

POLITECNICO DI MILANO
Polo Regionale di Como
Facoltà di Ingegneria dell'Informazione
Corso di Studi in Ingegneria Informatica



WHEREWIIARE:
SISTEMA DI LOCALIZZAZIONE
IN AMBIENTE PREDEFINITO
BASATO SUL DISPOSITIVO WII REMOTE

AI & R Lab
Laboratorio di Intelligenza Artificiale
e Robotica del Politecnico di Milano

Tutor universitario: Prof. Andrea Bonarini

Elaborato finale di:
Alberto Bottinelli, matricola 679380
Matteo Botta, matricola 679182

A.A. 2007-2008

Alle nostre famiglie...

Ringraziamenti

E' difficile ringraziare in poche righe tutte le persone che hanno contribuito al raggiungimento di questo traguardo, ma è ancora più difficile trovare le parole adeguate a fare questo. Ringrazio in primo luogo tutti i miei famigliari che mi sono stati vicino durante tutto questo periodo e mi hanno saputo sopportare nonostante il mio brutto carattere. A loro vanno tutta la mia stima, il mio rispetto e la mia riconoscenza. Ringrazio anche chi purtroppo non c'è più, ma il fatto che non ci sia più non vuol dire che non mi sia stato vicino in questo percorso. Ringrazio la mia ragazza Mirella per il supporto morale che ha saputo darmi e per la comprensione mostrata, Elisa e Matteo per tutte le torte che mi hanno preparato e che mi prepareranno. Come poi non ringraziare il mio socio di tesi Botta, con cui ho passato 4 mesi divertenti e indimenticabili. Non è stato affatto facile portare a termine questo lavoro iniziato da una semplice idea e tante speranze. Nonostante le innumerevoli difficoltà incontrate siamo comunque riusciti nell'impresa e sicuramente un grande grazie va a Botta. Un grazie ai miei compagni più cari: Mery, Rocco, Volo, Sghez, Claudio e Pini, che hanno sempre trovato il tempo di darmi una mano quando ne avevo più bisogno. Un ringraziamento anche ai veterani della biblioteca che mi hanno insegnato a vivere l'università in modo divertente e proficuo.

Per ultimo ,ma non per questo meno importate, ringrazio il Prof. Andrea Bonarini per averci permesso di poter sviluppare questa idea e per la disponibilità mostrata durante tutto il periodo di lavoro.

GRAZIE.

Alberto

Per primi ringrazio i mie genitori, mio fratello e tutti i miei familiari che mi hanno sempre permesso di scegliere da solo la strada da percorrere appoggiandomi. Ringrazio la mia ragazza Claudia che mi è stata sempre vicina nonostante io fossi intrattabile. Ringrazio il mio socio Botti che mi ha sempre sopportato anche quando dal nulla uscivo con la faticosa frase “Botti mi è venuta in mente un’idea”, lui mi ascoltava e puntualmente mi “smontava” dicendomi che non si poteva fare, ma lo ringrazio soprattutto perché è stato divertente portare avanti questo lavoro con lui. Ringrazio il professore Andrea Bonarini che ci ha permesso di sviluppare questo progetto. Inoltre volevo ringraziare i miei compagni di disavventure, week-end di studio e progetti infiniti: Rocco, Meri, Volo, Sghez, Claudio, Max, Pini, Tim, Joe e Luca. Ringrazio i “veterani” della biblioteca rumorosa che mi hanno insegnato con scherzi e innumerevoli cinque minuti di degenero a prendere l’università in modo migliore. Infine ringrazio tutti quei compagni che ormai, dopo tre anni di studi, posso chiamare amici.

Matteo

Indice

Ringraziamenti	I
1 Introduzione	1
2 Strumenti e tecnologie utilizzate	3
2.1 Il Wii Remote	3
2.1.1 La Telecamera IR	5
2.1.2 L'accelerometro	8
2.1.3 Comunicazioni pc Wii Remote	9
2.2 La libreria Motej	11
2.2.1 Connessione	11
2.2.2 Gestione telecamera IR	13
2.2.3 Gestione accelerometro	16
2.3 I Led ad infrarossi	17
2.4 I Catarifrangenti	18
2.5 L'auricolare bluetooth	19
3 Problematiche di sviluppo	20
3.1 La matrice di led	20
3.1.1 Il circuito: analisi ed implementazione	22
3.2 I tag	25
3.2.1 Forma e dimensione	25

3.2.2	Posizionamento nell'ambiente	27
3.3	Ambiente	30
3.3.1	Analisi	30
3.3.2	Disturbi	30
3.3.3	Limiti	31
4	Il sistema di localizzazione: WhereWiiAre	32
4.1	L'idea	32
4.1.1	Che cosa deve fare	32
4.1.2	A chi è destinato	33
4.2	Struttura e Funzionamento	33
4.2.1	Connessione dispositivo	34
4.2.2	Test dispositivo	37
4.2.3	Gestione mappa	38
4.2.4	Gestione Localizzazione	42
4.2.5	Utilizzo del Wiimote	46
5	Scenario di utilizzo	47
5.1	Una simulazione	47
5.2	Possibili utilizzi	49
5.2.1	Domotica	49
5.2.2	Robotica	50
5.2.3	Intrattenimento	51
6	Conclusioni e sviluppi futuri	52
6.1	Sviluppi futuri	52
6.1.1	Potenziamento Wiimote	52
6.2	Valutazioni conclusive	54
	Bibliografia	55

A	Il manuale utente	56
A.1	WhereWiiAre	57
A.2	Testa Wiimote	58
A.3	Gestione mappa	59
A.4	Start WWA	61

Capitolo 1

Introduzione

Lo scopo della nostra tesi è quello di studiare il controller della console Nintendo Wii, il Wii Remote, al fine di sviluppare ulteriori utilizzi al di fuori dell'ambito prettamente ludico. La scelta di analizzare questo particolare controller è stata determinata dalla semplicità di utilizzo da parte dell'utente e dalla presenza della tecnologia bluetooth che permette al Wiimote (chiameremo così il Wii remote all'interno del nostro elaborato) di interfacciarsi con altri dispositivi. Inoltre, l'accelerometro e la telecamera ad infrarossi, presenti al suo interno, permettono il suo utilizzo in svariati domini applicativi. A fronte di ciò, si è sviluppato un software per la localizzazione in un ambiente strutturato, che permette d'identificare la posizione del Wiimote utilizzando unicamente i dati acquisiti dal controller. In questo modo è possibile guidare persone non vedenti all'interno di un ambiente noto, tra i vari goal prefissati. Il Wiimote acquisisce tramite la telecamera ad infrarossi i Tag precedentemente posizionati, il computer riceve le informazioni e le elabora determinando la posizione dell'apparecchio. I Tag sono figure ottenute utilizzando materiale catarifrangente. Al fine di rendere i Tag visibili al Wiimote si è realizzata una fonte luminosa ad infrarossi applicata

direttamente al corpo del controller. Il software è stato sviluppato in Java e la comunicazione col Wiimote è stata realizzata sulla base della libreria Motej adeguatamente modificata. Di seguito viene fornita una breve descrizione delle sezioni che compongono la tesi.

“Strumenti e tecnologie utilizzate”

Nella sezione vengono descritti in dettaglio gli strumenti utilizzati quali il Wiimote in ogni sua parte, la libreria Motej usata per interfacciare il Wiimote al computer e gli altri componenti presi in considerazione.

“Problematiche di sviluppo”

Nella sezione si mostrano le problematiche riscontrate durante la realizzazione del progetto e le soluzioni adottate.

“Il sistema di localizzazione: WhereWiiAre”

In questa sezione vengono mostrate la struttura e la realizzazione del nostro sistema, dall'idea iniziale fino alla forma definitiva. Inoltre vengono descritte le varie parti che compongono il software e il suo funzionamento.

“Scenario di utilizzo”

Nella sezione si analizza un esempio di utilizzo del nostro sistema e si considerano i possibili scenari applicativi.

“Conclusioni e sviluppi futuri”

Nella sezione si discutono i risultati ottenuti e si descrivono ulteriori sviluppi futuri dell'applicazione legati all'evoluzione del Wiimote.

“Il manuale utente”

Nell'appendice è riportato il manuale utente per l'utilizzo dell'applicazione.

Capitolo 2

Strumenti e tecnologie utilizzate

2.1 Il Wii Remote

Il Wii Remote è il principale controller della console Nintendo Wii. Proprio grazie ad esso la console Wii si differenzia dalle altre in commercio poiché permette all'utente di interfacciarsi col videogioco in modo semplice e intuitivo. Per giocare basta compiere movimenti simili a quelli usati nella vita reale o puntare semplicemente lo schermo della tv impugnando il Wiimote. Questo è permesso grazie ai due sensori contenuti all'interno del dispositivo. Il primo è un accelerometro che riconosce i movimenti a cui è soggetto il Wiimote; il secondo è una telecamera sensibile ai raggi infrarossi che permette di determinare il punto del televisore mirato dal controller, grazie a dei led IR incorporati nelle estremità della Wii



Figura 2.1: Il Wii Remote.

Sensor Bar (da porre sopra o sotto la televisione). Vi sono inoltre a disposizione del giocatore 10 tasti digitali standard, un tasto “power” che permette di spegnere la console e infine un tasto “sync” creato per connettere via bluetooth il Wiimote alla console. Impugnando il dispositivo si notano inoltre 4 led e uno speaker. All’interno del dispositivo, oltre ai due sensori principali, sono presenti una memoria EEPROM di 16 KiB, un piccolo motore elettrico interno, collegato ad una massa eccentrica, che permette al Wiimote di vibrare e un chip bluetooth che rende possibile la comunicazione controller console. Nella parte inferiore del dispositivo è situata una porta di espansione alla quale è possibile collegare altri dispositivi, come ad esempio il Nunchuk che a sua volta contiene un accelerometro ed è dotato di 2 tasti digitali e uno stick analogico. Il controller è alimentato da due batterie Standard di tipo AA. Queste caratteristiche fanno sì che il Wiimote possa essere utilizzato in svariati domini applicativi dove è richiesta un’interfaccia uomo applicazione istintiva.

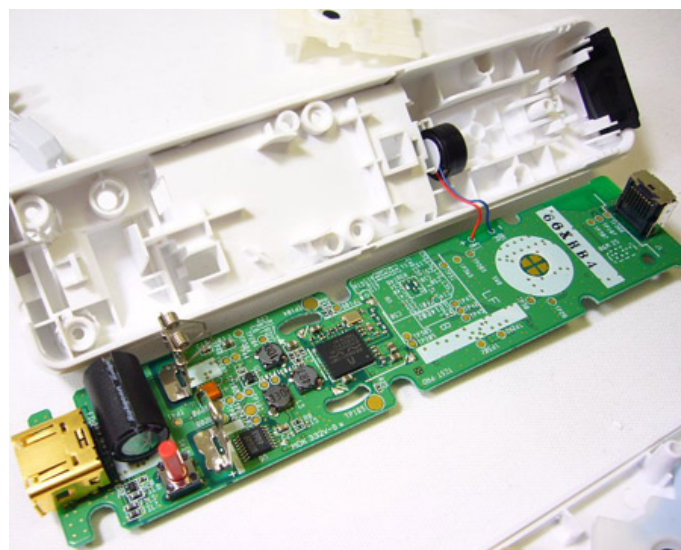


Figura 2.2: Interno del Wii Remote.

La scelta di utilizzare questo controller è stata dettata dal fatto che permette di reperire tutte le informazioni necessarie per il funzionamento del nostro sistema di localizzazione, potendo contare su di un apparecchio robusto e a basso costo. Di seguito vengono spiegati nel dettaglio i sensori principali e maggiormente coinvolti nel nostro lavoro.

2.1.1 La Telecamera IR

Sulla parte anteriore del Wiimote è montato un sensore Pixart¹ che svolge la funzione di telecamera. Più precisamente si tratta di una telecamera monocromatica con definizione 128x96 con applicato un filtro che rende visibile al Wiimote solo luci infrarosse con una lunghezza d'onda

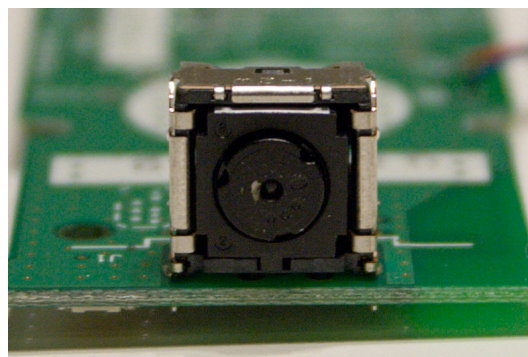


Figura 2.3: Telecamere del Wii Remote.

definita (intorno ai 940 nm). Integrato nel Wiimote e collegato alla telecamera è presente un processore che si occupa di tracciare il movimento di 4 punti. Questo significa che il Wiimote non percepisce l'intera immagine acquisita dalla camera, ma solo i punti tracciati dal processore. Per tracciare l'andamento di questi punti è applicata una 8x subpixel analysis, che rende la dimensione spaziale di 1024x768.

