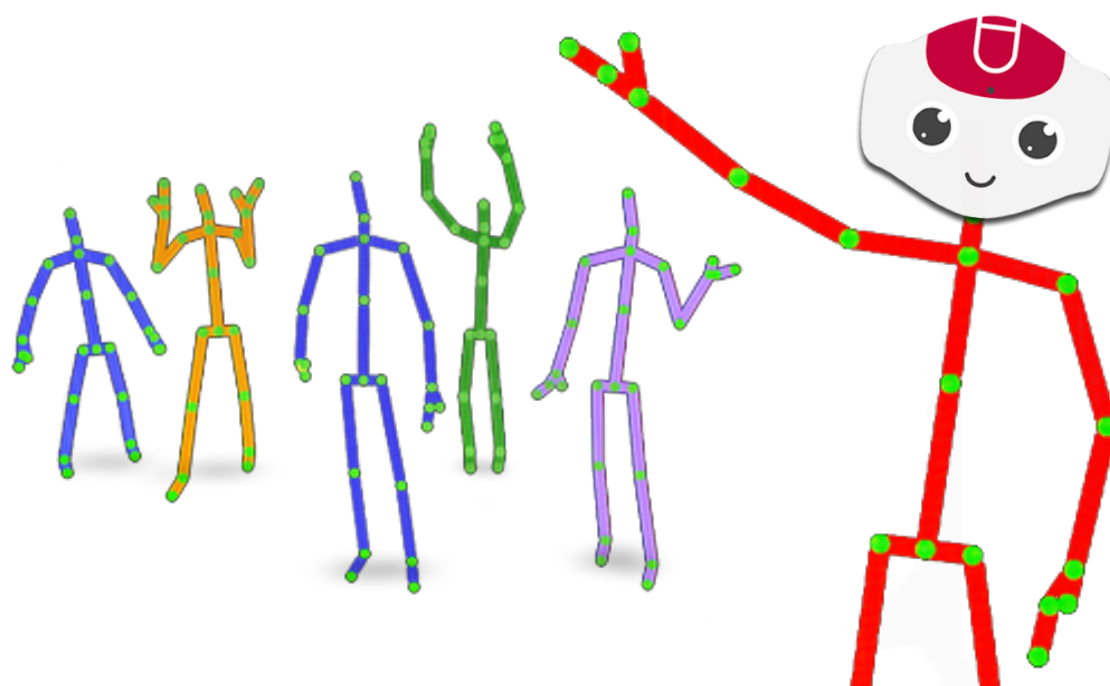


NAO e Kinect

Lorenzo Daidone
I.I.S. "I. Calvino", Genova

A.S. 2016-2017



NAO e Kinect

Lorenzo Daidone

A.S. 2016/2017

I.I.S. "I. Calvino", Genova

Classe: 5[^]BEA

Abstract [ITA]: In situazioni di pericolo o di impossibilità, risulta essenziale comandare in remoto dei robot. Il software si propone come un framework per lo sviluppo di applicazioni interattive.	Abstract [ENG]: In situations of danger or impossibility, it is essential to remotely control robots. The software is intended as a framework for the development of interactive applications.
--	---

Keywords:

- *Aldebaran NAO robot*
- *Microsoft 360[®] Kinect*

Attenzione: le parole mostrate in color rosso sono definite nell'appendice "Glossario", al fondo di questa tesina.

Introduzione:

