

POLITECNICO DI MILANO
FACOLTÀ DI INGEGNERIA DELL'INFORMAZIONE

PROGETTO DI LABORATORIO DI HUMAN COMPUTER INTERACTION

PROF. GARZOTTO
PROF. PAOLINI

ROBOGAME DESIGN

ROBOWII 2.1

PROGETTO DI:
DIEGO MEREGHETTI
ALESSANDRO MARIN

2008-09

Indice

1	Introduzione	5
2	Risorse	6
3	Stato dell'Arte	7
3.1	RoboWII	7
3.1.1	RoboWII 1.0	7
3.1.2	RoboWII 2.0	8
3.1.3	RoboWII 2.0.1	8
3.1.4	RoboWII 2.0.L	10
3.1.5	RoboWII 3.0 (Jedi Training)	10
3.2	Considerazioni	10
4	Descrizione del gioco	13
4.1	Obiettivi	13
4.2	Requisiti	13
4.3	Regole di base	14
4.4	Regole avanzate	15
4.4.1	Movimento del robot	15
4.4.2	Waypoint	15
4.4.3	Effetto dei colpi del robot	15
4.4.4	Effetto dei colpi del giocatore	16
4.4.5	Armi disponibili	17
5	Interfaccia utente	19
5.1	Funzionalità	19
5.1.1	Associazione funzionalità - interfaccia	20
5.2	Altre interazioni	22
6	Specifiche tecniche e studio di fattibilità	25
6.1	Macchina a stati	25

6.1.1	Armi del giocatore	26
6.1.2	Puntamento delle armi del giocatore	32
6.1.3	Energia del robot	35
6.1.4	Stamina del robot	36
6.1.5	Aggressività del robot	38
6.1.6	Vittoria del robot	39
6.2	Hardware occorrente	40
6.2.1	Robot autonomo mobile	41
6.2.2	Controller WiiMote	43
6.2.3	Giubbotto di materiale rifrangente	43
6.2.4	Identificatore waypoint e casa base	43
6.2.5	Unità di elaborazione esterna	44
6.2.6	Comunicazione	44
6.3	Studio di fattibilità	44
6.3.1	Riconoscimento movimenti del gioco	45
6.3.2	Riconoscimento del giocatore da parte del robot	48
6.3.3	Riconoscimento dei waypoint da parte del robot	49
6.3.4	Riconoscimento della casa base da parte del robot	50
6.3.5	Riconoscimento degli ostacoli da parte del robot	51
6.3.6	Puntamento del WiiMote verso il robot	52
6.3.7	Rilevamento della eccessiva vicinanza del giocatore al robot	52
7	Esempi di scenario	53
7.1	Scenario: vittoria robot	53
7.2	Scenario: vittoria giocatore	55
8	Descrizione del prototipo	57
9	Conclusioni	59
A	Glossario	60

Elenco delle figure

3.1	Robot utilizzato per il progetto RoboWII 1.0	8
3.2	Spykee, utilizzato per il progetto RoboWII 2.0	9
3.3	LEGO NXT	10
3.4	Schema gioco RoboWII 3.0	11
5.1	Controller WiiMote	20
5.2	Posizione dei tasti sul WiiMote	21
6.1	Macchina a stati armi	27
6.2	Macchina a stati puntamento armi del giocatore	33
6.3	Macchina a stati energia del robot	35
6.4	Macchina a stati stamina del robot	37
6.5	Macchina a stati aggressività del robot	38
6.6	Macchina a stati vittoria del robot	39
6.7	Rovio	42
6.8	Spykee	42
6.9	Estrazione delle armi	46
6.10	Rifodero delle armi	47
6.11	Ricarica dell'arma	48
6.12	Riconoscimento del giocatore da parte del robot	49
6.13	Led piano	50
6.14	Riconoscimento di uno dei waypoint da parte del robot	50
6.15	Riconoscimento della casa base da parte del robot	51
6.16	Led infrarossi montati sul robot	52
7.1	Esempio di partita vinta dal robot	54
7.2	Esempio di partita vinta dal giocatore	56

Capitolo 1

Introduzione

RoboWII 2.1 è un gioco basato sull'interazione tra un robot mobile autonomo e giocatori umani muniti di *Wii Remote* — *WiiMote* — il principale controller della console *Nintendo Wii*.

RoboWII 2.1 tramite alcune semplici regole consente al giocatore di interagire con l'automa, il quale è in grado di prendere decisioni in maniera autonoma, tramite l'utilizzo del sistema di controllo offerto dal *WiiMote*.

Capitolo 2

Risorse

Il progetto è stato realizzato da due persone, suddividendo il carico di lavoro — circa di 320 ore — in modo equo, quindi assumendo circa 160 ore di lavoro a testa.

I ruoli sono stati equivalenti tra di loro ed il lavoro è stato portato a termine tramite diverse fasi di discussione, analisi delle problematiche e realizzazione del prototipo — sia dal punto di vista programmatico che da quello più strettamente meccanico —, cercando di suddividere dove possibile le attività in modo da poter parallelizzare lo sviluppo.

Capitolo 3

Stato dell'Arte

In questo capitolo verrà svolta una breve panoramica relativa ai lavori già esistenti che trattano diversi aspetti legati al lavoro qui trattato, in maniera da fornire un punto di vista su cosa è già disponibile e cosa è oggetto di ricerca in questo campo.

3.1 RoboWII

Il progetto RoboWII 2.1 consiste nell'estensione e nell'esplorazione di differenti alternative di design, relative al più generale progetto RoboWII [5] dell'AIRLab [1] all'interno del Politecnico di Milano, esso consiste nella realizzazione di giochi in cui sia presente una forte componente di interazione tra utente umano e robot autonomo.

Lo strumento principale che il giocatore ha a disposizione per interagire con il robot consiste nel WiiMote, che quindi costituisce la principale interfaccia di gioco.

Nel seguito di questo capitolo verranno analizzati i diversi lavori RoboWII già in corso.

3.1.1 RoboWII 1.0

Il progetto RoboWII 1.0 [10] è stato realizzato da Antonio Bianchi e Ben Chen. Esso consiste in un gioco nel quale il giocatore deve puntare un led posizionato sul robot tramite la telecamera a infrarossi del WiiMote mentre il robot cerca di raggiungere una posizione determinata, senza essere intercettato.

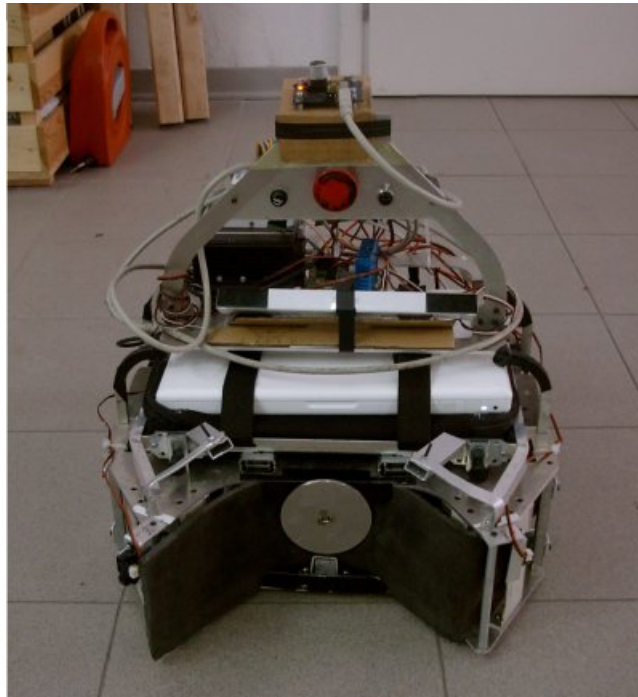


Figura 3.1: Robot utilizzato per il progetto RoboWII 1.0

Per la realizzazione è stato utilizzato uno dei robot impiegati nella RoboCup [4], rappresentato nella Figura 3.1, mentre per gli aspetti relativi alla comunicazione e alla localizzazione del robot sono stati impiegati rispettivamente le librerie Wiiuse e del software originariamente sviluppato per il progetto LURCH [3].

3.1.2 RoboWII 2.0

RoboWII 2.0 [12] è stato sviluppato da Antonio Micali, con l'obiettivo di poter portare i concetti e le dinamiche di gioco del progetto RoboWII 1.0 in un gioco realizzato con un robot a basso costo, venduto tramite i consueti canali commerciali e quindi con una potenziale maggior facilità di diffusione. Il robot impiegato per la realizzazione di questo progetto è Spykee [7] — Figura 3.2

3.1.3 RoboWII 2.0.1

Il progetto RoboWII 2.0.1 [13] è gestito da Andrea Pontecorvo, questo progetto ha l'obiettivo di realizzare un nuovo gioco robotico tramite l'utilizzo

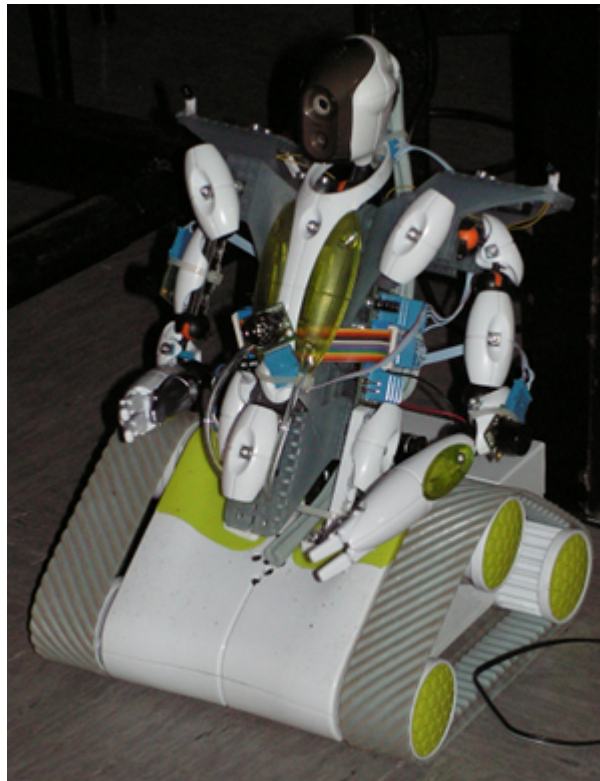


Figura 3.2: Spykee, utilizzato per il progetto RoboWII 2.0



Figura 3.3: LEGO NXT

del robot Spykee. Attualmente il lavoro è volto verso la realizzazione di un gioco appartenente alla tipologia *Hide-and-Seek*.

3.1.4 RoboWII 2.0.L

Il progetto RoboWII 2.0.L [11] è gestito da Gabriele Pallotta e da Luigi Parpinel. Il progetto si pone come obiettivo quello effettuare la valutazione dell'impiego dei prodotti *LEGO Mindstorms* [2] ed in particolare del *kit NXT* — Figura 3.3 — per la realizzazione di giochi robotici. Successivamente alla fase di valutazione, nel caso abbia esito positivo, verranno via via implementate le diverse funzionalità per la realizzazione di un gioco.

3.1.5 RoboWII 3.0 (Jedi Training)

Il progetto RoboWII 3.0 [9] è gestito da Adrien Servier e da Andrea Nicotra presso la sede di Como del Politecnico di Milano. Il progetto consiste nell'implementazione di un sistema che simuli l'addestramento Jedi visto nei film di *Guerre Stellari*, ovvero respingere con la propria spada laser il colpo diretto verso l'apprendista Jedi lanciato dal drone mobile. Il concetto del gioco è rappresentato in maniera schematica nella Figura 3.4.