¹Per informazioni dettagliate sul sensore Pixart[10]

Il campo visivo della telecamera è di 45° sull'asse X e di 35° sull'asse Y e la frequenza di scansione è di 100 *hz*. Una volta abilitata la camera è possibile impostare la sensibilità e il formato di dati con il quale si desidera operare. I formati sono i seguenti:

- **BASIC** Nel formato Basic la camera restituisce 10 byte contenenti le coordinate X Y di tutti i 4 punti relative al piano di riferimento della telecamera. Le informazioni dei punti sono trasmesse a coppie, 5 byte per ogni coppia. La posizione di ogni punto è codificata in 10 bit sia per le coordinate sull'asse X che per quelle dell'asse Y con un range che va da 0 - 1023 per il primo e da 0 - 767 per il secondo.
- **EXTENDED** Nel formato Extended, la camera oltre a restituire la posizione X Y del punto, ne fornisce anche la dimensione. Per ogni punto il Wiimote invia una stringa di 3 byte, 10 bit per X e 10 bit per Y con lo stesso range del protocollo Basic, più 4 bit rappresentanti la dimensione del punto in base ai pixel occupati sulla matrice del CCD della camera. Questo valore ha un range compreso tra 0 e 15, per un totale di 12 byte per tutti i 4 punti.
- **FULL** Nel formato Full, oltre a tutti i dati forniti dal formato Extended, sono presenti le informazioni riguardanti il bounding box ($\max(X)$, $\max(Y)$, $\min(X)$, $\min(Y)$) e l'intensità dei punti. Il bounding box rappresenta un rettangolo contenente tutti i punti visti dalla telecamera. Per ogni punto questa modalità richiede 9 byte per un totale di 36 byte per i 4 punti.

Al fine di tracciare i punti con maggiore precisione, è stata utilizzata la massima sensibilità permessa della telecamera. Per quanto riguarda invece il formato dei dati, è stato scelto l'Extended in quanto includeva l'informazione riguardante la dimensione del punto. Purtroppo però, dalle prove eseguite, è emerso che questo valore non è affatto affidabile. Avendo a disposizione 4 bit ci si aspetta un range tra 0 e 15, ma purtroppo nella realtà questo varia da 0 a 6 e non segue una distribuzione lineare. La Nintendo Wii utilizza come fonte di illuminazione la Sensor Bar, un dispositivo da posizionare sopra o sotto il televisore composta da 2 gruppi di 5 led infrarossi posti a una distanza predefinita. Questi 2 gruppi di led sono recepiti dal Wiimote come 2 punti e in base a questo la console determina il suo posizionamento rispetto alla Sensor Bar. La possibilità di tracciare solo 4 punti è stato un vincolo determinate per la nostra applicazione, che è stato superato con il posizionamento dei tag ad hoc.

2.1.2 L'accelerometro

All'interno del Wiimote è inoltre presente un accelerometro² che rileva i movimenti a cui il controller è sottoposto in riferimento a una terna di assi solidale ad esso (Figura 2.5). Il sensore è posizionato sotto alla superficie superiore del Wiimote leggermente alla sinistra del pulsante A. L'accelerometro è l' ADXL330 inserito all'interno di un circuito integrato

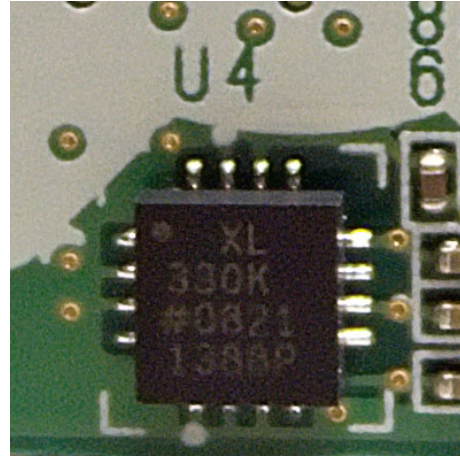


Figura 2.4: Accelerometro del Wii Remote.

(Figura 2.4) prodotto dalla Analog Devices. Il dispositivo è progettato per rilevare accelerazioni che vanno da $-3g$ a $+3g$ con un'incertezza del 10%. L'accelerometro calcola le accelerazioni in base ai movimenti delle piccole masse racchiuse all'interno dell'integrato.

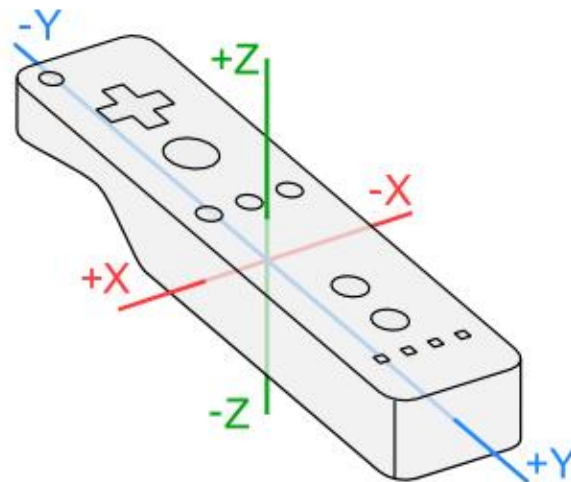


Figura 2.5: Sistema di riferimento solidale al Wiimote.

²Per informazioni dettagliate sull'accelerometro[8]

Il sensore registra sempre un'accelerazione su di un'asse a causa della forza di gravità e l'unica posizione in cui registra il valore 0 è nel caso in cui il Wiimote sia in caduta libera. All'interno della memoria EEPROM del Wiimote è memorizzato il valore equivalente a 0g e a +1g su ognuno dei 3 assi, poiché è necessario, in fase di fabbricazione, calibrare ogni singolo accelerometro. Ogni accelerazione dispone di 8 bit per la sua memorizzazione. Come la telecamera a IR anche l'accelerometro deve essere esplicitamente attivato dopo aver connesso il Wiimote al computer.

2.1.3 Comunicazioni pc Wii Remote

La comunicazione tra il Wiimote e gli altri dispositivi, quali computer o Nintendo Wii, avviene in modalità wireless, più precisamente utilizzando la tecnologia bluetooth. L'apparato che si occupa della comunicazione è un chip Broadcom BCM2042³. Questo provvede a gestire la comunicazione dei dati provenienti dai sensori del Wiimote quali accelerometro, camera IR, tasti digitali, led, vibrazione, speaker e le informazioni

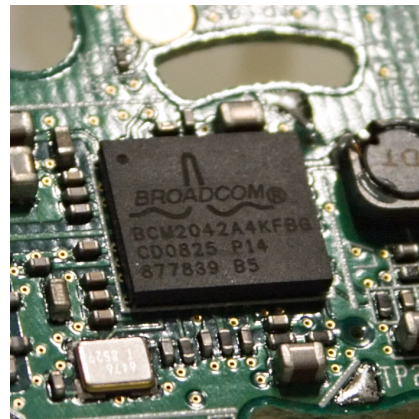


Figura 2.6: Il Chip Bluetooth.

riguardanti il controller. Il Wii remote utilizza uno standard bluetooth HID (Human Interface Device) basato sul protocollo USB HID utilizzato per le comuni periferiche I/O (tastiera, mouse e joystick). Questo protocollo permette all'host di inviare una stringa di byte al Wiimote con la quale attivare i sensori desiderati e definire la modalità di tra-

³Per informazioni dettagliate sul dispositivo Bluetooth Broadcom BCM2042[9]

smissione dei dati elaborati. A questo punto l'host si mette in ascolto dei pacchetti di informazioni provenienti dal Controller. Per connettere il Wiimote al pc non è necessaria nessuna autenticazione. Vi sono 2 modalità con il quale il controller può mettersi in discovery mode, cioè cercare un dispositivo al quale connettersi. La prima chiamata Soft Syncing, si attiva premendo simultaneamente i pulsanti 1 e 2, la seconda detta Hard Syncing si avvia premendo il tasto sync posto all'interno del vano batteria. Per connettere il Wiimote viene normalmente utilizzato il Soft Syncing. Una volta connesso, il Wiimote invia all'host le seguenti informazioni:

Name Nintendo RVL-CNT-01

Vendor ID 0x057e

Product ID 0x0306

Major Device Class 1280

Minor Device Class 4

Service Class 0

Summary of all Class Values 0x002504

Queste informazioni identificheranno il Wiimote e, nel caso si voglia connettere più controller, ci si baserà sull'indirizzo MAC.

2.2 La libreria Motej

Per scrivere la nostra applicazione si è scelto di utilizzare come linguaggio di programmazione Java, in quanto è multiplatforma. Per far sì che Java riesca a interfacciarsi e utilizzare il Wiimote ci siamo basati sulla libreria open-source Motej 0.8⁴ da noi adeguatamente adattata e modificata. Di seguito trattiamo i punti principali da noi sviluppati.

2.2.1 Connessione

Qui di seguito è riportata una parte del codice riguardante la connessione del Wiimote. Una volta trovato un device, vengono scritte sul file di log le seguenti informazioni: Name, Bluetooth Address, Major Device Class, Minor Device Class e Service Classes. Queste informazioni identificano in modo univoco il dispositivo. Per verificare che il dispositivo collegato sia di tipo Wii remote bisogna controllare che il Name inizi con Nintendo, che il Major Device Class corrisponda a 1280 e il Minor Device Class sia 4. Se le informazioni rilevate corrispondono, il device verrà accettato e connesso al software, il quale gli assocerà la variabile “Mote” attraverso la quale sarà possibile ottenere e inviare tutte le informazioni necessarie.

```
protected final DiscoveryListener listener = new DiscoveryListener() {  
  
public void deviceDiscovered(RemoteDevice device, DeviceClass clazz) {  
    try {  
        if (log.isInfoEnabled()) {  
            log.info("found device: "  
+ device.getFriendlyName(true) + " - "
```

⁴Per tutte le informazioni riguardanti libreria Motej[5]

```
+ device.getBluetoothAddress() + " - "  
+ clazz.getMajorDeviceClass() + ":"  
+ clazz.getMinorDeviceClass() + " - "  
+ clazz.getServiceClasses());  
}  
  
if (clazz.getMajorDeviceClass() == 1280 &&  
clazz.getMinorDeviceClass() == 4 &&  
device.getFriendlyName(true).startsWith("Nintendo")) {  
    if (advancedDiscovery) {  
        Mote mote = new Mote(device);  
        fireMoteFound(mote);  
    } else {  
        mote = new Mote(device);  
        discoveryAgent.cancelInquiry(this);  
    }  
}  
} catch (IOException ex) {  
    throw new RuntimeException(ex.fillInStackTrace());  
}  
}
```

2.2.2 Gestione telecamera IR

Byte	Bit							
	7	6	5	4	3	2	1	0
0	X<7:0>							
1	Y<7:0>							
2	Y<9:8>	X<9:8>						S<3:0>

Figura 2.7: Suddivisione dei Byte in base alle informazioni riguardanti gli assi nel formato Extended.

0x33: Core Buttons and Accelerometer with 12 IR bytes

This mode returns data from the buttons, accelerometer, and IR Camera in the Wii Remote:

```
(a1) 33 BB BB AA AA AA II II II II II II II II II II II
```

BBBB is the core Buttons data. AA AA AA is the Accelerometer data. The 12 II bytes are from the built-in IR Camera.

Figura 2.8: Struttura dati da noi utilizzata per leggere i dati relativi al Wiimote.

