

## GameBug 1.0 User Manual

Robert Wytttenbach, Cornell University

### Getting Started

To start with an empty game, double-click GameBug.exe. The Examples folder contains several pre-defined games. To open one, double-click its document icon (shown at the right). GameBug documents have the extension .gbg.



**parental care**

File Edit View Windows Help

**Matrix** Results Notes

**1. Select Game Type**  
asymmetric 2x2

**2. Define Roles**  
#1 Male #2 Female

**3. Define Strategies**  
Male: Caring Deserting  
Female: Caring Deserting

**4. Define Variables**  
t two-parent care s single-parent care n no-parent care  
m male desertion f female desertion

**5. Create Payoff Matrix**

**Payoffs for Male**

	Caring	Deserting
Caring	t	s
Deserting	s+m	n+m

**Payoffs for Female**

	Caring	Deserting
Caring	t	s
Deserting	s+f	n+f

payoff for Deserting Male against Deserting Female

**Figure 1.** Defining a matrix.

### Defining a Game

A game opens on the *Matrix* tab, in which you may define a game or view the definition of an existing game (Fig. 1, above). The parts are as follows.

1. *Select Game Type.* Use the popup menu to determine whether the game is symmetric or asymmetric and set the number of strategies. In a symmetric game, all players have the same fitness payoffs. In an asymmetric game, players have different roles with different fitness payoffs. See the examples below.
2. *Define Roles.* For an asymmetric game, give the names of the two roles. These names will label the payoff matrices and strategies in the simulation. This section is not used in symmetric games, in which there is only one role.

3. *Define Strategies.* Give names of the strategies available to each player. The popup menu defines the color for each strategy, which will label the payoff matrices and simulation graphs. In asymmetric games, each role has its own set of strategies.
4. *Define Variables.* Payoffs can be determined by up to eight variables. Use the popup menu to create a variable (all variables are represented by single letters), and then describe it in the adjacent box. To delete a variable, select the first (blank) line in the popup menu beside it.
5. *Create Payoff Matrix.* Setting the game type creates a grid for the number of roles and strategies involved. Payoff cells may contain algebraic expressions containing numbers and variables. Use parentheses as needed to group expressions. You can also use  $\ln()$  for natural log and  $^$  for powers. If a cell is not algebraically valid, it will be marked in red and will be treated as zero when the simulation is run. To be reminded of the meaning of a cell in the grid, hold the mouse over it and message describing it will appear. Once the matrix is filled in, go to the *Simulation* tab to set variable values and watch the game play out.

If you open an existing game file, all of the above will already be filled in. You may then modify the game or proceed to the *Simulation* tab to view the outcome. The parental care game in the Examples folder illustrates an asymmetric game; all the other examples are symmetric games.

### Seeing the Results

To view the outcome of the game, select the *Simulation* tab (Fig. 2, below). At the top is a graph of the results. Around the graph are several sections that control the simulation: *Results.* The graph shows how each strategy fares over time. The number of generations shown (up to 16000) is controlled by the popup menu below the graph; some games take a long time to reach equilibrium. The proportion of the population playing each strategy can be shown either as area on the graph (first ) or as a curve ranging from 0 to 1 (below right). This is set by the buttons below the graph.

