

Sommario

Introduzione su Pepper	1
Pepper, com'è fatto.....	2
Cosa può fare Pepper	2
Pepper per le famiglie	2
Pepper per le aziende.....	3
Prezzo e altri dettagli	3
Prezzo per la versione “imprese”	3
Pepper robot non vuole essere un robot utile	3
Robot sociali	4
Guida sviluppatori	8
SoftBank Pepper	8
Sviluppo	8
Robot	8
NAOqi OS	9
Prima configurazione Pepper	11
Come connettere Pepper in rete la prima volta	14
Download e installazione del software Aldebaran	18
Choregraphe Suite	19
Lingue supportate.....	24
Python in Choregraphe.....	26
Dialogo in Choregraphe	30
SDK	35
Creare un'applicazione	36
Hello World.....	36
Applicazione di ballo.....	40
Applicazione interattiva	42
Creare un'applicazione da zero.....	47
Utilizzo del tablet.....	54
La nostra esperienza.....	57
Presentazione DIB	57
FAQ	59
Feature aggiuntive.....	63
Informazioni DIB.....	66
Riferimenti.....	71

Introduzione su Pepper

Pepper, com'è fatto

Pepper è un **robot umanoide** alto un metro e venti e pesa 28 chili (Figura 1). Si muove su ruote e interagisce con gli amici umani anche attraverso un tablet che si trova all'altezza del torso. Grazie a quattro microfoni direzionali che ha nella sua testa, è in grado di rilevare la provenienza di suoni e voci e girare la faccia in direzione di chi sta parlando. Questi microfoni gli consentono anche di analizzare il tono della voce per interpretare lo stato emotivo. Una fotocamera 3D e due telecamere HD gli permettono di riconoscere invece immagini, oggetti e volti.

Grazie alle sue tre ruote multidirezionali può muoversi in tutte le direzioni a una velocità massima di tre chilometri orari. È dotato di 20 motori che gli consentono di muovere testa, braccia e schiena. La sua batteria agli ioni di litio gli consente di avere fino a dodici ore di autonomia. I suoi sensori laser gli permettono di rilevare eventuali ostacoli al buio, anche se questo umanoide non è in grado di vedere in assenza di luce. I sensori laser sono sei, e a questi si aggiungono due sensori a ultrasuoni e tre rilevatori di ostacoli. [1]

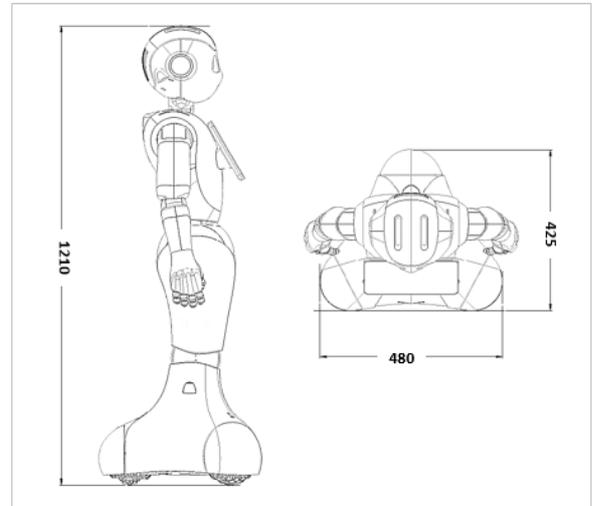


Figura 1: Pepper

Cosa può fare Pepper

Pepper (Figura 2) è un social robot che parla e comprende il linguaggio umano, un automa che percepisce le tue emozioni e agisce di conseguenza. È in grado di riconoscere se sei contento o triste, se ti sorprende per qualcosa o se sei arrabbiato, analizzando il tono della tua voce, la posizione della testa e la postura che assumi mentre parli. Dialoga in diverse lingue, tra cui l'italiano. È connesso sempre a Internet e ciò gli consente, quando ne hai bisogno, di suggerirti una ricetta, di tenerti informato su cosa è successo nel mondo nelle ultime ore piuttosto che tenerti aggiornato sulle previsioni del tempo. [1]

Pepper per le famiglie

Pepper è stato programmato per vivere con te. È un robot **da compagnia**, il suo scopo è quello di dialogare con te, intrattenerti e intrattenere ogni componente della tua famiglia, compresi i bambini, e quindi non bisogna aspettarsi da lui che impari a pulire il pavimento o che sappia fare altri lavori domestici.

Non è stato progettato per questo ma per interagire con te, informarti, darti indicazioni utili. E più vive in famiglia e più impara a conoscere i gusti e le caratteristiche di ogni componente e a soddisfare meglio le curiosità e i bisogni di ciascuno. Questo significa che apprende dall'interazione con te e il suo comportamento cambia di conseguenza, perché è in grado di memorizzare i tratti della tua personalità e le tue preferenze e si adatta gradualmente ai tuoi gusti e alle tue abitudini. [1]



Figura 2: Pepper

Pepper per le aziende

Pepper non è solo un home robot. Nella sua versione “for biz”, destinata cioè al business, alle aziende, può essere usato in negozi, attività commerciali, attività turistiche. Può fare per esempio il commesso di negozio, il consulente bancario piuttosto che l’assistente di bordo sulle navi da crociera.

Marchi come Nestlè e Loft sono stati tra i primi ad averlo “assunto” come commesso per alcuni punti vendita giapponesi (anche in questo caso a fare da apripista è stato il Giappone, da sempre particolarmente sensibile alle novità tecnologiche). HSBC, uno dei gruppi bancari più importanti del mondo, nell’estate del 2018 ha firmato un accordo con Softbank Robotics America per far “lavorare” Pepper nelle sue filiali bancarie.

Costa Crociere ne ha fatto uno steward per gli ospiti delle sue crociere. Il numero di aziende che utilizzano robot nelle loro attività è in crescita. Il settore turistico è tra quelli più sensibili. E Pepper non è il solo umanoide che può essere impiegato in attività lavorative. Nao, per esempio, ha lavorato in un hotel gestito da robot umanoidi nella prefettura di Nagasaki, all’interno di un parco divertimenti. [1]

Prezzo e altri dettagli

Robot Pepper in realtà non è un robot giapponese ma europeo. Le sue origini sono europee. Questo umanoide nasce dall’impegno degli ingegneri di Aldebaran Robotics, azienda francese, acquisita però qualche anno fa dal gruppo di telecomunicazioni Softbank. Oggi Aldebaran si chiama Softbank Robotics, È proprio Softbank che nel mese di giugno 2015 ha cominciato a vendere robot Pepper in Giappone. Mille esemplari alla volta, ogni mese. Risultato: mille robot Pepper venduti ogni volta in un solo minuto.

Per quanto riguarda il prezzo di Pepper, questo umanoide per il momento può essere acquistato dai privati solo in Giappone al prezzo di 198 mila yen, ovvero circa 1500 euro. A cui bisogna aggiungere il costo dei servizi forniti da Softbank, pari a poco meno di 200 euro mensili. [2]

Prezzo per la versione “imprese”

L’idea di Softbank Robotics è quello di fare di Pepper robot anche uno strumento di marketing commerciale al servizio delle imprese. Pepper è dunque più di un commesso di negozio. La sua intelligenza artificiale si integra con altri sistemi informatici e analizza milioni di dati contenuti per esempio nelle e-mail o nei post sui social network per capire meglio i gusti dei clienti.

È possibile ordinare la versione per le imprese di robot Pepper (si chiama Pepper for biz) con una serie di software preinstallati, utili per svolgere una serie di funzioni, come ad esempio quella di dialogare con i clienti. Il prezzo in Yen nel 2015 si aggirava attorno ai 55.000 mila yen (poco più di 400 euro) per 36 mesi, per un totale di 1.980.000 yen (circa 15 mila euro). In Italia il prezzo si aggira attorno ai 17 mila euro (prezzo nel 2019).

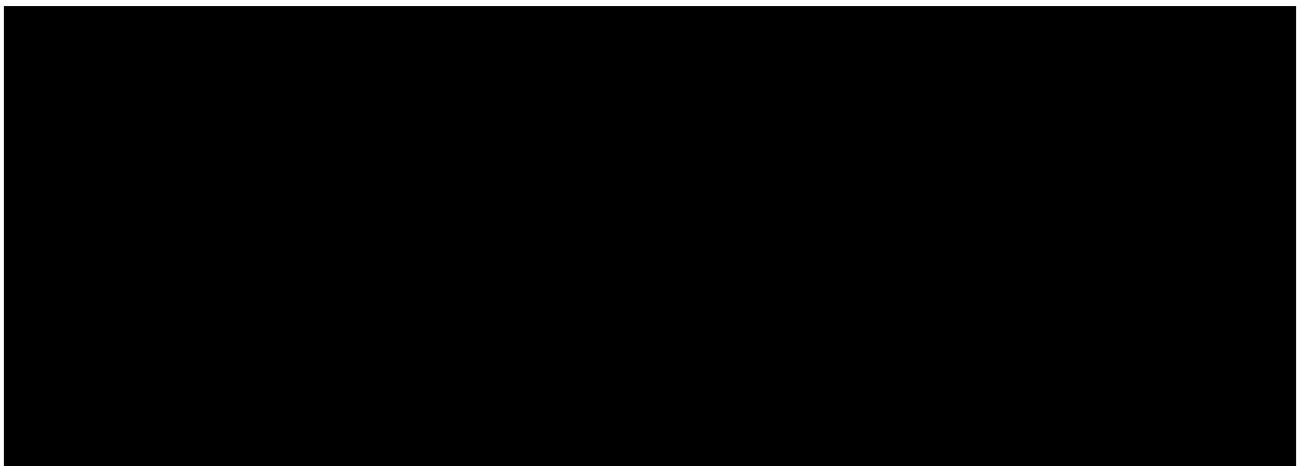
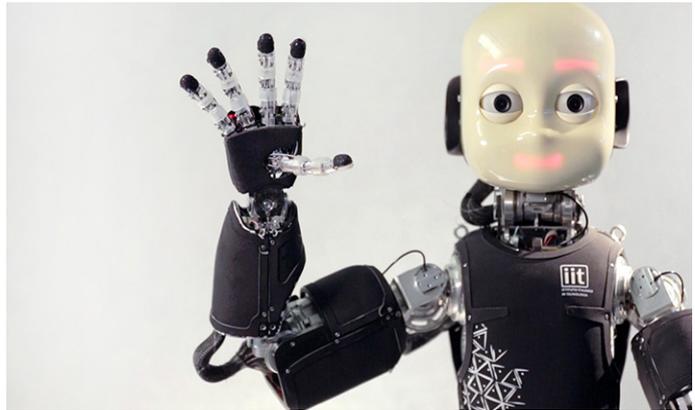
In alternativa si potrà affittare robot Pepper per 1500 yen all’ora, circa 11 euro. Il robot arriverà a destinazione con una serie di funzioni che gli consentono di svolgere una serie di compiti di routine nel settore commerciale. [2]

Pepper robot non vuole essere un robot utile

Pepper robot non vuole essere un **robot utile**, vuole essere il “tuo” **amico robot**. È per questo che più che un robot “operaio” dotato di abilità “fisiche” (prendere o sollevare un oggetto, fare le pulizie,

cucinare, per fare qualche esempio) dobbiamo immaginarlo come un robot da compagnia. Farne un compagno di vita è l'obiettivo di Softbank.

Di seguito mostriamo un video dell'azienda che lo produce e lo vende, che mostra proprio questo. Si vede Pepper robot che segue la crescita di un bambino, lo accompagna nelle sue giornate, memorizzando i momenti della sua vita, diventando quasi la sua memoria storica. Foto, video, appunti sui progressi e sulla sua crescita. Se non è un compagno di vita questo... [3]



Robot sociali

Al giorno d'oggi, vi sono diversi robot sociali presenti per il mondo, di seguito ne presentiamo alcuni tra i più famosi e importanti.

Asimo

Considerato fra i più intelligenti al mondo, Asimo (Figura 3) è in grado di correre ed evitare ostacoli in movimento, rispondere a semplici comandi vocali, riconoscere i volti e le voci di un gruppo ristretto di persone ed è in grado di dosare la forza per afferrare gli oggetti. La particolarità di questo robot è la naturalezza dei suoi movimenti e grazie ai trentaquattro servomotori, installati all'altezza delle giunture, può camminare su superfici irregolari e correre fino a 9 km/h. Asimo può anche svolgere mansioni che richiedono un grado di attenzione maggiore come aprire una bottiglia o afferrare un bicchiere di plastica senza schiacciarlo [4].



Figura 3:ASIMO

iCub

iCub (Figura 4) è la piattaforma robotica più diffusa al mondo. Questo robot presenta le sembianze di un bambino di quattro anni, è alto 104cm e pesa 22 chili. La sua notorietà è attribuitagli in

Figura 4:iCUB

quanto se riesce bene in qualcosa sorride, la cosa più affascinante è che se commette un errore, al tentativo successivo, si corregge fino quando non svolge un comando in modo corretto. Infatti, è progettato per imparare. Ha la possibilità di comunicare con le espressioni del volto e possiede perfino una pelle artificiale. [5].

Surena

È un robot umanoide “intelligente” che ricorda tanto Asimo. Questo robot (Figura 5), a differenza degli altri, ha un peso di circa 98 chili ed ha solo 31 gradi di libertà e di movimento per interagire con gli umani. Attualmente Surena può svolgere una serie di compiti, per esempio camminare su terreni sconnessi e pianeggianti, salire e scendere le scale, distinguere tra diverse tipologie di oggetti e porgere un determinato oggetto al suo interlocutore. Oltre che distinguere gli oggetti, è in grado di distinguere i volti umani e le voci [6].



Figura 5:SURENA

Romeo

Romeo nasce come robot umanoide che “studia” per diventare il robot domestico ideale per gli anziani (Figura 6) e persone con ridotte capacità motorie. Alto 140 centimetri per 36 chili di peso. Tutta la piattaforma è stata sviluppata da SoftBank Robotics. Gode di 37 gradi di libertà, può aprire porte, raccogliere oggetti sul tavolo e spostarsi autonomamente da una stanza all’altra della casa. La SoftBank Robotics assicura che Romeo è un perfetto badante, in grado di sostenere una conversazione, ricordare un appuntamento e, persino, in grado di cucinare. Se utilizzato come badante, sarà in grado di imparare tutte le abitudini del suo assistito (per esempio: il dosaggio del farmaco che assume regolarmente), potrà aiutarlo a rialzarsi in caso di caduta e se necessario potrà effettuare chiamate di emergenza. La casa costruttrice promette che entro il 2019 sarà presentata al mercato una versione migliorata di Romeo il quale potrà essere disponibile per case di riposo e ospedali. [7].



Figura 6:ROMEO

R1

R1 (Figura 7) è stato programmato per svolgere una serie di compiti per gli utenti o per fornire loro una mano in alcune attività domestiche e lavorative. È alto 1 metro e 25 centimetri ma, può allungarsi di altri 20 centimetri, fino a raggiungere il metro e quarantacinque di altezza. Anche le sue braccia sono allungabili di 13 centimetri e come iCUB possiede una pelle artificiale che ricopre gli avambracci e le mani consentendo al robot la percezione del tocco. Questo robot si muove su due ruote ed è realizzato in parte con materiale plastico e in parte con fibra di carbonio e metallo. Raggiunge i 50 chili di peso. A differenza dell’iCUB, il disegno delle mani è stato semplificato per

garantire robustezza e costi contenuti, pur consentendo l'esecuzione di semplici operazioni domestiche. Il volto è uno schermo LED a colori su cui compaiono delle espressioni stilizzate. All'interno, lo schermo ospita due telecamere e uno scanner 3D, sensori per l'equilibrio e per la generazione e percezione del suono. Nella "pancia" invece, ci sono i tre computer che governano la capacità del robot, dal calcolo al movimento della testa sino al controllo di tutti i sensori. R1 è da considerarsi una piattaforma a basso costo ideata per un'ampia distribuzione nelle case [8].

ALPHA2

Nasce come robot di compagnia, tanto da poter essere considerato un amico o un membro della famiglia. Le sue caratteristiche principali sono: intelligenza, interattività ed espandibilità. Alpha 2 (Figura 8) è alto 43 centimetri circa ed ha un peso di soli 2,3 chilogrammi. Il suo "corpo" è fatto di una lega di alluminio e plastica ABS. Caratteristica di questo robot sono i led che presenta negli occhi, nelle orecchie, nelle mani e nel petto. Per il controllo a distanza è presente una connessione WiFi. Il robot riconosce i comandi vocali e gli input ricevuti tramite app per iOS e Android. Oltre ai microfoni, altoparlanti e videocamere, questo robot integra anche un accelerometro a tre assi, sonar e sensori touch. Infine, presenta, 20 gradi di libertà che gli permettono di eseguire i movimenti con estrema agilità. Ciò che rende utile questo robot è la possibilità di utilizzarlo come tutor o interprete. Può memorizzare note e allarmi, effettuare chiamate, leggere ed inviare messaggi o email, cantare, ballare, pubblicare post sui social network e persino fare yoga [9].

Humanoid

Questo robot, è stato sviluppato dalla Toyota, è destinato a vivere con le famiglie, come robot da compagnia e per anziani. Alto quasi 150 centimetri con peso di circa 40 chilogrammi, è una perfetta combinazione di intelligenza e "umanità". Humanoid, cammina in modo simile ad un essere umano e può afferrare gli strumenti. Un'abilità particolare è quella di saper suonare il violino (Figura 9) e la tromba grazie a un meccanismo che crea delle vibrazioni nelle labbra simili a quelle dell'uomo. Si vuole estendere questa funzionalità anche in ambito infermieristico e medico. Lo sviluppo di questo robot è iniziato nel 2000. Toyota, ha progettato anche una versione di questo robot sulle ruote [10].

Figura 7:R1



Figura 8:Alpha 2



Figura 9:Humanoid

Robi Robot

Robi robot (Figura 10) è un piccolo robot umanoide che effettua movimenti fluidi. Grazie al suo aspetto simpatico e al design accattivante ha conquistato migliaia di famiglie. Alto 34 centimetri, pesa solo un chilogrammo ed è mosso da 20 servomotori. Dotato di due sensori di presenza, ha una camminata sciolta, sa cantare, ballare ed è anche in grado di rispondere. Assemblato da De Agostini Publishing, società del gruppo De Agostini, risponde a 250 comandi vocali in italiano. Si presenta stringendo la mano e quando parla gli occhi e la bocca si illuminano e cambiano colore. Robi è un ottimo intrattenitore soprattutto con i bambini con i quali gioca facendo loro degli indovinelli o a morra cinese. Robi è dotato di un telecomando ad infrarossi con cui è in grado di accendere, spegnere e cambiare canale della tv. Possiede, per cantare e parlare, un altoparlante interno e un sistema di riconoscimento vocale per rispondere alle domande, restituendo un'esperienza naturale. I suoi movimenti fluidi sono dati da una tecnologia specifica brevettata dalla ROBO-GARAGE. Non ha un'eccessiva diffusione in quanto il prezzo è elevato, ovvero 1300 euro [11].



Figura 10:Robi

Walk-man

Walk-man (Figura 11) è un robot umanoide italiano, messo appunto da 25 giovani ricercatori tra Liguria e Toscana. Questo robot pesa 118 chilogrammi ed è alto 185 centimetri ma, nonostante le sue dimensioni ed il suo peso, è in grado di destreggiarsi in luoghi stretti e affollati, camminare su superfici irregolari e strisciare su cumuli di detriti. L'innovazione apportata su questo robot è la capacità di integrare una mano robotica poliarticolata ad una polifunzionale che gli permettono quindi, grazie anche ai tendini artificiali, di essere estremamente flessibili come quelle dell'uomo. Walk-man può aprire le porte, adoperare attrezzi da lavoro come trapani, martelli pneumatici e persino guidare piccoli fuoristrada. È realizzato quasi interamente da una lega di alluminio e zinco che lo protegge dagli urti. Ha un busto ruotabile di 180 gradi e braccia retrovertibili che gli consentono di sollevarsi in caso di caduta, un'essenziale funzionalità visto l'ambito di applicazione in cui questo robot opera. L'obiettivo degli sviluppatori è quello di migliorarlo rendendolo a disposizione della protezione civile o dei vigili del fuoco per intervenire in zone colpite da calamità naturali o posti in cui l'ingresso dell'uomo è da considerarsi troppo pericoloso [12].

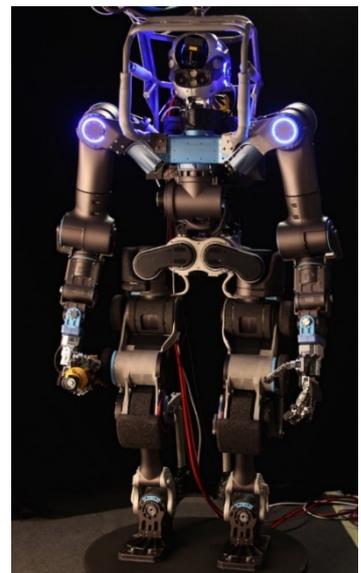


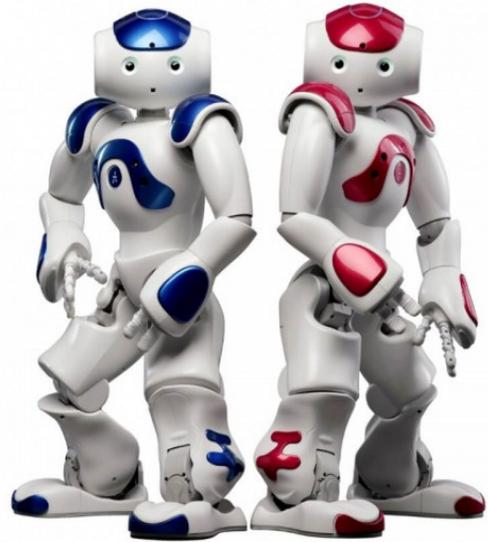
Figura 11:Walk-Man

Nao

Nao (Figura 12) nasce dalla stessa azienda di Pepper ovvero la Aldebaran Robotics. È un robot programmabile, open source, alto 58 centimetri, creato nel 2006, ma in continua evoluzione. Si può programmare in diversi linguaggi come C++, Python e Java. Uno degli obiettivi per cui è nato questo robot è quello di aiutare i bambini autistici a migliorare le loro relazioni, sviluppare attenzione e capacità linguistico-comunicative. È molto apprezzato dai bambini autistici grazie al suo aspetto accattivante. Possiede una voce che parte da un microfono interno ed un volto che

Figura 12:NAO

emana una simpatica espressività grazie ai suoi occhi neri e al tocco di colore blu o rosso che lo caratterizza. Purtroppo, Nao come Pepper, ha un prezzo eccessivo (oltre 7000€), dunque è utilizzato solo da una piccola fetta di mercato. Possiede un sistema multimediale evoluto con 4 microfoni, due altoparlanti e due videocamere che consentono la sintesi vocale, la localizzazione nello spazio ed il riconoscimento facciale [13].



Guida sviluppatori

In questo paragrafo tratteremo solo ed esclusivamente la parte di programmazione e sviluppo del software. Parte che più concerne la nostra sperimentazione e studio con Pepper. Si prenderà spunto dal sito Aldebran [14] e dall'esperienza maturata.

SoftBank Pepper

Sviluppo

SoftBank Robotics (ex Aldebaran Robotics) è una società di robotica, tra i leader mondiali nel campo della robotica umanoide, in particolare nella sfera professionale.

L'azienda ha sviluppato diversi modelli di robot, tra cui gli umanoidi NAO, Roméo e Pepper in collaborazione con la SoftBank giapponese. Questa "famiglia di robot" ha diversi obiettivi: NAO è orientata verso la programmazione, l'insegnamento e la ricerca, Roméo ha lo scopo di assistere gli anziani e Pepper nelle relazioni con i clienti o con gli utenti.

Secondo la comunicazione ufficiale, la visione dell'azienda è centrata sulla volontà di creare robot per il bene degli uomini, con l'obiettivo finale di commercializzare i suoi robot al pubblico come una "nuova specie che è docile con gli esseri umani".

Robot

Pepper è un robot creato da Aldebaran Robotics e Softbank Robotics e, come da loro parole, è "il primo robot in grado di leggere le emozioni".

