

---

# Insegnare il Moto con Boxer

---

Andrea A. diSessa  
Università della  
California, Berkeley  
USA

*Un corso sulla matematica del moto basato sulla premessa che tutte le persone coinvolte - studenti, insegnanti e progettisti - dovrebbero conoscere un computational medium generale e programmabile, Boxer.*

## INTRODUZIONE

L'essere istruito, nell'accezione convenzionale di capire ed essere capace di produrre il linguaggio scritto, pervade profondamente il nostro sistema educativo. Infatti, è difficile immaginare cosa succederebbe se leggere e scrivere non fossero un aspetto ovvio delle conoscenze di insegnanti e studenti. Ovviamente, studenti ed insegnanti, apprendono ed insegnano le discipline umanistiche leggendo e scrivendo, ma la situazione non è sostanzialmente diversa per la matematica e le scienze. Il libro di testo la fa da padrone in ogni fase dell'apprendimento, dall'organizzare quella che chiamiamo lezione, al fornire testi da leggere e problemi da risolvere. Piccole estensioni tecniche dell'istruzione linguistica, (compatibili con la tecnologia di carta e matita) quali l'algebra, il disegno di grafici e simili completano l'insieme degli strumenti per pensare che costituiscono e sviluppano la capacità di apprendere. Sembra inconcepibile che le proprietà di questa parte fondamentale del pensiero e dell'apprendimento non determinino, in modo rilevante, ciò che si insegna e come lo si apprende.

Per il prossimo secolo ed oltre, possibilità complementari costituiscono una prospettiva eccitante. E cioè, estendendo la struttura lineare del linguaggio scritto nelle forme inte-

rattive e grafiche, interconnesse, dinamiche, e ricche consentite dal computer, potremmo ampliare in modo sostanziale le basi materiali per pensare e apprendere, e con esse l'intera pratica educativa.

Semplificando molto si può dire che ci sono due concezioni contrastanti di un'istruzione rafforzata dall'uso del computer. Quella oggi dominante io la descriverei con l'aggettivo superficiale. Questa concezione vede esperti e progettisti di software fornire alla gente strumenti software eleganti e specializzati che impariamo ad usare nelle nicchie per le quali sono stati pensati - manipolatori di simboli, generatori di grafici, simulazioni e generatori di simulazioni, e simili.

Una visione contrapposta di istruzione completa con il computer offre una risorsa decisiva in più - accesso, in linea di principio per tutti, alla creazione e modifica delle caratteristiche interattive e dinamiche del *medium*, proprio quelle stesse caratteristiche che, in primo luogo, fanno di questo *medium* un'estensione ed un miglioramento del testo. In termini metaforici, leggere senza scrivere è solo un'istruzione a metà. Un'istruzione completa con il computer significa "scrivere" oltre che "leggere", creare e usare.

Molti chiamerebbero questa risorsa aggiuntiva e l'insieme di attività ad essa collegate programmazione. Io preferisco non pregiudicare le forme di accesso alla creazione

e modifica di strutture dinamiche ed interattive usando una parola, ormai connotata negativamente e riferita ad attività considerate per addetti ai lavori (p. es. “la programmazione è difficile e i programmatori la fanno per te”). Invece preferisco far riferimento al sistema globale caratterizzato da una programmabilità completa - fondamento di una nuova istruzione - come a un *computational medium*.

I problemi che riguardano l'utilità e la possibilità di sviluppare una istruzione completa con il computer sono complessi. Due questioni importanti sono accennate di seguito.

L'istruzione, nel senso convenzionale, non è un processo facile. In ogni caso le competenze si formano in modo cumulativo e l'apprendimento successivo dipende da quello precedente. Allo stesso modo padroneggiare un *computational medium* non è facile, tuttavia importanti risultati successivi si baseranno su quelli precedenti. Il secondo punto è correlato ai nostri obiettivi. Vogliamo che sia gli studenti che gli insegnanti abbiano la sensazione di possedere il medium, e che esso soddisfi i loro obiettivi e le loro necessità espressive. Il medium non deve solo servire gli istinti creativi degli sviluppatori.

In questo articolo esamino l'uso di un *computational medium*, **Boxer**<sup>1</sup>, in due versioni di un corso sulla matematica del moto. Descrivo semplicemente le molte forme di software che noi, insegnanti e studenti, abbiamo sviluppato all'interno di queste due classi. Alcune di queste forme (micromondi, tutoriali) risulteranno familiari. Altre che descrivo, sono meno note (per esempio forme di rappresentazione interattive) e più specifiche dell'uso di un *computational medium*. In tutti questi casi, verranno sottolineati l'accumulo graduale di competenza e senso di possesso espresso da studenti e insegnanti.

## DESCRIZIONE DELLE CLASSI

Entrambe le classi descritte avevano come obiettivo l'insegnamento della cinematica, la descrizione matematica del moto. Gli argomenti del corso includevano concetti di velocità istantanea e accelerazione in una e due dimensioni; moti relativi, sistemi di riferimento e composizione del moto; vettori e forma vettoriale della velocità e accelerazione; disegno di grafici (per esempio grafici di posizione, velocità e accelerazione e loro re-



**1 Boxer** è stato sviluppato da diSessa ed il suo gruppo come successore del Logo. Infatti i primi documenti di progetto di Boxer fanno esplicito riferimento a quei limiti del Logo come uno stile d'interfaccia legato ai limiti imposti dalle telescriventi. In molti Logo, ancora oggi, si danno comandi alla macchina in una finestra, una riga alla volta, mentre la grafica appare in un'altra. Un programma si scrive come sequenza di istruzioni con

un'editore, magari in un'altra finestra. Non c'è insomma una continuità concettuale e di interfaccia fra le modalità grafiche, iconiche ed interattive dei moderni computer ed il sistema programmabile che continua a ragionare come se l'attività di programmazione avvenisse ancora e soltanto con una telescrivente. Boxer invece parte dal presupposto che prima di tutto bisogna fornire un sistema ipertestuale, arricchito delle potenzialità grafiche, dinamiche ed interattive. Il sistema è usabile subito e con poco sforzo come editore di ipertesti, e a mano a mano che se ne scoprono le potenzialità estenderne l'utilizzo. Si chiama Boxer perché fa riferimento alla metafora di box (scatole) dentro altri box che corrisponde alla scomposizione gerarchica di un problema dove posso decidere di lavorare al livello di dettaglio voluto aprendo e/o chiudendo i vari box. L'altra innovazione è che non si comincia mai dal livello base di Boxer per poi partire ad esplorare il mondo, ma si comincia ad aprire e curiosare tra i vari “box” a disposizione che possono essere un programma di disegno geometrico (tipo Cabri), un libro interattivo di Fisica o altri documenti reperibili sulla scrivania del proprio computer o in rete, oppure un sistema di posta elettronica integrato con cui inviare messaggi sotto forma di box. Boxer non è ancora un prodotto commerciale, gli interessati possono esplorarlo su Mac presso l'Istituto per le Tecnologie Didattiche o richiedere una licenza d'uso all'università della California.[ndt]

lazioni); e quanto meno versioni qualitative delle idee di base di integrazione e differenziazione. Anche se, in generale, abbiamo cercato di evitare gli aspetti di causa del moto (dinamica), entrambi i corsi sono finiti con versioni della seconda legge di Newton, che specifica le relazioni tra forza, massa e moto.