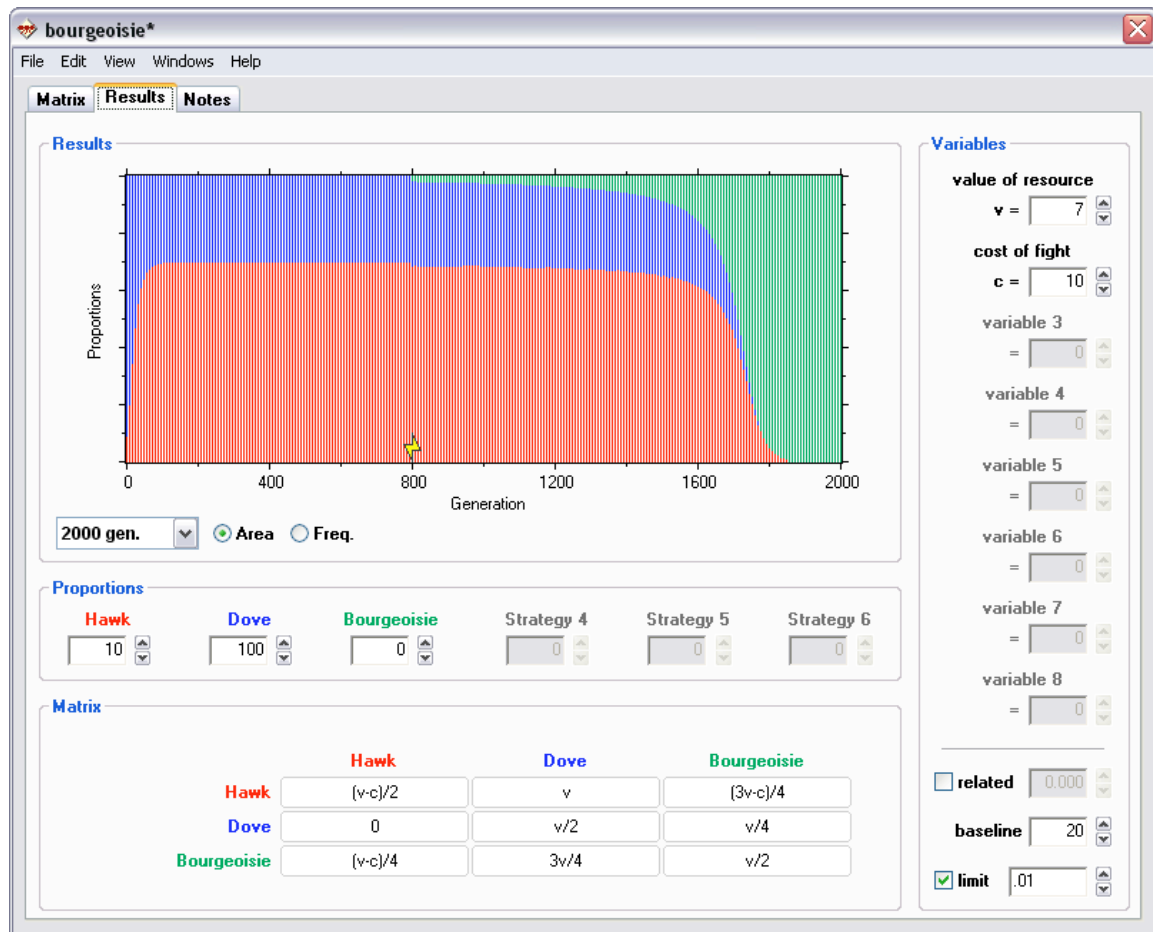
*Mutations.* To introduce a mutation, right click the graph (control-click on Macintosh).

This opens a menu giving you the option of removing a strategy (setting its proportion to zero at the generation where you clicked) or introducing a strategy that is absent in the generation where you clicked. The newly introduced strategy is set to 1% of the population. The mutation is marked in yellow on the graph as shown in Fig. 2.

*Proportions.* By default, all strategies start out in equal proportions. Because some games depend strongly on the initial conditions, you may find it instructive to vary these proportions. These numbers are relative. Thus three strategies with initial proportions of 200, 100, and 100 are 50%, 25%, and 25%, respectively. To test whether a novel strategy can invade, set its proportion to zero, allow the remaining strategies come to equilibrium, and introduce the novel strategy as a mutation.

*Variables.* Set the value of each variable here. To change values gradually without typing numbers, use the up/down arrow control beside each variable. The graph is updated as variables change.

*Relatedness.* To test the effect of genetic relatedness, check this box and specify the relatedness of the players (0 to 1). If the box is not checked, the value is ignored. (Technical Note: When relatedness is used, payoffs are calculated by neighbor-modulated fitness in symmetric games and by Hamiltonian inclusive fitness in asymmetric games.)



