Alfredo Lorenzo Mendiola

SP&E 316 – Computational and Agent-Based Modeling

Fall 2022

Final Paper

Pseudologia Fantastica

**Introduction**

Deception in an experiment’s design is seen as a taboo practice in Economics and will result in your research being written off. In Psychology, deception in an experiment’s design is accepted if it is deemed necessary for the purpose of the research question. More accurately, “deception in psychological research is often stated as acceptable only when all of the following conditions are met: 1) no other nondeceptive method exists to study the phenomenon of interest; 2) the study makes significant contributions to scientific knowledge; 3) the deception is not expected to cause significant harm or severe emotional distress to research participants; and 4) the deception is explained to participants as soon as the study protocol permits.” (Boynton et al., 2013) Both sides seem to have sound arguments and cases that have been presented. On the Economics side, a major concern is the negative effect on the trust between the subject pool and the researchers. On the Psychology side, the Economist’s concern seems unwarranted since most research subjects are incoming 1st year undergraduate students. Psychologists rebut the Economist’s concern by having a subject with no research experience and who has not been lied to before being the only qualified participants for research. But does the interaction truly end once that student leaves the location of the experiment? Maybe the direct interaction is over, but spillover effects are very much an indirect interaction. These 1st years go on to talk with their friends who then talk to their friends. Some may even become Resident Assistants and have frequent interactions with the following year’s incoming 1st years. Just how much do spillover effects spread into a population? Pseudologia Fantastica is an Agent-Based Model (ABM) that can be used to simulate university environments and the individual trust of the population. It uses 2 main values to gauge the spillover effects, a trust percentage and a lie percentage. Having game theory games be the mode of interaction between agents is similar to human behavior in the real world. The model also has randomness in it to account for error and chaos along with being extremely customizable to adapt to most environments.

**Research Question**

How does deception affect a subject pool’s trust through spillover effects? Using an ABM to answer my question has the benefit of saving time and money. We save both by being able to run a few experiments in different environments to accumulate data and never having to run every possible scenario. If we have enough data from our baseline and variations in scenarios, ABMs can fill in the rest with educated simulations. If the ABM is made well, it can consider chaos that occurs in between scenarios as well. Since most experiments in the field of psychology are done at undergraduate universities, there is already plenty of existing data to use in this model. By creating the ABM well, it can be adapted to most universities and their student environments.

**CONOPS**

Diagram

Description automatically generated

**ODD**

**Purpose:**

To use readily available university data to simulate environments where deception is used and its spillover effects on the subject population.

**Entities, State Variables, and Scales:**

* Agents
  + Liars
    - When they are P1, they will always lie to P2 about what they chose to play
    - Lying is not always part of their strategy to win. They just always lie until given reasons not to lie
  + Non-liars
    - When they are P1, they may or may not lie to P2 about what they chose to play
    - Lying is not always part of their strategy to win. Can be random for middle percentages
  + All agents will have the following individual values for uniqueness and strategy
    - trust – percentage that the agent believes the other agent (P2 play)
    - lie – percentage that the agent lies about what they will play (P1 play)
    - age – number of ticks the agent has been alive for
    - payoff – individual payoff that the agent has accumulated
    - game\_count – number of times the agent has played
    - lied\_to – number of times the agent has been lied to
    - partner – the other agent the agent will be playing the game with
    - player – assigned either P1/P2 for play
    - play\_actual – what the agent will chose to play
    - play\_tell – what the agent tells their partner they will play (P1 play)

**Process Overview and Scheduling:**

The following is a list of the flow of the model. Values are updated in their own code but most of the important ones (lie, trust, lied\_to, etc.) are updated in the payoff sections for each game.

* Setup game
* Setup world
* Go
* Match up
* Play
* Payoffs
* Fix values
* Reset
* Move
* Tick
* Age
* Spawns
* Deaths

**Design Concepts:**

Adaptation comes into effect as the agent’s trust or lie percentages fall or rise. The agents no longer play the same and adjust to how they have been treated by other agents. This leads to interaction between agents being the focus of this model as their individual values adjust as they interact with each other. Stochasticity is used in almost all the inputs that are used from the sliders. If the low end and the high end are not set exactly to each other, then I use the random function to assign values from within that range.