La prima versione del corso ha coinvolto otto studenti di prima media<sup>2</sup>, quattro ragazzi e quattro ragazze, quattro giorni alla settimana durante l'intero anno scolastico. La scuola è una media inferiore privata di Oakland, California. Abbiamo organizzato il corso in modo che l'insegnante non avesse altri studenti da seguire, il corso era tenuto nella biblioteca della scuola che era comunque accessibile agli altri studenti. Poiché non avevamo computer a sufficienza (5-6), gli studenti hanno spesso lavorato al computer in coppia. Gli studenti coinvolti erano intelligenti e preparati, ma non eccezionali.

La seconda versione è stata sperimentata in un liceo privato di San Francisco ed è durata 15 settimane. Gli studenti coinvolti erano sei ragazzi e due ragazze. Come nel caso della prima media gli studenti hanno parteci-

*2 Il sistema scolastico americano prevede sei anni per la scuola elementare, la classe di cui si parla è una sesta (sixth grade). Il liceo è una high-school. La scelta di tarare i riferimenti al sistema scolastico italiano, da un lato semplifica la lettura dall'altro potrebbe trarre in errore evocando schemi e realtà educative non facilmente traducibili l'una nell'altra. Inoltre entrambe le scuole, dove è stato provato il corso con Boxer, hanno permesso l'organizzazione all'interno del normale anno scolastico di un corso sperimentale con solo 8 allievi.[ndt]*

pato all'esperimento su base volontaria, ma, contrariamente al gruppo della media, questi non brillavano dal punto di vista del profitto. Questa versione del corso è stata molto compressa se paragonata alla precedente. Infatti solo dieci settimane su di un totale di quindici sono state dedicate alla cinematica; il resto del corso è stato usato per imparare Boxer. La maggior parte degli studenti in entrambe le classi non aveva esperienze di programmazione e nessuno di loro aveva lavorato con Boxer.

Tra gli orientamenti che hanno caratterizzato questi corsi, a parte l'uso di Boxer, c'è stato un uso esteso di discussioni collaborative e di progetti sia individuali che di gruppo.

### MICROMONDI

Il micromondo ideale dovrebbe avere le seguenti proprietà: (1) una modalità d'interazione abbastanza semplice da essere facilmente appresa dagli studenti. (2) un ampio ventaglio di attività motivanti. (3) affidabilità nel mettere gli studenti in contatto con idee fondamentali della matematica o delle scienze esatte.

La figura 1 mostra l'interfaccia di un semplice micromondo chiamato "Number-

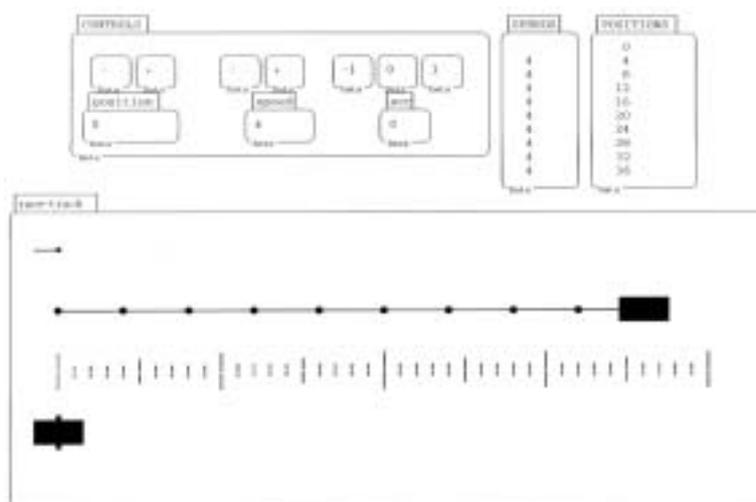
speed" progettato da Steve Adams, che è stato introdotto all'incirca dopo un terzo di entrambi i corsi. L'idea di base è che una rappresentazione della velocità in funzione del tempo espressa come lista di numeri costituisce un eccellente approccio alla comprensione degli elementi fondamentali di posizione, velocità e accelerazione. Gli studenti si esercitano con le seguenti idee: (1) la velocità determina una proprietà locale del moto; (2) la velocità fornisce l'unità di tempo d'incremento della posizione; (3) in maniera complementare, differenze di posizione determinano la velocità; e (4) lo stesso insieme di relazioni (2 e 3) è vero per accelerazione e velocità.

L'attività principale di questo micromondo è "corse di tartarughe". Gli studenti programmano liste di numeri selezionando i valori iniziali di posizione, velocità e accelerazione. I valori possibili per l'accelerazione sono più o meno 1 e 0. Si possono quindi ispezionare le liste di posizione e velocità e vedere il moto da esse determinato. Abbiamo incluso una serie di "sfide" a diversi livelli di difficoltà. Per esempio, gli studenti dovevano descrivere a parole particolari tipi di moto ai loro compagni i quali poi dovevano programmarli. La serie di "sfide" più complessa usava segnali posizionati lungo la pista della tartaruga ai quali l'una o l'altra delle due tartarughe programmate doveva essere in testa.

