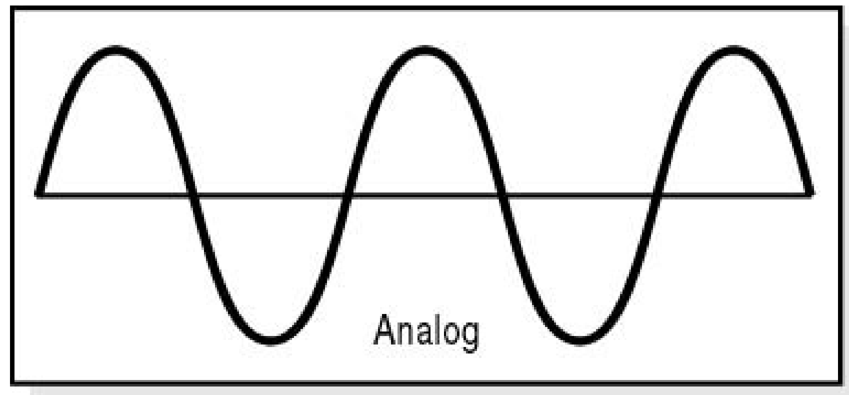
Simulations – Danger Prevention

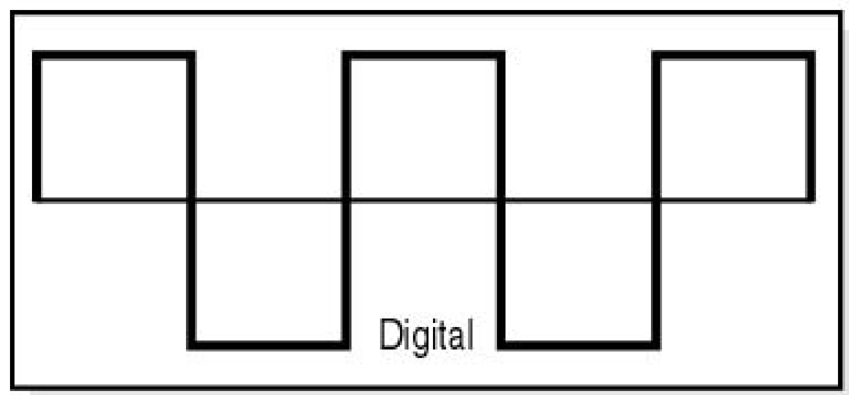
- On 21 May 1946, Louis Slotin, a researcher at Los Alamos, accidentally irradiated himself when an experiment with a sphere of plutonium took a wrong turn.
- Immediately realizing what had happened Slotin quickly disassembled the device, likely saving the lives of seven fellow scientists nearby.
- Slotin himself succumbed to radiation poisoning nine days later.

Many of the first computers and computer programs created/written were used to help calculate radiation yields for nuclear devices.

Discrete & Continuous

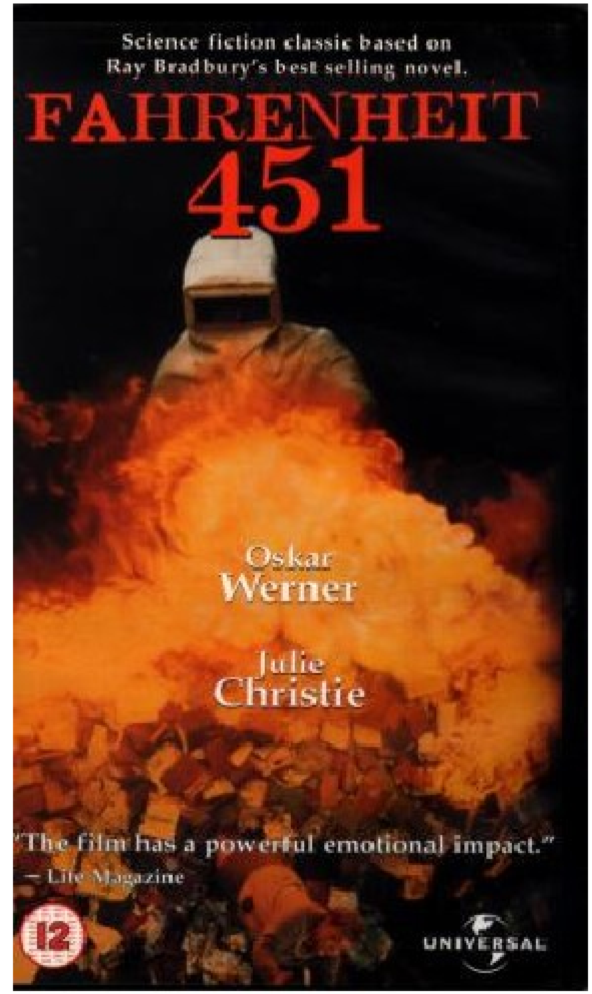
- Agents are sensing their environment. Input from sensors can come in the form of discrete (digital) and continuous (analog) signals.
- More information in signal isn't necessarily better
- More information in a signal may mean more processing will be required to make sense of that signal.





