Optimizing Micro-Management in RTS Games

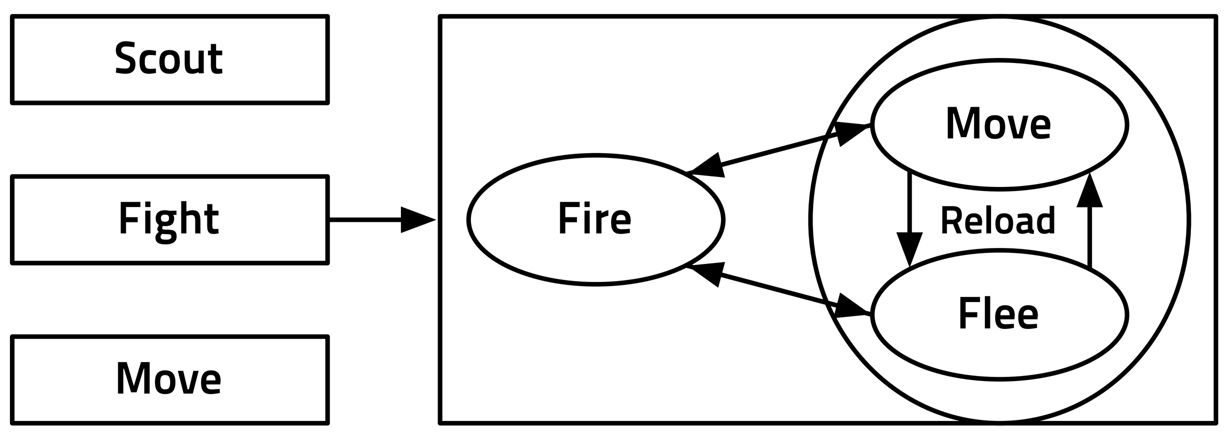
Gabriel Synnaeve and Pierre Bessière (University of Grenoble, INRIA, CNRS)

Introduction and Motivation

Micro-management is a central part of fast-paced RTS games in which it can change the course of whole matches. It is a task requiring coordination of multiple units as well as reasoning with incomplete information: we do not know what the opponent moves will be. We propose a reactive Bayesian model allowing decentralized moves selection and formations traversal (without the feeling that blocking units are being pushed away as they move themselves), damages dodging in fights, best fleeing local routes and it deals with collisions. The modeling approach that we present here can also be applied to FPS games (particularly involving squads).

Unit HFSM

Each unit is controlled by a hierarchical/model FSM. When in scouting mode, it flees all possible damages while steering towards its scouting goal. When in a move mode (no ennemy around), it either move simply or flock if there are many allied units. When in fight mode, it either fires, or flee or move towards a better position when reloading. The target selection algorithm is the only centralized (decided with help of the units group level) algorithm here.



Bayesian Model

We use Bayesian programming, a formalism able to describe any kind of Bayesian model to build our sensory-motor (sensors fusion) model:

Variables

- $Dir_{i\in[\![0..n]\!]} \in$ {True, False}: one variable for each atomic direction the unit can go to. $P(Dir_i) = 1$ means the unit will certainly go in direction i.
- $Obj_{i\in[\![0..n]\!]} \in$ {True, False}: direction of the objective, being it a scouting one, a priority target to move towards or the formation destination.
- $Dmg_{i\in[\![0..n]\!]} \in$ [DmgValues]: a subjective (DmgValues varies with unit types) potential fields to repulse the units from high (potential) damages directions.
- $A_{i\in[\![0..n]\!]} \in$ {None, Small, Big}: occupation of the direction i by an (interpolation of an) allied unit.
- $E_{i\in[\![0..n]\!]} \in$ {None, Small, Big}: same with enemy units.
- $Occ_{i\in[\![0..n]\!]} \in$ {None, Building, Terrain}: static occupation