La presentazione di questo micromondo consiste, per la maggior parte, in strutture generiche di Boxer. In altre parole usa strutture di base con cui ogni utente ha familiarità. Le liste di numeri sono semplicemente box di dati (variabili) che la maggior parte degli studenti incontra durante le prime sessioni di lavoro con Boxer. Box di dati sono creati semplicemente premendo il tasto appropriato e il loro contenuto è modificato direttamente dall'utente come si fa con il testo in un programma di videoscrittura. Box di dati possono anche essere cambiati da un programma, e queste modifiche sono immediatamente visibili all'utente. Poiché box di dati possono essere "visti" e cambiati sia dall'utente che dal programma, autori di materiale didattico possono usare, in Boxer, box di dati come strumenti per l'input e l'output per l'interfaccia utente senza ricorrere a primitive speciali di IO o programmazione. Spesso, la parte di programmazione dedicata a IO e interfaccia costituisce lo sforzo principale nello sviluppo di materiale didattico su computer.

Ci sono molti vantaggi nell'usare componenti generici di Boxer, come box di dati, nei micromondi. Il primo è che l'uso di componenti generiche di Boxer consente

Figura 1.  
In Number-speed due tartarughe sono programmate per correre generando liste di velocità e posizioni. Qui la prima tartaruga ha posizione iniziale 0, velocità 4, accelerazione 0.



speed" progettato da Steve Adams, che è stato introdotto all'incirca dopo un terzo di entrambi i corsi. L'idea di base è che una rappresentazione della velocità in funzione del tempo espressa come lista di numeri costituisce un eccellente approccio alla com-

una familiarità immediata sulla base dell'esperienza precedente. Gli studenti sanno già che cosa sia possibile fare e che cosa, probabilmente, verrà loro richiesto di fare. Nei nostri corsi sul moto, abbiamo avuto la netta sensazione di essere vicini allo stato ideale - cioè gli studenti, per iniziare a rapportarsi con un nuovo micromondo a livello concettuale, non hanno bisogno di imparare nulla di nuovo da un punto di vista tecnico (per esempio qual'è la natura di quello che vedono sullo schermo; come si dovrebbe interagire con esso). Number-speed ne è un esempio.

Il secondo vantaggio è la flessibilità creata dall'accumulazione di capacità. Quando gli studenti lavorano con sistemi costruiti con parti note, hanno accesso ai meccanismi interni per esplorare il funzionamento dei micromondi, per modificarli o per estenderli; cosa che non è generalmente possibile. Nel caso di Number-speed abbiamo osservato un esempio eclatante di queste potenzialità, che abbiamo scherzosamente chiamato "apprendere barando" (Adams & diSessa, 1991). Una coppia di studenti di prima media stava lavorando all'ultima "sfida" del micromondo, che proponeva tre cambi di guida (posizione di testa) in una corsa di tartarughe. Questo problema è insolubile (con accelerazione costante)! Questi studenti capirono l'insolubilità del problema cambiando a mano i valori della lista delle velocità (di solito generata automaticamente una volta assegnati i valori di velocità iniziale e accelerazione) in modo da ottenere tre cambi di guida. Fatto ciò argomentarono che il tipo di lista di velocità necessaria alla soluzione del problema non poteva essere generata dal programma all'interno dei limiti imposti. Abbiamo osservato molti altri esempi di studenti che hanno usato produttivamente i micromondi proposti in modi che noi non avevamo immaginato (alcuni saranno descritti in seguito). Molti di questi usi erano dovuti alla possibilità che gli studenti avevano di accedere alle parti più "interne" del micromondo.

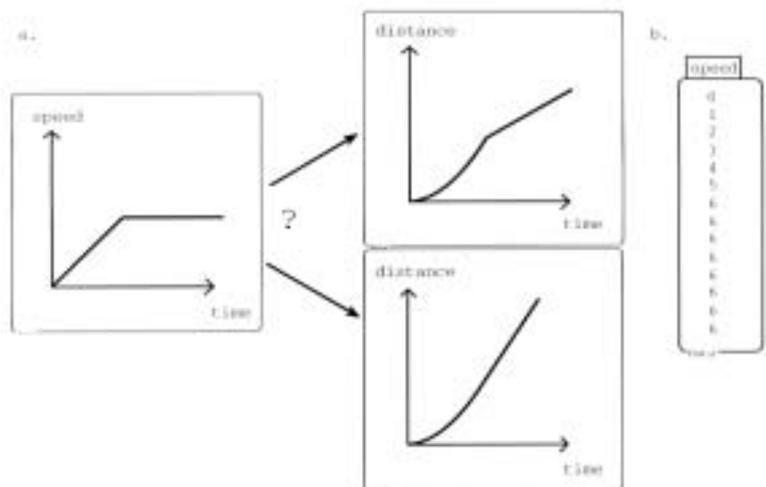
Il terzo vantaggio per gli studenti nell'usare strutture generiche, quali le liste di numeri di Boxer in Number-speed, è la cumulatività che scaturisce da ogni sistema di rappresentazione potente. Le liste di numeri sono rappresentazioni generali utili e efficaci sia dal punto di vista del calcolo che da quello intellettuale. Gli studenti troveranno molti contesti in cui in realtà la rappresentazione

lavora per loro. Liste di numeri di Boxer, oltre che in Number-Speed, sono state generate e usate dagli studenti in molte altre circostanze. Uno studente di prima media sviluppò un progetto autonomo di "analisi matematica" elaborando programmi generici per la manipolazione di liste che "derivavano" e "integravano" funzioni rappresentate da liste di numeri.

Number-speed è un esempio calzante della cumulatività della rappresentazione poiché gli aspetti concettuali del suo uso sono strettamente intrecciati con quelli tecnici. Nello stesso modo in cui imparando a produrre grafici si imparano anche le funzioni, le liste di numeri sono potenti strumenti concettuali per pensare il moto. Noi le abbiamo usate in maniera coerente in molte circostanze, per esempio come rappresentazione concettuale intermedia che permetteva agli studenti di passare più facilmente dalla velocità all'accelerazione.

In un episodio, studenti di prima media stavano discutendo su quale tipo di grafico della posizione potrebbe corrispondere ad una velocità che cresce linearmente fino ad un punto e poi rimane costante. Gli studenti sapevano che la prima parte del grafico sarebbe stata una curva concava all'insù e la seconda parte una linea retta e obliqua. Ma, nel punto di congiunzione di queste due parti del grafico, c'è un gomito oppure una transizione dolce (figura 2a)? Una ragazza pose fine alla discussione con il seguente ragionamento "la sua (della posizione) quantità di crescita ogni volta aumenta fino a un certo punto e poi resta costante." Il grafico deve inclinarsi sempre di più e poi continuare con la pendenza

Figura 2.  
Generare un grafico della posizione da quello della velocità.  
(a) La domanda. (b) Le liste di numeri sono uno strumento per decidere.



massima. Noi interpretiamo questo come un riferimento implicito ad una lista di velocità che incrementa le posizioni. È come se lei avesse costruito una lista di velocità (figura 2b) nella sua testa, corrispondente al grafico della velocità dato, quindi derivasse le implicazioni di ogni velocità per la lista delle posizioni e per il grafico.