Come si può notare (Figura 2.8) il primo byte di informazione riguardante la camera è il settimo. Questo contiene il valore della coordinata X del primo punto visto dal sensore, mentre il byte numero 8 contiene il valore della coordinata Y. Osservando il formato Extended (Figura 2.7) si può notare che ogni coordinata è rappresentata con 10 bit non consecutivi. La coordinata X è contenuta nel byte 0 più 2 bit nel byte 2, mentre la coordinata Y è contenuta nel byte 1 più 2 bit del byte 2. Inoltre 4 bit vengono occupati dalla dimensione del punto. A fronte di ciò il nostro software si preoccupa di estrapolare il valore delle coordinate svolgendo operazioni logiche sui bit in questione. Una volta estrapolati i valori e controllata la loro veridicità, questi vengono letti dal Listener che si occupa della loro gestione. Sempre in questa sezione di codice

è possibile notare la variabile `Numeropunti` che tiene il conteggio dei punti osservati in un determinato istante dal Wiimote.

```
protected void parseExtendedIrCameraData(byte[] bytes, int offset) {
    int x0 = bytes[7] & 0xff ^ (bytes[9] & 0x30) << 4;
    int y0 = bytes[8] & 0xff ^ (bytes[9] & 0xc0) << 2;

    source.numero_punti=0;
        int size0 = bytes[9]& 0x0f;

    if (x0 < 1023 && y0 < 1023) {
        source.fireIrCameraEvent(IrCameraMode.EXTENDED, 0, x0, y0, size0);
            source.numero_punti++;
    }

    int x1 = bytes[10] & 0xff ^ (bytes[12] & 0x30) << 4;
    int y1 = bytes[11] & 0xff ^ (bytes[12] & 0xc0) << 2;
    int size1 = bytes[12] & 0x0f;
    if (x1 < 1023 && y1 < 1023) {
        source.fireIrCameraEvent(IrCameraMode.EXTENDED, 1, x1, y1, size1);
            source.numero_punti++;
    }

    int x2 = bytes[13] & 0xff ^ (bytes[15] & 0x30) << 4;
    int y2 = bytes[14] & 0xff ^ (bytes[15] & 0xc0) << 2;
    int size2 = bytes[15] & 0x0f;
    if (x2 < 1023 && y2 < 1023) {
```

```
source.fireIrCameraEvent(IrCameraMode.EXTENDED, 2, x2, y2, size2);
    source.numero_punti++;
}

int x3 = bytes[16] & 0xff ^ (bytes[18] & 0x30) << 4;
int y3 = bytes[17] & 0xff ^ (bytes[18] & 0xc0) << 2;
int size3 = bytes[18] & 0x0f;
if (x3 < 1023 && y3 < 1023) {
source.fireIrCameraEvent(IrCameraMode.EXTENDED, 3, x3, y3, size3);
    source.numero_punti++;
}

}
```

2.2.3 Gestione accelerometro

Come si può notare (Figura 2.8) il primo byte di informazione riguardante l'accelerometro è il quarto. Questo contiene il valore dell'accelerazione misurata sull'asse X, mentre i byte numero 5 e 6 contengono rispettivamente il valore dell'accelerazione letti sull'asse Y e sull'asse Z. Osservando la struttura dati, è possibile notare che ogni asse ha a disposizione 8 bit per memorizzare l'informazione. A fronte di ciò il nostro software si preoccupa di estrapolare il valore delle accelerazioni sui relativi assi e, una volta estrapolati, questi vengono letti dal Listener che si occupa della loro gestione.

```
protected void parseAccelerometerData(byte[] bytes) {  
    int x = bytes[4] & 0xff;  
    int y = bytes[5] & 0xff;  
    int z = bytes[6] & 0xff;  
    source.fireAccelerometerEvent(x, y, z);  
}
```


2.3 I Led ad infrarossi

LED è l'acronimo di Light Emitting Diode (diode ad emissione luminosa)⁵. Il dispositivo sfrutta le proprietà ottiche di alcuni materiali semiconduttori per produrre fotoni a partire dalla ricombinazione di coppie elettrone-lacuna. Gli elettroni e le lacune vengono iniettati in una zona di ricombinazione attraverso due regioni del diodo drogate con impurità di tipo diverso e cioè di tipo n per gli elettroni e p per le lacune. Il colore della radiazione emessa è definito dalla distanza in energia tra i livelli energetici di elettroni e lacune e corrisponde tipicamente al valore della banda proibita del semiconduttore in questione. I Led sono uno speciale tipo di diodi a giunzione p-n, formati da un sottile strato di materiale semiconduttore drogato. Quando sono sottoposti ad una tensione diretta per ridurre la barriera di potenziale della giunzione, gli elettroni della banda di conduzione del semiconduttore si ricombinano con le lacune della banda di valenza rilasciando ener-

⁵Per ulteriori informazioni riguardanti i diodi LED[12]

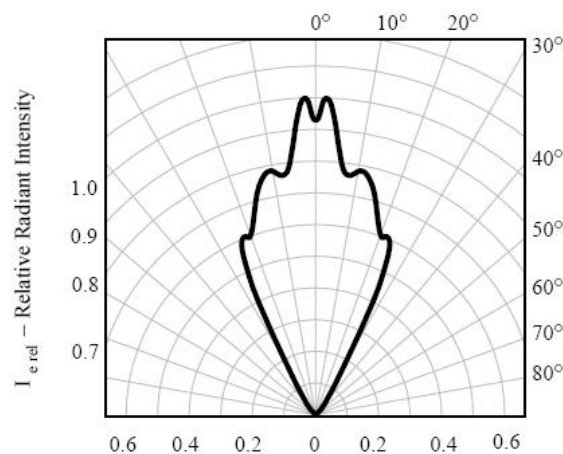


Figura 2.9: Angolo di irraggiamento del led TSAL6400.

gia sufficiente da produrre fotoni. A causa dello spessore ridotto del chip un ragionevole numero di questi fotoni può abbandonarlo ed essere emesso come luce. Alla telecamera del Wiimote sono visibili solo luci infrarosse quindi i led da noi utilizzati sono Led che emettono radiazioni luminose infrarosse. La radiazione infrarossa ha una lunghezza d'onda compresa tra 700 nm e 1 μm , in particolare i led da noi utilizzati, i Vishay TSAL6400, emettono una radiazione con una lunghezza d'onda di 940 nm con un angolo di irraggiamento che va da -25° a $+25^\circ$ (Figura 2.10).



Figura 2.10: Materiali catarifrangenti.

2.4 I Catarifrangenti

Il catarifrangente è un riflettore che ha la caratteristica di riflettere la luce nella stessa direzione da cui essa proviene. Un catarifrangente ideale riesce a rinviare la luce alla sorgente, qualunque sia l'angolo di incidenza. La costruzione dei catarifrangenti è effettuata ponendo l'una accanto all'altra delle microcelle catarifrangenti, così da ottenere superfici di dimensioni variabili a seconda delle necessità. Osservando una superficie catarifrangente si può facilmente vedere come essa sia in effetti costituita da più celle adiacenti. In caso di necessità di grande

capacità di riflessione, possono essere utilizzati dei catarifrangenti costituiti da un unico prisma in vetro di grandi dimensioni con superfici interne a specchio. Per la nostra applicazione sono state utilizzate delle strisce di nastro catarifrangente per riflettere la luce infrarossa emessa da una matrice di led. Tra tutti i materiali catarifrangenti da noi testati, quello che ha permesso di ottenere i migliori risultati è stato il nastro catarifrangente utilizzato in nautica. Questo materiale è soggetto all'omologazione SOLAS (Safety Of Life at Sea) e viene utilizzato per rendere visibili i dispositivi di salvataggio a grande distanza.

2.5 L'auricolare bluetooth

La nostra applicazione per interagire con l'utente utilizza segnali sonori. Al fine di ottenere una soluzione più compatta e flessibile possibile, è stato deciso di utilizzare un auricolare bluetooth. Il traffico audio viene reindirizzato all'auricolare permettendo così un ascolto dei segnali indipendentemente dalla posizione in cui ci si trova, evitando in tal modo di dovere cablare un impianto ad hoc.

Capitolo 3

Problematiche di sviluppo

3.1 La matrice di led

Il nostro sistema si localizza nell'ambiente in base ai tag letti dalla telecamera. Il primo problema affrontato è stato scegliere come rendere visibili i tag al Wiimote poiché esso riconosce solo fonti luminose infrarosse. Abbiamo in considerato due possibilità: posizionare nell'ambiente diodi led emettitori di raggi infrarossi, oppure tag di materiale catarifrangente. La prima scelta comporta notevoli complicazioni a livello di predisposizione nell'ambiente e manutenzione, dato che ogni tag richiede un apposito circuito e un'appropriata alimentazione.

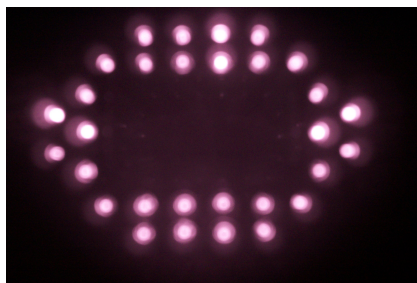


Figura 3.1: La Matrice di led accesa.

La seconda comporta il posizionamento di una fonte luminosa direttamente sul Wiimote, in grado di illuminare i tag, facendo sì che il loro riflesso sia visibile al sensore IR. Questo richiede un circuito appropriato ed una struttura applicabile al controller senza troppo comprometterne peso e dimensioni. In base a queste considerazioni, si è deciso di sviluppare la seconda ipotesi.

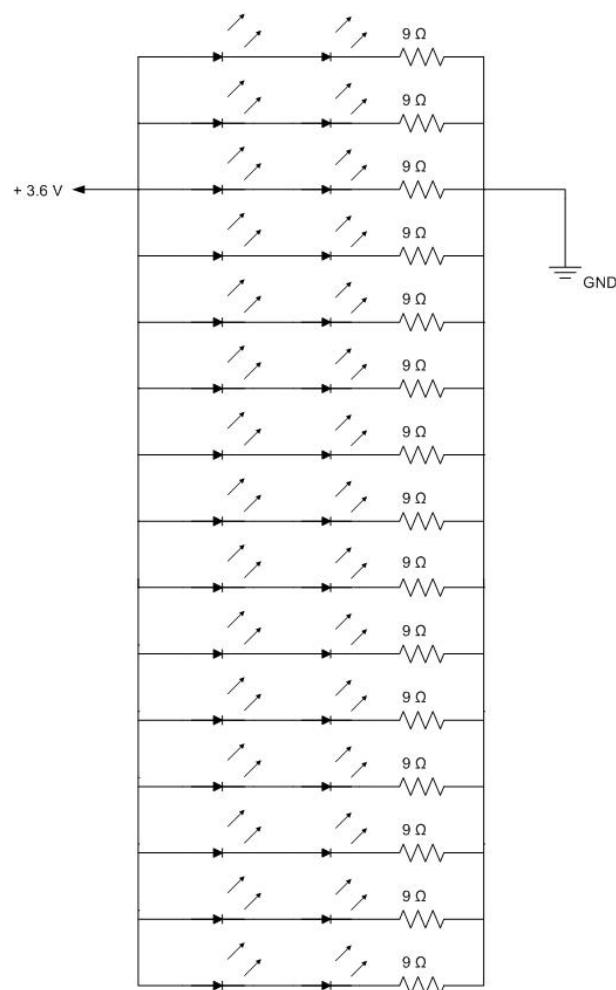


Figura 3.2: Schema del circuito.

3.1.1 Il circuito: analisi ed implementazione

Come sorgente di illuminazione si è progettato un circuito composto da led infrarossi Vishay TSAL6400¹, alimentato a batteria, da posizionare direttamente attorno alla testa del controller dove è situata la telecamera IR. L'obiettivo del circuito è quello di illuminare in modo adeguato i tag, mantenendo dimensioni e consumi ridotti. Il circuito è alimentato da 3 batterie standard AA ricaricabili da 1.2 V e 2400 mA/h poste in serie tra loro. Queste sono posizionate all'interno di un pack indipendente dal circuito e collegato ad esso tramite un cavo USB, in modo da poter essere tenuto in tasca senza così aggiungere peso e volume al Wiimote. Il circuito è composto da 30 led suddivisi in 15 rami in parallelo fra loro, con 2 led ciascuno più una resistenza. Su ogni ramo cade una tensione di 3.6 V e scorre una corrente di 100 mA. Il punto ottimale di lavoro dei led richiede 1.35 V e 100 mA. Ne consegue che il valore della resistenza posta su ogni ramo è di 9Ω (3.1 applicazione legge di Ohm)².

$$V_{tot} = 3.6 V$$

$$V_{led1} = 1.35 V$$

$$V_{led2} = 1.35 V$$

$$I_{led} = 100 mA$$

$$V_{resistenza} = V_{tot} - V_{led1} - V_{led2}$$

$$V_{resistenza} = 0.9 V$$

¹Per ulteriori informazioni riguardanti possibili circuiti ad infrarossi legati all'utilizzo del Wiimote[4]

²Per chiarimenti riguardante il funzionamento del circuito[7]

$$V = RI \quad R = \frac{V}{I} \quad R = \frac{0.9V}{0.1A} = 9\Omega \quad (3.1)$$

$$R_{resistenza} = 9 \Omega$$

Con questa configurazione la matrice di led richiede 1500 mA/h , per questo motivo sono state utilizzate della batteria da 2400 mA/h che garantiscono più di un ora di autonomia. Il circuito è progettato per essere posizionato direttamente sul corpo del controller. In questo modo il fascio luminoso non interferisce con il campo visivo della telecamera, nonostante quest'ultimo sia contenuto all'interno del primo (Figura 3.3).

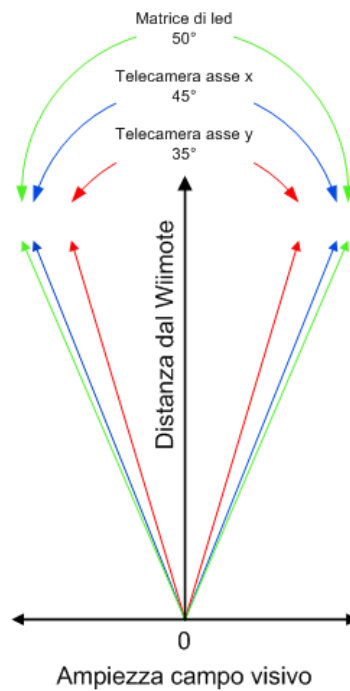


Figura 3.3: Relazione tra i campi visivi del Wiimote.

