Brent Barbachem and Mariaclara Ludovici

BNAL 722: Agent Based Modeling and Simulation

Wednesday 4:20 – 7:00 p.m.

Semester Project

***Introduction and Motivation***

Current, world, news has displayed the mass amount of immigrants attempting to enter into Europe from the southern nations around Africa and the Middle East. Many immigrants have been pardoned and allowed to enter the European nations, but there has been little or no discussion about the lasting effects that meshing cultures could bring. It would not be fair to the Europeans or the immigrants to declare that only one culture may survive, but is there a way to peacefully coexist? Will one culture inadvertently take over the other culture? The purpose of this project is to use Agent Based Modeling and Simulation to study the possible effects of immigration into Europe from outside nations.

Mariaclara authored a paper on the subject of the possible or probable effects of immigration into Europe in February 2016. Mariaclara investigated possible scenarios using Game theory techniques in her paper, but we will use agent based modeling and simulation to explore the issues from a different angle. We will simply use the possible outcomes of the games proposed by Mariaclara, but we will not directly use game theory. The agents in our simulation will have the ability to learn which game they like to play the most, and the game outcomes may be weighted by the percentage of the population that supports a specific culture. The ability to learn and provide significant weights will allow us to distance our approach from that of strict game theory.

***Interface Variable Descriptions***

The model includes numerous interface variables that the user can alter in order to vary the execution outcomes, shown in *Figure 8* in *Appendix B*. The first, and easiest, interface variables to alter (with no effect on the outcome) are the eu\_color and im\_color. The user can select the display colors for the agents in order to identify the agents. It is important, and obvious, to mention that the agents should be colored differently, and the agent colors should not be black (unless the background color is changed).

The initial\_eu\_number is a slider that will indicate the number of agents, recognized as favoring the European culture, to be created when the setup procedure is executed. The initial\_immigrants\_number is a slider that will indicate the number of agents, recognized as favoring the immigrant culture, to be created when the setup procedure is executed. The search\_area variable indicates how large the circle will be for an agent to search for another agent to play a game against. The smaller the search area, the less likely the agent will find an agent favoring the opposite culture. The preference slider indicates the amount that an agent will increase their preference for a particular game when they select that game.

The preference variable is very important, because it allows the agents to adapt and favor certain games over others. Implicitly defined with the preference variable is a variable that subtracts one quarter of the preference value from each of the games percentage of being selected. Once a game has achieved a probability of 1.0, the game will be selected on each subsequent tick.

The harmony-prec variable is the initial percentage of the time that the game will be chosen; it is the probability of selecting or using the harmony game. The deadlock-prec variable is the initial percentage of the time that the game will be chosen; it is the probability of selecting or using the deadlock game. The prisoners-prec variable is the initial percentage of the time that the game will be chosen; it is the probability of selecting or using the prisoner’s dilemma game. The coordinate-prec variable is the initial percentage of the time that the game will be chosen; it is the probability of selecting or using the coordination game. The chicken-prec variable is the initial percentage of the time that the game will be chosen; it is the probability of selecting or using the chicken game.

***Operating Instructions***

If user does not like the color choices for the immigrants and Europeans, they can alter the colors for each agent using the two appropriate inputs. The user should select an appropriate number of European and immigrant agents using the sliders. The user can select a search area that they feel is appropriate for searching for other agents. It is advised that the user does not set the preference variable too high, because it will, essentially, eliminate the agent’s ability to learn. In order to draw the learning process out over a longer period of time, the preference variable should be set no higher than 0.20. Another important quirk about the model is that all of the percentages for the games must add up to 1.0 or the program will automatically redistribute the numbers to be 0.20 each. A warning will be generated when the user fails to properly set the values, but it should be mentioned before the user attempts this action. After all of the interface variables have been adjusted to meet the user’s requirements, click the setup button to run through the setup procedure. Proceed with the simulation by clicking the Go button to run the go procedure. It is important to note that there is no current case that will end the simulation, but this will be addressed below in the *extending the model* section.

