

# Progetto: RoboGP

RoboGP intende essere una versione elettronica del gioco da tavolo RoboRally.

Nel seguito descriveremo innanzitutto il gioco (in una versione già adattata per poter essere implementata). Successivamente spiegheremo come immaginiamo l'applicazione.

In breve, il gioco consiste in una corsa di robot. Il **robodromo** – la “pista” su cui avviene la corsa – è un vero e proprio campo minato, irto di ostacoli e pericoli, che i robot devono scansare per “toccare” in un ordine prestabilito i checkpoint numerati. Il primo a raggiungere l'ultimo checkpoint sarà il vincitore.

I giocatori hanno a disposizione istruzioni con cui programmare le prossime mosse di ciascun robot, ma quando i programmi vengono avviati le istruzioni di un robot potrebbero interferire con quelle degli altri, sortendo effetti inaspettati e... indesiderati.

## Il gioco

RoboGP è un gioco per 2-8 giocatori. Ciascun giocatore può controllare 1 o più pedine (robot) che si muovono su un tabellone (robodromo) suddiviso in caselle. Il tabellone rappresenta un percorso ad ostacoli. Lo scopo del gioco è toccare per primi con uno dei propri robot le bandierine (checkpoint) numerate presenti sul robodromo, nell'ordine da esse indicato.

## Le basi del gioco

### I robot

Ciascun giocatore (il gioco prevede da 2 a 8 giocatori in generale) controlla uno o più robot. In

figura si possono vedere i “ritratti” dei segnalini-robot originariamente previsti dal gioco.

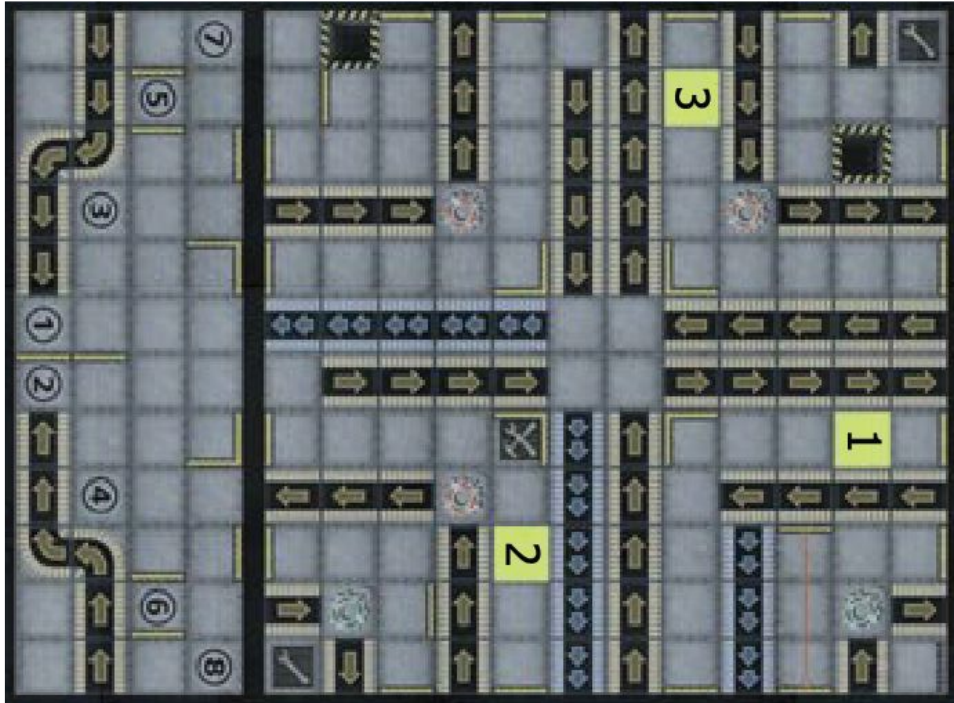


In RoboRally ciascun robot ha **cinque registri programmabili** (I-V), e inizia la gara con **dieci punti salute** e **tre vite**.

Ciascun robot è anche dotato di un **laser** che spara dritto dinanzi a lui e si interrompe solo quando incontra un ostacolo (tipicamente, una parete, o un altro robot...)

