technische universität
dortmund

Supporting the Player by Means of Intelligent Group Behavior in Realtime Strategy Games

Nicola Beume, Boris Naujoks, Georg Neugebauer, Nico Piatkowski, Mike Preuss

Computational Intelligence, Chair of Algorithm Engineering
Department of Computer Science, TU Dortmund

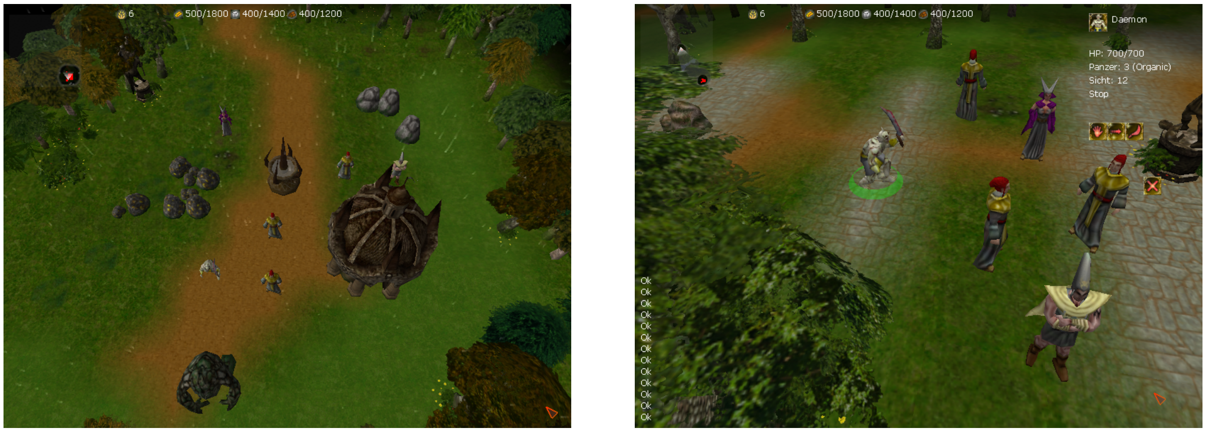
July 4, 2008

1 Introduction

As any player of RTS (realtime strategy) games knows, the management as well as the in-game behavior of units in most of these games is improvable in terms of movement, offense and defense. We modified the open-source RTS game Glest (www.glest.org) in order to estimate the potential of CI methods helping us to get out of this unsatisfactory situation. In this demo, we introduce our modifications on selecting, moving and attacking with groups of battle units.

2 Glest

Glest is settled in a fantasy mediaeval world. The players can choose to control either a magic or a technical faction. The magic faction uses wizards as basic units, which can be trained to become powerful warlocks and may conjure dragons and daemons as fighting units. The tech faction uses laborers as basic units, who may be trained to become knights, archers or engineers which are able construct (flying) battle machines.



Screenshot of Glest. The left picture shows a standard starting situation of the magic faction. The right picture shows an in-game situation from a fancy camera perspective.

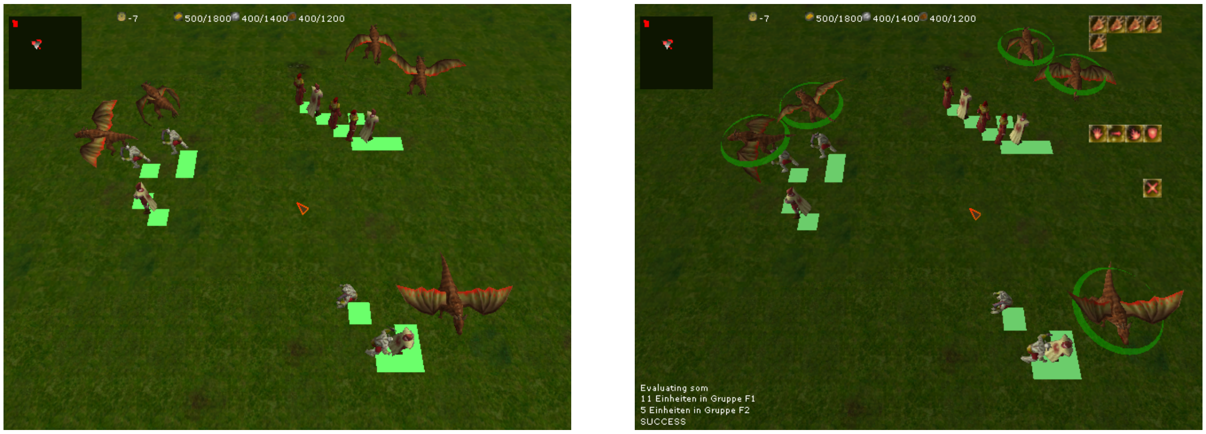
The game economy is based on four resources. Gold, stone, wood, and food are used by both factions and can be harvested by workers, technicians (tech faction) and initiates (magic faction). The player is able to select single units or groups of units an give commands to them.

3 Improvements of group behavior

We will present the following improvements in our demonstration.