3.2 Considerazioni

In questo capitolo è stata svolta una veloce panoramica dei lavori facenti parte del progetto RoboWII, è stato indicato come ognuno di essi si focalizzi su differenti aspetti delle problematiche di gioco, ovvero come si sia partiti

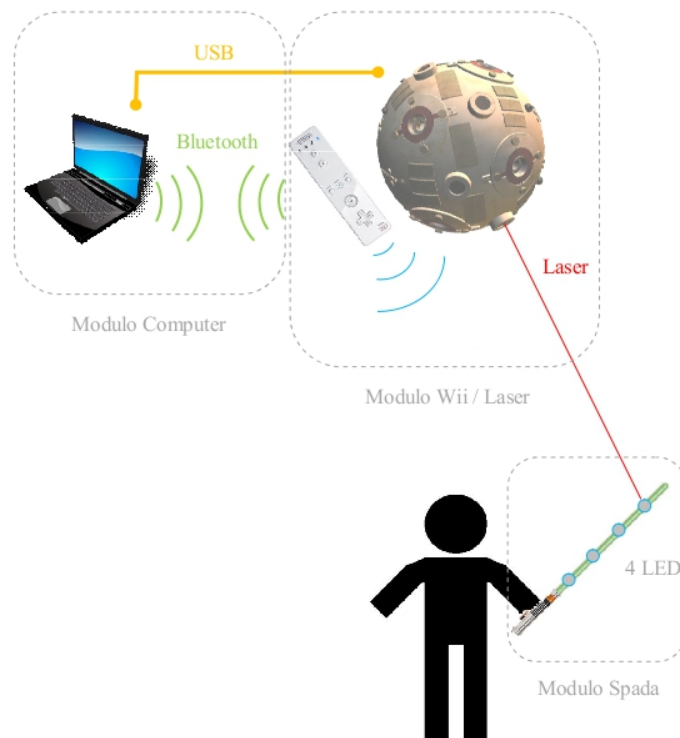


Figura 3.4: Schema gioco RoboWII 3.0

da un primo prototipo di gioco e successivamente si sia passati ad esplorare differenti alternative relative alla realizzazione dello stesso. Nel capitolo successivo e nei seguenti verrà introdotto il lavoro oggetto di questo documento, ovvero il progetto RoboWII 2.1.

Capitolo 4

Descrizione del gioco

In questo capitolo verrà descritto il gioco RoboWII 2.1 e le dinamiche collegate ad esso. Verranno descritti inizialmente gli obiettivi degli attori facenti parte del gioco, successivamente verranno date alcune informazioni di carattere generale sullo stesso ed infine verranno descritte le regole del gioco, sia quelle basilari che gli aspetti più avanzati delle stesse.

4.1 Obiettivi

Il gioco consiste in una caccia al robot, nella quale il giocatore ha l'obiettivo di impedire al robot di raggiungere la propria *casa base*.

Il giocatore, tramite la propria arma rappresentata dal WiiMote deve sparare e cercare di colpire il robot, fino a fargli perdere tutta la propria *energia*.

Il robot deve invece cercare di evitare i colpi del giocatore e raggiungere la *casa base* prima di essere messo fuori uso. Il robot oltre a fuggire dai colpi del giocatore è in grado di rispondere al fuoco del giocatore e può inoltre raggiungere particolari siti detti *waypoint* in grado di fornirgli degli aiuti.

4.2 Requisiti

Il gioco è adatto ad un target di utenza di età compresa tra gli 9 ed i 13 anni.

L'ambiente di gioco può consistere in una qualunque area — sia all'aperto che al chiuso — a disposizione del giocatore, tuttavia lo svolgimento del gioco è influenzato dalle possibilità di movimento del robot, quindi l'ambiente non deve avere superfici sulle quali il robot non può muoversi con agio o sulle quali il movimento lo danneggerebbe, quali ad esempio superfici erbose o

in asfalto. Trattandosi di un gioco nel quale occorre sparare ed evitare di essere colpiti, utilizzare un ambiente che offre diversi ripari ed ostacoli aiuta a rendere lo stesso più coinvolgente.

4.3 Regole di base

Sia il robot sia il giocatore umano sono soggetti a delle regole durante lo svolgimento del gioco, vengono di seguito elencate le regole che descrivono il corretto svolgimento del gioco:

- Il robot per raggiungere la *casa base* può muoversi nell'ambiente di gioco, ma deve evitare gli ostacoli presenti;
- Il robot non conosce a priori la posizione della *casa base* e dei *waypoint* presenti nell'area di gioco ma deve ricercarli, esplorando quindi l'area di gioco;
- Il robot è in grado di sapere quando è puntato dal WiiMote e la direzione in cui essa si trova;
- Il robot può rispondere al fuoco sparando a sua volta verso il giocatore;
- Il robot possiede un proprio valore di *energia*, esaurito il quale esso si considera fuori uso e quindi sconfitto;
- Il robot possiede un proprio valore di *stamina* che utilizza nello svolgimento delle proprie azioni, ognuna delle quali è associata ad un certo consumo di stamina;
- Il giocatore tramite l'utilizzo del WiiMote può mirare e sparare al robot;
- Il WiiMote non è in grado di sparare appena il robot viene inquadrato, ma necessita di una fase di *targeting* del robot che consiste nel mantenerlo puntato per un certo ammontare di secondi prima di poterlo colpire — *tempo di targeting*;
- Il giocatore dispone di tre differenti armi per affrontare il robot, ognuna con modalità di utilizzo ed effetti differenti;
- Gli effetti di un colpo andato a segno sono dipendenti dal tempo di puntamento e dalla tipologia di arma utilizzata;

- Il giocatore non può mirare al robot da una distanza inferiore alla *distanza di targeting*, altrimenti il robot innalza il suo scudo di energia e la funzionalità di sparo del giocatore verrà inibita fino alla fine della sua permanenza nell'area proibita;
- Il contatto fisico tra il giocatore umano ed il robot non è permesso.

4.4 Regole avanzate

In questa sezione verranno trattate le regole di gioco avanzate, tramite le quali verranno approfonditi in dettaglio alcuni degli aspetti discussi nelle sezioni precedenti.

4.4.1 Movimento del robot

Il robot può muoversi a differenti velocità, nel tentativo di fuggire dal giocatore e di raggiungere il proprio obiettivo.

Il movimento — a qualunque velocità esso avvenga — richiede una certa quantità di *stamina*, maggiore è la velocità maggiore è la quantità di *stamina* utilizzata. La stamina viene rigenerata automaticamente con il passare del tempo, pertanto il robot deve ottimizzare le proprie mosse in maniera da non rimanere mai con un basso livello di *stamina* e quindi con ridotte possibilità di azione.

4.4.2 Waypoint

Il robot dispone di alcuni traguardi intermedi — *waypoint* — che una volta raggiunti aumentano la stamina oppure l'energia del robot di una quota prefissata, questo valore è impostato in base alla difficoltà di raggiungere il particolare *waypoint* e può coincidere con il ripristino integrale dei valori di stamina e di energia.

4.4.3 Effetto dei colpi del robot

Il robot è in grado di sparare al giocatore, anche nel caso del robot è richiesto un *tempo di targeting* prima di poter sparare, tuttavia esso è notevolmente ridotto rispetto a quello del giocatore. Nel caso il colpo vada a segno, il giocatore non potrà sparare per un intervallo di tempo determinato — *tempo di ripresa* —; durante questo intervallo il robot può cercare di raggiungere i suoi obiettivi senza essere disturbato.

Ogni colpo effettuato dal robot ne abbassa lievemente il livello di *stamina*, questa scelta è fatta in maniera tale da evitare che il robot continui a sparare ininterrottamente, rendendo troppo difficile al giocatore di prendere parte all'azione.

Il robot deve operare quindi continuamente delle scelte su come impiegare la propria *stamina*: per sparare verso il giocatore oppure per muoversi. Tale scelta è influenzata dalla situazione in cui il robot si trova: in situazioni critiche — ad esempio bassa *energia* e bassa *stamina* — esso sarà più propenso a sparare, in maniera tale da poter bloccare il giocatore e poter raggiungere ad esempio un *waypoint* dove poter ricaricare la propria *energia/stamina*; viceversa in situazioni più tranquille esso tenderà a sparare con meno frequenza e a dedicarsi maggiormente alla ricerca della casa base.

La quantità di *stamina* a disposizione del robot in un dato istante non è direttamente fornita la giocatore ma questo — come nella realtà, ad esempio in una partita di calcio o basket — dovrà intuire questo valore dal vigore delle reazioni dell'avversario e da questo dedurre l'effettivo *affaticamento* dovuto alla bassa *stamina*.

4.4.4 Effetto dei colpi del giocatore

Sul robot sono presenti alcuni led che mostrano al giocatore lo stato corrente del puntamento della propria arma. In base a quanti led sono accesi il colpo, una volta andato a segno avrà un differente effetto, come indicato nella Tabella 4.1.

Led accesi	Effetto
1 led	Piccolo abbassamento della <i>stamina</i> del robot.
2 led	Discreto abbassamento della <i>stamina</i> del robot.
3 led	Grande abbassamento della <i>stamina</i> del robot.
4 led	Abbassamento di un livello di <i>energia</i> del robot.

Tabella 4.1: Effetto dei colpi del giocatore

Oltre ai led indicanti lo stato del puntamento, sul robot sono presenti ulteriori 4 led i quali indicano lo stato dell'*energia* del robot. Nel caso il robot venga colpito nella situazione in cui sono accesi i 4 led di puntamento, la sua *energia* si abbasserà di un livello, l'effetto del colpo andato a segno viene segnalato al giocatore dall'accensione di un nuovo led relativo all'*energia* del robot.

4.4.5 Armi disponibili

Il giocatore ha la possibilità di scegliere differenti armi, esse si differenziano in base al danno causato, alla *politica per la ricarica* ed altri aspetti che saranno descritti nel seguito del documento.

Il danno inflitto da un'arma è indicato tramite il valore *Danno* relativa all'arma: esso funge da moltiplicatore per i valori riportati nella Tabella 4.1. Ad esempio un'arma che infligge un danno $2x$, raddoppierà il danno base indicato nella Tabella 4.1.

La politica di ricarica indica le modalità con le quali è necessario provvedere ad eseguire la ricarica dell'arma.

La ricarica consiste in un particolare movimento del WiiMote, una volta effettuato il quale, l'arma viene completamente ricaricata. E' possibile effettuare la ricarica in qualsiasi momento anche se si hanno ancora a disposizione dei colpi nel caricatore.

La scelta di quale arma utilizzare è basata su dei movimenti appositi del WiiMote descritti nel Capitolo 5.

Le armi previste sono le seguenti:

Pistola

La pistola è l'arma standard del giocatore.

- **Danno:** $1x$;
- **Politica di ricarica:** Deve essere ricaricata una volta ogni 4 colpi;

Fucile

Il fucile è la seconda arma del giocatore, essa infligge danni più consistenti rispetto alla pistola ma possiede delle maggiori limitazioni.

- **Danno:** $2x$;
- **Politica di ricarica:** Deve essere ricaricata una volta ogni 2 colpi;

Spada

La spada è l'unica arma bianca a disposizione del giocatore, essa non causa danno ma porta la stamina del robot ad un livello minimo nel caso il colpo vada a segno. Il colpo della spada si differenzia rispetto a quello realizzato dalla pistola o dal fucile in quanto non è necessario eseguire la fase di targeting

del robot, ma il colpo si intende andato a segno se la lama della spada si trova a passare sulla zona del robot provvista del relativo sensore.

- **Danno:** Stamina portata a livello minimo;
- **Politica di ricarica:** N/A;

Capitolo 5

Interfaccia utente

In questo capitolo verrà definita l'interfaccia che il gioco pone verso il giocatore umano, con particolare riferimento ai comandi e alle funzioni presenti sul WiiMote.

5.1 Funzionalità

Il controller WiiMote, rappresentato nella Figura 5.1 dispone di diversi tasti posizionati sulla parte superiore ed inferiore dello stesso che possono essere utilizzati da parte dell'utente per svolgere le attività di gioco. In aggiunta a ciò il WiiMote dispone di alcuni accelerometri posizionati al proprio interno che possono essere utilizzati per identificare la tipologia di movimento che il giocatore sta compiendo in un determinato istante. Infine esso dispone di una telecamera a raggi infrarossi posizionata nella parte superiore del dispositivo in grado di rilevare la presenza e la posizione di quattro punti, fonti di emissione infrarossa.

La creazione dell'interfaccia per il gioco RoboWII 2.1 si basa su queste caratteristiche del controller WiiMote per fornire le funzionalità necessarie al giocatore umano.