## Il robodromo

Il robodromo è un tabellone di gioco a base quadrata o rettangolare diviso in caselle, che rappresenta un "campo di gioco" con ostacoli e pericoli. Sul robodromo sono rappresentate anche le posizioni di partenza (dock) dei concorrenti-robot e i checkpoint numerati che i robot devono toccare (nell'ordine della numerazione) per procedere nella gara.



Questo tabellone di esempio può avere da 2 a 8 concorrenti, e prevede 3 checkpoint.

Si vedono anche altri elementi del percorso:

- i **buchi neri**, che costringono il robot a ripartire dall'ultimo checkpoint attraversato, o dal via;
- i **nastri di scorrimento** (singolo e doppio) che spostano automaticamente il robot
- le **stazioni di riparazione**, che riparano il robot da eventuali danni subiti
- i **muri**, che limitano il passaggio fra le caselle
- le **rotatorie**, che fanno girare il robot su se stesso
- i **raggi laser**, che danneggiano il robot

Il gioco include diversi robodromi predefiniti.

## La programmazione dei robot

In RoboGP i giocatori assumono il ruolo dei programmatori dei robot.

Ad ogni manche, il giocatore può programmare il proprio robot inserendo nei suoi cinque registri delle schede-istruzione, scelte dal proprio **pool personale**, che è normalmente composto di nove schede. Il programmatore può quindi definire le prossime cinque mosse del suo robot. Le schede-istruzione vengono assegnate casualmente ai giocatori estraendole da un **pool globale** di 84 schede-istruzione. I tipi disponibili di schede-istruzione sono i seguenti:



- **Back-up:** il robot fa un passo indietro senza voltarsi. Ci sono in tutto 6 schede **Back up**.
- **Move X (X=1,2,3):** il robot fa X passi avanti. Ci sono in tutto 18 schede **Move 1**, 12 schede **Move 2** e 6 schede **Move 3**.
- **Turn left/right:** il robot si gira di 90 gradi a sinistra/destra. Ci sono in tutto 18 schede per ciascuno dei due tipi di **Turn**.
- **U-turn:** il robot si gira su se stesso di 180 gradi. Ci sono in tutto 6 schede **U-turn**.

Ogni scheda ha anche una priorità (riportata in alto, sopra l'istruzione); quando più robot eseguono istruzioni che interagiscono l'una con l'altra, quella a priorità più alta viene eseguita per prima. Le schede hanno tutte diversa priorità e le priorità sono tutte multipli di 10.

La seguente tabella ricapitola i tipi di schede-istruzione e le loro priorità:

Tipo di Scheda	Numero	Priorità (step 10)
U-turn	6	da 10 a 60
Turn left	18	da 70 a 410 (step 20)
Turn right	18	da 80 a 420 (step 20)
Back-up	6	da 430 a 480
Move 1	18	da 490 a 660
Move 2	12	da 670 a 780
Move 3	6	da 790 a 840

Alla fine di ogni manche, le schede usate vengono scartate (potranno venire rimesse in circolo nel momento in cui i giocatori avranno esaurito le 84 inizialmente disponibili).

All'inizio di ogni manche, il programmatore riceve tante nuove schede quante ne servono a ripristinare il pool, che come detto è normalmente di 9 schede. La dimensione del pool tuttavia può calare quando il robot perde punti salute (dimensione pool = punti salute-1). Inoltre, se i punti salute si riducono troppo, i registri possono **bloccarsi**: la scheda rimane incastrata e non è più possibile rimuoverla o sostituirla sino a che il robot non viene riparato. Per riparare il robot si può terminare la manche di gioco su una casella Riparazione (o Upgrade&Riparazione) (simbolo della chiave inglese, eventualmente accompagnata da un martello) o Checkpoint (bandierina numerata) – che fanno riacquisire un punto salute – o decidere di spegnere il robot per una manche, per ripararlo completamente.

Vedi la sezione [Funzionamento del robodromo](#) per maggiori dettagli sulle caselle Riparazione, Upgrade&Riparazione e Checkpoint, e la sezione [Robot danneggiati o distrutti](#) per maggiori dettagli su punti salute, vite, e sul blocco dei registri.

## Il gioco in breve

Il gioco inizia estraendo a sorte su quale rampa di lancio partirà ciascun robot (se ci sono meno di 8 giocatori, si usano solo le rampe 1-N, dove N è il numero di giocatori). Per rendere il gioco

più movimentato, si può anche decidere di dotare ciascun robot di un **upgrade** scelto a caso sin dall'inizio del gioco (vedi sezione [Upgrade](#)). In caso contrario, per ottenere un upgrade sarà necessario fermarsi su una casella Upgrade&Riparazione

Il gioco si svolge in più manche, che proseguono finché un robot (o i primi 3, se si vuole stabilire un "podio" di vincitori) non ha tagliato il traguardo.