La figura 3 mostra un micromondo per

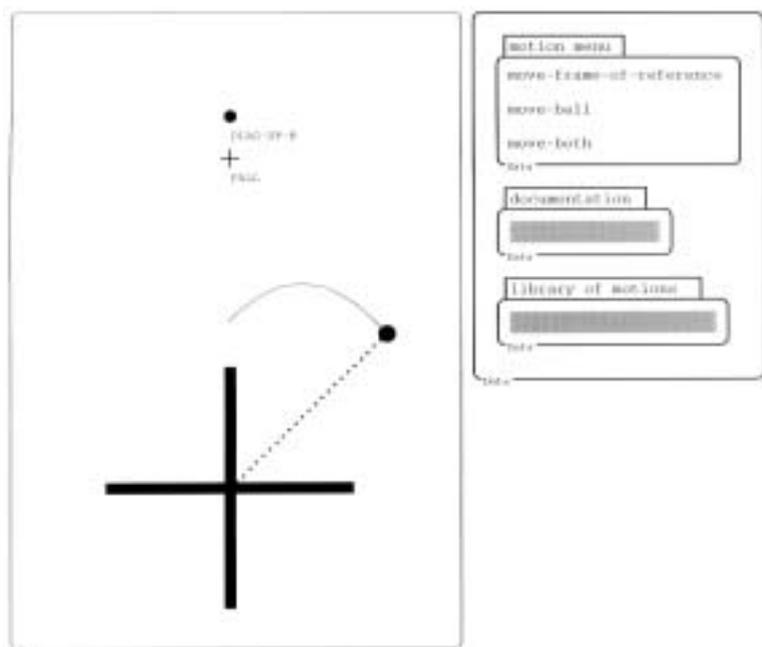


Figura 3.  
La croce rappresenta un sistema di riferimento (in questo caso sta "cadendo"). La palla si muove in diagonale verso l'alto (linea punteggiata) nel sistema di riferimento, la composizione dei due moti è un percorso a parabola.

moti relativi chiamato Elmira. Il box grande sulla sinistra è un box grafico, dove è mostrato il risultato di un programma tipo istruzioni Logo per grafica della tartaruga. Le figure che si muovono all'interno del box grafico sono "folletti"<sup>3</sup> (un tipo di tartaruga) la cui forma è stata modificata. La croce è un "folletto" e rappresenta un sistema di riferimento in cui la pallina (un'altro folletto) si muove. L'utente definisce sia il moto del sistema di riferimento che quello della pallina. Questi moti si possono visualizzare separatamente. Abbiamo chiesto agli studenti di prevedere il moto combinato che la pallina ha rispetto ad uno sfondo fisso quando sia la pallina che il suo sistema di riferimento si muovono simultaneamente. Gli studenti possono rallentare il moto, un passo alla volta, o usare una serie di strumenti analitici quali mostrare il percorso della palla nel sistema di riferimento in movimento, mostrare il percorso del sistema di riferimento in movimento o disegnare il per-

<sup>3</sup> In Boxer, come in molte versioni di Logo, si possono avere più tartarughe programmabili o folletti (sprites in inglese) [ndt]

corso della pallina nel sistema di riferimento dello sfondo fisso.

Elmira è stato inizialmente programmato in un giorno. Revisioni successive con l'introduzione di nuove o mutate caratteristiche in risposta ai primi esperimenti formativi non sono costate più di due giorni di programmazione. Questo non è fuori dalla norma con Boxer. Molti dei nostri micromondi sono stati realizzati al computer in pochi giorni. Tutto ciò è stupefacente se paragonato agli anni-uomo tipici dello sviluppo di software educativo. Le ragioni principali del ridotto tempo di sviluppo sono le potenti strutture generali (per es. per editare testi e grafica dinamica) che, in Boxer, sono sempre disponibili per sviluppatori, insegnanti e studenti. Il risultato dell'uso del "prototipo veloce" è che la maggior parte del tempo di sviluppo di un micromondo può essere dedicata al progetto didattico invece che alla programmazione. Inoltre pensiamo che l'uso di molti piccoli micromondi, invece che di pochi e complessi, semplifichi la situazione sia per gli sviluppatori che per gli studenti.

Uno dei "segreti" che rende possibile lo sviluppo veloce in Boxer è che possiamo far affidamento su conoscenze dello studente senza le quali sarebbe necessario molto lavoro di programmazione. Per esempio, non è necessario approntare procedure elaborate di gestione degli errori in quanto questi non sono eventi eccezionali per studenti che hanno programmato un *computational medium*. La maggior parte di questi non interferiscono con il lavoro intellettuale principale del micromondo.

Abbiamo presentato Elmira all'insegnante della prima media solo il giorno prima che lei lo usasse in classe. Anche se noi pensavamo di avere un buon progetto, l'insegnante ebbe molte reazioni forti. Per prima cosa lei voleva riorganizzare e semplificare i menu per gli studenti. Poi chiese di togliere tutta la documentazione, poiché riteneva che fosse un fattore di distrazione e di complicazione invece che un aiuto per gli studenti. Infine voleva modificare e riorganizzare la sequenza di problemi proposti. Noi facemmo tutti i cambiamenti e il "nuovo" micromondo fu sperimentato con successo il giorno dopo.

