APPRENDIMENTO E TECNOLOGIE III[1]

PROGRAMMARE PER APPRENDERE: I MICROMONDI

di Laura Casulli

approfondimenti di Luca Navone sui programmi logo *freeware*

CHE COSA SONO?

I Micromondi possono essere considerati i figli multimediali della prima attività informatica introdotta nella scuola elementare dalla fine degli anni Settanta: il linguaggio di programmazione Logo, ideato da S. Papert e B. Silverman del MIT[2] di Boston.

La parte più caratteristica e conosciuta della programmazione in Logo consiste nella "grafica della tartaruga": Papert costruì inizialmente un automa cibernetico "da pavimento" collegato al computer, la cui forma ricordava davvero una specie di tartaruga. Dando una serie di comandi al computer, essa si spostava sul piano in cui giaceva e, mediante una penna a inchiostro, lasciava una traccia del suo passaggio. Con una serie di comandi appropriati era possibile eseguire dei disegni.

Per motivi di praticità e di economia si preferì poi sostituire tale robot con una sua rappresentazione sullo schermo, anche se in alcune attuali versioni di Logo la forma di tartaruga è stata sostituita da un triangolino isoscele, che continua però a essere chiamato tartaruga. Il principio di base è comunque lo stesso: per disegnare sullo schermo si

[1] Questo articolo, insieme a quelli precedenti pubblicati nella stessa rubrica sul sito, costituisce un commento e una sintesi problematizzata e integrata del testo di B. M. Varisco e V. Grion, "Apprendimento e tecnologie nella scuola di base", Utet, Torino, 2000.
[2] Cfr. http://www.mit.edu.

devono impartire alcuni comandi tenendo presente che la tartaruga, come entità geometrica, è caratterizzata dalla posizione nel piano e dall'orientamento.

La creazione di ambienti di apprendimento denominati Micromondi[3], che ha recentemente risvegliato l'interesse della scuola per gli aspetti didattici legati alla programmazione, risponde all'esigenza di coniugare i punti di forza delle più recenti acquisizioni relative alle tecnologie per la didattica, alla diffusione della videoscrittura, degli ipertesti e della multimedialità, con quel memorabile Logo che entrò nella scuole con i *Commodore* 64 .

Ma che cosa sono questi Micromondi?

Il termine "Micromondi", che venne utilizzato per la prima volta proprio da Papert, è applicato a una riproduzione, effettuata su computer, del comportamento di un sistema reale (che può essere un sistema economico, sociale, scientifico, ecc.). Un Micromondo è quindi, in definitiva, una simulazione su computer basata sullo studio di un caso reale. In un Micromondo si possono esplorare molteplici alternative, effettuare la verifica di ipotesi e intuizioni e sperimentare quali azioni possono avere maggiori probabilità di riuscita nella risoluzione di un problema. I Micromondi sono pertanto piccoli universi, pezzi di realtà, luoghi sicuri da esplorare in cui è possibile creare applicazioni interattive, sotto forma di storie animate, quiz, percorsi didattici, simulazioni geometriche e molto altro ancora.

Nei programmi Logo attuali (Micromondi Logo, MSWLogo, IperLogo, etc.) si possono gestire un numero illimitato di "tartarughe" (oggetti programmabili) e più procedure parallele anche complesse, nonché creare pulsanti per collegamenti ipermediali o per eseguire istruzioni assegnate dall'utente.

Una delle versioni del linguaggio di programmazione Logo, largamente utilizzata nelle scuole, è sicuramente "Micromondi Logo" della Garamond[4]: si tratta, in effetti, di uno strumento davvero potente e completo, studiato appositamente per consentire la libera creazione di micromondi. L'interfaccia si presenta semplice e vivace, ricca di funzioni che il bambino può liberamente esplorare: disegni a mano libera, creazione di brani musicali, inserimento di testi, immagini, suoni e filmati video, oltre ai comandi di ridimensionamento e trasformazione delle "tartarughe" in innumerevoli figure e forme colorate che, disposte in sequenza, creano effetti di animazione. La versione "MicroMondi Pro" può inoltre includere anche *hyperlink* a pagine Web e permette l'inserimento di fogli di calcolo *Excel*. Inoltre, il notevole valore aggiunto dei progetti MicroMondi di tipo interattivo è che possono essere

[3] Per le diverse versioni di Logo e Micromondi cfr. gli ultimi paragrafi del presente articolo.
[4] Cfr. http://www.garamond.it/index.php?risorsa=micromondi.

pubblicati su Internet per essere utilizzati e condivisi da qualsiasi utente, attraverso un *player* scaricabile gratuitamente.

La significatività dei risvolti cognitivi e pedagogici del giocare con la programmazione a scuola, attraverso Micromondi e in generale la microrobotica, è stata inoltre ribadita recentemente dalle scelte della Direzione Regionale del Piemonte proprio nel progetto "*Dschola*", che prevede tra le aree di interesse e di formazione ai docenti un Centro di Animazione[5], con funzione di riferimento sul territorio piemontese, che contribuisce allo sviluppo di metodi, sistemi e contenuti innovativi relativamente ai temi e agli ambienti digitali della tradizione costruzionista.

PERCHÉ I MICROMONDI?

La filosofia pedagogica del gruppo di ricercatori del MIT, coordinato da S. Papert, dalla quale sono scaturite le premesse cognitive fondamentali dei molteplici ambienti per l'apprendimento, che si possono rappresentare con il simbolo della Tartaruga geometrica (*Micromondi*, *Logo*, *LegoLogo*, *LegoWriter* ecc.), è il cosiddetto "costruzionismo". Vale la pena di soffermarsi sulla fitta rete di collegamenti, sovrapposizioni e discontinuità che esso presenta, in relazione alle principali visioni dell'uomo e della conoscenza che si sono sviluppate e affermate nel Novecento.