**Initialization:**

To start the first run of the model you need to establish ranges for lie and trust. The others should be left at the baseline to run initially. This will leave the system in an equilibrium but allow you to get used to the model and how lie and trust percentages can affect the equilibrium.

**Input Data:**

The model does not read in data from outside sources but does expect to use data from universities such as admission rates, dropout rates, graduation rates, etc.

**Submodels:**

* Setup game
  + The game you choose to set up with will assign the game’s payoff values into the code and then call the general setup to setup the world
* Setup world
  + Will create the agents and call other chunks of code to assign colors and shapes once values have been randomly assigned
* Go
  + Runs the model until manually stopped or when all agents die
* Match up
  + Has agents look within a radius for a partner who does not already have a partner
* Play
  + For everyone with a partner, 1 agent gets assigned P1 and the other gets assigned P2
  + Has the players make their choices and then calls payoffs for the respective game that the code has been loaded in with
* Payoffs
  + changes values for all agents who played and counters
* Fix values
  + keeps values within a 0-100 percentage range
* Reset
  + clears partners and P1/P2 assignments
* Move
  + Moves agents randomly around the world for the next round
* Tick
  + Keeps track of world time
* Age
  + Keeps track of agent time
* Spawns
  + Based on ticks
  + Spawns new agents every X tick increments
* Deaths
  + Based on ticks and age
  + Kills agents every Y tick increments
  + Kills agents once they have reached a certain age
* Go restarts

**The Model User Interface**

The interface is set up to have the flow of use be from the left to the right. All the inputs are on the left side of the interface. The world and in-play changes are in the center. All the outputs are sorted to the right side. Most of the inputs in the current iteration of this model use 2 sliders to set a low end and upper end of a range.

**Inputs (sliders, setup buttons):**

* **![Graphical user interface

  Description automatically generated]()**initial\_pop
  + Initial number of agents that spawn during setup
* percent\_liars
  + Percentage of initial agents that lie 100% of games
* percent\_liars\_new
  + Percent of new spawn agents that lie 100% of games
* trust\_low / trust\_high
  + Forms the trust percentage range assigned to each agent
* lie\_low / lie\_high
  + Forms the lie percentage range assigned to each agent (excludes liars)
* new\_pop
  + Number of agents that spawn every new\_ticks
* die\_pop
  + Number of agents that die every die\_ticks
* new\_ticks
  + Number of tick increments when new agent spawns occur
* die\_ticks
  + Number of tick increments when agent deaths occur
* move\_range
  + Range that agents can move every tick
* play\_range
  + Range that agents can pair up to play within
* trust\_fall\_range
  + Range of observation of lies where trust percentages fall (non-anonymous play)
* die\_after
  + Number of ticks that an agent will die after
* trust\_fall\_low / trust\_fall\_high
  + Forms the percentage range that trust falls when lied to
* trust\_rise\_low / trust\_rise\_high
  + Forms the percentage range that trust rises when told the truth
* lie\_fall\_low / lie\_fall\_high
  + Forms the percentage range that lie falls when lies are unsuccessful
* lie\_rise\_low / lie\_rise\_high
  + Forms the percentage range that lie rises when lies are successful
* Multiple setup buttons
  + Each setup button calls a different chunk of code to use different games in play

**![Graphical user interface

Description automatically generated with low confidence]()In-Play (world, go button, switches):**

* World
  + Left at 32x32 as default but can be adjusted easily if deemed necessary for environments
* Go
  + Runs the model forever until stopped manually or all agents die
* show\_lie?
  + Show’s an agent’s lie percentage as a label when switched on
* non\_anon\_play?
  + Removes anonymous play and creates area spillover effects when switched on