***Model Description and Technical Implementation***

At the start of each tick, the agent set for each cultural preference is constructed, shown below in *Code Block* 2 in *Appendix* B. It is important to mention this feature, because, as one may be thinking, the agent could switch cultural preferences multiple times during a single tick. Exactly! We wanted to allow the agents to be able to alter their cultural preference for each encounter with another agent whose cultural preference differs. The design consideration to query all of the agents and set them into agent sets at the beginning of each tick was also chosen to alleviate the computational power required and to decrease the runtime. The computational power decreases, because the query is run once instead of running once for each agent.

The agents will begin by wiggling, *Code Block* 3 in *Appendix* B, to add variability and ensure that agents experience different encounters on each iteration. The agents will then move forward by 1 space using the move procedure in *Code Block 4* in *Appendix B*. We have created what is known as a wrapper function or procedure for the forward procedure with the specific intention of model extension (see *model extensions* for further explanation). New behaviors may be experienced if certain agents are given the ability to move at different speeds or distances.

All of the agents are asked to search for another agent that does not share the same cultural beliefs within the user defined search area (from the interface). In the dual procedure, *Code Blocks 6* and *7* in *Appendix B*, the current agent will ask for one-of the agents that is both in their [user-defined] radius and that does not share their cultural preference.

If another agent is found, then a game will be selected at random from a list of 5 predetermined games. The results of the games are predetermined from Mariaclara’s original report, and they are mentioned in the section directly below. The results can be any of the following: both agents support the European culture, both agents support the immigrant culture, or one agent supports one culture while the other agent supports the opposing culture.

The prisoner’s dilemma and coordination games contain extra logic that will skew the end results by attempting to select the culture that is dominating the simulation at the current tick. After the results have been applied, the agent’s probabilities of choosing each type of game are updated; this behavior can be observed in *Code Block 8* in *Appendix B*. The agent is coded with the ability to adapt or choose only games that they have encountered most often.

***Agent Based Modeling vs. Game Theory Approaches***

Now the question that you may be wondering is, does agent based modeling and simulation offer a novel approach to this problem? Our implementation, using agent based modeling and simulation, contains different features such as learning, adapting, and alternative scope from the game theory approach.

The feature of learning can be seen in the adjust procedure, *Code Block 8* in *Appendix B*. The agent’s probabilities of choosing each game will be altered depending on the game that was currently chosen. The more often that a game is chosen, the more likely it will be chosen in the future. Once an agent’s probability of choosing a particular game reaches a value of zero, they will never be able to choose that game again; conversely the maximum probability of choosing any particular game is 1.0.

The feature of adapting can be examined in the dual procedure, *Code Blocks 6* and *7* in *Appendix B*. If the game that is selected is either the prisoner’s dilemma or the coordination game, then the agent’s cultural preference will be skewed by the percentage of the population that supports a particular culture. The agent is given the ability to be influenced by the other agents in the simulation. It is important to note that we simulated the ability to resist influence by randomly picking a value and comparing it to the population percentage.

The agent based modeling and simulation approach has also taken the scope of the problem away from the game theory approach completely. The ABMS approach utilizes the two features described previously, which are dependent upon random values. Unlike most game theory techniques, there is no guaranteed solution to this problem and the results between simulation runs will never be identical (unless we use a random seed). The original game theory techniques simulated the possible outcomes of entire populations, while the ABMS approach simulates the games using individual players or agents.

***Experiments, Results, and Observations***

User(s) should notice that no matter how many agents are added to the model, the number of agents favoring the host culture and the number of agents supporting the immigrant culture will always teeter around the fifty percent line if the percentages are left to the default 20 percent each. We believe that this happens because of the balance of the results from the predetermined game theory results. In other words, the ability to learn and adapt may add more variability to the model, but their effects can be cancelled or neglected. The reason for this phenomenon is that the random numbers may lead each agent to favor one particular game. Since the results are balanced, then the agents may change their preference for a particular culture, but the chances are that for each agent that switches there is also an agent that reverses this effect. The slight teeter of the number of cultural supporters can be referred to as a negative or balancing feedback loop.