Italiano: Nella maggior parte dei casi, quando si approccia un software per il controllo di un robot, l'attenzione che gli sviluppatori pongono nella progettazione dell'interfaccia grafica (GUI), risulta essere scarsa. L'innovazione introdotta con la NUI (<i>Interfaccia Utente Naturale</i>) rende ora possibili una vasta gamma di applicazioni che consentono all'utente di interagire con il dispositivo utilizzando i movimenti e un linguaggio naturale. Una possibile applicazione del framework, potrebbe essere quella di pilotaggio remoto di robot, sia per dispositivi antropomorfi che non. Inoltre, la piattaforma potrebbe essere utilizzata per applicazioni mediche, ad esempio nel campo delle disabilità gravi, per tenere traccia dei progressi, sviluppando applicazioni che sottopongano all'utente giochi al fine di affinare motricità grossa e fine. In questa tesi, il framework è stato implementato nel seguente modo: l'umano si muove davanti al dispositivo di acquisizione. Mediante una comunicazione senza fili, il robot riproduce in tempo reale ciò che l'umano esegue.	Inglese: In most cases, when approaching a robot controlling software, the attention paid by developers in designing the GUI turns out to be poor. The innovation introduced with the NUI , now makes, nowadays, possible a wide range of applications that allow the user to interact employing movements and a natural language. A potential application, could be a remotely controlled robot, both for anthropomorphic robots and not. Furthermore, the platform could be used for medical treatment, as an example in the field of severe disabilities, in order to keep track of progress, developing applications that present to user games in order to improve fine and large motricity. In this thesis, the framework has been implemented in the following way: the human moves in front of capture device. Through a wireless communication system, the robot reproduces in real time what the human performs.
--	--

Data la complessità del problema, ho suddiviso il problema in tre macro-problemi:



1. Acquisizione giunti arti corpo umano

Per acquisire la posizione nello spazio dei "giunti" del corpo umano si è deciso di utilizzare il dispositivo Kinect, distribuito dalla Microsoft a partire dal 2010.



Figura 1 - Kinect v. 1414

Il dispositivo possiede un proiettore e un ricevitore ad infrarossi, telecamera e un array di microfoni che permettono di individuare l'angolo da cui l'utente sta parlando. È possibile variare l'angolo di visione verticale, agendo sul motore di *tilt*. L'escursione è di $\pm 27^\circ$.
Costo: ~ 150 \$.

Specifiche del Kinect V1 (2010):

Kinect	
Portata	MAX: ~ 4 m, MIN: ~ 80 cm
Angolo di visione	43,5°
Camera	VGA resolution 640 * 480 pixel. Profondità colore: 8 bit.
Velocità di acquisizione frame (stream profondità e colore)	30 frame per secondo (FPS)
Formato audio	16-kHz, 24-bit mono pulse code modulation (PCM)
Caratteristiche audio input	Array di quattro microfoni con convertitore analogico-digitale a 24 bit e processamento interno del suono con eliminazione dell'eco e riduzione del rumore
Caratteristiche accelerometro	Range configurato di $\pm 2g$

Microsoft distribuisce gratuitamente il kit di sviluppo software (SDK: *Software Development Kit*). L'SDK comprende:

- driver per utilizzare il dispositivo
- application programming interfaces (APIs) e interfacce per il dispositivo

- Esempi, strumenti e altre risorse per lo sviluppo

Gli esempi contenuti nel pacchetto sono scritti nei seguenti linguaggi: C#, C++, VB.

Dal software a corredo nel *Kinect SDK*, è stato modificato un esempio. L'esempio utilizzato per realizzare il tracking dei giunti del corpo umano è "Skeleton Basics"; esso è scritto in linguaggio C# e permette di tracciare un massimo di due persone.

Per ogni persona sono rappresentati 20 giunti.

Il programma, così come viene distribuito, non permette l'invio delle coordinate dei giunti via rete.

Per modificare il progetto è stato utilizzato Visual Studio, l'ambiente di sviluppo integrato (**IDE**) sviluppato da Microsoft che permette la realizzazione di applicazioni, siti web, applicazioni web e servizi web.