Se l'insegnante avesse o no ragione nel riprogettare il micromondo non è rilevante. Quello che lei aveva fatto di importante fu di rendere suo il micromondo. Il suo stile e le sue sensibilità nel rapporto con gli studenti

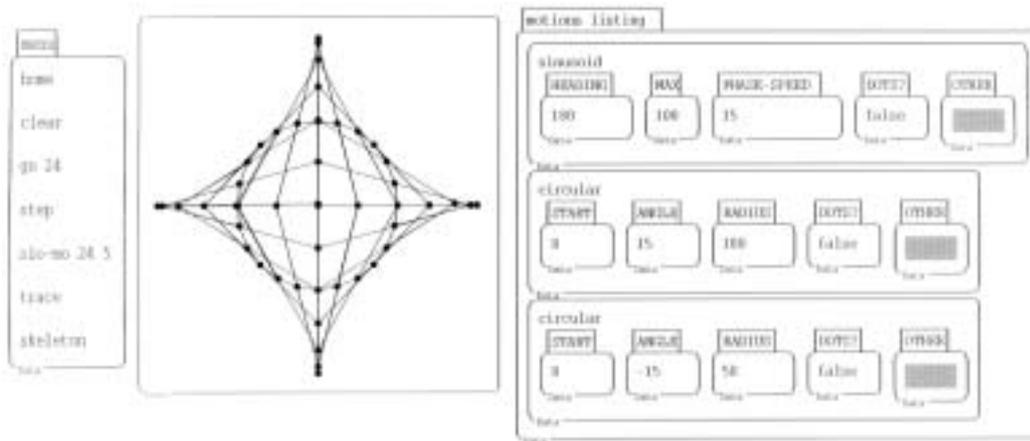


Figura 4.  
La traccia di una serie di “bastoncini,” che hanno il moto specificato nei tre box della “motions listing.” “Trace” (mostra il percorso del bastoncino finale) e “Skeleton” (lascia un’immagine di tutte le posizioni dei bastoncini) sono attivi.

poté stabilirsi facilmente. In generale, noi ci immaginiamo gli insegnanti come progettisti seri e cooperativi di materiale didattico di Boxer, non solo nelle fasi iniziali di progetto ma anche nel suo uso in classe.

È importante poter produrre materiali semplici e modificabili e, da parte degli insegnanti, poter riutilizzare ciò che apprendono nel modificare un micromondo. Nel seguito forniremo altri esempi delle possibilità e della potenza della modificabilità dei materiali da parte di insegnanti e studenti.

Qui descriviamo una caratteristica particolare di Elmira che ha favorito le modifiche da parte degli studenti. La libreria di moti che contiene tutti i moti disponibili nel micromondo è completamente accessibile ed estendibile dagli studenti. La libreria appare come un piccolo box grigio “ridotto”<sup>4</sup> (figura 3), che può essere aperto e ispezionato con un semplice click del mouse. Al suo interno ci sono tutti i moti predefiniti, ognuno dei quali è un box di dati con il nome. La rappresentazione interna dei moti è costituita da liste di numeri, in consonanza con Number-speed e più in generale con il curriculum. Poiché gli studenti potrebbero usare questo micromondo prima di conoscere le liste di numeri, abbiamo incluso nella libreria un traduttore capace di convertire una rappresentazione più familiare del moto (un programma Logo tipo “spostati in avanti”, “gira a destra”, ...) in liste di numeri. Nella nostra esperienza agli studenti piace sperimentare moti inusuali come “gravità inversa” (un sistema di riferimento che “cade verso l’alto”) e moti irregolari. Grazie alle caratteristiche di Boxer la programmazione del traduttore è costata solo 15 minuti.

Elmira è uno dei micromondi meglio studiati tra quelli prodotti per questi corsi (Di Sessa, 1989; Metz and Hammer, 1993). Questi studi dimostrano che gli studenti possono apprendere in modo efficace Elmira usando le loro intuizioni spaziali dinamiche piuttosto che costrutti formali quali equazioni e funzioni. Ciò contribuisce a confermare un assunto centrale dello sviluppo di un *computational medium* e cioè che nuove strutture grafiche, dinamiche ed interattive possono sviluppare forme di intelligenza autentiche che l’istruzione standard non riesce ad ottenere.

La figura 4 mostra un micromondo che noi abbiamo introdotto quasi contemporaneamente a Elmira. Si chiama Tinker, contiene magici bastoncini ognuno dei quali ha un moto caratteristico. Alcuni hanno sia lunghezza che direzione fissi. Altri crescono in lunghezza con tasso costante. Altri ancora girano perennemente in circolo. Gli studenti possono agganciare una serie di questi bastoncini e osservare il moto complesso risultante all’estremità dell’ultimo bastoncino agganciato. Questo micromondo si riferisce alla composizione dei moti e fornisce un’immagine estremamente concreta della scomposizione vettoriale del moto. Usiamo Tinker come un kit di modellizzazione dove lo studente esplora moti complessi derivandoli da moti più semplici. Per esempio, la composizione di un bastoncino che cresce costantemente con uno che “accelera verso il basso” produce tutte le forme di lancio di proiettili, e cioè parabole di ogni tipo, come per esempio un semplice lancio verso l’alto e successiva caduta, o una semplice caduta da posizione di riposo. L’appropriata

<sup>4</sup> I box di Boxer possono essere espansi fino a coprire l’intero schermo o collassati fino a diventare un piccolo box grigio che visualizza solo il suo nome e nasconde il contenuto. Espandendo e collassando i box l’utente controlla la complessità dell’informazione decidendo su cosa focalizzare l’attenzione. [ndt]

---

composizione di moti circolari genera sinusoidi (o più in generale epicicli), e la composizione di sinusoidi dà origine a cerchi o ellissi.

Tinker, ancor più di Number-speed o Elmira, si basa sulla capacità di mettere a disposizione degli studenti strutture generiche di Boxer. La lista di bastoncini che gli studenti generano e modificano è una sequenza di box contenuta in un box (tutti box di dati); ognuno di questi contiene i parametri attivi su cui opera il programma del micromondo. In pratica gli studenti modificano direttamente il codice del micromondo invece di una sua interfaccia. Mentre abbiamo fornito scorciatoie per operare i cambi più comuni al sistema di bastoncini, gli studenti potevano, e lo hanno fatto, accedere al “programma dei bastoncini” per operare i cambi più strani. A volte, gli studenti evitavano le scorciatoie suggerite in quanto riuscivano a capire e quindi a editare direttamente i box contenenti la rappresentazione dei bastoncini.

Tinker può produrre bei disegni e moti interessanti (figura 4). Infatti è stato progettato per aumentare queste possibilità; alcuni studenti hanno lavorato, divertendosi, alla sua esplorazione, estensione e perfezionamento. Ancora una volta un coinvolgimento così personale è un obiettivo esplicito del *computational medium*, e la struttura aperta di Boxer lo facilita.

