



Costruire microrobot e programmarli

di Giovanni Marcianò (<http://margi.bmm.it>)

BIOGRAFIA

dal 1992 segue l'impiego didattico delle TIC, dal 1999 si occupa di metodologia didattica nelle azioni di FAD per il personale della scuola, rappresenta il mondo della Scuola nel Direttivo della *SIeL - Società Italiana per l'e-Learning*. Promuove da anni il costruttivismo come modello d'impiego delle TIC nella scuola dell'infanzia ed elementare.

SOMMARIO

Premessa e guida alla lettura

Dal Logo al mattoncino programmabile

Kit e linguaggi per la microrobotica

I tre teoremi della microrobotica

Conclusioni

Premessa e guida alla lettura

L'impiego della microrobotica ¹ nella scuola italiana non è molto diffuso. Vi sono esperienze sparse, nate spesso in forza di iniziative sperimentali promosse da Enti di ricerca ² e Associazioni esterne alla scuola. In altri casi è stata la passione di singoli docenti che han dato vita a "laboratori di microrobotica" coniugando questa attività a curricoli disciplinari in area tecnica e scientifica.

L'impiego in classe della microrobotica è una scelta non semplice, in quanto presuppone una dotazione specifica che, seppur non eccessiva sul piano finanziario, va proposta e deliberata, quindi condivisa e compresa. E in questo caso si rischiano due fraintendimenti:

- che l'oggetto "microrobot" sia banalizzato al rango di "giocattolo tecnologico", e come tale da tenere fuori dalle aule e dalla programmazione didattica;

¹ Mi piace usare il termine "microrobotica" per due motivi. Primo, la robotica è disciplina complessa, che ha un preciso assetto scientifico a cui è difficile riferire le attività di cui possiamo trattare a livello di scuola primaria. Secondo, il termine crea un'assonanza con altri oggetti impiegabili didatticamente - i micromondi - con cui la microrobotica convive perfettamente. Micromondi e microrobotica sono figli dello stesso padre: S. Papert; l'assonanza aiuta a cogliere lo stretto legame metodologico che spesso richiamerò in queste pagine.

² Si veda l'interessante e completa esperienza promossa a Milano, in cui molti soggetti, tra cui l'ITD del CNR di Genova, han dato vita al progetto "Costruiamo un robot". V. <http://www.museoscienza.it/microrobotica>

- che si ritenga la "robotica" un tema da Istituto superiore, considerato che coinvolge competenze complesse sia sul piano meccanico, che elettronico ed informatico.

Ma è anche un'avventura entusiasmante da vivere con la propria classe. Si sta ripetendo la dinamica già vissuta alla prima introduzione dei computer nelle scuole dell'obbligo, dell'Infanzia ed Elementari in particolare. Erano "cose da grandi", oppure "servono solo per giocare e perdere tempo". La storia della scuola dimostra che quelle opinioni erano infondate.

Per questo nella primavera di quest'anno mi son fatto promotore di una raccolta di sparse realtà nate nelle scuole italiane ³, presentandole come esperienze didattiche significative non tanto per la passione tecnologica, quanto perché in ognuna di esse si sentiva quel "senso del Logo" ⁴, quell'approccio costruttivista che sempre più si rivela una valida risposta al desiderio di apprendere dei nostri alunni.

Sono proprio loro, gli alunni, l'elemento primo di attenzione che dobbiamo avere. La loro risposta, non solo emotiva ma cognitiva, la crescita di abilità e il maturare di competenze sono l'oggetto primo della valutazione di un uso didattico della microrobotica. E in tutti i casi che conosco gli alunni hanno fatto osservare ai loro docenti evoluzioni significative; una costante in contesti scolastici, socio-culturali, didattici anche differenti.

Troverete in queste pagine i principali criteri di riferimento generale sul piano pedagogico e metodologico - didattico da porre alla base di un progetto d'impiego della microrobotica a scuola. Ma anche riferimenti specifici agli aspetti operativi, di organizzazione e promozione di un uso efficace delle potenzialità che i microrobot oggi disponibili possono offrire all'insegnante. Oltre a alcuni elementi di carattere divulgativo per comprendere le potenzialità insite nella tecnologia, da cui trarre spunto per progettualità proprie.

Essendo un tema nuovo si rischia di assistere a tante "partenze da zero". Come amo spesso ripetere non mi piace quando si comincia da zero, in una società come la nostra in cui la massa di informazioni disponibili è impressionante. Vale anche qui, sul fronte della microrobotica. Non si può dire di essere all'anno zero né dal punto di vista pedagogico, né da quello tecnologico. È bene esserne coscienti, se si desidera partecipare a questa esperienza.

L'impiego di piccoli robot programmabili è presente da sempre nelle teorie di Papert relative all'impiego delle nuove tecnologie per l'apprendimento e pochi sanno che il linguaggio LOGO ⁵ nacque per pilotare un piccolo robot ⁶, pratica molto più semplice dell'impiego del linguaggio sui computer di allora. L'oggetto aveva forma emisferica e per questo veniva chiamata "tartaruga", nome che poi, nelle versioni per computer nate successivamente, passò al triangolino rappresentato sullo schermo, l'oggetto grafico virtuale programmabile.

³ v. Marciànò G., "Robotica a scuola", Rassegna dell'Istruzione, Le Monnier, a. 2003/04, 4, pp. 6-20.