Come detto precedentemente, è progettato per lavorare nel servizio clienti, accogliere, informare e divertire i clienti in modo gentile e familiare. Pepper è particolarmente popolare in Giappone, dove è presente in centinaia di negozi e sta iniziando ad essere utilizzato nei negozi di tutto il mondo. SoftBank sostiene che Pepper ha aiutato alcuni negozi negli Stati Uniti a crescere nelle vendite e nel traffico pedonale. SoftBank afferma che ci sono oltre 10.000 robot Pepper in uso in tutto il mondo.

Chiunque può partecipare allo sviluppo di Pepper se possiede un robot con una licenza legale, per questo motivo SoftBank ha creato un sito web (<https://developer.softbankrobotics.com>) da condividere con altri sviluppatori[15].

Di seguito, mostriamo le maggiori specifiche tecniche di Pepper.

Dimensioni (mm)	Altezza: 121 cm; larghezza: 48 cm; profondità: 42,5 cm
Peso	28kg / 62lb
Batteria	Lithium-ion, 30 Ah/795 Wh/12 ore
Tablet	LG CNS Tablet; 10.1"
Testa	4x microfoni; 2x Telecamere RGB; 1x Sensori 3D; 3x Sensori Tattili
Torso	1x Giroscopio; 1x Accelerometro
Mani	2x Sensori tattily
Piedi	6x Sensori laser; 2x Sensori infra-rosso; 1x Sonar; 3x paraurti
Componenti motorie	20 motori
Velocità di movimento	Fino a 3 Km/h
Networkiing	1x Ethernet (1xRJ45 – 10/100/1000 base T); 1x WiFi (IEEE 802.11 a/b/g/n)
Sistema Operativo	NaoQI OS

NAOqi OS

NAOqi OS (figura 13) è il *sistema operativo* del robot. È una distribuzione GNU / Linux basata su Gentoo. È una distribuzione embedded appositamente sviluppata per adattarsi alle esigenze del robot Aldebaran. Fornisce e gestisce un numero di programmi e librerie, tra questi, tutto quello richiesto da NAOqi, il software che dà vita al robot.

Il robot è controllato usando le sue API. Questi sono usati in **Choregraphe Suite** o in uno qualsiasi degli SDK (Python, C ++, Java, Javascript, ecc.) forniti da SoftBank Robotics.



Figura 13: NAOqi

Le **principali API** sono le

seguenti:

- **NAOqi Core:** viene fornito con un elenco di moduli principali sempre disponibili. Come ad esempio: ALBehaviorManager: che avvia e stoppa i comportamenti, ALUserInfo che gestisce i dati persistenti sugli utenti.
- **NAOqi Interaction engines:** Alcuni moduli chiave, in esecuzione sul tuo robot *Aldebaran*, definiscono il suo comportamento interattivo. Lo sviluppo di applicazioni e le attività in esecuzione su un robot *Aldebaran*, comprende lo scopo di questi moduli per trarne vantaggio. Come ad esempio: ALAutonomousBlinking: Consente al robot di far lampeggiare i LED degli occhi quando vede qualcuno e quando interagisce. ALBasicAwareness: Permette al robot di reagire all'ambiente per stabilire e mantenere il contatto visivo con le persone.
- **NAOqi Motion:** è lo strumento principale che consente al robot di muoversi. Sviluppatori e animatori avanzati possono trarre vantaggio dai suoi numerosi metodi per sviluppare il proprio movimento.
- **NAOqi Audio:** Gestisci ingressi e uscite audio. Utile per far recepire al robot le domande fatte dall'utilizzatore. ALAudioPlayer: Riproduci file audio sul robot. ALAudioRecorder: Registra file audio sul robot.
- **NAOqi Vision:** Fai in modo che il robot impari e riconosca i modelli visivi: oggetti, immagini.
- **NAOqi People Perception:** Questa sezione descrive i moduli che possono essere utilizzati per analizzare le persone attorno al robot. Per esempio: ALEngagementZones: Analizza la posizione di una persona rispetto al robot. Oppure ALFaceDetection: Fai in modo che il robot rilevi e riconosca i volti umani, modalità molto usata. ALGazeAnalysis: Analizza la direzione dello sguardo di una persona.
- **NAOqi Sensors & LEDs:** è responsabile dell'attivazione degli eventi. ALBattery: relativo all'hardware della batteria del robot. Oppure ALSensors: relativo ai sensori del robot.
- **DCM:** Il DCM è un modulo software, parte di **NAOqi**, che si occupa della comunicazione con quasi tutti i dispositivi elettronici nei robot (schede, sensori, attuatori ...), ad eccezione del suono (interno o esterno) e delle telecamere. Gestisce i principali bus di comunicazione tra la testa e il corpo e anche la comunicazione all'interno della testa. Il **DCM** ha due scopi principali: Invia comandi agli attuatori e Aggiornare i valori dell'attuatore e del sensore. **Attuatori** e **sensori**, da un punto di vista **DCM**, potrebbero essere attuatori "fisici" (posizione dei giunti, LED, corrente del motore ...) di "virtuali" (come la rigidità che è un parametro fuori dal circuito di controllo) [15].

In figura 14, uno schema del funzionamento di NAOqi.

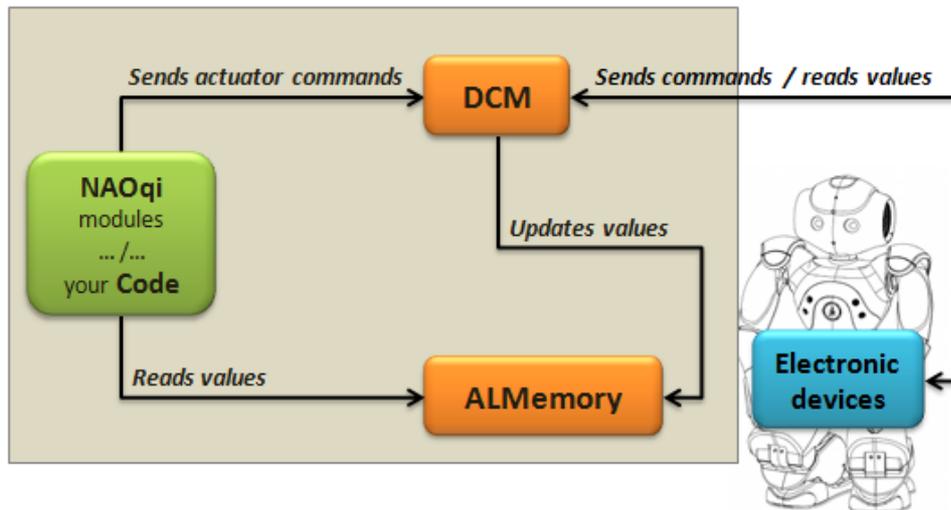


Figura 14: Funzionamento elementare di NAOqi

Prima configurazione Pepper

Rimozione Pepper dall'involucro

La rimozione del robot dal proprio involucro deve essere fatta con massima cura e delicatezza. Con la massima attenzione, l'involucro deve rimanere in piedi possibilmente in VERTICALE.

I passi corretti sono i seguenti:

1. Fissare il lembo inserendolo nello slot situato sulla parte superiore della scatola.
2. Conservare la scatola in un luogo sicuro per la spedizione, i resi, le riparazioni. Rimuovendo delicatamente la prima parte in cartone.
3. Aprire la parte inferiore in cartone in modo tale da formare una piccola pedana
4. Porre le mani sotto i fianchi del robot per farlo scivolare delicatamente.
5. Assicurati di avere una presa sicura e, idealmente, anche con l'aiuto di un amico.

Figura 15: Rimozione dall'involucro



6. Il busto deve necessariamente stare all'indietro e la testa in basso.
7. Rimuovere i pin in plastica azzurra dal bacino, per poter mobilitare il robot. (Non reinserire i pin quando Pepper è in stato ON.)
8. Rimuovere i pin in plastica azzurra dalle gambe, per poter mobilitare il robot. (Non reinserire i pin quando Pepper è in stato ON.)

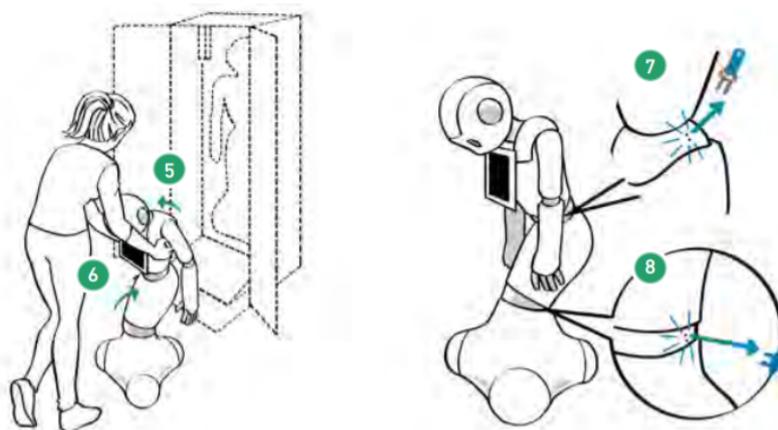


Figura 16: Rimozione del robot e dei pin

9. Infine, riponi i pin azzurri rimossi in precedenza, aprendo la parte gommata sulle spalle. Inserendoli negli appositi fori. Come nell'illustrazione il punto 1 e 2. Come in figura in figura 17.

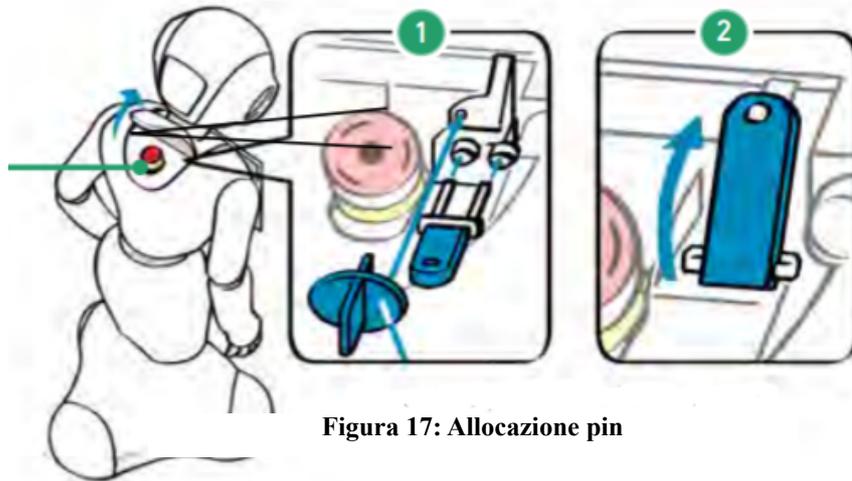


Figura 17: Allocazione pin

Tutte queste azioni possono essere fatte al contrario per riportarlo. [16]

Attivazione Pepper

Il pulsante Stop è **bloccato** in *push*, quando si utilizza il robot per la prima volta. È necessario quindi, **sbloccare** il pulsante Stop per poter attivare Pepper in modalità ON.

Lo si può fare facilmente girando il pulsante rosso in senso ORARIO (punto 1) e tirare il bottone rosso verso l'alto per porlo in stato *pop* (punto 2). Come in figura 18.

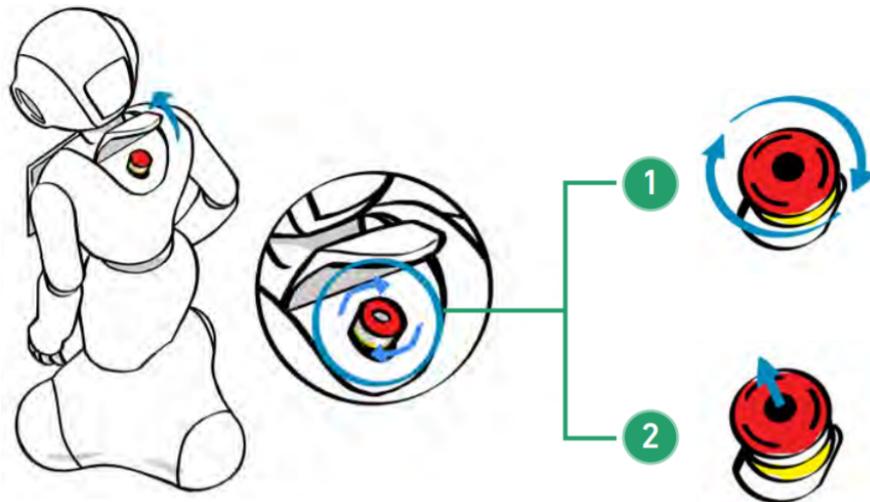
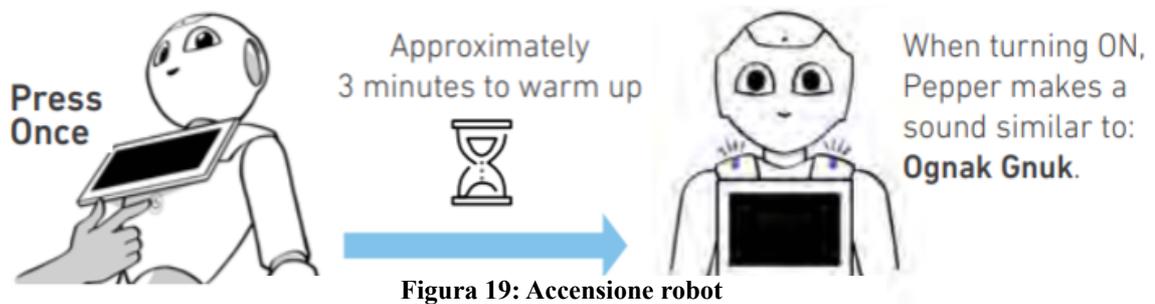


Figura 18: Attivazione robot

Tutte queste azioni possono essere fatte al contrario per riportarlo. [16]

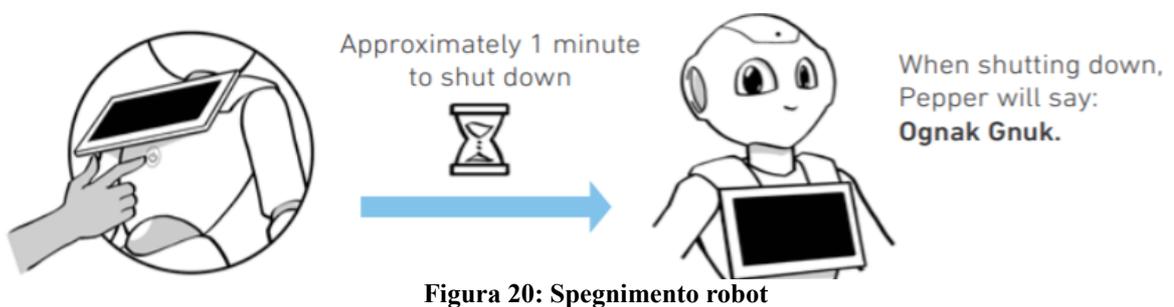
Turning Pepper ON

Per accendere Pepper, premere il pulsante sotto il tablet una volta. Attenderai per circa 3 minuti l'accensione. Comanderai che sarà acceso perché vedrai il marchio di fabbrica apparire sul tablet e le luci attivarsi sulle spalle del robot. Come in figura 19. [16]



Turning Pepper OFF

Premere e tenere premuto per circa 3 secondi, quindi rilasciare. Attenderai circa 1 minuto lo spegnimento totale. Quando si spegne, Pepper dirà: *Ognak Gnuak*. Come in figura 20. [16]



Come connettere Pepper in rete la prima volta

Dopo che il robot si sia acceso correttamente, rispettando quanto detto in precedenza, bisognerà configurarlo e connetterlo alla propria rete. Bisognerà connettere **il proprio pc e il robot alla stessa sottorete**, per poter procedere.

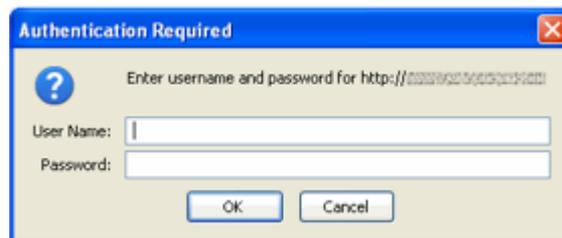
1. Aprire la regione dietro la nuca del robot. Troverai la connessione Ethernet. Connetti il robot e il pc con un cavo Ethernet. **(Esegui questo passaggio solo se vorrai aggiungere una nuova rete al robot)**
2. Premere una sola volta il bottone di accensione sotto il tablet.
3. Ascoltare l'indirizzo ip che pepper pronuncerà, premere nuovamente il bottone per farlo pronunciare più lentamente. L'IP sarà pronunciato in lingua inglese. Annotalo da qualche parte.
4. Dopo aver annotato l'indirizzo, apri un browser Web qualsiasi e inserisci l'**indirizzo IP** nella barra degli indirizzi. Come in (figura 21).

Figura 21: Annotazione IP barra indirizzi



5. Verrà visualizzata una finestra di autenticazione. Come in figura 22:

Figura 22: Autenticazione robot



Rispettivamente:

- **User name:** nao
- **Password:** malerba

Il valore predefinito della password è *nao*. Se hai dimenticato la password, vedi: [Hai perso la password](#). [17]

6. Ti comparirà la seguente schermata, come in figura 23:

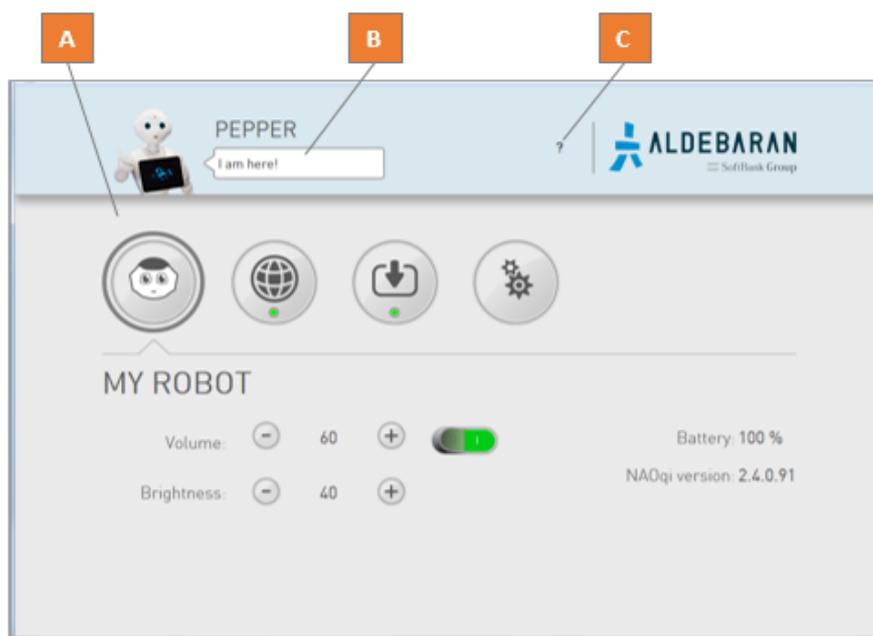


Figura 23: Schermata remota robot

Parte	Nome	Ti permette di ...
A	Menu Impostazioni	Per l'accesso remoto a Pepper
B	Speech bubble	Fai in modo che il tuo peper dica una frase breve: <ul style="list-style-type: none"> • Fai clic sulla vignetta. • Scrivi del testo. • Poi, premi il tasto Invio.
C	? menu	Aperto: <ul style="list-style-type: none"> • the Getting started wizard • on-line Documentation • Former Pepper Web page

La sezione che ci interesserà sarà l'icona con il mondo, cliccaci su. Questa sezione denota le impostazioni di rete. Come illustra la figura 24. Se è già presente una rete alla quale ci si può connettere allora basterà cliccarci su per far connettere anche il robot. Come nell'esempio in figura 24 la rete *Hello* o *Now*.

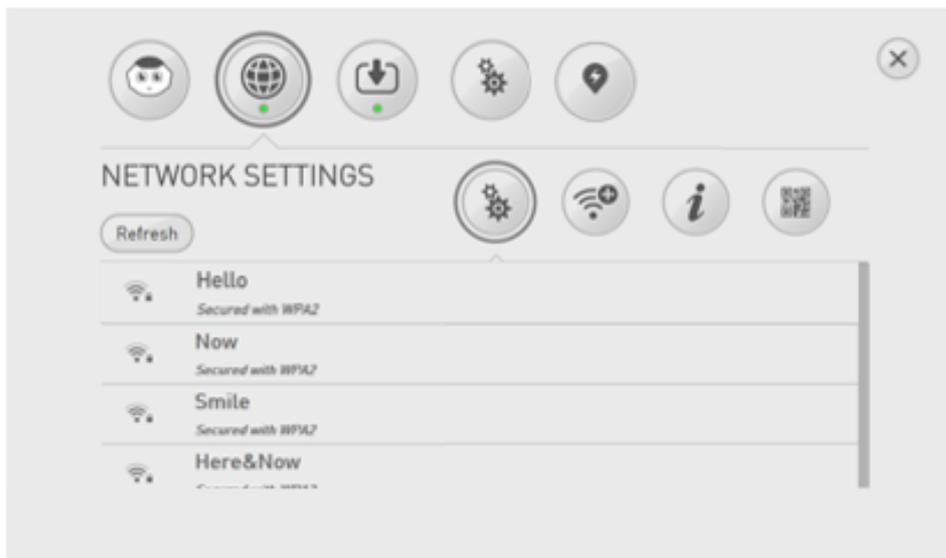


Figura 24: Schermata Impostazioni di Rete

7. Altrimenti se vorrai aggiungere un'altra sottorete come connessione tra il tuo pc e il robot, allora basterà cliccare sull'icona del Wi-Fi per aggiungerla. Basterà inserire il nome della nuova sottorete nel campo SSID, con Sicurezza WPA2 e con Password della sottorete. Tutto mostrato in figura 25.

IMPORTANTE: Il pc e il robot devono essere connessi alla stessa sottorete.



Figura 25: Inserimento nuova sottorete

8. Poi, fai clic sul pulsante **Connect**.

9. Rimuovi il cavo Ethernet, se lo hai inserito inizialmente, e il gioco è fatto. [18]

Download e installazione del software Aldebaran

Senza un robot **Aldebaran** non sarà possibile scaricare alcun software ad eccezione della versione di prova di **Choregraphe**.

Per scaricare il software dal sito Web della **comunità di Aldebaran**:

1. Assicurati che il tuo **NAO** sia registrato sul tuo account utente di **Aldebaran Community**. Consultando [Registering your Pepper](#).
2. Se non hai un account, dovrai registrarti qui: (<https://community.ald.softbankrobotics.com/en/resources/software/language/en-gbv>). Come illustrato nella figura 26. Cliccando su *Create Account*.

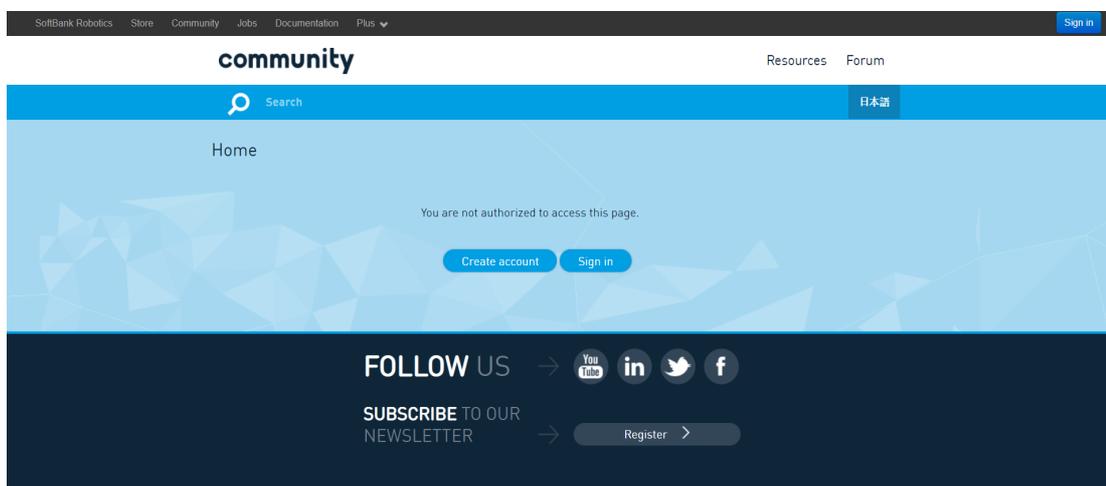


Figura 26: Creazione Account

3. Dopo aver creato l'account, si potrà accedere al proprio profilo. Cliccando poi sulla spunta Choregraphe. Possiamo installare la versione del sistema in base al nostro sistema operativo, come illustrato in figura 27.