3.1 Easy group selection by Self-Organizing Maps

Controlling and grouping units in a strategy game is quite difficult, especially when the number of units increases. We address this problem by an attempt to automatize the grouping process. The player can concentrate on strategy issues instead of managing his units, which is also time-consuming. For developing such an autonomous process we choose an unsupervised learning approach, the Self-Organizing Maps (SOM). Using SOMs works fine in clustering processes, which is a key part in building good groups of units in strategy games.



```
Evaluating som
11 Einheiten in Gruppe F1
5 Einheiten in Gruppe F2
SUCCESS
```

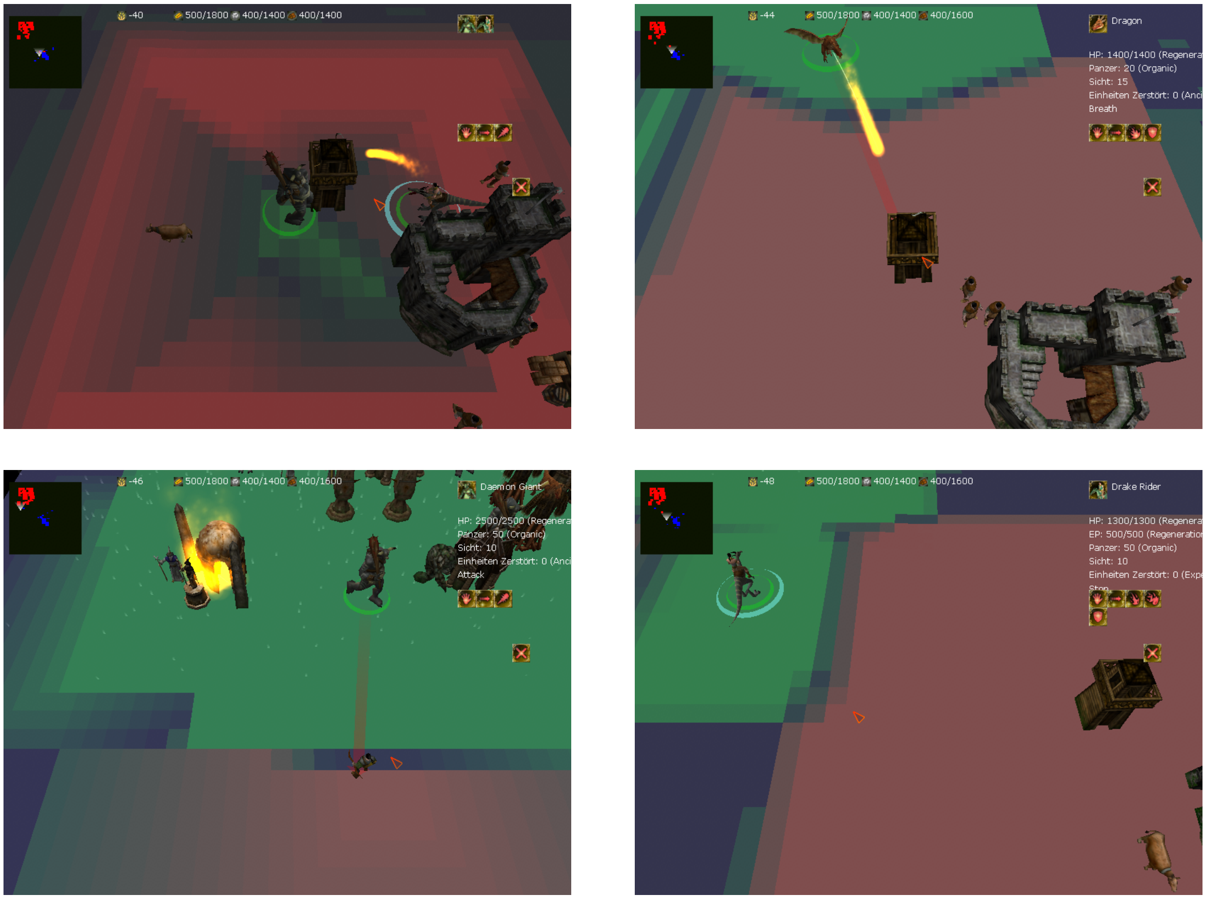
Screenshot of the SOM based selection. The left picture shows a group of friendly units, loosely gathered. The right picture shows a selected group of dragons (encircled) calculated by the SOM.

3.2 Smoother moving by Flocking

As mentioned, the player has the possibility to select groups of units and give every unit of this selection the same target position that it should move to. In this case, the player wants the group members to move cohesive rather than independently. This is especially important when the group is about to attack an opponent and single units are likely to loose the battle. Obviously, they should achieve their target as a group and the behaviour of the selection should be realistic, resembling the birds in a flock.

3.3 Smarter moving by considering Influence Maps

An Influence Map holds the quantified influence value for each faction on each tile at any time of the game. We applied these influence values to the weights, used in the A*-Pathfinding Algorithm to calculate "safe" paths, which means paths crossing tiles influenced by the own faction or with no or only weak influence of the adversary.



Activated IM visualization in four situations. The calculated influence values are transformed to colour codes and used to colour the ingame map cells. One can easily distinguish his own influence domain (green), the enemies influence domain (red) and the influence free areas (grey).