Robot World Server Requirements

The world is a **grid**.

1. Distance in the world is measured in a unit called a `step`.
2. The width the world is the length of the top or bottom side measured in steps.
3. The height of the world is the length of the left or right side measured in steps.
4. The size of the grid is up to you. However, the size cannot be hard-coded. It must be configured using a configuration file which is read by the server program on start-up.
5. For example, the world size can be 100 steps wide and 50 steps wide.

World Coordinate System

1. The **centre** of the world is at coordinates [0,0].
2. The top left of the world is at [-width/2, height/2].
3. The bottom right is at [width/2, -height/2].
4. For example, if the world has width of 200 steps and a height of 50 steps, then the top left is [-100, 25] and the bottom right is [100,-25].

Directions

1. The world uses a standard compass with north, south, east and west.
2. North points to the top edge of the world.

Visibility

1. The world controls the range of **visibility** of all robots.
2. Visibility is configured in the server's configuration file as a value in *number of steps*.
3. A Robot cannot see further than the server's configured number of steps in each direction - ahead of it (i.e. in the direction it is facing), to either side and behind.
4. If an object is larger than the visible range, then the robot cannot see the extent or ends of the obstacle.
5. For example, if the world configures visibility of 10 steps, then a robot cannot see anything that is 11 steps or further on.

6. **Optional:** A robot can remember stationary objects it has seen already seen since joining the world.

Obstacles

1. The world has **obstacles**.
2. Obstacles are rectangular or square in shape.
3. Obstacles are positioned in the world by specifying the top-left coordinate and bottom-right co-ordinate.
4. You can construct more complex shapes by placing obstacles next to each other.
5. Obstacles *cannot* overlap each other.
6. Obstacles block the path of robots.
7. The world must have at least one obstacle.

Pits

1. The world has **bottomless pits**.
2. If a robot enters a bottomless pit, it dies.
3. Pits are rectangular or square in shape.
4. Obstacles are positioned in the world by specifying the top-left coordinate and bottom-right co-ordinate.
5. You can construct more complex shapes by placing pits next to each other.
6. Pits *can* overlap each other.
7. The world must have at least one bottomless pit.

World Commands

Managing the world is done via a **console interface** on the server program. After the program is started, the program must allow for world commands to be issued.

The following commands must be implemented:

1. `quit` - disconnects all robots and ends the world.
2. `robots` - lists all robots in the world including the robot's name and state (See Protocol for state information).
3. `purge <robot name>` - kill the robot and remove it from the game.
4. `dump` - displays a representation of the world's state showing robots, obstacles, pits and any-

thing else in the world that you programmed. The output format is your choice. It can either be a text list or graphic image that you construct. It can even be a JSON or YML pretty-printed string.

Other World Configuration Parameters.

In addition to the visibility, the server must also configure the following:

- How long it takes to repair shields (in seconds)
- How long it takes to reload weapons (in seconds)
- How long it takes to set a mine (in seconds)
- The maximum strength of a shield (in hits)

See client requirements and protocol for more information.