Kevin Cooper (cooperka)
Final Assignment Report

Introduction

I decided to make this model because of my enjoyment of computer games when I was in middle school. My favorite game during that time was called Age of Mythology, in which you chose a particular Greek civilization and built it up to form a great army, then used the army to fight with other humans online. This model is a much-simplified version of that game.

The results I plan to get out of this model involve which strategies are optimal for winning a game such as Age of Mythology. The user of the model can choose a particular build strategy (e.g. building more workers or building more soldiers) as well as a fighting strategy (e.g. roam randomly, group together in a mass, or offensively attack the enemy). These results are relevant enough that they can likely be applied to the real game, in order to have better luck when playing against actual humans.

Methods (ODD)

Overview

Purpose

The purpose of the model is to examine several different strategies for civilization-building and conquest within the context of role-playing computer games (such as Age of Empires, Age of Mythology, and Starcraft). Artificially intelligent agents follow the given strategy and battle each other until only one civilization remains.

State variables and scales

The world is a 101-by-101 grid (over 10 000 total patches) that wraps on all sides. Each patch may either be empty (colored black) or contain a resource. Possible resources are food, wood, and stone (colored yellow, green, and grey respectively). Occupying this world are turtles—which may be civilizations, workers, or soldiers. Workers and soldiers are collectively called "units."

Civilizations are shaped like a star, and serve as the "home base" for all of the workers and soldiers of the same color. These turtles spawn from the base, and workers must return resources to the base when they have been collected.

Civilization variables include the amount of resources they currently have, and the known locations of enemy civilizations. It costs resources to build more units, depending on the type of unit.

Workers, shaped like people, run back and forth from their civilization to nearby resources. If they see a resource in their line of sight (set by the user), they will pick it up and take it to their base. They only have 1 health (health is explained later).

Worker variables include information about their home (so they can return when they pick up a resource) as well as what resource, if any, they are carrying.

Soldiers look like rings with circles around them, and they roam the map looking for enemies. They begin with 3 health, represented by the circles around their ring; if they lose health the number of circles decreases.

Solider variables include information about their home and how much of their health remains.

Units also have a "line of sight," specified by the user, which controls how far they can see. Workers look for resources within this radius, and soldiers look for enemies. Both types also look for the location of enemy bases, in order to concentrate their offensive efforts (depending on the strategy chosen).

Process Overview & Scheduling

During runtime, the following processes occur on each tick.

- *check-loss*: if no workers and no soldiers exist for a particular color, that colored civilization has lost. The simulation runs until only 1 civilization remains.
- *move-workers*: worker action depends on if they are currently carrying a resource or not. If a worker is not carrying anything, they will continue walking forward until they see a resource, in which case they will head towards it. If a worker is carrying something, they will return to their home base and drop off the resource, increasing the total resource count for that civilization.
- *move-soldiers*: soldiers will fight if possible, otherwise they will continue moving. If an enemy exists within the 8 neighboring patches of the soldier, the soldier will do 1 point of damage to them. If the number of health points remaining for the injured turtle is 0, the turtle dies. If no enemies are in the 8 neighboring patches, the soldier will head toward the nearest enemy in sight.
- *spy*: units in range of an enemy civilization will "discover" it, and their home civilization will remember the coordinates for future attacks.
- *spawn-more*: if enough resources are available, civilizations will build more units. They preferentially choose a type of unit to build depending on the chosen build strategy.

Design Concepts

Fitness: A civilization's fitness is measured mainly by the number of resources it has, combined with the number of units it has. No single resource or single unit type is beneficial on its own; you must have soldiers to fight AND workers to forage in order to be successful.

Objectives: The objective of each civilization is to defeat all the other civilizations. To do this they need soldiers, and to build soldiers they need to gather resources, which can only be done with workers. Therefore, the civilization must prioritize the creation of these units in order to have a winning strategy. Workers may preferentially look for a specific type of resource depending on what their civilization needs, and soldiers may form a mass or target the enemy. This all depends on the strategy chosen by the user.

Sensing: Units can see around themselves in a radius known as their "line of sight." This is specified by the user of the model. Workers look for resources, while soldiers look for enemies. Both units look to discover the location of enemy bases.

Interaction: Soldiers injure other units, while workers do nothing but collect resources. Interaction may only occur through direct contact (within the 8 neighboring patches). Units also have a sort of "omniscient" knowledge of the status of their civilization (e.g. how many total workers exist in order to choose which resource to gather, or knowledge of where the enemy base is located).