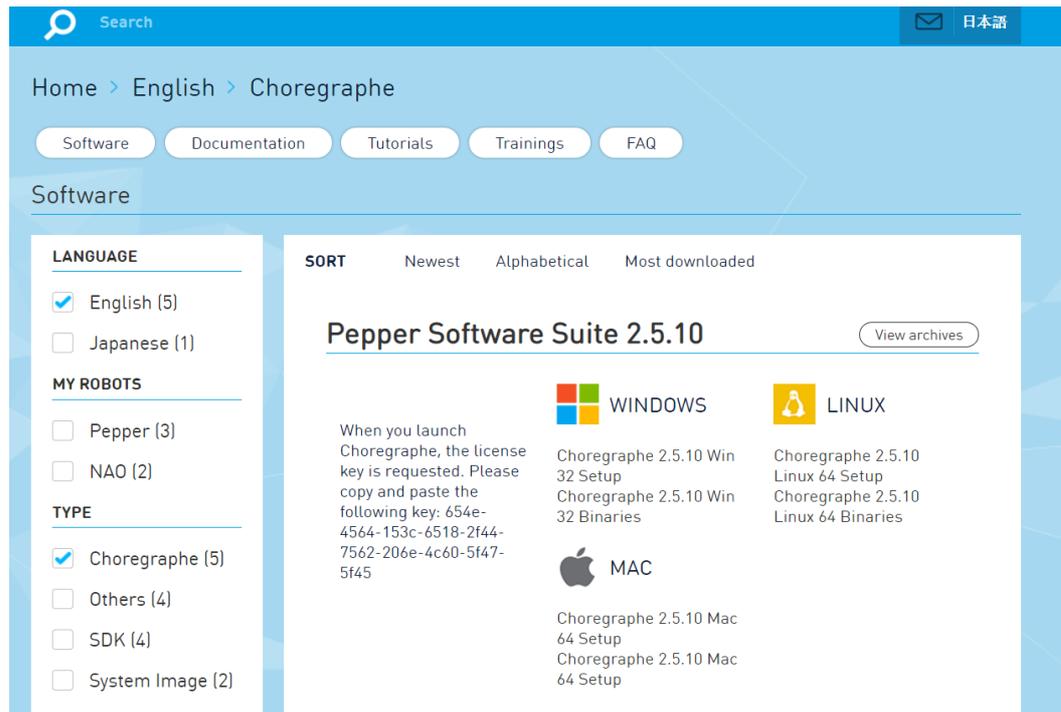


Figura 27: Profilo Account

Dopo l'installazione, all'avvio del programma verrà richiesta una chiave di licenza: puoi immettere la seguente:

654e-4564-153c-6518-2f44-7562-206e-4c60-5f47-5f45

Choregraphe Suite

Choregraphe Suite è un'applicazione desktop multiplatforma che consente all'utente o allo sviluppatore di lavorare con Pepper in modo semplice, riducendo le competenze di programmazione necessarie con un'interfaccia amichevole e una creazione del comportamento di facile comprensione.

Permette all'utente di fare quanto segue:

- Creare animazioni, comportamenti e dialoghi,
- Provarle su un robot simulato, o direttamente su un robot reale,
- Monitoraggio e controllo del robot,
- Arricchire i comportamenti di Choregraphe con il codice Python.

Requisiti per Choregraphe:

Un computer collegato a un punto di accesso WiFi, con:

- CPU da 1,5 GHz
- 512 MB di RAM
- Scheda grafica OpenGL certificata [19]

SO supportati:

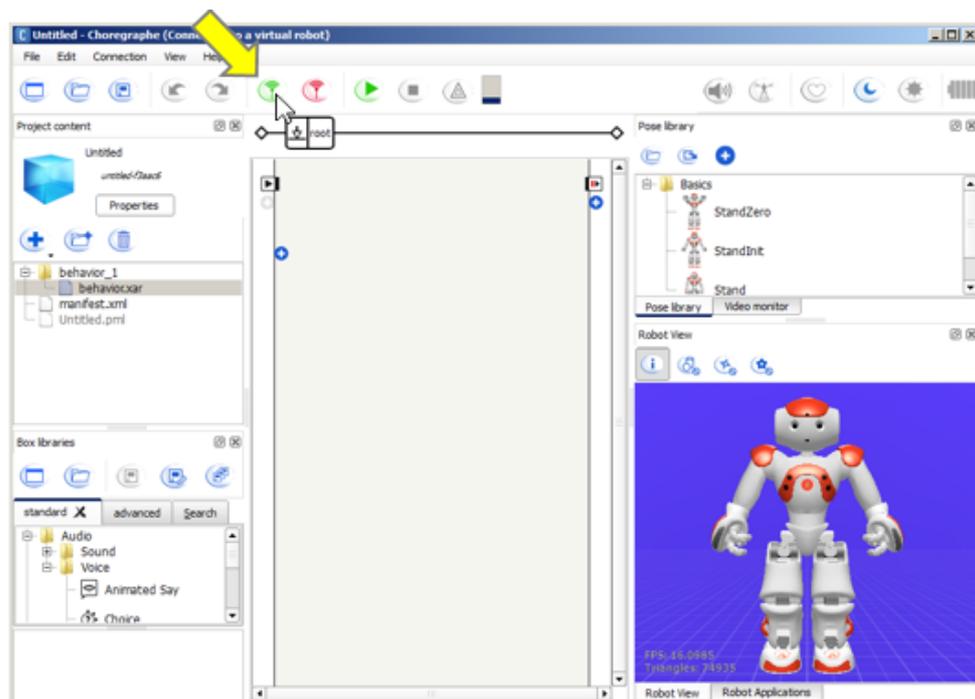
Linux	Ubuntu 14.04 (Trusty Tahr) e successivi
finestre	Microsoft Windows 7 e 8.1
Mac	Mac OS X 10.10.2 Yosemite

Prima di iniziare:

- Assicurati di avere un robot pronto per l'uso.
- Assicurati che **Choregraphe** sia installato sul tuo computer.

Passi da eseguire con *Choregraphe*:

1. Esegui **Choregraphe**
2. **Prima di procedere, dovresti controllare che il settaggio della rete di pepper sia connesso alla stessa sottorete del tuo pc.**
3. Ora hai la necessità di dover connettere il tuo pepper al programma. E molto semplice. Clicca sul pulsante verde "Connect to..." per connettere il tuo pepper come mostrato in figura 28.



4. Comparirà una nuova schermata che ti permetterà di cliccare la tua versione del pepper e di poter immettere l'ip del robot e la sua porta. Consigliamo di mantenere la porta settata con default. Mentre l'IP deve essere pronunciato da pepper, premendo il pulsante che ha sotto il tablet. Come illustrato in figura 29.

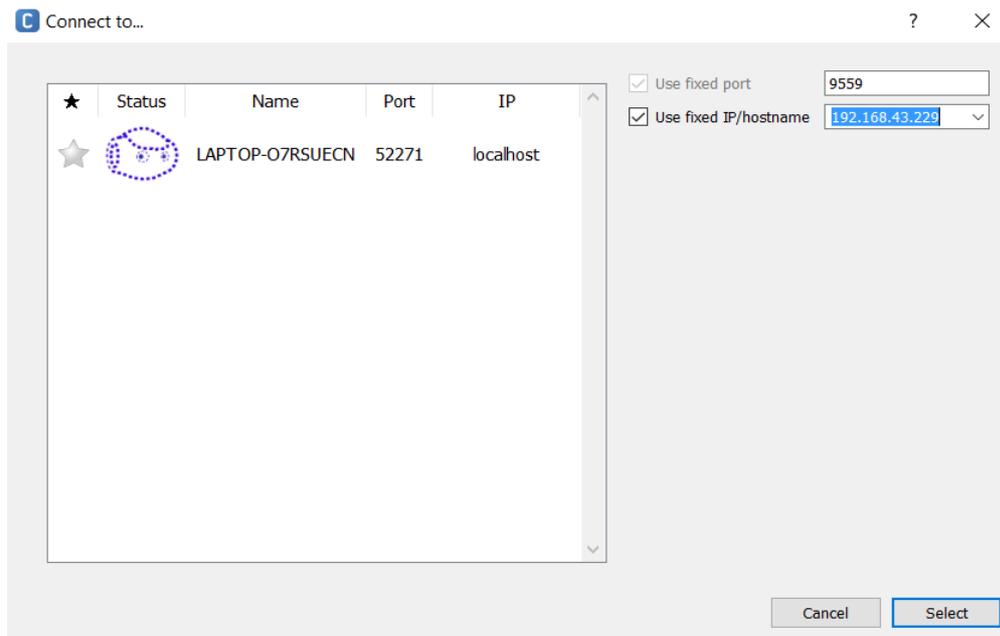


Figura 29: Indirizzo IP

5. Dopo di che, si cliccherà sul pulsante
 “Select” e il programma verrà connesso al robot. Te ne accorgerai che sarà connesso solo quando noterai in alto a sinistra la frase (Connected to “IPaddress”). Il tuo robot poi apparirà nella **vista Robot** (Robot view).
6. Ora possiamo notare nella parte sinistra del programma, il project Content Panel. Che è il luogo per la creazione e archiviazione di file di progetto, che possono essere file audio, video, file di immagine. Si possono creare file di directory, si possono creare nuovi comportamenti del robot, nuovi dialoghi che può fare e gestire. Si possono pianificare anche i comportamenti del robot. Vedere figura 30.

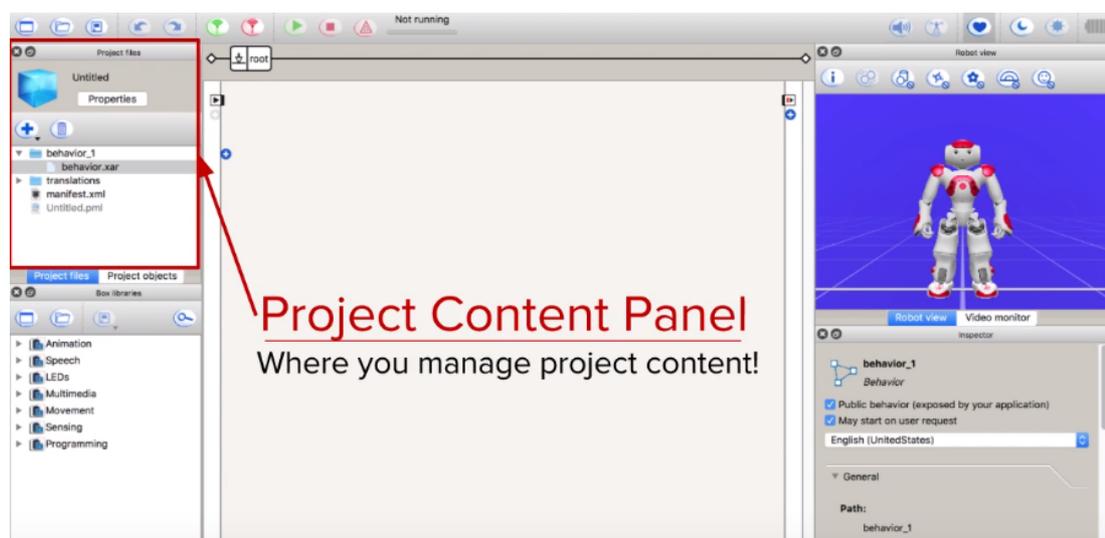
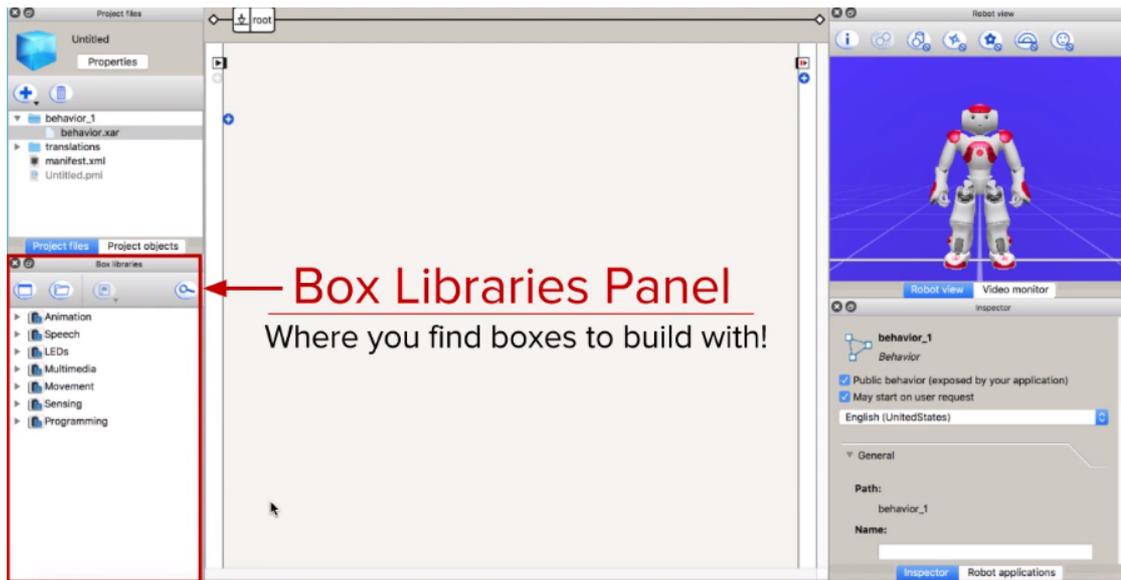


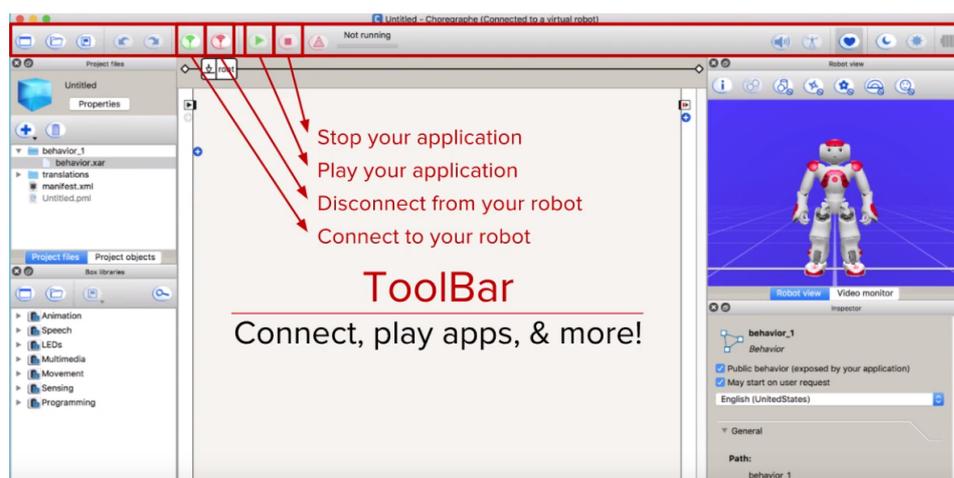
Figura 30: Pannello di Controllo

7. Visualizza le **Box Libraries** come in figura 31, che sono lo strumento principale per arricchire diversi comportamenti del robot. È possibile utilizzare le boxes predefinite per utilizzare semplici programmi, oppure è possibile anche creare nuovi movimenti e nuove boxes con nuove gestualità del robot. Puoi anche scrivere il tuo codice al loro interno, ed è possibile farlo in Python. Le



directory sono: *Animation, Speech, LEDs, Multimedia, Movement, Sensing, Programming.*

- La seguente ToolBar, consente di avviare la connessione con il robot, far partire la propria applicazione e disconnettere il robot stesso. Consente all'utente di connettersi a un robot virtuale o a un robot reale utilizzando la rete. Per **connettersi** al robot reale, è necessario essere connessi alla stessa rete del robot. Il tasto con il cuoricino, identificato con " **Autonomous Life On/Off**" è l'applicazione che dà vita al Robot. Il robot diventa visivamente vivo: "respira", si muove ed è consapevole del suo ambiente. Oltre a questa funzionalità di autonomia, si aggiunge anche la modalità programmabile, deselegzionando il *cuoricino* e attivando l'icona con il *sole* che decreta il " **Wake up**". Mentre il tasto " **Play**", carica il comportamento corrente sul robot in uno slot di memoria temporaneo e lo esegue. Vedere figura 32.



- Posizionando diverse boxes qui è possibile creare programmi in modo semplice. Il loro collegamento lo rende visivo e informativo per lo sviluppatore. Come per esempio trascinare la box (**Voce > Creazione > Pronuncia**) sulla griglia. Vedere figura 33.

Figura 32: ToolBar

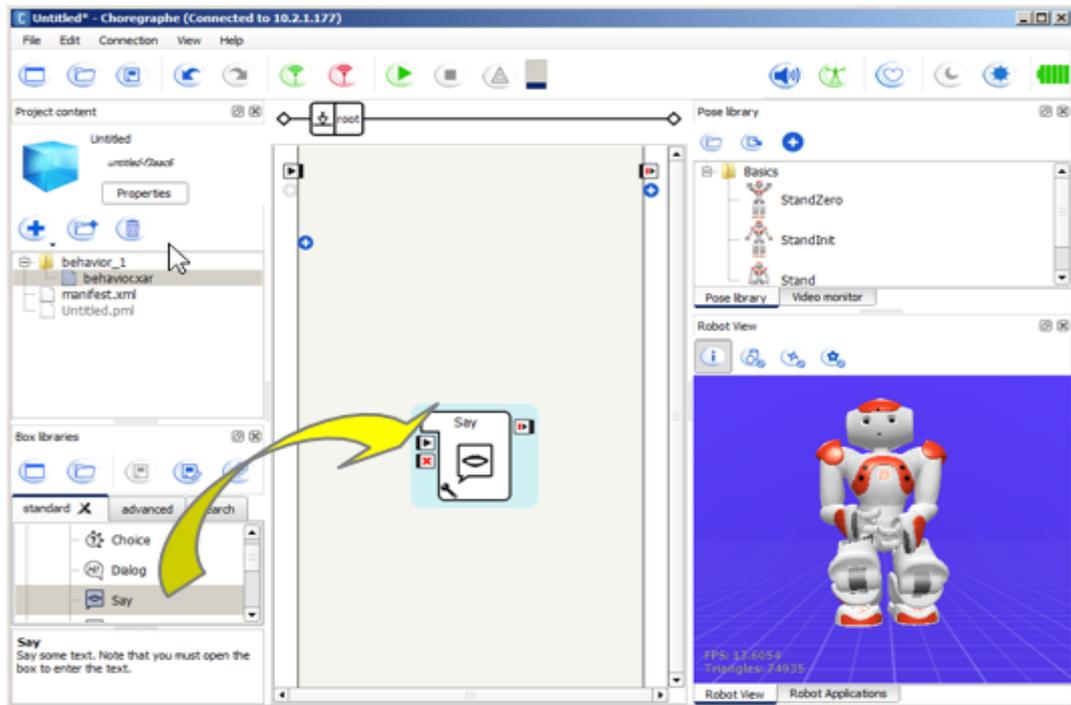


Figura 33: Boxes

10. Si va a creare un link in seguito. Vedere figura 34.

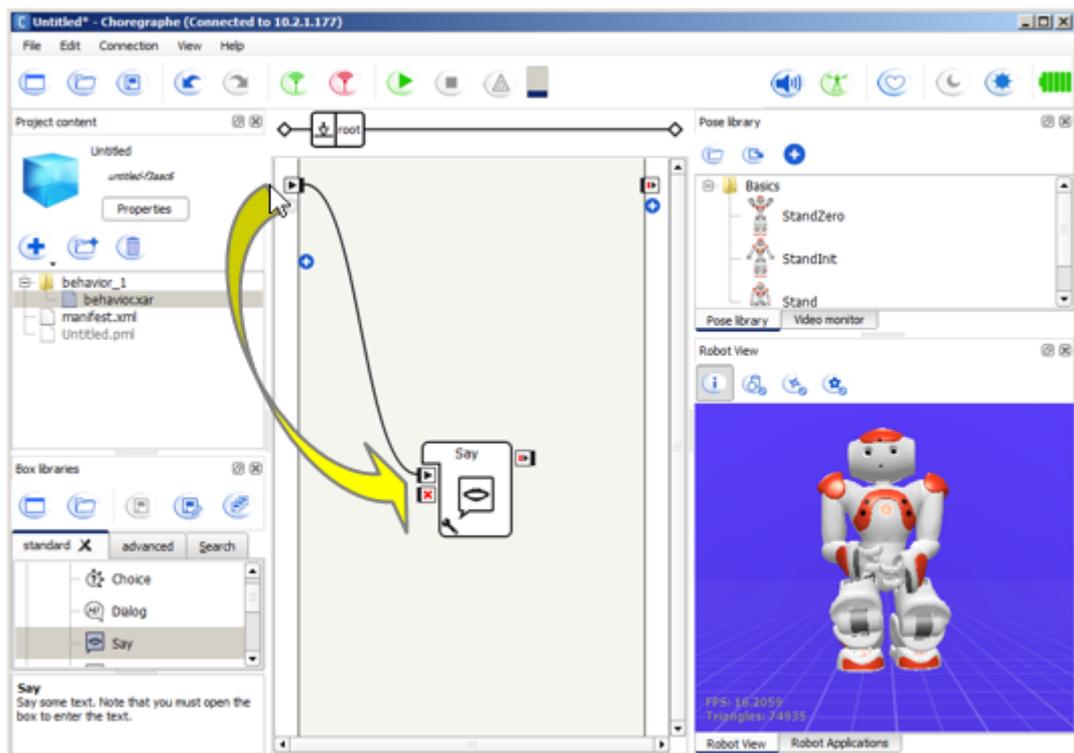
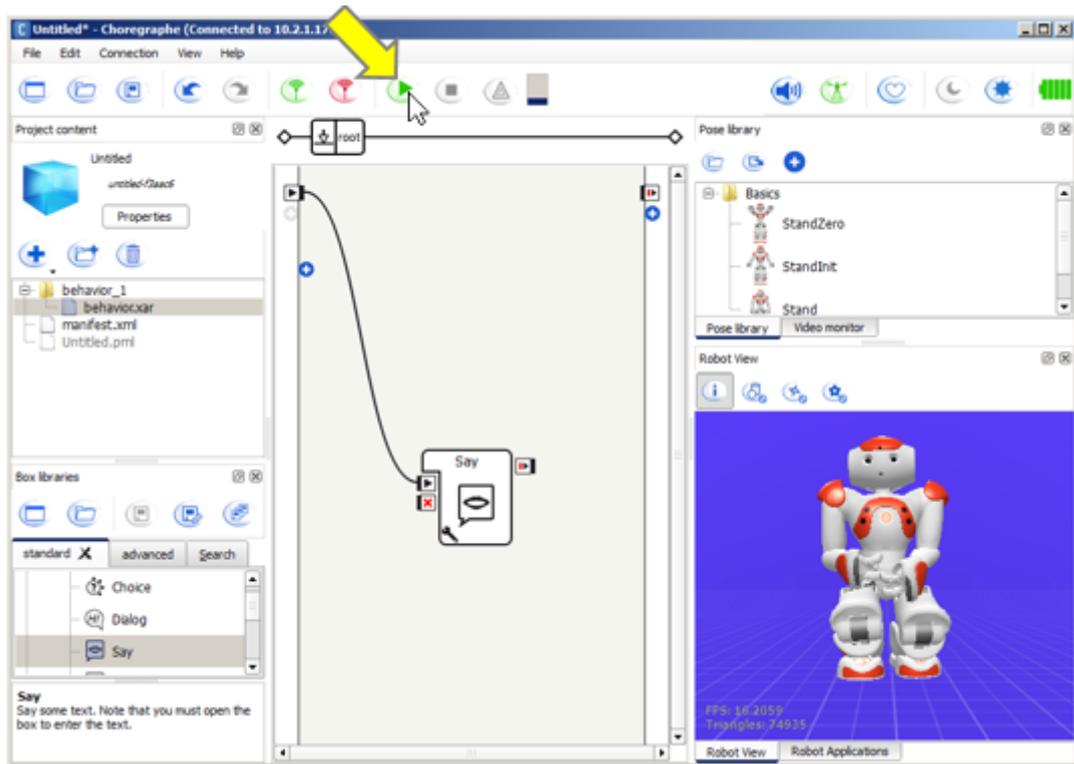


Figura 34: Creazione Link

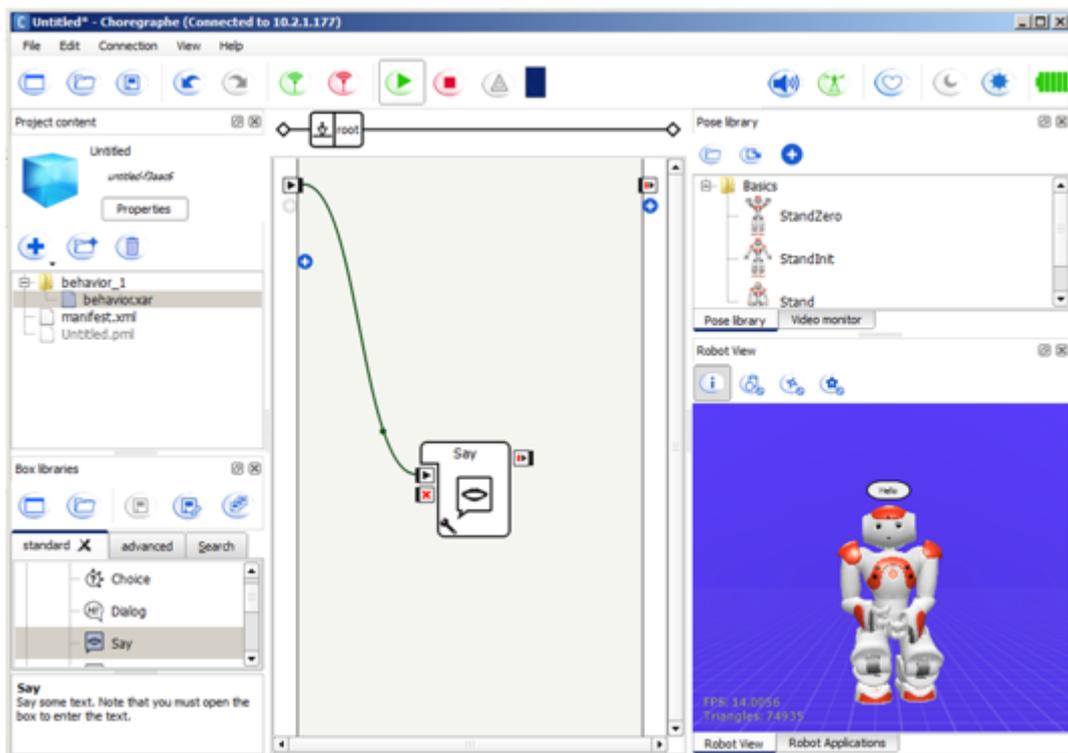
11. Ed infine si può eseguire, cliccando sul tasto "Play". Vedere

figura 35.