L'idea alla base della definizione dell'interfaccia, dato il buon numero di combinazioni di tasti e movimenti possibili del controller, consiste nell'introdurre dove possibile una leggera ridondanza nei comandi, ad esempio facendo corrispondere una data funzionalità sia ad un movimento eseguito dal giocatore, sia ad un tasto scelto appositamente. In questa maniera il giocatore è libero di scegliere il metodo a lui più comodo per svolgere le azioni del gioco.

Nella sezione seguente verranno definite nello specifico le associazioni tra



Figura 5.1: Controller WiiMote

le funzionalità proprie del gioco e come queste verranno richiamate tramite l'interfaccia utente.

5.1.1 Associazione funzionalità - interfaccia

Le azioni di base che il giocatore può eseguire consistono nelle seguenti:

- Avviare il gioco;
- Mirare verso il robot;
- Sparare verso il robot;
- Ricaricare la propria arma;
- Riporre l'arma corrente;
- Estrarre un'altra arma;
- Riavviare il gioco;
- Terminare il gioco.

Ad ognuna di queste azioni è stato associato un comando per essere richiamata tramite l'uso del WiiMote, in alcuni casi mediante il movimento dello stesso, in altri mediante i tasti ed in altri ancora tramite entrambi i metodi. La posizione dei differenti tasti sul WiiMote è indicata nella Figura 5.2, di seguito vengono invece descritti in modo più approfondito le azioni ed i relativi metodi per essere richiamate:



Figura 5.2: Posizione dei tasti sul WiiMote

- **Avviare il gioco**
 - Movimento WiiMote: *N/A*;
 - Tasto associato: Tasto 1;
- **Mirare verso il robot**
 - Movimento WiiMote: Puntamento del controller in direzione del robot;
 - Tasto associato: *N/A*;
- **Sparare verso il robot**
 - Movimento WiiMote: *N/A*;
 - Tasto associato: Tasto *B* — *grilletto*;
- **Ricaricare la propria arma**
 - Movimento WiiMote: Movimento veloce prima verso il *basso* e successivamente verso l'*alto*;
 - Tasto associato: Tasto *A*;
- **Riporre l'arma corrente - estrarre un'altra arma**

Arma	Movimento per riporre	Movimento di estrazione
Pistola	Inserimento fondina sulla destra	Estrazione fondina sulla destra
Fucile	Inserimento fondina sulla schiena	Estrazione fondina sulla schiena
Spada	Inserimento fondina sulla sinistra	Estrazione fondina sinistra

Tabella 5.1: Movimenti per riporre l'arma corrente ed estrarre una nuova arma

- Movimento WiiMote: Si assume che il giocatore abbia tre diverse guaine posizionate su se stesso: due sui lati del bacino ed una posizionata sulla propria schiena; mediante dei movimenti relativi a queste posizioni il giocatore può cambiare la propria arma corrente. Nella Tabella 5.1 sono descritti i movimenti definiti;
- Tasto associato: I tasti *sinistro* e *destro* della *croce direzionale*;

- **Riavviare il gioco**

- Movimento WiiMote: N/A;
- Tasto associato: Pressione del tasto *2* — utilizzando l'impugnatura tipica per questo gioco risulta poco accessibile e di conseguenza difficilmente attivabile per errore;

- **Terminare il gioco**

- Movimento WiiMote: N/A;
- Tasto associato: Pressione del tasto *home*;

5.2 Altre interazioni

Il WiiMote, oltre che essere utilizzato per le interazioni descritte precedentemente, dispone di un motorino vibrante e di quattro led azzurri. Questi dispositivi possono essere utilizzati per fornire delle informazioni al giocatore e per rendere l'esperienza di gioco maggiormente immersiva.

Quando il giocatore esegue il cambio di arma, il WiiMote esegue una breve vibrazione, in modo da confermare al giocatore la corretta esecuzione del movimento per il cambio dell'arma; senza questa segnalazione il giocatore non sarebbe a conoscenza dell'arma che sta utilizzando correntemente se non provando ad utilizzarla.

I led posizionati sul WiiMote vengono invece impiegati per indicare al giocatore il numero di colpi disponibili attualmente per l'arma impiegata, ogni colpo eseguito tramite le armi da fuoco, che vada a segno oppure manchi il bersaglio spegne uno dei led. Quando tutti i led sono spenti l'arma corrente non dispone più di munizioni e deve essere ricaricata. Una volta eseguita l'operazione di ricarica, il nuovo numero di munizioni disponibili apparirà tramite i led del WiiMote in maniera da evidenziare al giocatore la corretta esecuzione della ricarica.

Oltre ai dispositivi presenti sul WiiMote il robot è dotato a sua volta di alcuni led luminosi, in particolare:

- una fila di quattro led gialli;
- una file di quattro led rossi;
- un led singolo verde;
- un led singolo bianco.

Questi led sono impiegati per fornire al giocatore informazioni utili relative all'andamento corrente della partita; inoltre il robot dispone di una piccola cassa audio capace di riprodurre suoni con una discreta qualità.

In particolare la fila di led gialli viene impiegata per indicare lo stato del *tempo di targeting*, in modo che il giocatore osservandoli possa capire quale effetto avrebbe il colpo nel caso fosse eseguito in quel momento, secondo gli effetti indicati nella Tabella 4.1. Il posizionamento di questi led sul robot inoltre permette al giocatore di non distogliere lo sguardo dal robot mentre lo tiene sotto mira.

La fila di led rossi indica invece la quantità di energia disponibile del robot, essa indica che il robot è in piena salute quando tutti i led sono spenti, mentre via via che esso subisce colpi i led si accendono. Quando i led rossi accesi arrivano a quattro, il robot ha terminato la propria energia ed il giocatore ha vinto la partita.

Il led verde indica al giocatore che si trova troppo vicino al robot e che questo ha attivato il proprio scudo, disattivando il puntamento dell'arma da parte del giocatore. Il robot può muoversi mentre il proprio led verde è acceso, ed essendo immune al puntamento da parte del giocatore può dirigersi senza problemi verso la casa base oppure puntare il giocatore senza

che questo posso fare nulla per difendersi — se non allontanarsi —, è quindi nell'interesse del giocatore rimanere ad una certa distanza dal robot in modo da non incorrere in questa situazione di svantaggio.

Infine l'accensione del led bianco indica al giocatore che il robot lo sta inquadrando e si sta preparando per eseguire il proprio colpo, il giocatore vedendo questa condizione deve togliersi il prima possibile da di fronte al robot, altrimenti entro pochi istanti questo avrà terminato la propria fase di targeting e colpirà il giocatore.

Il colpo andato a segno da parte del robot viene notificato al giocatore tramite un suono apposito riprodotto dalla cassa audio del robot.

Capitolo 6

Specifiche tecniche e studio di fattibilità

In questo capitolo verranno descritti gli aspetti più tecnici relativi alla realizzazione del gioco. Nella sezione seguente verrà introdotto la macchina a stati del gioco, in modo da definire in modo formale lo svolgimento dello stesso in tutte le sue fasi. Verrà successivamente descritto l'hardware occorrente per poter realizzare il gioco ed infine verrà presentato lo studio di fattibilità.

6.1 Macchina a stati

In questa sezione verrà analizzata la macchina a stati del gioco. Per rendere di più semplice lettura il documento, la macchina a stati generale del gioco è stata suddivisa in cinque diverse sotto-macchine, ognuna relativa ad un differente aspetto:

- Armi del giocatore;
- Puntamento delle armi del giocatore;
- Energia del robot;
- Stamina del robot;
- Aggressività del robot;
- Vittoria del robot.

La macchina a stati totale del gioco risulterà quindi definita dall'unione delle cinque sotto-macchine definite.

6.1.1 Armi del giocatore

La macchina a stati relativa alle armi del giocatore definisce come quest'ultimo può passare tra le differenti armi a sua disposizione, tenendo in considerazione i colpi disponibili per ognuna delle armi e considerando gli eventuali colpi ricevuti dal robot. Il diagramma della macchina a stati è rappresentato nella Figura 6.1.

Le transizioni della macchina a stati sono definite dai seguenti segnali:

RIPONI_PISTOLA Viene generato dall'atto di riporre la pistola nella fondina, viene generato all'esecuzione del relativo movimento, definito nel Capitolo 5;

RIPONI_FUCILE Viene generato dall'atto di riporre il fucile nella fondina, viene generato all'esecuzione del relativo movimento, definito nel Capitolo 5;

RIPONI_SPADA Viene generato dall'atto di riporre la spada nella guaina, viene generato all'esecuzione del relativo movimento, definito nel Capitolo 5;

ESTRAI_PISTOLA Viene generato dall'atto di estrarre la pistola dalla fondina, viene generato all'esecuzione del relativo movimento, definito nel Capitolo 5;

ESTRAI_FUCILE Viene generato dall'atto di estrarre il fucile dalla fondina, viene generato all'esecuzione del relativo movimento, definito nel Capitolo 5;

ESTRAI_SPADA Viene generato dall'atto di estrarre la spada dalla guaina, viene generato all'esecuzione del relativo movimento, definito nel Capitolo 5;

RICARICA Viene generato dall'atto di ricaricare la propria arma, il movimento è definito nel Capitolo 5;

SPARA Viene generato dall'atto di sparare verso il robot, indipendentemente dal fatto che sia sotto puntamento o meno, il movimento è definito nel Capitolo 5;

COLPITO_DAL_ROBOT Viene generato nell'istante in cui il giocatore è stato colpito dal robot;

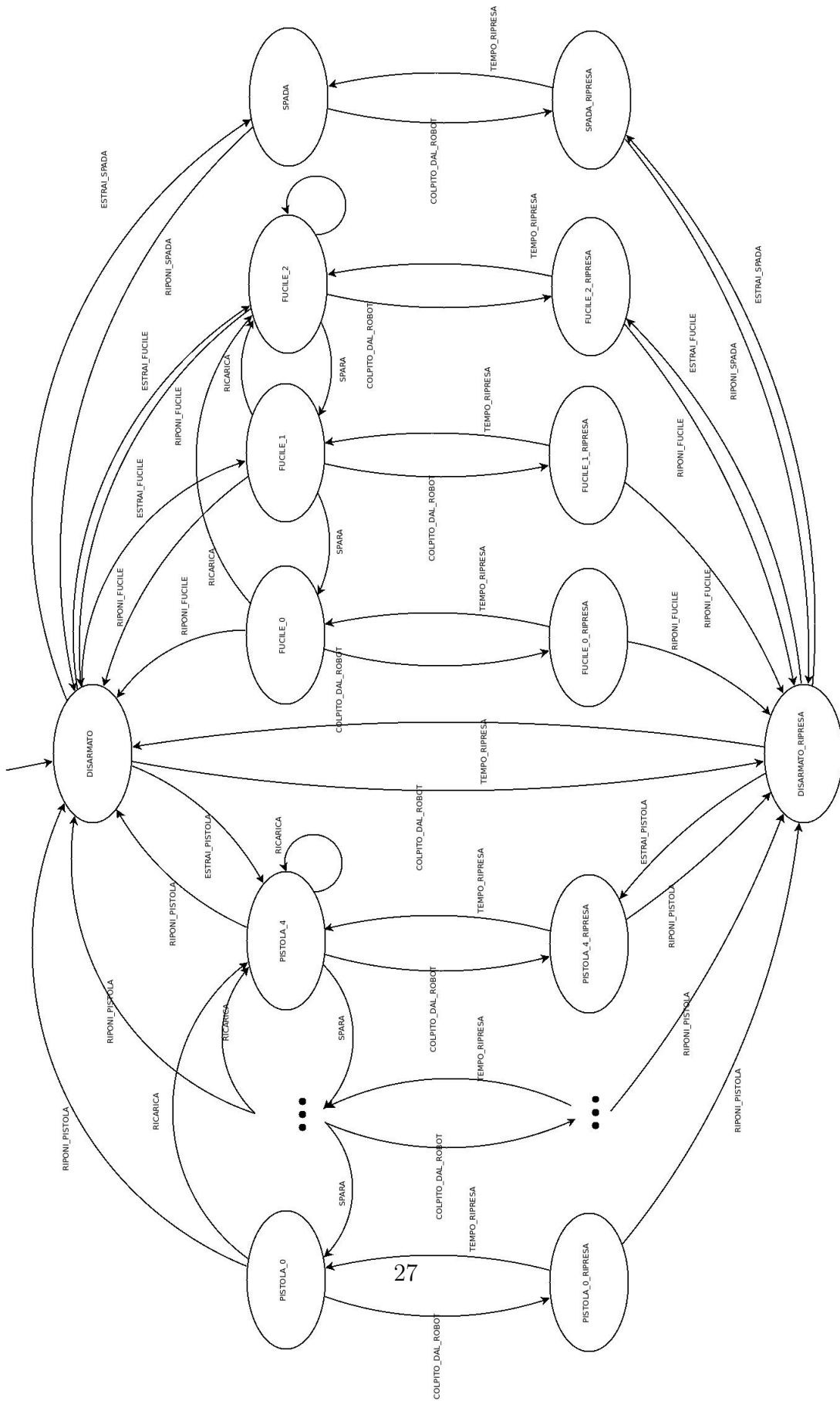


Figura 6.1: Macchina a stati armi

TEMPO_RIPRESA Viene generato dal *sistema* nel caso il giocatore sia stato colpito dal robot, una volta trascorso il *tempo di ripresa*.