⁴ v. Papert S., *Introduzione a "Logo Philosophy and Implementation"*, LCSl, Canada, 1999. Disponibile in rete su <http://www.microworlds.com/company/philosophy.pdf> - Traduzione italiana su <http://margi.bmm.it>

⁵ si veda in questo stesso ambiente di apprendimento "PuntoEdu" il materiale di studio "Usare il linguaggio LOGO per costruire micromondi"

⁶ L'ingegnere Grey Walter negli anni '50 costruisce la prima tartaruga elettronica, Elsa che obbedisce a semplici comandi. A Seymour Papert la tartaruga piacque a tal punto da utilizzarla per i suoi primi esperimenti con il linguaggio LOGO, costruendo appunto piccoli robot-tartaruga. In molte scuole i bambini hanno avuto a disposizione questi piccoli robot che obbedivano agli ordini impartiti con LOGO. (cfr. Ceserani G.P., *Gli automi, storia e mito*, Laterza, 1983)

Possiamo probabilmente dire che tutta l'esperienza del LOGO è un ripiego virtuale derivato dalla disponibilità di hardware a basso costo come l'home computer. Mentre gli apparati robotici restavano un lusso per ricercatori, i computer divenivano dagli anni '80 in poi la tecnologia a basso costo conciliabile con l'esigenza delle scuole. E allora si dovette riportare nel virtuale, nel micromondo rappresentato sul monitor del computer, l'oggetto digitale programmabile. Solo alla soglia del 2000 sul mercato sono apparsi kit robotici con prezzi accessibili alla scuola. Ed allora si può "tornare all'origine"? Vedremo nelle pagine seguenti come sia possibile.

1) Dal Logo al mattoncino programmabile

Seymour Papert ha coniato il termine "mattoncino programmabile" pensandone un uso didattico in continuità con la tarta del LOGO e, prima ancora, con Elsa, la tartaruga elettronica di W. Grey. Intervistato a Venezia il 7 marzo 1997 alla domanda "La matematica, dunque, dovrebbe essere insegnata in un modo più pratico?" Papert rispose:

*Dovremmo insegnare una matematica diversa. La matematica che insegniamo a scuola è completamente inutile, mentre essa costituisce un modo di pensiero attivo. E' quindi inutile cambiare solo il modo di insegnare la stessa materia. Dobbiamo creare, viceversa, un nuovo contesto. Penso, per esempio, che i bambini possano apprendere un modo per realizzare dei meravigliosi progetti con i computer, come costruire dei robot, o fare dell'arte computerizzata con la realizzazione di spettacoli multimediali. Per fare questo hanno bisogno di un'importante matematica: la matematica che una volta era impossibile insegnare, ma che ora i bambini possono apprendere. Ciò ha un senso solo cambiando l'intera struttura di quello che pensiamo.*⁷

L'obiettivo di Papert non è mai stato quello di formare dei "programmatore" più o meno precoci. In questo equivoco giacciono molte superficiali critiche alle didattiche costruttiviste, viste come banali esercizi tecnologici. Invece l'attenzione è centrata sull'obiettivo di far impiegare ai bambini tecnologie digitali programmabili come strumento potente per concepire ed esprimere progetti personali, carichi di significato, quasi una bacchetta magica per esprimere l'infinita creatività dei bambini.

Queste intenzioni – dagli anni '60 in poi – han dovuto confrontarsi con l'hardware disponibile e accessibile alle scuole, come pure a specifiche versioni di linguaggi di programmazione che dovevano essere adeguati all'uso da parte dei bambini, come fu il caso del Logo. Nacquero quindi progetti essenzialmente grafici comprensivi di animazioni (la geometria della tartaruga e i micromondi), pagine di testi scritti con illustrazioni composte da disegni elaborati dai bambini al computer (LogoWriter), progetti di robotica (LegoLogo), sino ad includere la possibilità di elaborare oggetti ipermediali (MicroWorlds).⁸

Ora si rende possibile un ritorno al contatto diretto, non più reso virtuale dal computer, manipolando l'hardware e i linguaggi di programmazione specifici. Questo contatto diretto con qualcosa di concreto, programmabile, mobile è possibile grazie ai microrobot. E quindi, come successe negli anni '70 e '80 all'apparire sul mercato dei

⁷ v. Papert, S. *Bambini e adulti a scuola con il computer*, MediaMente - Biblioteca digitale, <http://www.mediamente.rai.it/home/bibliote/intervis/p/papert.htm>

⁸ cfr. Varisco B. M., *Nuove tecnologie per l'apprendimento*, Garamond, 1998.

primi home-computer, così ora è possibile portare a diretta disposizione dell'alunno apparecchiature sino a ieri riservate a centri universitari o al massimo a istituti tecnici o professionali.

Sul fronte del software specialistico, la programmazione di un robot richiede alcuni elementi peculiari per pilotare attuatori e leggere dati dai sensori. Ma per il resto sono presenti gli elementi comuni ad ogni linguaggio di programmazione. Prima di entrare nel merito della proposta didattica, è comunque necessario esplicitare maggiormente i caratteri di fondo che differenziano la microrobotica dall'informatica. In una logica divulgativa, ma anche con sufficiente dettaglio perché l'insegnante che intende procedere all'approccio alla microrobotica in classe possa farlo con coscienza di causa.

a) Hardware robotico

Tra un robot e un computer l'elemento comune è rappresentato dal microprocessore, la differenziazione sta invece sul campo delle periferiche di input/output. Mentre un computer riceve i dati da una tastiera, o da uno scanner, ad esempio, il robot riceve dati da sensori che reagiscono all'ambiente.

Ci sono sensori molto semplici, come quello che segnala al robot di aver toccato un ostacolo, che in pratica è solo un piccolo interruttore a pressione, che scatta se viene "premutato" dal robot stesso in movimento contro l'ostacolo.

Ci sono sensori molto complessi che - ad esempio - sono in grado di "sentire" caratteristiche fisiche (temperatura) o chimiche (acidità) dell'ambiente in cui si trovano ad operare.

Mentre il computer usa i dati in entrata per realizzare delle elaborazioni (un testo ben impaginato, una tabella ordinata e corretta, una lista in perfetto ordine alfabetico, una immagine senza difetti, un certo suono o motivo musicale), il robot usa i dati in entrata dai suoi sensori per attivare i suoi motori e - quindi - muoversi. O decidere di non muoversi.

Mentre il computer necessita di stampanti e monitor per fornire il risultato dell'elaborazione, il robot necessita di motori, leve, ingranaggi per compiere i movimenti che lo qualificano come tale.