Il costruzionismo, che si pone in continuità con molti presupposti del costruttivismo piagetiano (*learning by making)*, va tuttavia oltre l'idea di una conoscenza intesa come costruzione progressiva di strutture, ponendo invece l'accento sull'importanza di un *setting* che favorisca l'apprendimento consapevole attraverso una costruzione collettiva, pubblica, condivisa. Esso condivide anche con l'attivismo di Dewey il classico "*learning by doing"*, ma aggiunge "pensando e parlando intorno a quello che si fa"[6].

La programmazione diventa in questo senso uno strumento per l'apprendimento significativo, "sintonico", centrato sull'allievo e non sull'insegnante, un modo "potente" per utilizzare il computer in classe, differente da quello classico comportamentista che lo considerava una sorta di *tutor* meccanico, con scarse o nulle possibilità di interazione con l'allievo. Si tratta, come per il costruttivismo, del riconoscimento della centralità educativa affidata alla "continuità" tra la conoscenza e l'attualizzazione di un progetto, secondo il paradigma galileiano classico di circolarità tra Ipotesi e Verifica. Tutto ciò tenendo conto

[5] http://www.bdp.it/~toee0014/.
[6] I. Harel, S. Papert, *Software Design as a Learning Environment*, in I. Harel, S. Papert (a cura di) *Constructionism*, Ablex, Norwood, 1991.

dell'importanza della "risonanza culturale" all'interno delle realtà scolastiche, ovvero del legame che si deve instaurare tra progetti che prevedono l'uso delle tecnologie informatiche e la programmazione curricolare.

Si tratta di un'ottica in antesignana polemica con alcune caratteristiche della recentissima introduzione dell'Informatica nelle scuole italiane, che pur viene sbandierata come segno di svecchiamento e avvicinamento a prospettive didattiche europee di più ampio respiro. Infatti l'utilizzazione didattica dell'informatica pensata negli anni Settanta da Papert si schierava già, e in modo esplicito, contro il tecnocentrismo, che iniziò a intravedersi dopo poco nelle utilizzazioni scolastiche del Logo, e che contraddistingue le attuali "riforme" che stanno investendo tutti i livelli del sistema formativo italiano. Nei primi anni Novanta, infine, Papert si sarebbe espresso con amarezza, commentando la mancata rivoluzione pedagogica: "Così, a poco a poco, le caratteristiche 'sovversive 'del computer vennero scalzate; invece di superare le barriere tra le materie, e di mettere in discussione il concetto stesso di separazione, il computer finì col diventare una nuova materia, invece di spostare l'accento da un programma impersonale alla ricerca attiva e entusiasta degli studenti, il computer servì a rafforzare i tradizionali metodi scolastici…"[7].

L'informatica come disciplina oggi è entrata a pieno titolo anche nei curricoli universitari, con il fine della cosiddetta alfabetizzazione informatica. Purtroppo viene coniugata in termini di in mere competenze tecniche (e ciò anche negli indirizzi di tipo filosofico, di Scienze Sociali e della Comunicazione), mentre viene disconosciuta la centralità della problematizzazione e contestualizzazione di queste in un orizzonte culturale più ampio, dei risvolti filosofici, cognitivi, pedagogici. Siamo pertanto di fronte ad uno scenario ben più allarmante di quello condannato da Papert, in quanto le tendenze al tecnocentrismo passivizzante non sono più solo l'esito di gruppi di insegnanti e educatori ma hanno preso forme istituzionali, radicandosi nella storia**.** Gli ambienti per l'apprendimento, i "micromondi" generati da tali politiche educative, sono punti, in ordine sparso ma non (àinoi) casuale, di un Macromondo la cui *Console* dei comandi di "base" rischia di non essere più alla portata di individui, gruppi, parti politiche, a cui restano forse ancora le modifiche "accessorie".

È dunque fondamentale declinare i principi dell'apprendimento costruzionista, che trovano riscontro nella filosofia del Logo[8], anche per non perderli di vista nella pratica

[7] S. Papert (1993), *I bambini e il computer*, Rizzoli, Milano, 1994.
[8] B.M. Varisco (a cura di), *Nuove Tecnologie per l'apprendimento. Guida all'uso del computer per insegnanti e formatori*, Garamont, Roma, 1998.

didattica dato che, come tutto, possono cambiare di segno e colore a seconda dello scenario nel quale operano:

- il protagonismo dello studente-programmatore;
- l'inversione epistemologica - che sostituisce "imparare per usare" con "usare per imparare" -, con la conseguente rivalutazione del pensiero operatorio concreto su quello formale logico-deduttivo di Piaget. Sono le esperienze concrete e casuali che, secondo Papert, hanno originato anche le teorizzazioni di tipo scientifico;
- l'apprendimento "sintonico" coinvolge il soggetto a livello corporeo nell'apprendimento "corpo-sintonico", che nasce dall'armonizzazione dell'orientamento spaziale del proprio corpo con quello della tartaruga, e lo indirizza anche relativamente all'identità personale nell'apprendimento "ego-sintonico", che deriva dall'attitudine al "decentramento cognitivo" - secondo la definizione di Piaget -, e soprattutto dall'acquisizione di visioni pluriprospettiche sullo stesso problema o dominio[9].
- i micromondi sono "palestre cognitive", nelle quali si ricercano incessantemente problemi e soluzioni emergenti (*problem finding/ problem solving*);
- l'epistemologia prevalente è quella "dell'indeterminatezza gestita", che favorisce l'esperienza metacognitiva descrivibile come "avere vagamente ragione" *vs* quella dello "sbagliare con precisione". Si tratta di un procedere analogico, per aggiustamenti continui, a discapito della logica dicotomica digitale del vero o falso;
- tale modo di procedere si configura come una apologia dell'errore, definita "matetica" e descrivibile come "apprendere riflettendo sui propri errori".

La conoscenza non è dunque un insieme di regole formali, in cui gli insegnanti si muovono in qualità di tecnici, e l'apertura verso metodi diversi costituisce il cardine didattico. Le tecnologie informatiche sono considerate come attrezzi sociali (*convivial tools*)[10], ambienti che aiutano a pensare, esprimersi, comunicare nel contesto sociale.