Ciascuna manche si compone dei seguenti passi:

1. **Ricezione delle schede-istruzione.**
2. **Programmazione:** Ciascun giocatore programma il proprio robot. In questa fase ciascun giocatore indica anche se vuole o meno spegnere il proprio robot nella manche successiva.
3. **Esecuzione dei registri:** I registri vengono eseguiti uno ad uno, da I a V. Per ciascuno si effettuano:
  - A. **Dichiarazione:** si rivela a tutti i giocatori il contenuto del registro
  - B. **Mossa:** i robot eseguono la propria istruzione in ordine di priorità
  - C. **Attivazione robodromo:** gli elementi del robodromo si attivano
  - D. **Laser & Armi:** i laser sparano
  - E. **Touch & Save:** i checkpoint e i punti riparazione vengono "toccati".
4. **Fine manche:** Si **gestiscono** gli effetti di fine manche e si resettano i robot per la manche successiva.

## Esecuzione di una manche di gioco

### 1. Ricezione delle schede-istruzione

I giocatori ricevono tante schede-istruzione quante ne servono per riempire il loro pool (ricordiamo che la dimensione del pool è pari a numero di punti salute rimasti al robot – 1).

### 2. Programmazione

Come detto in precedenza ciascun giocatore sceglie (senza che gli altri vedano quello che fa) quali schede-istruzione inserire in ciascun registro (salvo eventuali registri bloccati a causa di danni subiti). Il giocatore inoltre può indicare se desidera che il proprio robot si spenga, per ripararsi, nella manche successiva. **Attenzione: se il robot dovesse venire distrutto in questa manche, l'indicazione di spegnimento viene invalidata.**

Quando un giocatore ha terminato lo **dichiara**. Quando tutti i giocatori tranne uno hanno terminato parte un conto alla rovescia di 30 secondi al termine del quale, se un giocatore non ha finito, i registri del suo robot verranno riempiti con schede scelte a caso dal suo pool.

Quando tutti i robot sono programmati o il conto alle rovescia è terminato, si dà il via alla fase di **Esecuzione**.

### 3. Esecuzione

All'inizio della fase di esecuzione, i **robot spenti** riguadagnano tutti i propri punti salute. Siccome, anche se sono spenti, i robot sono presenti sul tabellone, essi potrebbero comunque

ri-perdere punti salute nel corso della manche.

La fase di esecuzione effettua un ciclo sui registri. Per ogni registro R (R=I, II, III, IV, V) si susseguono le seguenti sotto fasi.

#### **A. Dichiarazione**

La scheda contenuta nel registro R di tutti i robot viene mostrata a tutti i giocatori e resta visibile sino a fine manche.

#### **B. Mossa**

I robot eseguono l'istruzione in R **in ordine di priorità** (dalla più alta alla più bassa). La scheda-istruzione in corso di esecuzione viene evidenziata e il robot si muove secondo quanto indicato da essa. Il movimento si interrompe se incontra una parete. Se il robot ne incontra un altro sul suo cammino, lo spinge (ma non lo danneggia in questa fase, a meno dell'uso dell'apposito Upgrade); non può però spingerlo oltre una parete. Se il robot spintonato incontra una parete, il movimento di entrambi si interrompe.

#### **C. Attivazione robodromo**

Il robodromo ha elementi attivi, che possono spostare i robot dalla loro posizione. Questi elementi sono: i **nastri trasportatori semplici**, i **nastri trasportatori express**, i **respingenti** e le **rotatorie** (per i dettagli, vedere il [Funzionamento del robodromo](#)). Essi effettuano le proprie azioni nel seguente ordine:

- i nastri trasportatori express si attivano una prima volta e spostano i robot che si trovano su di essi di una posizione nella direzione delle frecce.
- si attivano nuovamente i nastri trasportatori express, insieme a quelli normali, e spostano i robot che si trovano su di essi di una posizione nella direzione delle frecce
- si attivano i respingenti
- si attivano le rotatorie, ruotando il robot di 90 gradi nella direzione delle frecce