Gli stati sono invece definiti come segue:

- **DISARMATO**

- Stato iniziale.
- Descrizione: In questo stato il giocatore non sta imbracciando alcuna arma.
- Transizioni:
 - * *ESTRAIPISTOLA* → PISTOLA_4;
 - * *ESTRAIFUCILE* → FUCILE_2;
 - * *ESTRAISPADA* → SPADA;
 - * *COLPITO_DAL_ROBOT* → DISARMATO_RIPRESA.

- **PISTOLA_4**

- Descrizione: In questo stato il giocatore sta imbracciando la pistola e ha a disposizione quattro colpi nel caricatore.
- Transizioni:
 - * *SPARA* → PISTOLA_3;
 - * *RICARICA* → PISTOLA_4;
 - * *RIPONIPISTOLA* → DISARMATO;
 - * *COLPITO_DAL_ROBOT* → PISTOLA_4_RIPRESA.

- **PISTOLA_3**

- Descrizione: In questo stato il giocatore sta imbracciando la pistola e ha a disposizione tre colpi nel caricatore.
- Transizioni:
 - * *SPARA* → PISTOLA_2;
 - * *RICARICA* → PISTOLA_4;
 - * *RIPONIPISTOLA* → DISARMATO;
 - * *COLPITO_DAL_ROBOT* → PISTOLA_3_RIPRESA.

- **PISTOLA_2**

- Descrizione: In questo stato il giocatore sta imbracciando la pistola e ha a disposizione due colpi nel caricatore.

- Transizioni:
 - * *SPARA* → PISTOLA_1;
 - * *RICARICA* → PISTOLA_4;
 - * *RIPONI_PISTOLA* → DISARMATO;
 - * *COLPITO_DAL_ROBOT* → PISTOLA_2_RIPRESA.

- **PISTOLA_1**

- Descrizione: In questo stato il giocatore sta imbracciando la pistola e ha a disposizione un colpo nel caricatore.
- Transizioni:
 - * *SPARA* → PISTOLA_0;
 - * *RICARICA* → PISTOLA_4;
 - * *RIPONI_PISTOLA* → DISARMATO;
 - * *COLPITO_DAL_ROBOT* → PISTOLA_1_RIPRESA.

- **PISTOLA_0**

- Descrizione: In questo stato il giocatore sta imbracciando la pistola ma non ha colpi disponibili nel caricatore.
- Transizioni:
 - * *RICARICA* → PISTOLA_4;
 - * *RIPONI_PISTOLA* → DISARMATO;
 - * *COLPITO_DAL_ROBOT* → PISTOLA_0_RIPRESA.

- **FUCILE_2**

- Descrizione: In questo stato il giocatore sta imbracciando il fucile ed ha a disposizione due colpi nel caricatore.
- Transizioni:
 - * *SPARA* → FUCILE_1;
 - * *RICARICA* → FUCILE_2;
 - * *RIPONI_FUCILE* → DISARMATO;
 - * *COLPITO_DAL_ROBOT* → FUCILE_2_RIPRESA.

- **FUCILE_1**

- Descrizione: In questo stato il giocatore sta imbracciando il fucile ed ha a disposizione un colpo nel caricatore.

- Transizioni:
 - * *SPARA* → FUCILE_0;
 - * *RICARICA* → FUCILE_2;
 - * *RIPONIFUCILE* → DISARMATO;
 - * *COLPITO_DAL_ROBOT* → FUCILE_1_RIPRESA.

- **FUCILE_0**

- Descrizione: In questo stato il giocatore sta imbracciando il fucile ma non ha colpi disponibili nel caricatore.
- Transizioni:
 - * *RICARICA* → FUCILE_2;
 - * *RIPONIFUCILE* → DISARMATO;
 - * *COLPITO_DAL_ROBOT* → FUCILE_0_RIPRESA.

- **SPADA**

- Descrizione: In questo stato il giocatore sta imbracciando la spada.
- Transizioni:
 - * *RIPONISPADA* → DISARMATO;
 - * *COLPITO_DAL_ROBOT* → SPADA_RIPRESA.

- **DISARMATO_RIPRESA**

- Descrizione: In questo stato il giocatore non sta imbracciando alcuna arma ma anche imbracciandone una avrebbe sia il puntamento sia la possibilità di fare fuoco sono disattivati.
- Transizioni:
 - * *ESTRAIPISTOLA* → PISTOLA_4_RIPRESA;
 - * *ESTRAIFUCILE* → FUCILE_2_RIPRESA;
 - * *ESTRAISPADA* → SPADA_RIPRESA;
 - * *TEMPO_RIPRESA* → DISARMATO.

- **PISTOLA_4_RIPRESA**

- Descrizione: In questo stato il giocatore sta imbracciando la pistola con quattro colpi a disposizione nel caricatore, ma sia il puntamento sia la possibilità di fare fuoco sono disattivati.
- Transizioni:

- * *TEMPO_RIPRESA* → PISTOLA_4;
- * *RIPONI_PISTOLA* → DISARMATO_RIPRESA.

- **PISTOLA_3_RIPRESA**

- Descrizione: In questo stato il giocatore sta imbracciando la pistola con tre colpi a disposizione nel caricatore, ma sia il puntamento sia la possibilità di fare fuoco sono disattivati.
- Transizioni:
 - * *TEMPO_RIPRESA* → PISTOLA_3;
 - * *RIPONI_PISTOLA* → DISARMATO_RIPRESA.

- **PISTOLA_2_RIPRESA**

- Descrizione: In questo stato il giocatore sta imbracciando la pistola con due colpi a disposizione nel caricatore, ma sia il puntamento sia la possibilità di fare fuoco sono disattivati.
- Transizioni:
 - * *TEMPO_RIPRESA* → PISTOLA_2;
 - * *RIPONI_PISTOLA* → DISARMATO_RIPRESA.

- **PISTOLA_1_RIPRESA**

- Descrizione: In questo stato il giocatore sta imbracciando la pistola con un colpo a disposizione nel caricatore, ma sia il puntamento sia la possibilità di fare fuoco sono disattivati.
- Transizioni:
 - * *TEMPO_RIPRESA* → PISTOLA_1;
 - * *RIPONI_PISTOLA* → DISARMATO_RIPRESA.

- **PISTOLA_0_RIPRESA**

- Descrizione: In questo stato il giocatore sta imbracciando la pistola senza alcun colpo a disposizione nel caricatore, ma in ogni caso sia il puntamento sia la possibilità di fare fuoco sono disattivati.
- Transizioni:
 - * *TEMPO_RIPRESA* → PISTOLA_0;
 - * *RIPONI_PISTOLA* → DISARMATO_RIPRESA.

- **FUCILE_2_RIPRESA**

- Descrizione: In questo stato il giocatore sta imbracciando il fucile con due colpi a disposizione nel caricatore, ma sia il puntamento sia la possibilità di fare fuoco sono disattivati.
- Transizioni:
 - * *TEMPO_RIPRESA* → FUCILE_2;
 - * *RIPONI_FUCILE* → DISARMATO_RIPRESA.

● FUCILE_1_RIPRESA

- Descrizione: In questo stato il giocatore sta imbracciando il fucile con un colpo a disposizione nel caricatore, ma sia il puntamento sia la possibilità di fare fuoco sono disattivati.
- Transizioni:
 - * *TEMPO_RIPRESA* → FUCILE_1;
 - * *RIPONI_FUCILE* → DISARMATO_RIPRESA.

● FUCILE_0_RIPRESA

- Descrizione: In questo stato il giocatore sta imbracciando il fucile senza alcun colpo a disposizione nel caricatore, ma in ogni caso sia il puntamento sia la possibilità di fare fuoco sono disattivati.
- Transizioni:
 - * *TEMPO_RIPRESA* → FUCILE_0;
 - * *RIPONI_FUCILE* → DISARMATO_RIPRESA.

● SPADA_RIPRESA

- Descrizione: In questo stato il giocatore sta imbracciando la spada, ma la possibilità di colpire il robot è sono disattivata.
- Transizioni:
 - * *TEMPO_RIPRESA* → SPADA;
 - * *RIPONI_SPADA* → DISARMATO_RIPRESA.

6.1.2 Puntamento delle armi del giocatore

La macchina a stati relativa al puntamento delle armi del giocatore definisce gli stati di puntamento quando viene imbracciata la pistola oppure il fucile; essa definisce lo svolgimento del gioco e determina quale sarà l'effetto del

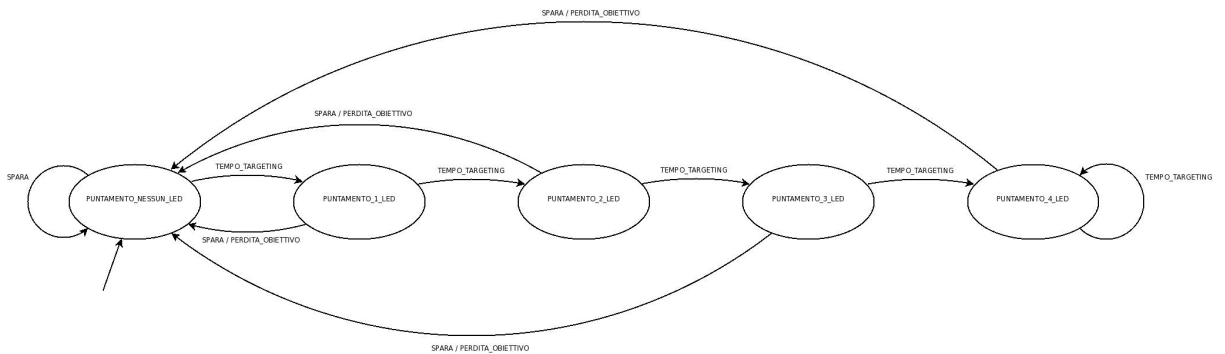


Figura 6.2: Macchina a stati puntamento armi del giocatore

colpo nel caso vada a segno. Il diagramma della macchina a stati è rappresentato nella Figura 6.2.

Le transizioni della macchina a stati sono definite dai seguenti segnali:

TEMPO_TARGETING Viene generato dal sistema nel caso il giocatore tenga puntato il robot per un tempo pari al *tempo di targeting*;

SPARA Viene generato all'atto di sparare verso il robot nelle modalità definite nel Capitolo 5;

PERDITA_OBIETTIVO Viene generato nel caso il giocatore perda il puntamento del robot.