Per realizzare il circuito abbiamo utilizzato una scheda millefori di 6.5x10 cm, 30 led del modello sopra indicato, 15 resistenze da 10 Ω (il valore in commercio più vicino alle nostre necessità) e un vano porta batterie. Per il collegare il circuito all'alimentazione si è scelto un cavo USB, in quanto di facile reperibilità e d'immediato utilizzo. Delle 4 linee presenti nel cavo, ne vengono utilizzate 2, dato che non vengono trasferiti dati.

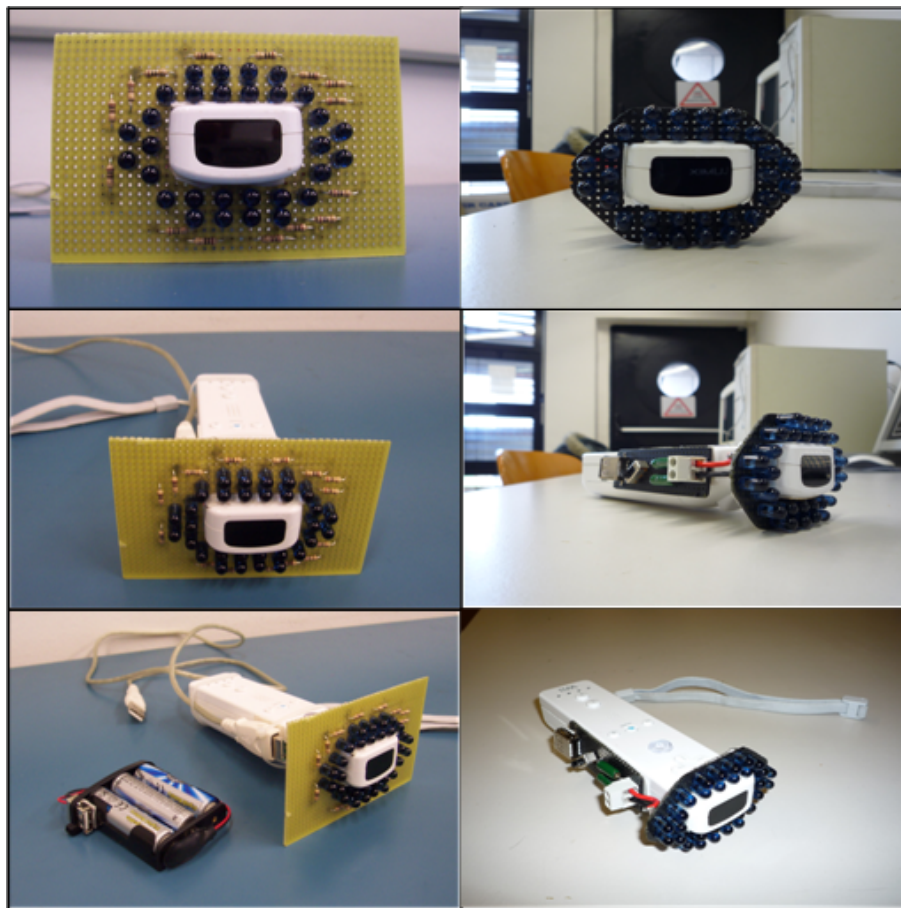


Figura 3.4: Varie viste dal Wiimote da noi modificato.

3.2 I tag

Per localizzarsi il nostro software deve riconoscere, tramite il sensore del Wiimote, i tag appositamente posizionati nell'ambiente e, in base ad essi, elaborare la possibile posizione dell'utente in relazione alla mappa inserita nell'elaboratore. Sono stati quindi ideati 4 modelli di tag univoci tra loro, con forma e dimensioni proporzionati alla fonte infrarossa utilizzata e all'ambiente in cui il sistema opera.

3.2.1 Forma e dimensione

Le peculiarità del sensore IR hanno imposto forti vincoli sulla geometria dei tag adottati. Come illustrato in precedenza, i fotogrammi acquisiti dal sensore sono elaborati dal processore integrato presente nel controller, che ne limita la visione a 4 punti. Questa caratteristica è stata vincolante, in quanto impedisce l'utilizzo di figure complesse, ma impone l'uso di combinazioni costituite per lo più da 4 punti. Anche la possibilità di utilizzare conformazioni di dimensioni diverse non è stata attuabile poiché le informazioni ricavate dal Wiimote sulla grandezza del tag (basate sul Formato Extended) non sono affidabili. Inoltre, non è stato possibile sviluppare combinazioni basate sulla geometria di punti, in quanto la forma elaborata del sensore varia in relazione alla visuale del controller. Un secondo problema nasce dal tipo di gestione con cui il processore acquisisce i punti per eseguirne il tracking. Infatti se nel campo visivo del Wiimote sono presenti più di 4 punti, il processore sceglierà in modo casuale i 4 da acquisire. Questo rende del tutto inaffidabile il rilevamento di forme tramite tag, in quanto la geometria acquisita potrebbe essere del tutto diversa rispetto a quella reale.

A fronte dell'analisi fatta si è deciso di utilizzare come discriminante per i marker il numero di punti presenti in ognuno di essi. I tag (Figura 3.4) sono cartoncini sui quali vengono applicati in linea retta quadrati di carta catarinfrangente. Per permettere al controller di rilevare il numero corretto di punti, questi devono essere posti a 4 cm di distanza fra loro (questo valore è stato estrapolato da uno studio sperimentale). Con le 4 tipologie di marker siamo in grado di marcare l'intero spazio e renderlo riconoscibile via software.

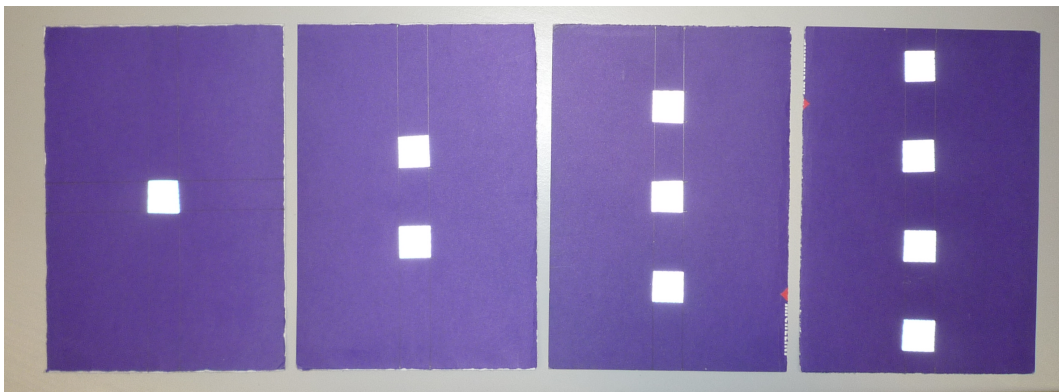


Figura 3.5: I quattro tag.

3.2.2 Posizionamento nell'ambiente

I marker sono stati realizzati per essere posizionati o sulle pareti o sul soffitto. Il sistema si comporta esattamente allo stesso modo indipendentemente dalla configurazione di posizionamento scelta, l'unica avvertenza è impugnare il Wiimote nel modo corretto. In questo progetto si è optato per la prima soluzione per due principali motivi. Il primo per praticità, poiché risulta più semplice e veloce posizionare i marker al muro piuttosto che al soffitto; la seconda è legata al fatto che l'utente, tenendo il Wiimote puntato verso l'alto potrebbe occultarne la visuale con capelli, maniche o altro.

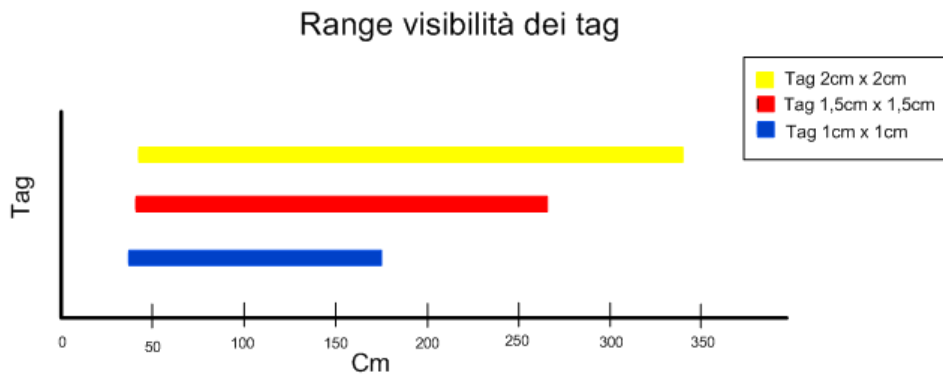


Figura 3.6: Range di visibilità dei tag in base alla dimensione dei quadrati di materiale catarifrangente.

Dimensione (cm)	Distanza min (cm)	Distanza max (cm)
1 x 1	37	175
1.5 x 1.5	41	264
2 x 2	42	340

Tabella 3.1: Tabella raffigurante il rapporto dimensione, distanza massima e distanza minima.

Prima di realizzare i tag è necessario valutare l'ambiente dove questi saranno utilizzati, poichè l'intervallo in cui il tag rimane visibile al controller dipende dalla dimensione del materiale catarifrangente. La dimensione del lato del quadrato è direttamente proporzionale alla massima distanza con cui è possibile utilizzare il controller e inversamente proporzionale alla minima distanza di utilizzo dello stesso. Di seguito è riportato il grafico con dei valori significativi (Figura 3.5). La misura adottata in questo studio è di 2 cm. Il tag è posto nell'ambiente in una posizione tale per cui la direzione del materiale catarifrangente risulti essere perpendicolare a quella del percorso. Questa disposizione fa sì che tutti i punti presenti sul marker entrino simultaneamente nel campo visivo della telecamera, evitando così errori dovuti ai tempi di acquisizione del sensore. Infatti ponendo i quadrati con un'altra disposizione, questi verrebbero acquisiti in tempi diversi poichè il sensore non li vedrebbe nello stesso istante. Tale logica rende il meccanismo di lettura dei tag più robusto e affidabile. Tra marker consecutivi è necessario lasciare una distanza minima, per evitare che questi siano simultaneamente acquisiti dal sensore, comportando così una lettura errata. Questa distanza è stata ricavata dalla seguente formula (3.2).

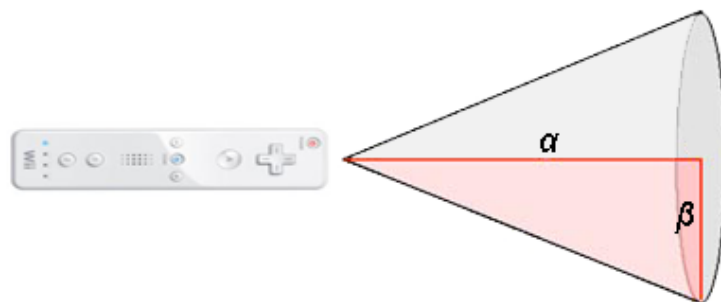


Figura 3.7: Campo visivo del Wiimote.

AngoloAsseX = 22.5°

Distanza α *cm*

RaggioCampoVisivo β *cm*

$$\beta = \sqrt{\left(\frac{\alpha}{\cos 22.5}\right)^2 - \alpha^2} = \beta \quad (3.2)$$

Per fare questa stima, è stato scelto l'asse X come asse di riferimento, poiché possiede un campo visivo maggiore e quindi necessita di una maggiore distanza tra i marker. Inoltre, va considerata la distanza di utilizzo tag Wiimote. Sapendo che questa distanza rappresenta un cateto di un ipotetico triangolo e, che l'angolo compreso tra cateto e ipotenusa corrisponde a metà campo visivo dell'asse di riferimento, applicando la formula si ricava il cateto restante. La distanza minima a cui verranno posti 2 tag consecutivi equivale al doppio del cateto trovato. Ad esempio con un distanza di utilizzo pari a 2 *m* i tag dovranno essere posti tra loro almeno a 166 *cm* (Tabella 3.2).

Distanza dal tag (<i>m</i>)	Distanza tra tag (<i>m</i>)
0.5	0.4
1	0.82
1.5	1.24
2	1.66
2.5	2.08
3	2.50
3.5	2.92

Tabella 3.2: Tabella raffigurante il rapporto tra la distanza tag-Wiimote e la distanza tra tag.

3.3 Ambiente

Il nostro sistema è soggetto a rumori e disturbi provocati dall'ambiente nel quale ci si trova ad operare. In questa sezione illustriamo le problematiche riscontrate.

3.3.1 Analisi

WhereWiiAre è stato pensato per essere usato in ambiente chiuso, con una superficie circoscritta a causa dei limiti imposti dalla tecnologia usata. Infatti il dongle bluetooth in uso ha un range operativo di 100m e quindi è consigliabile che sia posto al centro della struttura dove viene utilizzato. Questo lavoro si colloca in un ambiente domestico, dove non sono presenti conformazioni strutturali impervie che renderebbero non rappresentabile lo spazio di movimento sul pc, o spazi troppo grandi, dove i nostri dispositivi potrebbero avere problemi di portata.