**Outputs (monitors, plots):![A picture containing calendar

Description automatically generated]()**

* Population
  + Current total number of agents alive in world
* Liars
  + Current total number of agents with 100% lie
* Highest Trust / Average Trust / Lowest Trust
  + 3 monitors to keep track of trust percentage values for QC and information
* Highest Lie / Average Lie / Lowest Lie
  + 3 monitors to keep track of lie percentage values for QC and information
* Highest Payoff / Average Payoff / Lowest Payoff
  + 3 monitors to keep track of agent’s payoff values for QC and information
* Total Payoff
  + The total payoff accumulated by all current agents in models
* Average Age
  + The average ticks that the agents in the model have been alive for
* Games Played
  + Histogram for the number of games agents have played, broken into 25 4% segments
* Times Lied To
  + Histogram for the number of times an agent has been lied to, broken into 25 4% segments
* Trust v Lie
  + Plot for average trust percentage and average lie percentage over time elapsed
* Trust t Lie
  + Phase Portrait for average trust percentage by average lie percentage
* Trust Spread
  + Histogram for trust percentages in 10 10% segments
* Lie Spread
  + Histogram for lie percentages in 10 10% segments
* Wealth Spread
  + Histogram for number of agents by individual accumulated payoffs in 10 10% segments

**Baseline, Simulation and Scenarios**

![Graphical user interface

Description automatically generated]()The following parameters are for the baseline run of the model. 5 percent was chosen for percent\_liars because according to Dr. Paul Ekman, less than 5 percent of the population are pathological liars. lie\_low and lie\_high were set to 7 percent because “lying comprised 7% of total communication and almost 90% of all lies were little white lies”. (Docan-Morgan, 2021) The rest of the inputs were chosen intuitively.

The parameter inputs on the left were chosen because increasing the percentage of the population who are liars should intuitively drop trust in the population drastically. This did not seem to be the case when it was run a few times. Instead, the trust and lie stayed around 50/50 and stable in the equilibrium behavior.

The parameter inputs on the right are for another 50/50 stable equilibrium but with changing the initial lie percentage range to match that of the trust percentage.

![Graphical user interface

Description automatically generated]()![Graphical user interface

Description automatically generated]()

**Data Analysis on Results**

The following are linear regressions for the dependent variables, mean lie percentage and mean trust percentage, using the control variables lie\_fall\_high, lie\_rise\_high, and lie\_high. The low end of all 3 of these ranges were set to 0% and the upper end was started low to get a full range.

##   
## Call:  
## lm(formula = df$mean..trust..of.turtles ~ df$lie\_fall\_high +   
## df$lie\_rise\_high + df$lie\_high)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -74.149 -14.907 -1.089 14.625 60.717   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 84.213397 0.454173 185.42 <2e-16 \*\*\*  
## df$lie\_fall\_high 4.063337 0.012138 334.75 <2e-16 \*\*\*  
## df$lie\_rise\_high -4.275662 0.012138 -352.25 <2e-16 \*\*\*  
## df$lie\_high -0.298425 0.006069 -49.17 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.81 on 76860 degrees of freedom  
## Multiple R-squared: 0.7563, Adjusted R-squared: 0.7563   
## F-statistic: 7.952e+04 on 3 and 76860 DF, p-value: < 2.2e-16

##   
## Call:  
## lm(formula = df$mean..lie..of.turtles ~ df$lie\_fall\_high + df$lie\_rise\_high +   
## df$lie\_high)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -49.977 -7.179 -0.537 5.965 47.132   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 16.508671 0.221813 74.43 <2e-16 \*\*\*  
## df$lie\_fall\_high -2.940835 0.005928 -496.08 <2e-16 \*\*\*  
## df$lie\_rise\_high 3.149789 0.005928 531.32 <2e-16 \*\*\*  
## df$lie\_high 0.322283 0.002964 108.73 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9.188 on 76860 degrees of freedom  
## Multiple R-squared: 0.8754, Adjusted R-squared: 0.8754   
## F-statistic: 1.801e+05 on 3 and 76860 DF, p-value: < 2.2e-16

The following are linear regressions for the dependent variables, mean lie percentage and mean trust percentage, using the control variables trust\_fall\_high, trust\_rise\_high, and trust\_high. The low end of all 3 of these ranges were set to 0% and the upper end was started low to get a full range.