Gli stati sono invece definiti come segue:

- **PUNTAMENTO_NESSUN_LED**

- Stato iniziale.
- Descrizione: In questo stato il giocatore non sta puntando il robot o lo sta puntando da un tempo inferiore al *tempo di targeting*; qualsiasi colpo dato in questo stato non colpisce il robot.
- Transizioni:
 - * *TEMPO_TARGETING* → PUNTAMENTO_1_LED;
 - * *SPARA* → PUNTAMENTO_NESSUN_LED;

- **PUNTAMENTO_1_LED**

- Descrizione: In questo stato il giocatore sta puntando il robot, il danno inferto da un eventuale colpo sarà pari a quello definito dall'arma corrente nel caso di un led acceso, come definito nel Capitolo 4.
- Transizioni:
 - * *TEMPO_TARGETING* → PUNTAMENTO_2_LED;
 - * *SPARA* → PUNTAMENTO_NESSUN_LED;
 - * *PERDITA_OBIETTIVO* → PUNTAMENTO_NESSUN_LED;

● PUNTAMENTO_2_LED

- Descrizione: In questo stato il giocatore sta puntando il robot, il danno inferto da un eventuale colpo sarà pari a quello definito dall'arma corrente nel caso di due led accesi, come definito nel Capitolo 4.
- Transizioni:
 - * *TEMPO_TARGETING* → PUNTAMENTO_3_LED;
 - * *SPARA* → PUNTAMENTO_NESSUN_LED;
 - * *PERDITA_OBIETTIVO* → PUNTAMENTO_NESSUN_LED;

● PUNTAMENTO_3_LED

- Descrizione: In questo stato il giocatore sta puntando il robot, il danno inferto da un eventuale colpo sarà pari a quello definito dall'arma corrente nel caso di tre led accesi, come definito nel Capitolo 4.
- Transizioni:
 - * *TEMPO_TARGETING* → PUNTAMENTO_4_LED;
 - * *SPARA* → PUNTAMENTO_NESSUN_LED;
 - * *PERDITA_OBIETTIVO* → PUNTAMENTO_NESSUN_LED;

● PUNTAMENTO_4_LED

- Descrizione: In questo stato il giocatore sta puntando il robot, il danno inferto da un eventuale colpo sarà pari a quello definito dall'arma corrente nel caso di quattro led accesi, come definito nel Capitolo 4.
- Transizioni:
 - * *TEMPO_TARGETING* → PUNTAMENTO_4_LED;
 - * *SPARA* → PUNTAMENTO_NESSUN_LED;
 - * *PERDITA_OBIETTIVO* → PUNTAMENTO_NESSUN_LED;

6.1.3 Energia del robot

La macchina a stati relativa all'energia del robot definisce la quantità di energia ancora a disposizione del robot e determina l'eventuale vittoria del giocatore nel caso questa arrivi a zero, oltre a questo contribuisce a definire il livello di aggressività del robot. Il diagramma della macchina a stati è rappresentato nella Figura 6.3.

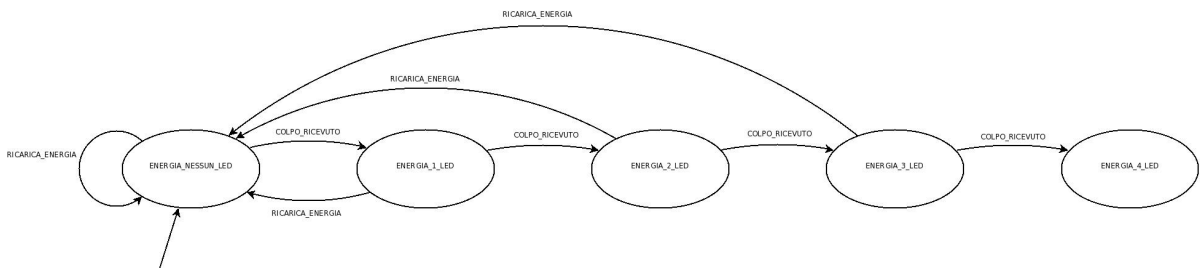


Figura 6.3: Macchina a stati energia del robot

Le transizioni della macchina a stati sono definite dai seguenti segnali:

COLPO_RICEVUTO Viene generato nel caso il giocatore spari verso il robot ad un livello di puntamento tale da causare un danno al livello di energia, secondo quanto definito nel Capitolo 4.

RICARICA_ENERGIA Viene generato nel caso il robot raggiunga un traguardo intermedio che ne ripristini il livello di energia.

Gli stati sono invece definiti come segue:

- **ENERGIA_NESSUN_LED**

- Stato iniziale.
- Descrizione: In questo stato robot è al livello massimo di energia.
- Transizioni:
 - * *COLPO_RICEVUTO* → ENERGIA_1_LED;
 - * *RICARICA_ENERGIA* → ENERGIA_NESSUN_LED.

- **ENERGIA_1_LED**

- Descrizione: In questo stato robot ha perso un quarto della propria energia.

- Transizioni:
 - * *COLPO_RICEVUTO* → ENERGIA_2_LED;
 - * *RICARICA_ENERGIA* → ENERGIA_NESSUN_LED.

- **ENERGIA_2_LED**

- Descrizione: In questo stato robot ha perso metà della propria energia.
- Transizioni:
 - * *COLPO_RICEVUTO* → ENERGIA_3_LED;
 - * *RICARICA_ENERGIA* → ENERGIA_NESSUN_LED.

- **ENERGIA_3_LED**

- Descrizione: In questo stato robot ha perso tre quarti della propria energia.
- Transizioni:
 - * *COLPO_RICEVUTO* → ENERGIA_4_LED;
 - * *RICARICA_ENERGIA* → ENERGIA_NESSUN_LED.

- **ENERGIA_4_LED**

- Descrizione: In questo stato robot ha perso tutta la propria energia; vittoria del giocatore.
- Transizioni: Nessuna.

6.1.4 Stamina del robot

La macchina a stati relativa alla stamina del robot tiene traccia di questo valore, e ne sottrae una determinata quota a seconda delle azioni eseguite dal robot. E' stato scelto di utilizzare una scala di cento valori, ad ogni azione eseguita dal robot il *sistema genererà* un determinato numero di richieste unitarie che porteranno il robot verso il nuovo valore di stamina. Il valore corrente della stamina contribuisce a definire il livello di aggressività del robot. Il diagramma della macchina a stati è rappresentato nella Figura 6.4.

Le transizioni della macchina a stati sono definite dai seguenti segnali:

UTILIZZO_STAMINA Viene generata dal sistema in base alle azioni eseguite dal robot, più l'azione è dispendiosa, più segnali di *UTILIZZO_STAMINA* verranno generati;

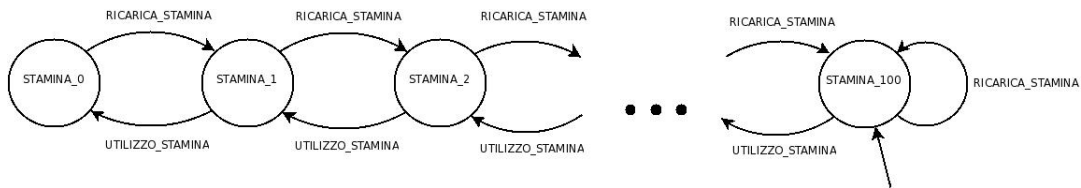


Figura 6.4: Macchina a stati stamina del robot

RICARICA_STAMINA Viene generata nel caso il robot raggiunga un traguardo intermedio che ne ripristini — eventualmente anche solo in maniera parziale — il livello di stamina.

Gli stati sono invece definiti come segue:

- **STAMINA_100**

- Stato iniziale.
- Descrizione: In questo stato robot è al livello massimo — 100 — di stamina.
- Transizioni:
 - * *UTILIZZO_STAMINA* → STAMINA_99;
 - * *RICARICA_STAMINA* → STAMINA_100.

- **STAMINA_1**

- Descrizione: In questo stato robot è al livello 1 di stamina.
- Transizioni:
 - * *UTILIZZO_STAMINA* → STAMINA_0;
 - * *RICARICA_STAMINA* → STAMINA_2.

- **STAMINA_0**

- Descrizione: In questo stato robot è al livello 0 di stamina.
- Transizioni:
 - * *RICARICA_STAMINA* → STAMINA_1.

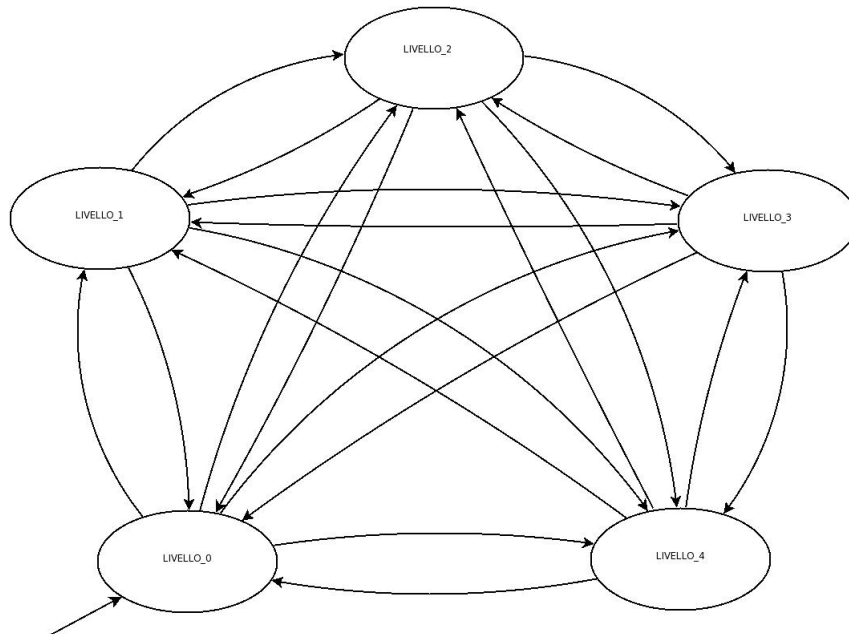


Figura 6.5: Macchina a stati aggressività del robot

6.1.5 Aggressività del robot

La macchina a stati relativa all'aggressività del robot definisce il comportamento del robot in base alle proprie condizioni. Il diagramma della macchina a stati è rappresentato nella Figura 6.5.

Le transizioni della macchina a stati sono dettate dall'utilizzo di una funzione — eventualmente in logica fuzzy — che tiene in considerazione parametri quali:

- Energia del robot;
- Stamina del robot;
- Prossimità di un eventuale *waypoint*;
- Prossimità della *casa base*.

Gli stati sono invece definiti come segue:

- **LIVELLO_0**
 - Stato iniziale.

- Descrizione: In questo stato il robot ricerca semplicemente la *casa base*, senza curarsi della presenza del giocatore.

- **LIVELLO_1**

- Descrizione: In questo stato il robot ricerca la *casa base* oppure uno dei *waypoint*, senza curarsi della presenza del giocatore.

- **LIVELLO_2**

- Descrizione: In questo stato il robot ricerca la *casa base* oppure uno dei *waypoint*, ma nel caso individui il giocatore cerca di colpirlo.

- **LIVELLO_3**

- Descrizione: In questo stato il robot si pone come obiettivo primario quello di colpire il giocatore, e di ricercare la *casa base* oppure uno dei *waypoint* solamente come obiettivo secondario.

- **LIVELLO_4**

- Descrizione: In questo stato il robot cerca di colpire ad ogni costo il giocatore, non curandosi della prossimità di *waypoint* o della *casa base*.

6.1.6 Vittoria del robot

La macchina a stati relativa alla vittoria del robot definisce la condizione nella quale il robot vince la partita; essa inoltre definisce il riavvio della partita. La macchina a stati è rappresentata nella Figura 6.6.

La condizione della vittoria della partita da parte del giocatore è invece definita dalla macchina a stati relativa all'energia del robot, illustrata nella Sottosezione 6.1.3.

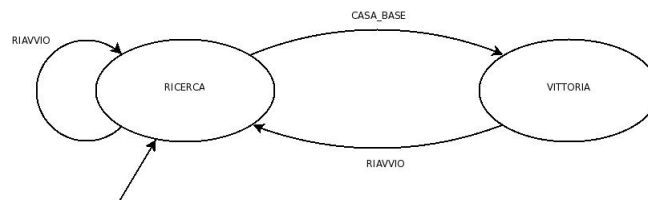


Figura 6.6: Macchina a stati vittoria del robot

Le transizioni della macchina a stati sono definite dai seguenti segnali:

CASE_BASE Viene generata dall'atto del raggiungimento della *casa base* da parte del robot;

RIAVVIA Viene generata dall'atto di riavvio della partita da parte del giocatore, come definito nel Capitolo 5.

