

Game Environment for Command and Control Operations (GECCO) User's Manual

**Joel Brynielsson, Henrik Bäärnhielm, Andreas Enblom,
Jing Fu Zi, Niklas Hallenfur, Karl Hasselström,
Henrik Hägerström, Oskar Linde, Klas Wallenius
and Jon Åslund**

Department of Numerical Analysis and Computer Science
Royal Institute of Technology
SE-100 44 Stockholm
Sweden
gecco@nada.kth.se

May 14, 2001

Contents

1	Introduction	2
1.1	What is GECCO?	2
1.2	Games	2
1.3	Requirements	2
2	Starting the game	3
2.1	Installation	3
2.2	Starting the server	3
2.3	Starting the client	4
3	Playing the game	4
3.1	Connecting to a game server	4
3.2	The game board	5
3.3	Units	5
3.4	Controlling a unit	7
3.5	Ghost units	7

1 Introduction

1.1 What is GECCO?

GECCO is a platform for simulating strategic games. It consists of two parts; the *server* and the *client*. The server is a program running on some computer that simulates the entire game. The client programs are used to control the actions of the game. Every player starts a client on his/her computer, and the program contacts the server program. The *game board* (see section 3.2) is then presented to the player. The board contains a *map* and several *units* (see figure 1). The player may see the properties of the units, and control some of them.

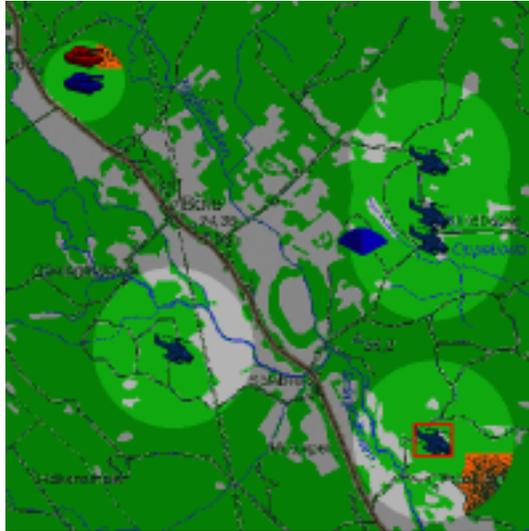


Figure 1: A typical game board.

A typical game is a wargame. The players control some number of military units that engage in combat against each other.

For a more thorough and general discussion of games and their properties, see the related document *GECCO General Description*[1].

1.2 Games

As stated above, GECCO is a platform for simulating games. It is not itself a game. To be able to play a game, somebody has to implement a game for the platform, and deliver it together with the rest of the platform. It will be assumed that you have access to a game implementation.

1.3 Requirements

The game is written in Java. It requires Java 2 runtime version 1.3 to run and should be runnable on any system that supports Java. Since the game involves communication between computers, the game also requires a TCP/IP network (it is usually enough to be connected to the internet or some intranet).

2 Starting the game

To start the game, three things are needed; the general server package, the client package, and a game implementation package. Furthermore, the game constructor usually supplies startup scripts that makes it easier to start the game.

The general server package contains the server components that are in common to all games. It consists of a single Java Archive (jar) file, named `server.jar`.

The client package The client package contains the complete client program. Again, it consists of a single Java Archive file, named `client.jar`.

The game implementation package contains all the data specific to the game. It usually has a number of image (`.gif`) files, a number of configuration (`.conf`) files and a Java Archive, named something like `game.jar` (where “game” is the name of the game). The game implementation should also include documentation about the game, containing information about how the game is started (using the startup scripts described below).

The startup scripts are the only things of the system that are platform-dependent, and they are used to start the game. For info about how to use these, see the documentation of the game implementation.

2.1 Installation

The installation is really simple: put all the files of the server, client and game implementation packages along with the startup scripts in some folder.

2.2 Starting the server

To start the game, use the information in the documentation of the game implementation, and the supplied startup scripts for your platform. Make sure you start the program from the folder where the game has been installed.

