

Appendix D: Simulation User Instructions

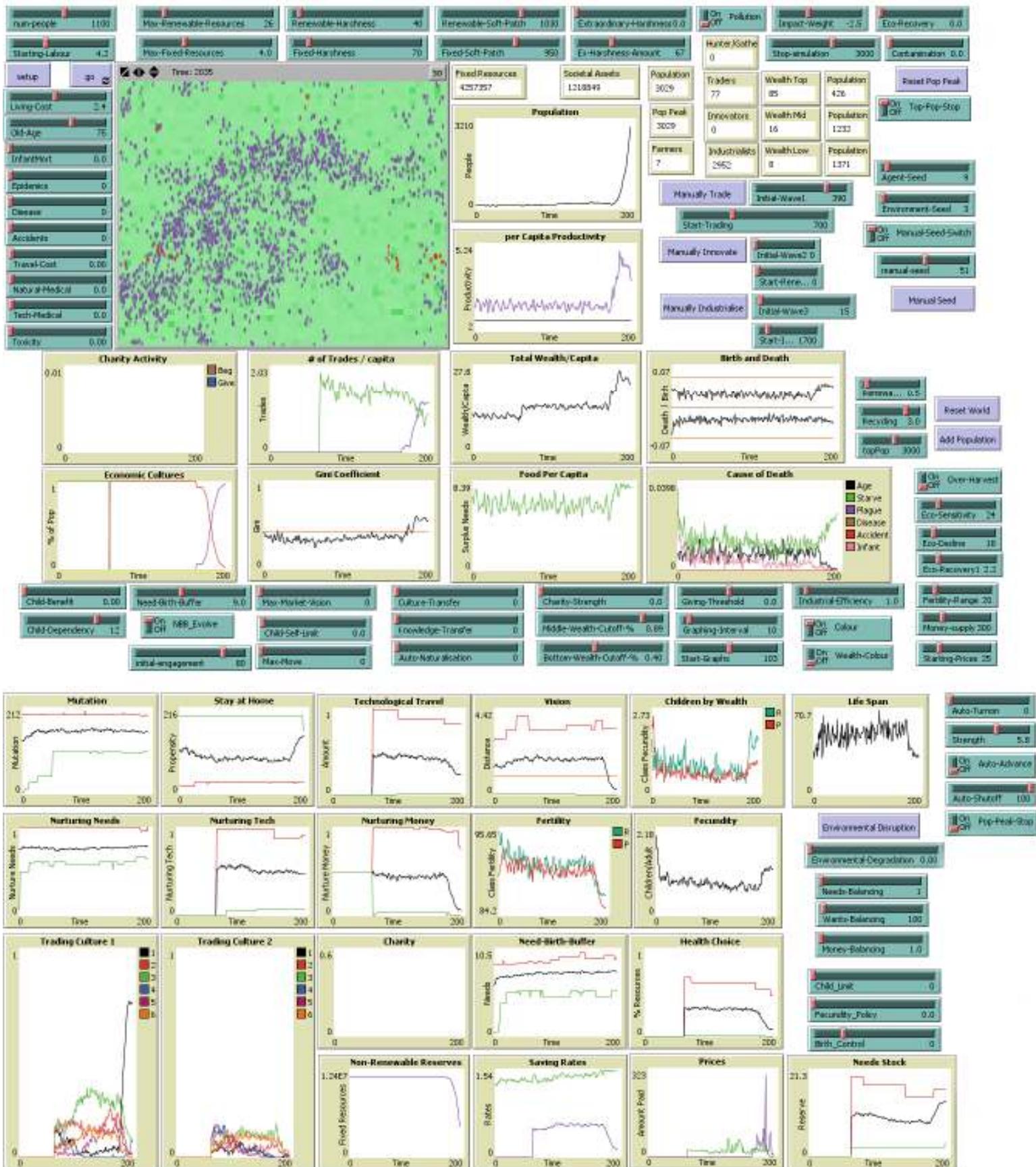


Figure 35. Full interface for Societal Simulator v203 with random seed setting 9, 3, 51.

