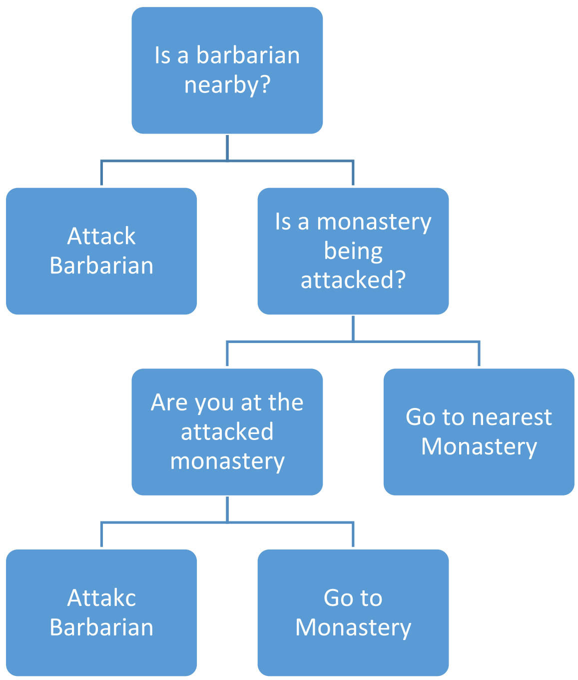
Steven Santorelli, Dajin Lee, Joshua Kwiatkowski

Monks Knights Barbarians:

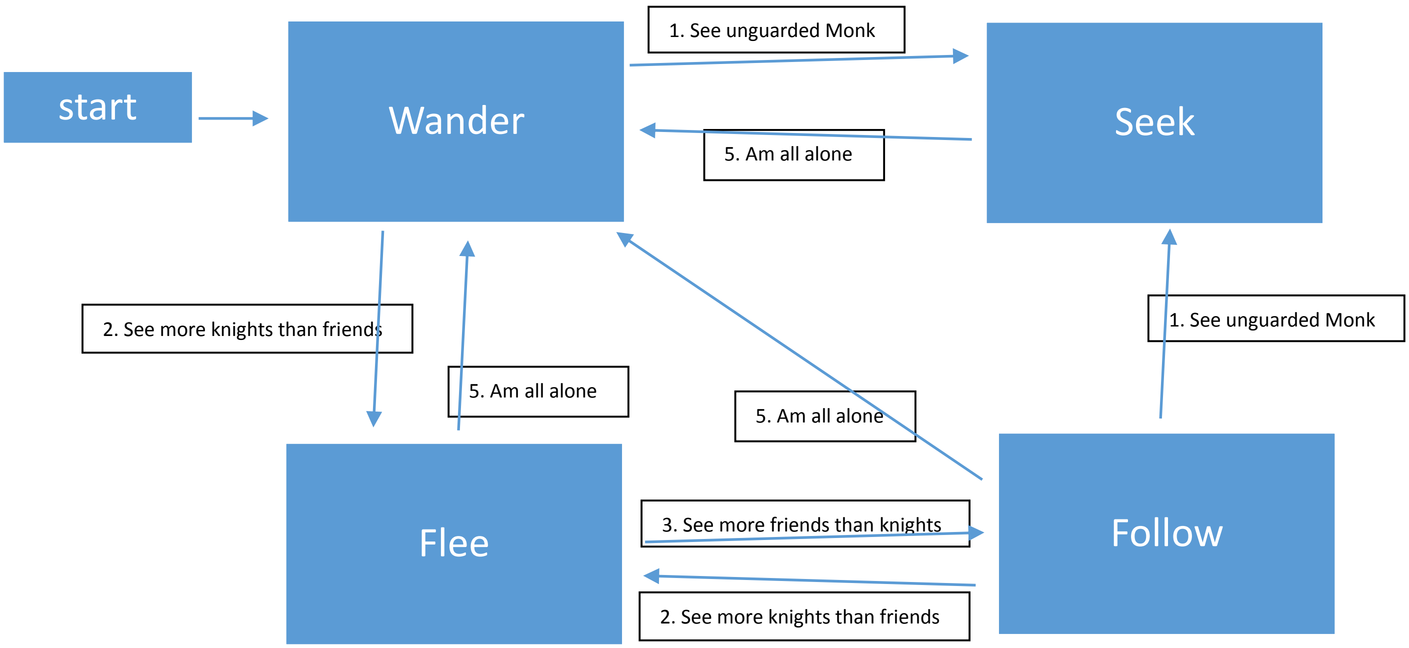
Path following:
Originally, the monks used path following. They moved along a set path going from on monastery to the next in a never ending loop. However, we took it out of the game because it was not helping the game in any way and replaced it with path finding.

Path finding:
The monks, knights, and the barbarians all use path finding. We use a navigation mesh for the movement, with the positions of the monasteries baked into the mesh.