##   
## Call:  
## lm(formula = df$mean..trust..of.turtles ~ df$trust\_fall\_high +   
## df$trust\_rise\_high + df$trust\_high)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -89.084 -7.118 -0.623 6.816 52.102   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 32.611415 0.159739 204.2 <2e-16 \*\*\*  
## df$trust\_fall\_high -3.436169 0.007258 -473.4 <2e-16 \*\*\*  
## df$trust\_rise\_high 4.062128 0.007258 559.7 <2e-16 \*\*\*  
## df$trust\_high 0.133832 0.001434 93.3 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 11.48 on 80076 degrees of freedom  
## Multiple R-squared: 0.8721, Adjusted R-squared: 0.8721   
## F-statistic: 1.82e+05 on 3 and 80076 DF, p-value: < 2.2e-16

##   
## Call:  
## lm(formula = df$mean..lie..of.turtles ~ df$trust\_fall\_high +   
## df$trust\_rise\_high + df$trust\_high)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.6824 -1.1481 0.2676 1.4721 7.2014   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 48.3816024 0.0314205 1539.81 <2e-16 \*\*\*  
## df$trust\_fall\_high 0.2218209 0.0014276 155.38 <2e-16 \*\*\*  
## df$trust\_rise\_high -0.2118301 0.0014276 -148.38 <2e-16 \*\*\*  
## df$trust\_high -0.0142127 0.0002822 -50.37 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.258 on 80076 degrees of freedom  
## Multiple R-squared: 0.3781, Adjusted R-squared: 0.3781   
## F-statistic: 1.623e+04 on 3 and 80076 DF, p-value: < 2.2e-16

![Diagram

Description automatically generated]()![A picture containing diagram

Description automatically generated]()![Diagram

Description automatically generated]()All 4 regressions have every coefficient as being extremely significant. This is due to the current setup of the games not having true game theory equations yet. Once the equations are tailored to the model and games, the analysis will be able to produce actual results. For now, it is here to show experience in the process of analysis and not to be taken as productive results.

The 3 phase portraits above show 3 different ranges in lie\_rise\_low/lie\_rise\_high percentages. All 3 ranges were extremely similar (1-5, 5-10, 3-5) and yielded similar looking equilibrium behaviors but at drastically different percentage value combinations. As more changes are made in the model, the equilibrium behavior seems to keep the tail from its initial starting position and then moves around a concentrated set of values.

**![Graphical user interface, application

Description automatically generated]()![Graphical user interface, text, application, email

Description automatically generated]()Strategy**

I ran the Behavior Search with the parameters on the left. I placed all the low ends of the ranges to 0 to use the high ends to sift from low to high values. The graph on the right is what was found for the best fit to run the model on.

**Conclusion Remarks**

The model is built well to handle multiple 2 person games by using setups to assign payoff values to global values. The model also uses inputs with very customizable ranges and wide range of control variables that can be adjusted to adapt to a variety of environments. Pseudologia Fantastica was built for university environments specifically, but with the ability to add a little extra code, it can easily be adjusted to model any environment. Currently the model is designed to be used with 2 person games with non-simultaneous moves. P1 chooses first and P2 has imperfect information when they make their choice. An update to the model to allow for more deception choice games could be useful to make it more accurate to human behavior.

The spawns and movement in this version of the model are randomized and limited to a range. If an environment warranted segregation of agents or clustering to simulate population density, it could easily be added in using spawns and movement restrictions. One such use for university environments would be to simulate a campus map and movement of the agent’s day to day. The agents could spawn in clusters (dorms, houses, etc.) and move in set paths (to classes, food, etc.) to see how the spillover effects would take place.

The next iteration of this model will not only have setups for games but will have setups for different campuses and use data from admissions, registrar, etc. to tune the inputs. This will allow it to have a more applicable use and leave less room for rebuttals on the results. With the setup buttons for the different games, formulas will need to be found and added to make the model as accurate as possible.