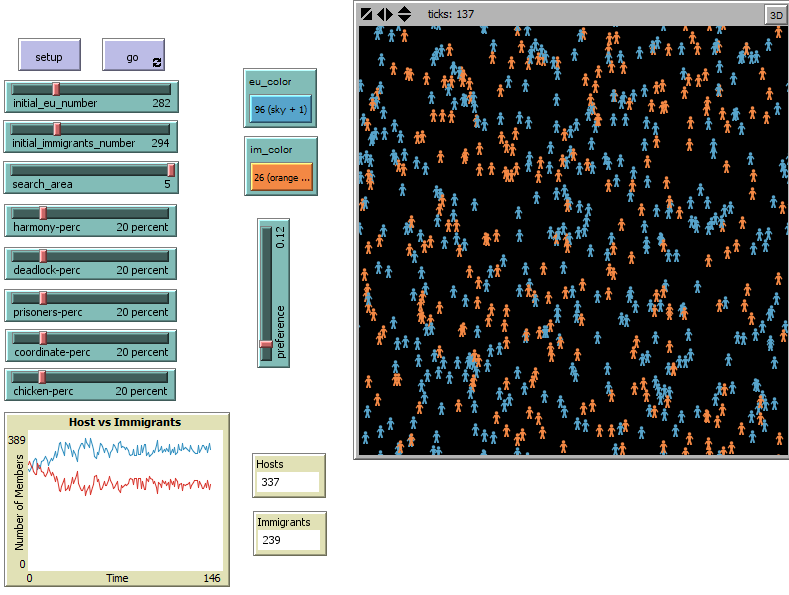
Depending on the percentages, it is possible to observe each culture completely take over and control the population. This result is most likely due to the ability to adapt. If one culture starts to take over and the number of people favoring that culture grows too large, then the games will favor that particular culture. This creates a positive or reinforcing loop in the simulation.

Try moving the sliders that initialize the model with different numbers of agents including the initial immigrants and European numbers. The search area can be adjusted while the model is executing, but the runtime alterations do not appear to greatly affect the model. In order for the results to be noticed as the search area is altered, try setting the initial immigrants and European numbers very low. You may also wish to slow down the execution speed in order to be sure to notice the differences.

Try altering the percentages in order to observe different scenarios. It is important to note that the user will also see different results even leaving all of the values the same between different runs of the simulation.

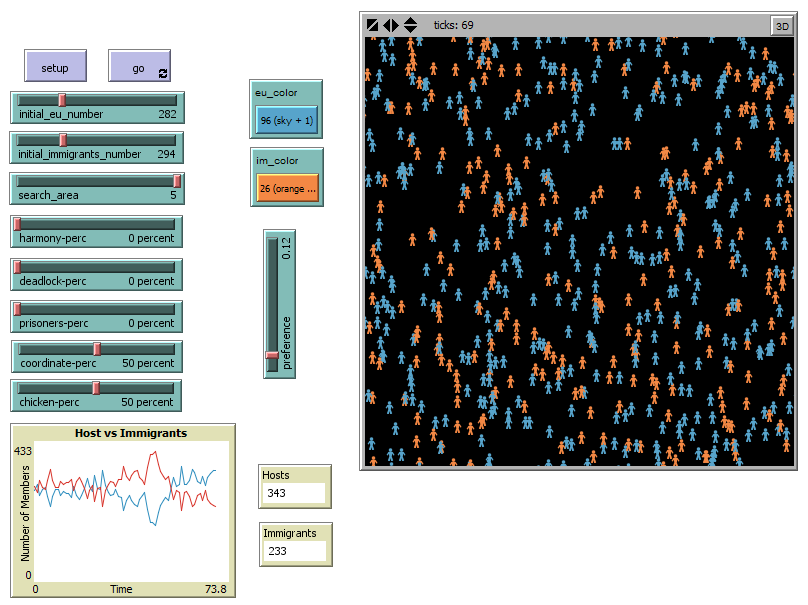
As mentioned previously, it is advised that the user alter the preference variable, but please be cautious. A very high value for the preference will lead to the agents finalizing their favorite game very quickly. In fact, when the preference is set too high the number of agents supporting each culture will only change slightly with no true growth or decay. After testing the model, we decided that a value of 0.20 or lower should be used as the preference value to ensure that the agent does not find a favorite game too quickly.

