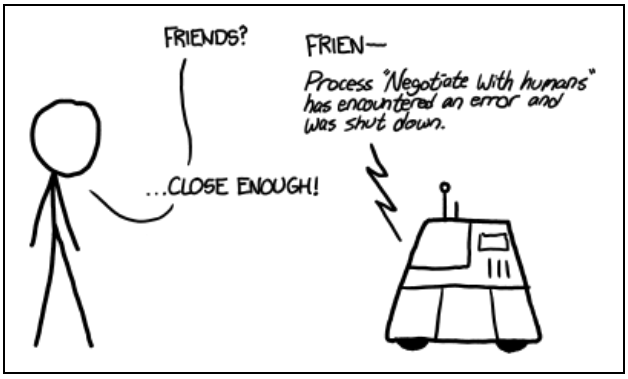
Attack Robots: Part 1



In this project, you will be creating 2 teams of 4 robots each that will attack one another. The team that does not lose all its robots (that is, the other team's robots lose all of their health) will win.

Setup Stage

The setup for this project is as follows:

- Each team will be an array of type "robot", each of which will hold 4 robot objects
- Each robot object will have a name, an amount of damage they inflict, and some amount of health
- The name and health will be scanned in by the user for each robot.
    - RESTRICTION: There is only 30 points of health that can be given out to each team. This can be spread among the robots in any way, but each team cannot have more than 30 health points in total. Your code must enforce this rule by disqualifying any team that has over 30 health points.
- The damage inflicted (the damage_given variable, if you have been following the format in class), will be a random number between 1 and 10. This can be set up either in the constructor, or in a mutator method.

Battle Stage

For the battle stage, there will be "simultaneous" attacks. Team 1 will choose which robot will attack, and which robot will be attacked. *Hint: Use the positions in the array to choose which robot will be attack / be attacked*.

Then, Team 2 will have the same choice. After both teams make their decisions, teams will attack "simultaneously". *By simultaneous, this means all repercussions are felt after both teams attack. For example, if a robot from Team 1 attacks and kills a robot from Team 2, that "dead" robot can still attack on this turn.*

After each attack, check to see which robots are alive or not. If any robots are dead, they cannot be used in another attack. If all robots on one team are dead, then the game is over, with the other team winning. If all robots on both teams are dead, then the game is over in a tie. If the game is not over, repeat the battle stage.

Hints:

Use vectors: https://www.codeguru.com/cpp/cpp/cpp_mfc/stl/article.php/c4027/C-Tutorial-A-Beginners-Guide-to-stdvector-Part-1.htm to store each team.

Get this code working with ONE robot on each team first. Then, build up to 4 robots on each team.

Robot methods can be used on any robot object. That is, if an array named "team1" is of type robot, then team1[0].getHealth() will get the health of the robot in position 0.