**Attenzione:** alcuni nastri trasportatori oltre che uno spostamento provocano anche una rotazione, evidenziata dalla freccia che è curva anziché dritta. **La rotazione avviene solo se il robot arriva su questa casella da un altro nastro trasportatore e viene applicata quando il robot arriva, non quando parte.**

#### **D. Laser & Armi**

I robot che si trovano sulla linea di uno o più raggi laser al termine della fase di esecuzione perdono un punto salute per ciascun raggio da cui sono colpiti. I raggi laser possono essere emessi dal robodromo o da un altro robot. Il raggio laser di un robot colpisce in linea retta davanti al robot stesso. I raggi laser si interrompono quando colpiscono una parete o un ostacolo, quindi ciascun raggio laser può colpire un solo robot alla volta (salvo eccezioni dovute a upgrade).

**Attenzione:** un robot fra il movimento suo, le spinte degli altri e i nastri trasportatori, può attraversare diverse caselle nel corso dell'esecuzione di un solo registro. **Perché i laser lo danneggino deve trovarsi sulla linea di fuoco in questa sotto-fase.** Se vi transita soltanto durante le sottofasi precedenti non perde punti salute.

**Attenzione:** se il robot dispone di un upgrade, il giocatore può decidere di sacrificarla invece di far perdere un punto salute al suo robot.



## E. Touch & Save

I Checkpoint e i punti Riparazione (o Upgrade&Riparazione) vengono "toccati", ossia la posizione del robot viene salvata in quel punto e il progresso nella gara viene registrato. Se il robot ha toccato tutti i checkpoint nell'ordine, ha terminato la gara. Se il robot dovesse perdere una vita, ricomincerà dall'ultima posizione salvata.

**Attenzione:** perché la funzione di riparazione e/o upgrade di queste caselle si attivi, è necessario che il robot si trovi ancora sulla casella al termine della manche. Non è sufficiente che ci passi sopra.

## Attivazione upgrade

Durante la fase di esecuzione, il giocatore può dichiarare in qualunque momento (=attivare) se vuole utilizzare un **upgrade** ; l'upgrade verrà in ogni caso consumato e applicato nella prima sottofase valida **successiva** a quella di attivazione. Se l'esecuzione termina prima che possa essere utilizzato, esso verrà "sprecato".

**Esempio:** un giocatore ha ricevuto 5 bombe come upgrade. Le bombe vengono utilizzate durante la sottofase "Laser & Armi" (D). Supponiamo che il giocatore voglia far esplodere una bomba durante l'esecuzione del registro I: dovrà attivarla **prima** che inizi la sottofase D, quindi durante una delle sottofasi A-C. Se il giocatore si sbaglia e attiva la bomba ad es. nella sottofase D, essa non potrà venire utilizzata in quella stessa sottofase, ma verrà utilizzata nella sottofase D del registro successivo, ossia il II. Se il giocatore, per errore, attiva la bomba nella sottofase D o E del registro V, non sarà possibile effettivamente utilizzarla; tuttavia al giocatore la bomba risulterà utilizzata (ossia possiederà una bomba in meno).

I dettagli sul funzionamento di ciascun upgrade sono indicati nelle istruzioni specifiche ad esso relative.

## 4. Fine Manche

Una volta terminata l'esecuzione del registro V:

- si applicano gli effetti di riparazione delle caselle Checkpoint, Riparazione e Upgrade&Riparazione, ossia il robot riacquista un punto salute
- si applicano gli effetti di upgrade delle caselle Upgrade&Riparazione: il robot riceve un upgrade e il giocatore può visualizzarne le istruzioni. Gli altri giocatori vengono notificati dell'upgrade installato.
- si resettano i registri che non sono bloccati, scartando le schede-istruzione che contenevano.

## Robot danneggiati o distrutti

Un robot può essere danneggiato nel corso della gara (da un laser) e in tal caso perde punti salute.

Quando un robot perde un punto salute, il pool di schede a disposizione del programmatore, che normalmente è di 9 schede, si riduce di uno: il pool di schede ha dunque una dimensione pari a punti salute – 1.

Quando il numero di punti salute scende a 5, oltre che ridursi il pool di schede, vengono anche bloccati ad uno ad uno i registri al loro stato attuale, partendo dall'ultimo (V) e andando verso

il primo (I).

Il blocco di un registro significa che la scheda è incastrata, non viene eliminata al termine della manche né sostituita in quella successiva.