Le modifiche apportate al progetto SkeletonBasics originario, sono le seguenti:

- Aggiunto metodo **JointToString** alla classe principale del software: `MainWindow`; la quale eredita i metodi dalla classe `Window`. Il metodo `JointToString` crea una stringa composta da nome del giunto e le rispettive coordinate del giunto nello spazio.
- Aggiunta classe **Server.cs** al progetto. Essa prevede alcune funzionalità di rete quali: creazione socket lato server, invio dati, ricezione dati, chiusura socket.
 - creazione socket: crea un socket lato server accettando come parametro il numero di porta. Di default la porta utilizzata è la n° 20248, scelta arbitrariamente. Il tipo di socket è stream, protocollo TCP/IP. Vedi appendice: "Focus su socket e indirizzi IP".
 - invio dati: accetta una stringa come parametro. Invia tale stringa e attende un echo dal server.
 - chiusura comunicazione: annuncia al server uno shutdown, disabilitando sia invio che ricezione e poi chiude la connessione all'host remoto, rilasciando tutte le risorse gestite e associate al socket.
 - calcolo e invio dimensione messaggio (metodo non utilizzato): per avere una maggiore sicurezza dell'integrità del messaggio, potrebbe essere utile inviare al client la dimensione del messaggio precedentemente inviato.

2. Invio informazioni al robot

Il computer e il robot sono connessi ad ad una rete locale.

Collegare il robot ad una rete locale utilizzando un cavo, potrebbe essere limitante per alcune operazioni e la lunghezza del cavo potrebbe discriminare la posizione del robot stesso. Pertanto NAO è connesso alla rete locale via Wi-Fi.

Il router utilizzato per testare la comunicazione dei due dispositivi è il TL-WR841N prodotto dalla TP-Link. La sicurezza utilizzata per l'autenticazione e la cifratura dei dati è la WPA2-PSK.

Il server è connesso via cavo perché non fondamentale una connessione senza fili.

Al fine di far comunicare il client con il server in rete, è stato creato un socket di tipo stream, protocollo TCP in cui gli indirizzi IP siano della versione 4. Vedi appendice: "Focus su socket e indirizzi IP".

Il formato con cui il server invia il nome dei giunti e le rispettive coordinate, per un'opportuna elaborazione, è stato creato ad hoc. Non si utilizza un protocollo standardizzato.

Il messaggio ASCII inviato a NAO è una concatenazione di nome giunto e rispettive coordinate nello spazio. Il sistema di riferimento per tali coordinate è il seguente: sistema di riferimento destrorso, fronte rivolta al dispositivo, asse x verso destra dell'utente e asse z direzionato verso l'utente [fig. 2].

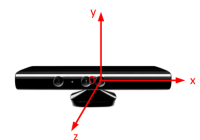


Figura 2 - Sistema coordinate

In generale, la regola per il protocollo è la seguente:

$$\text{NomeGiunto}_n \quad X_n \quad Y_n \quad Z_n /$$

Esempio di messaggio inviato dal server al client:

```
HipCenter 0,02213011 -0,1516186
3,294796/Spine 0,0270134 -0,0878974
3,345237/ShoulderCenter 0,02703401
0,2609315 3,324159/Head 0,01493661
0,4477339 3,266301/ShoulderLeft
-0,1625187 0,1579514 3,344531/
ElbowLeft -0,3824059 0,02974367
3,294011/WristLeft -0,5916628
-0,04512607 3,192345/HandLeft
-0,6193663 -0,04728847 3,160089/
ShoulderRight 0,2109764 0,1430885
3,312534/ElbowRight 0,371758
-0,01951969 3,216079/WristRight
0,5452417 -0,1115142 3,066102/
HandRight 0,6481183 -0,1721237
2,975143/HipLeft -0,06597368
-0,236739 3,2796/KneeLeft
-0,06226847 -0,7335333 3,314915/
AnkleLeft -0,02930086 -1,104403 3,362409/FootLeft -0,05503895
-1,183514 3,299364/HipRight 0,1072637 -0,2343558 3,289687/KneeRight
0,1137078 -0,6194826 3,058723/AnkleRight 0,110105 -1,020778
2,865749/FootRight 0,1243722 -1,058315 2,742855/
```