3.3.2 Disturbi

La telecamera è sensibile, come ormai largamente spiegato, a radiazioni di tipo infrarosso. Queste radiazioni sono emesse non solo da fonti dedicate, come i led da noi utilizzati, ma anche dal sole, da candele e da altre fonti luminose convenzionali presenti nella nostra vita quotidiana. Di conseguenza, queste come le altre, possono essere rilevate dal Wiimote e indurre errate rilevazioni. Ad esempio è possibile sostituire la Sensor Bar della Nintendo Wii con 2 candele posizionate sotto il televisore, senza che questo influisca sul funzionamento della console. Un altro problema è quello dovuto a superfici riflettenti come finestre o specchi normalmente presenti in ogni abitazione. Queste superfici riflettono sempre un unico punto e, grazie a questa particolarità è possibile aggirare il problema, evitando di posizionare marker con

un singolo quadrato di materiale catarifrangente dopo tale superficie, poiché potrebbe falsarne la lettura. Purtroppo, oltre a questi oggetti, ve ne sono altri che riflettono in maniera diversa creando letture imprevedibili, come ad esempio superfici lucide. Nel caso fossero presenti, l'unica soluzione adottabile è quella di evitare che queste entrino nel raggio d'azione del controller oppure studiarne il comportamento effettivo.

3.3.3 Limiti

La struttura dell'ambiente può limitare il funzionamento del nostro sistema. Sia la comunicazione Wiimote pc che pc auricolare avviene tramite bluetooth. Il dongle che gestisce questo tipo di tecnologia wireless è progettato per funzionare con un range d'azione di circa 100 metri in ambiente aperto. Logicamente all'interno di un edificio il campo di azione diminuisce drasticamente, questo a causa della presenza di pareti e ostacoli che interferiscono riducendo il range di operatività dei dispositivi. In modo particolare muri in cemento armato e strutture in ferro limitano lo spazio di utilizzo dell'applicazione. Quindi è importante analizzare la struttura dell'ambiente e situare l'elaboratore in un punto centrale rispetto allo spazio di lavoro.

Capitolo 4

Il sistema di localizzazione: WhereWiiAre

4.1 L'idea

Dopo avere analizzato a fondo il Wiimote, si è deciso di sviluppare un sistema di localizzazione intuitivo e basso costo, ma allo stesso tempo robusto ed efficace. L'obiettivo è quello di guidare l'utente all'interno dell'ambiente tramite segnali acustici, col fine di raggiungere goal prefissati.

4.1.1 Che cosa deve fare

Il software deve permettere all'utente di creare una propria mappa in base alle specifiche esigenze, al fine di rappresentare nel migliore dei modi l'ambiente reale di utilizzo. Oltre a questo, il programma deve garantire la completa personalizzazione della mappa, attraverso l'aggiunta di goal, suoni e indicazioni. Una volta avviata l'applicazione, l'utente potrà scegliere attraverso l'uso del controller la destinazione da raggiungere, selezionandola tra i goal presenti.

A questo punto il sistema calcolerà il percorso migliore e guiderà l'utente a destinazione attraverso indicazioni sonore.

4.1.2 A chi è destinato

L'applicazione è dedicata a persone con ridotte capacità visive che necessitano di aiuto per muoversi all'interno della propria abitazione. Il controller, grazie alla sua forma simmetrica, può essere utilizzato indifferentemente da utenti destri o mancini. Inoltre presenta un numero di tasti ridotto, posti in posizioni chiare e difficilmente confondibili. Nel caso si abbia difficoltà a utilizzare i tasti, è possibile interagire con l'applicazione attraverso i movimenti della mano, che verranno letti dall'accelerometro.

4.2 Struttura e Funzionamento

Qui viene mostrata la struttura e il funzionamento del software creato, diviso per sezioni principali, illustrando scelte e ragionamenti utilizzati. Si rimanda al manuale utente per una spiegazione dettagliata su come utilizzare il sistema.

4.2.1 Connessione dispositivo

Per gestire la connessione bluetooth è stata utilizzata la libreria Bluecove¹. Questa libreria permette di interfacciarsi con innumerevoli piattaforme quali Mac OS X, WIDCOMM, BlueSoleil e Microsoft Bluetooth stack se si utilizza Windows XP o Windows Vista, oppure WIDCOMM e Microsoft Bluetooth stack se si utilizza Windows Mobile. In questo modo l'applicazione potrà essere utilizzata su qualsiasi piattaforma, indipendentemente dall'hardware usato (Figura 4.1). Si consiglia di utilizzare un dongle bluetooth basato su driver WIDCOMM poiché risultano più stabili e completi rispetto all'alternativa BlueSoleil.

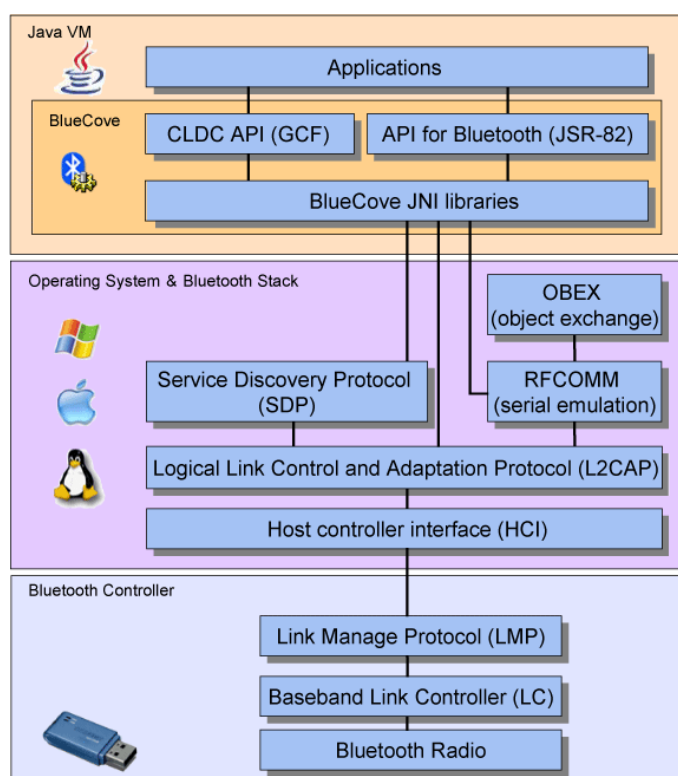


Figura 4.1: Bluecove stack-diagram.

¹Per tutte le informazioni riguardanti la libreria Bluecove[1]

Una volta iniziata la ricerca di dispositivi bluetooth, verrà inizializzato un ascoltatore che resterà in attesa di un device che abbia le seguenti caratteristiche :

Name Nintendo RVL-CNT-01

Vendor ID 0x057e

Product ID 0x0306

Major Device Class 1280

Minor Device Class 4

Service Class 0

Summary of all Class Values 0x002504

Dopo aver posto il controller in discovery mode (cioé visibile all'applicazione), il software effettuerà il pairing, instaurando così una connessione tra i due dispositivi senza bisogno di nessuna autenticazione. Se tutti i parametri sono rispettati, al controller verrà associata una variabile attraverso la quale sarà gestito. Infine viene settato il Report Mode, cioè il formato con cui pc e Wiimote si scambieranno i dati. In questo caso la modalità scelta è la 0x33, in quanto permette l'uso simultaneo di pulsanti, accelerometro e telecamera IR con formato EXTENDED.



Figura 4.2: Schermata con Wiimote connesso.

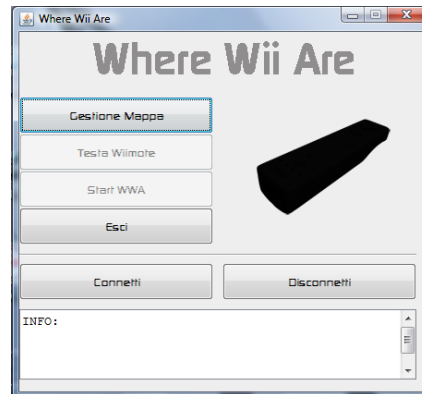


Figura 4.3: Schermata con Wiimote disconnesso.

A questo punto attraverso l'oggetto creato si ha il pieno controllo del Wiimote. All'utente apparirà la seguente finestra (Figura 4.2 e 4.3) dove, dopo aver messo il Wiimote in discovery mode, potrà connetterlo al software. Nella finestra di dialogo è possibile osservare lo stato della connessione e l'indirizzo MAC associato al controller.

4.2.2 Test dispositivo

La sezione test Wiimote è stata creata per permettere di testare il controller in tutte le sue parti. In questo modo è possibile visualizzare il corretto funzionamento della telecamera IR, dell'accelerometro, dei led e della vibrazione. Per gestire i vari componenti del controller vengono creati tre ascoltatori, attraverso i quali è possibile accedere e gestire le informazioni necessarie. A video è possibile vedere un quadratino colorato per ogni punto visibile dal controller, che seguirà i movimenti della fonte infrarossa a cui è associato. Si ricorda che questi movimenti sono tracciabili unicamente all'interno della visuale del Wiimote, quindi all'interno di una finestra di 1024 x 768 pixel. Inoltre è possibile attivare la vibrazione e l'accensione dei led attraverso i check button presenti. Per osservare le accelerazioni a cui è soggetto il controller sono visualizzate tre barre sulle quali è possibile osservare i cambiamenti in tempo reale.



Figura 4.4: Ambiente test Wiimote.

4.2.3 Gestione mappa

Questa sezione permette la creazione e la modellizzazione della mappa raffigurante l'ambiente di utilizzo. Sullo sfondo è possibile porre una planimetria della propria abitazione per rendere più reale possibile il modello. La struttura si basa su una matrice di grandezza proporzionale alla planimetria inserita, composta da oggetti che possono assumere tre tipologie di riempimento: vuoto, strada e goal. Per fare ciò è stata creata una classe blocco, nel quale vengono salvate tutte le informazioni riguardanti l'oggetto, come ad esempio il tipo di riempimento, la posizione, l'audio associato e il relativo tag posto nell'ambiente reale. Per aiutare l'utente durante la guida, è possibile associare ad ogni cella un clip audio, scegliendolo dalla lista presente o registrandolo al momento.

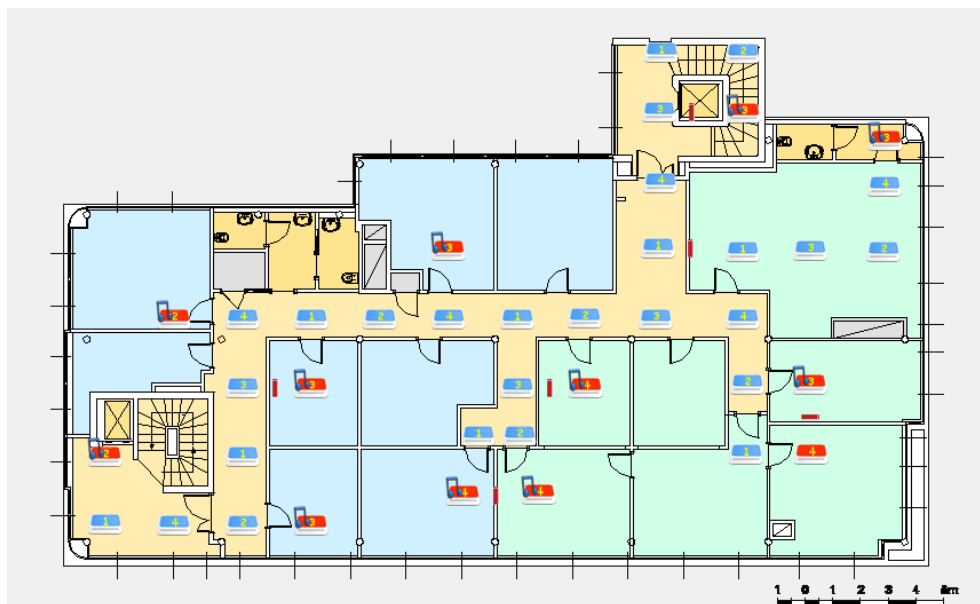


Figura 4.5: Strada taggata correttamente.

La gestione dell'audio è stata realizzata attraverso l'uso della libreria JLayer² che permette la selezione e l'ascolto di audio in diversi formati. Una volta completata la mappa, è possibile memorizzarla su file. Il software creerà due strutture dati che memorizzeranno, una il modello dell'ambiente, l'altra la mappa relativa all'audio. La principale difficoltà riscontrata in questa parte di applicazione, è stata realizzare un algoritmo per la predisposizione automatica dei tag nell'ambiente. Ad ogni blocco è associato un marker e, al fine di ottenere percorsi validi, bisogna far sì che ogni tag sia posto con coerenza e non generi ambiguità sul percorso che si sta seguendo, indipendentemente dalla direzione scelta (Figura 4.5). Nella Figura 4.6 viene mostrato un esempio di errore.