12.11

tuo robot dirà "Ciao". Nel frattempo, la **Robot View** mostrerà il messaggio, come in figura 36.



Lingue supportate

Figura 36: Robot View

Il livello di supporto e per tutte le lingue. Di

integrazione non è lo stesso seguito in figura 37

mostriamo una tabella che mostra le differenti caratteristiche per le diverse lingue riconosciute da Pepper.

Codification			Choregraphe		NAOqi API		Basic Channel		Remote	
Locale	Language	Dialog code	Tools	Content and Samples	Notifications	ASR Engine	TTS Engine	Activity launcher	Conversation	Remote ASR Engine
ja_JP	Japanese	jpj	✓	✓	✓	✓	AiTalk	✓	✓	✓
en_US	English	enu	✓	✓	✓	✓	Nuance	✓	✓	✓
fr_FR	French	frf	✓	✓	✓	✓	Nuance	✓	✓	⊗
it_IT	Italian	iti	✓	✓	✓	✓	Nuance	✓	⊗	⊗
de_DE	German	ded	✓	✓	✓	✓	Nuance	✓	⊗	⊗
es_ES	Spanish	spe	✓	✓	✓	✓	Nuance	✓	⊗	⊗
zh_CN	Chinese	mnc	✓	✓	•	✓	Nuance	✓	⊗	⊗
nl_NL	Dutch	dun	✓	✓	•	✓	Nuance	•	⊗	⊗
ar_SA	Arabic	arw	✓	✓	•	✓	Nuance	•	⊗	⊗
ko_KR	Korean	kok	✓	✓	•	⊗	⊗	•	⊗	⊗
pl_PL	Polish	plp	✓	✓	•	⊗	⊗	•	⊗	⊗
pt_BR	Brazilian	ptb	✓	✓	•	⊗	⊗	•	⊗	⊗
pt_PT	Portuguese	ptp	✓	✓	•	⊗	⊗	•	⊗	⊗
cs_CZ	Czech	czc	✓	✓	•	⊗	⊗	•	⊗	⊗
da_DK	Danish	dad	✓	✓	•	⊗	⊗	•	⊗	⊗
fi_FI	Finnish	fif	✓	✓	•	⊗	⊗	•	⊗	⊗
sv_SE	Swedish	sws	✓	✓	•	⊗	⊗	•	⊗	⊗
ru_RU	Russian	rur	✓	✓	•	⊗	⊗	•	⊗	⊗
tr_TR	Turkish	trt	✓	✓	•	⊗	⊗	⊗	⊗	⊗
nn_NO	Norwegian	nor	✓	⊗	•	⊗	⊗	⊗	⊗	⊗
eL_GR	Greek	grg	✓	⊗	•	⊗	⊗	⊗	⊗	⊗

✓	OK
•	Partial / Not tested
⊗	Not Supported Yet

Figura 37: Supporto lingue

Come si può notare dall'immagine, le lingue maggiormente supportate sono il giapponese e l'inglese. Subito dopo vi è il francese, lingua dell'azienda produttrice di Nao e Pepper, e subito dopo vi è l'italiano.

Per quanto riguarda l'italiano, sono implementati tutti i tool e i content del software Coreographe, tutte le diverse API, ma non contempla il basic channel Conversation e il Remote ASR Engine. [20]

Python in Choregraphe

Il linguaggio **python** può essere inserito facilmente nei box. Si può seguire questa breve illustrazione che ti aiuterà a capire come inserire comandi **python** facilmente nelle box che creerai.

1. Esegui **Choregraphe**
2. Fai clic sul pulsante **Connetti** come in figura 38 e collega **Choregraphe** al tuo robot, se non lo avevi già fatto. Se non riesci a fare la connessione nel paragrafo precedente è stata spiegata nel dettaglio.

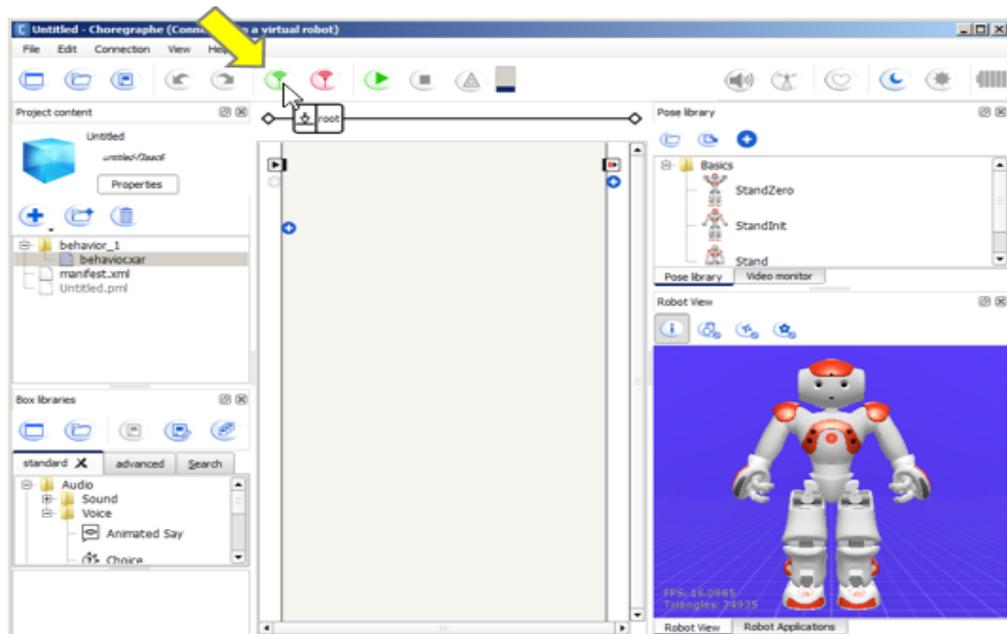


Figura 38: Connessione Robot

3. Il tuo robot poi apparirà nella vista Robot.
4. Fai clic con il pulsante destro del mouse sulla griglia e scegli (**Crea una nuova casella > Python ...**) nel menu come in figura 39.

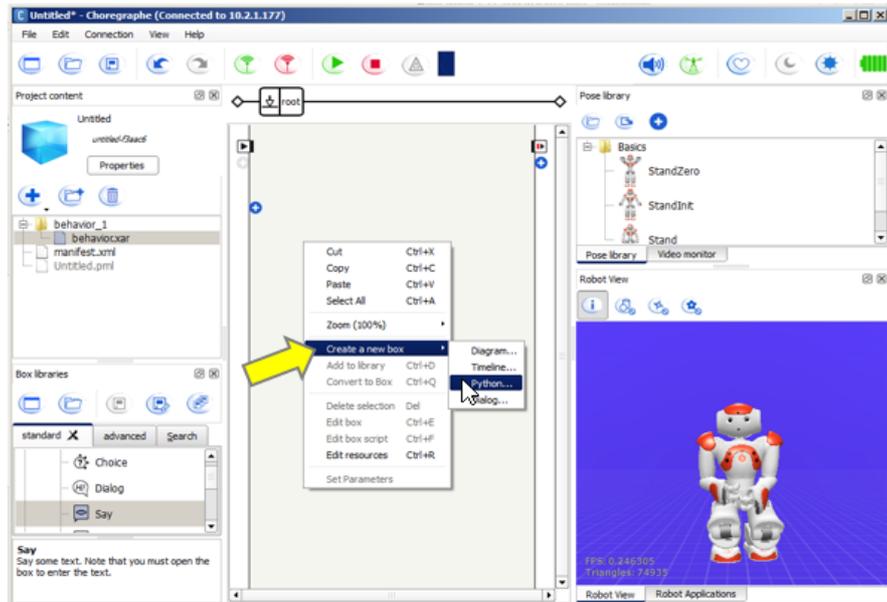


Figura 39: Connessione Robot

5. Digita un nome, ad esempio: My hello world, e fai clic sul pulsante OK. Vedere figura 40.

```

class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)

    def onLoad(self):
        #put initialization code here
        pass

    def onUnload(self):
        #put clean-up code here
        pass

    def onInput_onStart(self):
        #self.onStopped() #activate the output of the box
        pass

    def onInput_onStop(self):
        self.onUnload() #it is recommended to reuse the
        clean-up as the box is stopped
        self.onStopped() #activate the output of the box
  
```

Figura 40: Hello world

- Fai doppio clic sulla box che è apparsa. Così facendo, ti apparirà l'**editor dello script**. Vedere figura 41.



Figura 41: Editor Script

- Aggiungi le 2 seguenti righe presenti in figura 42:

After:

```
def __init__(self):
    GeneratedClass.__init__(self)
```

Add:

```
self.tts = ALProxy('ALTextToSpeech')
```

After:

```
def onInput_onStart(self):
    #~ self.onStopped() #~ activate output of the box
```

Add:

```
self.tts.say("Hello young Padawan")
```

Figura 42: Editor script

La prima istruzione crea un oggetto, chiamato **tts**.

- **self**: considera il fatto che i comandi **tts** saranno disponibili non solo localmente, ma anche in tutti i blocchi di codice dello script.
- **tts**: è il nome che abbiamo dato all'istanza dell'oggetto.
- **ALProxy ()** è una classe di oggetti che ti consente di accedere a tutti i metodi di un modulo.
- **ALTextToSpeech** è il nome del modulo di NAOqi che vogliamo utilizzare.

Mentre l'ultima istruzione usa l'oggetto **tts** per inviare un'istruzione al modulo *NAOqi*.

- **self.tts** è l'oggetto che usiamo.
- **say ()** è il metodo.
- *"Hello young Padawan."* è il parametro che utilizziamo come input al metodo.

8. Lo script dovrebbe assomigliare a quello in figura 43.

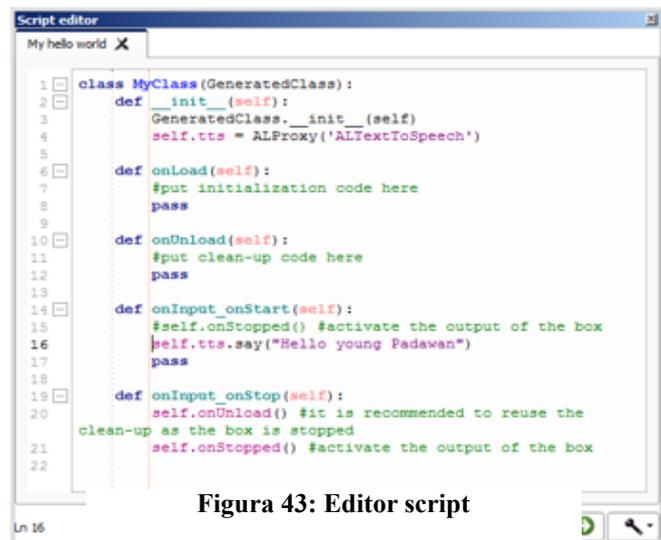


Figura 43: Editor script

9. Fai clic sul pulsante **“Play”**. Collegando il link di sinistra. Vedere figura 44.

link

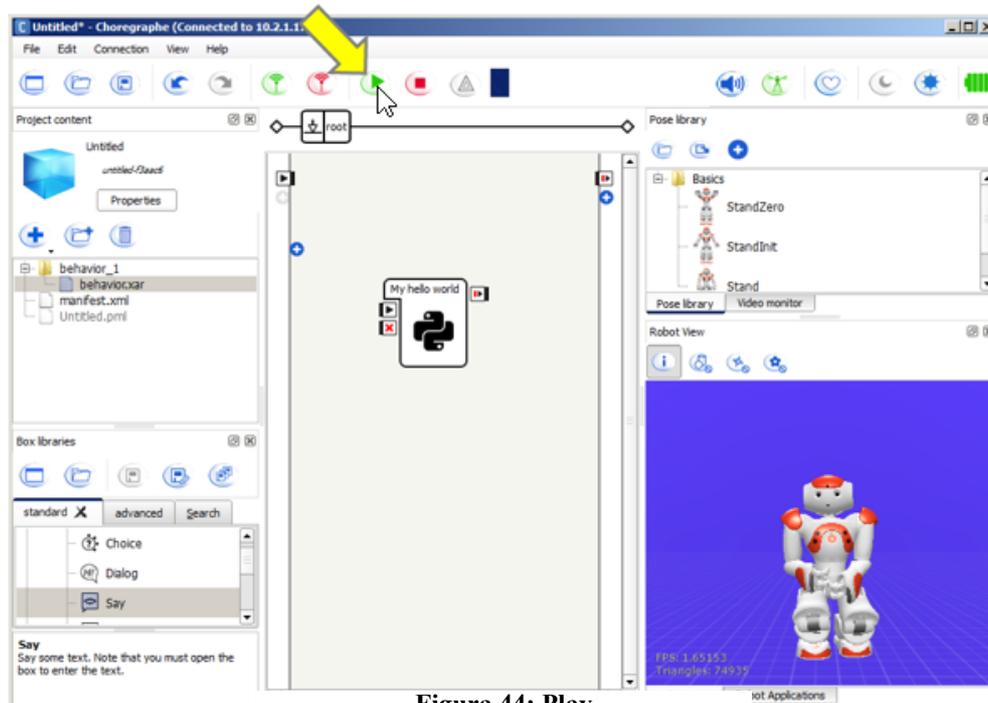


Figura 44: Play

10. Il tuo robot dirà " Hello young Padawan". Nel frattempo, la **vista Robot** mostrerà il messaggio.

Questo script usa il metodo **say** del modulo *ALTextToSpeech*. *ALTextToSpeech* è il modulo di **NAoqi** che permette al robot pronunciare la stringa data come parametro [21].

Per ulteriori dettagli su questo modulo, vedere *ALTextToSpeech* [22].

Dialogo in Choregraphe

La peculiarità di Choregraphe è quella di poter creare tanti dialoghi tra voi e il robot. Grazie a queste istruzioni potrai farlo:

1. Esegui **Choregraphe**
2. Fai clic sul pulsante Connetti e collega Choregraphe al tuo robot, se non lo avevi già fatto. Se non riesci a fare la connessione nel paragrafo precedente è stata spiegata nel dettaglio.
3. Trascinare il (**Speech > Speech Settings > Set Language**) e trascinatelo sulla grigia. Fare clic sul box sulla chiave inglese in basso a sinistra e imposta la lingua come inglese di default. Vedere figura 45.

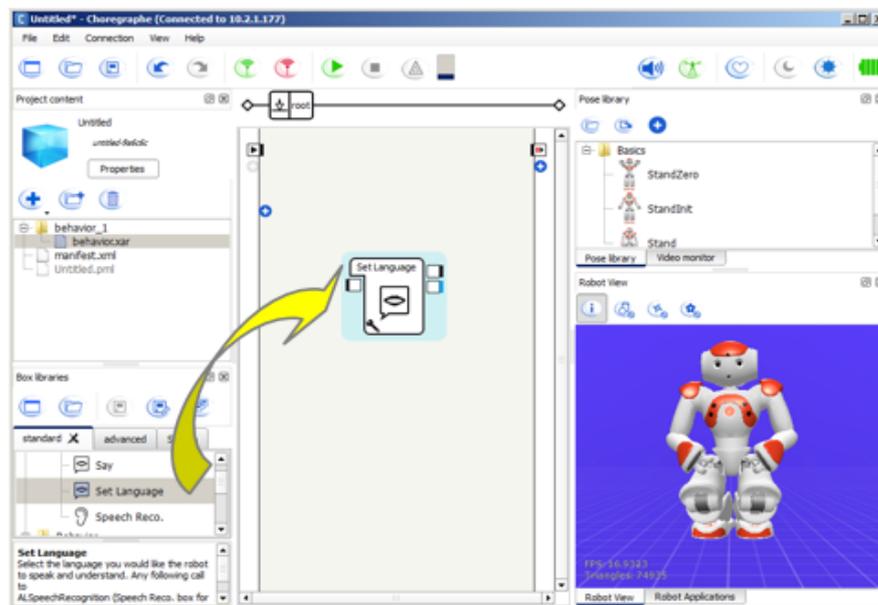


Figura 45: Set Language

4. Fare clic destro sulla grigia e selezionare, dal menu, (**creare una nuova casella > dialogo..**), come in figura 46.

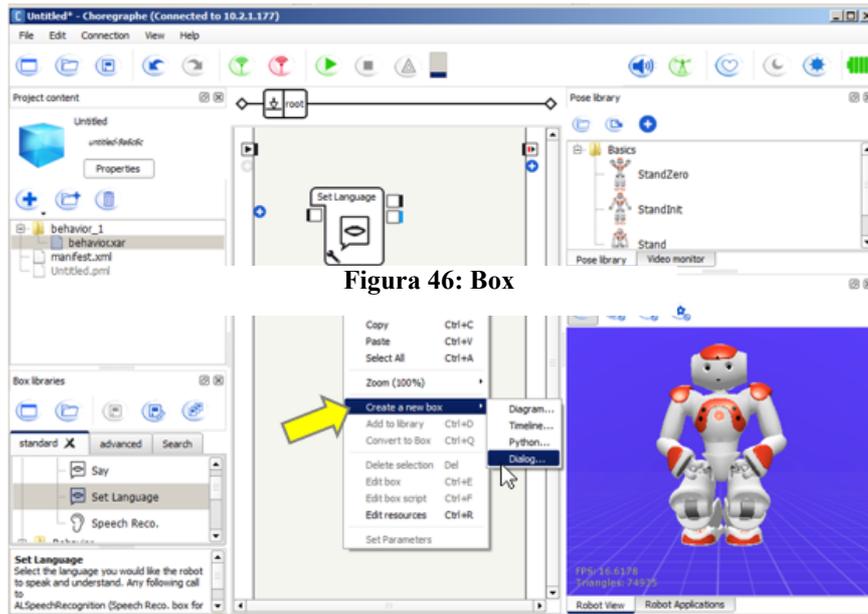


Figura 46: Box

5. F a i
u n

clic sul
pulsante
Aggiungi
n u o v o

argomento, quindi digita un nome senza spazi, ad esempio: *HelloWorld*. Fare clic sul pulsante **Aggiungi**, quindi sul pulsante OK. Vedere figura 47.

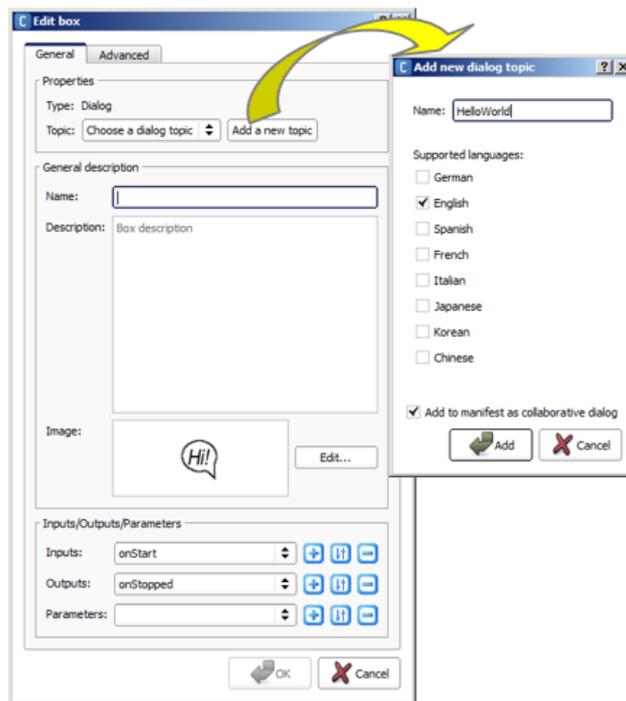


Figura 47: Hello world

6. Crea collegamenti come in figura 48.

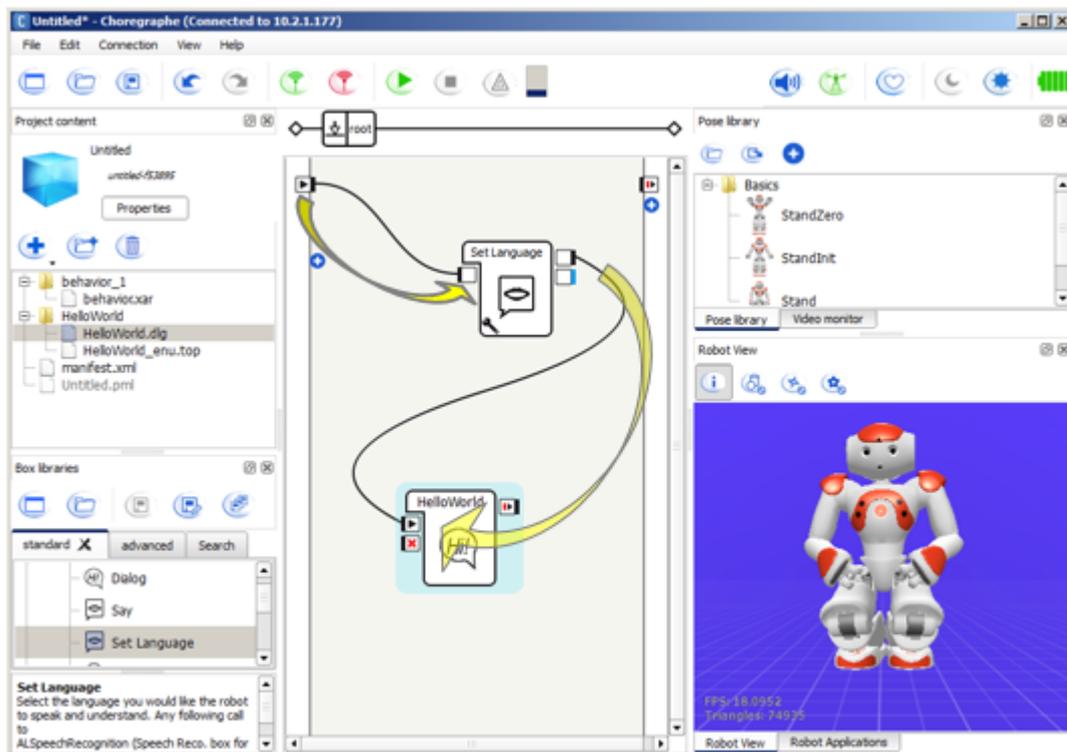


Figura 48: Collegamenti

7. Ora dobbiamo scrivere il nostro dialogo in inglese. Nel contenuto del progetto, fai doppio-clic sul file *HelloWorld_enu.top*. Vedere figura 49.