If there are no startup scripts the server has to be started manually, using the Java interpreter program and the Java Archive files. See the Java documentation for information about how to load jar-files and starting Java programs. The Java Archive (jar) files that should be loaded are the game implementation Java Archive and `server.jar`. It is important that the game implementation archive is loaded first and that the folder where the game is installed is included in the classpath. The name of the class to be executed is `server.startup.StartServer`. On a UNIX system a manual start of the server would look something like:

```
> java -classpath game.jar:server.jar:. server.startup.StartServer
```

When the server is started, several messages are printed to the screen, and when the server is ready to receive connections from the client programs, a message indicating that is displayed.

The network name of the machine where the server runs is necessary to know to be able to connect to the game (see section 3.1). The server sets up its communication on some network *port*, and it is also necessary to know that port number. The information about port number should be stated in the game documentation, and if it is not, the port is probably 7000. The port number is also among the information displayed when the server starts up.

2.3 Starting the client

The client is started pretty much like the server, using the documentation of the game implementation and the supplied startup scripts. Again, the game should be started from the folder where the game has been installed.

If no startup scripts are supplied, a manual Java interpreter startup is necessary (see section 2.2). The necessary Java Archive is `client.jar` and the name of the class to be executed is `client.Game`. On a UNIX system this would look like:

```
> java -classpath client.jar client.Game
```

When the client has been started, you are ready to connect to a game and start playing!

3 Playing the game

When playing a game, you take control over some units, moving and acting on a map. Which units you control depends on what role you play. The role also determines what part of the map that you can see, and which units you can get information about.

In a typical wargame a role is something like “Blue team player 1”, and the role controls some number of units, and can only see what is going on in areas close to those units.

3.1 Connecting to a game server

When the game is first started, the game connection dialog pops up (see figure 2). To bring up this dialog at another time, choose “Connect to game...” from the Game menu. In the dialog, type in the name of the host machine (the machine where the server runs) and the port at which the server has set up its communication (see section 2.2, or try using port 7000).

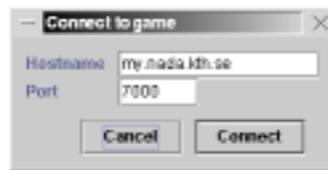


Figure 2: The game connection dialog.

When the connection has been established (the opposite is stated by a message on the screen), the role selection dialogs pops up (see figure 3), displaying

the available roles. Choose a role from the list by clicking on it, and then clicking the “Select” button.

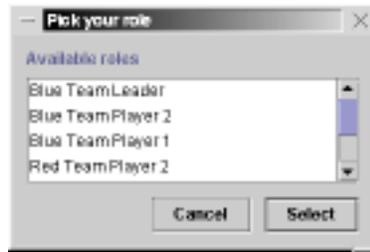


Figure 3: The role selection dialog.

Be aware of that it may take some time from the moment when the connection to the server is established to the moment when the role selection dialog shows up.

When the role has been selected, the map of the game is loaded, and you are ready to start playing!

To disconnect from the current game server, choose “Disconnect” from the Game menu. If you connect to a new game before having disconnected from the current game, the connection to the current game will be disconnected.

3.2 The game board

The game board (see figure 4), displays the map and the units of the game. Typically, large areas of the map will be darker than the rest. This indicates that you currently can not see what is there, because you have no units close enough.

If the map is too big to fit the board, the scrollbars can be used to see different areas of the map.

You can use the zoom buttons to zoom in or out on the map. If a unit is selected (see section 3.3), the map will be zoomed in or out around it.

3.3 Units

To select a unit, click on it. A red rectangle around the unit will indicate that it is selected. The unit info panel on the right will now contain information about the unit (see figure 5). The information is:

- Name (displayed on top of the panel)
- Type (displayed under the name)
- Properties (like fuel or health)
- Actions
- Messages