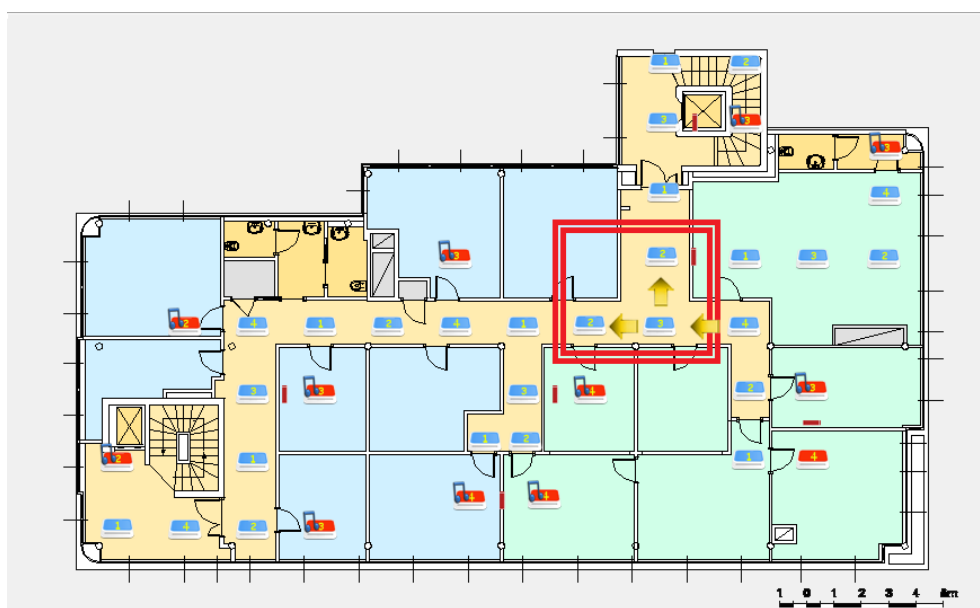


Figura 4.6: Strada taggata in modo errato.

²Per informazioni riguardanti la libreria JLayer[2]

Come è possibile notare nel rettangolo, una volta arrivati sul blocco con tag 3, sia che si giri a destra, sia che si vada dritti (ipotizzando di arrivare dal marker 4) si incontrerà in qualunque caso il tag 2. In realtà, l'utente che direzione avrà preso? In questo modo è impossibile determinarlo, a meno di aspettare la prossima mossa, a fronte però di un software poco efficiente e robusto. Per evitare questo bisogna predisporre in modo adeguato i marker nell'ambiente. Prima di associare un tag a un blocco, bisogna verificare la presenza e il tipo di tag dei blocchi adiacenti. L'algoritmo controlla i marker associati alle celle circostanti, li memorizza e in base ai valori riscontrati sceglie, se possibile, il tag ottimo da associare al blocco in esame. Il controllo non si limita ad analizzare solo le celle adiacenti a quella in analisi ma, controlla tutte i blocchi mostrati in figura 4.7. A questo punto è possibile posizionare i marker nello spazio. A causa del numero ridotto di marker a disposizione, non è possibile gestire un incrocio a 4 vie, poiché un tag sarebbe ripetuto ricadendo così nell'errore spiegato in precedenza.



Figura 4.7: Blocchi sottoposti al controllo.

In figura 4.8 è mostrata l'interfaccia grafica che si occupa di questa gestione. Sulla sinistra dello schermo è possibile selezionare il tipo

di riempimento e completare la mappa semplicemente cliccando sul percorso desiderato. Inserito un blocco è possibile, cliccando sui lati, inserire un “muro” che blocca la navigazione verso quella direzione. Una volta conclusa la modellizzazione dell’ambiente, cliccando sul comando “Elabora”, verranno adeguatamente disposti i tag sul percorso. A questo punto è possibile aggiungere i goal e i clip audio desiderati, sempre utilizzando il menù posto sulla sinistra dello schermo. In questa distribuzione sono presenti alcuni clip audio predefiniti, ma niente vieta all’utente di aggiungerne di nuovi. Completata la stesura del modello, cliccando su “salva”, la mappa viene memorizzata in tutte le sue parti. Sempre in questa sezione, è possibile modificare una mappa già esistente utilizzando il tasto “carica”.

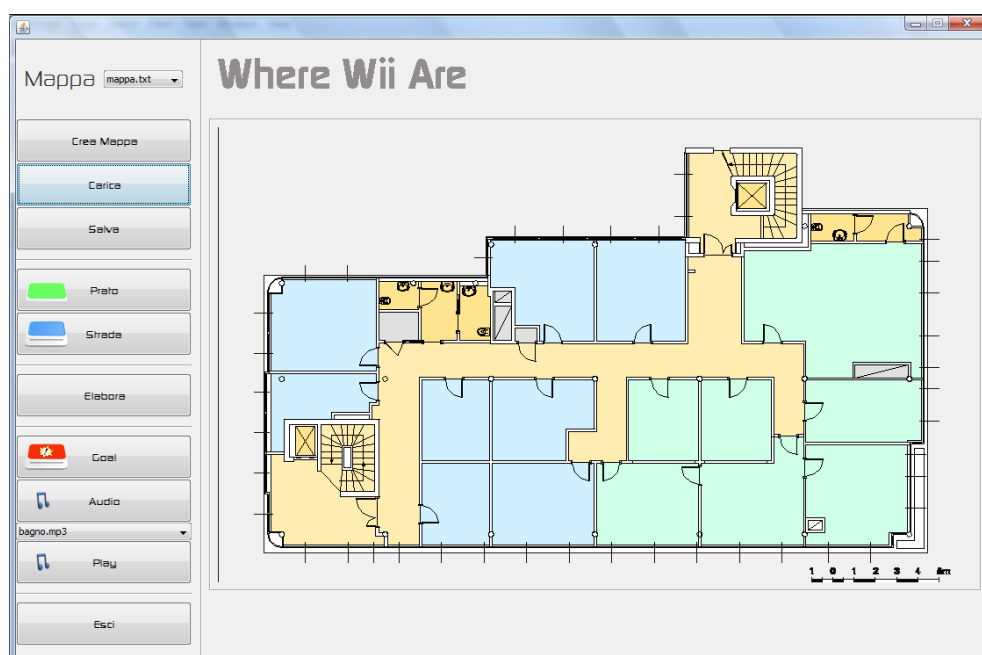


Figura 4.8: Ambiente gestione mappa.

4.2.4 Gestione Localizzazione

Una volta creata ed elaborata la mappa è possibile avviare la modalità di guida. Anche qui per gestire il controller sono presenti diversi thread che si occupano della gestione e della sincronizzazione dei componenti del Wiimote. Un primo thread controlla i movimenti letti e i tasti premuti sul controller. Per gestire il controllo dell'applicazione tramite movimenti del Wiimote, vengono monitorate la forza e il verso a cui è soggetto il controller sull'asse X. Nel momento in cui il valore letto supererà la soglia limite (+7 *N* sinistra, -7 *N* destra), il thread si preoccuperà di gestire l'evento e invocare i metodi associati. Viene utilizzata una soglia per impedire che movimenti naturali e minimi vengano interpretati come comandi. Si è inoltre implementata la possibilità di invocare gli stessi metodi attraverso l'uso del Wiimote. Per permettere la lettura corretta degli input del controller, a fronte di un evento rilevato, il thread viene "congelato" per un 1.5 secondi per poi riprendere il normale funzionamento. Questo perché ogni ascoltatore che gestisce gli eventi del Wiimote rileva lo stato di attivo per tutto il tempo in cui l'evento è verificato, cosa che rende difficile il controllo via software. Per ovviare a questo problema si è deciso di temporizzare la lettura degli eventi del controller, impostando un tempo di sleep al thread. Per determinare il numero di fonti infrarosse presenti nel campo del Wiimote, viene utilizzata una variabile che viene incrementata ogni qual volta un punto entri nel campo visivo, e decrementata nel momento in cui la fonte luminosa esca dalla sua visuale. Questo controllo è gestito da un secondo thread. Per ottenere una lettura corretta e robusta, indipendentemente dal tempo di esposizione del marker alla telecamera IR, si è deciso di temporizzare la lettura del numero di quadrati nel tag dividendola in due acquisizioni.

Il marker sarà ritenuto valido, se solo se, il numero di punti letto nelle due acquisizioni sarà il medesimo. Il valore di default di questa temporizzazione è di 60 *ms*, ma può essere in qualunque momento personalizzato dall'utente.

A un terzo thread è affidato il compito di gestire l'aspetto audio, i vari menù di selezione e le indicazioni utilizzate durante la guida. Altro aspetto fondamentale di questa sezione, è il calcolo del cammino ottimo tra 2 goal definiti. Per farè ciò si è scelto di implemetare, tra i vari algoritmi analizzati, A*³ poichè è completo e ottimamente efficiente. A* è un algoritmo path-finding il cui scopo è quello di trovare il percorso più breve tra due punti. L'algoritmo esamina ripetutamente il blocco inesplorata più promettente disponibile ad ogni passo. Se la locazione esplorata è il goal A* termina, altrimenti l'algoritmo esamina le locazioni vicine per la prossima esplorazione. Più precisamente, A* tiene traccia di due liste: OPEN per le celle esplorate e CLOSED per le celle

³Per informazioni riguardanti l'algoritmo[3]

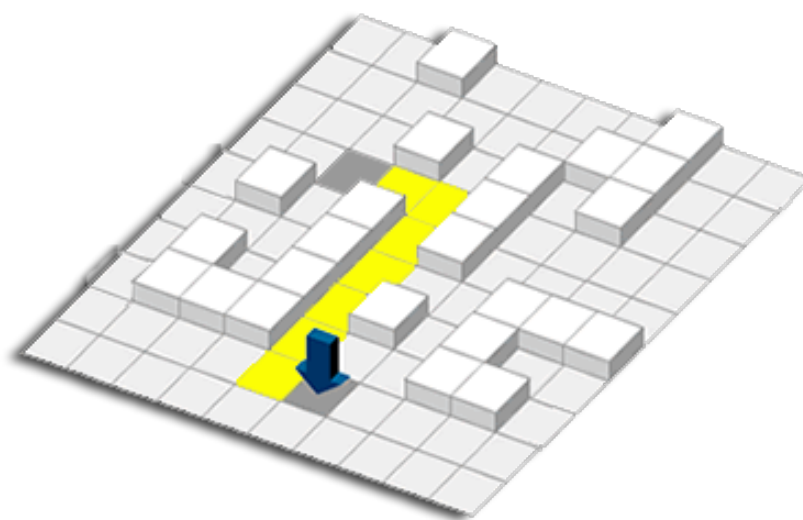


Figura 4.9: Esempio di percorso calcolabile con A*.

inesplorate. Inizialmente la lista CLOSED è vuota e la lista OPEN contiene solo il punto di partenza. Ad ogni iterazione l'algoritmo rimuove lo stato più promettente della lista OPEN. Se questo è il goal l'algoritmo termina, altrimenti vengono esaminati i suoi vicini: se non sono stati ancora incontrati vengono posti nella lista OPEN, altrimenti le loro informazioni vengono aggiornate (solo se il percorso risulti essere migliore). Le celle presenti in CLOSED sono ignorate se il loro costo è maggiore di quello corrente, altrimenti vengono spostate nella lista OPEN. Se la lista OPEN risulta vuota significa che non esiste nessun percorso che colleghi partenza a destinazione. Ogni stato X include le seguenti informazioni per determinare il cammino migliore: il costo del percorso che collega l'inizio a X , la stima euristica della distanza tra X e il goal e infine il costo totale che è la somma di questi due. Inoltre ogni stato tiene un riferimento dello stato precedente; quando il goal viene raggiunto questi riferimenti sono usati per tracciare la strada tra il goal e la partenza. L'idea che sta alla base dell'euristica è quella di stimare il costo reale tra un dato nodo e il goal. E' determinante scegliere una buona funzione euristica. Se si conosce sempre il reale costo per raggiungere il goal, A^* seguirà solamente la strada migliore senza espandere gli altri nodi, non perdendo così tempo in ricerca. Se invece l'euristica è sovrastimata rispetto al costo reale, questa diventa inammissibile e l'algoritmo potrebbe non trovare il percorso ottimo, a fronte però di una maggior velocità di calcolo. Un modo per garantire che il costo dell'euristica non sia sovrastimato è quello di calcolare la distanza euclidea tra il nodo e il goal. Grazie all'utilizzo dell'euristica A^* non trova semplicemente un percorso, ma trova il percorso ottimo.

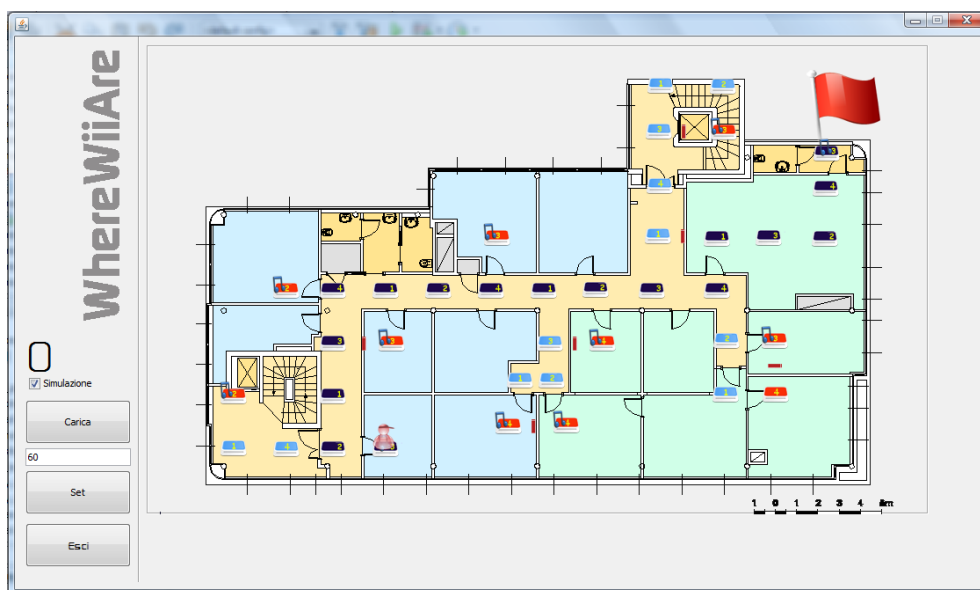


Figura 4.10: Interfaccia del sistema WhereWiiAre in modalita guida.

WhereWiiAre utilizza come euristica la Manhattan distance:

$$MANHATTAN = \xi(|X - GOALX| + |Y - GOALY|) \quad (4.1)$$

ξ rappresenta il minimo costo di spostamento tra una cella e quella adiacente, per noi quindi sempre uguale ad 1 in quanto non c'è alcuna differenza di costo tra le varie direzioni. L'implementazione software segue questa descrizione adeguatamente modellata sulla struttura della mappa utilizzata. Una volta trovato il percorso ottimo, l'utente viene guidato al fine di raggiungere il goal prestabilito. Nel caso in cui, per un qualsiasi motivo, il soggetto commetta un errore e segua quindi la direzione sbagliata, l'algoritmo ricalcherà il percorso minimo dalla posizione attuale alla meta prefissata. Per poter dare indicazioni corrette durante la guida, è necessario memorizzare la direzione dell'utente e, aggiornarla ogni qual volta questa venga modificata. L'interfaccia che si occupa di questa sezione, mostra a video la mappa in uso, dando al possibilità di poter osservare in qualunque momento la posizione e i

movimenti dell'utente. Una volta selezionati il punto di partenza e di arrivo, su di essi verranno posti due icone e, a video verrà evidenziato il percorso calcolato. L'animazione sullo schermo segue l'utente in ogni suo spostamento, permettendo un controllo a real time della posizione.

4.2.5 Utilizzo del Wiimote

Per un corretto funzionamento del sistema WhereWiiAre è necessario utilizzare il controller con semplici accorgimenti. L'obiettivo è quello di permettere al Wiimote di acquisire in modo semplice e corretto il numero di quadratini catarifrangenti presenti nel tag. È importante valutare la distanza media di utilizzo e la velocità di spostamento dell'utente. Attualmente è utilizzata una frequenza di lettura dei tag standard pari a 60 *ms*, che permette una corretta acquisizione, ma in ogni caso è personalizzabile dell'utente secondo le proprie caratteristiche. Regola fondamentale per il funzionamento del sistema, è quella di impugnare il controller rivolto sempre in direzione dei tag e, tenerlo alla stessa altezza del marker posto sul muro. Logicamente non è necessaria una precisione elevata, in quanto pensato per persone non vedenti. Si raccomanda dimensionare i tag in base all'ambiente in cui verranno utilizzati.

Capitolo 5

Scenario di utilizzo

5.1 Una simulazione

Di seguito viene descritto un normale utilizzo dell'applicazione. Avviato il programma è possibile accedere alla sezione riguardante la gestione della mappa. Qui è possibile creare e personalizzare la struttura attraverso tre semplici passaggi. Per prima cosa si effettua la stesura del percorso (Figura 5.1) sul quale ci si vorrà spostare. Fatto questo si passa all'elaborazione della struttura, che consiste nell'assegnare in modo automatico le tag ad ogni blocco (Figura 5.2). Infine è possibile personalizzare ulteriormente la mappa aggiungendo goal e clip audio. A questo punto è possibile disporre i marker nell'ambiente seguendo

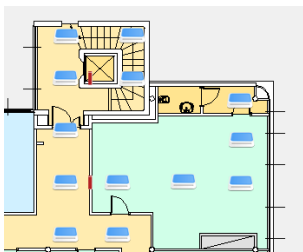


Figura 5.1: Mappa con strada.

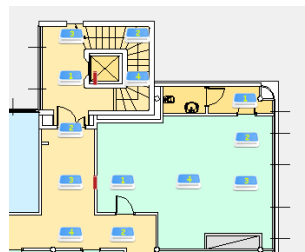


Figura 5.2: Mappa con strada elaborata.

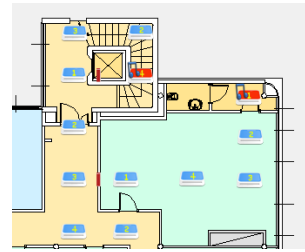


Figura 5.3: Mappa finale con goal e audio.

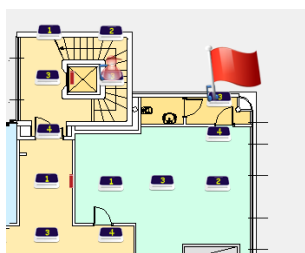


Figura 5.4: Utente al punto di partenza.

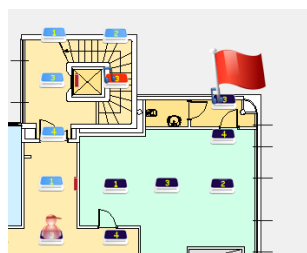


Figura 5.5: Utente a metà percorso.

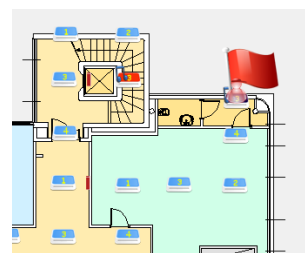


Figura 5.6: Utente al punto di arrivo.

lo schema elaborato precedentemente. Per passare in modalità guida è necessario connettere il Wiimote al pc. Fatto questo e caricata la mappa desiderata, è possibile iniziare ad utilizzare il sistema di navigazione. Premendo il pulsante **1** una voce permetterà all'utente di selezionare la posizione corrente e la meta da raggiungere. Al termine di questa operazione a video un "omino" indicherà la posizione attuale, mentre una bandiera verrà posta in posizione di arrivo (Figura 5.4). Il percorso ottimo calcolato verrà evidenziato sullo schermo. Durante la navigazione l'omino sullo schermo mostrerà i movimenti eseguiti dall'utente (Figura 5.5, 5.6). Durante la navigazione l'utilizzatore sarà assistito da una voce attraverso l'auricolare bluetooth. Nel caso in cui l'utente sbagli direzione, il percorso verrà ricalcolato automaticamente e la voce guida continuerà ad assistere il soggetto fino al raggiungimento della meta.

5.2 Possibili utilizzi

Il sistema sviluppato può essere utilizzato in diversi domini applicativi, mantenendo invariato il suo funzionamento di base, ma andando a personalizzare le azioni associate agli eventi in questione. Questo è reso possibile dalle innumerevoli combinazioni di azioni che possono essere associate agli eventi del dispositivo. In questo modo, allo stesso input, possono corrispondere output diversi in base alla posizione individuata dal software. Di seguito vengono illustrati possibili utilizzi del Wiimote.

5.2.1 Domotica

La domotica¹ è la scienza interdisciplinare che si occupa dello studio delle tecnologie atte a migliorare la qualità della vita nella casa. Il nostro sistema nasce in quest'ambito come guida per persone non vedenti. Sfruttando sempre lo stesso dispositivo è possibile realizzare ad esempio un telecomando universale per gestire dispositivi elettronici, utilizzando relè comandati attraverso canali USB dal pc. L'utente puntando il Wiimote in direzione dell'apparecchio desiderato (taggato in precedenza) e dando uno specifico input (come ad esempio un tasto o un movimento), permette al sistema di riconoscere l'evento e gestire le relative operazioni associate. Questo rende l'uso di diversi dispositivi semplice e divertente. Inoltre, la gestione di apparecchi domestici potrebbe essere associata al sistema di localizzazione. In questo modo si potrebbe gestire automaticamente l'apertura di porte o finestre incontrate durante la navigazione.

¹Per informazioni riguardanti questa disciplina[11]

5.2.2 Robotica

Uno dei problemi principali della robotica mobile è quello della localizzazione. Per risolvere parzialmente questo problema è possibile sfruttare WhereWiiAre. Il sistema, infatti, può essere utilizzato per guidare un robot nell'ambiente, nello stesso modo con cui viene guidato un soggetto umano. Il software, anziché inviare segnali sonori all'auricolare, invia l'informazione riguardante la posizione al controllore del comportamento del robot che, in base a questi dati, sceglierà i movimenti da compiere. Per far ciò è necessario posizionare in modo corretto il Wiimote, ed è possibile scegliere se installare direttamente l'applicazione sull'elaboratore del robot, o utilizzare un computer esterno e connetterlo via wireless. Oggi all'interno di grossi impianti di stoccaggio si usano robot che si occupano dell'immagazzinamento delle merci in modo automatico. Il nostro sistema potrebbe essere un'alternativa al posizionamento di binari per lo spostamento e la guida del robot all'interno del magazzino.



Figura 5.7: Robot per lo stoccaggio di merci all'interno di un magazzino.

5.2.3 Intrattenimento

Il Wiimote è stato progettato per giocare utilizzando la console Nintendo Wii. Anche il nostro software può permettere la realizzazione di giochi basati unicamente su dialogo computer controller. Sfruttando sempre l'utilizzo di tag è possibile discriminare ogni giocatore assegnandogli un marker diverso. Questo permette l'identificazione del player nel momento in cui entri nel campo visivo di un altro partecipante. In funzione di ciò è possibile creare diverse tipologie di giochi, come ad esempio un gioco stile "sparatutto", o semplicemente emulare gli stessi giochi della console. Sviluppando determinati script è inoltre possibile sostituire i comuni dispositivi di input come mouse, joystick e tastiera con il Wiimote. Altro possibile utilizzo è quello di adoperare il dispositivo in musei e altre strutture dove è richiesta una voce guida, permettendo così a chiunque una visita autonoma.

Capitolo 6

Conclusioni e sviluppi futuri

6.1 Sviluppi futuri

In questa sezione si analizzano possibili sviluppi futuri legati al potenziamento del controller e dell'applicazione.

6.1.1 Potenziamento Wiimote

La casa produttrice Nintendo ha reso noto che a breve verrà messo in commercio un nuovo dispositivo. L'accessorio Wii MotionPlus¹(Figura 6.1), ridefinisce i controlli di gioco grazie all'immediata e fedele riproduzione dei movimenti in uno spazio tridimensionale. Da connettere alla base del telecomando Wii, il Wii MotionPlus si unisce alla tecnologia dell'accelerometro e della barra sensore, permettendo di registrare con accuratezza la posizione e l'orientamento del braccio del giocatore, offrendo un livello di precisione elevatissimo. Ogni minimo movimento del polso o del braccio del giocatore è automaticamente riprodotto in tempo reale sullo schermo, garantendo una risposta 1:1 effettiva del gioco. Con queste nuove capacità, si potrebbe implementare un soft-

¹Per tutte le informazioni riguardanti gli accessori Nintendo Wii[6]

ware in grado di gestire con maggior precisione i movimenti compiuti dall'utente, ampliando così i possibili scenari di utilizzo dell'applicazione. Questo permetterà di identificare e riconoscere in maniera precisa vari movimenti, come ad esempio il cammino, la corsa e il salto...

Queste informazioni non sono ricavabili dall'accelerometro attuale, poiché quest'ultimo non è sufficientemente preciso. Un altro possibile miglioramento è la sostituzione della telecamera IR con una di maggiore qualità, per avere così una risoluzione migliore. Questa modifica permetterebbe di sfruttare le informazioni del formato EXTENDED e FULL in modo ottimale. Andando inoltre ad incrementare il numero di punti identificati durante il tracking, sarebbe possibile utilizzare un alfabeto di tag più complesso, non imponendo più vincoli alla rappresentazione di qualsiasi tipo di mappa, permettendo così l'utilizzo di incroci ad N vie. Se verranno implementate queste modifiche, WhereWiiAre risulterebbe molto più preciso e potente e, il suo utilizzo potrebbe essere esteso in ambiti più complessi. Inoltre, se la teleca-



Figura 6.1: Il Wii Motionplus.

mera presente all'interno del controller trasmettesse l'immagine vista e non il numero di punti individuati, si potrebbero apportare i miglioramenti elencati senza alterare la struttura hardware, ma lavorando sulla decodifica software.