The image below, Figure 1, shows the results of running the model with all of the probabilities set to an initial value of 0.20 and the preference set to 0.12. The graph in the bottom left corner indicates that the Host or EU culture will take over. It is interesting to note that the initial number of agents supporting each culture is relatively even, but the EU culture quickly takes over. The nature of the graph leads us to believe that the prisoner’s dilemma, harmony, and coordination game are selected most often in this early period. We can determine this nature, because the harmony game will ensure that both players favor the EU culture. The results of the Prisoner’s Dilemma and Coordination games favor the culture that is currently dominating the simulation.



**Figure [1]**: Results of run with equal probabilities for all games.

The image below in Figure 2 displays a simulation run with only the coordination and chicken games selected. The simulation run was a little more interesting than most of the simulation runs. In the majority of the simulations, we were able to notice that there were possible crosses in the graph for the number of immigrant and EU supporters. In this particular run, we can notice that the normal majority culture switches, but it is interesting to see how large the separation can become and still be able to switch. Around tick 45, we can see the largest separation occurs, but the EU culture is able to recover.



**Figure [2]**: Results of run with only the coordination and chicken games.

The results of our simulation indicate that it is possible for both cultures to coexist without one culture overtaking the other culture. In certain scenarios, the culture that starts with a significant advantage over the other, will quickly take over and it will become impossible for the other culture to advance or recover. Conversely, the simulation runs indicate that it is possible to start with as little as one agent and the culture can grow; the culture can reach an achievement of about fifty percent of the population. The purpose of the model was not to predict the future, instead the purpose was to explore the possible scenarios that can occur. We believe that our model was successful in determining and conveying the possible scenarios that could develop from conflicting cultures.

***Extending the Model***

There are several ways that the user can extend the model or alter the way that the model executes. The first and easiest extension is to modify the move procedure, *Code Block 4* in *Appendix* B, so that the agents can take in different values for the forward command. Once the procedure has been altered, further modify the model by sending in values to the function for different agent sets. We have chosen this action as an extension, because it can be a simple entry to successful model extensions. We ask that you observe and report and changes from running the simulation in this manner. It is important to note that the user must run a lot of iterations in order to grasp true changes, because the model is based on random numbers thus no iteration will be exactly the same (even with no variable alterations).

The second extension of our model is that the user can alter the number of preset game theory techniques by rearranging the order or decrementing the number of possible techniques. The extension is listed before several others, because we believe that this will take the far less time to accomplish and it will have the least total impact on the current code. The user can make simple changes to the code by commenting specific sections out or altering a single number.

The third model extension is the addition of new game theory techniques. In fact, there is a comment in the code for the dual procedure that indicates where the new techniques may be added. The user must know all of the possible Nash Equilibrium points in the game, or, if there are not any equilibrium points, the possible solutions to the game. The extension seems rather simple at first glance, but an additional game will require the addition of code to the adjust and random-game procedures. The user will also be forced to change the error checking code in the setup procedure, and add a new slider to the interface tab.

The fourth and final extension is the addition of a termination case. In the event that a user would like to run multiple iterations of the model, a proper termination case should be established. We have explored the code for this extension, but we have opted to leave the code out. We will however provide simple guidance and instructions to assist in the development. The user should add a measure of the average number of agents supporting each culture for a given amount of time. For example, keep track of the numbers over a period of 100 ticks. If the average number of supporters for the European culture is within a tolerance level of the current value, then the simulation has reached a steady state.

***Similar Models***

All Prisoner’s Dilemma models under the Social Science portion of the model library were considered to be similar to our model. While we did not use any of the code from the models, and we strived to move the scope away from the game theory approach, we still considered the prisoner’s dilemma models to be helpful in the development of our model. The prisoner’s dilemma models were advantageous to study, because they allowed us to examine the possible solutions based on rational human behavior. Discussed in Appendix A, the assumption of rationality was important in our study. The models from the Netlogo library that we consider similar to our model include: PD Basic, PD Basic Evolutionary, PD N-Person Iterated, and PD Two Person Iterated.