Important Note: this is an active experimental tool rather than a polished demonstration of a single experiment. Changes to the code and the charts are made frequently, sometimes from run to run. While these instructions are quite comprehensive, maintaining these instructions to reflect the code in every experiment is not a primary concern.

These instructions are better viewed in the simulation itself (see the *Info* tab in the simulator). This has been included in the appendix for anyone who does not have the simulator but may be curious of aspects of the simulator not discussed in the paper.

SECTIONS

PROGRAMME OVERVIEW
BUTTONS, SWITCHES AND SLIDERS
GRAPHS AND MONITORS
THE EVOLUTIONARY GRAPHS
COPYWRITE NOTICE

Please note: the names of the charts, monitors and the variables in the code were primarily chosen as an aid in remembering what they do. Some of these names do not attempt to reflect neutral scientific concepts. Please do not read too much into how things are named.

PROGRAMME OVERVIEW

This simulator is designed first and foremost to be an experimental tool. The controls can be extremely sensitive in some setups and then seem barely functional in other setups. It is easy to setup the simulator so that it fails to do anything similar to human society. It is recommended you save the interface [file --> export --> interface] before you make changes so that you can easily find a working setting if new settings become dysfunctional.

If an interesting run results from your settings, it is advisable to use the 'Reset World' and 'Repopulate' buttons to test the robustness of the trends as the environment will remain unchanged. Using 'Setup' will create a new randomised environment based on your settings. Most rigorously, an experimenter can save the entire world [file --> export --> world] after 'Setup' and before starting a simulation run. This will permit the experiments with the exact same condition. For example, after the initial run, the experimenter could reload the setup from the saved file, cause a single agent to die (type 'ask one-of people [die]' in the command centre), and then do a second run to see what happens.

This simulator is designed so that you can set it up and walk away from it such that it will run for a user set amount of time and then automatically shut off. This way you don't have to sit and watch it. See the 'Pop-Peak-Stop' button and the 'Stop Simulation' slider.

The core of this simulation is designed around three environmental elements and one agent:

'Needs' – a single environmental variable designed to represent renewable resources, such as food, wood (when harvested renewably) and water.

'wants' – a single environmental variable designed to represent the benefits non-renewable resources such as oil or Uranium, medical technology, vehicles, etc.

'Money' – a fixed resource of no intrinsic value but can be traded for either 'wants' or 'needs'. The money supply is fixed after startup in this version.

'agent' – an evolutionary entity that navigates the simulation environment. It is meant to capture a few key forces generated by human beings in the environment of the world today.

The agents in the simulation can exist in four different states which, in turn, create four different possible agent-generated simulation environments. Somewhat arbitrarily, these four states are called 'Hunter/Gatherer', 'Trading', 'Innovating' and 'Industrial'.

The key characteristics of these four states in their simplest forms are:

'Hunter Gatherer' – (Consisting of Agents and 'Needs')

Agents look for and harvest food, reproduce (asexually), and die from either lack of food or old age. The environment consists of patches that contain the single renewable food source 'needs' and a single non-renewable source called 'wants'. Each has a random amount of resistance covering them that the agent needs to overcome with their 'labour' in order to harvest. 'needs' is available in set amounts that is randomly assigned to each patch at the beginning of a simulation run and are renewed once per iteration.

Trading – (Consisting of Needs and Money)

Agents do all of the above and, in addition, they can now buy and sell 'needs' using money.

Innovating – (Consisting of Needs, Wants and Money)

Agents do all of the above but they now have the choice of converting their 'needs' into 'wants' simulating the force of turning renewable resources into fixed assets that deteriorate without human intervention. This only occurs in the 'innovating' state.