Stochasticity: Initialization of the model is stochastic in that resources and civilization coordinates are randomly assigned. This, along with the strategy chosen, is what mainly defines who wins a given simulation.

Collectives: Soldiers may group together and form a "mass" if such a strategy is chosen. This mass is much more likely to be successful in battle, though it is also able to patrol less ground than it would if each individual soldier walked in a different direction.

Observation: Observation is done via several monitors and plots. These show the number of resources each civilization has, as well as the total population of each civilization and the number of resources remaining on the map.

Details

Initialization

During setup, resources are randomly distributed around the map and civilizations are given a number of starting units. Resources are grouped together based on type: each time, a random coordinate on the map is selected, and a particular resource is placed on the map in the area surrounding that coordinate. This is done repeatedly, the number of times being defined by the user of the model. A user-specified number of civilizations are then created, and each is given 3 initial workers in order to begin accumulating resources.

Global variables are also assigned during initialization, which represent the colors of the turtles and resources. Several reporter functions exist to access these globals and convert information from one form to another (e.g. turning the color yellow, represented by the number 44, into the word "food").

Submodels

<u>Check loss</u>

If no workers and no soldiers exist for a particular color, that colored civilization has lost. The simulation runs until only 1 civilization remains. When a civilization is removed from the map, the other civilizations must forget about it so their soldiers don't continue to group around a no-longer-existing base.

Move workers

Worker action depends on if they are currently carrying a resource or not.

- *Forage*: if a worker is not carrying anything, they will continue walking forward until they see a resource, in which case they will head towards it. If there is a build strategy set, they will preferentially choose resources that help them obtain that strategy (e.g. head towards food instead of wood if the strategy is "workers").
- *Return Home*: if a worker is carrying something, they will return to their home base and drop off the resource, increasing the total resource count for that civilization.

Move soldiers

Soldiers will fight if possible, otherwise they will continue moving.

- *Fight*: if an enemy exists within the 8 neighboring patches of the soldier, the soldier will do 1 point of damage to them. If the number of health points remaining for the injured turtle is 0, the turtle dies.
- *Move*: if no enemies are in the 8 neighboring patches and thus the soldier cannot fight, the soldier will head toward the nearest enemy in sight. If no such enemy exists, they will move depending on the strategy chosen: "mass" means they will all head toward the "leader" soldier (the one created earliest in the simulation); "attack" means they will head toward an enemy civilization, if any such civilization is known.

Spy

Units in range of an enemy civilization will "discover" it, and their home civilization will remember the coordinates for future attacks. These coordinates are used when soldiers are moving (as described above).

Spawn more

If enough resources are available, civilizations will build more units. They preferentially choose a type of unit to build depending on the chosen build strategy (preferring either workers or soldiers). If "balanced" is chosen, they attempt to keep even numbers of workers and soldiers.

Model Implementation

The model was implemented in NetLogo v. 5.0.4.

Kevin Cooper (cooperka)
Final Assignment Report

Analysis

Several BehaviorSpace experiments were run in attempt to determine the optimal strategy for winning. Each experiment measured whether or not you (the blue team) won, depending on what strategy you chose. Since there are 3 players, the expected value for the win percentage is 33.3% (i.e. completely random).

The first set of experiments was performed in order to determine which build strategy is best. The options are no strategy; preferring workers; preferring soldiers; and trying to keep the numbers balanced. 100 runs of each type were performed. It appears that preferring soldiers is optimal, with balanced close behind.

	Win Percent
None	35 %
Workers	31 %
Soldiers	41 %
Balanced	39 %

The second set of experiments was performed in order to determine which fighting strategy is best. The options are no strategy; grouping together; and offensively attacking. It appears that attacking is optimal. 100 runs of each type were performed. Again, note that 33% is the expected value. It appears that offensively attacking is by far the best strategy here.

	Win Percent
None	31 %
Mass	19 %
Attack	42 %

The final set of experiments was performed with the optimal strategy determined above—that is, preferring soldiers and offensively attacking. 200 identical runs were performed to ensure accuracy. It got the largest percent of wins of any strategy, a full 48% better than expected (49% versus 33%).

	Win Percent
Optimal	49 %

Conclusion

The purpose of this model was to examine different strategies that can be implemented by players of games such as Age of Mythology, and this was definitely accomplished. It appears that the best strategy is to favor the creation of a large army rather than build workers who can gather resources more quickly, and also to be on the offensive rather than defensive. This conclusion makes sense conceptually, and in fact is exactly the strategy I would use when playing the game myself back in middle school.