Quindi con 5 punti salute viene bloccato solo il registro V, con 4 punti salute vengono bloccati i registri IV e V, e così via: un robot con 1 punto salute ha tutti i registri bloccati. Un robot con 0 punti salute viene **distrutto**. La seguente tabella ricapitola cosa succede:

Punti Salute	Effetto
10	Pool di 9 schede
9	Pool di 8 schede
8	Pool di 7 schede
7	Pool di 6 schede
6	Pool di 5 schede
5	Pool di 4 schede, registro V bloccato
4	Pool di 3 schede, registri IV e V bloccati
3	Pool di 2 schede, registri III, IV e V bloccati
2	Pool di 1 scheda, registri II, III, IV e V bloccati
1	Pool di 0 schede, tutti i registri bloccati
0	Distrutto

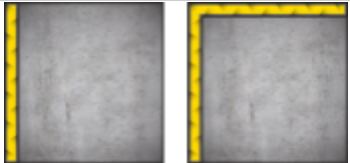
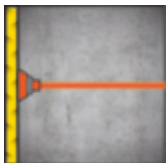

Un robot può venire distrutto anche se **cade in un buco nero** o **finisce fuori dal robodromo**. Per cadere in un buco nero è sufficiente passarci sopra.






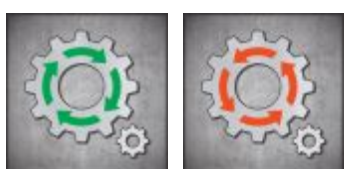
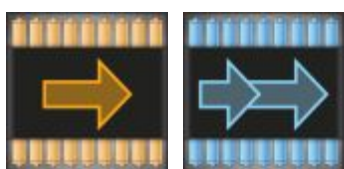
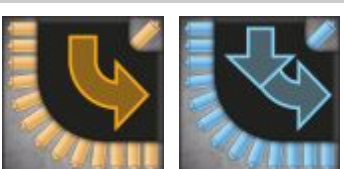
Un robot distrutto perde una vita e ricomincia dall'ultima posizione salvata, o dalla posizione di partenza se non ha mai incontrato un punto di salvataggio (ricordiamo che i punti di salvataggio sono le caselle Checkpoint, Riparazione e Upgrade&Riparazione, e che i salvataggi avvengono nella sottofase **Touch (E)** della fase di **Esecuzione (3)**).

Se il robot esaurisce le vite, esce dal gioco.


## Funzionamento del robodromo

Riportiamo qui i tipi di elementi che si possono trovare nel robodromo.

	<b>Pareti:</b> impediscono il passaggio del robot
	<b>Laser:</b> danneggia il primo robot che si trova sulla sua linea di fuoco
	<b>Respingenti:</b> spingono il robot di una posizione. I respingenti

	<p>pari si attivano solo durante l'esecuzione dei registri II e IV, i respingenti dispari solo durante l'esecuzione dei registri I, III e V.</p>
	<p><b>Dock o Rampa di partenza:</b> indica la posizione di partenza dei robot nel circuito. I dock sono numerati da 1 a 8.</p>
	<p><b>Checkpoint:</b> per vincere la gara i robot devono "toccare" i checkpoint nell'ordine di numerazione. Su un robodromo possono esserci fino a 4 checkpoint. I checkpoint sono anche punti di riparazione.</p>
	<p><b>Riparazione:</b> un robot che tocca un punto di riparazione vede la sua posizione salvata. Se un robot termina la <b>manche</b> su un punto riparazione, riacquisisce un punto salute.</p>
	<p><b>Upgrade&amp;Riparazione:</b> si comporta come un punto Riparazione. In più, se un robot termina la <b>manche</b> su un punto upgrade, acquisisce un upgrade estratto a sorte.</p>
	<p><b>Buco nero:</b> se il robot ci cade dentro viene immediatamente distrutto.</p>
	<p><b>Rotatoria:</b> il robot viene ruotato di 90° nella direzione delle frecce (verde = senso orario, arancione = senso antiorario).</p>
	<p><b>Nastri trasportatori:</b> traslano il robot nella direzione delle frecce. I nastri trasportatori semplici (color arancione) si attivano una volta sola nel corso dell'esecuzione di un registro, mentre i nastri express (color azzurro) si attivano per due volte. Il robot viene traslato se al momento dell'attivazione del nastro si trova su questa casella.</p>
	<p><b>Nastri trasportatori con curva:</b> <u>in aggiunta al comportamento dei nastri trasportatori normali</u>, se il robot viene depositato su di essi da un altro nastro trasportatore proveniente dalla direzione di ingresso indicata dalla freccia, lo ruotano <u>immediatamente</u> nella direzione della freccia. Dunque parte dell'effetto di questi nastri viene operato quando il robot <b>arriva</b> sulla casella a causa dell'attivazione dei nastri, non quando ci è</p>



