* Server
  + Summary of class:
    - The server is responsible for all the interactions that the players have with the system. The server regulates the flow of the program by determining who’s turn it is and what they can and cannot do. By utilizing the server the process of running and updating the game is streamlined and done solely in one class.
  + Fields:
    - - <List> turnQueue
      * The designated order of how each round will start and end.
    - - <array of Tiles> board
      * The game board broken into individual tiles. We chose to use a array because unlike a list the board size will not add or remove tiles as the game continues.
    - - Display display
      * ????
    - - Array playerStats
      * Holds statistics for each player, human or AI every movement and fire action will have their corresponding statistics updated.
    - - Array playerInfo
      * Holds the information regarding the player’s units and the player’s id number.
  + Methods:
    - - getNextPlayer()
      * Summary: Gets the next player in the turnQueue.
      * Pre conditions: The game has not been resolved.
      * Post conditions: Return the next player in the queue.
    - - removePlayerFromQueue( playerID ID)
      * Summary: Removes a player from turnQueue, this is called whenever a player is removed from the game.
      * Parameters:
        + ID: the player that will be removed from the turn queue.
      * Pre conditions: A player has lost all their robots.
      * Post conditions: The player has been removed from the turnQueue parameter.
    - - updateRobotPosition(Robot movedRobot , Tile newLocation)
      * Summary: If a robot has moved successfully it’s position must be updated.
      * Parameters:
        + movedRobot: The robot that has moved.
        + newLocation: The new tile location the robot has moved to.
      * Pre conditions: A robot has moved to a different tile.
      * Post conditions: The location of the robot and the tile it now occupies has been updated.
    - - moved(tile destination)
      * Summary: Handles a received move request from a player.
      * Parameters:
        + Destination: the tile that the player wants to move their tank to.
      * Pre conditions: Received a move request.
      * Post conditions: Approves or denies the request based on if it is a legal move and the player’s corresponding playerStats is updated & updateRobotPostion is invoked.
    - - fired(tile destination)
      * Summary: Handles a fire request from the current player. If the tile is within range and the player has not attacked this turn the destination tile is damaged.
      * Pre conditions: The player has not attacked this turn & the tile is within range.
      * Post conditions: All robots in the fired tile (if any) receive damage.
    - - updateFOW(playerID player) //do we need player
      * Summary: After a robot has moved or been destroyed the fog of war must be recalculated for the effected player.
      * Parameters:
        + Player: the player that will have their fog of war updated.
      * Pre conditions: A robot has moved or been destroyed.
      * Post conditions: The fog of war for that player has been updated accordingly.
    - - playerKilled(player)
      * Summary: Removes a player from the game.
      * Pre conditions: All of the player’s robots have been destroyed.
      * Post conditions: The player will no longer appear in the turn queue and their playerInfo parameter is updated.
    - - checkStalemate()
      * Summary: Checks whether a stalemate has been reached which is when there are only AI players and there have not been any shots fired or movement after x turns.
      * Pre conditions: Only AI players remain.
      * Post conditions: If the stalemate has been met terminate the game, else continue game.
    - + const.(<List> options)
      * Summary:
      * Pre conditions:
      * Post conditions: