

Laboratorio di Human Computer Interaction

## ROBOGAMES DESIGN

### Robo Hide\_and\_Hunt

*Matteo Botta*

*Alberto Bottinelli*

*Massimo Luraschi*

A.A. 2008-2009

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Progettazione della user experience</b>	<b>2</b>
2.1	Target User . . . . .	2
2.2	Numero di utenti . . . . .	3
2.3	Targent Environment . . . . .	3
2.4	Regole Principali . . . . .	3
2.5	Livelli di difficoltà . . . . .	4
2.6	Scenario di gioco . . . . .	4
2.7	Macchina a stati . . . . .	5
2.7.1	Descrizione stati . . . . .	6
2.7.2	Descrizione eventi . . . . .	7
2.8	Interazione . . . . .	8
<b>3</b>	<b>Analisi delle tecnologie utilizzate</b>	<b>10</b>
3.1	Tecnologie utilizzate . . . . .	10
3.2	Tecnologie trasmissione dei dati . . . . .	11
3.3	Specifiche tecniche del gioco . . . . .	12
3.3.1	Il Robot . . . . .	13
3.3.2	Il Controller . . . . .	13
3.3.3	Il Computer . . . . .	14

# Capitolo 1

## Introduzione

Lo scopo del nostro progetto è sviluppare un Robogame nel quale il robot agisca in modo autonomo. In questo studio illustreremo una proposta e ne analizzeremo la fattibilità tecnologica. L'obiettivo principale è far sì che l'interazione uomo-robot, sia essa di tipo competitivo o collaborativo, porti a massimizzare il divertimento e il coinvolgimento dell'utente.

## Capitolo 2

# Progettazione della user experience

*Descrive le soluzioni proposte per soddisfare i requisiti, sia con testo descrittivo che con altri strumenti visivi, ad esempio regole del gioco, scenari - sequenze di immagini rappresentative della user experience, eventualmente a progressivi livelli di astrazione (ad esempio: prima design scenarios ad alto livello che descrivono i task principali della user experience, poi interaction scenarios che descrivono sequenze di interazione con l'interfaccia del mobile e con l'ambiente)*

### 2.1 Target User

L'applicazione è pensata per un intervallo di età compreso tra i 6 e 10 anni poiché, come i classici giochi puerili da cui deriva, oltre una certa soglia di età non suscitano più un interesse continuo. L'età minima, invece, è giustificata dalla necessità di comprensione del regolamento di gioco.

## 2.2 Numero di utenti

Il gioco nasce come uno contro uno (giocatore-robot) ma è possibile espandere la giocabilità a più giocatori umani ed eventualmente più cacciatori robotici.

## 2.3 Targent Environment

Il gioco è pensato per essere utilizzato in ambienti interni. Questa scelta è dovuta al fatto che i sensori presenti sul robot potrebbero non funzionare correttamente e rilevare informazioni sbagliate se utilizzati all'aperto. Inoltre non bisogna trascurare le possibili difficoltà di movimento riscontrate dal robot in caso di asperità del terreno, che renderebbe impossibile lo svolgimento del gioco.

## 2.4 Regole Principali

- **Vittoria Player:** l'utente deve colpire il robot un numero prefissato di volte. Il colpo viene considerato valido se effettuato quando l'utente non è nel campo visivo del robot. I colpi non possono essere sparati troppo frequentemente poiché c'è un tempo di ricarica del 'fucile'.
- **Vittoria Robot:** lo scopo del robot è stanare il giocatore e mantenerlo inquadrato nel campo visivo per un tempo prefissato o per un numero prefissato di avvistamenti. Il tempo è cumulabile tra i vari avvistamenti.
- Il fucile necessita di un tempo di ricarica dopo ogni sparo, superiore o al più pari al tempo di avvistamento che decreta la vittoria del robot.
- Affinché un colpo vada a segno il giocatore non deve trovarsi all'interno del campo visivo della telecamera del robot.

## 2.5 Livelli di difficoltà

E' possibile personalizzare il gioco settando diversi parametri. Ad esempio è possibile scegliere il numero di colpi con cui colpire il robot modificando così la durata della partita, oppure si può agire sulla difficoltà del gioco andando a modificare il tempo di avvistamento, l'ampiezza dell'angolo di tiro del fucile (angolo telecamera Wiimote) e il cooldown dell'arma.

