Retrieving Information About the SpuCraft

Some Spumone challenge worlds are set up so that you can retrieve information about the dynamic state of the SpuCraft. In some worlds, you only have access to partial information. In other worlds you do not have access to any information.

Lengths are measured in meters, time in seconds, and angles in radians.

getX() [Returns double.] Retrieves the x (horizontal) coordinate of some point (usually center of mass) of the spuCraft. Units: meters (m)

getY() [Returns double.] Retrieves the y (vertical) coordinate of some point (usually center of mass) of the spuCraft. Units: meters (m)

getVelX() [Returns double.] Retrieves the horizontal component of velocity of some point (usually center of mass) of the spuCraft. Units: m/s

getVelY() [Returns double.] Retrieves the vertical component of velocity of some point (usually center of mass) of the spuCraft. Units: m/s

getAccelX() [Returns double.] Retrieves the horizontal component of acceleration of some point (usually center of mass) of the spuCraft. Units: m/s^2.

getAccelY() [Returns double.] Retrieves the vertical component of acceleration of some point (usually center of mass) of the spuCraft. Units: m/s^2.

getAngle() [Returns double.] Retrieves the orientation angle of the spuCraft or part of the spuCraft. Units: radians (rad).

getAngleDot() [Returns double.] Retrieves the first time derivative of orientation angle. Units: rad/s.

getAngleDDot() [Returns double.] Retrieves the second time derivative of orientation angle. Units: rad/s^2.

Retrieving Target Information

In several of the worlds, there are targets that you must somehow intercept. In others, you can set a target manually with a mouse. There is a set of functions which retrieve information about the target.

getXTg() [Returns double.] Retrieves the x (horizontal) coordinate of the target. Units: meters (m).

getYTg() [Returns double.] Retrieves the y (vertical) coordinate of the target. Units: meters (m).

getXClick() [Returns double.] Retrieves the x (horizontal) coordinate of the last place on the screen that the mouse was clicked. Units: meters (m).

getYClick() [Returns double.] Retrieves the y (vertical) coordinate of the last place on the screen that the mouse was clicked. Units: meters (m).

Sending Information to Spumone

You can send information back to Spumone in order to give the spuCraft its driving commands and to display information on the screen.

send(i,d) [takes int, double] This command is used to send double precision numbers through data channels to Spumone. Normally this is used to give the spuCraft commands. The number d is sent to channel i. Functionality will vary depending on the spuCraft and the event.

sendInt(i1, i2) [takes int, int] This command is used to send an integer i2 to Spumone through channel i1. Functionality will vary depending on the spuCraft and the event.

sendBool(i, b) [takes int, bool] This command is used to send a bool b to Spumone through channel i. will vary depending on the spuCraft and the event.

msg.set(i,s,d) [takes int, char*, double.] Used to send messages to be displayed on screen. The message index is i, s is a c-string character array that represents the label: e.g. “MyVar = “. D is the number to be displayed. Some events do not accept labels.

Interfacing with the Gamepad

In using Spumone, it is important that you be able to connect to the gamepad and read input signals from it. I have tested Spumone with the Logitech F310, F710, and Dual Action gamepads as well as the Xbox 360 gamepad (not wireless).

getStickHL() or **stick.horizL()** or **stick.horiz()** [returns double.] Returns a number between -1.0 and +1.0 indicating how much the left thumbstick is displaced horizontally. As indicated in the figure above, a value of -1.0 corresponds to the stick all the way to the left. A value of +1.0 corresponds to the stick all the way to the right. When the stick is released and allowed to return to the center, it returns a value of 0.0.

getStickVL() or **stick.vertL()** or **stick.vert()** [returns double.] Returns a number between -1.0 and +1.0 indicating how much the left thumbstick is displaced vertically. As indicated in the figure above, a value of -1.0 corresponds to the stick all the way to down. A value of +1.0 corresponds to the stick all the way up. When the stick is released and allowed to return to the center, it returns a value of 0.0.

getStickHR() or **stick.horizR()** [returns double.] Returns a number between -1.0 and +1.0 indicating how much the right thumbstick is displaced horizontally. As indicated in the figure above, a value of -1.0 corresponds to the stick all the way to the left. A value of +1.0 corresponds to the stick all the way to the right. When the stick is released and allowed to return to the center, it returns a value of 0.0.

getStickVR() or **stick.vertR()** [returns double.] Returns a number between -1.0 and +1.0 indicating how much the right thumbstick is displaced vertically. As indicated in the figure above, a value of -1.0 corresponds to the stick all the way to down. A value of +1.0 corresponds to the stick all the way up. When the stick is released and allowed to return to the center, it returns a value of 0.0.

stick.button(i) [takes int, returns bool.] The Logitech Dual Action gamepad has twelve buttons. Buttons 1 through 10 are labeled on the gamepad. Buttons 11 and 12 are activated by pushing down the left and right thumbsticks respectively. When given an integer between 1 and 12, the function returns a boolean indicating whether the corresponding button is pressed. A value of true is returned if the button is pressed, false if not pressed.

stick.buttonA(), stick.buttonB(), stick.buttonX(), stick.buttonY(), stick.buttonLB(), stick.buttonRB(), stick.buttonLT(), stick.buttonRT(), stick.buttonBack(), stick.buttonStart(). [returns bool.] Same as stick.button(), except it checks the corresponding buttons on the Logitech F310, F710, and Xbox gamepads.

stick.buttonLStick() and **stick.buttonRStick()**. [returns bool.] Checks status of the buttons connected to the left and right thumbsticks respectively.

More will be posted.