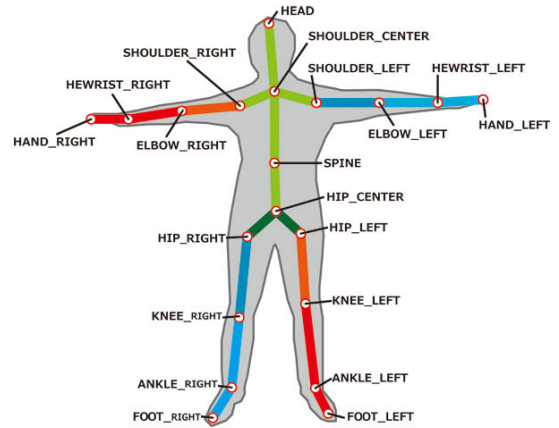


Figura 3

Una volta che il software “Skeleton Basics” è stato dotato di connettività su rete locale e della capacità di creare un messaggio che il client possa correttamente interpretare, la parte server si può considerare terminata.

3. Riproduzione dei movimenti da parte del robot

Kinect ritorna solamente le coordinate dei giunti di un corpo umano. Invece, le funzioni Python che gestiscono il movimento dei giunti del robot, accettano unicamente angoli come parametro.

Nao è un robot umanoide progettato da Aldebaran (ora SoftBank Robotics) nel 2006.

Caratteristiche principali	
Altezza	58 cm
Connettività	Wi-fi, ethernet, USB
Sistema operativo	OpenNAO, distribuzione GNU/Linux di Gentoo
Gradi di libertà	25
Videocamere	1280 * 960 @30fps(x2)
Altoparlanti	2
Microfoni	4



Figura 4 - NAO Robot

Caratteristiche principali	
Massa	5,4 kg
Sensori	Ultrasuoni, infrarossi, di pressione, di tatto
Costo	~ 7300 €

Il software lato client è stato realizzato con Choregraphe, l'ambiente di sviluppo distribuito gratuitamente da Aldebaran, che fornisce una serie di strumenti per sviluppare agevolmente applicazioni. Nell'**IDE** è compreso anche un simulatore.

Per avere una maggiore libertà di sviluppo, tutto il programma è stato scritto in Python.

Per decodificare il messaggio inviato dal server e rendere i dati fruibili da NAO sono necessarie le seguenti operazioni:

1. il messaggio è frammentato in una lista di stringhe, considerando il carattere slash / come un carattere di interruzione
2. le stringhe ottenute vengono nuovamente frammentate usando lo spazio come carattere di interruzione.
3. nei numeri che rappresentano le coordinate le virgole sono sostituite con dei punti (notazione anglosassone)
4. effettua un cast di tipo convertendo le coordinate, rappresentate come stringhe, in numeri di tipo `float` (reali)

Una volta trasformato il messaggio in una matrice bidimensionale che contiene i nomi dei giunti e le rispettive coordinate, occorre passare da coordinate ad angoli.

L'intento è di rendere il framework il più generale possibile, pertanto è stata creata una classe di nome "Angles". I metodi della classe permettono di effettuare calcoli su vettori e da questi leggere i valori dei relativi angoli.

Ottenimento di un angolo fra due vettori

Per passare da coordinate dei **punti in tre dimensioni** ad **angoli** è stata utilizzata la formula inversa del prodotto scalare.

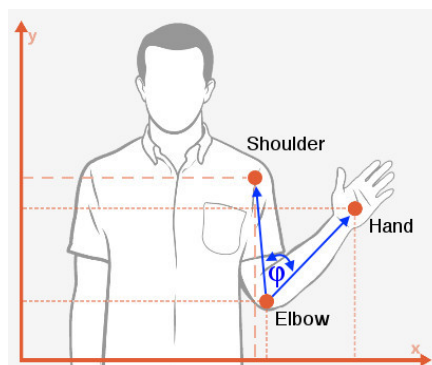


Figura 5

Prodotto scalare:

$$v \cdot w = |v| |w| \cos(\varphi)$$
 Formula inversa:

$$\cos(\varphi) = \frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}| |\mathbf{w}|}$$