Decomposition

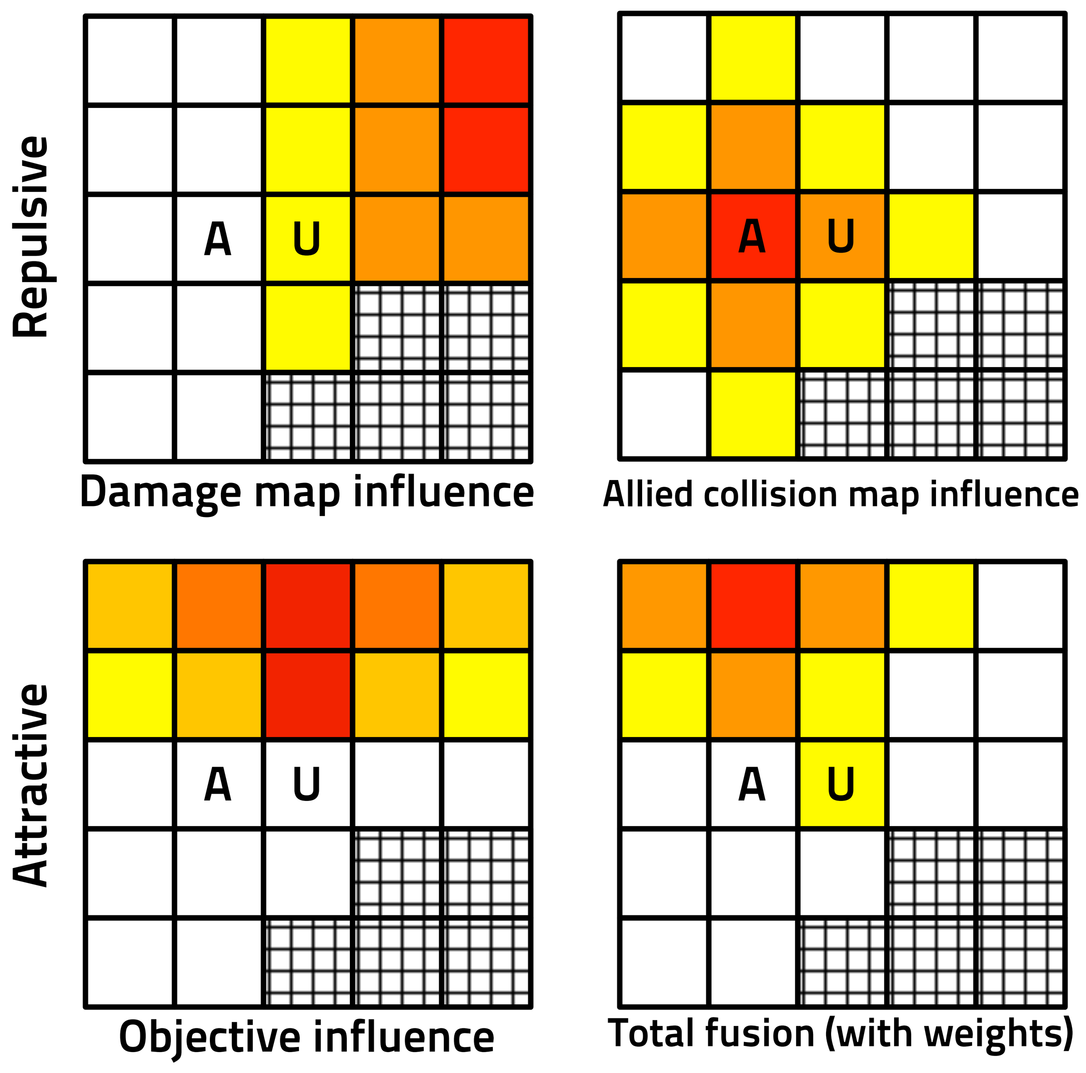
$$P(Dir_{1:n}, Obj_{1:n}, Dmg_{1:n}, A_{1:n}, E_{1:n}, Occ_{1:n})$$
$$= \prod_{i=1}^{n} P(Dir_i).P(Obj_i|Dir_i).P(Occ_i|Dir_i)$$
$$.P(A_i|Dir_i).P(E_i|Dir_i).P(Dmg_i|Dir_i)$$

$P(Dmg_i | Dir_i)$ can be learned or hand-specified

Question (fight mode)

$$P(Dir_{1:n}|Obj_{1:n}, Dmg_{1:n}, A_{1:n}, E_{1:n}, Occ_{1:n}, Prio_{1:n})$$

Example



Different sensory influences separated (repulsive on the left and attractive on the right), which combine to form the unit behavior. The group behavior emerges and is piloted by higher level sensory inputs (objectives).

Results

Real battle prob. table of a dragoon in a fight setup. White directions are of the highest probabilities and deep blue are the lowest. Effects of above variables.

Future Work and Conclusion

Future work on this model would be to have a generic and efficient mechanism to learn the conditional probability tables alltogether. through reinforcement learning and/or genetic optimization.

Our model is used in our BroodwarBotQ Starcraft's bot and won 2 of the 3 micro-managements setups against current AIIDE 2010 micro-management champion with hand-specified P(Dmg|Dir) and P(A,E|Dir) probability tables (worse than learned ones). This model allows for any kind of sensory input to have a weighted effect over the unit behavior in a unified, distributed framework.