Gli stati sono invece definiti come segue:

- **RICERCA**

- Stato iniziale.
- Descrizione: In questo stato il robot cerca di raggiungere i propri obiettivi, in modo da ottenere la vittoria della partita.
- Transizioni:
 - * *CASA_BASE* → VITTORIA;
 - * *RIAVVIO* → RICERCA.

- **VITTORIA**

- Descrizione: In questo stato il robot ha vinto la partita.
- Transizioni:
 - * *RIAVVIO* → RICERCA.

6.2 Hardware occorrente

In questa sezione verrà definito l'hardware occorrente per la realizzazione del gioco RoboWII 2.1. Il materiale occorrente consiste in:

- Robot autonomo mobile;
 - Telecamera;
 - Fila di quattro led rossi;
 - Fila di quattro led gialli;
 - Led singolo verde;
 - Led singolo bianco;
 - Dieci led infrarossi;
 - Piccola cassa audio;
 - Sensori di prossimità;

- Controller WiiMote;
- Giubbotto di materiale rifrangente per il giocatore;
- Waypoint;
 - Led piani da circa 90-100 lumens;
- Casa base;
 - Cilindro di colore facilmente identificabile;
- Unità di elaborazione esterna;
- Comunicazione;
 - Interfaccia *Bluetooth* WiiMote;
 - Interfaccia *Bluetooth* unità di elaborazione esterna;
 - Interfaccia *Wifi* robot;
 - Interfaccia *Wifi* unità di elaborazione esterna;
 - Interfaccia *Xbee* per la gestione dei sensori di prossimità e dei led;
 - Interfaccia *Xbee* unità di elaborazione esterna.

Nelle sezioni seguenti verranno descritti in dettaglio i componenti specificati ed i loro ruoli all'interno del gioco.

6.2.1 Robot autonomo mobile

Il robot è assieme al giocatore l'attore principale del gioco, il suo ruolo consiste nel competere con il giocatore per la vittoria del gioco. Esso è dotato di movimento in maniera da poter raggiungere i propri traguardi ed evitare i colpi del giocatore. Il movimento deve essere preferibilmente di tipo *olonomico*, in maniera da permettere al robot di spostarsi attorno al giocatore senza dover alternare rotazioni e traslazioni; questo permette al robot azioni quali ad esempio lo schivare i colpi del giocatore, mantenendolo sotto tiro — movimento di *strafe*. Per questo motivo uno dei robot indicati per l'implementazione del gioco è *Rovio*[6] — Figura 6.7.

Rinunciando alla possibilità di fare eseguire questo movimento al robot è possibile utilizzare robot quale ad esempio *Spykee* — Figura 6.8.

Il robot impiegato dovrà disporre di alcuni ulteriori componenti.



Figura 6.7: Rovio



Figura 6.8: Spykee

Dovrà essere presente una telecamera posizionata sul robot stesso per poter individuare i propri obiettivi ed il giocatore.

Il robot dovrà avere montata su di se due file di quattro led: una fila di led rossi per mostrare al giocatore lo stato corrente dell'energia del robot ed un'altra fila gialla per mostrare al giocatore lo stato corrente del puntamento.

Il robot dovrà essere munito di led infrarossi, in maniera da poter essere riconosciuto dalla telecamera del controller WiiMote.

Inoltre, dovrà essere munito di due led singoli, uno di colore verde, per indicare al giocatore il fatto di essere troppo vicino ed uno bianco, che starà ad indicare che in quel momento il robot sta eseguendo il puntamento del giocatore.

La cassa audio del robot sarà invece impiegata per notificare al giocatore che è stato colpito dal robot.

Infine, per poter riconoscere gli ostacoli il robot dovrà essere equipaggiato da una cintura di sensori di prossimità.

6.2.2 Controller WiiMote

Il controller WiiMote è lo strumento principale utilizzato dal giocatore per interagire con il robot. Tramite i sensori presenti all'interno di esso il giocatore potrà eseguire i movimenti ed i comandi descritti nel Capitolo 5, mentre con la telecamera presente potrà mantenere il robot sotto puntamento. Inoltre, i led presenti sul WiiMote indicheranno la quantità di colpi presenti nel caricatore dell'arma correntemente imbracciata dal giocatore.

6.2.3 Giubbotto di materiale rifrangente

Il giubbotto serve al robot per poter identificare il giocatore e per poterlo eventualmente colpire. Il giubbotto è realizzato con un materiale rifrangente, di facile individuazione da parte della telecamera del robot.

6.2.4 Identificatore waypoint e casa base

Per definire la posizione dei waypoint e della casa base e rendere questa informazione chiara sia per il giocatore che per il robot, è stato scelto di impiegare rispettivamente dei led luminosi piani ed un cilindro di colore facilmente riconoscibile dal robot.

I led piani devono avere una buona luminosità, circa di 90 lumens in maniera da essere facilmente riconosciuti da parte del robot, mentre per il cilindro utilizzato per la casa base può essere utilizzato semplicemente del

cartoncino colorato.

La scelta di utilizzare due soluzioni così differenti è stata dettata dalla necessità di esplorare differenti alternative per la soluzione del problema della individuazione dei punti importati per lo svolgimento del gioco da parte del robot; le considerazioni riguardo all'efficacia delle soluzioni scelte ed un commento ai risultati saranno trattati nella Sezione 6.3.

6.2.5 Unità di elaborazione esterna

Tutta l'elaborazione dei dati e della immagini, essendo particolarmente onerosa in termini di risorse, verrà gestita da un calcolatore esterno, che provvederà a ricevere i dati, ad elaborarli e a trasmettere i risultati alle interfacce che ne facciano richiesta. L'unità di elaborazione esterna svolgerà il ruolo di *sistema*.

6.2.6 Comunicazione

Il protocollo di comunicazione scelto è il Bluetooth per la comunicazione tra WiiMote e l'unità di elaborazione esterna, mentre la comunicazione tra robot e unità di elaborazione esterna è stato scelto di utilizzare Wifi per la maggiore affidabilità e velocità di trasmissione dati, soprattutto in relazione al trasferimento delle immagini per l'elaborazione.

Infine la comunicazione tra il sistema di controllo dei sensori di prossimità, dei led infrarossi, dei led colorati e l'unità esterna è gestita tramite Xbee. E' stato scelto di utilizzare questo tipo di comunicazione in quanto i dispositivi citati non sono presenti nel robot, ma sono stati e aggiunti e programmati manualmente; inoltre un'altro aspetto che ha fatto prediligere questo standard rispetto ad altri di tipo equivalente è stato il suo basso consumo energetico.

6.3 Studio di fattibilità

Per la realizzazione del gioco così come descritto occorre considerare una serie di differenti problematiche legate alla reale realizzabilità di ogni caratteristica trattata, in modo da stabilire se con le tecnologie disponibili sia possibile realizzare un prodotto così definito.

Viene di seguito indicata la lista degli aspetti che influiscono sulla realizzabilità del progetto:

- Riconoscimento movimenti di gioco;
- Riconoscimento del giocatore da parte del robot;
- Riconoscimento dei waypoint da parte del robot;
- Riconoscimento della casa base da parte del robot;
- Riconoscimento degli ostacoli da parte del robot;
- Puntamento del WiiMote verso il robot;
- Rilevamento della eccessiva vicinanza del giocatore al robot.

Ognuno di questi punti verrà trattato qui di seguito analizzando le conclusioni ed i risultati a cui si è giunti durante questo lavoro.

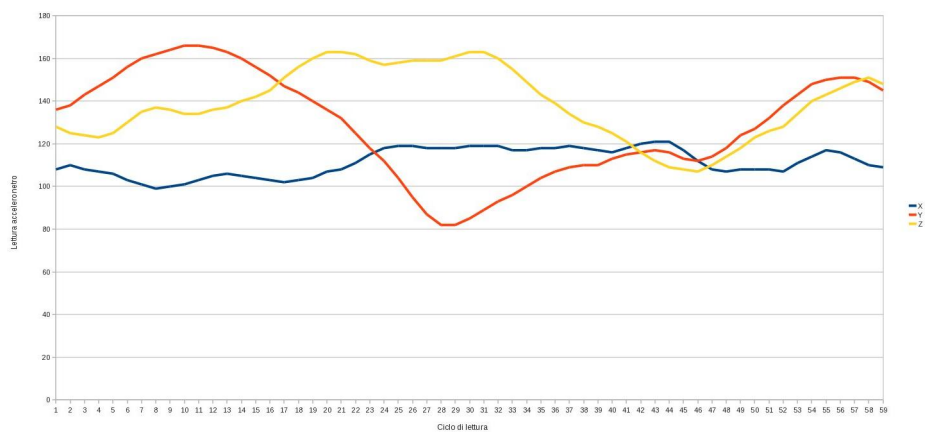
6.3.1 Riconoscimento movimenti del gioco

Il riconoscimento dei movimenti del gioco, quali cambio di arma e ricarica della stessa viene effettuato tramite l'elaborazione dei valori dagli accelerometri presenti all'interno del WiiMote. Questi segnali — uno per ogni asse — rappresentano quindi con una discreta precisione — data dai sensori — il movimento eseguito dal giocatore.

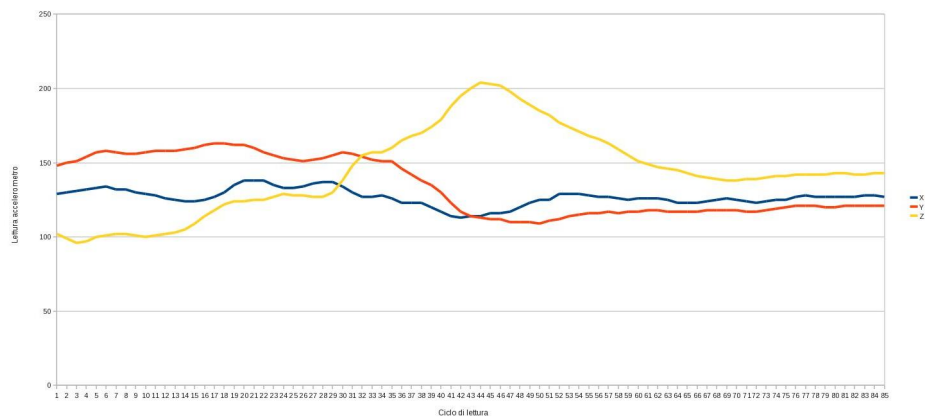
Come si vede nelle Figure 6.9 – 6.10 – 6.11, raffiguranti rispettivamente la lettura degli accelerometri sui tre assi per i movimenti relativi all'estrazione, al rifodero ed alla ricarica delle armi, le curve ottenute sono molto differenti le une dalle altre, è quindi possibile sfruttare questo aspetto per realizzare un algoritmo per il riconoscimento dei movimenti.