Industrial – (Consisting of Needs, Wants and Money)

Agents can now harvest non-renewable 'wants' (non-renewable technology) directly from the patches. In this version of the simulation, agents in an 'Industrial' state do not convert 'needs' into 'wants'.

How the agents progress through these stages are controlled by a series of buttons and switches. The user can also introduce a number of complicating elements through the use of the buttons and switches.

BUTTONS, SWITCHES AND SLIDERS

Accidents – This is a random death that is unmitigated by wealth or medicine. Increasing the number increases the strength. A setting of '0' turns it off.

Agent-Seed – This slider determines the seed for the random number generator that initiates the agent population. A setting of '0' turns it off.

Auto-Advance – This switch creates a probability of randomly advancing individual agents through the different economic stages by assigning a random agent one level above its current level. Works in conjunction with the 'Strength' slider.

Auto-Naturalisation – This allows the simulation to randomly convert agents to previous levels of economic activity. For example, an 'industrialist' might get converted to a 'trader'. A setting of '0' turns it off. Strength is determined by the position of the slider.

Auto-Shutoff – Once there is a population of 'industrialists' equal to this setting, 'Auto-Advance' is automatically turned off.

Auto-Turnon – This slider automatically switches on 'Auto-Advance' at a given iteration.

Birth-Control – This creates a pressure for the agents to have less children than they naturally would. It differs from 'child-limit' in that 'birth-control' is effective all the time while 'child-limit' becomes progressively more effective as an agent has more children.

Bottom-Wealth-Cutoff-% - See Middle-Wealth-Cutoff-%.

Charity-Strength – Works on setup only. Above '0' this slider causes agents to give 'needs' to agents who are 'begging'. The amount of charity that an agent gives is determined by randomising the number indicated by this slider. See also the 'Giving-Threshold' slider.

Child-Benefit – This simulates taxation and redistribution to children. The higher the slider, the greater the benefit and taxation.

Child-Dependency – This causes new agents to rely on their parents for food. Works in conjunction with 'New-Labour'. If 'Child-Dependency' is set above zero, the effects of the 'infantmortality' slider will be constrained to agents within the 'Child-Dependency' range.

Child-Limit – This limits the number of children any agent can have. It simulates the effects of the 'Chinese Policy' at a global scale. See also the 'Birth-Control' slider.

Child-Self-Limit – This creates a global force that automatically lowers the fecundity by effecting an agent's probability of having children. It simulates a society-wide wilful decision to have less children.

Colour – This switch colours the agents based on their economic activity. 'Hunter/gatherers' are green, 'traders' are red, 'innovators' are yellow and 'industrialists' are purple. Turning on this switch over-rides the 'Wealth-Colour' switch.

Contamination – This spreads the effects of 'pollution' to neighbouring patches. See 'Pollution'.

Culture-Transfer - This determines the strength of the desire of agents to change their ways to match the richest agent around them. Setting it to '0' turns it off. This is a new slider. The effectiveness of the slider and the effectiveness of using isolated individual characteristics has not yet been thoroughly tested and tuned.

Disease – This simulates individual disease. A setting of 0 turns it off. An agent's resistance to 'disease' can be mitigated by wealth if either of the 'medical' sliders are turned on.

Eco-Recovery – This determines how quickly patches recover from the environmental disruption caused by the harvesting of 'wants' (brings down the 'land-usage' variable). Works in conjunction with the 'Pollution' switch and the 'Impact-Weight' slider.

Eco-Sensitivity – This sets up how sensitive a patch can be to over-harvesting of 'needs'. The 'Over-Harvest' switch needs to be turned on for this to work.

Eco-Decline – This determines how quickly patches decline if they are over-harvested. The 'Over-Harvest' switch needs to be turned on for this to work.

Environment-Seed – This slider sets the random number generator seed used for the generation of the simulation's patches. A setting of '0' turns it off.

