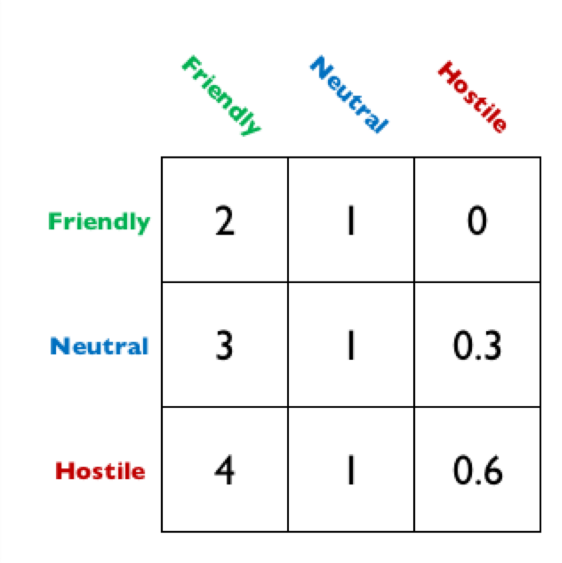
ABM – Week 8 – Seminar – LVL2

Purpose

This task will allow you to use links and to model agent interactions using game theory. Agents will interact in pairs and will grow or shrink based on their success in those interactions. The code uses the ‘matrix’ extension.

Model

Open the model international_relations_baseline.nlogo. You will develop this model to represent nations interacting periodically in pairs. The interactions will take the form of the following game, where nations may choose to be friendly, neutral or hostile (payoffs are to the row player):



	Friendly	Neutral	Hostile
Friendly	2	1	0
Neutral	3	1	0.3
Hostile	4	1	0.6

Take a couple of minutes to look at the code and take particular note of the following things:

- What are the global variables?
- What variables do the turtles have?
- How does the model define the payoff matrix?
- How has the shape of the turtles been changed?
- How have the turtles been arranged across the world?

Exercises

1. Create directed links between all the turtles. At each tick, select a link and make the turtles at either end play the game shown above. Since the nations all have the “random” strategy, they should choose their row/column at random. Based on the outcome of the game, make each turtle gain the appropriate amount of utility. Complete the set-sizes procedure so that a nation’s size is proportional to its utility.

2. Introduce a second strategy option for my-strategy, called “hostile”. Make sure that some turtles use this strategy. Turtles with this strategy should always be hostile in their interactions. Colour nations differently according to their strategy.

Step-By-Step Guide

EXERCISE ONE

Creating directed links between all turtles

Look at the Netlogo manual's section on links to see how to achieve this.

Selecting a link at random and identifying the turtles at each end

Edit the interaction procedure to achieve this. You will need to store the link and the two turtles as local variables. It will also be useful to visualise which link has been selected, so also add code to make the link increase in thickness after it is identified and return to its original thickness at the end of the procedure. Again, you should be able to find all the useful primitives and build-in variables in the 'links' sections of the Netlogo manual and dictionary.

Coding the random strategy

Create a reporter called 'random-strat', which returns a random number in the range 0, 1, 2. This reporter will choose a row at random for a turtle who is playing the game. It is convenient to create a separate reporter for this (even though it is very simple), because it will be cleaner to have a different reporter for each strategy when others are introduced.

Playing the game

Create a procedure called 'play-game', which takes two arguments to represent the two interacting nations (i.e. the arguments will be turtles). The procedure should create some local variables to represent each player's chosen row of the payoff matrix and these should be set equal to the output of the 'random-strat' reporter (all turtles have the "random" strategy at this stage). The 'play-game' procedure should be called in the 'interaction' procedure.

[Note that the 'play-game' procedure plays the game, in the sense that it makes each player choose a row (or column), but it does not yet calculate the payoff of the game to each player.]

Updating turtle utility

Create an empty procedure called 'update-utility' that takes four arguments, representing two interacting turtles and their chosen rows (or columns) of the payoff matrix. The procedure should pick out the correct payoff for each player from the payoff matrix and the turtles should each update their 'utility' by this amount. This procedure should be called in the 'play-game' procedure.

[Necessary procedure from matrix extension: matrix:get]

Making turtle size reflect utility

Edit the 'set-sizes' procedure to set the turtles' sizes in proportion to their 'utility'. Do this in such a way that the average size of all turtles remains equal to the 'average-turtle-size' global variable.

EXERCISE TWO

Coding the hostile strategy

Create a reporter for this strategy in the same form as the 'random-strat' reporter created earlier.

Setting up turtles with either the random or the hostile strategy

Edit the 'set-strategy' procedure to give turtles different strings for 'my-strategy' ("random" or "hostile"), and colour the turtles by this variable.

Making the hostile turtles use the hostile strategy

Create a turtle-context reporter called 'choose-strategy', which reports the output of the strategy reporter ('random-strat' or 'hostile-strat') that corresponds to the turtle's value of 'my-strategy'. Edit the 'play-game' procedure so that the local variables storing each player's choice of row use this new reporter to ensure that turtles behave as they should.