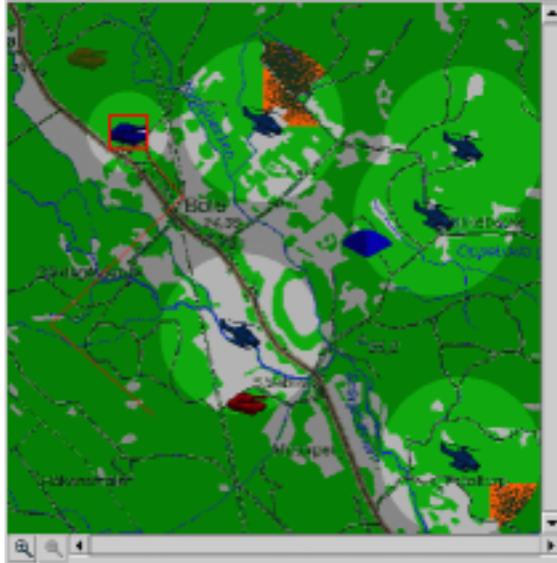


Figure 4: An example of a game board.

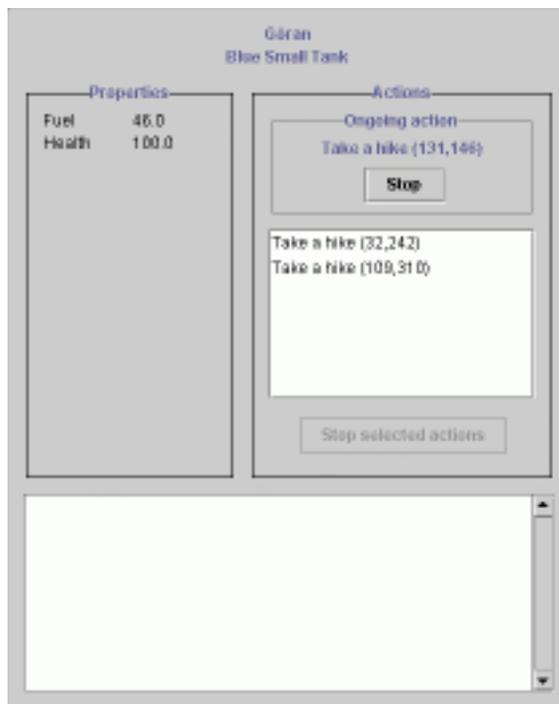


Figure 5: The unit info panel.

It is, however, not necessary that there is information available on the unit, since it may be a unit controlled by somebody else.

The actions are things that the unit is about to do. The ongoing action is what the unit is currently doing, and the actions in the list are what the unit will do later, in order. Red lines are drawn between the points where the actions will take place, starting at the unit itself (see figure 4). To stop some action from being executed, select the action to stop, and click the “Stop selected actions” button. To stop the ongoing action, click the stop button below it. At some point of time, some action may fail to be executed correctly (like, for example, when a tank is trying to move across water), and then every action in the list of the unit will be stopped automatically. In section 3.4, it is described how to assign actions for a unit to do.

3.4 Controlling a unit

To control a unit means that you assign actions for it to do. The actions will be executed in order, and be displayed in the info panel as described above. Some actions however, are instantaneous, and will be executed the moment you select them. They will not appear in the list.

To assign an action to a unit, first select the unit, and then press the right mouse button somewhere on the map. A list of available actions will appear. When an action has been selected, it will sometimes require an argument (i.e. against what to perform the action), and in such cases the mouse cursor will change to a crossbar. It should be clear from the name of the action if the argument should be a point on the map, or another unit.

3.5 Ghost units

When a unit disappears from your field of vision, it is replaced by a semi-transparent *ghost unit* (see figure 6). The ghost units are used to mark places where a unit was last seen, or places where a unit is believed to be.



Figure 6: A ghost unit.

You can put ghost units of your own on the map by right-clicking where you want it (when no unit is selected, of course, see sections 3.3 and 3.4), and selecting the type of ghost unit you want.

You can drag the ghost pieces with the mouse, to place them where you think there are other units. To remove a ghost unit, select it and choose action “Remove ghost” (see section 3.4).

The ghost unit feature can be turned off by un-checking “Use ghost units” in the Game menu.

References

- [1] Joel Brynielsson, Henrik Bäärnhielm, Andreas Enblom, Jing Fu Zi, Niklas Hallenfur, Karl Hasselström, Henrik Hägerström, Oskar Linde, Klas Wallenius, and Jon Åslund. *GECCO General Description*. Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, May 2001.