The segregation models are also an important set of models that we consider to be similar to our model. These models are related, because our model uses the idea that people want to be near people with the same characteristics. This feature was described as the adaptability feature of our model. The agents’ results of their game will be skewed by percentage of the population that supports a particular culture.

***Netlogo Features***

The model that we developed did not include special or unseen features of Netlogo, but we included features that were essential to the current implementation of the model. The features that we considered essential were the on-of, ask, and nobody keywords.

The one-of procedure was particularly useful in order to obtain a second player to the game. In combination with the in-radius procedure we were able to attempt to pick out a target. The use of the one-of command can be observed in *Code Block 6* in *Appendix B*.

Usually the ask command is used with the set of turtles or patches, but in our case we used ask with a specific target in order to change the variables for that specific turtle without ever knowing the turtle number or id. The explicit use of the ask command in combination with a specific target can be observed in *Code Blocks 6* and *7* in *Appendix B*.

The nobody keyword was also used in order to ensure that there was actually a turtle found that met the search criteria. The nobody keyword was used as a safeguard, because when no turtle was found errors could occur. The use of the nobody keyword can be observed in *Code Block* 6 in *Appendix* B. We refer to this feature as error-checking or error avoidance, because passing nobody to the program will make the system fail; we wanted to ensure that the simulation would run as long as it could.

***References and Credits***

Büthe, Tim. “Basic Games,” Duke University. <<http://people.duke.edu/~buthe/downloads/teaching/basic_games.pdf>>

DeCanio, Stephen J. and Anders Fremstad. “Game Theory and Climate Diplomacy,” Ecological Economics 85 (2013).

Farahani, Shahrabi M. and M. Sheikhmohammady. “A Review On Symmetric Games: Theory, Comparison and Applications,” International Journal of Applied Operational Research 4, no. 3 (Summer 2014).

Prisner ,Erich. Game Theory through Examples (Washington, DC: The Mathematical Association of America, 2014), 16.

Requejo-Martinez, Ruben J. “Evolutionary Game Theory and the Tower of Babel of Cooperation: Altruism, Free-Riding, Parasitism and the Structure of the Interactions in a World with Finite Resources.,” *Physics of Life Reviews* (December 13, 2013): 10, <<http://arxiv.org/pdf/1312.3447.pdf>>

***Appendix A***

Under the context of immigration into the EU it is interesting to consider the game of “Harmony”, shown below in Figure 3 (Büthe).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Player 2 (Immigrants) | | |
| Player 1  (EU) | Strategy | EU-ID | I-ID |
| EU-ID | 4  4 | 2  3 |
| I-ID | 3  2 | 1  1 |

**Figure [3]**: Harmony Game Space

In this condition, cooperating is the best solution and best rational outcome for both agents, because the Harmony Game maximizes the payoffs that correspond to the same strategies for each player. Since none of the agents will be worse off than the other, the situation does not produce any dilemma for the evolution of the game (Requejo-Martinez, 2013). In fact, when both agents opt to adopt or keep the EU culture, they will each have the highest payoff value; this strategy is known as the dominant strategy.

In the game of “Deadlock” the situation is the exact opposite of the Harmony game; opting to adopt the immigrant culture will earn the highest payoff for both agents (Büthe). Shown below in Figure 4, one can notice how the bottom right corner of the grid will earn both players the highest payoff. Similar to the Harmony game, a dominant strategy arises. A dominant strategy is present when each player chooses the choice for themselves that will have the highest possible outcome (no matter what the other player chooses) (Prisner, 2014).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Player 2 (Immigrants) | | |
| Player 1  (EU) | Strategy | EU-ID | I-ID |
| EU-ID | 1  1 | 3  2 |
| I-ID | 2  3 | 4  4 |

**Figure [4]**: Deadlock Game Space

According to the Prisoner’s Dilemma, both players tend to defect as a form of maximizing their payoffs (DeCanio, 2013). Examining Figure 5 below, one can notice that there is no dominant strategy or Nash Equilibrium in the Prisoner’s Dilemma (Farahani, 2014).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Player 2 (Immigrants) | | |
| Player 1  (EU) | Strategy | EU-ID | I-ID |
| EU-ID | 3  3 | 4  1 |
| I-ID | 1  4 | 2  2 |

**Figure [5]**: Prisoner’s Dilemma Game Space

We simulate the highest payoff in the case where one player defects while the other remains the same. In *Code Block 6* and *7* in *Appendix B*, one can see that the outcome of the agent that defects is dependent upon the proportion of the population that supports each culture.