---

[9] La geometria della tartaruga non è euclidea, non possiede una struttura assiomatico-deduttiva che, a partire dagli assiomi e dai postulati, deduce tutti i teoremi secondo una concatenazione logica rigorosa. Euclide aveva posto infatti la sua l'attenzione sugli oggetti (piano, rette, punti, triangoli...), viceversa l'attenzione è qui posta sulle trasformazioni geometriche che spostano o modificano tali oggetti nel piano e nello spazio; cfr. *http://www.emsf.rai.it/interviste/interviste.asp?d=335; http://www.math.ethz.ch/~michele/AZ/it/hyp.html; http://www.olivetocitraic.it/Formanet/BUCCINO/storia%20geometrie.htm*.

[10] I. Illich, Tools for Conviviality, Harper & Row, New York, 1973.

Un corollario di tale "pluralismo cognitivo" è costituito dall'approfondimento degli stili cognitivi, come emergono dall'osservazione dell'attività programmatoria. La tipologizzazione che è emersa propone alcuni stili-modello differenti, lo studio dei quali ne evidenzia anche i collegamenti con il genere, le culture etniche, i tipi di personalità.

Una prima dicotomia è quella che contrappone lo stile *hard*, pianificatore, che tende a programmare seguendo alcune linee procedurali fisse, a quello *soft*, che negozia le proprie procedure in modo contestuale, con il lavoro stesso, come un pittore o un *bricoleur*[11].

La fortuna delle interfacce iconiche è infatti stimolo e risultato della crescita di questa ultima modalità di navigare la conoscenza sotto l'impulso delle tecnologie, e anche l'attuale "programmazione ad oggetti" (OOP)[12] degli ambienti ipermediali si muove in tale direzione: sostituisce metafore algebriche con metafore biologiche e sociali e considera il programma come insieme di elementi (detti oggetti) incorporanti sia i dati che le procedure o funzioni che operano su di essi. Gli oggetti interagiscono così tra di loro in un ordine gerarchico nel quale gruppi di livello inferiore possono ereditare caratteristiche da quelli di livello superiore. Essi sono considerati non di per sé, per le loro proprietà, ma per le loro funzioni e interazioni di sistema. La programmazione orientata ad oggetti permette dunque di costruire sistemi flessibili ed economici, con sviluppo e aggiornamento più veloce, più affidabili e dotati di miglior interfaccia utente.

Per Papert e il suo gruppo di ricercatori il ragionamento formale, *hard*, non è dunque l'apice dello sviluppo come per Piaget, non è il paradigma dell'intelligenza scientifica, logica, matematica, dell'oggettività scaturita dal distacco con l'oggetto di studio. Esso viene piuttosto considerato uno *stile*, sullo stesso piano dello stile operatorio concreto. Il programmatore *soft* non agisce solo sugli oggetti ma sui loro rapporti, pensa piuttosto in termini di relazioni che di proprietà, pensa attraverso i problemi (*to think from with problems*), ha una propensione al ragionamento abduttivo, ecologico, sistemico. L'intelligenza stessa non è ridotta dunque a quella del modello cognitivista HIP (*Human Information Processig*), che la considera quale capacità di elaborare informazioni, ma si configura come un'intelligenza emergente, come una società della mente[13]. Che

[11] L'idea della scienza come *bricolage* ha una notevole tradizione di riferimento: tra gli altri C. Lévi-Strauss, *The Savage Mind*, University of Chicago Press, Chicago, 1968; P. Feyerabend, *Against Method: The Outline of an Anarchistb Theory of Knowledge*, NLB, Londra, 1975; L. Wittgenstein (1953), *Ricerche filosofiche*, Einaudi, Torino, 1967.
[12] http://www.macocoa.omitech.it/download/MaCocoa.PDF.
[13] Cfr. M. Minski, http://helios.unive.it/~cired/crossing/somente.htm, http://lgxserver.uniba.it/lei/rassegna/990822d.htm.

l'accesso alla verità non sia da pensare solo come "momento cartesiano" in termini di conoscenza, ma che sia necessario il modificarsi, trasformarsi, nel corso della propria esperienza di soggetto, dando forma a se stesso e al mondo artigianalmente, è peraltro uno dei temi suggeriti e sviscerati dalla filosofia contemporanea[14], dalle scienze sociali[15] e dalla scienza stessa, quella con la S maiuscola[16].

In sintesi, le teorie di Piaget e Papert condividono il costruttivismo di fondo e anche la prospettiva evoluzionista, l'attenzione alla genesi dello sviluppo cognitivo e la concezione dell'intelligenza come equilibrio tra stabilità e cambiamento. Ma l'attenzione di Papert non è alla stabilità quanto alle dinamiche del cambiamento, agli stili d'apprendimento individuali e culturalmente connotati: essere intelligente vuole dire essere un *soft master*, situato, connesso, sensibile, immerso. Il bambino epistemico di Piaget si perfeziona, invece, nell'abilità di manipolare mentalmente oggetti simbolici in mondi ipotetici attraverso fasi di sviluppo generali e decontestualizzate.

L'osservazione compiuta dal gruppo di Papert ha portato anche a formulare connessioni di genere, sulla preponderanza dello stile *hard* nel genere maschile e di quello *soft* presso il genere femminile. In chiave psicoanalitica si può considerare, tra le varie possibili spiegazioni, che i bambini vengono abituati a giocattoli da manovrare, controllare, gestire con distacco, mentre le bambine crescono con bambole da allevare; tali distinzioni, in epoca di "Tamagogi", risultano tuttavia riduttive quanto mai.

In una visione sociologica si può altresì rilevare che i modelli di comportamento maschile dominanti sono la forza, il potere, la decisionalità, mentre quelli femminili piuttosto la negoziazione e l'adattamento. Si può quindi provocatoriamente sostenere che la diffusione delle tecnologie educative che tendono a implementare il pluralismo

---

[14] Oltre al filone ermeneutico classico, cfr. M. Foucault, *L'ermeneutica del soggetto*, Feltrinelli.

[15] Cfr. diffusione di metodi qualitativi e narrativi *vs* quantitativi.