	già sopra. (La figura mostra due rotazioni a sinistra, ci sono anche le rotazioni a destra)
	<b>Nastri trasportatori misti:</b> si comportano come i precedenti, ma ci sono più direzioni di ingresso che attivano l'eventuale rotazione. (Quelli in figura sono solo alcuni esempi delle combinazioni possibili)

## Upgrade

Gli upgrade sono componenti usa e getta che permettono a un robot di effettuare azioni particolari durante la fase di esecuzione.

### Utilizzo nel gioco

Quando un robot **termina una manche** su una casella Upgrade&Riparazione, riceve un package upgrade a caso fra quelli disponibili. Il package contiene 3 o 5 istanze usa e getta di un determinato upgrade.

Opzionalmente, si può scegliere di iniziare la gara dotando già ciascun robot di un package upgrade, sempre scelto a caso.

Gli upgrade devono essere esplicitamente attivati dal giocatore che controlla il robot **durante la fase di Esecuzione (3)**; perché siano efficaci l'attivazione deve avvenire *prima dell'inizio della sottofase di utilizzo* (secondo le istruzioni specifiche dell'upgrade).

L'upgrade viene consumato quando viene attivato, anche qualora non dovesse avere nessun effetto a causa di un'attivazione erronea.

### Upgrade disponibili

Upgrade + dim. package	Sottofase di utilizzo	Effetto	Disponibilità (nr. package)
Random Switch (x5)	A	La scheda istruzione del prossimo registro viene sostituita con una presa a caso dal pool globale	2
Copia & Incolla (x3)	A	La scheda istruzione del prossimo registro viene sostituita con una copia di quella attuale.	1
Freni (x5)	B	Non esegue l'istruzione nel registro	2
Acceleratore (x3)	B	Se nel registro c'è una Move X, esegue invece Move (X+1)	1
Retromarcia	B	Se nel registro c'è un Back-up, arretra di due	1

(x5)		caselle anziché di una.	
Shift (x5)	B	Se nel registro c'è un Turn left/right, avrà invece l'effetto di una Shift left/right (ossia il robot si sposta lateralmente senza ruotare)	2
Dual Core (x3)	B	L'istruzione nel registro viene eseguita due volte in immediata successione.	1
Disturbatore di frequenze	C	I nastri trasportatori semplici non si attivano. I nastri trasportatori express si attivano una volta sola. Il resto del robodromo si comporta normalmente.	2
Giroscopio (x5)	C	Il robot non viene ruotato da rotatorie né nastri trasportatori	2
Scudo (x3)	D	Assorbe 1 danno ricevuto dal robot durante la sottofase Laser&Armi immediatamente successiva.	2
Bomba	D	Invece di sparare il laser, il robot lancia una bomba, che colpisce qualunque altro robot si trovi nel quadrato 5x5 di caselle intorno ad esso.	2
Superlaser (x3)	D	Il laser del robot dopo aver colpito il primo ostacolo sul suo percorso (parete o altro robot) ne colpisce un secondo.	1
Doppio Laser (x3)	D	Il laser del robot provoca due danni.	1
Raggio respingente (x5)	D	Il raggio del robot non provoca danni ma spinge il robot colpito di una posizione (a meno che non ci sia una parete a impedirlo).	1
Raggio attrattore (x5)	D	Il raggio del robot non provoca danni ma attrae il robot colpito verso di sé di una posizione (a meno che non ci sia una parete a impedirlo).	1
Raggio controllante (x3)	D	Il raggio del robot non provoca danni ma sostituisce completamente il programma del robot colpito con quello di chi ha sparato.	1
Raggio disturbante (x3)	D	Il raggio del robot non provoca danni ma rimescola fra loro le schede istruzione del robot colpito.	1
Retro Laser (x3)	D	Il robot spara un colpo dinanzi a sé e un colpo dietro di sé.	1



# L'applicazione RoboGP

L'applicazione RoboGP dovrà essere un'applicazione desktop multi-piattaforma.