The coordination game is another technique where no dominant strategy or Nash Equilibrium exists. Figure 6 below shows the possible outcomes for the Coordination game (Büthe).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Player 2 (Immigrants) | | |
| Player 1  (EU) | Strategy | EU-ID | I-ID |
| EU-ID | 1  1 | 0  0 |
| I-ID | 0  0 | 1  1 |

**Figure [6]**: Coordination Game Space

The players will align their beliefs to support either the EU or immigrant culture. The manner in which this solution is simulation is similar to that of the prisoner’s dilemma. A random number is selected based on the proportion of the population that favors a particular culture. The dominant culture is more likely to gain the support of the two players from this game.

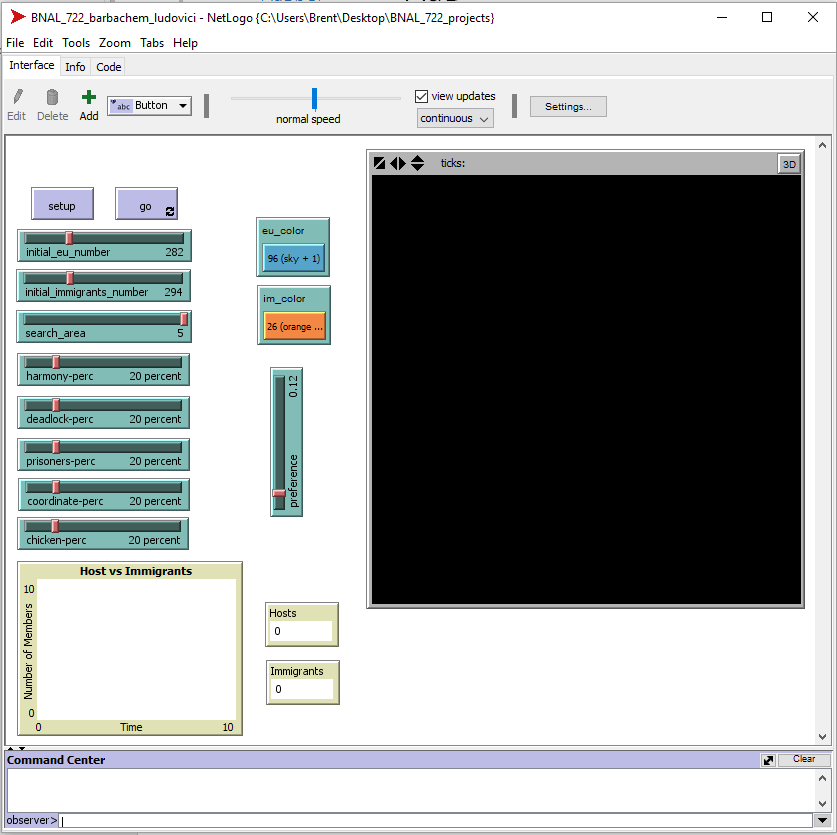
In the game of Chicken the Nash Equilibrium of the game are when one agent cooperates and the other agent does not cooperate (Requejo-Martinez). When one player chooses EU cultural preference, the other agent will in turn support the immigrant culture (Büthe). The top right and bottom left corners of Figure 7, below, are the choices that we simulation. We randomly selected a uniformly distributed number. If the number is less than one half the current agent will support the EU culture, while the target agent will support the immigrant culture. The opposite will remain true when the random number is above 0.5.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Player 2 (Immigrants) | | |
| Player 1  (EU) | Strategy | EU-ID | I-ID |
| EU-ID | 3  3 | 4  2 |
| I-ID | 2  4 | 1  1 |

**Figure [7]**: Chicken Game Space

The strongest and most important assumption that was made about our model was that of rational behavior. All of the outcomes or results for the games that were played or simulated were based on the rational behavior of human agents. The basis for this assumption is two-part. First, the assumption alleviates the need to code all possible outcomes, thus allowing the scope to drift away from the game theory approach. The second reason that we made this assumption is based on the fact over a large number of iterations, a real human would learn and choose the answers that we have explicitly coded into the model.