Environmental-Degradation –. This slider works with 'Environmental Disruption'. The number set by this slider is the percentage reduction of the maximum 'needs' available. For example, setting the slider to .1 will decrease the maximum possible on a patch by 10% and increase the maximum harshness possible on a patch by 10%.

Environmental Disruption – This button recalculates the distribution and resistance covering the 'needs' in the patches using the original setup settings.

Epidemics – This simulates the probability of epidemics. The higher the number, the greater the strength. A setting of 0 turns it off. An agent's resistance to 'epidemics' can be mitigated by wealth if either of the 'medical' sliders are turned on.

Extraordinary-Harshness – This creates patches where it is impossible to harvest 'needs' without a great deal of 'technology'.

Ex-Harshness-Amount – This determines how great the 'extraordinary harshness' is.

Fecundity-Policy – This prevents the global fecundity from increasing beyond a certain level. It is a fantasy population control policy that was included purely for experimentation purposes.

Fertility-Max – This highest possible fertility an agent can have when the simulator is first setup.

Fixed-Harshness – On setup, the programme randomises this number to randomly the resistance covering the fixed 'wants' on each patch.

Fixed-Soft-Patch – This is the number of patches that will have its resistance covering 'wants' forced to become lower than a single agent's 'labour'.

Giving-Threshold – This determines the level of an agent's 'needs' below which the agent will begin to 'beg'. Works in conjunction with 'charity-strength'.

Go – Starts and stops the simulation.

Graphing-Interval – This determines how often a plot is made in the graphs. If it is set to '10' the simulation will average the variable for those 10 iterations and graph the result. This was done to make the graphs more readable. Set it to '1' if you wish to see every plot, but the relatively small populations required by personal computers makes the graphical trends much harder to read at this setting.

Impact-Weight - This weights the impact of harvesting 'wants' on a patch. Operates during setup only. Works in conjunction with and the 'Pollution' switch and the 'Eco-Recovery' slider.

Industrial-Efficiency – This determines how powerful 'wants' are in the environment.

InfantMort – increases the mortality of agents less than an age of 5 iterations or those considered 'dependent' if the 'Child-Dependency' slider is greater than zero. An agent's resistance to infant mortality can be mitigated by wealth if either of the 'medical' sliders are turned on.

Initial-Wave1 - this determines approximately how many agents will be turned into 'Traders' when either the 'Manually Trade' button is pressed or the simulator iteration reaches the setting of the 'Start-Trading' slider. See also 'Auto-Advance'.

Initial-Wave2 - this determines approximately how many agents will be turned into 'Innovator' when either the 'Manually Innovate' button is pressed or the simulator iteration reaches the setting of the 'Start-Renewable-Economy' slider. See also 'Auto-Advance'.

Initial-Wave3 - this determines approximately how many agents will be turned into 'Industrialists' when either the 'Manually Industrialise' button is pressed or the simulator iteration reaches the setting of the 'Start-Industrialisation' slider. See also 'Auto-Advance'.

Knowledge-Transfer - This determines the strength of the economic transfer to other agents. This will only work if 'Pollution' is turned on in the simulator. Setting it to '0' turns it off. See article for more information. This is a new slider introduced and has yet not been fully tested.

Life-Expectancy-Max – the approximate maximum age an agent can reach.

Living-Cost – how much 'needs' is subtracted from the agent each iteration.

Manual-Seed – This slider sets the random number generator seed used to control agent decision making.

Manual-Seed – This button allows the user to set the random number generator seed used to control agent decision making while regenerating a population. Used in conjunction with the 'Add Population' button.

Manual-Seed-Switch – Controls whether 'Manual-Seed' is active or not.

Manually Trade - A button that instantly converts agents into 'Traders'. Works in conjunction with the 'Initial-Wave1' slider.

Manually Innovate - A button that instantly converts agents into 'Innovators'. Works in conjunction with the 'Initial-Wave2' slider.

Manually Industrialise - A button that instantly converts agents into 'Industrialists'. Works in conjunction with the 'Initial-Wave3' slider.