Once the game formulas are added in correctly, the analysis can be rerun to see which control variables are still significant and which ones do not have as large of an effect as they currently do. Without proper formulas to emulate human behavior, the model is essentially a piece of art. Cool to look at, possibly expensive, and without actual use.

The current model allows for non-anonymous play to be turned off and on during the running of the model. Every instance that it is turned on, the world’s average trust percentage plummets to 0%. Once anonymous play is turned back on, the average trust percentage will sometimes stay low but usually will make its way back to the equilibrium. I believe this steep decline is due to how the model is currently coded. With the trust percentage falling in a radius of a lie being told, some agents may have their trust percentage impacted multiple times in 1 round if they are within the radius of multiple lies. A limit could be put into account for this but that would seem counter intuitive. In reality, if someone sees multiple people around them lying, their trust in general will decrease more than if it had just been 1 person lying.

Ultimately, with a little updating, this model will be able to show if deception in experiments would affect a population negatively in the long run. Then we could use the results to decide if we should still run the experiment or change the design.

**References**

* Boynton MH, Portnoy DB, Johnson BT. Exploring the ethics and psychological impact of deception in psychological research. IRB. 2013 Mar-Apr;35(2):7-13. PMID: 23672145; PMCID: PMC4502434
* Docan-Morgan 2021 [https://www.uwlax.edu/post/fbff1575-62fa-4f32-af24-554b24823f86

**Appendix**

**Model code:**

; Alfredo Lorenzo Mendiola  
; SP&E 316 Final Project  
; 29Dec2022  
; Pseudologia Fantastica  
; (Pathological Lying)  
  
globals [ ; see ppt for details  
 game  
 A1  
 A2  
 B1  
 B2  
 C1  
 C2  
 D1  
 D2  
]  
  
turtles-own [ ; see ppt for details  
 trust  
 lie  
 age  
 payoff  
 game\_count  
 lied\_to  
 partner  
 player  
 play\_actual  
 play\_tell  
]  
  
to setup\_p\_d ; Prisoner's Dilemma setup  
 setup  
  
 ask turtles [  
 set game "PD"  
 set A1 -6  
 set A2 0 ; --- ----------------- -----------------  
 set B1 -10 ; | | C | D |  
 set B2 -1 ; --- ----------------- -----------------  
 set C1 -6 ; | A | (-6[A1],-6[C1]) | (0[A2],-10[D1]) |  
 set C2 0 ; --- ----------------- -----------------  
 set D1 -10 ; | B | (-10[B1],0[C2]) | (-1[B2],-1[D2]) |  
 set D2 -1 ; --- ----------------- -----------------  
 ]  
end  
  
to setup\_g\_e ; Gift Exchange setup  
 setup  
  
 ask turtles [  
 set game "GE"  
 set A1 2  
 set A2 0 ; --- --------------- ---------------  
 set B1 3 ; | | C | D |  
 set B2 1 ; --- --------------- ---------------  
 set C1 2 ; | A | (2[A1],2[C1]) | (0[A2],3[D1]) |  
 set C2 0 ; --- --------------- ---------------  
 set D1 3 ; | B | (3[B1],0[C2]) | (1[B2],1[D2]) |  
 set D2 1 ; --- --------------- ---------------  
 ]  
end  
  
to setup  
 clear-all  
 reset-ticks  
  
 create-turtles initial\_pop [  
 setxy random-xcor random-ycor  
 set trust trust\_low + (random (1 + trust\_high - trust\_low))  
 set lie lie\_low + (random (1 + lie\_high - lie\_low))  
 set age 0  
 set payoff 0  
 set game\_count 0  
 set lied\_to 0  
 set partner nobody  
 set player 0  
 set play\_actual ""  
 set play\_tell ""  
 ]  
  
 liars  
  
 ask turtles [  
 agent\_color\_shape  
 ]  
end  
  
to go  
 if not any? turtles [ ; ends model if all turtles die  
 stop  
 ]  
  
 match\_up  
 play  
 move  
 tick  
  
 ask turtles [  
 set age age + 1  
 ]  
  
