Hunting Game - A parallel implementation in C#

In this project we propose a C# implementation of the wolf and sheep game described in class. We will create an array representing the board, and associate two mutexes with every cell: one for the wolves and the other for the sheep. Thus, any wolf can never be on the same cell as another at the same time, but it can enter cells in which there is a sheep. Each wolf has its own thread, as well as every sheep.

Wolves have two states: Search and Hunt. Sheep also have two states: Graze and Escape. Each state has a different movement algorithm.

We will combine the Strategy and Factory design patterns for the all movement algorithms for both the sheep and the wolves. This will allows us to implement and experiment multiple scenarios easily. At the same time, with minimum work, real wolves and sheep will be able to implement algorithms and research the best way to hunt/evade in a pack/herd.

The wolves will have a pack behavior implemented – they will cooperate to hunt a sheep in order to minimize the time spent searching. At the simplest level, cooperation will consist of a “shared view” of the map between wolves. This will be implemented using a shared, non-locking array.

For sheep, similar considerations exist. An interesting idea to be explored is whether one can create an algorithm that will maximize the life of a herd, as opposed to each sheep simply running for its life.