La soluzione suggerita, per alcuni aspetti simile a quella adottata per il rilevamento delle azioni del giocatore in differenti giochi commerciali per Wii, consiste nei seguenti passi:

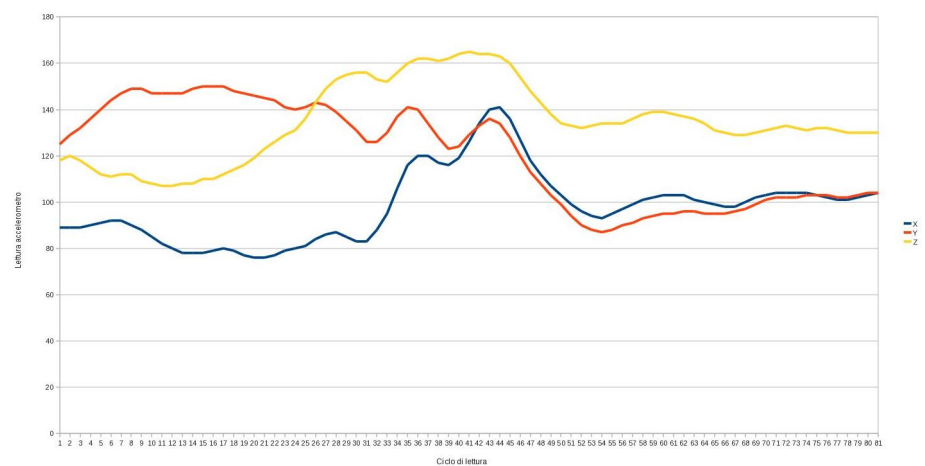
- Definire un set di azioni che il giocatore può eseguire, nel nostro caso quindi le estrazioni, rifoderi e ricarica delle armi;
- Associare ad ognuno di questi movimenti una curva di riferimento, ad esempio quelle indicate nelle Figure 6.9 – 6.10 – 6.11;
- Acquisire il movimento effettuato dal giocatore tramite la lettura dei valori degli accelerometri;



(a) Estrazione della pistola

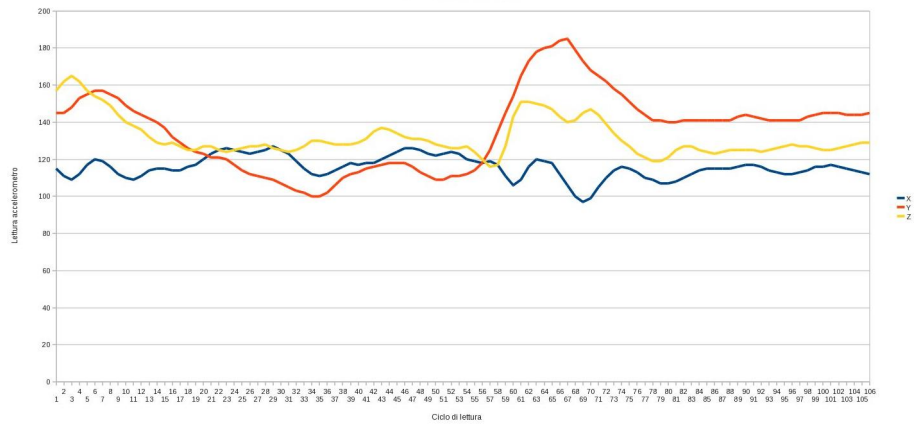


(b) Estrazione del fucile

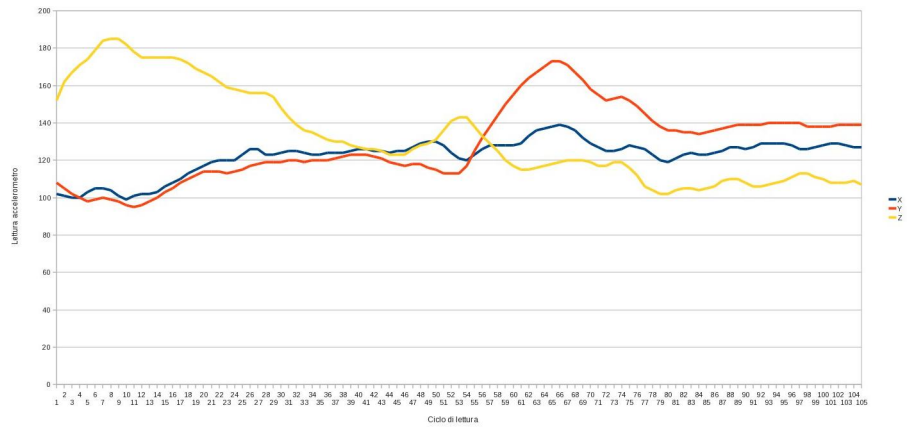


(c) Estrazione della spada

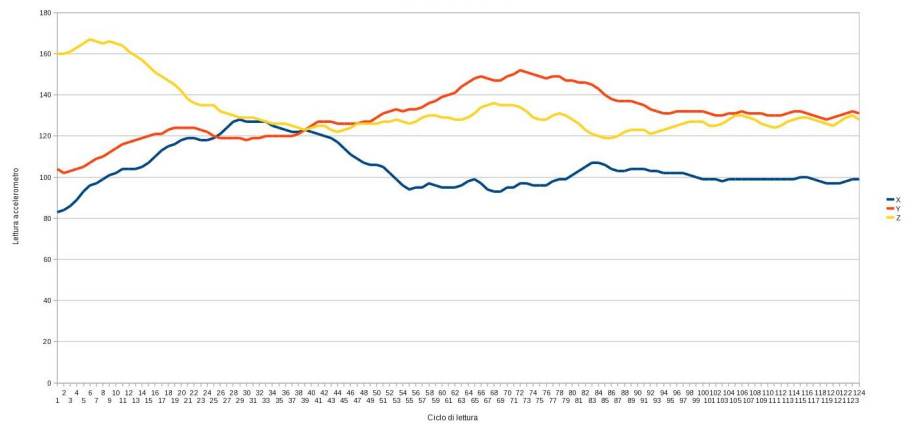
Figura 6.9: Estrazione delle armi



(a) Rifodero della pistola



(b) Rifodero del fucile



(c) Rifodero della spada

Figura 6.10: Rifodero delle armi



Figura 6.11: Ricarica dell'arma

- Eseguire una normalizzazione della curva letta dagli accelerometri, in modo da portare questa in una dimensione temporale paragonabile con le curve di riferimento;
- Scegliere tra le curve di riferimento quella più somigliante a quella letta (considerando misure quali ad esempio la deviazione standard per ogni asse);
- Se nel passo precedente è stata scelta una delle curve di riferimento, eseguire l'azione relativa ad essa, altrimenti nel caso non sia stata trovata nessuna curva somigliante, ignorare il movimento corrente e proseguire con l'analisi del successivo.

6.3.2 Riconoscimento del giocatore da parte del robot

Un altro aspetto da considerare per la realizzazione del gioco consiste nel trovare un modo per riconoscere il giocatore tramite la telecamera montata sul robot.

L'idea alla base del riconoscimento del giocatore consiste nel ricercare il particolare colore del giubbotto indossato dal giocatore. Inizialmente è stato considerato un filtraggio nello spazio di colore *RGB*, ma visto che la soluzione ottenuta era particolarmente sensibile alle variazioni di luminosità, è stato deciso di eseguire il riconoscimento nello spazio di colore *HSV*, maggiormente robusto a questo genere di variazioni.

E' stato quindi sviluppato durante il lavoro un algoritmo basato sulle librerie OpenCV [8] che a partire dall'immagine ottenuta dalla telecamera, ottenga una nuova immagine in bianco/nero dove i pixel bianchi adiacenti tra di loro indichino la posizione nella quale si trova il giocatore.

Un esempio dei risultati dell'algoritmo implementato è mostrato nella Figura 6.12.



Figura 6.12: Riconoscimento del giocatore da parte del robot

6.3.3 Riconoscimento dei waypoint da parte del robot

Il riconoscimento dei waypoint da parte del robot è un altro aspetto fondamentale del gioco. L'idea per la realizzazione è basata sull'utilizzo dei led piani, come quello rappresentato nella Figura 6.13, in grado di emettere una buona quantità di luce — quelli utilizzati nel lavoro variavano tra i 90 e i 150 lumens. E' stato calcolato che l'utilizzo di questo tipo di led, collegato a 4 batterie di tipo *cr2032* permette un utilizzo di oltre 35 ore di gioco continuativo.

Il filtraggio è stato eseguito nello spazio di colore YCbCr, filtrando in base alla componente *luma*, ricercando i punti più luminosi e considerando la posizione relativa del candidato waypoint nell'immagine ottenuta.

Anche in questo caso è stato sviluppato un algoritmo per questo compito ed i risultati sono quelli mostrati nella Figura 6.14.

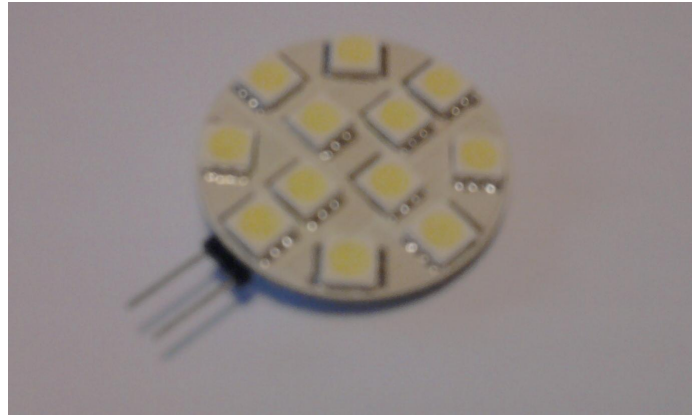


Figura 6.13: Led piano



Figura 6.14: Riconoscimento di uno dei waypoint da parte del robot

6.3.4 Riconoscimento della casa base da parte del robot

Per il riconoscimento della casa base è stato scelto un approccio differente, volto ad analizzare i risultati di una diversa scelta per l'identificazione dei punti dell'ambiente da parte del robot.

In particolare in questo caso si è provato ad utilizzare un cilindro di un particolare colore, realizzato tramite del cartone, e successivamente si sono visti i risultati ottenuti tramite il filtraggio nello spazio di colore *HSV*, tramite lo stesso metodo impiegato anche per il riconoscimento del giubbotto del giocatore.

I risultati ottenuti sono mostrati nella Figura 6.15.



Figura 6.15: Riconoscimento della casa base da parte del robot

L'utilizzo del cilindro per individuare la casa base ha dei vantaggi e degli svantaggi. Tra i vantaggi ci sono sicuramente il fattore costo e il fatto di non necessitare di batterie o di altra fonte di alimentazione per il funzionamento, mentre gli svantaggi consistono nella più difficile identificazione del cilindro da parte del robot, soprattutto quando questo è molto distante.

A differenza del giubbotto del giocatore, il cilindro è di dimensioni ridotte relativamente al campo visivo del robot, di conseguenza diventa maggiormente complesso stimare se alcuni dei punti rilevati come candidati per appartenere al cilindro, effettivamente appartengono ad esso, piuttosto che a del possibile rumore nell'immagine.

Dopo aver eseguito diversi test per individuare la scelta migliore, siamo giunti alla conclusione che i waypoint, per la loro natura puntiforme, per il fatto di essere in un numero maggiore ad uno e perchè un riconoscimento basato sul colore sarebbe poco affidabile per delle zone puntiformi, trovano nei led piani la loro migliore implementazione. La casa base invece può essere realizzata tramite il cilindro, rendendo leggermente più difficile per il robot individuarla precisamente da lontano, in questo modo il robot è costretto ad esplorare maggiormente lo spazio che lo circonda.

6.3.5 Riconoscimento degli ostacoli da parte del robot

Il riconoscimento degli ostacoli da parte del robot viene effettuato tramite la cintura di sensori di prossimità montata sul robot, la gestione, l'interpretazione e l'utilizzo dei dati provenienti da tale strumentazione sono già stati oggetto di studio dei progetti RoboWII 2.0 e RoboWII 2.0.1 con esito positivo.

6.3.6 Puntamento del WiiMote verso il robot

Il puntamento del WiiMote verso il robot ha come idea alla base di esso lo sfruttamento della telecamera a infrarossi montata sul WiiMote. Tramite questa telecamera vengono rilevati i led infrarossi montati sul robot, in particolare è possibile ottenere le coordinate di un numero di sorgenti di emissione fino a quattro.

Il posizionamento dei led sul robot è indicato nella Figura 6.16 dai led di colore blu.



Figura 6.16: Led infrarossi montati sul robot

6.3.7 Rilevamento della eccessiva vicinanza del giocatore al robot

Per il rilevamento dell'eccessiva vicinanza del giocatore al robot è stato deciso di sfruttare il particolare posizionamento dei led infrarossi: su ogni spalla è montato un led mentre i restanti 8 sono posizionati, rivolti in ogni direzione, sul petto del robot.

Quando il giocatore si trova alla giusta distanza dal robot, puntandolo con il WiiMote, esso riconoscerà 3 punti come sorgenti di emissione; quando invece il giocatore si troverà troppo vicino, la telecamera del WiiMote riuscirà a distinguere altri dei led posizionati sul petto del robot, portando quindi il conteggio delle sorgenti individuate a 4: in questo caso si rileva l'eccessiva vicinanza. Questa soluzione è stata provata durante il lavoro ed ha avuto esito positivo.

Capitolo 7

Esempi di scenario

In questo capitolo verranno illustrati alcuni scenari d'uso del gioco, in particolare verranno presentate due partite tipiche, una nella quale il robot vincerà la partita ed una nella quale sarà il giocatore umano ad aggiudicarsi la vittoria.