[16] Warner Heisemberg (1901-1976) nel 1925, è all'origine della meccanica quantistica. Con questa teoria, Heisemberg indicava una nuova meccanica, una modificazione di quella classica, che interpretava la struttura della materia in termini di "quanto", ovvero d'unità indivisibili, d'atomi d'energia. Mentre la meccanica classica postulava che un oggetto fisico dovesse in ogni istante possedere una determinata posizione e una ben determinata velocità, la meccanica quantistica perviene, con la formulazione del principio d'indeterminazione di Heisemberg, all'affermazione dell'impossibilità di una misurazione contemporanea della posizione e della velocità di una particella. Il principio d'indeterminazione evidenzia dunque che in microfisica i fenomeni non possono essere studiati a prescindere dagli effetti, dalle azioni di disturbo provocati dall'osservatore, che va pertanto assunto come parte integrante del fenomeno. Una delle formulazioni del principio classico di causalità fisica, afferma che è sempre possibile, data la conoscenza precisa dello stato presente di un sistema isolato, prevedere esattamente il suo stato futuro. Per Heisemberg, invece, l'interazione tra oggetto e apparato di osservazione genera un comportamento apparentemente indeterministico degli oggetti microscopici; sarebbe però *insensato* porsi la questione di come si comportino questi oggetti quando nessuno li osserva, quando nessuno strumento li disturbi, e chiedersi se in realtà il loro comportamento è di tipo deterministico

metodologico possa condurre a colmare il divario di stile tra i generi, a superare la dicotomia tra pensiero astratto e pensiero concreto, in favore di un modo di concepire la mente che trae spunto e origine dalla biologia e non dalla logica.

Il costruzionismo condivide dunque molte delle proposte del socio-costruttivismo nella sua forma culturale-situazionista[17], anche nello spazio accordato agli stili individuali di collaborazione tra pari e nei contesti di apprendistato cognitivo, dove l'esperto modella e conduce l'esperienza. Viene riconosciuta l'importanza della "collaborazione attraverso l'aria", ovvero del libero flusso di idee, della creazione di una zona confortevole in cui esplorare e sperimentare le proprie intuizioni, i progetti embrionali, nel rispetto delle proprie necessità di interazione e delle diverse attitudini al lavoro di gruppo.

MICROMONDI E MACROMONDI

Vi invito a entrare in un Micromondo, esplorarlo, auscultare e riflettere sulla vostra attitudine a costruire e programmare come ci suggeriscono le opzioni ambientali di cui è

oppure no, in quanto è evidente che lo scienziato non ha nulla da dire circa quello che fa la natura allorquando nessuno la osserva.

[17] Cfr. L. Casulli, *Apprendimento e tecnologie*, nella stessa rubrica "Incroci" sul sito.

dotato il programma “Micromondi Logo”[18], ovvero le tre *Console:* la *Console* dei Comandi, la *Console* delle Forme e quella di Disegno.

L’oggetto su cui si agisce nasce con la forma di una tartarughina (nera in figura) che può essere mossa secondo istruzioni semplici e predeterminate (avanti, destra, sinistra, indietro) o alla quale si possono insegnare nuovi comandi partendo da queste. Il tutto può essere accompagnato da effetti multimediali (musica, video, immagini, testi, suoni), gratificanti per il soggetto-creatore anche grazie all’evidente fruibilità esterna.

Esprimo alcuni miei dubbi, di carattere cognitivo ma soprattutto filosofico, che non intendono tuttavia costituire un’obiezione teorica strutturata, quanto piuttosto spunti di riflessione sparsi in modo *soft*, che possono essere raccolti e rilanciati.

Mi immergo nella realizzazione di uno scenario fantastico: creo un oggetto-tartarughina e poi scelgo quale altra forma darle, tra quelle previste o disegnandola direttamente; opto per l’albero, lievemente inclinato a destra come da disegno proposto (vedi figura), ovvero con una inclinazione di partenza differente da quella della tartarughina, che in stato di quiete assume invece sempre la posizione testa-in-alto parallela all’ asse delle y (vedi figura); decido come programmare i suoi spostamenti. In tutto ciò, tengo presente che, in realtà, l’albero è una tartarughina camuffata, sotto mentite spoglie, che come lei si comporta e si posiziona nello spazio: altro che apprendimento corpo-sintonico!

La sensazione di disagio cresce e prende forma : ma l’albero non è una tartaruga!

O, se devo ammettere che la celi, allora la tartaruga, che è quella che io sto in realtà comandando, è l’anima dell’albero!

Ma se abituo, o continuo ad abituare i bambini, a pensare che gli enti abbiano un’essenza, un’anima, un’idea platonica aristotelicamente incorporata, mi pongo in un’ottica innegabilmente sostanzialista. E’ pur vero che ciò che determina in questo caso l’oggetto non sono solo proprietà in sé (nome, dimensione, colore, forma), ma anche le sue funzioni, ovvero le istruzioni secondo le quali si muove la tartaruga, il suo specifico rapporto con l’ambiente circostante: dunque, una delle proprietà sostanziali dell’oggetto è proprio il suo essere programmabile secondo alcuni parametri, il suo poter entrare in relazione con l’ambiente, ma sempre secondo le medesime modalità, quelle della tartaruga.

[18] Mi riferisco alla versione della *Garamond* in quanto maggiormente diffusa nelle scuole.

Resta infatti lo spaesamento di fondo, peraltro di solito aggirato in quanto la *Garamond* nelle istruzioni all'utilizzo specifica che - per motivi certamente di ordine tecnico, e non credo per evitare spaesamenti collaterali -, conviene usare la tartaruga mentre si sta preparando il programma e, poi, trasformare tutto nelle forme finali desiderate. La forma dell'oggetto, ciò che appare, risulta dunque essere qualcosa che si può cambiare a piacimento in qualunque momento, senza alterare nulla dell'interno. Essa si configura come la più opzionale, la meno importante, come se non fosse sulle forme che si gioca la riuscita della costruzione del Micromondo da parte del bambino. Tra l'altro, la tartaruga è l'unica forma in grado di mostrare, di svelare, la direzione e i cambiamenti di spessore e colore.