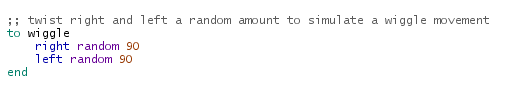
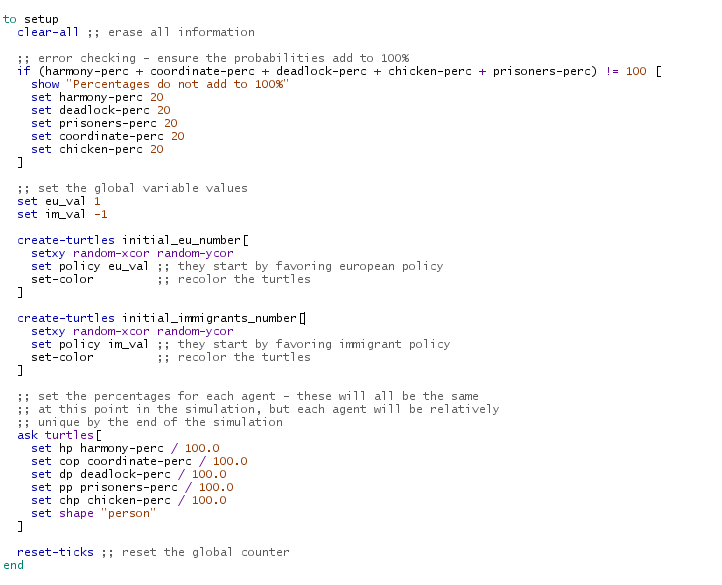
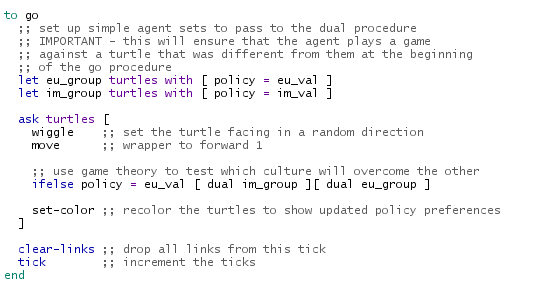
***Appendix B***



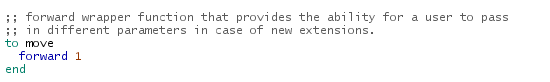
**Figure [8]**: The Netlogo interface tab that was custom designed for our model. The user has the option to alter 11 variables. The user can also monitor the output by examining the plot and two monitors.

**Code Block [1]**: The code represents the setup procedure.

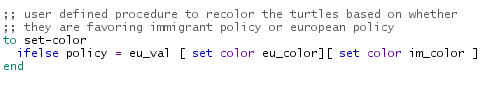
**Code Block [2]**: The code represents the go procedure.



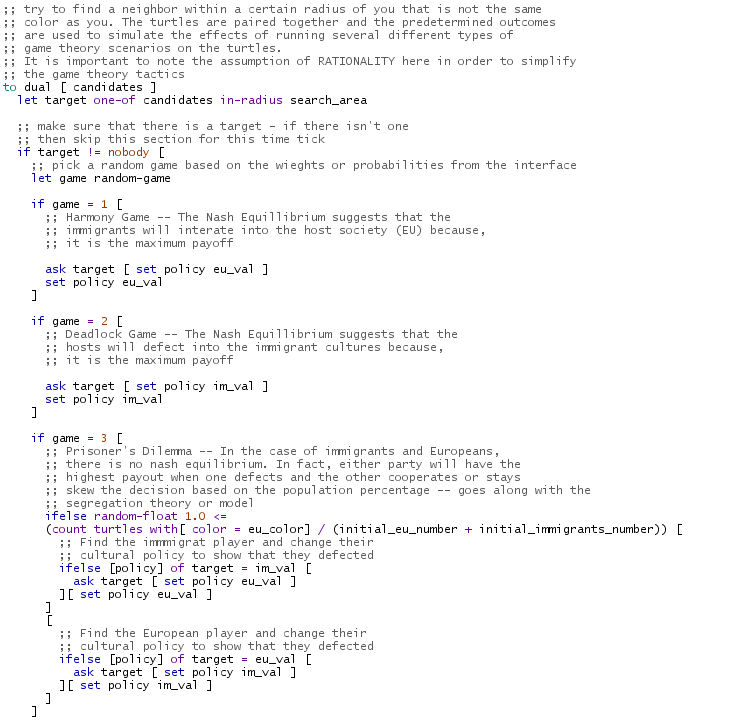
**Code Block [3]**: The code represents the wiggle procedure.



