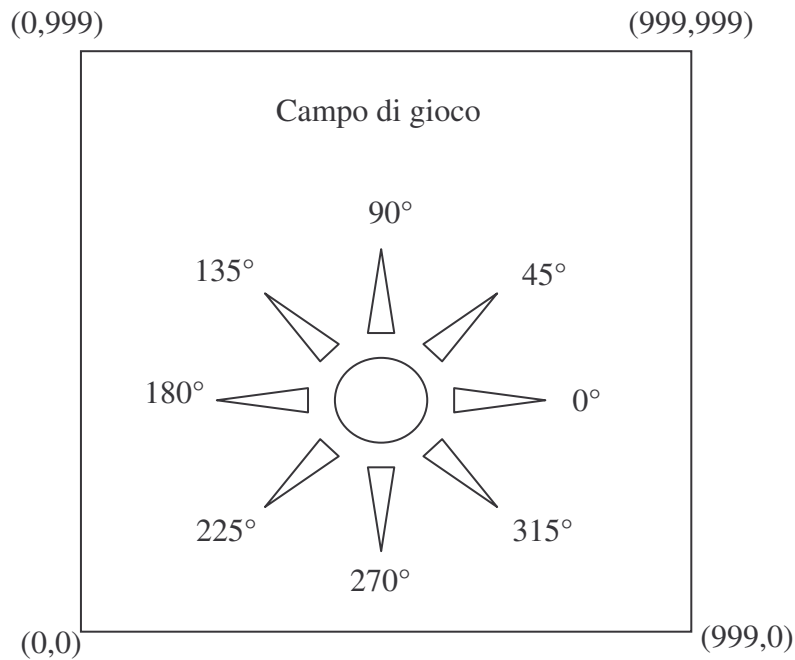


# Manuale essenziale di CROBOTS

(prof. Alessandro Bugatti)



## **Danni**

I danni arrivano ad un massimo di 100 e non possono essere riparati. Il robot mantiene tutte le sue capacità con qualsiasi percentuale di danno.

- 2% Collisione con un altro robot o con un muro (inoltre il robot si ferma)
- 3% Un missile esplode nel raggio di 40 metri
- 5% Un missile esplode nel raggio di 20 metri
- 10% Un missile esplode nel raggio di 5 metri

## **Limiti del linguaggio rispetto all'ANSI C**

Mancano

- preprocessore (niente define e include)
- variabili floating point
- variabili di tipo char e stringhe
- strutture
- unioni
- puntatori
- array
- typedef
- for statement
- do..while statement
- switch..case statement
- break
- continue
- goto e label

- operatore ternario e operatori virgole
- costanti ottali e esadecimali

La lunghezza massima del programma è limitata a 2000 istruzioni macchina.

La dichiarazione delle funzioni è come quella del vecchio C cioè ad esempio la funzione

```
int distance(int x1,int y1,int x2,int y2)
```

diventa:

```
distance(x1,y1,x2,y2)
int x1;
int y1;
int x2;
int y2;
```

dove il valore di ritorno non deve essere dichiarato, basta ritornarlo nel corpo della funzione ad esempio con:

```
return (d) /* da notare le parentesi */
```

## Funzioni

### int scan(int gradi, int risoluzione)

Scandaglia il territorio in cerca di robot nemici. Lo scanner viene diretto verso la direzione individuata da *gradi* (0-359) con un angolo di apertura pari a *risoluzione*. Il valore di ritorno è la distanza del robot più vicino individuato dallo scanner o 0 se nessun robot viene individuato.

La precisione nella risoluzione dello scanner è di +/- 10 gradi

Esempi:

```
range = scan(45,0); /* guarda a 45°, con un apertura di 0° */
range = scan(365,10); /* guarda nella zona tra 355° e 15° */
```

### int cannon(int gradi, int distanza)

Spara un proiettile nella direzione indicata da *gradi* a una distanza indicata da *distanza*. Il massimo valore di distanza è 700, valori superiori vengono troncati.

Ritorna 1 se ha sparato, 0 se il cannone è in fase di ricarica.

Esempi:

```
degree = 45; /* setta la direzione da guardare a 45° */
if ((range=scan(degree,2)) > 0) /* se trova un robot nemico */
    cannon(degree,range); /* gli spara */
```

### void drive (int gradi, int velocita)

Muove il robot nella direzione indicata da *gradi* alla velocità indicata da *velocita*. La velocità è espressa come una percentuale da 0 a 100. I cambiamenti di direzione sono consentiti a una velocità minore di 50.

Esempi:

```
drive(0,100); /* va a est alla velocità massima */
drive(90,0); /* si ferma con il muso girato a nord */
```

## int damage()

Ritorna il danno attuale, compreso tra 0 e 99 (a 100 e oltre il robot muore).

Esempi:

```
d = damage();          /* d conterrà il valore del danno */
```

## int speed()

Ritorna la velocità corrente del robot compresa tra 0 e 100. Potrebbe essere diversa da quella impostata con `drive` a causa di accelerazione e decelerazione.

Esempi:

```
drive(270,100);      /* va verso sud */
... /* altre istruzioni */
if (speed() == 0) /* il robot si è scontrato */
{
    drive(90,20); /* riparto in un'altra direzione */
}
```

## int loc\_x()

## int loc\_y()

Ritornano rispettivamente il valore della posizione x e y del robot, comprese tra 0 e 999

Esempi:

```
while (loc_x() > 20); /* non far niente finchè non sei vicino al muro ovest*/
```

## int rand(int limite)

Produce un numero casuale compreso tra 0 e *limite*, con *limite* al massimo 32767

## int sqrt(int numero)

Calcola la radice quadrata di *numero* (se *numero* è negativo gli cambia segno). Da notare che il risultato è approssimato ad un intero.

## int sin (int gradi)

## int cos (int gradi)

## int tan (int gradi)

Calcolano rispettivamente seno, coseno e tangente di *gradi*. Ritornano il valore moltiplicato per 100000 (cioè se il seno di 52 gradi è 0.788010 ritorna 78810)

## int atan (int rapporto)

Ritorna l'arcotangente di *rapporto*. Rapporto deve essere moltiplicato per 100000 prima di essere passato a `atan`. Il valore di ritorno è compreso tra -90 e +90.