Max-Fixed-Resources – On setup, the programme randomises this number to generate the number of 'wants' available on each patch.

Max-Move – This limits the technological boost to movement to the number of patches to which this slider is set. A setting of '0' allows the simulator to choose how far an agent can move. Usually, it chooses a great deal of movement.

Max-Renewable-Resources – On setup, the programme randomises this number to generate the number of 'needs' available on each patch.

Max-Vision-Market – This limits the reach of agents for trading. Set to '0' turns this feature off and when the technology boom begins the richest agents are likely able to trade with any agent in the environment.

Middle-Wealth-Cutoff-% and Bottom-Wealth-Cutoff-% -- These two sliders define the agent's wealth class used by the graphs. It divides the agents by the percentage of population. For example, if Middle-Wealth-Cutoff-% is set to .90 and Bottom-Wealth-Cutoff-% is set to .40, then the wealthiest will be defined as the wealthiest top 10% of the population, the middle class will be 40% to 90% and the poorest will be defined as the lowest 40%. These sliders influence the graphs only and have no impact on the simulation.

Money-Balancing – see Needs-Balancing

Money-Supply – This determines the amount of money each agent possesses at start up which then determines the money supply for the duration of the simulation.

Natural-Medical – This uses excess 'needs' to diminish the chance of dying from 'disease', 'InfantMort' and 'Epidemics'.

NBB_Evolve - This switch determines whether the amount of 'needs' an agent possesses influences its reproduction rate. See 'Need-Birth-Buffer'.

Need-Birth-Buffer – This slider prevents births unless an agent has a minimum amount of 'needs'. If NBB_Evolve is turned on, on setup the agents randomises the 'Need-Birth-Buffer' setting which then evolves. If it is turned off, all agents accept the 'Need-Birth-Buffer' setting during the simulation. This can be changed during the simulation.

Needs-Balancing, Wants-Balancing and Money-Balancing – These sliders give weight to each item in regards to what the simulation considers wealth. These sliders influence the graphs only and have no impact on the simulation. Using the internal prices to determine wealth requires larger populations than my computer can sustain to create usable charts.

New-Labour – This sets the 'labour' of child dependence to the fraction set by this slider. This is effective when 'Child-Dependency' is set above 0.

num-people – this determines the number of randomly generated agents at the beginning of a simulation run.

Old-Age – The age above which the likelihood of death drastically increases for the agent.

Over-Harvest – This turns on the effect of diminishing harvests if a patch is over harvested. This is used in conjunction with 'Eco-Sensitivity' and 'Eco-Decline'. At the time of writing (20-10-2012) this is a new feature and its effects are yet to be evaluated and calibrated.

Pollution – This turns on the effect of the patch's land-usage variable which increases the resistance to harvesting 'needs'.

Pop-Peak-Stop – This automatically stops the simulator if the population drops below a certain percentage of the population peak. It is currently set at 15% in the programming. I'm tempted to include a slider to allow the user to alter this at will.

Recycling – This determines how quickly non-renewable resources erode from the system, representing wear and waste. The higher the number, the lower the wear (the more recycling).

Renewable-Economy – This determines how efficient the agents are at converting their renewable resources into technology. This is only relevant to agents working as 'Innovators'.

Renewable-Harshness – On setup, the programme randomises this number to generate the resistance covering the 'needs' on each patch.

Renewable-Soft-Patch – On setup, this is the number of patches that will have its resistance (harshness) covering 'needs' set lower than a single agent's 'labour'. It allows for a greater range of resistance while maintaining a viable population at the beginning of the simulation.

Re-populate - This generates the number of new agents set by the 'num-people' slider. It is meant to be used in conjunction with the 'Reset World' button.

Reset Pop Peak – Resets the peak population monitor to the current population. This effects 'Pop-Peak-Stop'.