Ma se devo pianificare tutto prima di dare la forma all'oggetto, non sono costretto a utilizzare uno stile *top down, hard,* a discapito di quello immersivo, *soft*?

Non mi è chiaro, allora, se la costruzione di Micromondi si collochi in una visione della realtà e della conoscenza vicina alla prospettiva ermeneutico/costruttivista, senza centri gravitazionali, aperta e pluriprospettica o non, piuttosto, in una visione della verità frontale, cartesiana, positivista, che considera il problema scomponibile in parti dal semplice al complesso e finisce per riproporre le dicotomie tradizionali interno/esterno e centro/periferia.

Sembrano questioni eccessivamente complicate e dunque lontane dalla didattica quotidiana, ma sono altresì fondanti tutta la prassi. Porsi questi interrogativi significa mettersi di fronte al senso più ampio di ciò che facciamo in classe con gli studenti, piccoli e grandi, ed essere consapevoli di fornire, proprio attraverso le pratiche, visioni del mondo latenti ma non per questo meno incisive.

La questione da non sottovalutare è che diamo, anche attraverso il lavoro in classe con ambienti digitali, come nel caso di Micromondi, risposte a domande di amplissima portata che, se fossero poste in modo esplicito, non ci sentiremmo di liquidare né negandoci, né riducendone la portata e la complessità, ma soprattutto non disconoscendone la significatività autentica: gli enti (che tra l'altro, nelle opzioni del programma, non sono distinti in "viventi" e "non viventi" ma in "attivi" e "non attivi") hanno un'anima? Gli alberi, le nuvole, i castelli, i fiori, i pesci, le automobili, sotto sotto sono delle "tartarughe"?

Mi piace l'idea che l'utilizzo didattico dell'ambiente Micromondi possa davvero essere ri-valorizzato come "palestra per pensare", tenendo presente questi interrogativi e anche tutte le critiche che vengono mosse sulla possibilità di ri-attualizzare il Logo secondo le intenzioni del suo ideatore[19]. "Pensare" però è un verbo di portata molto ampia e deve essere anch'esso ri-declinato, anche perché è quanto meno ovvio che a scuola si pensi.

Le domanda che ci aiutano a cercare chiarezza si trasformano allora in: pensare a cosa?

Pensare a come programmare, costruire, muovere, far interagire più oggetti secondo logiche coerenti, esplorare, esprimere e scoprire il proprio modo di conoscere, divertendosi, collaborando, ma anche imparare ad ampliare i contesti, a porsi e a porre interrogativi fondanti partendo dagli oggetti dell'esperienza. Perché lasciarsi allora sfuggire l'occasione di "filosofare" con i bambini attraversando ed esplorando i loro micromondi?

Le tecnologie sono strumenti, ma anche occasioni potenti per dare vita nelle scuole non a laboratori di informatica ma a laboratori di un pensiero[20] attento[21], coraggioso, riflessivo, che agisce nei confini e abita gli interstizi dei "manuali per uso didattico", forzando non solo i limiti della dicotomia conoscenza/azione, come nei presupposti delle filosofie pedagogiche attiviste, costruzioniste, costruttiviste, ma anche di quella fra particolare e generale[22].

---

[19] "…Non penso che il Logo possa più essere riproposto, nemmeno attraverso un'operazione di "*lifting*" multimediale come quella di MicroMondi, come "una palestra della mente", come vorrebbe un certo fondamentalismo neocognitivista assai pervasivo in ambito pedagogico; o che si debba necessariamente insegnare ai bambini la logica della programmazione; penso piuttosto che rimanga valida l'idea di offrire ai bambini l'opportunità di dare un'occhiata più da vicino, più attraverso la pratica dello smanettamento che attraverso la logica strutturalmente rigorosa degli algoritmi, a come funzionano i programmi "dal di dentro", per orientarli verso un uso più autonomo e disincantato delle tecnologie informatiche" , L. Rosso, *Effetti digitali*, http://www.bdp.it/~toee0014/progetti/lem/stage/rosso/rosso.htm.

[20] L. Casulli, *Perché un laboratorio di pensiero a scuola?,* gennaio 2002 in http:// http://irre.scuole.piemonte.it/materiali/set_offerta_formativa/piani/lab_pens/casulli/il_laboratorio_di_pensiero.htm.

[21] "...il primo dovere della scuola è quello di sviluppare la facoltà di attenzione ricordando agli studenti incessantemente che devono essere attenti per poter più tardi essere giusti.*"*, S. Weil, *Piccola cara. Lettere alle allieve.*

[22] "L'ora della filosofia è quella delle domande non generali o particolari, ma singolari, che colgono l'evento in quanto tale o le cose in quanto eventi"; su questo e sul divenir-bambino del filosofo, cfr. G. Deleuze, *Mille Piani. Capitalismo e schizofrenia*, (1980), tr. italiana di G. Passerone, Castelvecchi, Roma, 1997, vol. II, 197-183 e G. Deleuze, *Critica e Clinica* (1993), tr. It. Di A. Panaro, Raffaello Cortina Editore, Milano 1997, cap. IX.

ALCUNE VERSIONI DI LOGO E LINK UTILI

MSW LOGO 810k 02-09-94 [Win 3.x]. LOGO *programming lang*. (Windows 3.1, PD) dall'Universita' di Berkeley.

Aggiunte per MswLogo 283k 02-09-94 [Win 3.x]. *Files* aggiuntivi per MSWLogo w/Win3.0.
Sorgenti MSWLogo 355k 02-09-94 [Win 3.x].*Source code for* MSWLogo (Borland C"> ">)

Winlogo v.1 187k 01-24-94 [Win 3.x]. Semplice versione del Logo per *Windows* (v.1 del MswLogo)

MswLogo v.2 606k 10-14-93 [Win 3.x]. 2 versione del Logo MSW di Berkley

micromondo Marta 11k 02-02-94 . Micromondo MARTA (procedure in Logo 'italianizzato') da adattare ai differenti interpreti e provare.