Signal to Symbol Problem

- The signal to symbol problem describes the following:
  - The output of a sensor, doesn't tell an agent what it should do.
  - Symbols are used to make information "abstract" and not "sensor-specific".
- Example:
  - Thermal sensor -> 451 degrees Fahrenheit
  - Implies that -> The library is on fire



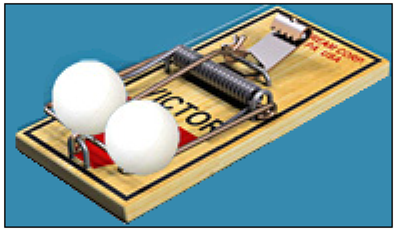
Creating a Simulation in NetLogo

- For your final project in this unit you will create your own simulation using NetLogo. You could:
  1. Model a chemical reaction (alcohol production by yeast).
  2. Model an ecosystem (termite mound)
  3. Model a planetary system (comet).
  4. Model a famous battle (fun with 3D settings).
  5. Model a social situations
     1. Spread of a virus (Zombie Outbreak)
  6. Create a simulation of any of a robot agent.
     1. Have a robot follow a line or navigate a maze
     2. Have a group of robots make music, draw a picture, etc…
  7. Create a game (that uses lots of agents).

Understanding the Problem

- Whatever simulation you decide to create, you need to make sure that you thoroughly understand (and can visualize) the situation that you are trying to model.

- Videos of/about this type of experiment:
  - http://youtu.be/0v8i4v1mieU
  - http://youtu.be/hgO1-lqzrZY
  - http://youtu.be/vjqIJW_Qr3c
  - http://youtu.be/ESpRFkXon7g

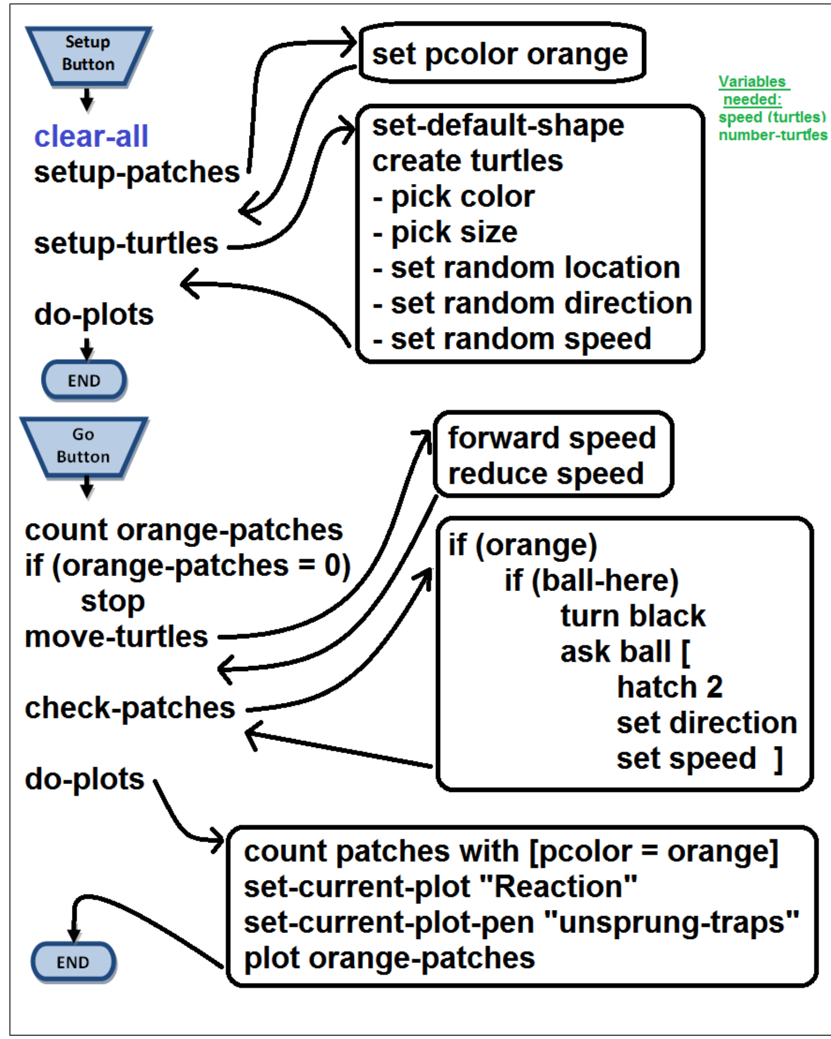


Planning the Model

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

Flowchart (PseudoCode)

1. What is the sequence of events that must happen.
2. Assume that you are going to use the setup button to setup the simulation.
3. Assume that you are going to use the go button (with the forever option selected) to run the simulation.
4. Identify any new variables needed



Programming the Model (step 1-4)

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.
   1. 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:
   1. Example: *extensions [sound]*
4. Create breeds (if needed).
   1. Breeds are turtle sub-groups that have unique names.

Programming the Model (steps 5-7)

5. Adjust world (screen) settings.
   1. How many patches do you need (size of world)?
   2. What size should your patches be?
   3. Where will location (0, 0) be?
   4. Torus or rectangle (does the world "wrap")?
6. Create **setup** and **go** buttons.
   1. Standard NetLogo simulations have these two buttons.
   2. These buttons will call the setup and go procedures.
7. Create interface controlled variables (sliders, etc..)
   1. Some variables you will want the users to be able to change.

Programming the Model (steps 8-14)

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.

Testing, Evaluating & Expanding

<u>Once you have completed your model you should test it:</u>

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

<u>Evaluate your hypothesis using your model:</u>

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 need to run the model to test it?
4. Does the model validate or invalidate your hypothesis?

<u>Expand your model:</u>

1. How can you make your model more realistic/accurate?

DON'T PANIC

- You will have several labs on NetLogo.
- Lost of help is available online:
  - Fantastic (short) NetLogo Quick Guide
  - Full Netlogo Manual
  - NetLogo Dictionary

The End