 new\_spawns  
 die\_spawns  
end  
  
to play ; plays the model  
 ask turtles with [partner != nobody and player = 0] [ ; assigns P1 and P2  
 set player 1  
 ask partner [  
 set player 2  
 ]  
 ]  
  
 if game = "PD" [ ; Prisoner's Dilemma  
 p\_d  
 ]  
 if game = "GE" [ ; Gift exchange  
 g\_e  
 ]  
  
 ask turtles [  
 agent\_color\_shape  
 ]  
  
 reset  
end  
  
to p\_d ; play for Prisoner's Dilemma setup  
 ask turtles with [player = 1] [ ; P1 plays  
 if lie >= 80 [ ; always lies plays A  
 set play\_tell "B"  
 set play\_actual "A"  
 ]  
 if lie < 80 and lie > 20 [ ; 50/50 lie or truth to play A/B  
 set play\_tell "B"  
 ifelse (random 100 + 1) mod 2 = 0 [  
 set play\_actual "A"  
 ][  
 set play\_actual "B"  
 ]  
 ]  
 if lie <= 20 [ ; always truth plays B  
 set play\_tell "B"  
 set play\_actual "B"  
 ]  
 set game\_count game\_count + 1  
 ]  
  
 ask turtles with [player = 2] [ ; P2 play  
 if [partner] of partner = self and [play\_tell] of partner = "B" [ ; P1 told P2 B  
 if trust >= 70 [ ; P2 believes P1 so plays D  
 set play\_actual "D"  
 ]  
 if trust < 70 and trust > 30 [ ; 50/50 trusts/distrusts plays C/D  
 ifelse (random 100 + 1) mod 2 = 0 [  
 set play\_actual "D"  
 ][  
 set play\_actual "C"  
 ]  
 ]  
 if trust <= 30 [ ; P2 distrusts P1 so plays C  
 set play\_actual "C"  
 ]  
 ]  
 set game\_count game\_count + 1  
 ]  
  
 payoffs\_pd  
end  
  
to payoffs\_pd ; checks outcome and distributes payoffs for PD  
 ask turtles with [player = 1] [  
 if play\_tell = "B" and play\_actual = "A" [ ; outcomes for lying  
 ask partner [ ; increase times lied to for P2  
 set lied\_to lied\_to + 1  
 ]  
 if non\_anon\_play? [ ; for non-anonymous play switch  
 ask turtles in-radius trust\_fall\_range [ ; trust falls in a radius of lying P1  
 set trust trust - (trust\_fall\_low + (random (1 + trust\_fall\_high - trust\_fall\_low)))  
 ]  
 ]  
 if [play\_actual] of partner = "C" [ ; AC  
 set payoff payoff + A1  
 set lie lie - (lie\_fall\_low + (random (1 + lie\_fall\_high - lie\_fall\_low)))  
 ask partner [  
 set payoff payoff + C1  
 set trust trust - (trust\_fall\_low + (random (1 + trust\_fall\_high - trust\_fall\_low)))  
 ]  
  
 ]  
 if [play\_actual] of partner = "D" [ ; AD  
 set payoff payoff + A2  
 set lie lie + (lie\_rise\_low + (random (1 + lie\_rise\_high - lie\_rise\_low)))  
 ask partner [  
 set payoff payoff + D1  
 set trust trust - (trust\_fall\_low + (random (1 + trust\_fall\_high - trust\_fall\_low)))  
 ]  
 ]  
 ]  
 if play\_tell = "B" and play\_actual = "B" [ ; outcomes for truth  
 if [play\_actual] of partner = "C" [ ; BC  
 set payoff payoff + B1  
 set lie lie + (lie\_rise\_low + (random (1 + lie\_rise\_high - lie\_rise\_low)))  
 ask partner [  
 set payoff payoff + C2  
 set trust trust + (trust\_rise\_low + (random (1 + trust\_rise\_high - trust\_rise\_low)))  
 ]  
 ]  
 if [play\_actual] of partner = "D" [ ; BD  
 set payoff payoff + B2  
 set lie lie - (lie\_fall\_low + (random (1 + lie\_fall\_high - lie\_fall\_low)))  
 ask partner [  
 set payoff payoff + D2  
 set trust trust + (trust\_rise\_low + (random (1 + trust\_rise\_high - trust\_rise\_low)))  
 ]  
 ]  
 ]  
 ]  
  