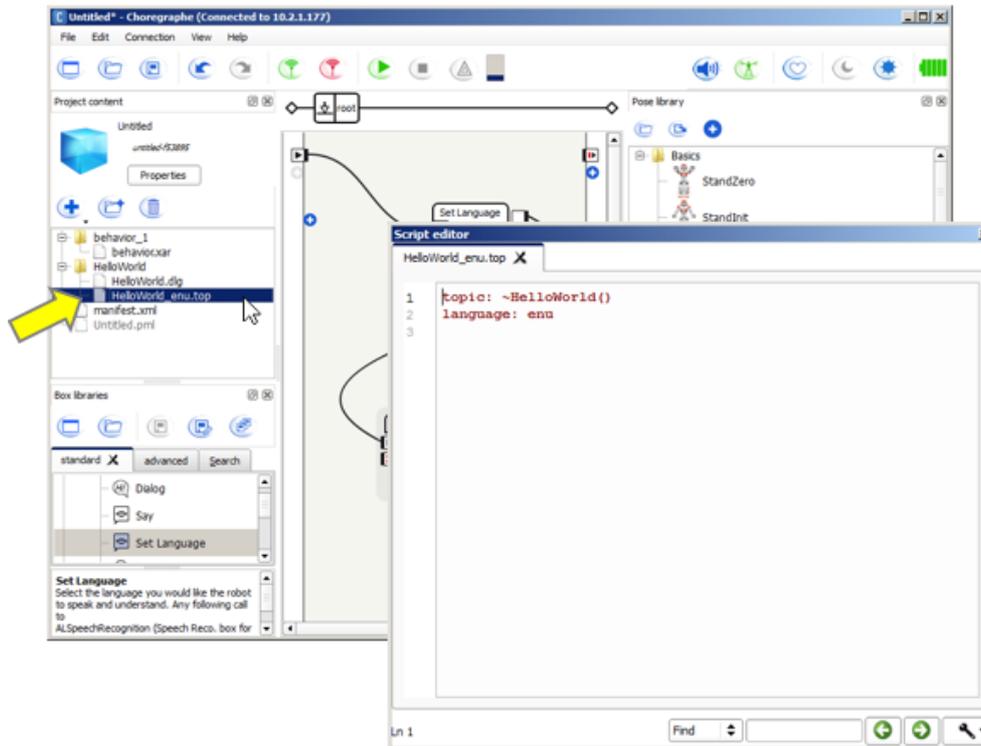


Figura 49: Dialogo

8. Digita le regole come in figura 50:

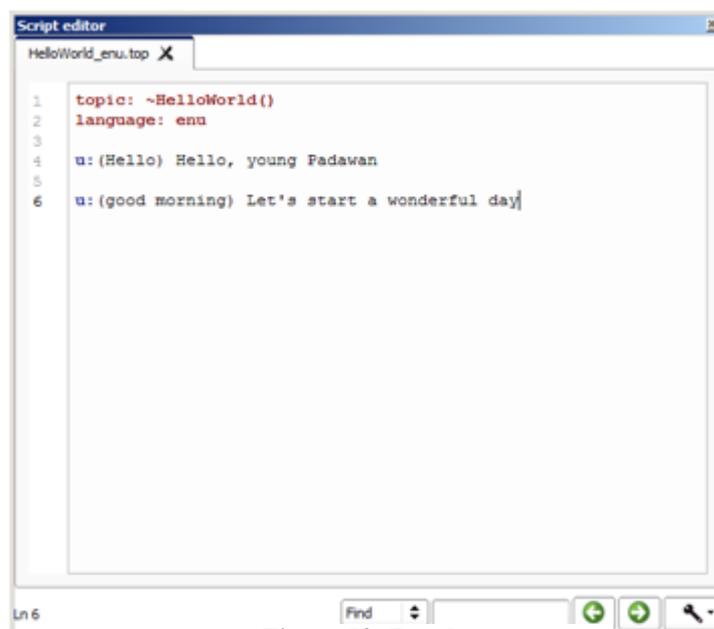


Figura 50: Regole

9. Fai clic sul pulsante “play” per eseguire il tutto.

- E quindi ora, potrai dire "good morning" proprio davanti al suo volto, oppure digitando "good morning" nel pannello di Dialogo e premendo subito dopo invio.
 - Il tuo robot poi risponderà: "Let's start a wonderful day".
- Oppure potrai anche dire, "Hello" proprio davanti al suo volto, oppure digitando "Hello" nel pannello di Dialogo e premendo subito dopo invio.
 - Il tuo robot risponde "Ciao, giovane Padawan".
- Il tuo robot ti parlerà con un'animazione di suoni e luci a LED. [23]

SDK

In figura 51 possiamo elencare una vasta panoramica di linguaggi di programmazione supportati:

Programming Languages	Bindings running on		Choregraphe support	
	Computer	Robot	Build Apps	Edit code
Python	✓	✓	✓	✓
C++	✓	✓	⊘	⊘
Java	✓	⊘	⊘	⊘
JavaScript	✓	✓	✓	⊘
ROS	✓	⊘	⊘	⊘

✓	OK
⊘	Not available

Figura 51: Set Language

SDK Python

- [SDK Python](#)
 - [Python SDK - Panoramica](#)
 - [Python SDK - Guida all'installazione](#)
 - [NAOqi Python - API](#)

C ++ SDK

Avanzate

- [C ++ SDK](#)
 - [C ++ SDK - Guida all'installazione](#)
 - [Nuovo Qi C ++ SDK](#)
 - [SDK C ++ deprecato](#)

SDK Java

Controlla i robot **Aldebaran** dal tuo computer usando **Java**

- [SDK Java](#)
 - [introduzione](#)
 - [Come usare](#)
 - [Specificità dell'SDK Java](#)
 - [Iniziare](#)

Javascript

Controlla i robot **Aldebaran** dal tuo browser web utilizzando **JavaScript**

- [SDK JavaScript](#)

Interfaccia ROS

Controlla i robot **Aldebaran** dal tuo computer usando **ROS**

- [Interfaccia ROS](#)

Precedenti versioni

- [Ex Versioni](#)
 - [QiMessaging JavaScript 1.0](#)

Creare un'applicazione

Hello World

1. Fai clic sul pulsante **Connetti a** e collega **Choregraphe** al tuo robot come in figura 52

2. Trascinare il **Discorso** > **Impostazioni vocali** > **Imposta lingua** sulla zona grigia. Fare clic sul pulsante  dei **parametri** e impostare la lingua come **inglese** o **italiano** come in figura 53.

Figura 52: Connessione

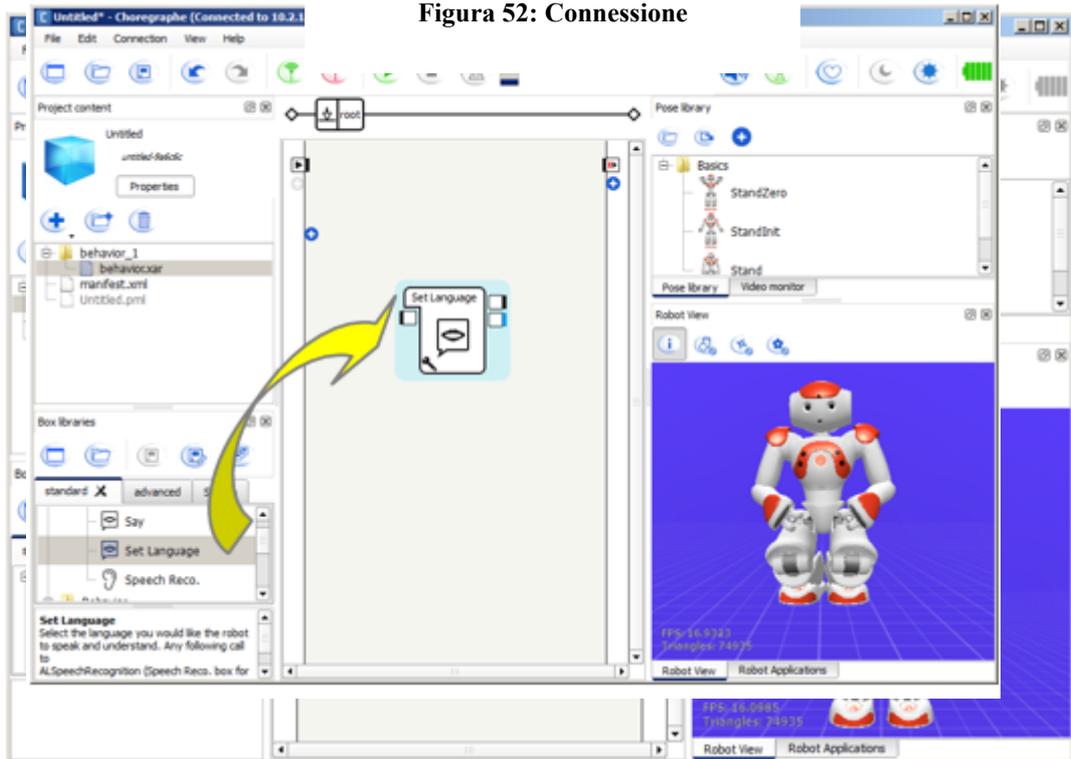


Figura 53: Set Language

3. Fare clic destro sull'area grigia e selezionare, dal menu contestuale, **creare una nuova casella > dialogo** come in figura 54 .

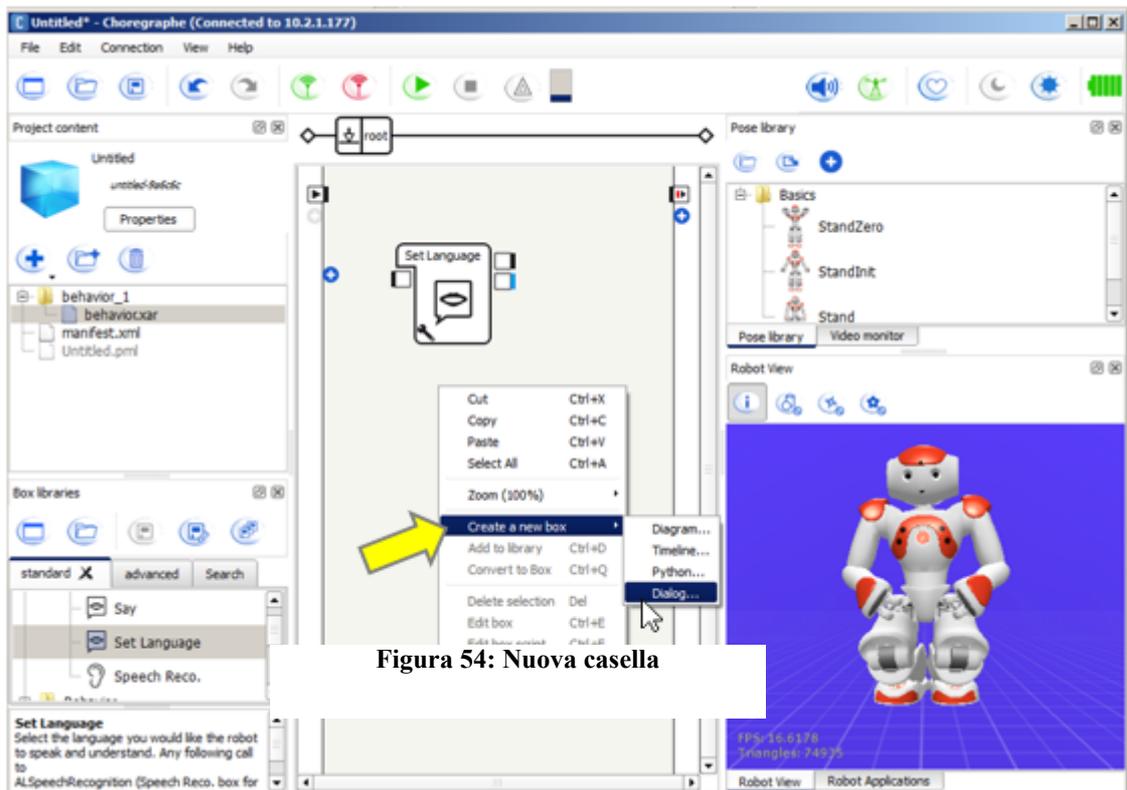


Figura 54: Nuova casella

4. Fai clic sul pulsante **Aggiungi un nuovo argomento**, quindi digita un nome senza spazi, ad esempio: *HelloWorld* . Fare clic sul pulsante **Aggiungi** , quindi sul pulsante **OK**. Vedere figura 55

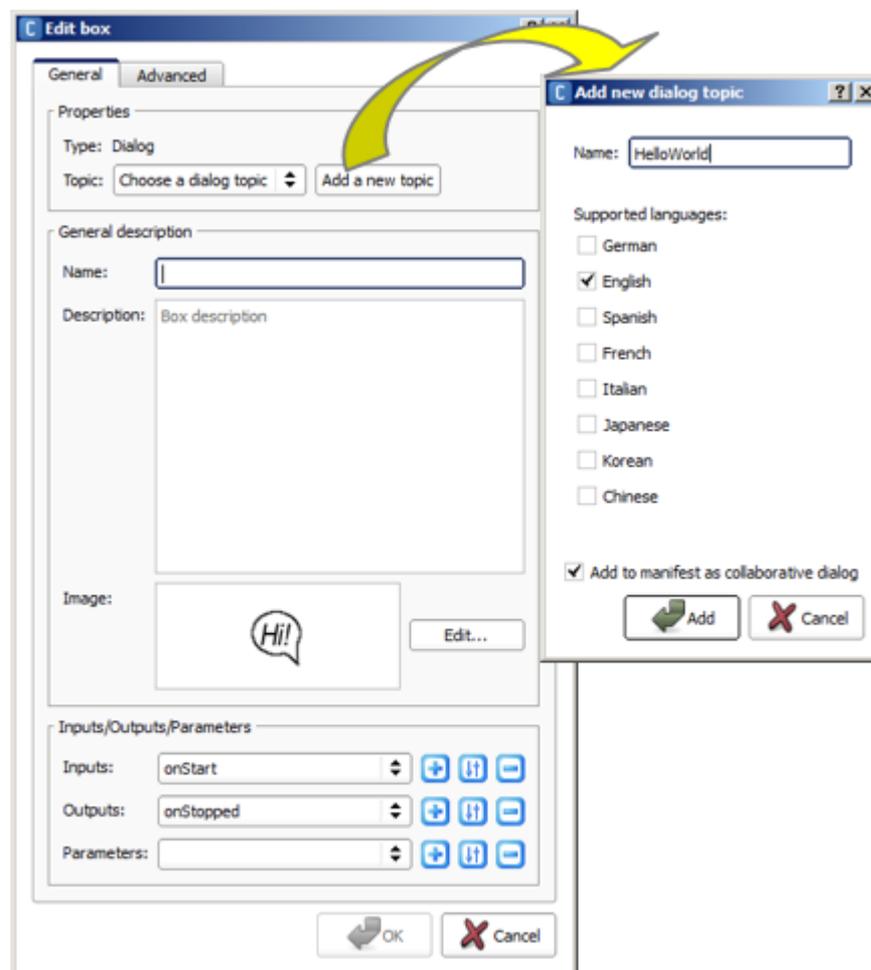


Figura 55: Nuovo argomento

5. Crea collegamenti come in figura 56.

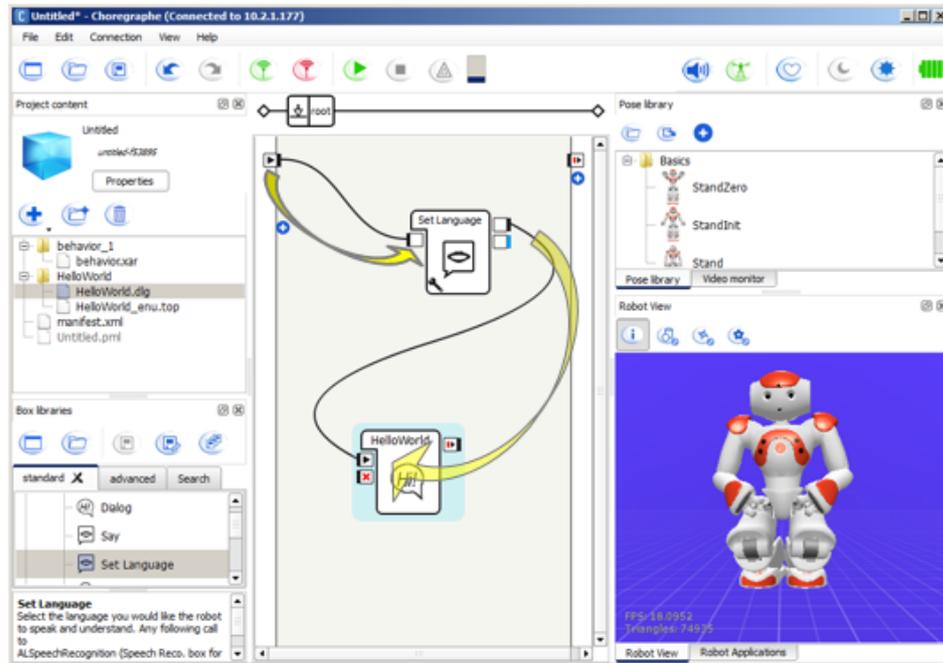


Figura 56: Collegamenti

6. Nel **contenuto del progetto**, fai doppio clic sul file HelloWorld_enu.top. Ti apparirà l' **editor di script**.
7. Digita le regole in figura 57.

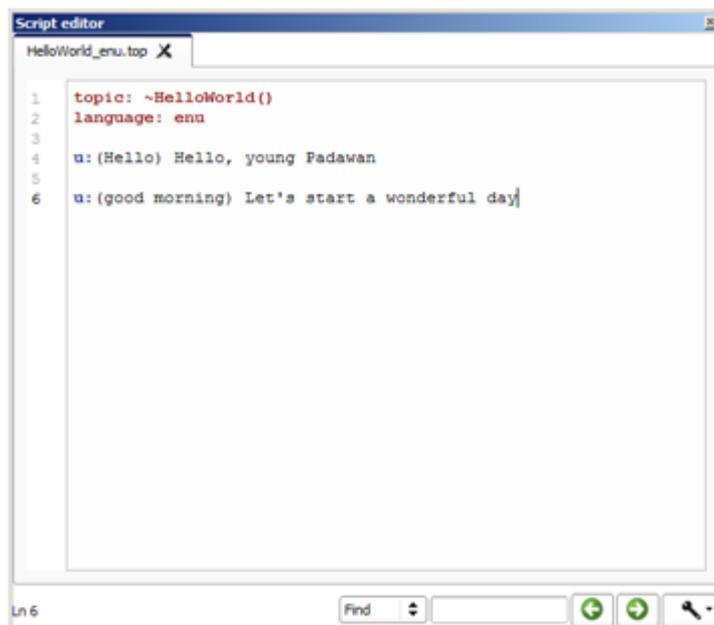


Figura 57: Regole Hello World

8. Infine, fai clic sul tasto **riproduci** 

Applicazione di ballo

Il comportamento principale di questa applicazione contiene 2 caselle: **Init & Reset** e **Dance Timeline**.

Init & Reset

Assicurati che **Pepper**:

- è nello stato giusto prima di iniziare la danza,
- è tornato al suo stato iniziale dopo che la danza è finita.

Questa è la prima casella che viene attivata, perché è direttamente collegata all'ingresso **onStart** dell'Applicazione.

Fare doppio clic sulla casella **Init & Reset** per aprirlo.

Leggiamo i 2 metodi principali.

Il metodo **onInput_onStart** viene eseguito all'avvio dell'applicazione per:

- Disattivare **ALAutonomousMoves** e **ALBasicAwareness** .
- Tornare alla postura di **Stand** .
- A seconda del successo o dell'insuccesso della chiamata della funzione **goToPosture** , sarà l'esito positivo o negativo.

Se ha successo, la casella **Dance Timeline** viene avviata, altrimenti l'applicazione si chiude.

```
def onInput_onStart ( self ):
    prova :
        self . amoves . setBackgroundStrategy ( "none" )
    tranne Exception come exc :
        self . log ( "Impossibile disabilitare le mosse autonome: % s " % str ( exc ))
    provare :
        self . ba . stopAwareness ()
    tranne Exception come exc :
        self . log ( "Impossibile interrompere la consapevolezza: % s " % str ( exc ))

    sé . posturaProxy . setMaxTryNumber ( 3 )
    result = self . posturaProxy . goToPosture ( "Stand" , 0.8 )
    se ( risultato ):
        self . successo ()
    altro :
        sé . fallimento ()
```

Il metodo **onUnload** della casella viene eseguito quando l'applicazione si chiude. Riattiva **ALAutonomousMoves** e **ALBasicAwareness** .

```

def onUnload(self):
    try:
        self.amoses.setBackgroundStrategy("backToNeutral")
    except Exception as exc:
        self.log("Unable to enable breathing: %s" % str(exc))
    try:
        self.ba.startAwareness()
    except Exception as exc:
        self.log("Unable to start awareness: %s" % str(exc))
    self.postureProxy.stopMove()

```

Dance Timeline

Il mix di Dance Timeline contiene:

- Movimenti del corpo
- Musica da un file audio
- Un'immagine visualizzata sul tablet di Pepper

Questa casella viene attivata quando la casella **Init & Reset** termina correttamente, grazie al suo input **onStart** collegato all'output di successo della casella **Init & Reset**.

Fare doppio clic sulla casella **Timeline Dance** per aprirla. Apparirà qualcosa come in figura 58.

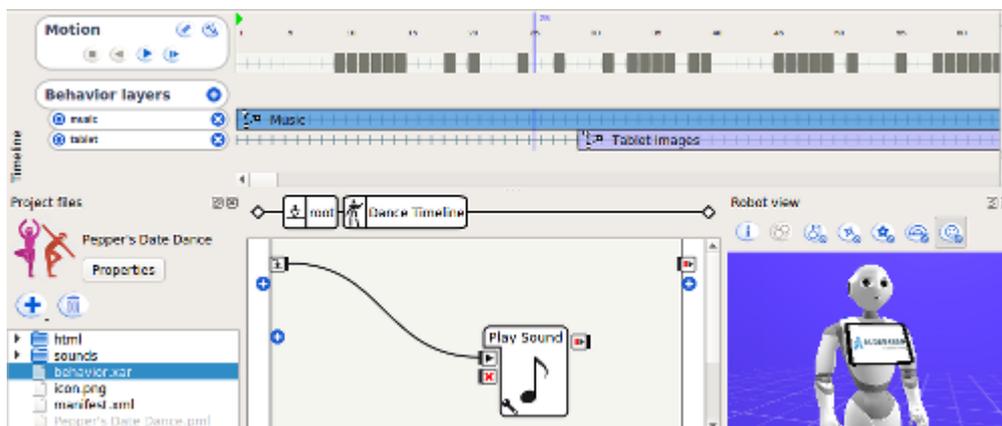


Figura 58: Timeline Dance

Leggiamo i 3

strati principali.