## 2.6 Scenario di gioco

Avviato il gioco il robot concede un intervallo di tempo predefinito al giocatore per allontanarsi e nascondersi (conta). Terminato questo lasso di tempo il robot inizia la ricerca del player sfruttando un algoritmo di ricerca casuale fino a che non si presenta uno di questi due eventi: l'utente spara oppure il robot percepisce il giocatore nelle sue vicinanze. Nel primo caso bisogna distinguere se il colpo va a vuoto oppure se il robot viene colpito. In caso di mancato bersaglio il robot compie un giro completo su se stesso al fine di individuare se il giocatore è presente nelle vicinanze. In caso di rilevamento eseguirà le azioni spiegate in seguito. Se invece lo sparo va a segno il robot rimarrà 'stunato' per qualche secondo permettendo così al player di nascondersi poiché tra uno sparo e l'altro bisogna attendere il tempo di 'cooldown'. Infine viene aggiornato il punteggio. Nel caso in cui il robot percepisce l'utente, Spike cercherà di posizionarsi in modo tale da inquadrare il giocatore. Nel momento in cui il giocatore risulti essere nel campo visivo del robot, il punteggio di quest'ultimo (numero di avvistamenti e/o tempo di avvistamento) viene incrementato finché l'utente non uscirà dalla sua visuale. Il gioco continua con il susseguirsi di queste azioni fino a che uno dei due soggetti non arriva al punteggio finale.