Logo su Internet 1k 04-06-96 dove trovare alcuni interpreti LOGO sotto Internet

LOGO Foundation

Brian Harvey's UCBLogo, European mirror

MSWLogo, European mirror

LOGO related materials presso il MIT include versioni *free* di LOGO

European mirror of the MIT Logo Site

StarLogo, a new Logo version with thousands of turtles

George Mills, autore di MSWLOGO dove è possibile scaricare tutte le versioni

Bardassano, settembre 2003

PROGRAMMI LOGO FREEWARE

di Luca Navone

UCBLOGO

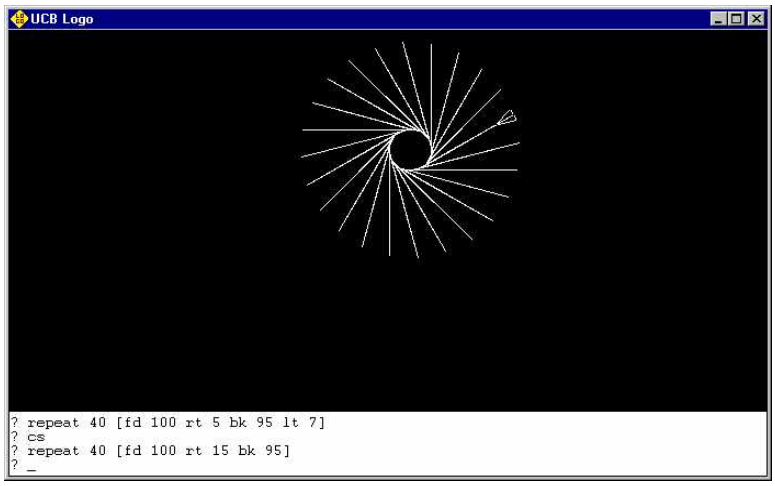
UCBLogo è un programma Logo di impostazione grafica piuttosto vecchia, ma può essere ancora utilizzato con soddisfazione se si è disposti a rinunciare in partenza a prestazioni grafiche molto evolute. Brian Harvey, il programmatore che ha sviluppato UCBLogo, espone subito le sue possibilità e i suoi limiti nel manuale che accompagna il programma.

La caratteristica più importante posseduta da UCBLogo è quella di essere multipiattaforma: esso è infatti disponibile per ambienti Unix (Linux), Dos, Windows e Mac.

Le caratteristiche che non sono implementate nel Logo di Berkeley sono invece la musica, una interfaccia amichevole, animazioni e possibilità di interfacciare dispositivi multimediali.

Sono convinto che possa comunque essere un prodotto utilizzabile. E' vero che MSWLogo (che è stato programmato sfruttando come base di partenza proprio UCBLogo) sfruttando l'interfaccia Windows riesce ad essere più attraente e più completo nelle prestazioni, ma è anche vero che il fatto di avere un programma che gira indifferentemente su qualsiasi tipo di macchina non è per nulla da sottovalutare. E quando dico su qualsiasi tipo non intendo solamente le diverse piattaforme: UCBLogo è talmente snello da poter girare senza problemi anche su un vecchio 286 attrezzato magari con un sistema operativo FreeDos. Questo significa poter riutilizzare delle macchine anche obsolete per fare informatica ad un livello più che dignitoso.

Ecco la schermata di UCBLogo per Win9x:



MSWLogo

MSWLogo è un programma sviluppato da George Mills sull'ossatura del programma UCBLogo, creato da Brian Harvey. Mills ha curato l'interfaccia grafica a finestre del programma, che non è invece disponibile per UCBLogo.

Si tratta di un Logo classico, che implementa tutte le funzioni conosciute di questo tipico linguaggio, ma anche molte altre, rendendolo pertanto uno strumento per la creazione e lo studio di programmi anche particolarmente sofisticati.

E' disponibile solamente per ambienti Windows (a differenza di UCBLogo, che è disponibile per ambiente Dos, Win9X e anche per Linux e Mac) e questo ne costituisce un limite. (E' comunque possibile far girare MSWLogo sotto Linux sfruttando l'emulatore Wine, ma non è una cosa semplice)

MSWLogo è un linguaggio interpretato e, come del resto tutti gli altri programmi Logo-like, possiede una caratteristica unica che non è presente in nessun altro linguaggio di programmazione: la grafica della Tartaruga. La grafica della Tartaruga è un semplice e potente insieme di comandi che servono a manovrare gli spostamenti di una tartaruga. Nelle prime versioni del Logo la tartaruga era una tartaruga meccanica con una penna ad inchiostro, che muovendosi tracciava dei disegni su dei fogli di carta. Nel caso di MSWLogo viene chiamata tartaruga il cursore triangolare che è presente sullo schermo e che, al posto dei vecchi robot, serve per tracciare dei disegni sullo schermo.

La grafica della tartaruga riesce a riempire una grossa lacuna presente nel linguaggio tradizionale: essa rende possibile un feedback immediato. E questo feedback immediato rende l'apprendimento divertente e più facile. Lo scopo del Logo è quello di insegnare a giovani e vecchi come programmare, per rendere facilmente accessibili tutte le caratteristiche educative che possiede questa attività: abitudine a strutturare il ragionamento, sviluppo della capacità di astrazione, abitudine a strutturare rigorosamente (anche se in modo del tutto creativo) i vari comandi disponibili.

Tra le caratteristiche più avanzate di MSWLogo possiamo trovare la gestione indipendente di più tartarughe e l'interfacciamento MCI. La MCI (Media Control Interface) è un interfaccia di comando per dispositivi multimediali e relativi file. Le applicazioni scritte in MSWLogo potranno, quindi, controllare ogni dispositivo multimediale supportato, incluse playback audio e registrazioni audio.