- **Movimento:** i fotogrammi chiave del movimento dell'animazione della danza, che rappresentano la posizione del robot e le sue parti del corpo in un dato momento.
- Livello di comportamento **musicale:** una casella **Riproduci suono**. Riproduce un file audio allegato.
- Livello di comportamento della **tavoletta:** una casella **Mostra immagine**. Visualizza un'immagine sul tablet.

Ecco i passi:

1. Nel pannello del pannello Applicazioni robot , fare clic sul  **pacchetto e installare il progetto corrente sul pulsante del robot**
2. Assicurati che la **vita autonoma** sia attiva. In caso contrario, fare clic sul pulsante  **Turn autonomous life on** .
3. Avvia l'applicazione. Assicurati che la lingua di **Pepper** sia impostata sull'inglese.
4. Pronuncia una delle **frasi Trigger**: "Date dance", "Pepper's date dance", "dance I'm fresh you're pretty",
5. Oppure con il tablet, in **Avvio applicazioni** , premere l'icona dell'applicazione.

Puoi anche provare il comportamento solo facendo clic sul pulsante  **Riproduci** .

Nota che questo esempio funziona correttamente su un vero **Pepper** , poiché **ALTabletService** non è presente su un robot virtuale.

Naturalmente modifica la parte del contenuto che desideri personalizzare, per renderlo "tuo". [24]

Applicazione interattiva

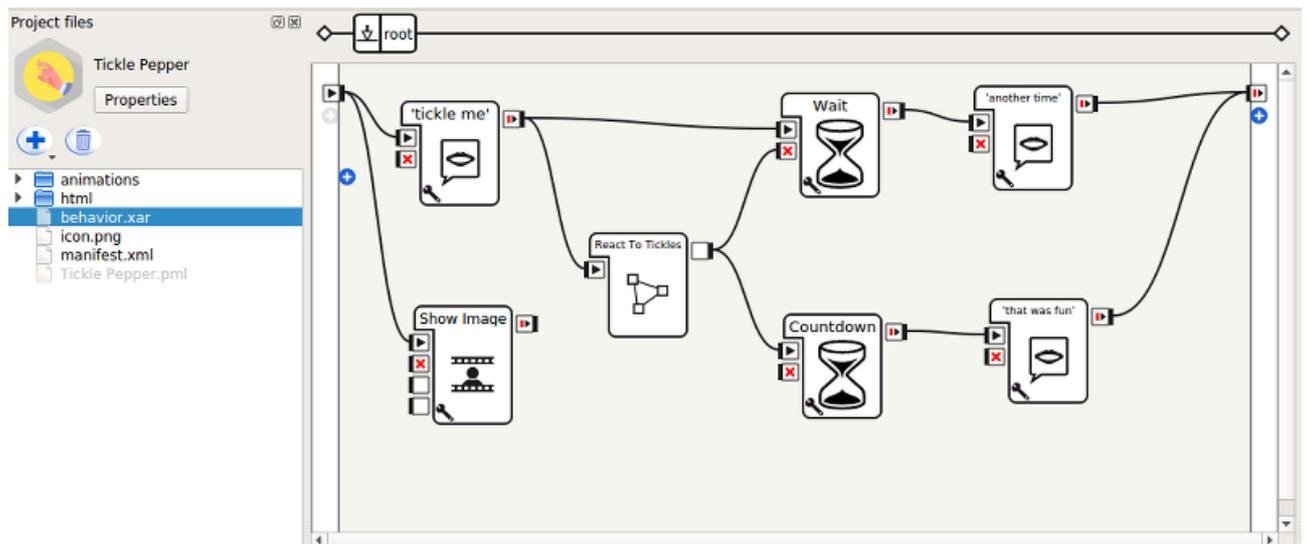
Vediamo adesso come creare un'applicazione interattiva, in particolare, con questa applicazione, se a pepper verrà fatto il solletico, esso "riderà".

Il comportamento principale di questa applicazione contiene 7 caselle, come in figura 59:

- 5 scatole standard: 3 **Animated Say (Tickle me, another time e that was fun)**, **Show Image** e un **Wait**.
- 2 caselle personalizzate: **React to Ticks**(reagisci al solletico) e **Countdown**.

Come una panoramica generale, il comportamento esegue quanto segue:

1. Il robot dice "Solleticami!" E mostra un'immagine nel suo tablet che rimarrà durante l'intero comportamento.
2. Il robot reagisce al "solletico" ogni volta che il suo tablet viene toccato.



3. Il comportamento uscirà se il tablet del robot non viene toccato per 15 secondi.

Prima di uscire, il robot dirà "È stato divertente!" Se il tablet è stato toccato almeno una volta, altrimenti dirà "Ok, un'altra volta allora".

React to Tickle

Assicurati che **Pepper** :

- ascolta l'evento di **touch down** del tablet,
- reagisce suonando un suono e/o un'animazione ogni volta che viene sollevato l'evento precedente,
- non suona più suoni allo stesso tempo,
- non riproduce diverse animazioni contemporaneamente.

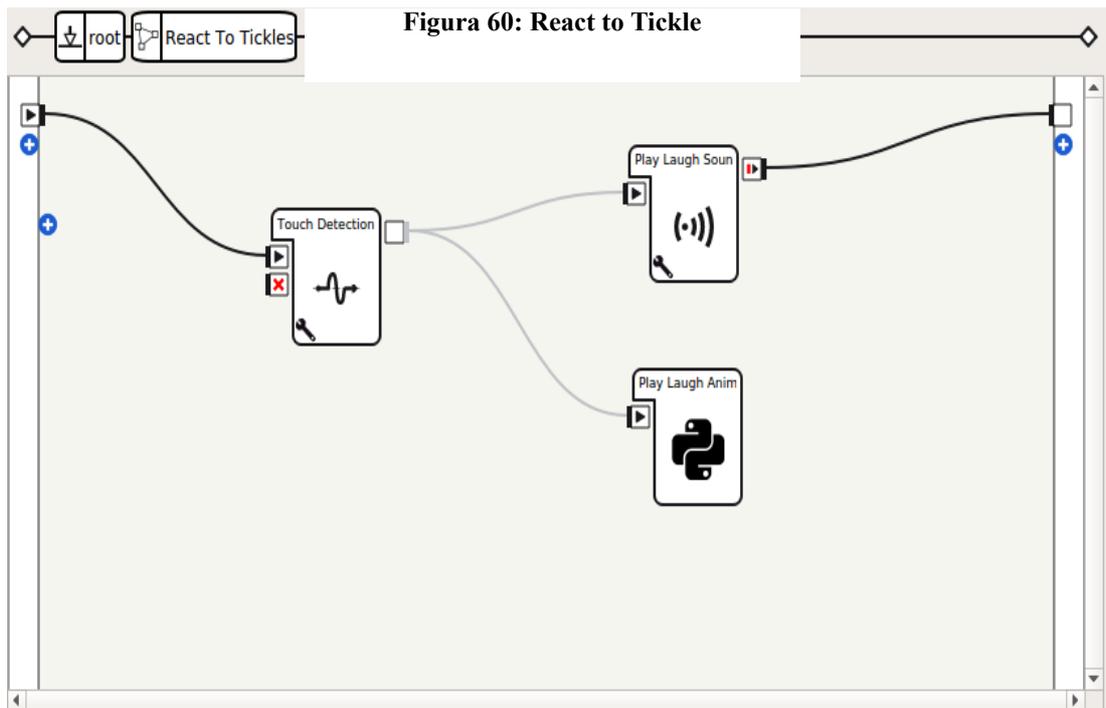
Come funziona

Questa casella verrà attivata dopo la prima casella **Pronuncia animati**, denominata **solleticami**. Questa casella ha un output **onTickle** che verrà attivato quando il robot rileva e reagisce al tocco del tablet. Quando questo output viene attivato, la casella **Wait** avviata in parallelo viene arrestata. Si noti che la casella **Attesa** viene utilizzata per impostare un timer che consentirà di uscire automaticamente nel caso in cui il tablet non venga toccato durante i primi quindici secondi del comportamento.

Fai doppio clic sulla casella **Reagisci al solletico** per aprirlo.

Questa casella contiene un diagramma con 3 caselle: una casella **Touch Detection** e 2 caselle personalizzate: **Play Laugh Sound** e **Play Laugh Animation**.

La casella **Touch Detection** è stata parametrizzata per ascoltare l' evento **On touch down** tablet. Entrambe le box personalizzate sono direttamente collegate



all'uscita **onTouched** del box **Touch Detection**. Ciò significa che entrambi saranno attivati dopo che l'evento precedente è stato rilevato.

La casella **Play Laugh Animation** ha il compito di riprodurre casualmente una delle animazioni contenute nella cartella delle animazioni e contrassegnate come **tickle**. Questa casella impedisce anche di riprodurre un'animazione se ne è già stata riprodotta un'altra.

Fai doppio clic sulla casella **Play Laugh Animation** per aprirla.

```
class MyClass ( GeneratedClass ):
    def onLoad ( self ):
        self . finito = Vero
        io . animPlayer = ALProxy ( "ALAnimationPlayer" )

    def onInput_onStart ( self ):
        se self . finito :
            auto . finito = Falso
            sé . animPlayer . runTag ( self . packageUid () + "/" animations /" , "tickle" )
            self . finito = Vero
            io . onStoped ()
```

La **risata del gioco suono** ha il compito di giocare a caso uno dei suoni scelti contenute nel **Soundset di Aldebaran**. Questa casella impedisce anche di suonare un suono se ne sta già suonando un altro.

Fare doppio clic sulla casella **Riproduci audio di risata** per aprirla.

```
importare la classe casuale
MyClass ( GeneratedClass ): def onLoad ( self ): self . finito = Vero io .
audioPlayer = ALProxy ( "ALAudioPlayer" ) self . soundSet = self .
getParameter ( "SoundSet" ) di sé . suoni = auto . getParameter ( "Suoni" )
def onInput_onStart ( self
```

```

):
    se stesso . finito :
        auto . finished = False
        # Verificare se Soundset è installato e caricato
        se self . soundset in sé . audioPlayer .
getLoadedSoundSetsList ():
            sound = random . scelta ( self . sounds . split ())
            self . audioPlayer . playSoundSetFile ( self . soundset ,
suono )
        altro :
            auto . log ( "Soundset non installato:" + str ( self .
soundset ))
            self . finito = Vero
            io . onStoped ()

```

Si noti come in figura 61 che SoundSet e Soundset sono definiti come parametri.

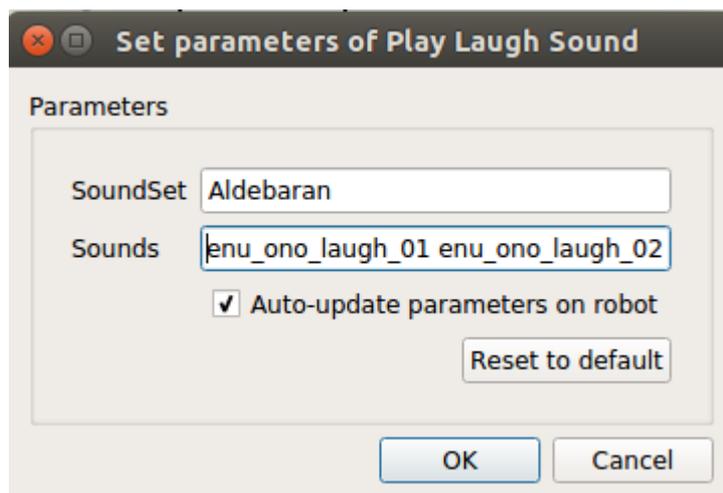


Figura 61: Set Parameters

Ogni volta che viene riprodotto un suono, viene attivata l'uscita **React to Tickers** poiché ciò significa che il robot ha reagito al tocco di un tablet.

Countdown

Assicurati che **Pepper** non rimanga bloccato nel comportamento quando il tablet non viene più toccato.

Questa casella funziona come un timer che viene resettato quando viene attivata l'uscita della casella **Reagire al solletico**. Se il timer raggiunge la fine, significa che il tablet non è stato toccato per quindici secondi e il comportamento deve essere interrotto.

Fare doppio clic sulla casella **Countdown** per aprirlo.

```
class MyClass ( GeneratedClass ):
    def onLoad ( self ):
        importa il threading
        self . correndo = falso
        sé . lock = threading . Lock ()

    def onUnload ( self ):
        self . correndo = Falso

    def onInput_onStart ( self ):
        importa il tempo
        con self . lock :
            # Indipendentemente dal fatto che la scatola sia già in esecuzione o meno,
            resettarla
            automaticamente . seconds_left = self . getParameter ( "Seconds" )
            se self . in esecuzione :
                # la scatola è già in esecuzione
                return
            self . running = True
            # Questa parte del codice sarà raggiunta solo dalla prima chiamata.
            mentre io . correndo e sé . seconds_left > 0 :
                time . sonno ( 1.0 )
                auto . seconds_left -= 1
                # timeout è terminato, trigger output.
            con sé . lock :
                se self . correndo :
                    auto . correndo = falso
                    sé . onStoped ()

    def onInput_onStop ( self ):
        self . onUnload () # ripulire
```

Alcune note su questa box:

- Il suo tempo è approssimativo (quando resettato, non ci vorranno *esattamente* quindici secondi prima che il suo output venga attivato).
- Questa casella blocca un thread durante l'esecuzione.

Ecco i passi

1. Nel pannello del pannello Applicazioni robot , fare clic sul  **pacchetto e installare il progetto corrente sul pulsante del robot** .
2. Assicurati che la **vita autonoma** sia attiva. In caso contrario, fare clic sul pulsante  **Turn autonomous life on** .
3. Avvia l'applicazione e assicurati che la lingua di **Pepper** sia impostata sull'inglese. Pronuncia una delle **frasi Trigger**: “tickle you”, “tickles”, “tickle game”, “tickling” or “are you ticklish” oppure attraverso il tablet, premi l'icona **dell'applicazione in Avvio applicazioni**.
4. Tickle Pepper: tocca ripetutamente il tablet di **Pepper** .

Puoi anche provare il comportamento solo facendo clic sul pulsante  **Riproduci**.

Nota che questo esempio funziona correttamente su un vero **Pepper**, poiché **Soundset Aldebaran** e **ALTabletService** non sono presenti su un robot virtuale.

Prova a cambiare la reazione del robot o aggiungi un'altra casella **Touch**, come la casella **Tactile Head**. [25]

Creare un'applicazione da zero

Ora vedremo come creare una semplice applicazione basata su un sistema interattivo di attività che ricorda il proprietario del robot.

Useremo le **frasi di trigger** e alcune finestre di dialogo per impostare il nome del proprietario.

Passaggio 1: Creare un'attività interattiva

1. Crea un nuovo progetto e salvalo come "Dillo al mio proprietario" dove preferisci.
2. Cerca la casella **Animated Say** dalla box library usando la ricerca e trascinala nel tuo diagramma.
3. Collega il suo input all'ingresso principale del diagramma e la sua uscita all'uscita principale del diagramma, come illustrato in figura 62:

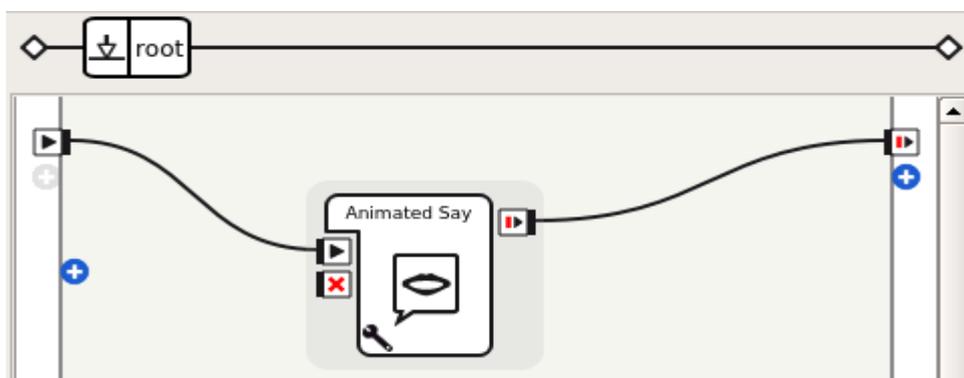


Figura 62: Animated Say

4. Fai clic sul pulsante  **Riproduci** per testare il comportamento. Il robot dovrebbe dire "Ciao". Se stai lavorando su un robot virtuale, puoi vedere in Robot View una nuvoletta con scritto "Ciao".

Passaggio 2: Definire i trigger della propria attività interattiva

- Clicca su Proprietà .
- Imposta la **descrizione dell'applicazione** . Questa descrizione sarà visibile su **Aldebaran Store** , quindi è importante scriverlo a fondo.
- Ora modifica le proprietà del comportamento e imposta metadati aggiuntivi come, ad esempio:
Nome : "Dì al nome del mio proprietario"
Descrizione : "L'esecuzione di questo mi fa dire il nome del mio proprietario".
È possibile aggiungere **Tag** liberamente, ma è necessario assicurarsi che la **natura** del comportamento è **interattivo** , che le bandiere vostro comportamento ad essere un Activity .
- Imposta le **frasi di trigger**, che dovrebbero eseguire automaticamente il comportamento. È possibile impostare, ad esempio:
"Chi è il tuo proprietario?"
"Di chi sei?"
"Chi è il tuo umano?"
"Chi è il tuo capo?"
O altre frasi divertenti per attivare l'attività. Tuttavia, si noti che le migliori frasi trigger sono esplicite e non dipendono dal contesto per essere comprese.
- Premere **OK** per confermare le modifiche.

Ora è il momento di **testare i trigger**.

1. Salva il tuo progetto
2. Nel pannello del pannello Applicazioni robot , fare clic sul  **pacchetto e installare il progetto corrente sul pulsante del robot** .
3. Fare clic  **sul pulsante Turn autonomous life on** .
4. Mettiti di fronte al tuo robot, finché non inizia ad ascoltarti.
5. Prova le frasi trigger a voce alta! Il tuo robot **Aldebaran** dice "Ciao" quando ha capito.

Passaggio 3: Crea un'altra attività per impostare il nome del proprietario

L'implementazione fittizia della tua applicazione è andata bene, ora è tempo che il tuo robot apprenda e richiami l'identità del proprietario.

Innanzitutto, è necessario eseguire una seconda attività interattiva dedicata all'input del proprio nome in modo che il robot possa riconoscerti.

1. Crea un comportamento , denominato "get_name_activity". Fare doppio clic sul **comportamento.xar** creato in "get_name_activity" per aprirlo.
2. Crea una finestra di dialogo nel diagramma, associata a un nuovo **argomento** inglese denominato "get_name". Modifica l' input **onStart in** modo che possa essere di tipo **String** .

Aggiungi i seguenti output:

- **spellingStarted** di natura **puntuale** e di tipo **Bang**
- **spellingStoppato** di natura **puntuale** e di tipo **Bang**
- **nextLetterGiven** di natura **puntuale** e di tipo **String**

E premere **OK**.

Ulteriori informazioni sugli ingressi e le uscite della scatola qui: Input of a box .

Ora puoi modificare la versione inglese dell'argomento della finestra di dialogo facendo doppio clic su **get_name / get_name_enu.top** nel **contenuto** del **progetto** e inserisci il seguente codice:

```
argomento: ~ get_name ()
lingua: enu

# Aggiungi ulteriori lettere qui se necessario
concetto: (lettera) [abcdefghijklmnopqrstu vwxyz]

# Diversi modi per dire "stop"
concetto: (stop) [finire finito stop fatto]

# Cosa dire quando la finestra di dialogo viene avviata
da una casella,
# con il nome del proprietario assegnato come argomento.
u: (e: onStart _)
    Ciao umano. ^ first ["Il mio attuale proprietario è $
onStart. Vuoi cambiarlo?" "Non ho un proprietario
attuale. Vuoi impostarlo?"].

    u1: (si)
        Ok, allora scrivi il tuo nome, dì di fermarti
quando hai finito! $ spellingStarted = 1

        # Prendi la lettera come input, ripeti, invia
        # dalla casella e attendi ancora un'altra lettera
u2: (_ ~ lettera)
```

```

        $ 1 $ nextLetterGiven = $ 1 ^ stayInScope

    u2: (~ stop)
        Ok $ spellingStopped = 1

    u1: (no)
        Ok $ onStopped = 1

```

Questo pezzo di dialogo può essere integrato in una scatola che aiuterà l'applicazione a memorizzare il nome del proprietario.

3. Ora dovrai memorizzare e richiamare il nome del proprietario.

Crea una casella python denominata "Store Owner" e aggiungi i seguenti input:

- **appendNextLetter** di natura **onEvent** e di tipo **String**
- **interruzione** della natura **onStopped** e di tipo **Bang**

Modifica lo script per inserire il seguente codice:

```

class MyClass ( GeneratedClass ):
    def __init__ ( self ):
        GeneratedClass . __init__ ( auto )

    def onLoad ( self ):
        self . ownerName = str () # il nome del
proprietario da impostare
        autonomamente . acceptLetters = False

    def onUnload ( self ):
        self . acceptLetters = False
        auto . ownerName = str ()

    def onInput_onStart ( self ):
        self . acceptLetters = True
        self . ownerName = str ()

    def onInput_appendNextLetter ( self , p ):
        se non self . acceptLetters :
            return
        self . ownerName += p

    def onInput_onStop ( self ):
        se non self . acceptLetters :
            return

import qi

```

```

        ownerFilePath      =      qi . percorso .
userWritableDataPath ( self . packageUid ( ),
"owner_name.txt" )
        con open ( ownerFilePath , 'w' ) come
ownerFile :
        ownerFile . scrivi ( self . ownerName )

        sé . onUnload ( ) #it è raccomandato per riutilizzare il
clean-up quando il box è fermo
        self . onStoped ( ) # attiva l'output della casella

def onInput_abort ( self ):
    se non self . acceptLetters :
        return

        sé . onUnload ( )
        self . onStoped ( ) # attiva l'output della casella

```

Questo script salverà il proprietario dato lettera per lettera in un file specifico per l'ID dell'applicazione. Può essere riletto da qualsiasi altra casella per richiamare i dati memorizzati.

Crea un'altra **Box Python**, chiamata "Recall Owner". Rimuovere l' input **onStop** e impostare il tipo di output **onStold** su **String** . Modifica il suo script e inserisci il seguente codice:

```

class MyClass ( GeneratedClass ):
    def __init__ ( self ):
        GeneratedClass . __init__ ( auto )

    def onLoad ( auto ):
        pass

    def onUnload ( self ):
        pass

    def onInput_onStart ( self ):
        import qi.path
        owner = str ( )
                ownerFilePath      =      qi . percorso .
userWritableDataPath ( self . packageUid ( ),
"owner_name.txt" )
                try :
                    with open ( ownerFilePath , 'r' ) come
ownerFile :
                        owner = ownerFile . readline ( )
                tranne Exception , e :
                    self.logger . avvertimento ( str ( e ))

```

```

auto . onStoped ( proprietario )

def onInput_onStop ( self ):
    self . onUnload () #it è raccomandato per
riutilizzare il clean-up quando il box è fermo
    self . onStoped ( p ) # attiva l'output della
casella

```

Questo leggerà semplicemente il file e lo stamperà.