Applico la funzione inversa di coseno:

$$\varphi = \arccos(\cos(\varphi))$$

Se i vettori sono espressi tramite componenti:

$$\mathbf{v} = (v_x, v_y, v_z) \text{ e } \mathbf{w} = (w_x, w_y, w_z)$$

il prodotto scalare diviene:

$$\mathbf{v} \cdot \mathbf{w} = (v_x, v_y, v_z) \cdot (w_x, w_y, w_z)^T = v_x w_x + v_y w_y + v_z w_z$$

Dalle relazioni precedenti si ottiene:

$$\cos(\varphi) = \frac{v_x w_x + v_y w_y + v_z w_z}{\sqrt{v_x^2 + v_y^2 + v_z^2} \sqrt{w_x^2 + w_y^2 + w_z^2}} \quad \text{[Relazione no.1]}$$

Coseni direttori

I coseni direttori di un vettore, sono i coseni degli angoli convessi che la retta (o la retta su cui giace il vettore) forma con gli assi cartesiani.

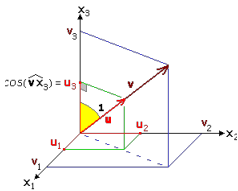


Figura 6

$$u_1 = \frac{\pm v_1}{\sqrt{v_1^2 + v_2^2 + v_3^2}}$$

$$u_2 = \frac{\pm v_2}{\sqrt{v_1^2 + v_2^2 + v_3^2}}$$

$$u_3 = \frac{\pm v_3}{\sqrt{v_1^2 + v_2^2 + v_3^2}}$$

Il denominatore della frazione è il modulo del vettore \mathbf{v} .

La relazione no.1, precedentemente trattata, si può anche esprimere in funzione dei coseni direttori.

$$\begin{aligned} \arcsin(\cos \varphi) &= \arcsin\left(\frac{v_x w_x + v_y w_y + v_z w_z}{\sqrt{v_x^2 + v_y^2 + v_z^2} \sqrt{w_x^2 + w_y^2 + w_z^2}}\right) = \arcsin\left(\frac{v_x w_x + v_y w_y + v_z w_z}{|\mathbf{v}| |\mathbf{w}|}\right) = \\ &= \arcsin\left(\frac{v_x}{|\mathbf{v}|} \cdot \frac{w_x}{|\mathbf{w}|} + \frac{v_y}{|\mathbf{v}|} \cdot \frac{w_y}{|\mathbf{w}|} + \frac{v_z}{|\mathbf{v}|} \cdot \frac{w_z}{|\mathbf{w}|}\right) \quad \text{[Relazione no. 2]} \end{aligned}$$

Con la relazione no. 2 sopra riportata, è possibile associare un angolo ad ogni giunto del robot, prendendo in considerazione due vettori.

La tabella seguente mette in relazione un angolo di NAO con i rispettivi vettori ottenuti da Kinect.

Arto destro				
Nao joint	Abbreviazione	v	w	Vertice in comune?
Shoulder pitch	rsp	anca-spalla	spalla-gomito	Sì
Shoulder roll	rsr	anca sx-anca dx	spalla-gomito	No
Elbow yaw	rey	anca-spalla	gomito-polso	No
Elbow roll	rer	spalla-gomito	gomito-polso	Sì

Non sempre i vettori presi in considerazione hanno punti in comune.

L'angolo ottenuto per ciascun giunto non è direttamente assegnabile al rispettivo motore mediante apposite istruzioni. È necessario introdurre un offset nell'angolo.

Giunto	Offset [rad]	Giunto	Offset [rad]
Left Elbow Roll	$-\pi$	Right Elbow Roll	0
Left Elbow Yaw	$-3\pi/2$	Right Elbow Yaw	$-\pi/2$
Left Shoulder Roll	$-\pi/2$	Right Shoulder Roll	$-3\pi/2$
Left Shoulder Pitch	$-\pi/2$	Right Shoulder Pitch	$-\pi/2$

Tali offset sono stati determinati empiricamente, utilizzando un widget compreso nell'**IDE Choregraphe: Motion**.