Ecco un elenco dei dispositivi supportati:

animation	Animation device
cdaudio	Audio CD player

dat Digital audio tape player

digitalvideo Digital video in a window (not GDI based)

other Undefined MCI device

overlay Overlay device (analog video in a window)

scanner Image scanner

sequencer MIDI sequencer

vcr Videotape recorder or player

videodisc Videodisc player

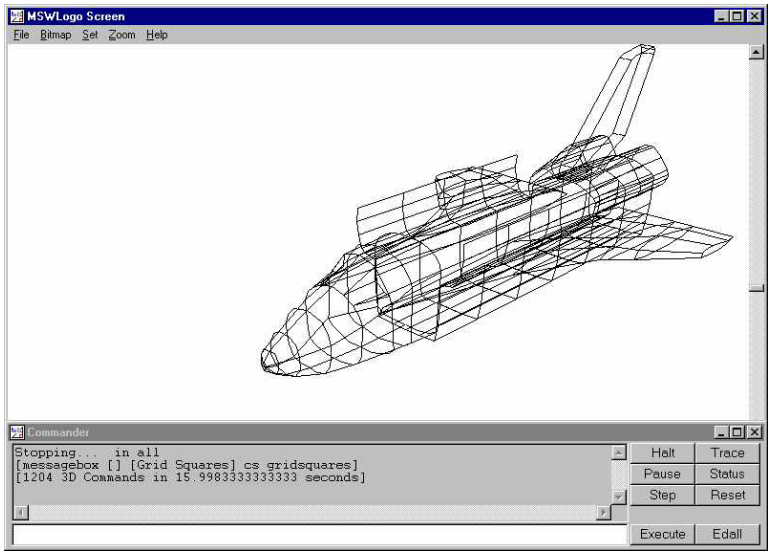
waveaudioAudio device that plays digitized waveform files

Come è possibile notare, MSWLogo offre delle possibilità non indifferenti. Il suo principale difetto, soprattutto nell'utilizzo con bambini delle scuole elementari, è l'assenza di una interfaccia che faciliti in qualche modo la programmazione: tutte le istruzioni devono essere scritte passo per passo e il minimo errore causa il malfunzionamento del programma (cosa che in ogni caso, anche se scomoda, personalmente ritengo educativa).

Non è un programma che può essere lasciato nelle mani di un bambino sperando che "giocandoci" riesca a riuscire a farci qualcosa, almeno non senza un periodo iniziale di apprendimento dei suoi comandi di base.

Esiste una versione di MSWLogo che può girare anche con sistemi operativi Windows 3.11 e una che può girare anche con computer di scarsa potenza. E', ad ogni modo, un programma che non richiede grosse risorse neppure nella versione per Win9x e può funzionare tranquillamente su qualsiasi macchina dal 486 al Pentium più veloce.

Questa è la schermata che presenta MSWLogo:



Drape

Drape è un programma per ambiente Win9X, che è stato scritto da Mark Overmars allo scopo di insegnare alcuni aspetti di base della programmazione. Penso che sia uno dei migliori programmi per l'apprendimento dell'informatica disponibili. L'impostazione concettuale ricorda molto da vicino il linguaggio Logo (come nel Logo, infatti, i programmi scritti con Drape creano dei disegni), ma con una differenza fondamentale: mentre nel Logo le istruzioni sono dei comandi scritti, Drape è interamente strutturato per mezzo di immagini; ad esempio se in Logo per disegnare un quadrato devo una serie di istruzioni simile a questa:

fd 10
lt 90
fd 10
lt 90
fd 10
lt 90
fd 10

Il programma equivalente in Drape avrà la seguente forma:

Il primo comando (la freccia verso destra) indica che una linea deve essere disegnata. (Per il momento è meglio ignorare la forma di diamante sulla sinistra, che è solamente un indicazione della procedura). Il comando seguente esegue una rotazione di 90 gradi, successivamente viene disegnata una seconda linea, etc.

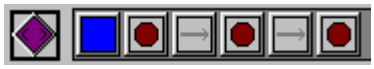
Come è possibile vedere, non ci sono più istruzioni scritte, ma solo piccoli simboli che prendono il posto delle solite righe di istruzioni dei linguaggi di programmazione. Questo rende Drape un linguaggio di programmazione davvero molto adatto anche ai bambini più piccoli.

Come nel linguaggio Logo, i programmi scritti con Drape creano dei disegni, ma Drape non si limita a questo. In questo ambiente sono infatti presenti tutte le caratteristiche importanti che devono essere presenti in un linguaggio di programmazione: comandi iterativi (ad es. il comando "repeat", ripeti), istruzioni condizionali, ecc. Inoltre, proprio come il Logo, dispone di comandi per richiamare altre procedure, consentendo l'inserimento di più procedure, una all'interno dell'altra.

Drape dispone di comandi che disegnano linee e forme, comandi che spostano il cursore in una particolare posizione, comandi che impostano proprietà come colore, larghezza e dimensione di linee, comandi per determinare azioni tramite i pulsanti del mouse. Ogni comando è, come già detto, rappresentato da una piccola figura e pertanto un programma consiste in una sequenza di queste figure. Per creare un programma è necessario trascinare con il mouse le figure nel posto appropriato della sequenza. E' possibile, comunque, visualizzare anche il programma "scritto" con le istruzioni corrispondenti ai disegnini (il cosidetto "listato"), anche se non è possibile modificare direttamente le istruzioni scritte. Inoltre le istruzioni "scritte" sono più simili ad una sintassi Basic che non ad un vero e proprio Logo.

Questo programma è stato pensato con tre differenti livelli di difficoltà, per poter "dosare" le sue potenzialità in base all'età e all'esperienza degli alunni; i livelli sono: easy, normal e advanced.

Ecco un altro esempio. Esso disegna 3 cerchi blu, uno vicino all'altro.