Come avviene in un computer, tra input e output, a regolare l'attività della macchina c'è il software.

b) Software robotico

Il software che regola un robot opera secondo gli stessi processi impiegati dal software di un computer. Come il software di un computer dipende dalla presenza o meno di alcune periferiche (inutile chiedere ad un programma di videoscrittura di stampare l'elaborato se non vi è una stampante collegata), ancora più strettamente il software robotico dipende dalla disponibilità di sensori e attuatori per poter funzionare correttamente.

Quindi un primo elemento importante del software robotico è la corretta definizione del robot da pilotare. Un po' come le "specifiche minime" che troviamo sulla confezione o sulla scheda tecnico di un software per computer: sistema operativo, versione, memoria, scanner e così via. Per il software robotico avremo modello, sistema operativo, sensori, attuatori.

Definite le caratteristiche di fondo, si possono programmare robot di diverse categorie, specializzati in compiti specifici. Qui si propongono tre tipologie che possono essere di riferimento all'impiego didattico: robot ripetitivi, robot obbedienti e robot intelligenti.

i) Robot ripetitivi

Tutti abbiamo almeno una volta visto l'immagine di un robot industriale. Esegue velocemente e con massima precisione una operazione che una volta era invece affidata all'operaio nella catena di montaggio. Quando dico "robot ripetitivi" mi riferisco a robot programmati per ripetere una operazione semplice o complessa con massima precisione, possibilmente in tempi ridotti. A cosa serve quello che fa non mi interessa, mi interessa che la programmazione del robot è di tipo deterministico, per poter avere il pieno controllo delle sue mosse, che dovranno corrispondere al mandato assegnato.

Il fatto che si debba programmare in modo prescrittivo rende l'operazione gestibile a livelli di competenza informatica elementare. In linguaggio LOGO è sufficiente conoscere i primi semplici comandi.

ii) Robot obbedienti

Cominciamo ad entrare un po' di più nella microrobotica. Un robot obbediente deve essere in grado di rispondere in modo definito (dal software) ad uno stimolo preciso: una luce, un suono, un tocco ...

A differenza del robot ripetitivo questo robot può fare molte cose diverse, in risposta a stimoli diversi. Facile programmarlo a dare una risposta ad uno stimolo, più complesso dare risposte a stimoli complessi.

Didatticamente in questo caso si presentano diversi compiti impegnativi che comportano da un lato l'impiego di una programmazione in grado di discriminare tra tanti input quelli significativi, a cui rispondere, dall'altro un adattamento costante alla presenza / assenza dello stimolo, o a sue variazioni di posizione, intensità ecc.

iii) Robot intelligenti

Qui giungiamo a quella che molti intendono essere la vera robotica: un apparato in grado di esprimere un comportamento in grado di risolvere specifici problemi. Non è pensabile, ovviamente, che a livello scolastico si possa giungere ai livelli ormai presenti nella robotica, tuttavia si può proporre un approccio che per quanto più semplice esprime un'intelligenza operativa.

Diamo all'alunno il compito di sviluppare procedure software che mettano in grado il robot di affrontare e risolvere problemi definiti, ma non determinati. Ovvero, non si dichiara esattamente cosa si deve fare, ma una situazione di tipo generico, che quindi può concretizzarsi in forme differenti, ma di fronte alle quali il robot deve riuscire a comportarsi in un determinato modo, raggiungendo un definito obiettivo.

iv) Teoremi microrobotici

Come ho proposto spunti di lavoro con la classe in forma di teoremi per la costruzione di micromondi con il linguaggio LOGO, ugualmente per la microrobotica ho raccolto in tre teoremi spunti di lavoro con la classe riferiti alle tre tipologie di microrobot qui indicati. Ma prima di affrontare una proposta didattica dobbiamo attrezzarci: e per farlo sarà bene orientarci tra i kit utilizzabili, e i linguaggi specifici disponibili.

2) Kit e linguaggi per la microrobotica

S. Papert vedeva nel "mattoncino programmabile" qualcosa di tecnologico che l'evoluzione della microelettronica rendeva possibile. Negli anni '70 i primi home-computer, poi a fine anni '80 i sistemi multimediali, e a fine XX secolo la microrobotica. Passi di un'evoluzione che ha reso tecnologie molto sofisticate alla portata di tutti. Nella premessa del volume *Logo philosophy and implementation*⁹ egli mantiene unite le diverse attività che queste diverse tecnologie permettono in classe nell'unico contesto del costruttivismo, per cui

"un bambino [impara] costruendo una torre, scrivendo un racconto, costruendo un apparato robotico o realizzando un videogame: sono tutti esempi di costruzioni e la lista potrebbe proseguire all'infinito. Tutte queste attività hanno molti aspetti in comune. Sono soggetti alla verifica della realtà; se non funzionano divengono una sfida a capire perché e a superare l'ostacolo."

Il riferimento metodologico didattico diviene importante perché unico è il processo di apprendimento, in questo contesto costruttivista come in altri contesti. E se "costruire" un robot o realizzare un videogioco sembrano cose diverse, in realtà attivano processi di apprendimento molto coerenti e simili. L'elemento unificante di questi processi di apprendimento è il linguaggio. Sia il linguaggio naturale, sollecitato dai fatti a esprimere relazioni logiche, ipotesi, progetti, sia il linguaggio informatico chiamato a dare corpo a quanto ipotizzato, immaginato, progettato.

Papert non ha mai pensato di formare dei "programmatori" in LOGO, eppure si trova nel 1999 a ragionare sulla "Filosofia del LOGO" affermando che

"... molti progetti che usano il Logo [sono] riferiti ad un certo "senso del Logo". E, d'altro canto, immagino, ma anche ho visto alcune volte, alcuni progetti basati sull'impiego del computer essere simili nello spirito e nelle finalità a quelli descritti in questo libro pur usando un altro linguaggio di programmazione. E allora la domanda diventa: 'Cos'è il senso del Logo?' E ancora 'Perché questo senso lo si ritrova raramente in attività didattiche con le TIC senza il Logo?' "

Questa premessa perché non è semplice affrontare il delicato problema della scelta di un kit e di un linguaggio di programmazione per apparati microrobotici, un linguaggio che possa essere appreso e usato in modo semplice - ma efficace - a livello di scuola elementare. Dato che negli anni '70 e '80 l'idea dell'impiego a scuola delle tecnologie si accompagnava alla disponibilità di un linguaggio - il LOGO - che permettevano l'utilizzo di un vero computer in aula e quindi di svolgere una didattica attiva, e non più limitata allo studio delle teorie informatiche, ora ci attenderemmo la disponibilità di una versione del LOGO "robotico", in grado di pilotare hardware robotico a basso prezzo.

