

Hugh McGinley
Software
Hw3 Design Rationale

Given the length of this assignment, there weren't too many design decisions that I had to make, but I did structure some of my classes to make it as simple to understand as possible. For example, I had a Race class that was able to instantiate as many Horses as needed and race them simultaneously to determine a winner. It goes without saying that I also had a Horse class that stored information (max speed, name, race strategy, etc) about the horse, and also provided functions to change any of the values, if needed.

The biggest design decision I made was to structure the Strategies with an interface with a generic runRace() method. From there, I overrode the runRace method in three different classes that represented the three different race strategies. This made it easy for me in the Horse class to instantiate one RaceStrategy, but modify it within the setRaceStrategy() method at any time.