6.2 Valutazioni conclusive

Lo scopo del progetto è quello di sviluppare un'applicazione a basso costo, intuitiva e affidabile, capace di localizzare un soggetto all'interno di un ambiente strutturato attraverso l'utilizzo del dispositivo Wiimote. WhereWiiAre raggiunge tali obiettivi integrando al suo interno, oltre a un sistema di localizzazione, un sistema di guida ideato per persone non vedenti. L'utilizzo del Wiimote come dispositivo principale, ha permesso di poter acquisire i dati necessari alla localizzazione senza utilizzare componenti elettronici aggiuntivi, avendo tutto il necessario in un unico dispositivo portatile e semplice. Gli obiettivi sono stati raggiunti grazie alla realizzazione di un'applicazione funzionale, intuitiva e di immediato utilizzo.

Bibliografia

- [1] Bluecove. Bluecove a java library for bluetooth.
- [2] JLayer. Jlayer - mp3 library for the java platform.
- [3] Jean-Claude Latombe. *Robot motion planning*. Springer, 1991.
- [4] Johnny Chung Lee. Johnny chung lee projects wii.
- [5] Motej. Motej - a wiimote library for java.
- [6] Nintendo. The wii motionplus.
- [7] Richard T. Corf Ralph J. Smith. *Circuit, device and system*. Hardcover, 1991.
- [8] Wiibrew. Accelerometr.
- [9] Wiibrew. Bluetooth communication.
- [10] Wiibrew. Ir camera.
- [11] Wikipedia. La domotica.
- [12] Wikipedia. Light-emitting diode.

Appendice A

Il manuale utente

In questo manuale è illustrato in modo semplice e chiaro il funzionamento del sistema WhereWiiAre in ogni sua parte, spiegando come utilizzare le funzioni di localizzazione e guida all'interno dell'ambiente scelto. Questo manuale è suddiviso in 4 sezioni.

WhereWiiAre: in questo paragrafo è illustrato come connettere il Wiimote al sistema e, come accedere a tutte le varie parti dell'applicazione.

Testa Wiimote: in questa sezione è spiegato come verificare tramite il software se il controller funziona correttamente.

Gestione mappa: in questa parte si spiega come inserire, personalizzare, modificare e salvare la mappa rappresentante l'ambiente dove verrà eseguita la navigazione.

Start WWA: In questo paragrafo è spiegato l'utilizzo del sistema di localizzazione e guida, illustrando le modalità con cui l'utente deve utilizzare il Wiimote.

A.1 WhereWiiAre

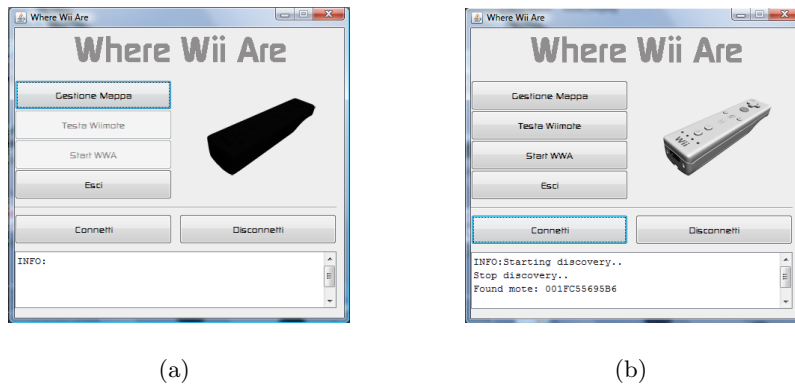


Figura A.1: Le due immagini rappresentano la stessa schermata, la prima senza Wiimote connesso (a), la seconda dopo aver connesso il dispositivo (b).

All'avvio del programma appare una schermata (Figura A.1) con le seguenti opzioni:

- Gestione mappa
- Testa Wiimote
- Start WWA
- Esci
- Connetti
- Disconnetti

Le prime tre opzioni permettono all'utente di accedere alle relative sezioni spiegate successivamente. I tasti connetti e disconnetti sono utilizzati per collegare o scollegare via Bluetooth il controller al pc. Si ricorda che prima di attivare la connessione via software è necessario premere simultaneamente i tasti **1** e **2** sul Wiimote. Se l'operazione è stata svolta correttamente i led del controller inizieranno a lampeggiare.

Una volta connesso il Wiimote, la sagoma nera, verrà sostituita con un'immagine raffigurante il dispositivo e nel campo informazioni sarà disponibile l'indirizzo MAC relativo al controller connesso. Si ricorda che è possibile accedere alla sezione Testa Wiimote e Start WWA solo se il controller è collegato al pc.

A.2 Testa Wiimote

Una volta cliccato sul tasto “Testa Wiimote” nella schermata principale, apparirà una finestra come quella mostrata in figura A.2. In questa sezione è possibile prendere confidenza col controller e verificare che tutti i sensori funzionino correttamente.



Figura A.2: Ambiente testa Wiimote.

Per testare il sensore IR è sufficiente puntare, con la matrice di led attiva, un tag ad una distanza adeguata e, se tutto funziona correttamente, sullo schermo apparirà un quadratino per ognuno di quelli rappresentati sul marker. Inoltre i punti visualizzati seguiranno i movimenti da noi eseguiti. Vengono inoltre mostrate le accelerazione impresse sugli assi del Wiimote attraverso tre indicatori a barra posti sulla sinistra dello schermo. Infine grazie alla presenza di 5 check box è possibile attivare e disattivare led e vibrazione del controller.

A.3 Gestione mappa

Questa sezione è stata sviluppata per permettere la creazione di una mappa, completamente personalizzata, nella quale muoversi grazie al WhereWiiAre. Sulla sinistra dello schermo, sono presenti quattro gruppi di tasti che permettono di compiere le operazioni desiderate. Il tasto “Crea mappa” permette di inserire una nuova mappa. Una volta creata è possibile personalizzarla coi seguenti tasti:

Prato: selezionando questo tasto è possibile inserire nella mappa una zona dove si prevede di non voler passare, quindi dove non è necessario inserire tag.

Strada: selezionando questo tasto è possibile inserire nella mappa un percorso sul quale l'utente potrà muoversi. E' inoltre presente per ogni blocco la possibilità di negare l'accesso verso una data direzione, semplicemente cliccando sul lato in questione. In questo modo a video viene visualizzato un muro, che non permette l'accesso al blocco di strada adiacente.

Una volta create le strade sulle quali l'utente vorrà muoversi, premendo il tasto “elabora”, il programma associerà ad ogni blocco strada un

numero, che indica il tipo di tag da collocare nell'ambiente. Marcata la mappa è possibile associare goal e audio. Per posizionare i goal è sufficiente cliccare prima sul tasto goal e poi sul blocco desiderato. Lo stesso procedimento sarà usato per collocare l'audio in una determinata posizione. Prima di associare il clip sonoro, è possibile selezionarlo tramite un menu a tendina e ascoltarlo cliccando su Play. Per aggiungere nuovi clip audio non presenti nel software di default, è sufficiente inserirli nella cartella audio del software. Infine cliccando su salva è possibile memorizzare la mappa, per permettere in futuro la possibilità di richiamarla utilizzando il tasto carica.

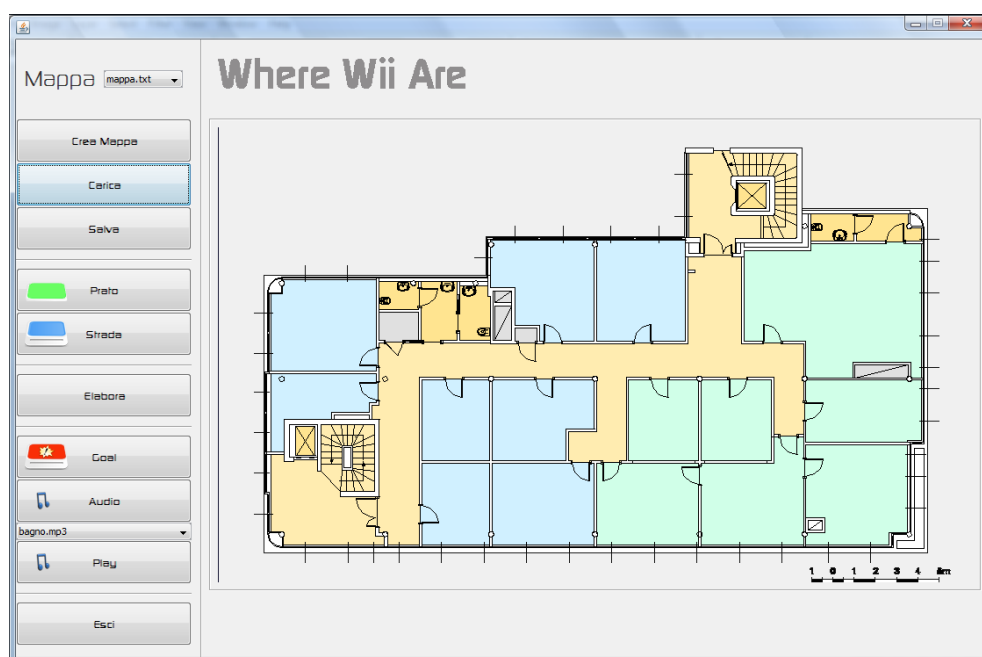


Figura A.3: Ambiente gestione mappa.

A.4 Start WWA

Una volta entrati in questa sezione è possibile scegliere la mappa dell'ambiente nel quale si desidera operare. Cliccando su “carica” apparirà una finestra in cui sarà possibile selezionare il file da caricare. Di default viene caricato il file col nome “mappa”. Per avviare il sistema di guida, dopo aver predisposto l'ambiente coi marker e indossato l'auricolare, è sufficiente premere il tasto **1** del Wiimote. A questo punto una voce guida chiederà di scegliere la posizione in cui ci si trova e la destinazione da raggiungere. Per selezionare il punto di partenza o di arrivo è sufficiente muovere il Wiimote verso destra o verso sinistra, oppure utilizzare i normali tasti direzionali.

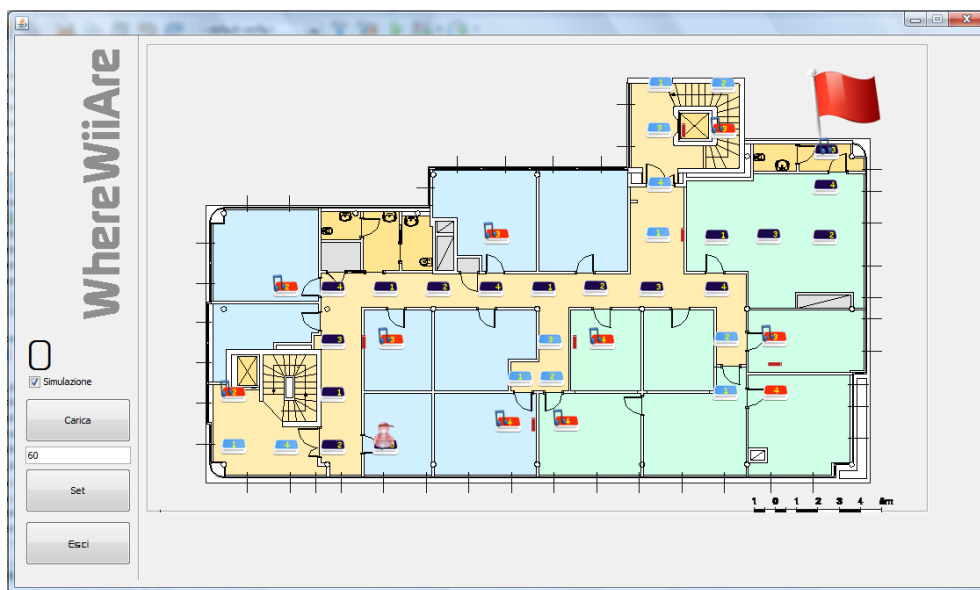


Figura A.4: Ambiente Start WWA.

Per confermare la scelta è necessario premere il tasto **A**. A questo punto a video verrà evidenziato il percorso ottimo che collega i due punti selezionati. Durante gli spostamenti una voce guiderà l'utente con semplici indicazioni: continua in questa direzione, destra, sinistra e torna indietro. Nel caso in cui si sbaglia direzione, il sistema ricalcolerà il percorso automaticamente e fornirà le nuove indicazioni. Una volta raggiunta la destinazione la voce comunicherà la posizione finale e, nel caso si voglia continuare la navigazione sarà possibile premere nuovamente il tasto **1** è selezionare una nuova meta. E' indispensabile, per il corretto funzionamento del sistema, tenere puntato il Wiimote verso in direzione dei tag durante tutto l'utilizzo.