Vorrei comunque chiarire che qualunque sia comunque il kit e il linguaggio scelto dalla scuola, la proposta che qui si fa di "teoremi microrobotici" a cui ispirare la programmazione didattica punta a **finalizzare la microrobotica allo sviluppo**

⁹ Papert, *op. cit.*

cognitivo degli alunni, e non solo, come è frequente osservare, allo sviluppo di competenze tecnologiche.

Se si accetta l'impiego della lingua inglese, allora sono facilmente reperibili in rete tool anche LOGO based per programmare i più diffusi mattoncino programmabili, come l'RCX della Lego ¹⁰ e Cricket sviluppato dal M.I.T. ¹¹. Segnalo qui solo alcune di queste risorse "Logo based", per mostrare i diversi approcci possibili per coniugare Logo e microrobot:

- Yellow brick logo ¹² – un micromondo per MicroWorlds 2 ¹³ che permette di programmare il mattoncino della Lego. Documentazione sul sito della scuola *Stokane National School* (Irlanda) ¹⁴
- MicroWorlds EX robotics – un'evoluzione specifica del pacchetto della LCSi ¹⁵, le cui potenzialità si possono intuire – oltre che dal sito della LCSi - dalla documentazione dei progetti seguiti dall'IML – *Independent Museum Learnings* (US) ¹⁶ con il kit Cricket
- MSWBricks ¹⁷ - che permette di programmare il mattoncino della Lego usando MSWLogo, una delle più diffuse versioni *free* del Logo
- Logo blocks ¹⁸ – un tool di programmazione visual, che permette di scrivere le procedure montando il diagramma di flusso con blocchi logici riferiti alle primitive Logo. Un sistematico impiego di Logo blocks con Cricket nella documentazione della sezione Robotica del progetto TEDDi ¹⁹ promosso dall'Università di Alicante.

Attorno all'hardware disponibile, sono anche nati diversi gruppi di appassionati che hanno - in questi anni - sviluppato linguaggi ad-hoc, molti secondo la logica dell'open-source e della collaborazione in rete. La ricerca del linguaggio migliore per usare a scuola uno dei kit microrobotici è possibile. Tra tool proprietari, versioni di Logo, e linguaggi open-source la scelta è ampia. In sitografia riporto alcune risorse. L'unico problema che tuttavia resta è la mancanza di una versione italiana, un falso-problema forse, ma che comunque ho voluto affrontare trovando anche una possibile soluzione.

Ho scelto tra i progetti open-source in corso NQC (Not Quite C), un linguaggio che permette, attraverso macro, di poter introdurre comandi in italiano per agli alunni. NQC ha una struttura di programmazione simile a quella del Logo di MicroMondi 2, ed è facilmente impiegabile tramite il tool BCC (Bricx Command Center). Nel BCC è disponibile un editor molto valido. Vediamo i dettagli di questa risorsa gratuita, e i possibili adattamenti alla portata di chi abbia un minimo di confidenza con l'impiego di macro.

¹⁰ La Lego TM ha un ampio catalogo di kit, tutti sviluppati attorno al mattoncino RCX, di cui sono distribuite due versioni: la 1.0 e la 2.0. Per approfondimenti <http://www.lego.com/eng/education/mindstorms/>

¹¹ Dalle ricerche del M.I.T. ha visto la luce una scheda robotica adatta ad impieghi didattici, denominata "Cricket". Informazioni e documentazioni sul sito <http://llk.media.mit.edu/projects/cricket/about/index.shtml> . Il M.I.T. ha poi rilasciato la licenza gratuita sul suo prodotto per usi didattici e di ricerca, Il che vuol dire che può anche essere acquistata al solo prezzo di costo. La versione ultima è "Handyboard". v. <http://www.handyboard.com/>

¹² <http://homepage.tinet.ie/~stokanens/robotics/robotics.htm>

¹³ purtroppo funziona solo con la rel 2.05, successiva alla 2.04 su cui è realizzata a versione italiana MicroMondi 2.

¹⁴ <http://homepage.eircom.net/~stokanens>

¹⁵ http://www.microworlds.com/solutions/demo_ex_rob.html

¹⁶ <http://lrc.smm.org:7780/mms/Students01/amy>

¹⁷ <http://www.southwest.com.au/~jfuller/rcx/rcx.htm>

¹⁸ <http://llk.media.mit.edu/projects/cricket/doc/>

¹⁹ <http://www.teddi.ua.es/>

a) NQC – un linguaggio open source per la microrobotica

Bricx Command Center si basa sul linguaggio NQC ²⁰sviluppato da David Baum ²¹. La prima cosa che mi ha colpito di NQC è il tutorial di Mark Overmars: sembrava scritto proprio per un impiego a livello elementare, sviluppandosi dal più banale programma gradualmente sino ai livelli più complessi ²². Il tutorial presenta molte procedure, spiegando per filo e per segno i comandi e la loro sintassi. Si parte dal far muovere il microrobot, sino a programmarne il comportamento in gruppo.