Abbiamo sviluppato, per questi corsi, più micromondi di quanti possiamo descrivere qui. Quindi chiudiamo questa sezione con un aneddoto e cioè una cosa interessante accaduta durante un incontro nella classe di prima. Un ragazzo ed una ragazza, che stavano usando un micromondo da me sviluppato, erano evidentemente annoiati e quindi cancellavano pezzi di micromondo per poi reincollarli. Chiesi se quest’attività non li preoccupasse. La ragazza rispose di no perché il micromondo era veramente semplice e lei poteva “ricomporlo” facilmente. Se, poi, potesse realmente farlo o no senza dover ricorrere ad una copia integra del micromondo è qui inessenziale. Evidentemente questi bambini avevano la sensazione che i micromondi di Boxer fossero loro e che quindi potevano farne quello che volevano.

## TUTORIALI ED ESERCIZI

Anche se tutoriali e “esercizi al computer” non sono particolarmente caratteristici del

nostro stile pedagogico, ne abbiamo sviluppato ed usato un certo numero. I tutoriali sono straordinariamente semplici da sviluppare e modificare in Boxer. La capacità di editare testi si usa dal primo incontro. Box dentro box sono creati premendo un tasto, e questi forniscono una buona struttura gerarchica da cui gli studenti hanno la visione d’insieme necessaria per una chiara organizzazione concettuale del materiale. Espandere e contrarre box è un modo per controllare la complessità sia per l’autore che per l’utente.

Poiché un *computational medium* è completamente integrato con un livello di programmazione, qualsiasi pezzo dinamico o interattivo può essere incluso in un tutorial. Questi possono essere semplicemente “immagini in movimento”, o intere simulazioni. Strumenti generali costruiti per altri fini possono essere facilmente inseriti o usati per costruire pezzi del tutorial. Se fosse utile usare uno strumento per la costruzione di grafici nel mezzo di un tutorial è sufficiente copiarlo e incollarlo. Un esempio, che descriveremo in maggior dettaglio in seguito, è costituito dai vettori che uno studente di dottorato ha programmato come estensione di Boxer. I vettori sono box grafici arricchiti con particolari capacità. (Fig.6). Essi contengono una freccia che rappresenta il vettore; questa può essere modificata in direzione o lunghezza con il mouse. Ogni box grafico, inclusi i vettori, può essere “girato” per accedere ad una struttura non grafica. In questo caso, appaiono le coordinate del vettore che possono essere modificate. La caratteristica fondamentale dei vettori di boxer è quella di poter compiere operazioni di calcolo. Si può eseguire “somma  $\mathbf{V}$  e  $\mathbf{W}$ ” e vedere il vettore risultante, o si può chiedere ad un oggetto grafico di Boxer di muoversi lungo un particolare vettore.

Per produrre un tutorial sui vettori in Boxer abbiamo scritto il testo e inserito i vettori nei posti appropriati premendo un tasto particolare. Una riga o due di programma può generare illustrazioni delle proprietà dei vettori, o immagini dinamiche di moti controllati da vettori.

Dal punto di vista dello studente i tutoriali di Boxer sono aperti e flessibili; è quindi possibile personalizzarli e dominarli. Gli studenti possono, in ogni momento, scrivere appunti o rispondere per iscritto a domande dell’insegnante. Un buon esempio della flessibilità di Boxer nel sostenere l’iniziativa de-

gli studenti è quanto accaduto durante il tutoriale sui vettori. Nella sezione sulle proprietà dei vettori, un paio di studenti del liceo ha deciso autonomamente di analizzare cosa succede a cambiare di posto a tre vettori, invece che a due come proposto dal tutoriale per illustrare la proprietà commutativa. È stato facile per loro “modificare il programma” nel tutoriale.

Spesso tutoriali e esercizi in Boxer sono il posto dove gli studenti non solo imparano un concetto, ma anche come implementarlo in un programma. I vettori ne forniscono anche alcuni esempi. Una parte del tutoriale sui vettori chiedeva agli studenti di completare una o due righe di codice per creare un particolare moto. Programmare con i vettori diventò la base per molti progetti individuali degli studenti.

Un’altro esempio di un esercizio che abbiamo costruito è un programma, quasi completo, assemblato in pochi minuti e costruito con i vettori, di una nave che naviga in un mare agitato. Lo studente deve scrivere la singola riga di programma che fa muovere in modo corretto la nave, dati il suo moto e quello del mare espressi in termini di vettori. Gli esercizi si sono dimostrati efficaci nel collegare i moti relativi all’addizione di vettori.

## UN LINGUAGGIO DI MODELLIZZAZIONE FLESSIBILE

Programmare, in quanto parte dell’istruzione generale di uno studente, è un eccellente linguaggio generale di modellizzazione. In un certo senso, è un linguaggio universale di modellizzazione. Abbiamo fatto uso della programmazione in quest’accezione un paio di volte durante il corso sul moto. Infatti, il primo esercizio per gli studenti era di creare simulazioni di un numero di moti familiari, come un libro lasciato cadere, o che, dopo una spinta, scivola su di un tavolo fino ad arrestarsi, oppure un punto disegnato sul copertone di un pneumatico che rotola. Per gli studenti, ciò costituiva l’introduzione alla complessità (o semplicità) della descrizione analitica dei moti del mondo reale. Per noi, apriva alcuni importanti spiragli su come i nostri studenti iniziavano a pensare al moto. Solo un paio di studenti di prima media hanno pensato, inizialmente, che fosse importante visualizzare l’accelerazione di un oggetto in caduta libera. E hanno scelto di mo-

strare l’accelerazione con la rappresentazione di un oggetto che in realtà rallenta durante la caduta. Rallentava perché scelsero di rappresentare “la maggiore velocità” con più punti disegnati dall’oggetto in movimento! (figura 5) Questa è una rappresentazione non convenzionale, ma convincente, che in effetti è comune tra i bambini e gli adulti tecnicamente meno competenti. La rappresentazione potrebbe essere facilmente fraintesa da un insegnante. Finestre nelle capacità concettuali e rappresentative degli studenti, come quelle offerte qui da un semplice compito di programmazione/modellizzazione, sono importanti in ogni istruzione di tipo costruttivista. Inoltre quest’esempio mostra che la modellizzazione non deve, per forza, essere un’attività per addetti ai lavori condotta con strumenti specialistici e di difficile comprensione.