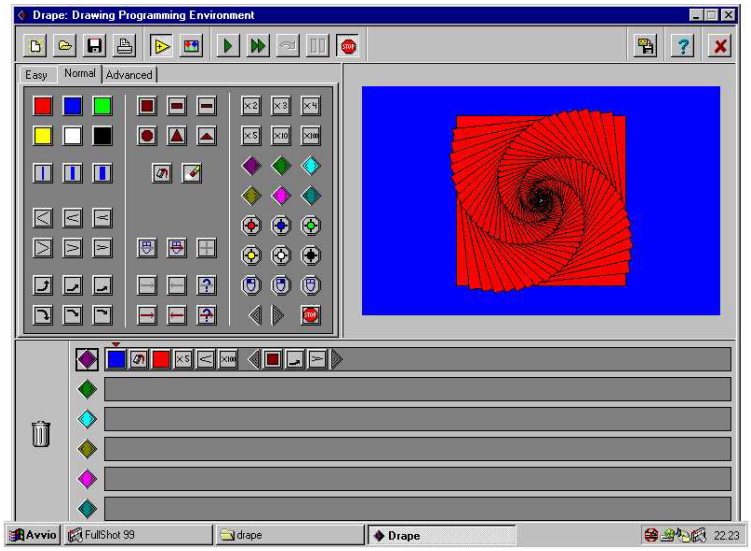
Il primo comando imposta il colore blu. Poi viene disegnato un cerchio. Il terzo comando sposta il cursore sulla destra. Viene disegnato un altro cerchio, e così via.

Drape è semplice da usare, ma richiede comunque una certa pratica per poterlo utilizzare. Si può incominciare a giocare con gli esempi forniti modificandoli e variandoli a piacere, per poter prendere confidenza con il programma. Si possono trovare numerosi esempi nella cartella "Examples".

Per utilizzare Drape è richiesto un moderno PC (è sufficiente anche un Pentium della prima serie, anche se le prestazioni saranno rallentate) con installato Windows 95 o NT 4.0 (o successivi). Dopo l'installazione il programma usa meno di un Megabytes di spazio su disco. Inoltre non modifica in nessun modo i registri di Windows, rendendo la sua gestione e disinstallazione una semplice operazione di cancellature di file.

Drape può essere usato gratuitamente, in qualsiasi modo, per qualsiasi scopo. Può essere distribuito con qualsiasi mezzo, a condizione che i file originali come forniti dall'autore rimangano intatti e non venga richiesto nessun pagamento se non quello per un ragionevole costo di distribuzione.

Questa è la schermata che presenta Drape:



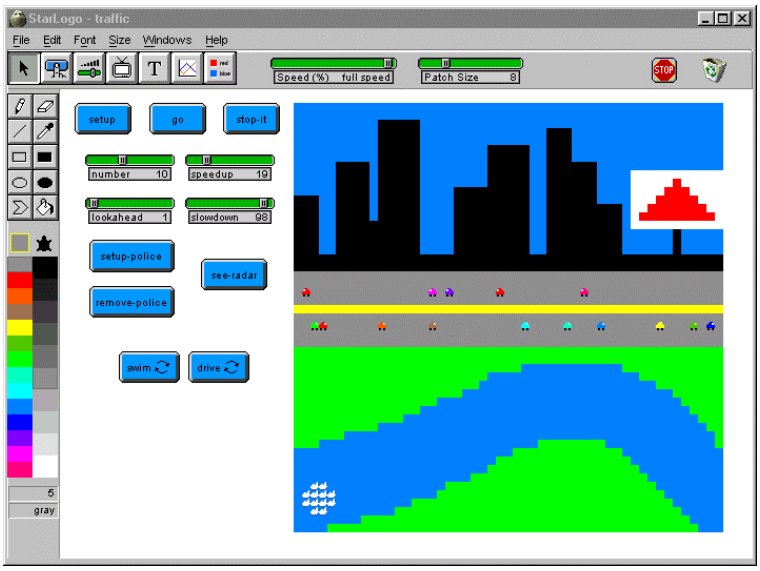
STARLOGO

StarLogo è un ambiente di programmazione per la creazione di simulazioni, sviluppato dal MIT (Massachussets Institute of Technology) nato con lo scopo di esplorare il funzionamento di sistemi decentralizzati (sistemi che sono organizzati sebbene non ci siano organizzatori, sistemi che posseggono una coordinazione senza la presenza di un coordinatore). Con StarLogo è possibile creare ed analizzare modelli di moltissimi tipi di fenomeni di vita reale, anche molto diversi tra di loro, come ad esempio il volo organizzato degli stormi di uccelli, la formazione di code nel traffico automobilistico, le vita nelle colonie di termiti e anche alcuni fenomeni dell'economia di mercato.

Nel settembre 2002 è stata rilasciata la versione 2.0 del programma. StarLogo è sviluppato in linguaggio Java e presenta pertanto la caratteristica di essere multipiattaforma: esistono infatti versioni per Windows, per Linux e per Mac. Richiede computer abbastanza potenti altrimenti la velocità di esecuzione può risentirne (è consigliabile, in ambiente Windows, almeno un Pentium III). Data la complessità delle tematiche affrontate è un linguaggio di programmazione adatto soprattuto per ragazzi dai 12-13 anni in poi.

I modelli decentralizzati complessi stanno registrando sempre più utilizzatori all'interno della comunità scientifica, essendo spesso l'unico modo possibile di spiegare o modellizzare fenomeni complessi. Ma molti continuano a resistere a queste idee, presupponendo controlli centralizzati anche dove non ne esistono (ad esempio è comune convinzione che negli stormi di uccelli ci sia un leader). StarLogo è stato progettato proprio

per aiutare gli studenti (ma è anche adatto alla ricerca vera e propria) a sviluppare nuovi modi di pensare e a comprendere i sistemi decentralizzati.

La caratteristica che lo distingue totalmente dagli altri sistemi Logo consiste nel dover dividere concettualmente in due parti distinte la programmazione: la programmazione delle tartarughe (che possono essere migliaia, comandate in parallelo) e la programmazione di un “ambiente” (costituito da migliaia di “patches”) con il quale le tartarughe interagiscono. Le patches possono essere, di volta in volta, pezzetti di terreno, erba, lo sfondo del cielo, etc. Anche le tartarughe assumono in ogni progetto un ruolo diverso: termiti, automobili, uccelli, etc. ed è possibile assegnare loro proprio le forme desiderate.

Con StarLogo è possibile programmare le tartarughe per andare in giro a “sentire“ il loro mondo e di cambiare il loro comportamento basandosi su cosa sentono nelle patch sottostanti.