2.7 Macchina a stati

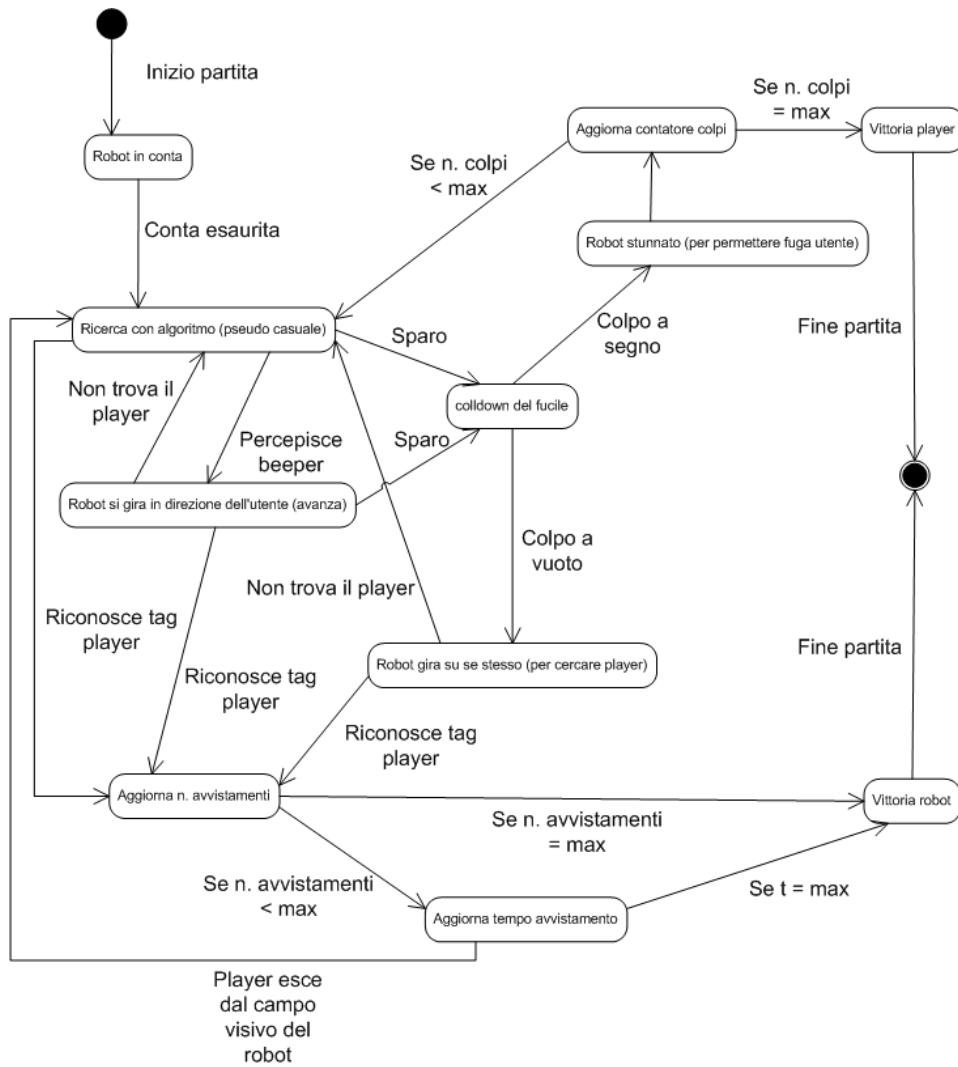


Figura 2.1: Macchina a stati che rappresenta lo svolgimento della partita

### 2.7.1 Descrizione stati

**Robot in conta:** all'inizio della partita il robot lascia un margine di tempo di fuga all'utente.

**Robot in ricerca (con algoritmo pseudo casuale):** il robot cerca il player all'interno dell'ambiente di gioco attraverso un algoritmo di esplorazione.

**Robot si gira in direzione dell'utente (avanza):** quando il robot percepisce il beeper cerca di scovare l'utente nelle vicinanze della direzione percepita.

**Cooldown del fucile:** ogni volta che l'utente spara il fucile rimane inattivo per un tempo prefissato.

**Robot stunnato (per permettere fuga utente):** dopo che il robot viene colpito rimane in uno stato di stun (sensori beeper e telecamera disattivati) per un breve intervallo di tempo, che permette all'utente di scappare.

**Robot gira su se stesso (per cercare player):** dopo uno sparo mancato il robot analizza lo spazio circostante alla ricerca del player.

**Aggiorna numero avvistamenti:** incrementa il numero di avvistamenti del robot.

**Aggiorna contatore colpi:** incrementa i colpi messi a segno dal player.

**Aggiorna tempo avvistamento:** incrementa il tempo di avvistamento del robot.

**Vittoria robot:** il robot vince la partita.

**Vittoria player** il player vince la partita.



### 2.7.2 Descrizione eventi

**Inizio partita:** il gioco inizia.

**Conta esaurita:** termina il periodo di tempo lasciato al player per scappare.

**Non trova il palyer:** il robot non percepisce nessun seganle dall'utente.

**Percepisce il beeper:** il robot percepisce il segnale emesso dal beeper dell'utente.

**Riconosce tag player:** il tag dell'utente entra nel capo visivo del robot.

**Sparo:** il palyer spara.

**Colpo a vuoto:** il player spara e manca il robot.

**Colpo a segno:** il player spara e colpisce il robot.

**Se  $n$ . avvistamenti  $<$  max:** il numero di volte che l'utente è entrato nel campo visivo del robot è minore del numero necessario di avvistamenti per la vittoria del robot.

**Se  $n$ . avvistamenti  $=$  max:** il numero di avvistamenti dell'utente ha raggiunto il numero necessario per la vittoria del robot.

**Se  $t =$  max:** il tempo in cui l'utente è stato all'interno del campo visivo del robot è uguale al tempo di avvistamento necessario per far vincere il robot.

**Se  $n$ . colpi  $<$  max:** il numero di spari andati a buon fine sono minori di quelli necessari per vincere.

**Se  $n$ . colpi  $=$  max:** il numero di spari andati a buon fine sono uguali a quelli necessari per la vittoria.

**Fine partita:** il gioco termina.

## 2.8 Interazione

In questa sezione vengono illustrati i comportamenti attuati dall'applicazione per interagire con l'utente in risposta ai principali eventi che si verificano durante il gameplay. Quasi tutte le azioni sono accompagnate da segnali vocali e visivi che aiutano il giocatore a comprendere le situazioni in modo semplice, intuitivo e divertente. L'interazione è obbligata all'interno del game per evolvere nelle fasi del gioco ed arrivare alla vittoria. Il gioco è pensato per porre sullo stesso livello uomo e robot, cercando di eliminare il divario intrinseco che li differenzia.