Reset World - This button clears the graphs and the removes all the agents while resetting the world to an identical state to the last time the 'Setup' button was pressed. If you use this button while the simulation is running you will get an error.

Setup – Sets up the simulation environment and generates the agents.

Start-Graphs – This determines the iteration that the graphs begin to plot. This is included because it takes several iterations for the simulator to settle down and can create extremes that subsequently make the graphs hard to read.

Start-Industrialisation - This slider determines on which iteration a number of agents will be converted into 'Traders'. Works in conjunction with the 'Initial-Wave3' slider.

Start-Renewable-Economy - This slider determines on which iteration a number of agents will be converted into 'Innovators'. Works in conjunction with the 'Initial-Wave2' slider.

Start-Trading - This slider determines on which iteration a number of agents will be converted into 'Traders'. Works in conjunction with the 'Initial-Wave1' slider.

Strength – This determines the strength of 'auto-advance'. The higher the number, the more likely agents will be raised to the next level of economic activity.

Starting-Labour –The number set on Starting-Labour determines the maximum amount an agent can potentially harvest each iteration before 'technology' is introduced.

Stop-Simulation – This slider allows the user have the simulation stop at a particular 'tick'. It is useful when to comparison runs where the shape of the graphs are important because this will force the graphs to have an identical compression. Also, this allows the user to 'set and forget' the simulation and come back later on to see what happened.

Tech-Medical – This uses 'wants' to diminish the chance of dying from 'disease', 'InfantMort' and 'Epidemics'.

Toxicity – This increases the effects of disease and epidemics as the patch variable land-usage increases. This simulates a health cost of pollution.

Travel-Cost – This is the cost of that travel in terms of the proportion of 'wants' (technology) that was used for that travel.

Wants-Balancing – see Needs-Balancing

Wealth-Colour – This switch colours agents according to their wealth band. Red is poor, yellow is middle income, and purple is rich. The bands are determined by 'Middle-Wealth-Cutoff-%' and 'Bottom-Wealth-Cutoff-%' sliders. The programming is currently set to divide the wealth bands by population (ie. the top 10% of the richest make up the top band) as opposed to using a proportion of total wealth (ie. those people with wealth above certain percentage of wealth are the richest). This switch cannot be turned on if the 'Colour' switch is turned on.

GRAPHS AND MONITORS

of Trades / capita – Sums the trades and divides by the population. Purple line is the per capita trading of 'wants' and the green line is the per capita trading of 'needs'.

Average Age – Ages of the agents has been summed and divided by the population.

Birth and Death – This graphs the total births in that turn divided by the population and the negative of total deaths divided by the population. The middle orange line is 0 and the two red reference lines are currently set at .9 for birth and -.9 for death.

Charity Activity – The brown line charts the percentage of agents who 'begged' and the blue line charts the percentage of agents who 'gave' in a given iteration. See the 'Charity-Strength' and the 'Giving-Threshold' sliders.

Children by Wealth – The turquoise line measures the fecundity of agents in the middle and top wealth bands while the red line measure the fecundity of agents in the bottom wealth band.

Farmers – This monitors the percentage of the population who choose harvesting 'needs' over 'wants'. As industrialists, these agents will only harvest 'wants' if they can't harvest any 'needs' during an iteration. See also the 'Trading Culture 1' and 'Trading Culture 2' graphs and the 'order-of-trade' variable under Evolutionary Variables.

Fecundity – Reports sum of 'children' had by the living agents divided by the sum of all agents above the age of 11 (the age they need to be before they can have children).

Fixed-Resources – Sums the patches' 'wants'. It is useful to see whether the agents are still harvesting 'wants' from the patches when the amounts are too small to register on the graph. See also the 'Non-Renewable Reserves' graph.

Food per Capita - This sums up all the food held by adults in the system and divides it by the population of adults and the 'living-cost'.

