Creature:
An abstract class, initially was an interface with modularity in mind. Due to certain methods that could be defined at the parent class I found that it made more sense to set it as an abstract class.

Hobbit:
Outside of the requirements, hobbits will lose a hunger value each turn unless vegetation was harvested. Replication also relies on hunger as opposed to just solely on time. Hobbits can cross the magical barrier to leave Sauron's garden and be safe from his wrath.

Nazgul:
They're kind of stupid. They will attack hobbits to gain food and harvest vegetation to get rewarded by Sauron. Replication is also based on food along with time. When they hit the barrier, they get stuck and are kept alive, until Sauron's wrath.

Vegetation:
Sauron's magical garden tends to overgrow not only with ease but at an astonishing pace. Vegetation can only replicate and it does so fairly often. Vegetation can grow over the barrier.

Sauron:
Sauron sits in the middle of his garden as it is one of his favorite pastimes to witness his garden overgrow (aside from attempting to conquer Middle-earth). However, Sauron gets hungry pretty easily and he's not happy with the hobbits and nazguls reaping too much of his crop. This causes him to devour anything within the magic square.

Item:
An abstract class to to clear up extended class files and ability to polymorphically equip inherited classes.

Sword:
Some hobbits will be granted a sword at the beginning of their harvest boosting their attack by 50 more points.

Armor:
Some hobbits will be granted armor at the beginning of their harvest boosting their health by 50 points.

Food:

Dropped by the vegetation the hobbits harvest. Maybe it will be good for tomorrow's quiche?

Map:

A separate class file that contains the 2D creature array where all the creatures exist. A magical barrier sits 10 units within the map's outermost limits. Hobbits will attempt to get over this barrier as they sense that danger is imminent.