Tutte le procedure citate nel tutorial sono pronte da essere caricate in BCC. Un aiuto prezioso per entrare subito in azione. Scoprire poi che Andrea Molteni ²³ aveva tradotto in italiano ²⁴ il tutorial mi ha convinto: è possibile un facile approccio a questo linguaggio nonostante che - sia chiaro - non sia nato per i bambini, non è certo il LOGO. Ma nonostante questo in classe ho potuto verificare con le insegnanti quanto la struttura delle procedure di NQC sia simile alle procedure di MicroMondi.

c) BCC – una collezione di tool

Il tutorial permette a tutti di mettere subito le mani nel linguaggio NQC. Il tool che permette di impiegare NQC è BCC (Bricx Command Center). Comprende oltre all'editor dei programmi anche una serie di strumenti che ampliano il ventaglio d'impiego del microrobot, con e senza ricorso a NQC. Come dice il nome BCC è davvero un "centro di comando" del mattoncino. Permette infatti non solo di scrivere programmi, ma attraverso il menu "Tools" di avere subito a disposizione altri modi di utilizzare il nostro microrobot.

Vari elementi a favore di NQC, quindi, rinforzati infine dall'interfaccia - anche questa non nata per i bambini - ma di certo molto standard: vi sono le normali icone impiegate in tutti gli altri software per aprire, salvare, copiare, incollare ecc. il listato del programma. Insomma, di certo né la barra dei menu, né la barra dei pulsanti disorienta gli alunni (e gli insegnanti) che hanno un minimo di confidenza col computer. Il fatto poi che siano già pronti tutti i listati utilizzati nel tutorial per prendere confidenza con la programmazione del microrobot non può che essere un'altra positiva scoperta.

Con "Direct control" si hanno sul video tutte le "parti" del robot sotto controllo: hardware e software. Con "Diagnostics" si può indagare facilmente a caccia di cosa non funziona, ma anche gettare una curiosa occhiata nel cuore del microrobot. Con "Watching the Brick" possiamo chiedere lo stato di ogni pezzo, sensore, motore, variabile, programma, impostando anche la visualizzazione grafica dei dati, da ricevere in un colpo solo, a intervalli, in flusso continuo.

Sin qui Tools un po' troppo sofisticati per gli alunni, ma che possono facilitare gli insegnanti nella "manutenzione" del microrobot. Subito dopo i tool tecnici troviamo "Brick Piano". Chi

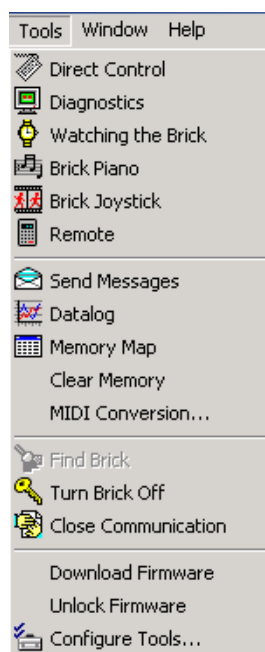


Figura 1 Il menu dei tool disponibili in BCC

²⁰ v. <http://bricxcc.sourceforge.net/nqc/>

²¹ David Baum opera presso il Dipartimento di Computer Science, all'Utrecht University nei Paesi Bassi.

²² v. <http://www.cs.uu.nl/people/markov/lego/> da cui è possibile scaricare il tutorial in varie lingue, anche in italiano.

²³ v. <http://counter.li.org/cgi-bin/runscript/display-person.cgi?user=245409>

²⁴ scaricabile da http://www.cs.uu.nl/people/markov/lego/tutorial_i.doc

conosce MicroMondi non può che fare un balzo ritrovando qui uno strumento decisamente simile a quello del pacchetto della LCSI. Si può far suonare il robot, salvare e utilizzare il brano dentro i nostri programmi NQC. Più sotto nella tendina del menu di BCC una voce "MIDI converter" preannuncia impieghi "musicali" decisamente efficaci e facili!

Dalla musica al pilotaggio: "Brick Joystick" dice da sé cosa permette di fare ... un uso ludico ma anche un approccio più semplice per chi comincia, per familiarizzare con l'RCX. Permette di usare un vero joystick, o il mouse o la tastiera per pilotare un robot. Per sterzare son previsti due configurazioni: l'uso di due motori (ruota destra e sinistra separate) oppure un motore per la trazione e uno per lo sterzo. I "bottoni di sparo" del joystick, o i pulsanti del mouse attivano i programmi che avremo scaricato nel robot. Sterminati spazi di attività didattiche possibili! Tutte incentivanti per conoscere meglio il robot e come programmarlo a nostro gusto!

E poi "Remote": un telecomando virtuale che nessuno avrà problemi ad usare, controllando così il microrobot. L'attuale versione pesa solo 2,9 Mb (avete letto bene ... anche con un semplice modem in dial-up si scarica facilmente)²⁵. La versione 3.3 build 3.3.7.7 è del gennaio 2004, e gira perfettamente sotto le diverse versioni di Windows™. Vi è poi una versione per computer Mac (MacNQC²⁶) ed anche le direttive per poter girare sotto Linux, con driver specifici²⁷ che fan ormai parte del linux kernel 2.6.1.

Una nota sull'editor. Sembra nato per far imparare chi lo usa. Già digitando i singoli elementi vengono controllati e - se digitati correttamente - colorati secondo un codice che può anche essere configurato a piacere: comandi in azzurro, valori in rosso ecc. Digitare una lettera minuscola mentre era da digitare maiuscola (NQC è case-sensitive) non fa colorare l'elemento (v. esempio a lato, Out invece che OUT), portando subito all'occhio l'errata digitazione.

<pre>OnRev(OUT_C); OnFwd(Out_B); Wait(10);</pre>
<p>Figura 2 nero = errore di battiture</p>

Oltre alla correzione ortografica, al momento della compilazione vi è un controllo anche sintattico sul listato. Errori logici vengono elencati al fondo della pagina. Cliccando la segnalazione viene evidenziata la riga in cui è scattato l'errore. Una funzione professionale ma resa in modo semplice ed efficace.

Tutto ciò che era possibile fare per semplificare il complesso processo di programmazione del microrobot è stato pensato. Il listato con un solo clic viene compilato, controllato, trasmesso del microrobot tramite la porta a infrarossi. Se tutto va a termine il microrobot emette un bip. A questo punto il pulsante di avvio fisicamente presente del microrobot, oppure un pulsante virtuale sulla barra di BCC lanciano il programma. Allo stesso modo - reale o virtuale - si opera per bloccarlo.

d) Le macro di NQC per avere le primitive in italiano

Grazie all'efficace tutorial in italiano di Andrea Molteni sarà un'operazione alquanto semplice dare con NQC quello che si faceva con il Logo: convertire le primitive originali in comandi in lingua italiana. Come accennavo, la microrobotica si presenta quale un "ritorno al passato" per molti aspetti.