Essa dovrà prevedere due modalità di utilizzo:

**Modalità Sfida**, dove più giocatori si sfidano in rete (un giocatore attiverà il server, a cui tutti poi si collegheranno come client) al gioco così come precedentemente descritto.

**Modalità Allenamento**, dove un giocatore da solo può allenarsi a programmare il proprio robot su un robodromo a sua scelta.

In release future si prevede di inserire anche una **Modalità creazione robodromo** per permettere ai giocatori di creare robodromi personalizzati, ma per il momento i giocatori potranno utilizzare solo i robodromi predefiniti messi a disposizione con l'applicazione.

## Modalità sfida

La modalità sfida richiede che uno dei giocatori attivi un server di partita a cui poi gli altri possano collegarsi per giocare. Il proprietario del server dovrà poter scegliere se accettare o meno le richieste di connessione (ossia, di partecipazione alla partita).

Il giocatore che attiva il server ha anche la responsabilità di scegliere le caratteristiche della partita, in particolare il numero di giocatori ammessi, il numero di robot controllati da ciascun giocatore, il robodromo su cui giocare, l'eventuale dotazione iniziale di upgrade dei robot, la modalità di conclusione della partita. In particolare si dovrà poter scegliere se terminare la partita al primo robot che taglia il traguardo, ai primi 2 o ai primi 3 (è possibile volendo fornire anche altre possibilità, ma tendenzialmente il gioco tende a diventare troppo lungo se si va oltre i primi 3).

Sul robodromo sono ammessi al massimo 8 robot, quindi il "gestore della partita" potrà decidere di ammettere fino a 8 giocatori con 1 robot a testa, o fino a 4 con 2 robot a testa, o 2 giocatori con 3 o 4 robot a testa.

Una volta presenti tutti i giocatori la partita potrà avere inizio e procederà secondo le regole del gioco sopra descritte.

La partita dovrebbe poter essere salvata (fra una manche e l'altra) e ripresa in un momento successivo.

## Modalità allenamento

In questa modalità il giocatore è solo e può allenarsi su un robodromo a sua scelta.

Allenarsi significa poter scrivere un programma, costruito mettendo in sequenza le istruzioni base presenti sulle schede-istruzione, ma senza limitazioni sulla scelta delle schede o sulla lunghezza del programma stesso, per poi eseguirlo. In fase di programmazione dovrà essere possibile manipolare liberamente la sequenza, aggiungendo, togliendo o modificando l'ordine delle istruzioni.

Eseguire il programma vuol dire eseguirne a una a una le istruzioni, attivando dopo ciascuna

gli elementi del robodromo (esclusi i laser, in allenamento i robot non si possono danneggiare).

Durante l'esecuzione il giocatore può mettere in pausa il programma, tornare a un punto precedente nella sua esecuzione, o fermarlo definitivamente per poterlo modificare.

Il giocatore deve anche poter scegliere liberamente da quale posizione far partire il robot e in quale direzione deve essere inizialmente voltato.

## Requisiti tecnici

Si richiede di implementare l'applicazione in linguaggio Java.

Si lascia libertà di scelta ai progettisti riguardo alla modalità e al formato di salvataggio dei dati della partita in corso.

Per quanto riguarda la modalità sfida, ai fini della regolarità del gioco, è importante che la logica del gioco stesso sia gestita dal server. In altre parole dovrà essere il server a vagliare la correttezza delle mosse dei giocatori e a determinarne l'esito. Il ruolo dei client è quello di permettere ai giocatori di scegliere le proprie mosse/azioni di gioco, e di mostrare l'andamento del gioco.