**Code Block [4]**: The code represents the move procedure.

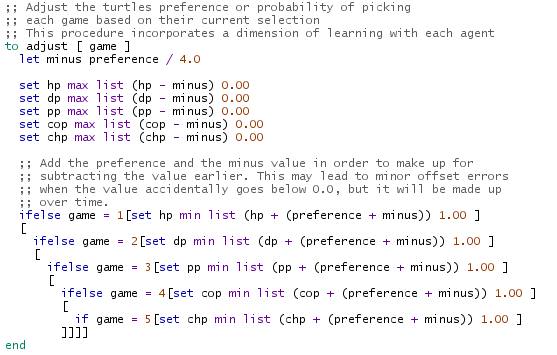
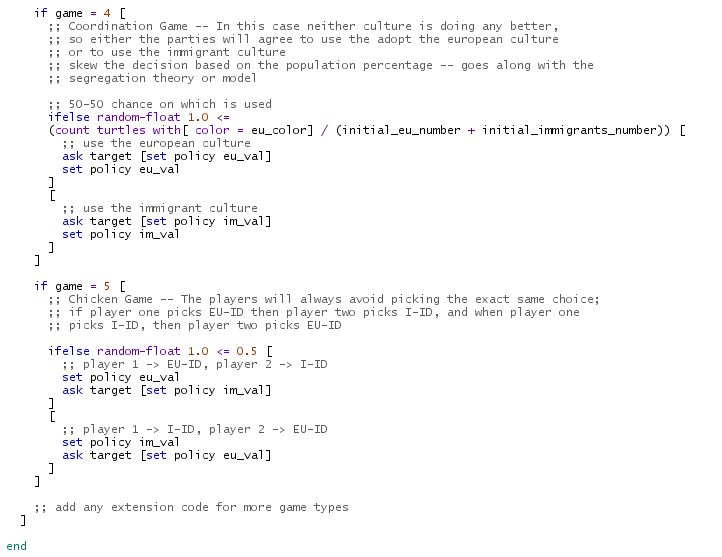


**Code Block [5]**: The code represents the set-color procedure.

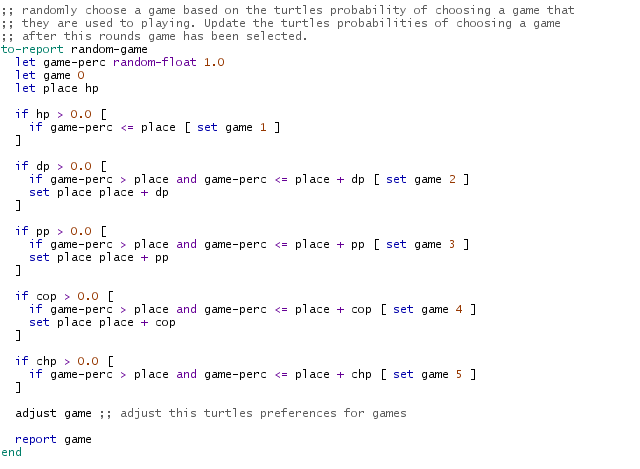


**Code Block [6]**: The code represents half of the dual procedure.

**Code Block [7]**: The code represents half of the dual procedure.



**Code Block [9]**: The code represents the random-game procedure.



**Code Block [8]**: The code represents the adjust procedure.