**Purpose:** The event manager is the common class used by all autonomous opmodes to control the other parts of the program. The event manager instantiates (creates an instance of) each EPS (Navigation and Actions), and starts each running in its own thread. This class uses list(s) of events from a configuration file or object specified by the opmode that creates it. Each configuration can contain a different combination and order of Navigation and Actions events, allowing for multiple different autonomous opmodes to be created fairly easily.

|  |  |
| --- | --- |
| **Priority:** High | **Reason:** This class sets up and assists the other important autonomous classes. |

**Primary Programmers:**

1. Holly L
2. Bryan B

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

* + None

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

* (<1> arguments)
  + Must Initialize: The HLQ (“High Level Queue”), which is generated by HLQGenerator.makeHLQ(configFile)
  + Arguments:
    - String configName
      * Provide the name of the event configuration file to use (the file name minus the extension, relative to the root config folder) to the HLQGenerator (which returns the HLQ object)

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

* public void start(0 arguments)
  + Purpose: Starts the navigation and action systems (calls their start() methods)
  + Priority: Medium (needs to start the other systems like navigation)
  + Arguments:
    - N/A
  + Returns: N/A
* public loop(0 arguments)
  + Purpose: Complete one cycle of events for autonomous, should call run() for navigation and actions systems when they have not already called navigationDone() or actionsDone(), and when both are done queue the next blocks of events
  + Priority: Very High (autonomous robot is a paper weight until this is implemented)
  + Arguments:
    - N/A
  + Returns: N/A

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

* public static void navigationDone(0 arguments)
  + Purpose: Called by navigation to indicate that it is done with its current block of events and is ready for the next block. When navigationDone and actionsDone have both been called, this class should proceed to the next block of events.
  + Priority: High (needed to transition to the next block of events)
  + Arguments:
    - N/A
  + Returns: N/A
* public static void actionsDone(0 arguments)
  + Purpose: Called by actions to indicate that it is done with it’s current block of events and is ready for the next block.
  + Priority: High (needed to transition to the next block of events)
  + Arguments:
    - N/A
  + Returns: N/A