## Requisiti di interazione

Il gioco è reso interessante dal fatto che non è sempre semplice valutare in modo sufficientemente rapido quali siano le istruzioni più corrette per portare il proprio robot dove si desidera, né come reagire – ammesso che sia possibile – alle interazioni impreviste fra i robot. Per questa ragione vanno tenute in conto le seguenti considerazioni:

- La fase di programmazione non deve dare ai giocatori tutto il tempo che vogliono, ma come da regole di gioco, deve iniziare un conto alla rovescia non appena il penultimo giocatore ha terminato di programmare. Nella fase di programmazione è inoltre importante che il giocatore veda quali altri giocatori hanno terminato e quali no.
- La fase di esecuzione deve avvenire sottofase per sottofase, da un lato dando il tempo ai giocatori di capire cosa sta succedendo e giocare gli upgrade, dall'altro senza diventare tediosa. Nelle sottofasi che prevedono una animazione in grafica (B,C,D) sarà probabilmente sufficiente lasciare qualche secondo, con un conto alla rovescia, in aggiunta al tempo di visualizzazione delle animazioni. Nelle sottofasi A e E, che sono visualizzate istantaneamente, si dovrà prevedere un tempo di reazione per i giocatori leggermente più lungo.
- Dovrà essere sempre mostrato chiaramente ai giocatori quali sono gli esiti di ciascuna sottofase (quindi: eventuali danni subiti, salvataggio posizione, riparazioni, upgrade acquisiti, checkpoint conquistati) **oltre** a quanto verrà naturalmente mostrato tramite la grafica del tabellone di gioco.

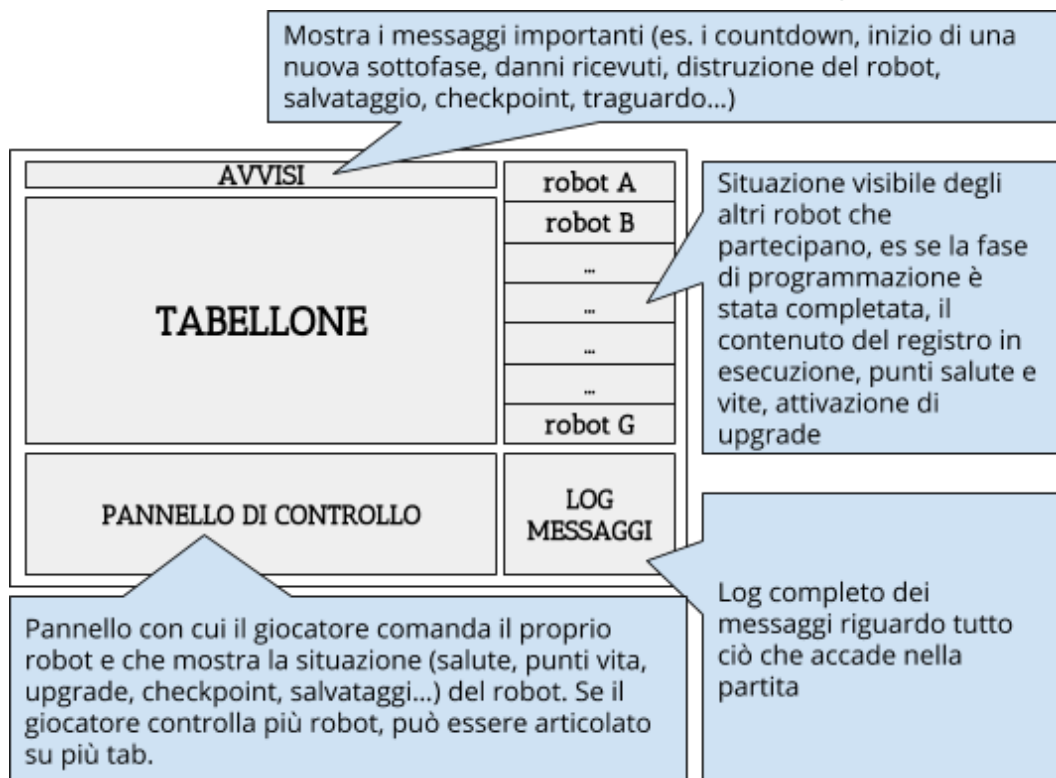


# GUI

I nostri grafici metteranno a disposizione un package per la visualizzazione del tabellone di gioco e l'animazione delle pedine, tramite il quale sarà possibile realizzare gli effetti necessari a mostrare l'andamento del gioco.

Le seguenti figure mostrano sketch di possibili GUI di gioco/allenamento studiate dai nostri grafici; sarà comunque possibile proporre varianti se dovessero emergere necessità diverse o idee migliori. Non sono state abbozzate le GUI per le fasi di configurazione (del server, della partita o dell'allenamento); si prevede non presentino particolari difficoltà.

## Proposta GUI per modalità sfida - fase di gioco



## Proposta GUI per modalità allenamento

