**Purpose:** Provide a layer over the servo classes provided by the FTC libraries. The intent and advantages of this class are similar if not identical to the Motor class (see its class definition for details).

|  |  |
| --- | --- |
| **Priority:** High | **Reason:** Servos will be used for many of the manipulators used for scoring. |

**Primary Programmers:**

* Adam

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

* int MAX\_POS = ???
  + Purpose: Provide the maximum position value a servo can be set to. (in degrees)
* int MIN\_POS = 0
  + Purpose: The lowest position value a servo can be set to. (Could be ≤ 0)

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

* public Servo(2 arguments)
  + Must Initialize: Whatever needs to be initialized in the FTC servo library for a servo to work, and store the information to access the servo, which could be a reference to the servo instance from the FTC libraries. Also should initialize the servo to the specified position.
  + Arguments:
    - ?String? servoName
      * The name the servo was registered with in the robot configuration
    - int startPos
      * The position the servo should be initialized to.

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

* public synchronized void setTarget(1 argument)
  + Purpose: To set the position the servo will move to. This should return immediately after setting the servo target; it should not wait for the servo to stop moving.
  + Priority: Very High (Needed for basic control of the servo)
  + Arguments:
    - int targetPos
      * The position to move the servo to
  + Returns: N/A
* public synchronized int getCurrentPos(0 arguments)
  + Purpose: Get what position the servo is at when this method is called.
  + Priority: Medium (not vital to using a servo, but could easily be used for tracking a servo during autonomous and possibly teleop)
  + Arguments: N/A
  + Returns: (int) the current position of the servo
* public synchronized int getTargetPos(0 arguments)
  + Purpose: Gets the last target position the servo was set to (see also: getCurrentPos)
  + Priority: Medium (likely to be used during autonomous or teleop, prevents classes from having to store the target position locally if needed later)
  + Arguments: N/A
  + Returns: (int) the current target position the servo is moving to

\*Type depends on final decision(s) made as a group.