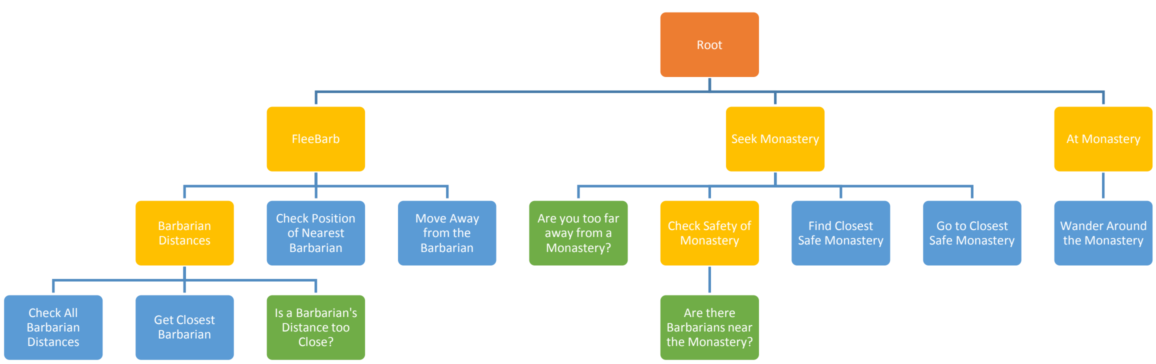
Decision Tree:
The knights use a decision tree to determine their behavior.



State Machine:
The barbarians use a state machine to determine their behavior



Behavior Tree:
We did not implement behavior trees into the game. We created a prolog behavior tree for the monks.



Genetic Algorithm:
All three use genetic algorithms to change their attributes every round. These attributes were all changed by a single fitness value that was evolved through a single threshold.

The traits evolved by each unit is as followed:
Monks – Sight range and move speed
Knights – Sight range, speed, and attack delay
Barbarians – Hit chance, sight range, speed, and attack delay

We used a population of 30 units, 10 of each class. Each chromosome had 10 bits and the phenotype range was 0 to 200. We calculated fitness based on the time that the unit survived and, for the knights and barbarians, how many units they killed. For the monks, since they could not kill, the time survived counted more heavily towards their fitness value. As a result, fitness values ended up in the range of about 10 to 180.

Only one threshold was evolved during all testing. During this testing, it took about 12 rounds for preferable behavior to evolve, which was noted by the increased variance in the win condition that was occurring each round. The values of the traits that we ended up with varied about 30% either higher or lower than our initial hard coded values.

I believe this feature is viable in the game as it adds a nice variance to each team's performance. Given our old scenario, it was extremely common for the Knights to win every round that was played, but when varied attributes were introduced, their win rate began to drop if they received lower fitness values. As a result, it can be said that genetic algorithms enhanced our game by making the teams more balanced. Balance is also an important factor in making a game more fun, as it reduces predictability and provides more unique occurrences as it plays out. A scenario involving this would likely revolve the changes between each situation a character is in, where sometimes your fitness is allowing you full control of the situation, and other times it makes you nearly helpless.

Bayes Classifier:
Bayes looks at the number of monks, knights, and barbarians within sight distance of the barbarian as well as whether or not there is a monastery within sight distance. The numbers on this are continuous, while the monastery check is a boolean. The action is a decision on whether or not the barbarian is going to be aggressive or not, a boolean.

Observations and decisions were made once every 20 seconds as well as upon death. If a barbarian killed something and didn't die that was considered a "good" observation, while a death and no kills resulted in a "bad" observation. There were other ways that could have eventually been added in, but these two results ended up giving a fair amount of responses. New decisions were made every 20 seconds, while the aggressive vs passive decision was kept for the entirety of the 20 second period. Upon death or the end of the round the barbarian would check to see whether or not it should store its decision in the history. While our algorithm did learn from its mistakes, mistakes weren't always mistakes as sometimes bad luck just happened (misses or lucky hits from knights), so bad data occasionally crept in and bad decisions were made or recorded.

Excerpt from the observation table:
Knights / Monks / Barbarians / Monastery / Aggressive (decision)
1 0 0 False False
0 0 0 False False
0 0 0 False False
0 0 0 False False
0 1 0 True False

0 0 0 False False
0 0 0 False False
0 0 0 False True
1 0 0 False False
0 0 0 False True
1 0 0 True False
0 0 0 False False
1 0 0 False False
0 0 0 False True
0 0 0 False False
0 0 0 False False
0 0 0 False False
2 0 0 False True
0 0 0 False False
1 0 0 False False

The end result ended up being that the barbarians started off being really really REALLY passive. They would run away from everything, including solo buildings, and the learning that we had off of that didn't have the right easy way to decide it was making a bad decision, as running away from a solo building didn't constitute a right or wrong decision, it was just a decision that was not recorded. As a result, only after the barbarians ended up dying a bunch while being passive did they finally realize that they could be aggressive. Once this happened it became much more interesting to see the outcome of a round. In the end the barbarians still make some weird passive decisions, but over time is definitely is much better.