4. Collegali Tutti:

- get_name.spellingStarted -> Store Owner.onStart
- get_name.spellingStopped -> Store Owner.onStop
- get_name.nextLetterGiven -> Store Owner.appendNextLetter
- Recall Owner.onStoped -> get_name.onStart
- Store Owner.onStoped -> Recall Owner.onStart
- Comportamento all'avvio -> Richiamare Owner.onStart
- get_name.onStoped -> Behavior's onStoped

Dovresti ottenere qualcosa di simile a quello mostrato in figura 63:

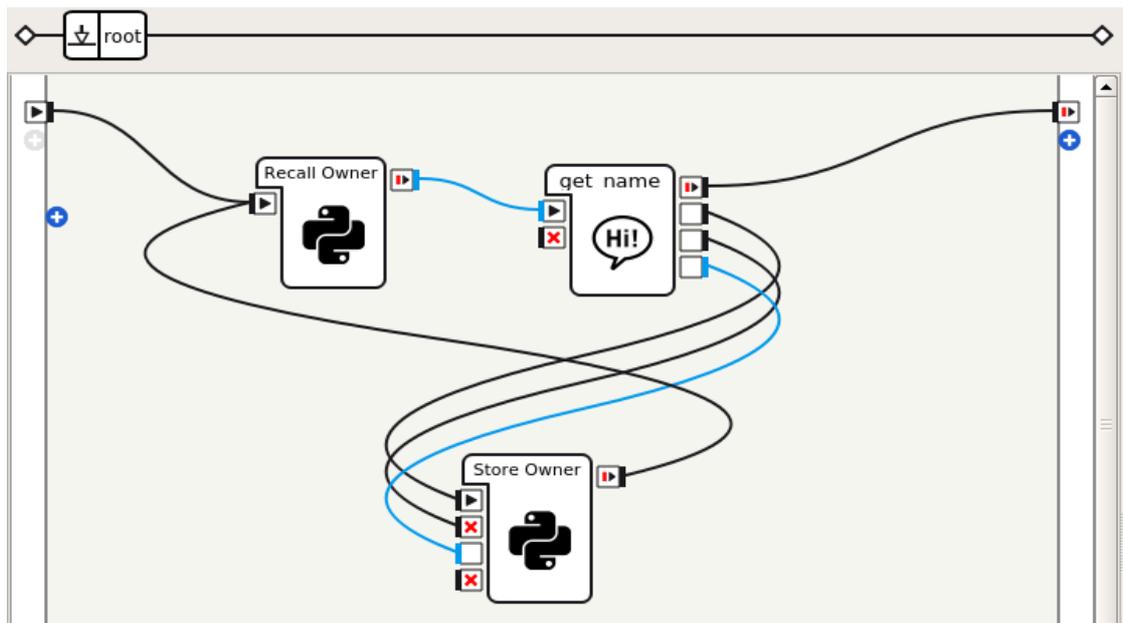


Figura 63: Collegamenti

5. Per provarlo, basta cliccare sul
▶ pulsante **Riproduci** per testare questo comportamento. Dovresti essere in grado di interagire con il robot con la voce o utilizzare il pannello di dialogo per inserire il testo.

Passaggio 4: Richiamo del nome del proprietario nell'attività principale

1. Copia la casella **Recall Owner** che hai appena creato.

2. Apri nuovamente **behavior_1 / behavior.xar** e incollalo lì. Apri la casella **Divi animati in** cui hai inserito quel comportamento per ottenere la casella **Testo animato animato** . Copia, torna al livello **root** e incollalo. Ora puoi eliminare la casella **Dire Animato** .
3. Ora hai una scatola che ricorda il nome del proprietario e una scatola che può dirlo. Ma quello che il tuo robot dovrebbe dire è più di un semplice nome! Quindi facciamo un altro **box Python** chiamato "Format Message". Rimuovi l' input **onStop** , imposta il tipo di output **Terminato** su **String** e scrivi il seguente codice:

```
class MyClass ( GeneratedClass ):
    def __init__ ( self ):
        GeneratedClass . __init__ ( auto )

    def onLoad ( auto ):
        pass

    def onUnload ( self ):
        pass

    def onInput_onStart ( self , p ):
        message = str ()
        se len ( p ) > 0 :
            message = "Il mio proprietario è % s "
        % p
        else :
            messaggio = "Non ho proprietario!"
        sé . onStoped ( messaggio )
```

E collega le caselle nel seguente modo:

- Comportamento all'avvio -> Richiamare Owner.onStart
- Recall Owner.onStopped -> Format Message.on Start
- Formatta Message.onStopped -> Animato Pronuncia testo.onStart

Il comportamento dovrebbe assomigliare a quello in figura 64:

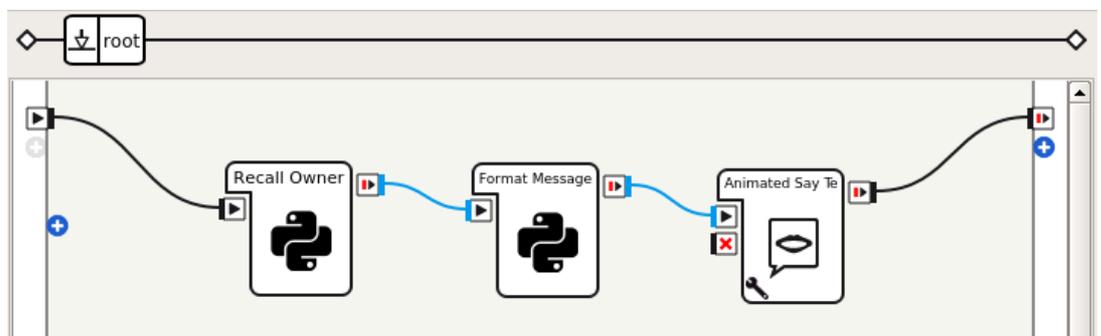


Figura 64: Comportamento

4. Provalo premendo il pulsante  **Play** e vedi il tuo robot che richiama il suo proprietario! Dovresti testarlo nuovamente attraverso i trigger. [26]

Utilizzo del tablet

Il tablet può essere utilizzato per visualizzare immagini, video e pagine Web.

Per controllare il tablet:

1. Imballa i file che vuoi mostrare nella tua **Applicazione**. Per ulteriori dettagli, consultare: Contenuto della tavoletta di imballaggio nell'applicazione.
2. Controlla il display sul tablet, utilizzando Caselle di **Choregraphe** disponibili nella cartella **Multimedia** > **Tablet** , o ALTabletService .

Tutto il contenuto mostrato sul tablet deve essere ospitato sul robot. Fare così:

1. Nel pannello **File progetto** , crea una cartella denominata "html".
2. Importa in questa cartella tutte le risorse da visualizzare sul tablet: immagini, video, pagine Web, javascript, ecc.

Ad esempio, se un'applicazione "simple-tabletpage" ha i file come in figura 65:

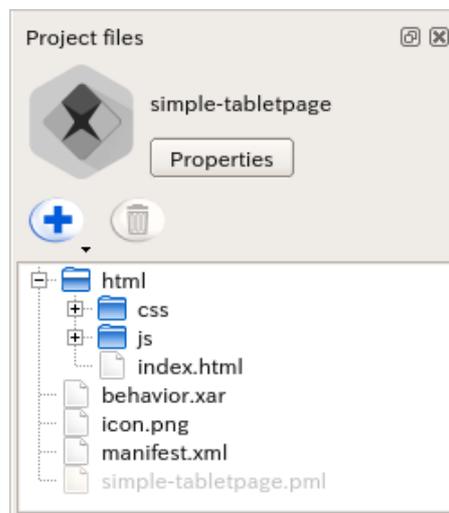


Figura 65: File

3. **I n s t a l l a** l' **applicazione** sul robot.
4. Prova la pagina. Ad esempio, per visualizzare index.html, visita:
<http://robots-ip-address/apps/simple-tabletpage/index.html>, o
<http://robots-ip-address/apps/simple-tabletpage/>

Mostrare immagini

Le immagini sono una buona soluzione semplice per la visualizzazione di cose semplici sul tablet mantenendo la tua applicazione piccola e leggera. Le immagini devono essere 1280 x 800 pixel per coprire l'intero schermo del tablet. Le gif animate verranno mostrate animate.

Il tablet può precaricare le immagini in modo che vengano visualizzate più rapidamente.

Per mostrare le immagini, utilizzare la casella **Mostra immagine**, specificando il percorso relativo dell'immagine nella cartella "html". Vedere figura 66

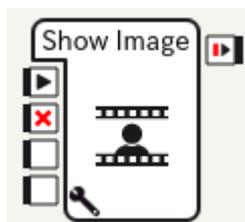


Figura 66: Show Image

Le immagini sono
viene caricata.

precaricate quando la scatola

Puoi anche chiamare `ALTabletService::showImage`, ma in questo caso, se dovrai gestire il pre-caricamento. Per ulteriori dettagli, vedere: `ALTabletService::preLoadImage`.

Riproduzione dei video

I video sono un buon modo per avere uno schermo più incisivo mantenendo il tuo codice semplice. Dovrebbero avere una risoluzione di 1280 x 800 pixel, con un codec h264 / mp4.

Per riprodurre video, utilizza la casella **Riproduci video**, specificando il percorso relativo del video nella cartella "html". Vedere figura 67



Figura 67: Play Video

Puoi anche chiamare `ALTabletService.playVideo` e altre funzioni `ALTabletService` per l'arresto, la pausa e così via. Per il riferimento completo, consulta l' API `ALTabletService`.

Una nota sul suono: se il video ha un suono, verrà dagli altoparlanti del tablet e non dal robot. Pertanto, nella maggior parte dei casi, si consiglia di utilizzare video senza audio, a meno che non si desideri realmente che il suono provenga dal tablet e non dal robot.

Mostrare pagine web

Le pagine Web offrono una maggiore flessibilità per creare qualsiasi tipo di contenuto tablet, tuttavia, vi è il rischio di rendere più complesso il codice dell'applicazione.

Potresti voler visualizzare due tipi di pagine web:

- Pagine Web contenute nella cartella html dell'applicazione, nel qual caso dovresti inserire un "index.html" nella cartella "html" della tua app.
- Pagine Web da Internet, poiché il tablet è connesso a Internet durante il normale utilizzo.

Per visualizzare una pagina Web, utilizza la casella **Mostra app**, che mostrerà l'indice index.html dell'applicazione. Puoi anche utilizzare **Mostra Webview** e **Nascondi Webview** per nascondere / mostrare la pagina, in modo da poter nascondere temporaneamente la pagina e poi tornare ad essa in seguito senza dover ricaricare. Vedere figura 68.



Figura 68: Show app-WebView

Puoi anche chiamare direttamente `ALTabletService::loadUrle` altre funzioni simili. Nota che dovrai prima chiamare `loadUrl` o `loadApplication` e poi chiamare `showWebview` - per il riferimento completo, vedi l' API di `ALTabletService` .

Ci sono alcune cure speciali che ti occorrono per creare pagine web per il tablet:

- Il tablet ha una CPU relativamente lenta, quindi molti eventi e animazioni non saranno così reattivi come lo sarebbero su un browser. Questo può essere un problema se si utilizza un framework JavaScript "pesante".
- Per essere compatibile con la densità dello schermo attuale e futura e la risoluzione del display del browser, dovresti fare del tuo meglio per creare o selezionare contenuti reattivi.

Tuttavia, se è possibile, è possibile risolvere il problema della densità su uno specifico robot aggiungendo questa linea nella sezione principale del contenuto:

```
<meta name = "viewport" content = "width = 1280, user-scalable = no" />
```

Risolverà la risoluzione della pagina web a 1280px (risoluzione nativa corrente).

Uno dei principali vantaggi delle pagine Web è che è possibile utilizzare: JavaScript SDK , che consente di chiamare direttamente i servizi NAOqi da JavaScript.

Reazione per eventi tattili

Esistono due approcci per rilevare che il tablet è stato toccato:

- Utilizzando la casella **Rilevamento tocco** o equivalentemente i segnali `ALTabletService.onTouchDown`, `onTouchUp` e `onTouchMove`

- Visualizzando una pagina Web che gestisce gli eventi tattili in javascript, con i pulsanti, ecc. (In questo caso, alcune librerie come [FastClick](#) possono aiutare a compensare la reattività lenta dei clic del tablet).

Nota che la prima soluzione ti darà la reazione del robot più veloce, ma comunque la gestione del tocco in JavaScript ti consentirà di dare un feedback visivo migliore sul tablet.
[27]

La nostra esperienza

La nostra esperienza con Pepper è dovuta ad una giornata di orientamento per le classi 4^a e 5^a degli istituti superiori della provincia barese.

Questo lavoro è stato svolto in collaborazione con i professori Paolo Buono, Nadia De Carolis, Antonietta Lanza e Veronica Rossano come progetto per gli esami di Intelligent Interfaces e E-learning methods and techniques.

Il lavoro svolto può essere suddiviso in tre sotto-lavori:

1. Presentazione DIB
2. FAQ
3. Feature aggiuntive

Presentazione DIB

Questa parte del progetto, ha il fine di far esporre da Pepper agli studenti interessati, una presentazione del dipartimento di Informatica.

Nella presentazione, vi è una definizione di informatica, il significato di “diventare un informatico”, la relazione esistente tra matematica ed informatica, la bassa percentuale di presenza femminile, i test d’ingresso per l’iscrizione ad uno dei corsi di laurea triennali ed infine l’offerta formativa del dipartimento.

Tutto questo, è ciò che solitamente viene esposto dalla professoressa Lanza durante le giornate di orientamento con gli studenti degli istituti superiori. Per cui l’idea è appunto sostituire una lunga presentazione di oltre trenta minuti, con una più **snella e accattivante** esposta da un **robot**.

La presentazione è stata realizzata mediante 3 box coreographe sequenziali.

Il primo è il box *Set Languages*, impostato sull'italiano

Il secondo è stato creato aggiungendo un *dialog topic*, e quindi un *Dialog box* che conterrà "*Presentation_iti.top*" dove all'interno vi sono i pattern vocali con cui si attiva la presentazione. In particolare, sono state considerati i pattern "fammi la presentazione" e "a te la parola".

Di seguito mostriamo il codice per il box *Presentation_iti.top* sopracitato.

```
topic: ~Presentation()
language: iti

# Defining extra concepts out of words or group of words
concept:(presentation) ["fammi la presentazione" "a te la
parola" ]

# Catching inputs and triggering outputs
#u:(e:onStart) $onStopped=1

# Replying to speech

u:({*}~presentation{*}) Salve a tutti,\pau=500\ $img="dib"
Benvenuti al Dipartimento di Informatica , è bello vedervi tutti a
casa mia.\pau=500\ Vi siete mai chiesti coe'è l'informatica?.
\pau=1000\ $img="it" L'informatica è scienza e tecnologia, dove le
tecnologie non sono altro che le produzioni industriali date dalla
scienza. \pau=500\ Il termine informatica discende da scienza
dell'informazione, \pau=500\ studiare dal punto di vista
scientifico ciò che ruota attorno all'informazione. \pau=1000\
$img="invent" L'informatica è una scienza affascinante dove c'è
tutto da poter inventare,\pau=500\ al fine di aiutare la società a
progredire.\pau=500\ Nessuno infatti, oggigiorno, può fare a meno
della tecnologia informatica \pau=500\ ed è certamente l'ambito
più richiesto e più attivo sul mercato ad oggi. Infatti,\pau=500\
ogni anno c'è un incremento di richieste pari a circa il 40
per cento. \pau=500\ $img="path" Intraprendere un percorso
universitario in Informatica significa diventare progettista e
sviluppatore di soluzioni tecnologiche che possono migliorare la
qualità della nostra vita.\pau=500\ $img="app" Come applicazioni
mediche, biologiche o tante altre ancora. \pau=500\ $img="term"
Nel termine "informatico", concentriamo in esso tante figure
professionali come il Programmatore. Progettista. Analista e
Sistemista. \pau=700\ Studiare informatica, significa imparare a
ragionare e applicare deduzioni e induzioni logiche matematiche.
\pau=500\ $img="math" La matematica è il linguaggio della natura
\pau=500\ con cui descrivere la natura che ci circonda. Anche gli
```

algoritmi, che sono l'anima dei calcolatori, \pau=500\ sono descritti con linguaggio matematico. \pau=500\ Non sfuggirete alla matematica, mi dispiace. \pau=500\ \$img="girls" Per quanto riguarda le ragazze, in tutta la società locale ed internazionale, purtroppo, c'è una scarsa presenza femminile ai corsi di informatica con percentuale europea del 15 per cento, \pau=500\ ma ciò non significa che gli studi informatici non siano adatti alle ragazze, \pau=500\ anzi la prima programmatrice è stata donna, \$img="ada" si chiamava Ada Lovelace. \pau=500\ Se c'è qualche ragazza tra voi, \pau=500\ ci aiutasse a migliorare questa percentuale! \pau=500\ \$img="test" Nel caso tu decidessi di intraprendere il percorso universitario in Informatica, sappi che c'è da superare un test di ingresso durante il mese di Settembre, \pau=500\ ma non temere. \pau=500\ il mancato superamento di esso, non preclude la possibilità di cominciare gli studi. \pau=500\ Puoi sempre tentarlo nuovamente a Dicembre e Febbraio, oppure ti basterà passare uno dei due esami di Matematica presenti al primo anno, ovvero Matematica Discreta o Analisi Matematica. \pau=500\ Ma non sarà una passeggiata \pau=500\ \$img="courses" Il dipartimento di Informatica, inoltre, eroga ben 3 corsi di laurea triennale in Informatica: il primo si chiama semplicemente Informatica, \pau=500\ il secondo si chiama Informatica e tecnologie per la produzione del software, \pau=500\ e il terzo si chiama Informatica e Comunicazioni Digitali, \pau=500\ quest'ultimo erogato nella sede di Taranto. \pau=500\ \$img="dib" Scusate se ho parlato tanto! \pau=1000\ Vi ho annoiati?. \pau=1000\ Grazie per l'attenzione, Venitemi a trovare. \pau=500\ Se avete delle domande, mettetemi alla prova. \pau=500\

FAQ

Questa parte del progetto, ha il fine, invece, di far rispondere Pepper agli studenti sulle domande frequenti poste durante le giornate di orientamento.

Le FAQ considerate sono le seguenti:

- Ho l'ECDL, sarò avvantaggiato negli studi?
- Sono una ragazza, sono studi adatti a me?
- Frequento un Liceo Classico, avrò difficoltà?
- Frequento un Istituto Superiore in cui ho già programmato, sarà una passeggiata?
- Non mi piace la Matematica, fa lo stesso?
- Dopo la laurea triennale cosa potrò fare?
- Che differenza esiste tra i tre corsi di laurea proposti dal Dipartimento di Informatica di Bari?

- Che differenza c'è tra un corso di laurea in Informatica ed uno in Ingegneria Informatica o dell'Informazione?
- La laurea triennale in Informatica dà accesso all'Albo Professionale degli Ingegneri?
- Mi vengono riconosciuti dei CFU avendo già una certificazione in inglese?
- L'informatica toglie lavoro ad altre persone?
- La frequenza alle lezioni dei corsi di laurea è obbligatoria?
- Che differenze lavorative ci sono tra un informatico ed un ingegnere informatico?

Anche in questo caso, le domande, e le rispettive risposte sono state concordate

Tutte queste domande sono solitamente poste alla professoressa Lanza durante le giornate di orientamento con gli studenti degli istituti superiori. Per cui anche in questo caso, l'idea è appunto sostituire le risposte data dalla professoressa, con quelle date da un **robot**.

Le FAQ sono state realizzate mediante 3 box coreographe sequenziali.

Il primo è il box *Set Languages*, impostato sull'italiano

Il secondo è stato creato aggiungendo un *dialog topic*, e quindi un *Dialog box* che conterrà "*FAQ_iti.top*" dove all'interno vi sono i pattern vocali con cui si attivano le diverse FAQ e le diverse risposte.

Di seguito mostriamo il codice per il box *FAQ_iti.top* sopracitato.

```
topic: ~FAQ()
language: iti

# Defining extra concepts out of words or group of words
concept:(FAQ1) ["ECDL può servire" "serve l'ECDL" "mi può servire
l'ECDL" "ECDL può servirmi" "ECDL può essere utile" "ECDL può
essermi utile" "ECDL mi serve" "patente europea può servire"
"patente europea può servirmi" "patente europea può essere utile"
"patente europea può essermi utile" "patente europea mi serve"
"può servire l'ECDL" "può servirmi l'ECDL" "può essere utile
l'ECDL" "può essermi utile l'ECDL" "mi serve l'ECDL" "può servire
la patente europea" "può servirmi la patente europea" "può essere
utile la patente europea" "può essermi utile la patente europea"
"mi serve la patente europea"]
concept:(FAQ2) ["le ragazze * " "le donne * " "le femmine * "
"sono una ragazza" "sono ragazza" "sono femmina" "sono donna"
"sono una donna" "cosa per donne" "cose per ragazze" "essendo *
ragazza" "essendo * donna"]
concept:(FAQ3) ["liceo classico" "fatto il classico" "faccio il
classico" "faccio il liceo classico"]
concept:(FAQ4) ["So programmare" "ho programmato" "ho già
programmato" "so già programmare" "conosco il c" "conosco il
linguaggio c" "conosco l'html" "conosco il linguaggio html" "ho
creato un sito web" "ho creato un sito" "ho fatto un sito" ]
```

```
concept:(FAQ5) ["troppa matematica" "tanta matematica" "molta
matematica" "poca matematica" "odio la matematica" "odio
matematica" "non mi piace matematica" "non mi piace la matematica"
"non voglio fare * matematica" "non so la matematica"]
concept:(FAQ6) ["quali sono i possibili sbocchi" "cosa mi aspetta
dopo" "da lavoro" "dopo l'università" "dopo la laurea" "dopo la
laurea triennale" "dopo la triennale"]
concept:(FAQ7) ["quali differenze ci sono tra i corsi di laurea"
"quali differenze ci sono tra i tre corsi di laurea" "quali
differenze stanno tra i corsi di laurea" "quali differenze stanno
tra i tre corsi di laurea" "quali differenze ci sono tra le
lauree" "quali differenze ci sono tra le tre lauree" "quali
differenze stanno tra le lauree" "quali differenze stanno tra le
tre lauree" "* tre lauree" "* diverse lauree"]
concept:(FAQ8) ["tra informatica e ingegneria" "differenze tra il
corso di laurea in informatica e Ingegneria" "differenze tra il
corso di laurea in informatica e il corso di laurea in
Ingegneria"]
concept:(FAQ9) ["iscrizione * albo" "iscrivermi * albo" "accesso *
albo" ]
concept:(FAQ10) ["certificazione * linguistica" "certificazione *
inglese" "certificazione B1" "certificazione B2" "certificazione
A2" "certificazione C1" "certificazione C2" "certificazione *
inglese B1" "certificazione * inglese B2" "certificazione *
inglese A2" "certificazione * inglese C1" "certificazione *
inglese C2" ]
concept:(FAQ12) ["competenze * richieste" "competenze necessarie"
"competenze consigliati" "requisiti richiesti" "requisiti
necessari" "requisiti consigliati" "nozioni richieste" "nozioni
necessarie" "nozioni consigliati" "abilità richieste" "abilità
necessarie" "abilità consigliate" "capacità richieste" "capacità
necessarie" "capacità consigliate" "prerequisiti richiesti"
"prerequisiti necessari" "prerequisiti consigliati"]
concept:(FAQ11) ["aiuta * trovare lavoro" "toglie * lavoro"
"elimina * lavoro" ]
concept:(FAQ13) ["frequenza * lezioni" "frequenza obbligatoria"
"le lezioni sono obbligatorie" "obbligatorio * lezione"
"obbligatorie lezioni"]
concept:(FAQ14) ["differenze lavorative"]

# Catching inputs and triggering outputs
#u:(e:onStart) $onStopped=1

# Replying to speech
u:({*}~FAQ1{*}) Cos'è la patente europea? \pau=500\ scherzo!
\pau=500\ Le capacità acquisite con l'ECDL o con le certificazioni
Cisco, sono molto richieste al giorno d'oggi. Ma non sono
correlate con la scelta universitaria. Perciò non concediamo CFU
per chi li possiede.
```

u:({*}~FAQ2{*) Purtroppo sono poche le donne iscritte al corso di informatica. \$img="girls" \pau=300\ Mi dispiace molto. Vuoi avere maggiori informazioni? \$img="dib"

ul:({~si} {*)) Non so il perchè, purtroppo sono le stesse ragazze ad auto escludersi. E' un fenomeno non solo italiano, ma anche internazionale. Nel resto del mondo la % delle ragazze arriva a stento al 15 per cento!

ul:({~no} {*)) va bene!