²⁵ http://sourceforge.net/project/showfiles.php?group_id=68600bricxcc_setup_3377.exe

²⁶ <http://homepage.mac.com/rbate/MacNQC/>

²⁷ <http://legousb.sourceforge.net/>

Premesso che ognuno può convertire come preferisce le primitive, assegnando nomi specifici ai comandi macro, propongo qui un primo set di elementi che sono sufficienti a gestire le attività proposte nel prossimo capitolo.

```
#define destra(t)      OnFwd(OUT_A); OnRev(OUT_C); Wait(t);
#define sinistra(u)   OnFwd(OUT_C); OnRev(OUT_A); Wait(u);
#define avanti(v)     OnFwd(OUT_A+OUT_C); Wait(v);
#define indietro(w)   OnRev(OUT_A+OUT_C); Wait(w);
#define avantisempre OnFwd(OUT_A+OUT_C);
#define fermatutto    Off(OUT_A+OUT_C);
#define faisempre     while(true);
#define ripeti(z)     repeat (z)
#define procedura     task main()
```

Grazie a queste macro poste in testa al foglio, programmare il microrobot a percorrere una traiettoria quadrata diventa un semplice esercizio per alunni che già hanno fatto esperienza con la tarta nei micromondi, o in altri contesti Logo. La figura 3 mostra le due procedure NQC, senza e con l'impiego delle macro:

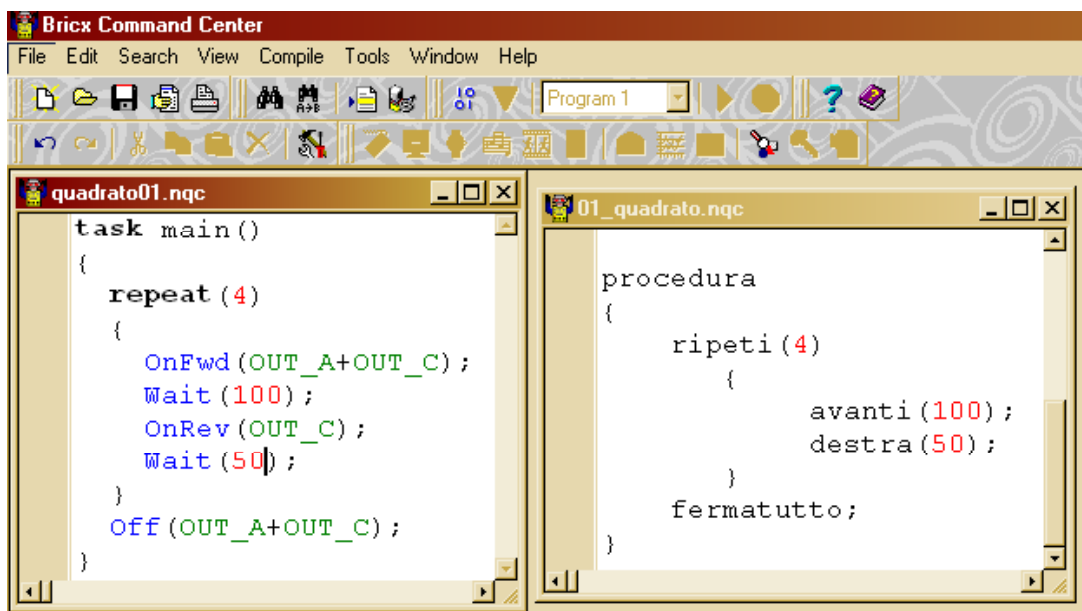


Figura 3 Un percorso quadrato con NQC e, a destra, con l'impiego delle macro.

3) I teoremi microrobotici

b) Il primo teorema robotico - velocità e precisione

A otto anni non è ancora chiaro - a molti esperti di didattica - se un bambino è in grado di affrontare il mondo della microrobotica. Con le insegnanti che hanno provato in classe questo approccio siamo convinti di sì. Sempre a condizione che tutto non parte e si esaurisce nella microrobotica, ovvero se questo contesto operativo si colloca in continuità / prospettiva con altre attività didattiche centrate sulle TIC.

In tal senso questa proposta in forma di teoremi si pone al fianco di quella su "I tre teoremi della tartaruga", che tratta di un approccio più ampio al linguaggio LOGO, teso a superare la banale applicazione alla programmazione di figure geometriche.

I bambini che hanno seguito le esperienze qui proposte avevano avuto modo di operare nell'ambito logico-matematico con il LOGO e la manipolazione di micromondi. Parallelamente alle esperienze di microrobotica procedono ancora nel padroneggiare il linguaggio LOGO e la programmazione di micromondi da loro stessi ideati e realizzati.

I problemi loro posti dai "Teoremi di microrobotica" sono sostanzialmente semplici, ma servono ad acquisire e consolidare i primi elementi del software di programmazione del robot, e le funzioni logico - concettuali connesse. Il raccordo dell'esperienza vissuta nella programmazione della tartaruga, prima di procedere a programmare il microrobot, è un raccordo forte.

Anche se non è stato possibile proporre il LOGO come linguaggio di programmazione del microrobot, abbiamo riscontrato che il "salto" da un linguaggio ad un altro è stato meno complesso di quanto ci immaginassimo.

Il primo teorema prevede che il microrobot debba percorrere un tracciato definito . Il percorso sarà necessariamente semplice, con angoli a 90°. Disponendo di più microrobot e dei computer per programmarli, si può proporre un lavoro a gruppi verificando che la programmazione lo guidi a realizzare un percorso poligonale perfettamente chiuso, in cui quindi alla fine ritorni alla perfetta posizione di partenza. L'obiettivo diventa quindi calibrare il programma per la perfetta chiusura del percorso, e rappresenta un forte stimolo all'attività di ricerca dell'ottimale programmazione. Con ciò acquisendo sempre maggiore confidenza con il linguaggio di programmazione e i diversi fattori in gioco nel movimento del robot.

