

Comandi supportati

RoboCompass ha 10 comandi di disegno e 12 comandi di supporto. I comandi di disegno possono essere assegnati a etichette e referenziati in altri comandi. In effetti, questi comandi possono essere combinati in modo creativo per produrre ogni sorta di costruzioni geometriche.

Comandi di disegno

`point (x, y)` Esempio: - `point (3,4)` o `A = point (3,4)` dove A è l'etichetta.

`line (x1, y1, x2, y2)` Esempio: - `line (0,0, -2,5)` o `C = line (A, B)` dove A, B sono etichette dei punti. La lunghezza può essere facoltativamente indicata come `line (A, B, 3)` dove 3 è la lunghezza del segmento.

`arc (originPoint, radius, angleFrom, upto)` Esempio: - `arc (point (2,2), 3,40,20)` Usa il punto (2,2) come origine, 3 come raggio e avvia l'arco da 40 gradi e aggiunge altri 20 gradi in senso antiorario, un valore negativo per il parametro "upto" si sposterà in senso orario. Per copiare una linea, assegnare il primo parametro come linea o due punti come mostrato qui `arco (point1, point2, punto origine, angolo da, fino a)`

`perp (line, passThroughPoint, length = 10)` Esempio: - `A = perp (line (1,2,3,4), point (1,2))` disegna una linea perpendicolare a una data linea (1,2,3, 4) passando per il punto (1,2). L'ultimo parametro facoltativo del comando `perp` è la lunghezza della linea perpendicolare che per impostazione predefinita è 10

`parallel (line, passThroughPoint, length = 10)` Esempio: - `parallel (line (1,2,3,4), point (1,2))`

`angle (punto1, punto2, gradi)` Esempio: - `angle (A, B, 45,1)` I due punti A e B definiscono la linea di base. Il rapporto è un parametro opzionale che controlla la posizione del goniometro. Un valore di 0 lo colloca in A, 1 in B e 0,5 nel mezzo.

`poligono (punti separati da virgola)` Esempio: - `poligono (A, B, C)` disegna un triangolo in cui A, B, C sono punti

`findangle (2 linee o un poligono)` Esempio: - `findangle (A, B)` trova l'angolo tra due linee A, B o un poligono `findangle (C)` (C è l'etichetta del poligono, in questo caso il metodo restituisce il primo angolo)

`fill (A, B, fillType = 0, output = 1)` riempie l'interno delle forme A, B. Le forme possono essere un arco, un poligono o un altro riempimento. Il comando accetta un numero qualsiasi di forme. Il `fillType` predefinito è 0 che riempie la regione intersecata, 1 è unione e 2 è differenza. `riempimento (A), riempimento (arco (0,0,3,0,360), D, poligono (2,3,4,6,1,0))`

`trace (punti separati da virgola)` traccia la curva che passa attraverso i punti indicati. `trace (punto (0,0), punto (1, sin (30)), punto (1, sin (60)), punto (1, sin (120)), B)`

Come regola generale, laddove ci si aspetta sempre un punto in un'espressione, possiamo usare sia point (2,3) o usare la sua etichetta o usare un comando helper che calcola e restituisce un punto. Analogamente ovunque ci si aspetta una linea, possiamo dare il etichetta di una linea o etichetta di due punti.

Ad esempio l'arco (A, B, punto (2,1), 50,40) utilizza la distanza da A a B e disegna l'arco dal punto (2,1). (Nota: - anziché il punto (2,1) possiamo usare 'C' se abbiamo un punto etichettato C.

Comandi di aiuto (utili per i calcoli)

dist (punto1, punto2) calcola la distanza tra due punti. Possiamo anche dare dist (C) se C è una linea (cioè un segmento)

X (punto) indica la coordinata X di un punto A A = X (punto (2,1))

Y (punto) indica la coordinata Y di un punto A B = Y (punto (1,2))

pos (poligono o linea o arco, indice) restituisce il punto in un determinato indice. Se un poligono A ha 4 punti, pos (A, 3) restituisce il terzo punto, analogamente se l'etichetta di una linea è B un indice di 2 restituisce il end point Esempio: - pos (B, 2)

intersect (object1, object2, index = 1) interseca due oggetti qualsiasi (tranne i punti). Esempio G = punto (interseca (D, E)) Per impostazione predefinita restituisce il primo punto di intersezione, per il secondo e il terzo punto di intersezione, dare 2 Or 3 Ad esempio, questa intersezione (D, E, 2) fornisce il secondo punto di intersezione.

reflection (oggetto, linea) riflette l'oggetto di una determinata linea. Esempio: - rifletti (D, A) dove D è l'etichetta del punto da riflettere e A è la linea riflettente

rotate (oggetto, angolo, withrespectTo = point (0,0)) ruota l'oggetto di un dato grado. Per impostazione predefinita, la rotazione è rispetto all'origine (0,0), che può essere facoltativamente sovrascritta dando il parametro withrespectPoint.

translate (oggetto, x, y, withrespectTo = point (0,0)) traduce l'oggetto in base a x, y rispetto all'origine o al parametro withrespectTo. tradurre (arco (2,3,3,0,180), 2,3, punto (2,1)) o tradurre (G, 2,4)

dilate (object, scaleFactor, withRespectTo = point (0,0)) dilata l'oggetto con un dato parametro scaleFactor. dilatare (punto (3,1), 2).

project (point1, line) proietta il punto di input sulla linea e restituisce il punto proiettato Esempio: - project (A, line (0,0,1,0)) proietta il punto A su una data riga.

interpolate (punto 1, punto 2, rapporto) Esempio: - interpolate (A, B, 0,5) Un rapporto di 0,5 restituisce il punto medio di A, B

hide (etichette separate da virgola) Esempio: - hide (A, B, C, D) nasconde gli oggetti A, B, C, D. Per mostrare di nuovo usa show (A, B, C, D)

gruppo (oggetti separati da virgola) Le trasformazioni possono essere applicate su più oggetti contemporaneamente creando un gruppo. Ad esempio per ruotare insieme un arco 'A' e un poligono 'B',

utilizzare $C = \text{gruppo}(A, B)$ e quindi $D = \text{ruotare}(C, 120)$ dove C è l'etichetta del gruppo. I gruppi possono anche essere nidificati.

È possibile creare una varietà di forme applicando operazioni booleane su forme semplici, che possono essere tassellate da comandi di trasformazione. Ad esempio $e(A, B, C)$ crea una regione poligonale comune (intersezione) in A, B, C dove A, B, C può essere un arco o un poligono. Allo stesso modo $o(A, E, \text{poligono}(0,0,2,3,4,1))$ e $\text{diff}(A, B)$ creano rispettivamente l'unione e la differenza di determinate regioni. Le aree con fori interni non sono supportate (risultanti da un'operazione xor) ma lo stesso risultato può essere creato combinando più poligoni usando il comando `group`.

Sono supportate tutte le funzioni matematiche standard come `sin`, `cos`, `tan`, `asin`, `acos`, `atan`, `log`, `sqrt`, `max`, `min`. Nota: le funzioni trigonometriche utilizzano i gradi come unità.

Osmosi delle Idee