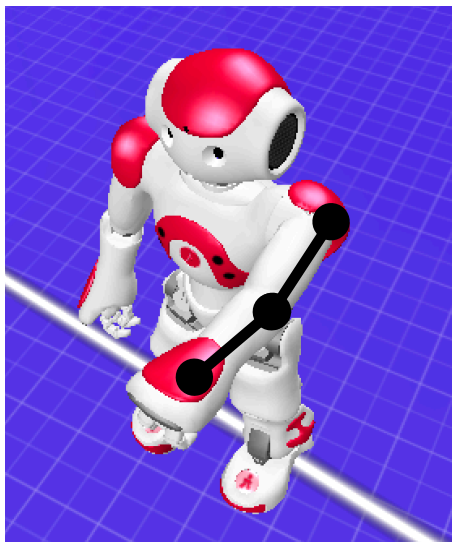


Figura 7a

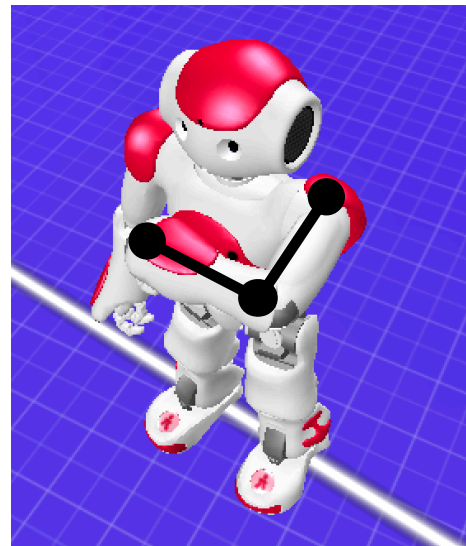


Figura 8a

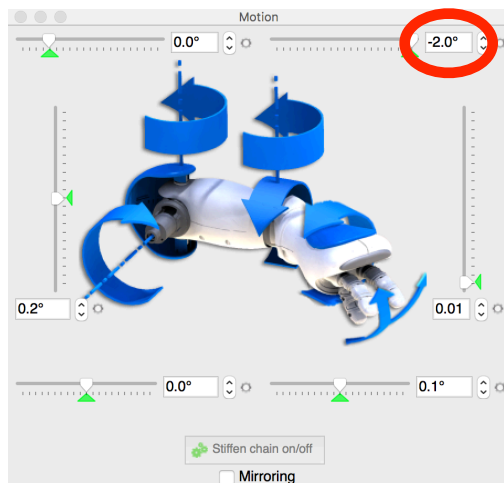


Figura 7b

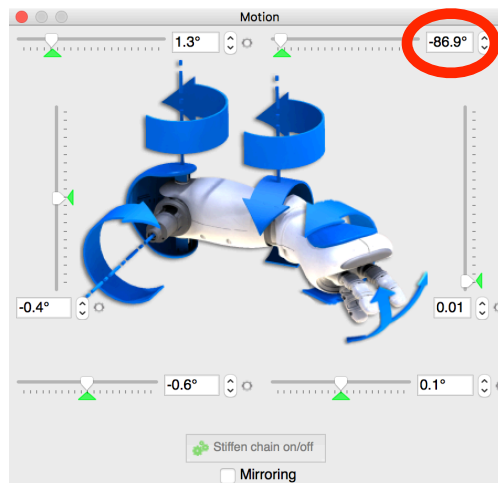


Figura 8b

Come si può notare nella **fig. 7a** l'angolo fra i due vettori è di circa 180° (π rad) mentre l'angolo del motion widget è di ~ 0 gradi [fig. 7b].

Analogamente, nella **fig. 8a** l'angolo fra i due vettori è di circa 90° ($\pi/2$ rad) mentre nel motion widget è indicato $\sim -90^\circ$ [fig. 8b].