## RAPPRESENTAZIONI COMPATTE E FORMALI

Forse l’aspetto più controverso del modo in cui abbiamo insegnato il moto con un computational medium è l’uso della programmazione essenzialmente come sostituto dell’algebra in quanto linguaggio di rappresentazione formale in cui esprimere le definizioni di base e le leggi fondamentali. Consideriamo il seguente ragionamento. Tutti sanno che l’algebra è difficile (uno strumento tutt’altro che banale). Nonostante ciò pochi mettono in discussione il suo potere “cumulativo” e decidono di non insegnarla solo perché difficile. Qui stiamo proponendo che lo stesso tipo di considerazioni e giudizio equilibrato debbano applicarsi alla programmazione. Il programmare potrebbe avere un ancor maggiore potere cumulativo se usato in situazioni diverse dalla rappresentazione formale e presumibilmente richiede meno studio dell’algebra, almeno nel modo in cui è stata usata in questi corsi.

Ci sono diversi nodi epistemologici e pedagogici cruciali sull’uso della programmazione come una rappresentazione formale compatta che non possiamo approfondire in quest’articolo (per esempio Sherin, di Sessa, & Hammer, 1992). Qui sosteniamo in forma sintetica, il concetto di un uso della programmazione nel senso sopra descritto.

1) Il moto è un argomento interessante che si può insegnare a bambini di età inferiore a quella in cui in genere viene inse-

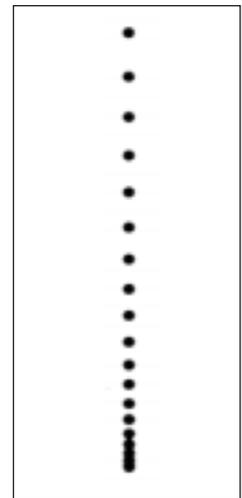


Figura 5.  
La rappresentazione costruita da uno studente di un oggetto che guadagna velocità cadendo. Una rappresentazione convenzionale sceglierebbe di graficare le posizioni ad intervalli di tempo uguali, il che produrrebbe punti sempre più distanti tra loro.

gnato. I bambini di prima media non conoscono l'algebra ma possono facilmente apprendere le poche nozioni di programmazione necessarie a far muovere gli oggetti. Se c'è un modo di descrivere il moto in forma precisa, generale e analitica per questi bambini, questo avviene attraverso la programmazione.

- 2) In realtà le rappresentazioni del moto attraverso la programmazione sono più interessanti dell'algebra. Un programma fa veramente muovere gli oggetti e permette di interagire con essi; inoltre permette di trattare moti più complessi di quelli che si ottengono risolvendo equazioni.
- 3) La programmazione usa modelli del moto discreti, passo passo, importanti da un punto di vista psicologico in quanto modelli intermedi e manipolabili di processi continui.

La parte fondamentale dell'uso che noi facciamo della programmazione come lin-

moto. Infine abbiamo incoraggiato gli studenti ad usarlo, quando necessario, nei loro progetti.

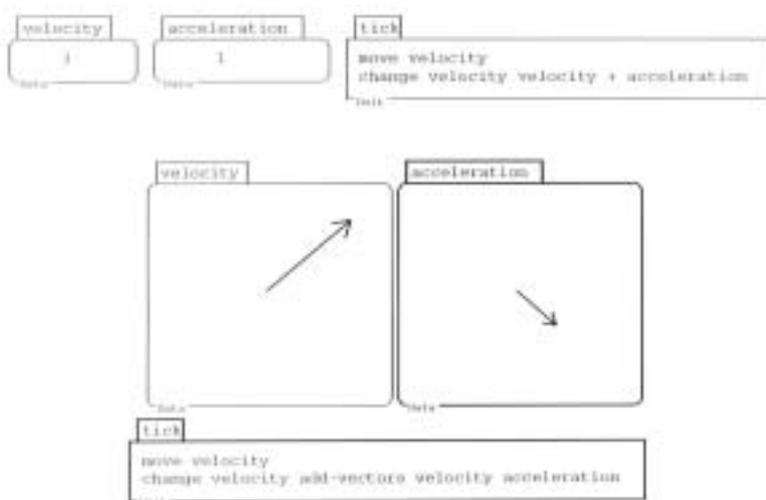
## UN MEDIUM PER LA PROGETTAZIONE COLLABORATIVA

Nel corso per il liceo abbiamo chiesto agli studenti di progettare in gruppo una versione del modello "a battiti" che esprimesse aspetti importanti della seconda legge di Newton - in assenza di forze il moto è un vettore costante, e le forze agiscono sommandosi vettorialmente alla velocità corrente. Questo lavoro in collaborazione è stato, con qualche limite, un successo ed è documentato in Sherin, diSessa & Hammer (1993) e diSessa (1995). In questa sede saremo sintetici. Dalla nostra analisi delle registrazioni video del lavoro di gruppo si evince che esso era fortemente influenzato da molti degli aspetti del corso che lo avevano preceduto. E cioè la cumulatività del processo di apprendimento è stata un fattore determinante di successo. Da una parte ci è sembrato evidente che l'uso dei vettori nel programma sia stato cruciale sia nel fare chiarezza che nel costringere alla precisione nella discussione. Dall'altra era chiaro che sia i programmi scritti dagli studenti (per esempio negli esercizi di moto di una nave sul mare - composizione di moti) che la comprensione concettuale ottenuta usando questi programmi siano stati di valido aiuto in questo sforzo.

Nel parlare degli aspetti collaborativi di "ricostruire la seconda legge di Newton" voglio sottolineare che un qualunque *medium* non può che offrire contributi limitati al successo dell'apprendimento. Boxer ha, come suggerito prima, aiutato gli studenti a capirsi l'uno con l'altro usando termini precisi e corretti. Comunque la collaborazione è un processo complesso e delicato. Nessun sistema di computer può "compensare" differenze culturali o simili ostacoli. D'altro canto dobbiamo onestamente riconoscere lo sforzo generoso e sistematico dei nostri insegnanti nell'incoraggiare spirito e strategie collaborative.

## STRUMENTI

I principi alla base della costruzione e dell'uso di strumenti sono simili a quelli dell'istruzione. La creazione di strumenti rappresenta la cumulatività effettiva in



guaggio formale è il modello "a battiti". La figura 6 mostra versioni scalari e vettoriali di questo modello. Due variabili  $v$  e  $a$  rappresentano la velocità e l'accelerazione. La procedura *tick* è eseguita ad ogni battito dell'orologio: (a) un oggetto si muove di una distanza  $v$  (definendo in questo modo la velocità come un pezzo di movimento); e (b) il valore della velocità è aggiornato sommando l'accelerazione.