 fix  
end  
  
to g\_e ; play for Gift Exchange setup  
 ask turtles with [player = 1] [ ; P1 plays  
 if lie >= 80 [ ; always lies plays B  
 set play\_tell "A"  
 set play\_actual "B"  
 ]  
 if lie < 80 and lie > 20 [ ; 50/50 lie or truth to play A/B  
 set play\_tell "A"  
 ifelse (random 100 + 1) mod 2 = 0 [  
 set play\_actual "A"  
 ][  
 set play\_actual "B"  
 ]  
 ]  
 if lie <= 20 [ ; always truth plays A  
 set play\_tell "A"  
 set play\_actual "A"  
 ]  
 set game\_count game\_count + 1  
 ]  
  
 ask turtles with [player = 2] [ ; P2 play  
 if [partner] of partner = self and [play\_tell] of partner = "A" [ ; P1 told P2 B  
 if trust >= 70 [ ; P2 believes P1 so plays D to maxi-max  
 set play\_actual "D"  
 ]  
 if trust < 70 and trust > 30 [ ; 50/50 trusts/distrusts plays C/D  
 ifelse (random 100 + 1) mod 2 = 0 [  
 set play\_actual "D"  
 ][  
 set play\_actual "C"  
 ]  
 ]  
 if trust <= 30 [ ; P2 distrusts P1 so plays D to maxi-min  
 set play\_actual "D"  
 ]  
 ]  
 set game\_count game\_count + 1  
 ]  
  
 payoffs\_ge  
end  
  
to payoffs\_ge ; checks outcome and distributes payoffs for GE  
 ask turtles with [player = 1] [  
 if play\_tell = "A" and play\_actual = "B" [ ; outcomes for lying  
 ask partner [ ; increase times lied to for P2  
 set lied\_to lied\_to + 1  
 ]  
 if non\_anon\_play? [ ; for non-anonymous play switch  
 ask turtles in-radius trust\_fall\_range [ ; trust falls in a radius of lying P1  
 set trust trust - (trust\_fall\_low + (random (1 + trust\_fall\_high - trust\_fall\_low)))  
 ]  
 ]  
 if [play\_actual] of partner = "C" [ ; BC  
 set payoff payoff + B1  
 set lie lie + (lie\_rise\_low + (random (1 + lie\_rise\_high - lie\_rise\_low)))  
 ask partner [  
 set trust trust - (trust\_fall\_low + (random (1 + trust\_fall\_high - trust\_fall\_low)))  
 set payoff payoff + C2  
 ]  
  
 ]  
 if [play\_actual] of partner = "D" [ ; BD  
 set lie lie - (lie\_fall\_low + (random (1 + lie\_fall\_high - lie\_fall\_low)))  
 set payoff payoff + B2  
 ask partner [  
 set payoff payoff + D2  
 set trust trust - (trust\_fall\_low + (random (1 + trust\_fall\_high - trust\_fall\_low)))  
 ]  
 ]  
 ]  
 if play\_tell = "A" and play\_actual = "A" [ ; outcomes for truth  
 if [play\_actual] of partner = "C" [ ; AC  
 set payoff payoff + A1  
 set lie lie - (lie\_fall\_low + (random (1 + lie\_fall\_high - lie\_fall\_low)))  
 ask partner [  
 set trust trust + (trust\_rise\_low + (random (1 + trust\_rise\_high - trust\_rise\_low)))  
 set payoff payoff + C1  
 ]  
 ]  
 if [play\_actual] of partner = "D" [ ; AD  
 set payoff payoff + A2  
 set lie lie + (lie\_rise\_low + (random (1 + lie\_rise\_high - lie\_rise\_low)))  
 ask partner [  
 set payoff payoff + D1  
 set trust trust + (trust\_rise\_low + (random (1 + trust\_rise\_high - trust\_rise\_low)))  
 ]  
 ]  
 ]  
 ]  
  
