## WHAT IS IT?

This is a model of a game of Telephone (also known as Chinese Whishpers in the UK), with agents representing people that can be asked, to play. The first player selects a word from their internal vocabulary and "whispers" it to the next player, who may mishear it depending on the current noise level, who whispers that word to the next player, and so on.

When the game ends, the word chosen by the first player is compared to the word heard by the last player. If they match exactly, all players earn large prize. If the words do not match exactly, a small prize is awarded to all players for each part of the words that do match. Players change color to reflect their current prize-count. A histogram shows the distribution of colors over all the players.

The user can decide on factors like

\* how many players there are,

\* whether they are laid out in a circle or just randomly,

\* how many players participate in a game,

\* whether to apply noise-distortion or not,

\* at what decibel level noise distortian starts interfering with the game,

\* how the first player to participate is chosen,

\* how further players are chosen, and

\* whether or not the games run quickly and silently or slowly and with commentary to explain what is happening.

These factors influence how likely players are to win a game and thus how the color of players will be distributed over time.

## HOW IT WORKS

The world has dimensions and also a noise level that moves up and down randomly at each time step, but cannot fall below 0. When the model is initiated, a number of characters are laid out across the dimensions according to a modeller input, which appears in the interface as "layout-style" and gives the options of random or circle. When created, all agents have a vocabulary (set in the code) of several 2 character "words" and an alphabet consisting of all the characters that appear in any position of any of the words in their vocabulary.

The first player is chosen according to a modeller input, which appears on the interface as "choose-starter" which gives the options of randomly, most prizes and least prizes (self-explanatory). That agent randomly selects one of the words in their vocabulary.

The next player is chosen according to a modeller input, which appears on the interface as "choose-next" which gives the options of randomly, most prizes, least prizes, nearest and near-ish (self-explanatory?). The first player "whispers" their chosen word to the next player, who will hear it correctly if the noise-level is below the "Appropriote-noise-level" as set by the modeller. If the noise level is above "Appropriote-noise-level"then a small test is performed for each part of the word, with a chance that the listener may mishear some but not all of the sounds. The listener then becomes the whisperer and the process is repeated until the number of players reaches the "game-length" as set by the modeller. At that point, the game ends, the word chosen by the first player and the word heard by the last player are compared and prizes are awarded.

Each player in the game earns a 5 point prize if the two words match exactly. Thus, [a a a] and [a a a] earn 5 points for all players.

Each player in the game earns a 1 point prize for each phoneme that matches when the two words do not match exactly. Thus, a maximum of 2 points can be awarded for partially matched words, so [a a a] and [a a b] would earn 2 points but [a a a] and [a b b] would earn 1 point.

Players do not earn any points for words that have no matching phoneme-positions in common. Thus, [a a a] and [b b b] would earn 0 points, as would [a b a] and [b a b].

After the game ends and any prizes are awarded, all agents lay themselves out again according to the modeller designated layout-style and adjust their color to reflect their current prize-count. 0 prizes = dark grey (color value 2) but players that have won a prize for exactly matching the first and final words would become light grey (color value 7). If they won again, they would be dark red (color value 12), etc. Any player whose prize count ends in 0 (such as 10, 20, 150, etc.) would become invisible against a black background, so their prize count is increased by 2.

A histogram shows the distribution of colors over all the players as a proxy for showing the distribution of prize counts.

## HOW TO USE IT

Adjusting the layout-style has no effect on the model functioning. This is purely an option to demonstrate the visual capabilities of NetLogo.

Similarly, adjusting Garrolous? To Yes or No has no real effect on the model functioning, although it does make the model run much slower and with much more writing appearing in the Command Center bar at the bottom of the screen. This is useful for those unfamiliar with the model to gain an understanding of the steps taken by the model. It is also useful for bug-testing during model development.

Adjusting the other sliders and switches DOES affect the model functioning. Choices about how

\* to select the first player or the subsequent players can skew the prize distribution.

\* long a game lasts can affect the chance that players will get a perfect match between first and final words.

\* to apply noise-distortion, and what level of noise is appropriate, influence the likelihood of earning (big) prizes.

\* many agents are available to play can effect how long it takes for feedback loops and large scale dynamics to display.

These are not always straightforward and can interact. For example, setting a long game means more players can win prizes but makes it much harder to win prizes if noise distortion is applied when Appropriate-noise-level is set low.

## THINGS TO NOTICE

What shape is the histogram? Does this change over time? Does it depend on the way the sliders and swiches are set?

## THINGS TO TRY

Try varying all of the switches and sliders in various combinations (excluding layout-style and Garrolous?, which only affect the modeller's experience of the model while it is running).

## EXTENDING THE MODEL

Add more words to the vocabulary.

Extend the code to make the words longer.

Adjust the prizes awarded for getting perfect and/or partial matches.

Adjust the likelihood that players can correctly hear/mishear a word when the noise-level is above the Appropriate-noise-level.

## NETLOGO FEATURES

## CREDITS AND REFERENCES

## HOW TO CITE

If you mention this model or the NetLogo software in a publication, we ask that you include the citations below.

For the model itself:

\* Kasmire, J. (2020). Telephone Game. UK Data Services and University of Manchester, UK.

Please cite the NetLogo software as:

\* Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

## COPYRIGHT AND LICENSE

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License. To view a copy of this license, visit https://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Commercial licenses are also available. To inquire about commercial licenses, please contact Uri Wilensky at uri@northwestern.edu.

This model was created as part of the project: CONNECTED MATHEMATICS: MAKING SENSE OF COMPLEX PHENOMENA THROUGH BUILDING OBJECT-BASED PARALLEL MODELS (OBPML). The project gratefully acknowledges the support of the National Science Foundation (Applications of Advanced Technologies Program) -- grant numbers RED #9552950 and REC #9632612.

This model was converted to NetLogo as part of the projects: PARTICIPATORY SIMULATIONS: NETWORK-BASED DESIGN FOR SYSTEMS LEARNING IN CLASSROOMS and/or INTEGRATED SIMULATION AND MODELING ENVIRONMENT. The project gratefully acknowledges the support of the National Science Foundation (REPP & ROLE programs) -- grant numbers REC #9814682 and REC-0126227. Converted from StarLogoT to NetLogo, 2001.