**Purpose:** To move the robot around on the field. The Navigation class receives navigation events from the task controller. The data received (contained inside an object) will include data like target coordinates/angles and max speed. Navigation will also have to deal with reflecting coordinates and angles depending on which alliance we are on, due to how the field is symmetrical over a center line instead of rotationally symmetrical (like in previous years).

|  |  |
| --- | --- |
| **Priority:** Medium-High | **Reason:** The Actions class will not be able to score much if the robot can’t move, and Navigation alone can get some points, but the EPS class needs to be created first. |

**Primary Programmers:**

1. Eamonn R
2. Bryan B

**Public Constants (public static final):**

* None

**Constructors (called when an object instance is created):**

* (1 argument)
* Initialize: The DriveSystem instance, which takes the hardwareMap argument
* Arguments:
  + HardwareMap hardwareMap
    - Passed to the drive system object when it is created so it can setup access to the drive motors

**Interface Instance Methods (used on an instance of this class):**

* Any abstract classes required by EPS

**Interface Static Methods (used without an instance of a class):**

* None

**Other Information:**

* Uses a grid system
  + Scale is inches (field minus the perimeter is a square, 11 feet 11 inches)
  + Origin is the center of the field
  + Orientation: The center line of the field (line of symmetry) has a slope of -1 and intersects the y-axis at the origin, the mountains are in the 1st and 3rd quadrants
  + Must be able to reflect points and angles depending on which alliance we are in for a match
    - Equations are as follows:
      * Reflected x coordinate = -yoriginal
      * Reflected y coordinate = -xoriginal
      * Reflected Angle = -Ɵoriginal – 270