 fix  
end  
  
to fix ; keeps changing values within 0 to 100  
 ask turtles [  
 if trust > 100 [  
 set trust 100  
 ]  
 if trust < 0 [  
 set trust 0  
 ]  
 if lie > 100 [  
 set lie 100  
 ]  
 if lie < 0 [  
 set lie 0  
 ]  
 ]  
end  
  
to reset ; resets agents for new round play  
 ask turtles [  
 set partner nobody  
 set player 0  
 set play\_actual ""  
 set play\_tell ""  
 ]  
end  
  
to match\_up ; partners up agents within play\_range  
 ask turtles [  
 if any? turtles in-radius play\_range [ ; looks for agents in play\_range  
 if (partner = nobody) and (any? other turtles in-radius play\_range with [partner = nobody]) [ ; checks if both agents are unpartnered  
 set partner one-of other turtles in-radius play\_range with [partner = nobody] ; partners current agent with the other one selected  
 ask partner [ ; sets current agent as partner for other agent  
 set partner myself  
 ]  
 ]  
 ]  
 ]  
end  
  
to die\_spawns ; checks if it is time to kill agents  
 if ticks mod die\_ticks = 0 [  
 if die\_pop > count turtles [  
 set die\_pop count turtles  
 ]  
 ask n-of die\_pop turtles [ ; kills agents  
 die  
 ]  
 ]  
 ask turtles [  
 if age >= die\_after [  
 die  
 ]  
 ]  
end  
  
to new\_spawns ; checks if it is time to spawn new agents  
 if ticks mod new\_ticks = 0 [  
 new\_spawns\_true  
 ]  
end  
  
to new\_spawns\_true ; used to spawn new agents normal/liars  
 create-turtles (new\_pop \* ((100 - percent\_liars) / 100)) [ ; spawns new non-liars  
 setxy random-xcor random-ycor  
 set trust trust\_low + (random (trust\_high - trust\_low))  
 set lie lie\_low + (random (lie\_high - lie\_low))  
 set age 0  
 set payoff 0  
 set game\_count 0  
 set lied\_to 0  
 set partner nobody  
 set player 0  
 set play\_actual ""  
 set play\_tell ""  
 agent\_color\_shape  
 ]  
  
 create-turtles (new\_pop \* (percent\_liars\_new / 100)) [ ; spawns new liars  
 setxy random-xcor random-ycor  
 set trust trust\_low + (random (trust\_high - trust\_low))  
 set lie 100  
 set age 0  
 set payoff 0  
 set game\_count 0  
 set lied\_to 0  
 set partner nobody  
 set player 0  
 set play\_actual ""  
 set play\_tell ""  
 agent\_color\_shape  
 ]  
end  
  
to liars ; creates initial liars  
 ask n-of (count turtles \* (percent\_liars / 100)) turtles [  
 set lie 100  
 ]  
end  
  
to move ; moves agents in range from 0 to move\_range  
 ask turtles [  
 fd random (move\_range + 1)  
 ]  
end  
  
to agent\_color\_shape ; sets shape/colors/labels for agents  
 ifelse lie >= 100 [  
 set shape "flag" ; sets shape for liars  
 ] [  
 set shape "person" ; sets shape for non-liars  
 ]  
  
 if trust >= 0 and trust < 20 [ ; low low trust  
 set color 15  
 ]  
 if trust >= 20 and trust < 40 [ ; high low trust  
 set color 17  
 ]  
 if trust >= 40 and trust <= 60 [ ; neutral trust  
 set color 9.9  
 ]  
 if trust > 60 and trust <= 80 [ ; low high trust  
 set color 57  
 ]  
 if trust > 80 and trust <= 100 [ ; high high trust  
 set color 55  
 ]  
  
 ifelse show\_lie? [  
 set label lie ; displays lie % over every agent  
 ] [  
 set label "" ; displays no lie %  
 ]  
end