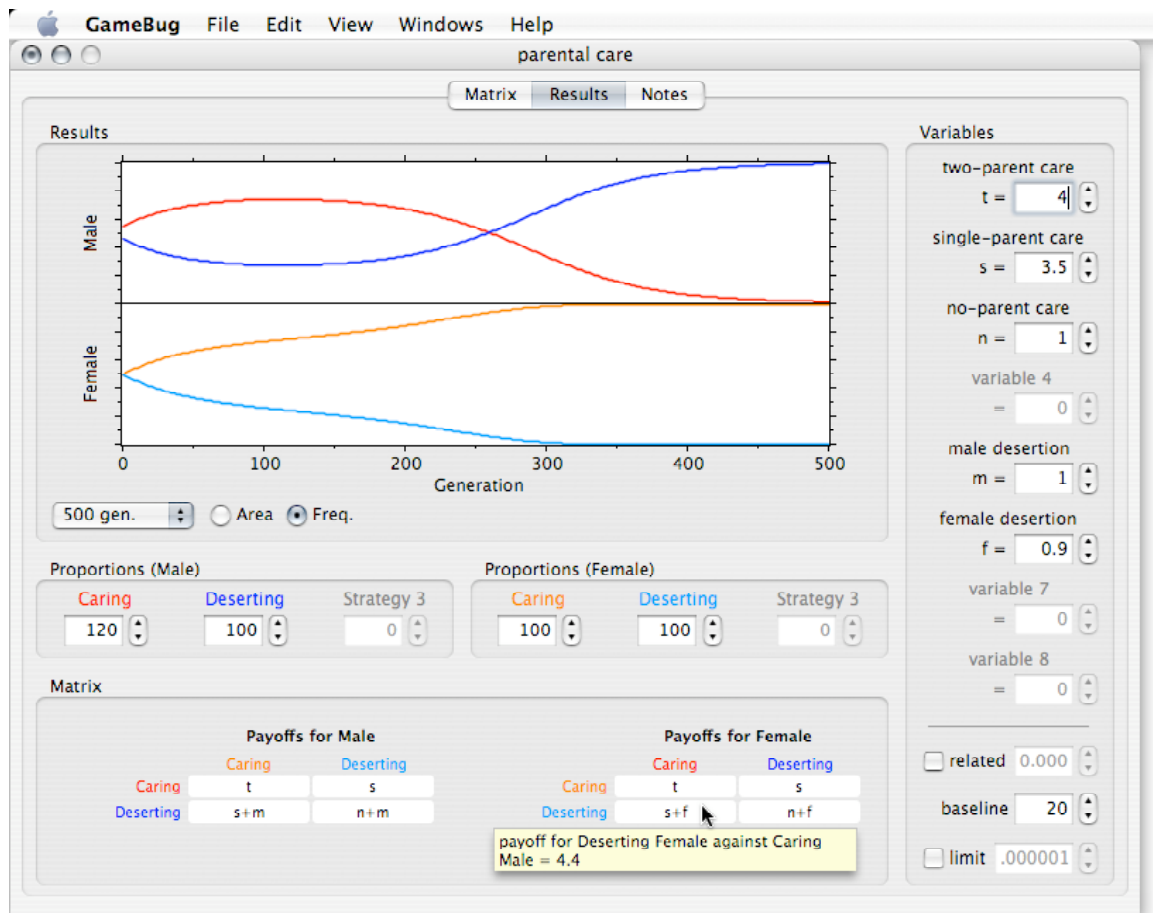
**Figure 2.** Results of a symmetric game with a mutation at generation 800.

*Baseline.* This is the fitness that each player starts with in each generation. It is increased or decreased by the payoff received when the game is played. Increasing baseline fitness slows change over generations but does not alter the outcome.

*Limit.* GameBug normally treats a population as infinite. Thus a strategy may fall to a frequency that is too small to display on the graph while still being present in the population (see rock.gbg for an example). To change this behavior, check the *limit* box; if it is checked, any strategy that falls below the specified proportion is set to zero. This mimics a limited population (e.g. a limit of 0.001 implies a population of 1000).

*Matrix.* The payoff matrix is shown for reference, but must be edited in the *Matrix* tab. To see the numerical value of a payoff given the current variable values, hold the mouse over the matrix cell in question as in Fig. 3.

To see the exact proportion of each strategy at a generation, hold the mouse over the graph. The generation number and list of proportions will appear below the graph. If the yellow mutation mark is present, hold the mouse over it for an explanation of the change that it represents.



**Figure 3.** Asymmetric game with proportions displayed as frequency rather than area.

### Printing and Saving Data (File Menu)

When you have created a new game or found a combination of variables that gives an interesting result, you can save or print your data. The *Print* command creates a single page showing the result graph, matrix, current variable values, and initial proportions.

If you modify an existing GameBug document by changing its matrix, variables, or initial proportions, you can save those changes back into the original file with the *Save* command. To save your changes into a new file, use *Save As* or *Save a Copy*. If you don't like the changes you made, the *Revert* command restores the most recently saved version of the data. The *Export Results* command creates a text file with the frequencies of each strategy for all generations displayed on the graph; the *Export Plot* command creates an image file with the graph in it.

### Example Files

We have included a few sample files with GameBug. Use these as a starting point and then explore on your own, adding strategies to them or building new games from scratch.

- |                |   |
|----------------|---|
| hawk.gb        | classic hawk-dove game where ess depends on values of $v$ and $c$ .   |
| bourgeoisie.gb | hawk-dove game with a new strategy for territory holders.   |
| rock.gb        | rock-scissors-paper, a three-strategy game with no stable outcome.  |
| parental.gb    | the asymmetric game of parental care, illustrating conditions that give rise to monogamy, polygyny, polyandry, or polygynandry. |

**Feedback**

This is the first version of GameBug to be tested outside Cornell University. While we have tested it extensively in our classes, it is likely that some bugs remain undiscovered. Please send bug reports to [rw12@cornell.edu](mailto:rw12@cornell.edu), including the following information: computer system (e.g. “OSX 10.3.5” or “Windows XP”), the exact words of any error message you saw, and what you were doing with GameBug at the time (e.g. “GameBug stopped responding after I set baseline to 0 in the Hawk-Dove game”). Be as specific as possible. If the problem occurs only in a specific game, save that game and send it to us. We are also interested in any suggestions or feature requests that you may have.

GameBug version 1.0, © 2005 Robert Wytttenbach (08/25/2005)