7.1 Scenario: vittoria robot

In questo scenario viene descritta una tipica partita a RoboWII 2.1, terminante con la vittoria del robot. Di seguito vengono elencati i passi:

- il robot parte e cerca la casa base;
- il giocatore impugna la pistola e punta il robot;
- il robot cerca di evitare il puntamento;
- il giocatore fa fuoco mentre i 4 led di puntamento sono accesi e colpisce il robot;
- il robot perde una unità di energia ed accende il primo del rosso;
- il robot individua uno dei waypoint e si dirige verso di esso;
- il giocatore ripone la pistola ed estrae il fucile;
- il giocatore attende il puntamento fino a 3 led gialli accesi e fa fuoco colpendo il robot;
- il robot riduce la velocità a causa della diminuzione della stamina dopo il colpo subito;

- il giocatore cerca di colpire nuovamente il robot con il fucile ma manca il bersaglio;
- il robot raggiunge il waypoint e ritorna a piena energia, spegne il led rosso;
- il giocatore ricarica il fucile, attende il puntamento fino a 4 led gialli e fa fuoco colpendo il robot;
- il robot perde due unità di energia ed accende due led rossi;
- il giocatore prende di nuovo la mira ma nel frattempo il robot ha raggiunto la casa base;
- il robot ha vinto.

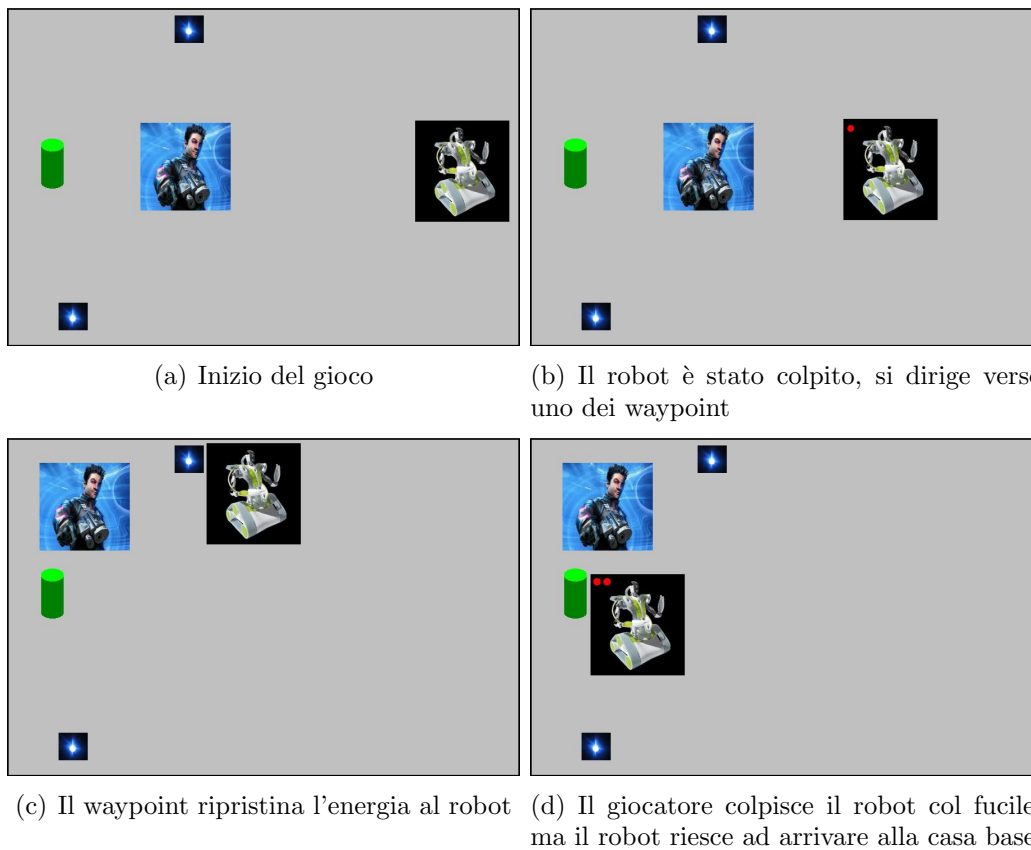


Figura 7.1: Esempio di partita vinta dal robot

7.2 Scenario: vittoria giocatore

In questo scenario viene invece descritta una tipica partita al gioco, dove però la vittoria va al giocatore.

- il robot parte e cerca la casa base;
- il giocatore estrae il fucile e punta il robot;
- il giocatore attende il puntamento a 4 led gialli accesi e fa fuoco;
- il robot perde due unità di energia ed accende due led rossi;
- il robot prende di mira il giocatore ed accende il suo led bianco;
- il giocatore non riesce a spostarsi in tempo ed il robot colpisce il giocatore, disattivandone l'arma;
- il robot procede verso la casa base;
- il giocatore, terminato il tempo di ripresa, torna a mirare il robot, ma è troppo vicino e viene acceso il led verde sul robot;
- il giocatore si allontana, ripone il fucile ed estrae la spada;
- il giocatore colpisce il robot con la spada;
- il robot, con la propria stamina ridotta al minimo si muove in maniera estremamente limitata;
- il giocatore ripone la spada ed estrae il fucile;
- il giocatore attende il puntamento fino a 4 led gialli e fa fuoco;
- il robot perde altre due unità di energia ed accende tutti i 4 led rossi;
- il giocatore ha vinto.



Figura 7.2: Esempio di partita vinta dal giocatore

Capitolo 8

Descrizione del prototipo

Per la realizzazione del prototipo del progetto descritto, è stato scelto di utilizzare un approccio di tipo *Wizard of Oz*. In questo paradigma, utilizzato nell'ambito dell'*human computer interaction*, tutte oppure solamente alcune delle funzionalità svolte dal robot sono in realtà gestite da una persona umana, la quale ha il compito di fare compiere al robot le corrette azioni, in maniera da simulare un comportamento di tipo intelligente da parte del robot. In questa maniera è possibile fornire ad un osservatore esterno una idea semplice e precisa di come debbano essere le meccaniche, le regole e lo svolgimento del gioco.

La realizzazione del prototipo secondo l'approccio descritto è stato realizzato all'interno del laboratorio AIRLab del Politecnico di Milano. E' stato realizzato un filmato che riprende le principali azioni di gioco singolarmente, in modo da indicarle in maniera chiara, e successivamente sono state svolte alcune simulazioni di vere e proprie partite.

Come giocatore umano è stato scelto un bambino della fascia di età al quale il gioco è rivolta.

Il robot utilizzato è stato Spykee, con apportate le modifiche descritte nel Capitolo 6; la sua gestione è stata affidata ad un utilizzatore umano avente il compito di provvedere al movimento in accordo con la situazione corrente di gioco, mentre gli altri aspetto del gioco sono stati quasi tutti completamente automatizzati, i dettagli dello stato di implementazione sono indicati nella Tabella 8.1.

Aspetto di gioco	Stato di implementazione
Puntamento del robot da parte del giocatore	Implementato
Visualizzazione munizioni disponibili sul WiiMote	Implementato
Visualizzazione stato del puntamento	Implementato
Visualizzazione energia del robot	Implementato
Rilevamento eccessiva vicinanza	Implementato
Rilevamento giocatore	Implementato
Rilevamento waypoint	Implementato
Rilevamento casa base	Implementato
Gestione della comunicazione WiiMote/unità di elaborazione	Implementato
Gestione della comunicazione unità di elaborazione/led	Implementato
Cambio armi e ricarica	Parziale
Rilevamento ostacoli	Non implementato
Guida automatica del robot	Non implementato

Tabella 8.1: Implementato: completamente automatizzato, Parziale: gestito in parte automaticamente, Non implementato: gestione tramite approccio Wizard of Oz

Capitolo 9

Conclusioni

La realizzazione di questo lavoro ha preso in esame diverse problematiche relative alla realizzazione di un gioco di interazione uomo – robot. La progettazione, la prova delle idee e la successiva modifica hanno dato modo di comprendere i diversi aspetti legati alla realizzazione, a partire dalla semplice idea del gioco, alla successiva individuazione delle caratteristiche che rivestono un ruolo maggiormente importante, fino alla progettazione dell'interfaccia ed alla esecuzione delle modifiche apportate al robot per renderne possibile l'impiego.

Durante la realizzazione è stata svolta una esplorazione delle alternative per la realizzazione dei concetti di gioco, come ad esempio l'impiego di RFID per identificare i waypoint e la casa base, che se da un lato avrebbero aggiunto maggiore precisione, dall'altro avrebbero significato rendere il gioco molto più costoso da realizzare ed avrebbero abbassato la durata delle batterie che alimentano i dispositivi montati sul robot.

Tramite l'analisi di queste ed altre alternative si è giunto ad identificare nelle scelte operate, quelle migliori per la realizzazione del gioco.

L'ultimo aspetto da considerare consiste nel capire se davvero il gioco così realizzato sia divertente; chiedendo al bambino che lo ha provato durante la registrazione delle partite se si sia divertito, questo ha risposto con un chiaro *si!*

Questo non può essere un giudizio assoluto sul gioco realizzato, ma il fatto che la persona che lo ha provato si sia divertita, non può essere che un segno positivo per noi che lo abbiamo realizzato.

Appendice A

Glossario

Vengono di seguito riportati alcuni dei termini chiave relativi al gioco Robo-WII 2.1 con la relativa definizione:

Energia Identifica il livello vitale del robot, una volta esaurito il quale, il giocatore vince;

Casa base Traguardo definitivo del robot, nel caso il robot riesca a giungervi vince la partita;

Distanza di targetting Distanza minima che il giocatore deve rispettare dal robot, nel caso il giocatore si avvicini oltre questa distanza, non potrà eseguire il targetting fino a che non uscirà dell'area;

Ostacolo Qualsiasi cosa o persona intralci o costituisca una difficoltà al movimento del robot;

Stamina Rappresenta la stanchezza del robot, indica la possibilità del robot di compiere una determinata serie di operazioni ognuna delle quali richiede una certa quantità di stamina; si rigenera con il passare del tempo;

Tempo di targetting Intervallo di tempo durante il quale il giocatore mantiene il robot sotto tiro, senza sparare;

Tempo di ripresa Intervallo di tempo durante il quale il giocatore avrà la funzione di sparo disabilitata in seguito ad un colpo ricevuto dal robot;

Sistema Entità arbitro del gioco, implementata tramite l'unità di elaborazione esterna;

Waypoint Traguardo intermedio per il robot, può ricaricare la stamina e/o l'energia nel caso venga raggiunto.

Bibliografia

- [1] Airlab. <http://airlab.ws.dei.polimi.it>.
- [2] Lego mindstorms. http://mindstorms.lego.com/eng/Copenhagen_Dest/default.aspx.
- [3] Lurch. http://airwiki.elet.polimi.it/mediawiki/index.php/LURCH_-_The_autonomous_wheelchair.
- [4] Robocup. <http://www.robocup.org>.
- [5] Robowii. <http://airwiki.elet.polimi.it/mediawiki/index.php/ROBOWII>.
- [6] Rovio robot. <http://www.wowwee.com/en/products/tech/telepresence/rovio/rovio>.
- [7] Spykee. <http://www.spykeeworld.com>.
- [8] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 2008.
- [9] Andrien Servier e Andrea Nicotra. Robowii 3.0. <http://airwiki.elet.polimi.it/mediawiki/index.php/RoboWII3.0>.
- [10] Antonio Bianchi e Ben Chen. Robowii 1.0. <http://airwiki.elet.polimi.it/mediawiki/index.php/RoboWII1.0>.
- [11] Gabriele Pallotta e Luigi Parpinel. Robowii 2.0.1. <http://airwiki.elet.polimi.it/mediawiki/index.php/RoboWII2.0.L>.
- [12] Antonio Micali. Robowii 2.0. <http://airwiki.elet.polimi.it/mediawiki/index.php/RoboWII2.0>.
- [13] Andrea Pontecorvo. Robowii 2.0.1. <http://airwiki.elet.polimi.it/mediawiki/index.php/RoboWII2.0.1>.