In alcuni casi Kinect potrebbe ritornare una lettura sfalsata dei giunti del corpo umano con la conseguenza di ottenere calcoli errati dei coseni direttori. L'errore più comune è avere il modulo del vettore uguale a 0 e quindi coseni direttori tendenti a infinito. Per evitare che sui giunti del robot compaiano angoli eccessivi, provocando la rottura degli stessi, viene prima effettuato un controllo. Se l'angolo è compreso in un determinato range, allora il suo valore viene inviato all'attuatore del giunto, altrimenti la posizione del giunto rimane invariata. In gergo quest'operazione viene chiamata **clipping**.

Considerazioni:

Uno dei miei obiettivi è di rendere il programma semplice da utilizzare e molto "**robusto**". L'utente deve solo avviare il programma server, il programma client ed essere pronto ad operare. Ecco perché sono state aggiunte alcune funzionalità che, sebbene non siano essenziali al funzionamento del programma, lo rendono maggiormente **user friendly**.

Disattivazione modalità vita autonoma: quando si accende l'umanoide, di default è attivata la "autonomous life". Questa funzionalità rende il robot più "vivo" facendolo ondeggiare sui fianchi e seguire perfino il volto dell'utente. Ai fini del programma, il controllo dell'hardware deve essere totalmente eseguito dal software dell'utente e non da quello della modalità autonoma; pertanto l'*autonomous life* viene disattivata non appena avviato il software client.

Set posizioni iniziali: Il robot possiede di default un modulo denominato "ALRobotPosture" che consente di portare il robot in una posizione predefinita.

Fra le posizioni disponibili è stata scelta la **StandInit** [fig. 9] come posizione iniziale perché garantisce un buon equilibrio al robot ed è disponibile su tutte le versioni di NAO.

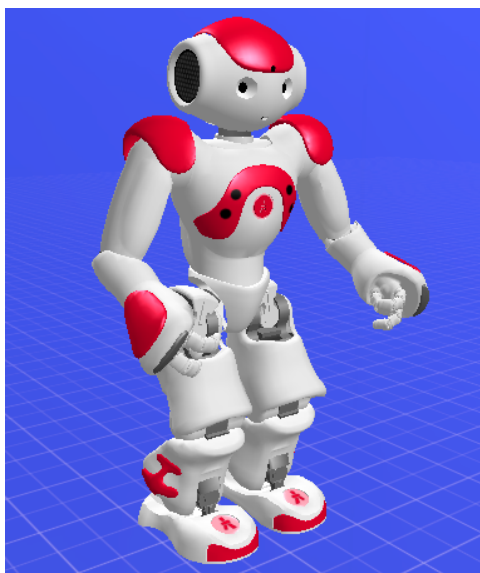


Figura 9 - StandInit

Glossario:

Il glossario presenta una veloce definizione e una dettagliata, sia in lingua inglese che in italiano.

- Clipping: “Spuntata, ritaglio”

Clipping, in the context of computer graphics, is a method to selectively enable or disable rendering operations within a defined region of interest.

Clipping, nel contesto della *computer grafica* è un metodo per abilitare e disabilitare selettivamente le operazioni di rendering in una regione di interesse definita.

- Framework: *struttura, intelaiatura*. Architettura logica di supporto su cui un software può essere progettato e realizzato.

In computer programming, a software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software.

In informatica, un framework software è un’astrazione in cui il software fornisce funzionalità generiche che possono essere modificate selettivamente con del codice scritto dall’utente, fornendo così un software specifico per l’applicazione.

- GUI: Graphic User Interface

The graphical user interface (GUI), is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators, instead of text-based user interfaces. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard.

L’interfaccia utente grafica, è un tipo di interfaccia utente che permette agli utenti di interagire con dispositivi elettronici attraverso icone e indicatori visuali, al posto di interfacce basate sul testo. Le GUI sono state introdotte in risposta alla curva di apprendimento delle interfacce a linea di comando, che richiedono comandi da digitare su una tastiera per computer.