Le tecnologie a nostra disposizione non permettono infatti di creare un'applicazione dove la componente fisica sia discriminante poiché il gioco risulterebbe impari e in conseguenza noioso e poco stimolante. Basandoci su questi ragionamenti si sono sviluppati i principali punti di interazione player-robot illustrati di seguito.

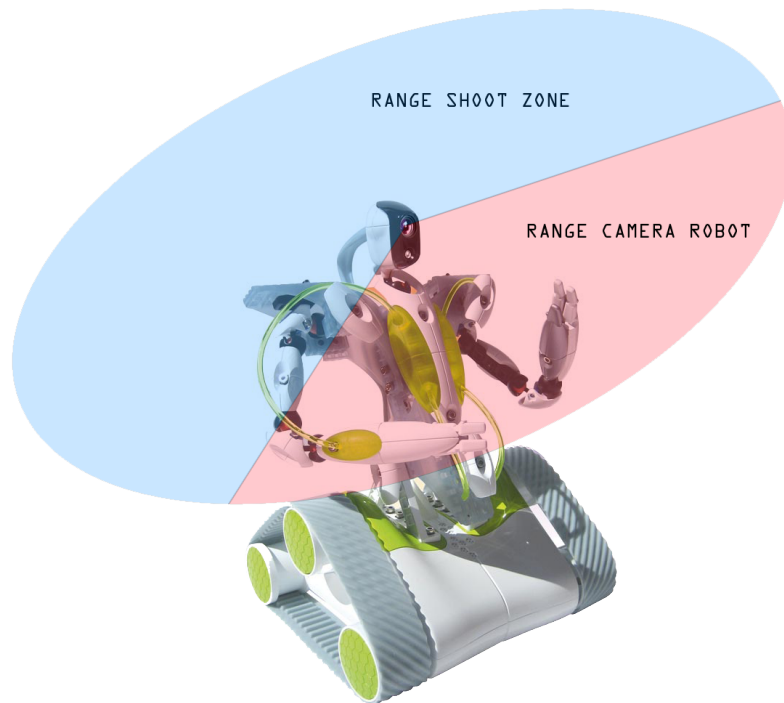
**Inizio ricerca:** Il robot scandisce il tempo della conta e ne annuncia il termine con annunci vocali. Finita la conta il robot inizia ad esplorare l'ambiente in cerca del player.

**Avvistamento:** quando il robot, durante la ricerca, percepisce il segnale del beeper lo annuncia con un particolare avviso vocale. Se il tag viene avvistato dalla camera del robot un altro segnale vocale avviserà l'utente. Inoltre, la barra dei led verrà accesa gradualmente in base al tempo di esposizione dell'utente e/o al numero di avvistamenti.

**Sparo:** l'azione dello sparo (attivata premendo il tasto B del Wiimote) da parte dell'utente, genera una molteplicità di eventi. Primo fra tutti, l'attivazione del cooldown del fucile, segnalata dai quattro led posti sul telecomando: allo sparo vengono accesi tutti contemporaneamente venendo successivamente spenti in modo graduale. Se il giocatore tenta di sparare durante questo lasso di tempo verrà avvisato da un evento sonoro. Nel caso in cui il giocatore manchi l'obiettivo un'esclamazione lo avviserà, viceversa se il robot viene colpito, esso esordirà con un'esclamazione per segnalare che il colpo è andato a segno e con una breve intermittenza delle due barre di led. Inoltre il numero di colpi

subiti verrà visualizzato attraverso una seconda barra di led (di colore diverso da quelli che segnalano gli avvistamenti). Infine se il giocatore tenta di sparare mentre si trova nel campo visivo del robot, quest'ultimo emette un altro segnale vocale che avvisa l'utente dell'impossibilità di mandare a segno il colpo.