Il primo ostacolo da superare, che mostra subito la forte differenza tra reale e virtuale, sta proprio nell'applicazione di misure. Se sullo schermo del computer le tarta si muovono per passi (pixel) con massima precisione, e ruotano in gradi senza rischio di errori, un microrobot che si muove nello spazio - tempo non può impiegare le stesse unità di misura!

L'unico modo per governare nello spazio il microrobot è possibile controllando la velocità dei suoi motori e il tempo per cui questi debbono essere attivi. Lasciando al valore di default la velocità dei motori, non resta che determinare quanto tempo serve per percorrere una certa distanza in linea retta (quindi con entrambi i motori accesi in avanti). Per ruotare di 90° dobbiamo trovare il tempo esatto necessario ²⁸, dopo aver attivato i due motori uno in avanti, l'altro in retromarcia ²⁹.

Un compito apparentemente semplice, ma che impegna alunni e insegnante in una forte immersione nella microrobotica, e nella sistematica applicazione del metodo euristico. E non si può copiare: peso, tipo di ruote, caratteristiche del pavimento sono tutte variabili che entrano in gioco a dare una soluzione diversa ogni volta che si prova.

e) Il secondo teorema robotico - il salto dalla scrivania

Una volta che gli alunni hanno preso confidenza su come far muovere il robot, impiegando in modo efficace i motori, possono iniziare a indagare e provare ad usare i sensori. Programmare quindi un robot in grado di "sentire" e conoscere il contesto in cui si sta muovendo.

²⁸ In NQC si esprime il tempo in centesimi di secondo. In altri linguaggi in decimi.

²⁹ Si veda la figura 3 con i listati pronti. In rosso i valori di tempo in centesimi di secondo.

Inizialmente si può utilizzare il sensore di contatto, un microinterruttore che ha due soli valori: 1 e 0, a secondo se il pulsante è premuto oppure no. I linguaggi di programmazione prevedono appositi comandi per interrogare in modo continuo il sensore, e quindi gli alunni potranno impegnarsi sull'impostare cosa fare nell'uno o nell'altro caso.

Programmare un robot che non si blocchi quando urta un ostacolo, ma sappia manovrare per aggirarlo, è una prima semplice attività propedeutica alla più impegnativa proposta del secondo teorema. Dall'esperienza con sensore di contatto si passa subito a quella col sensore di luce. In questo caso sarà utile tracciare a terra - con del nastro adesivo nero - una sorta di binario che il robot può seguire grazie al sensore che lo informa del valore di luce riflessa dal pavimento. Con pochi interventi sul programma il binario da seguire può divenire invece una barriera che respinge il robot. Ma queste esperienze propedeutiche servono solo per poter infine affrontare il problema del secondo teorema, una proposta che coinvolge - mi hanno evidenziato alcune insegnanti - anche emotivamente gli alunni, ma che proprio per questo motiva e rinforza in modo significativo l'apprendimento. Si fa leva sull'identificazione che - più di quanto avvenga con la tartaruga del LOGO - gli alunni hanno

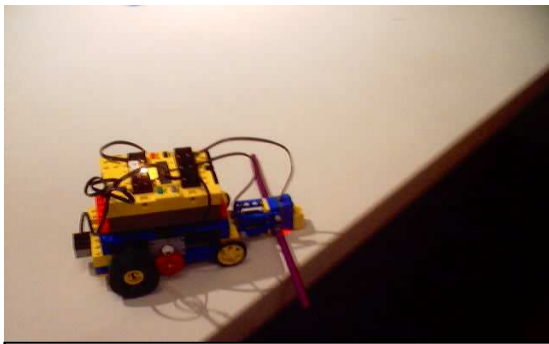


Figura 4 il precipizio è buio, quindi ...

verso il robottino costruito, che potenzialmente può fare molte cose, ma in questo caso deve essere istruito ad evitare di sbagliare e quindi di cadere.

La proposta è apparentemente semplice: data una scrivania, di una certa ampiezza, fare in modo che il robot vi si muova a caso³⁰ evitando di cadere giù.

Le strategie per fare ciò possono essere molte: basate sul sensore di contatto (che verifica la presenza del piano davanti al robot) o su quello ottico (legge il valore di luce riflessa dal piano, che varierà in presenza del vuoto). Se detto questo pensate sia un compito in fondo semplice, vi avvertiamo subito che così non è! Basti citare il fatto che l'avvicinamento al "precipizio" può essere con diversi

```
#define THRESHOLD 45

task main()
{
  SetSensor(SENSOR_1, SENSOR_LIGHT);
  SetPower(OUT_A+OUT_C, 2);
  OnFwd(OUT_A+OUT_C);
  Off(OUT_B);
  while (true)
  {
    if (SENSOR_1 < 45)
    {
      OnRev(OUT_C+OUT_A);
      Wait(100);
      Off(OUT_B+OUT_C);
      SetPower(OUT_A+OUT_B, 7);
      OnRev(OUT_A);
      OnFwd(OUT_C);
      Wait(100);
      until (SENSOR_1 >= 45);
      SetPower(OUT_A+OUT_C, 2);
      OnFwd(OUT_A+OUT_C);
    }
  }
}
```

³⁰ Qui il richiamo al secondo teorema della tartaruga raggiungere un risultato finale nonostante la c

angoli d'approccio per farvi intuire come si debba attentamente pensare alla posizione dei sensori, e sviluppare un programma ben congegnato, per avere buone probabilità di successo. Ovviamente a questo si giunge per successivi affinamenti e modifiche.

Il confronto tra diversi gruppi di ragazzi messi alla prova su questo fronte può trovare realizzazione in una prova a tempo: dato un tempo massimo (anche solo un minuto è un tempo lunghissimo in questo caso), per quanto il robot da loro programmato riuscirà a percorrere a caso la scrivania senza cadere? Ovviamente la velocità del robot sarà predeterminata e uguale per tutti.