- IDE: Integrated Development Environment

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger. Most modern IDEs have intelligent code completion.

Un ambiente di sviluppo integrato (IDE) è un’applicazione software che fornisce ai programmatori servizi completi per lo sviluppo software. Normalmente un IDE consiste in un editor di testo, strumenti per compilare il codice e un debugger. I più moderni IDE possiedono il completamento automatico del codice.

- NUI: Natural User Interface

In computing, a natural user interface, or NUI, or natural interface is a user interface that is effectively invisible, and remains invisible as the user continuously learns increasingly complex interactions. The word natural is used because most computer interfaces use artificial control devices whose operation has to be learned.

In informatica, un'interfaccia utente naturale, NUI, o interfaccia naturale è un'interfaccia utente che è completamente invisibile e rimane invisibile in quanto l'utente continua ad imparare interazioni complesse. La parola *naturale* è usata perché la maggior parte delle interfacce per computer usano dispositivi di controllo artificiale il cui funzionamento deve essere appreso.

- Robusto: in informatica, capacità di un software di comportarsi in modo ragionevole in situazioni impreviste

In computer science, robustness is the ability of a computer system to cope with errors during execution and cope with erroneous input. Robustness can encompass many areas of computer science, such as robust programming, robust machine learning, and Robust Security Network. Formal techniques, such as fuzz testing, are essential to showing robustness since this type of testing involves invalid or unexpected inputs.

In informatica, la robustezza è l'abilità di un computer di resistere agli errori durante l'esecuzione e resistere ad input errati. La robustezza comprende diverse aree dell'informatica come ad esempio la *programmazione robusta*, *machine learning robusto* e *rete di sicurezza robusta*. Tecniche come il *fuzz testing* sono essenziali per mostrare la robustezza poiché questo tipo di test comporta ingressi non validi o inaspettati.

- User friendly: di facile utilizzo anche per chi non è esperto.

In software engineering, usability is the degree to which a software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.

In ingegneria del software, l'usabilità è il grado in cui un software può essere utilizzato da consumatori specifici per raggiungere obiettivi quantificati con efficacia, efficienza e soddisfazione, in un preciso contesto di utilizzo.

Focus su socket e indirizzi IP:

I socket di tipo *stream* forniscono comunicazioni bidirezionali affidabili. Il protocollo di comunicazione standard per questo tipo di socket è il TCP (*Transmission Control Protocol*). Il protocollo TCP è progettato in modo che i pacchetti di dati arrivino a destinazione senza errori e in sequenza.

Attualmente, la maggior parte dei sistemi su internet ha un indirizzo IP (internet protocol), costituito dal noto schema a quattro byte (32 bit) nella forma xx.xx.xx.xx.

Esempio:

192.168.0.1

Dal 1981 si utilizzano indirizzi IPv4 i quali hanno uno spazio di indirizzamento a 32 bit.

$2^{32} \cong 4,3 * 10^9$ indirizzi

Riferimenti:

1. Specifiche Kinect: <https://msdn.microsoft.com/en-us/library/jj131033.aspx>
2. Kinect joint type enumeration: <https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx>
3. Autenticazione Wi-Fi: Hacker 7.0, McClure - Scambray - Kurtz. Edito da Apogeo
4. Indirizzo IP: pag 285, Arte dell'Hacking, Volume 1. Jon Erickson. Edito da Apogeo
5. Specifiche NAO robot: http://doc.aldebaran.com/2-1/family/robots/index_robots.html
6. Coseni direttori: http://it.wikipedia.org/wiki/Coseni_direttori
7. NAO joints: http://doc.aldebaran.com/2-1/family/nao_dcm/actuator_sensor_names.html
8. NAO joint control: <http://doc.aldebaran.com/1-14/naoqi/motion/control-joint.html>
9. StandInit: <http://doc.aldebaran.com/1-14/naoqi/motion/alrobotposture.html>