**Fuga:** un avviso sonoro segnala l'uscita del giocatore dal campo visivo del robot.



*Figura 2.2: zona di avvistamento comparata con la zona in cui il robot può essere colpito.*

## Capitolo 3

# Analisi delle tecnologie utilizzate

*discussione della tecnologia architettura hw e sw che permette la realizzazione del design proposto*

### 3.1 Tecnologie utilizzate

In questo paragrafo vengono illustrate e motivate le tecnologie coinvolte nella realizzazione di questo robogame.

**Il Robot:** in commercio esistono due robot adatti al nostro gioco: Spyke e Rovio, differiscono tra loro principalmente per le capacità di movimento (il secondo è ologonico mentre l'altro no).

**Il Wii remote:** Il telecomando principale della console Nintendo Wii. La presenza di sensori (accelerometro, telecamera sensibile ai raggi infrarossi), pulsanti e led insieme alla tecnologia bluetooth lo rendono uno strumento flessibile, intuitivo nell'utilizzo e facilmente interfacciabile sia con l'utente che con l'applicazione, perciò adatto ai nostri scopi anche per il suo costo relativamente basso.

**Cintura sonar:** andrà applicata al robot per l'evitamento ostacoli. Il principio di funzionamento consiste nell'emissione di un segnale sonoro

e nella registrazione dell'eco di ritorno consentendo di individuare ostacoli nelle vicinanze (come ad esempio i pipistrelli).

**Barra led luminosi:** sul robot verrà applicata una barra di led luminosi per comunicare all'utente eventi particolari del gioco.

**Led infrarossi:** il robot sarà reso visibile al Wiimote tramite un numero di led infrarossi posti in una posizione chiave in modo da essere riconoscibili dalla telecamera del controller.

**Cinturino player:** per rendere visibile e localizzabile il giocatore da parte del robot, dovrà indossare un particolare cinturino/braccialetto dotato di un beeper in grado di emettere un costante segnale sonoro (ad una frequenza prestabilita) che permetta la robot di localizzare approssimativamente il giocatore. Inoltre quando l'utente è nel campo visivo (telecamera a bordo) del robot dovrà essere individuato univocamente tramite un tag presente sullo stesso cinturino (led, materiale colorato, ecc).

**Riconoscitore del beeper:** per localizzare il giocatore attraverso il beeper, sono necessari dei microfoni, del tutto simili a quelli della cintura sonar, tarati sulla frequenza prestabilita del beeper.

## 3.2 Tecnologie trasmissione dei dati

La comunicazioni pc-robot avviene tramite tecnologie wi-fi, in quanto è necessario avere una connessione robusta e senza fili. Per quanto riguarda il Wiimote viene utilizzata la tecnologia bluetooth per collegarlo all'elaboratore. Con questa configurazione il computer centrale gestisce e regola tutte le parti dell'applicazione. (aggiungere limiti delle 2 tecnologie)

### 3.3 Specifiche tecniche del gioco

In questa sezione si illustrano le specifiche relative ai componenti utilizzati all'interno dell'applicazione. In particolare si analizza la struttura e l'utilizzo delle varie parti del robot e del controller.