Gini Coefficient - This charts the Gini coefficient for this simulated world. The code is borrowed from Uri Wilensky's Sugarscape (see the model library). The user can alter the weight of the three wealth variables (needs, wants, money) by using the 'balancing' sliders. See 'Needs-Balancing' under 'Buttons, Switches and Sliders'.

Hunter/Gatherer – This monitors the population of 'Hunter/Gatherers'.

Industrialists - This monitors the population of 'Industrialists'.

Innovators - This monitors the population of 'Innovators'.

Life Span – This plots the average age of death of the entire population.

Non-Renewable Reserves – Graphs the sum of the patch's 'wants'. See also the 'Fixed-Resources' monitor.

per Capita Productivity – Graphs the amount of harvesting of 'needs' and 'wants' in that cycle (tick) divided by the population. 'Wants' has been graphically repressed in order to prevent the graph from shooting meaninglessly straight up every time the industrialist take over.

Population (graph) – Graphs the simulation's population over time.

Population (monitor) – Reports the number of agents in the simulation at any one time.

Population (monitor beside Wealth Top, Wealth Mid and Wealth Low) - This monitors the population of each wealth band.

Pop Peak – Reports the highest population recorded since the Reset Peak Population button was last pressed. Peak is reset to '0' once the simulator reaches the 'Start-Graphs' iteration.

Prices – Adds up all the prices in that iteration and divides them by the number of transactions. The purple line is the price for 'wants' and the green line is the price for 'needs'.

Societal Assets – The sum of all non-renewable assets ('wants') held by agents.

Total Wealth per Capita – Sums the agents' 'needs' and 'wants' and divides by population. As in the per Capita Productivity graph, the 'wants' has been suppressed.

Trading Culture 1 - This graphs the six different ways an agent can trade for an agent that is emphasising 'wants' over 'needs' in their harvesting (or work).

Trading Culture 2 - This graphs the six different ways an agent can trade for an agent that is emphasising 'needs' over 'wants' in their harvesting (or work). This group is called 'Farmers' in the simulation.

Traders - This monitors the population of 'Traders'.

Wealth Top, Wealth Mid and Wealth Low – This is the average wealth of the population in each wealth band.

X (Lower Wealth) = Higher Wealth – Using the simulation's definition of wealth, X is the variable graphed, low wealth is the sum of wealth of the lowest wealth class divided by the population of that class and high wealth is the sum of wealth of the highest wealth class divided by the population of that class.

THE EVOLUTIONARY GRAPHS

For almost all these graphs, the black line is the system average (the sum of the agent's variable divided by the agent population) while the red line reports the highest value of this variable held by any agent in the system while the green line reports the lowest value.

Children by Wealth – Reports two variables: the sum of children had by people from the lower class divided by the sum of people in child bearing age in the class and the sum of children had by middle and upper classes divided by the sum of people of child bearing age of these two classes. The red line is the lower class while the turquoise line is the upper and middle classes.

Fertility – Reports the average 'fertility' of the lowest class by wealth and the combined average of the middle and upper class by wealth. The turquoise line is the upper and middle classes while the red line is the lower class.

Health-Choice – Reports the variable 'propensity-health'

Labour-Choice – Reports the variable 'propensity-labour'

Leisure-Choice – Reports the variable 'propensity-leisure'

Modern Travel – Reports the variable 'migration'. This is how intensively agents use technology to help them move.

Mutation – Reports the variable 'mutation'.

Nurturing-Needs – Reports the variable ‘nurturing-needs’

Nurturing-Tech – Reports the variable ‘nurturing-wants’

Nurturing-money – Reports the variable ‘nurturing-money’

Savings Rates – Reports two variables: the sum of savings rates for ‘needs’ divided by the population and the sum of savings rates for ‘wants’ divided by the population. The purple line is ‘wants’ and the green line is ‘needs’.

Stay at Home – Reports the variable ‘exploration’. The higher it is, the more likely an agent will be to stay at home.

Vision -- Reports the variable ‘vision’.