Potendo svolgere due o meglio tre sessioni di gioco si attenuano i fattori di casualità, potendo quindi differenziare tra le diverse costruzioni e programmazioni fatte dagli alunni quella con più alto tasso di successo.

Ho vissuto in prima persona un salto dalla scrivania. La routine sembrava funzionare benissimo, e la mia fiducia del fatto che il robottino potesse tranquillamente girovagare era totale. Sino a che un insieme di fattori - angolo di approccio al bordo, slittamento delle ruote sulla superficie, tempi di reazione del sensore - non ha ingannato il programma, provocando un salto di 75 centimetri sulla dura pietra del pavimento! Robot a pezzi, ma una volta rimontato nuovamente pronto a ripartire, non senza però un aggiornamento del programma!

Raccomandiamo quindi a tutti di dare per quasi certo il salto. Riuscire a raggiungere 5 minuti di vagabondaggio sulla scrivania è indice di grande perizia. Ma arriva per gradi, dopo qualche salto! Quella in figura 5 è una procedura per un microrobot dotato di sensore di luce. Ovviamente a solo titolo di riferimento. Come nel caso del percorso quadrato, non si può copiare, ma solo ispirarsi.

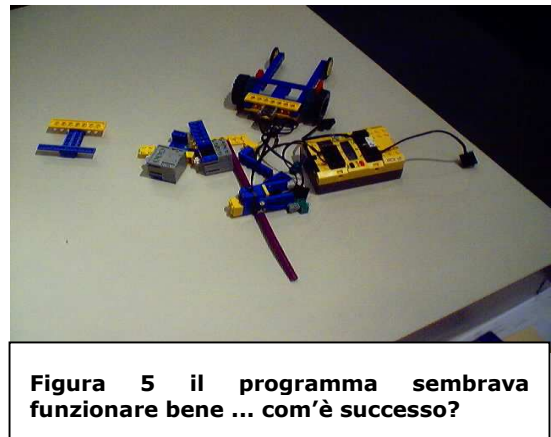


Figura 5 il programma sembrava funzionare bene ... com'è successo?

f) Il terzo teorema robotico - il solutore di labirinti

Come detto in premessa qui si inizia a programmare il microrobot in modo che sia in grado di risolvere un problema - quello del labirinto - vecchio come il mondo ma sempre affascinante. Il fatto che un certo Skinner li usasse per studiare l'apprendimento nelle cavie di laboratorio non è un caso. Ma qui - attraverso la sfida robotica - metteremo i nostri alunni a caccia della migliore strategia di soluzione da riportare nel programma per il microrobot. A lui poi dimostrare quanto la strategia scelta dai ragazzi sia efficace!

Per rendere il problema alla portata degli alunni, sarà bene definire alcuni elementi de labirinto: punto di partenza, punto di arrivo, tipologia degli ostacoli. Ad esempio il sapere che l'uscita del labirinto sarà sempre sul lato opposto a quello di partenza, semplificherà di molto il problema. Sapere che gli ostacoli saranno costituiti da segmenti orizzontali, trasversali alla direzione verso l'uscita, sarà un'altra semplificazione. Lo si afferra bene osservando le seguenti figure: a sinistra quella relativa ad un labirinto semplice, secondo le regole prima indicate, a destra invece un labirinto tradizionale, decisamente più complesso anche se non



impossibile. Fatta esperienza col primo semplice labirinto, si potranno proporre situazioni più complesse, sino anche ad arrivare ad un vero labirinto. Ma questa sarà una valutazione che sarete in grado di operare assieme agli alunni.

Il ricorso a labirinti, o altri problemi da far risolvere a microrobot è al centro di tante gare microrobotiche. Troverete tra i materiali lo studio di caso relativo all'esperienza milanese promossa dal Museo della Scienza e della Tecnica nell'ambito dell'iniziativa "Scienza Under 18"³¹, a cui partecipano squadre miste di alunni di scuole elementari e medie milanesi.

³¹ <http://www.scienza-under-18.org/>

BIBLIOGRAFIA

Ceserani G.P., *Gli automi, storia e mito*, Laterza, 1983

Chiocciariello A., Manca S., Sarti L., *La fabbrica dei robot - Bambini che costruiscono e programmano con un ambiente visivo*, TD 27, Genova, 3-2002, pp. 56-67

Lariccia G., *Le radici dell'informatica*, Sansoni, 1988

Marcianò G., *Robotica a scuola*, Rassegna dell'Istruzione, Le Monnier, a. 2003/04, 4, pp. 6-20

Papert S., *LOGO philosophy and implementation*, LCSJ, 1999

Varisco B. M., *Nuove tecnologie per l'apprendimento*, Garamond, 1998.

SITOGRAFIA

Hardware e kit robotici:

i) Basati su Lego RCX:

- <http://www.lego.com/eng/education/mindstorms/>

ii) Basati su M.I.T. Cricket:

- <http://llk.media.mit.edu/projects/cricket/about/index.shtml>
- <http://www.handyboard.com/>

Software e linguaggi robotici

Esperienze straniere

- CNICE - Osservatorio tecnologico del Ministero dell'Istruzione e della Scienza spagnolo – Gli articoli sulla robotica sono i più letti, e come tali accessibili dall'home page <http://observatorio.cnice.mecd.es> – per accedere direttamente: <http://observatorio.cnice.mecd.es/modules.php?op=modload&name=News&file=article&sid=117&mode=thread&order=0&thold=0>
- Stokane National School – Irlanda
<http://homepage.eircom.net/~stokanens/#projects>
- Università di Alicante – Spagna – Progetto TEDDi – robotica
<http://www.teddi.ua.es/lineasTrabajo/robotica.asp>

Esperienze italiane

- ITD – CNR di Genova <http://www.cnr.it/istituti/Focusperistituto.html?cds=102>
- Museo della Scienza e della Tecnica "Leonardo da Vinci" di Milano
<http://www.museoscienza.it/microrobotica/default.htm>