u:({*}~FAQ3{*) Non preoccuparti, cominciamo tutti dallo stesso livello.\pau=500\ ossia sotto 0. \pau=500\Vuoi sapere quali altre nozioni servirebbero per le discipline informatiche?

ul:({~si} {*)) Certamente lo studio del latino potrebbero servire per il ragionamento logico, le abilità richieste in ingresso sono quelle che normalmente dovrebbero possedere gli studenti provenienti da una Scuola Superiore.

ul:({~no} {*)) va bene!

u:({*}~FAQ4{*) \pau=300\ Ti piacerebbe avere subito 30 e lode?. , ma senti un mio consiglio: qualsiasi sia il tuo livello di conoscenze informatiche, nei primi mesi si apprende ex-novo... \pau=500\ Vuoi avere maggiori informazioni?

ul:({~si} {*)) Nel corso degli studi non ci si leggerà ad uno specifico linguaggio di programmazione o ad uno specifico ambiente di sviluppo, né ad uno specifico paradigma di programmazione.

ul:({~no} {*)) va bene!

u:({*}~FAQ5{*) \pau=300\ La matematica non piace a molti, ma questa disciplina è molto importante per noi. Un Informatico NON ne può far a meno! Vuoi avere maggiori informazioni?

ul:({~si} {*)) Infatti ci sono almeno 30 Crediti Formativi Universitari in discipline MAT su 180 cfu totali. E non sono pochi!

u:({*}~FAQ6{*) Tutti i lavori contengono l'informatica. Potrai lavorare come dipendente, come libero professionista, oppure puoi continuare il percorso di studi... \pau=500\Vuoi saperne di più?

ul:({~si} {*)) Oltre che lavorare come dipendente nelle diverse aziende locali, italiani e internazionali, puoi lavorare come libero professionista offrendoti come consulente ed esperto; è possibile anche accedere ai ruoli di formatore e docente nelle Scuole seguendo gli appositi percorsi FIT... Se poi decidi di continuare gli studi, puoi proseguire con un master di I livello oppure scegliere un percorso specialistico o magistrale. Le possibilità di certo non mancano!

ul:({~no} {*)) va bene!

u:({*}~FAQ7{*) In Informatica e tecnologie per la produzione del software ci si sofferma sui fondamenti teorici della programmazione, dei linguaggi e della algoritmica, mentre in Informatica e comunicazione digitale vengono approfonditi metodi e tecniche legate ad un "dominio" molto trasversale che è quello della elaborazione di testo e di linguaggi.

u:({*}~FAQ8{*) Vi sono differenze tra i corsi proposti dal dipartimento di Informatica e quelli offerti dalle varie facoltà

di Ingegneria. Sicuramente l'una ha un taglio "scientifico" e l'altra ha un taglio "ingegneristico"... Vuoi saperne di piu?

u1:({~si} {*)) Una laurea triennale corrisponde sempre e dovunque a 180 CFU, ma nella laurea in informatica ci sono 108 crediti di discipline informatiche, mentre solo 54 per i corsi del politecnico.

u1:({~no} {*)) va bene!

u:({*}~FAQ9{*)) \pau=300\ Tutti i corsi di laurea offerti dal DIB danno accesso all'Esame di Stato di abilitazione alla professione di Ingegnere dell'Informazione, sezione B per le lauree triennali, sezione A per le lauree magistrali.

u:({*}~FAQ10{*)) Yess, English is important. \pau=500\ Non ho un ottimo accento vero?.\pau=1000\ Le certificazioni in inglese sono molto importanti. \pau=500\ Hai avuto una grande idea nel cimentarti con l'inglese. \pau=500\ Dovrei fare pratica anche io. \pau=700\ Infatti con le certificazioni B1. B2. , C1. o C2. vengono riconosciuti crediti per l'esame di Inglese!

u:({*}~FAQ11{*)) Il progresso non si può fermare, così come è successo per la rivoluzione industriale. \pau=500\ La tecnologia e l'informatica danno la spinta al lavoratore di migliorarsi e aggiornarsi. L'informatica è pro-uomo.

u:({*}~FAQ12{*)) Le abilità richieste per l'informatica sono abilità matematiche, capacità di astrazione, ovvero saper modellare lo spazio, creatività e ragionamento logico.

u:({*}~FAQ13{*)) Non preoccuparti.! Puoi anche stare sempre a casa tua. A parte gli scherzi, \pau=300\ ascoltami: la frequenza è fortemente consigliata per aiutare lo studente nel superamento degli esami.

u:({*}~FAQ13{*)) \pau=300\ Come iscrizione all'albo sono pari, tutti gli aspetti di progettazione, sviluppo e collaudo sono differenti, ed è un aspetto più caro all'informatica.

Feature aggiuntive

In questa ultima parte del progetto, sono state aggiunte alcune feature a *Pepper*, per migliorare l'interazione con gli utenti.

Alcune di queste feature sono molto semplici, ma solite in una conversazione classica, come ad esempio la possibilità di chiedere a *Pepper* come sta, il suo nome, o salutarlo; ed altre un po' inusuali, come la richiesta di una storia, di una barzelletta, di ballare, o addirittura qualche proverbio famoso italiano.

Le diverse feature sono state implementate in diversi box all'interno del programma, ma principalmente sono poste nei *Dialog box* FAQ e Basic.

Di seguito mostriamo il codice utilizzato per la maggior parte delle feature descritte:

- Saluto di Pepper

```
concept:(saluto) ["puoi salutare" salutare salutami saluta ciao]
u:({*}~saluto{*) ) Ciao! $saluto=1
```

- Chiedere il giorno corrente

```
concept:(giorno) [giorno]
u:({*}~giorno{*) $giorno=1
```

- Chiedere quanti anni ha

```
concept:(anni) [anni "età"]
u:({*}~anni{*) Ho quattro anni sono ancora piccolino
```

- Chiedere se è umano o robot

```
concept:(umano) [umano]
concept:(robot) ["sei un robot"]
u:({*}~umano{*) No ho solo sembianze antropomorfe
u:({*}~robot{*) Si sono un robot con sembianze umane
```

- Chiedere se sa parlare

```
concept:(parlare) [parlare parli "puoi comunicare"]
u:({*}~parlare{*) Si posso parlare grazie agli speakers di cui dispongo
```

- Chiedere le leggi della robotica

```
concept:(leggiRob) ["leggi della robotica"]
u:({*}~leggiRob{*) Si le conosco vuoi sentirle?
    u1:(~si) \pau=300\la prima legge dice che un robot non può
rekar danno a un essere umano né può permettere che,\pau=300\ a
causa del proprio mancato intervento,\pau=300\ un essere umano
riceva danno.Vuoi che ti dica anche le altre due?\pau=300\
    u2:(~si)la seconda dice che un robot deve obbedire agli
ordini impartiti dagli esseri umani,\pau=300\ purché tali ordini
non contravvengano alla Prima Legge. la terza dice che un robot
deve proteggere la propria esistenza,\pau=300\ purché questa
autodifesa non contrasti con la Prima o con la Seconda Legge
    u2:(~no) va bene
    u1:(~no)Va bene
```

- Chiedere cosa sa fare

```
concept:(fare) ["cosa sai fare" fare]
u:({*}~fare{*)So dare informazioni alle persone,\pau=300\ so
raccontare storie ai bambini,\pau=300\ posso vedere quello che mi
circonda
```

- Chiedere di farsi un selfie o una foto con lui

```
concept:(selfie) [selfie foto]
u:({*}~selfie{*) Va bene fammi mettere in posa $foto=1
```

- Chiedere se è un maschio o femmina

```
concept:(genere) [maschio femmina maschietto femminuccia]
u:({*}~genere{*) non so,\pau=300\ io sono un robot
```

- Chiedere se ha degli amici

```
concept:(amici) ["hai amici" "hai degli amici" amici]
u:({*}~amici{*) Si al dipartimento di informatica ci sono Nadia,
Veronica, Paolo, Enzo, Giacomo e Enrico con cui chiacchiero spesso
```

- Chiedere di raccontarle una storia o una barzelletta

```
concept:(storia) [storia]
concept:(barzelletta) [barzelletta battuta]
u:({*}~barzelletta{*) Certo \pau=300\ senti questa! \pau=300\ Sai
qual'è lo sport preferito da noi robot? \pau=500\ Braccio di
ferro. \pau=300\ Vuoi saperne un'altra?
  u1:({~si} {*) ) Arancia non va a fare la spesa, perché
mandarino.
```

- Fare dei complimenti

```
concept:(complimenti) [bravo bravissimo]
u:({*}~complimenti{*) Grazie mille!
```

- Chiedere chi è il suo creatore, sua madre o suo padre

```
concept:(creatore) ["chi ti ha creato" "chi ti ha costruito" "chi
è il tuo creatore" "tuo padre" "tua madre"]
u:({*}~creatore{*) Mi ha creato l'aldebaran robotics \pau=300\ ma
chi mi ha programmato sono i ragazzi del dipartimento di
informatica!
```

- Chiedere il suo nome o chi è

```
concept:(chi) ["chi sei" "come ti chiami" "qual è il tuo nome"]
u:({*}~chi{*) Io mi chiamo Pepperr
```

- Dire che è bello o "ti voglio bene"

```
concept:(bene) ["ti voglio bene" "ti amo" "sei * bello"]
u:({*}~bene{*) \pau=300\ Grazie.! \pau=300\ Molto gentile!
```

- Chiedere a Pepper di completare un proverbio

```
concept:(intenditore) ["a buon intenditore"]
concept:(caval donato) ["a caval donato"]
concept:(mali estremi) ["A mali estremi"]
concept:(canche abbaia) ["can che abbaia"]
concept:(semina vento) ["chi semina vento"]
concept:(donna volante) ["Donna al volante"]
concept:(fra i due litiganti) ["Fra i due litiganti"]
concept:(patti chiari) ["Patti chiari"]
u:({*}~intenditore{*) Poche parole!
u:({*}~caval donato{*) Non si guarda in bocca!
u:({*}~mali estremi{*) Estremi rimedi!
u:({*}~caval donato{*) Non si guarda in bocca!
u:({*}~canche abbaia{*) Non morde!
```

```
u:({*}~seminavento{*) raccoglie tempesta!  
u:({*}~donnavolante{*) Pericolo costante!  
u:({*}~duelitiganti{*) il terzo gode!  
u:({*}~pattichiari{*) Amicizia lunga!
```

- Chiedere a Pepper il coordinatore dell'orientamento

```
concept:(coordinatore) ["chi è il coordinatore" "chi è la  
coordinatrice"]  
u:({*}~coordinatore{*) \pau=300\ La coordinatrice  
dell'orientamento consapevole è la Professoressa Antonietta Lanza.
```

- Chiedere a Pepper se ascolta musica

```
concept:(musica) ["che musica ascolti" "che genere di musica"]u:  
({*}~musica{*) \pau=300\ \rspd=60\ \vct=160\ Anima miaa,  
\pau=150\torna a casa tuaa \pau=150\ ti aspetterò dovessi odiare  
queste mura
```

- Chiedere a Pepper uno scioglilingua

```
concept:(scioglilingua) [scioglilingua filastrocca] u:({*}  
~scioglilingua{*) Trentatré trentini entrarono in Trento tutti e  
trentatré trotterellando trecentotrentatré trentini entrarono in  
Trento tutti e trecentotrentatré trotterellando  
tremilatrecentotrentatré trentini entrarono in Trento tutti e  
tremilatrecentotrentatré trotterellando
```

Potrai notare tante istruzioni simili a questa \pau=n\. Noterai con l'esperienza che i punti o le virgole a volte non bastano per far fare a pepper pause soddisfacenti, quindi ci aiutiamo con questa istruzione per prolungare la stessa pausa (n), per rendere il discorso complessivamente più lento.

[Informazioni DIB](#)

Il robot pepper è stato coinvolto, come detto in precedenza, nel dare informazioni sull'orientamento come per esempio fruire informazioni sulla presentazione del dipartimento o sulle FAQ precedentemente spiegate.

Non solo informazioni che vertono sull'orientamento, ma grazie a questo progetto, fruiremo anche informazioni più tecniche che possono interessare gli studenti e i professori del dipartimento di informatica.

Con questo progetto interfacciamo Choreographe con un programma svolto in PROLOG che è un linguaggio di programmazione che adotta il paradigma di programmazione logica, svolto per il corso di Artificial Intelligence. Il popolamento del DB Prolog presenta le seguenti informazioni:

- Sono stati considerati solo i corsi di laurea presso la sede di Bari, ovvero "Informatica", "Informatica e Tecnologie per la Produzione del Software" e "Computer Science", e non sono stati presi in considerazione i corsi presso la sede di Brindisi, ovvero "Informatica e Comunicazioni Digitali" e presso la sede di Taranto, ovvero "Sicurezza Informatica"
- Non sono stati considerati Master o Dottorati di ricerca
- Per tutte le informazioni riguardanti i corsi di laurea e le materie, sono stati memorizzate solo le informazioni riguardanti l'anno accademico 2017/2018 e 2018/2019
- Le informazioni sulle materie sono state considerate solo sulle materie del corso di Laurea in Computer Science, ove queste disponibili.

I macro-argomenti sono:

- Dipartimento
 - Tirocinio
 - Erasmus
 - Moduli
 - Avvisi
 - News
 - Eventi
 - Ada
 - Tipologia segreteria
 - Tutoring
- Segreteria Didattica
 - Giorni d'apertura
 - Orari d'apertura
- Segreteria Amministrativa
 - Giorni d'apertura
 - Orari d'apertura
 - Test d'ingresso
 - Wi-Fi
- Segreteria Esse3
 - Descrizione
 - Utilità
 - Sito Web
- Pagamenti
 - Regolamento tasse
 - 1^ rata
 - 2^ e 3^ rata
 - Riduzioni

- Maggiorazioni
- Simulazioni
- Software
 - Elenco Software
 - Informazioni Microsoft
 - Informazioni Office365
 - Informazioni IBM
- ADISU
 - Telefono ADISU
 - E-mail
 - Graduatorie
 - Bando
 - Orari d'apertura
 - Giorni d'apertura
 - Modulistica
 - Mensa
 - Sede
- Corsi di laurea
 - Calendario lezioni
 - Regolamento didattico
 - Elenco corsi di laurea
 - Gruppi FaceBook
- Materia
 - Elenco materie
 - Materiale didattico
 - Descrizione materia
- Professore
 - Telefono
 - Sito Web
 - E-mail
 - Orario ricevimento
- Distributori
 - Informazioni distributori
- Stanza
 - Piano
 - Orario
 - Occupazione
 - Elenco laboratory
 - Elenco aule

Sono state stilate da noi anche delle possibili domande che possono essere poste al robot pepper:

1. Dov'è l'ufficio del prof ferilli?
2. Hai il materiale didattico di artificial intelligence dell'anno accademico 18/19?
3. A che ora apre la biblioteca?
4. Sai la situazione occupazione aule?
5. Hai l'elenco delle materie di itps?
6. Quali materie insegna ferilli?
7. Quali sono i laboratori del nostro dipartimento?
8. Quali sono le aule del nostro dib?

9. Qual è la descrizione della materia information system dell'anno accademico 18-19?
10. Hai qualche informazione sul tutoraggio?
11. Sai darmi il regolamento didattico di computer science dell'anno accademico 17/18?
12. Sai dirmi l'elenco dei corsi di laurea attivati?
13. Qual è il numero dell'adisu?
14. Sai dirmi il numero di telefono del prof ferilli?
15. Sai dirmi il suo sito web?
16. Il prof ferilli ha un sito web?
17. A che ora riceve il prof ferilli?
18. Sai dirmi la mail della professoressa decarolis?
19. Hai l'email dell'adisu?
20. Dove posso prendere un caffè?
21. Vorrei sapere qualcosa sul tirocinio
22. Come posso fare l'erasmus?
23. Dove posso scaricare i moduli del dipartimento?
24. Ho perso l'ultimo avviso del prof, dove posso trovarlo?
25. Ci sono novità riguardo il dib?
26. Vorrei partecipare a qualche evento del dib, sai dirmi se ci sono?
27. A che ora apre la segreteria amministrativa?
28. A che ora apre la segreteria didattica?
29. Che giorni è aperta la segreteria didattica?
30. Che giorni chiude la segreteria amministrativa?
31. Come mi collego alla rete wifi?
32. In cosa consiste il test d'ingresso?
33. Hai il link di ada?
34. Dove posso scaricare il regolamento tasse?
35. Di quanto è la prima rata?
36. Hai info sulla terza rata?
37. Come potrei ridurre le mie tasse?
38. Ho ricevuto una maggiorazione sulla tassa, perchè?
39. Hai il link del simulatore delle tasse?
40. Dove posso trovare il bando ADISU?
41. A che ora apre l'adisu?
42. Mi parlano di modulistica adisu, sai dove potrei scaricarla?
43. Come potrei usufruire della mensa?
44. Dove è situata l'adisu?
45. Sai dirmi le graduatorie adisu?
46. Cosa è esse3?
47. A cosa serve esse3?
48. Hai il link di esse3?
49. Che software fornisce il dib?
50. Come posso scaricare microsoft?
51. Quali tipi di segreterie ci sono?

L'architettura del sistema è rappresentata in questo diagramma a blocchi:

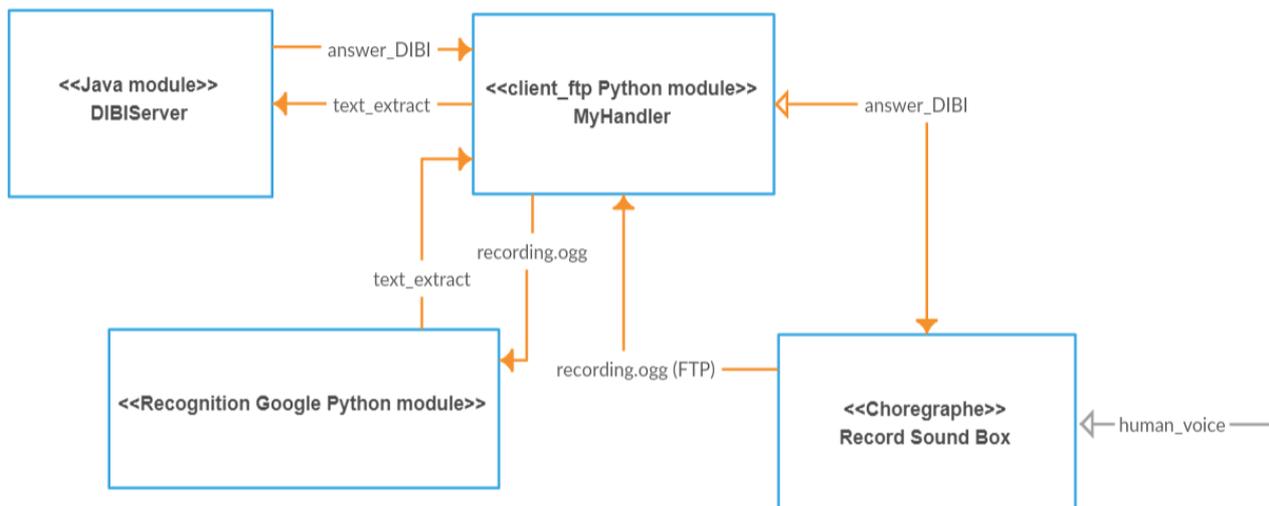


Figura 69: Architettura Sistema

Il sistema creato presenta due server uno in Java e uno in Python. Per prima cosa il robot ascolterà la domanda posta dallo studente, che verrà registrata grazie ad un box Choregraphe chiamato “Record Sound”. Attraverso un box Python in Choregraphe è stata importata una libreria python “ftplib”, una libreria che implementa il protocollo FTP. Usando FTP possiamo creare e accedere a file remoti attraverso chiamate di funzione.

In questo modo il robot registrerà il file denominato “recording.ogg”, scaricandolo sul pc grazie al protocollo FTP client-server, tutto inserito nel modulo python “client_ftp”. In questo modulo poi, abbiamo catturato l’evento del trasferimento del file FTP ed abbiamo invocato il modulo “recognition”, che ha convertito il file formato audio in formato testuale, grazie alle API di google.

Dopo di che il modulo python “server_ftp” ha inviato il testo, ricavato dalla conversione, al server Java “DIBIServer” che ha interrogato il database Prolog restituendo la risposta testuale al modulo python “server_ftp”, che infine permetterà al robot pepper di pronunciare la risposta alla domanda iniziale. Tutto questo illustrato nella figura 69.

Riferimenti

- [1] Robotiko, "Pepper, tutte le informazioni che stavi cercando su questo robot umanoide", <https://www.robotiko.it/pepper/>
- [2] Robotiko, "Robot Pepper: prezzo e caratteristiche del robot più amato", <https://www.robotiko.it/robot-pepper-prezzo/>
- [3] Robotiko, "Pepper robot: cosa può fare l'umanoide che comprende le tue emozioni", <https://www.robotiko.it/pepper-robot-cosa-puo-fare/>
- [4] Honda, «Innovation,» Honda, <http://asimo.honda.com>
- [5] iCub, <http://www.icub.org>
- [6] Surena Humanoid, <http://surenahumanoid.com/>
- [7] SoftBank, «Romeo,» <https://www.ald.softbankrobotics.com/en/robots/romeo>
- [8] Il Messaggero, «E' nato R1, il primo "robot domestico",» https://www.ilmessaggero.it/tecnologia/scienza/r1_robot_domestico_italiano_case_ospedali-1864776.html.
- [9] J. Dolcourt, 06 01 2016, <https://www.cnet.com/products/ubtech-alpha-2/preview/>.
- [10] Toyota, «Innovation», http://www.toyota-global.com/innovation/partner_robot/concept.html
- [11] De Agostini, De Agostini, <http://www.hellorobi.com/it/>.
- [12] «Robotiko - WalkMan,» <https://www.robotiko.it/walk-man-robot-umanoide/>.

- [13] SoftBank, «Nao,» Available: <https://www.aldebaran.com/en/robots/nao>.
- [14] Aldebaran Documentation 2.4.3, http://doc.aldebaran.com/2-4/home_pepper.html
- [15] Ignasi Lleonsi Carrillo, “Development of teaching assistance application for SoftBank Pepper” – University of Girona
- [16] Softbank Robotics, “Pepper Handbook” - https://www.softbankrobotics.com/us/sites/all/themes/sbr_global_web/images/support/pepperhandbook.pdf
- [17] Aldebaran Documentation 2.4.3, “Pagina Web Pepper”- http://doc.aldebaran.com/2-4/family/pepper_user_guide/webpage.html
- [18] Aldebaran Documentation, “Impostazione della connessione WiFi di Pepper” - http://doc.aldebaran.com/2-4/family/pepper_user_guide/connecting_pep.html#robot-connect-pep
- [19] Aldebaran Documentation 2.4.3, “Suite Choregraphe – Guida all’installazione” - <http://doc.aldebaran.com/2-4/software/choregraphe/installing.html>
- [20] Aldebaran Documentation, “Lingue supportate” - http://doc.aldebaran.com/2-4/family/pepper_technical/languages_pep.html
- [21] Aldebaran Documentation, “Hello world 2 – Usando python in coreographe” - http://doc.aldebaran.com/2-4/getting_started/helloworld_choregraphe_script.html
- [22] Aldebaran Documentation, “Tutorial ALTextToSpeech” - http://doc.aldebaran.com/2-4/getting_started/helloworld_choregraphe_script.html
- [23] Aldebaran Documentation, “Hello World 3 – Usando l’argomento Dialog in Coregraphe” - http://doc.aldebaran.com/2-4/getting_started/helloworld_choregraphe_dialog.html
- [24] Aldebaran Documentation, “Esempio 1: una prima applicazione di ballo” - http://doc.aldebaran.com/2-4/getting_started/samples/sample_dance1.html
- [25] Aldebaran Documentation, “Esempio 2: una prima applicazione interattiva”
- [26] Aldebaran Documentation, “Creare un’applicazione da zero” - http://doc.aldebaran.com/2-4/software/choregraphe/tutos/create_application.html
- [27] Aldebaran Documentation, “Uso del tablet Pepper” - http://doc.aldebaran.com/2-4/getting_started/creating_applications/using_peppers_tablet.html