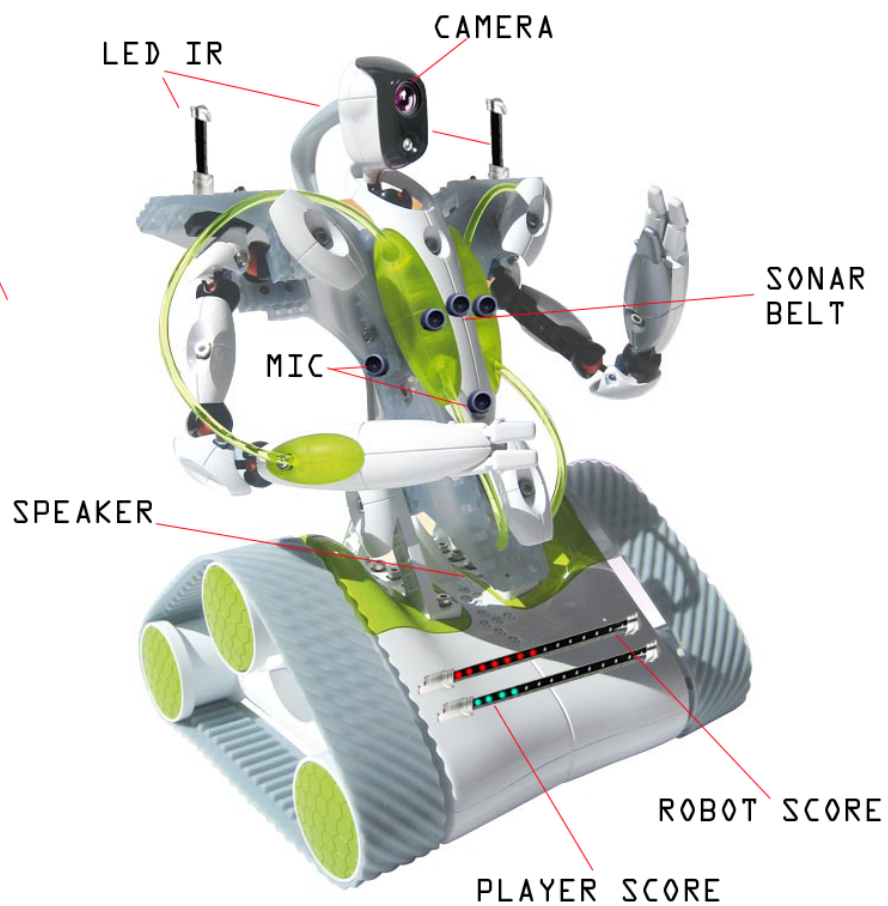


Figura 3.1: Spyke modificato

### 3.3.1 Il Robot

Per questo progetto si è deciso di utilizzare Spyke, un robot mobile presente in commercio, adeguatamente modificato sulla base delle nostre valutazioni. Sulla parte frontale del robot vengono applicate due barre di led luminosi di colori diversi, utilizzate per segnalare il punteggio relativi a player e robot. Sul corpo del robot viene inoltre aggiunta una cintura sonar per permettere a Spyke di evitare gli ostacoli presenti sul suo cammino. In aggiunta verranno applicati una serie di microfoni in grado di captare il segnale del beeper che permetteranno al robot di dirigersi verso la direzione con maggior intensità sonora, presumibilmente dove si trova il palyer. Sulle ‘spalle’ di Spyke verranno posizionati due led ad infrarossi (lunghezza d’onda 940 *nm*) per rendere visibile il robot al Wiimote. Seganli vocali e suoni, saranno emessi dallo speaker presente sul robot per aumentare il grado di interazione e coinvolgimento dell’utente duranre il gioco. Il player sarà identificato tramite i tag dalla telecamera presente sulla ‘testa’ di Spyke.

### 3.3.2 Il Controller

All’interno della nostra applicazione il giocatore interagisce con il robot grazie all’utilizzo del controller Wii Remote. Con questo dispositivo il player potrà mirare e sparare al robot per colpire. L’azione dello sparo va a buon fine se e solo se i led ad infrarossi presenti su Spikè entrano nel campo visivo della telecamera IR posta sulla testa del Wiimote. Premendo il tasto B situato sul retro del controller, il Wiimote ‘farà fuoco’ e lo seganlerà al giocatore vibrando. Dopo ogni sparo è necessario attendere un intervallo di tempo predefinito per poter esplodere un nuovo colpo. I led sul controller indicheranno il countdown del tempo di ricarica. Sul dispositivo o sul cinturino da portare al braccio, verranno applicati beeper e tag che permetteranno al robot di individuare il giocatore. I tag potranno essere eventualmente posizionati in zone diverse in base alle esigenze dell’utente ricordando però che questi devono essere facilmente visibili al robot (una collana, una cavigliera o un paio di occhiali).



Figura 3.2: il controller Wii Remote

### 3.3.3 Il Computer

L'applicazione sarà interamente caricata su un pc al quale verrà connesso il robot e il controller. L'intera elaborazione è affidata all'elaboratore, come anche l'analisi degli input forniti dall'utente tramite il controller e l'intera gestione del gioco (punteggi player e robot, inizio partita, ecc). Il computer deve essere posizionato in zona centrale rispetto al campo di gioco in realzione alla portata dei dispositivi wieless, in modo che il robot e il controller rimangano sempre connessi al pc.