Abbiamo insegnato il modello a battiti in tutoriali su moti ad una e due dimensioni (vettori). L'abbiamo usato più e più volte come un modello di riferimento per pensare al

Figura 6.  
Il modello "a battiti" in una e due dimensioni.

quanto include l'intelligenza del progettista. Spesso non è facile imparare ad usare gli strumenti ma è utile farlo per la potenza e espressività personale che liberano. Un *computational medium* è un meraviglioso contesto in cui sviluppare e condividere strumenti. Infatti, una delle mie immagini favorite di una futura pratica educativa basata sui *computational media* è quella di costruire e condividere strumenti per apprendere. I vettori, di nuovo, possono essere un eccellente esempio. Lo studente di dottorato Bruce Sherin, autore dei vettori in Boxer, ha reso possibile lo sviluppo di molti tutoriali e micromondi. Ne citiamo uno qui ed un altro nel seguito.

Decidemmo che il moto relativo era un eccellente introduzione alla cinematica a due dimensioni, così svilupparammo un semplice gioco come introduzione all'argomento.<sup>5</sup> "Cheeser" presenta un topo che corre su di un tavolo cercando di raggiungere e mordere un pezzo di formaggio, appeso ad un filo, che sfiora la superficie del tavolo. Il tavolo, a sua volta, si muove in modo sia regolare che irregolare, quindi il topo deve reagire per bilanciare o compensare ogni moto del tavolo. Gli studenti hanno "giocato al topo" manipolando un vettore che rappresentava la sua velocità corrente in tempo reale.

### ATTIVITÀ STILE "MBL"

L'aggiunta di interfacce poco costose a sensori esterni (Microcomputer-Based Laboratories) permette uno stile di software significativo che è facilmente inseribile in un *computational medium*. Invece di fornire strumenti per algoritmi di "smoothing" e per la generazione automatica di grafici, abbiamo deciso di dare agli studenti accesso diretto ai dati provenienti da un sensore di moto. La nostra opinione è che la flessibilità e concretezza dell'interazione diretta con i dati sperimentali, compresi gli errori strumentali, compensassero ampiamente le conseguenze negative di un'interfaccia di così basso livello. Questa scelta è coerente con la filosofia di Boxer di rendere disponibili, quando possibile, tutte le risorse del computer senza introdurre "scatole nere" che nascondono strutture potenzialmente illuminanti e potenti (come la singola riga di codice necessaria per disegnare il grafico dai dati). Un esercizio dimostratosi sorprendentemente difficile era semplicemente quello di avere un ogget-

to grafico che tracciasse una (versione in scala) del moto di un oggetto che veniva mosso di fronte ad un rilevatore di distanza.

Un'altra attività di laboratorio ha riguardato la produzione di foto stroboscopiche che presentano esposizioni multiple di oggetti lanciati e lasciati cadere. Avevamo preparato il terreno per quest'esperimento facendo discutere gli studenti su varie questioni: come un oggetto accelera cadendo, oppure se la componente orizzontale del moto di un oggetto lanciato diminuisce o meno. Alcuni studenti di prima media e del liceo credevano (correttamente) che la velocità di un oggetto in caduta venisse incrementata di una quantità costante a intervalli regolari. Alcuni pensavano che fosse moltiplicata per un fattore costante. Una ragazza di prima pensava che il moto dovesse avere un jerk<sup>6</sup> costante (velocità di accelerazione, derivata terza - che lei aveva scoperto estendendo il modello "a battiti", aggiungendo una variabile che incrementava l'accelerazione!). La maggior parte degli studenti credeva che la componente orizzontale del moto di un proiettile svanisse, o spontaneamente o come risultato dell'interferenza della gravità.

Le giornate precedenti l'esperimento con foto stroboscopiche non erano andate così bene come ci aspettavamo nella classe del liceo. Così la notte precedente l'analisi dei dati costruimmo uno strumento che doveva semplificare l'esercizio. Sostanzialmente gli studenti dovevano sistemare una serie di vettori (accelerazioni) per ottenere che un oggetto grafico seguisse le immagini degli oggetti lanciati e lasciati cadere digitalizzate ed inserite sul computer. Di nuovo uno studente ha avuto un'idea migliore: con un po' d'aiuto fu capace di estendere il nostro strumento in modo da lavorare con le componenti orizzontale e verticale separatamente.

Adattabilità e flessibilità sono contributi chiave di un *computational medium* per dare un senso di proprietà personale e di gruppo del mezzo.

### PROGETTI E LAVORO PERSONALE

Durante il corso gli studenti hanno particolarmente gradito progetti personali e discussioni in classe (come "ricostruire la seconda legge di Newton" descritto prima). Abbiamo lasciato molto spazio ai progetti. Naturalmente, in quanto educatori con l'obiettivo di sviluppare i concetti di moto, non sempre

*5 Non è un caso che moto relativo e composizione di moti fossero presenti in tante attività proposte agli studenti. Abbiamo deciso che questi sono concetti fondamentali e così li abbiamo enfatizzati.*

*6 Abbiamo introdotto questa terminologia convenzionale dopo che lo studente aveva spontaneamente esplorato l'idea.*

---

questi ci interessavano; tuttavia il coinvolgimento personale che i progetti comportavano è stato quasi sempre gratificante. I progetti sono un contesto importante in cui un *computational medium* fornisce risorse critiche per aiutare lo studente ad acquisire un senso di possesso delle idee e del software, stimolando un'appropriazione profonda di entrambi.

All'inizio del corso, in prima media, abbiamo invitato gli studenti a contribuire a un "museo delle velocità" descrivendo ed illustrando aspetti della velocità che loro conoscevano. Una ragazza si lanciò entusiasticamente in un gioco di "bambini che rimbalzano" dove "Adorable" e "Cutie" erano bambini lanciati fuori dalla finestra di un ospedale in fiamme. Il giocatore doveva muovere una rete al suolo per salvarli altrimenti sarebbero rimbalzati sul selciato. Questa ragazza, come altri, chiese di poter lavorare al di fuori dell'orario scolastico per completare il suo progetto.

Altre due ragazze hanno lavorato ad una simulazione in scala dell' "Autista pericoloso José Canseco" (con riferimento al famoso giocatore di baseball che più volte si era trovato nei guai per guida pericolosa). Il "Pericoloso José" investe l' "Innocente Sue" con la sua Corvette. È risultato concettualmente entusiasmante calibrare le velocità di Sue che cammina e José che guida in scala appropriata al contesto di un pezzo di marciapiede lungo 60 metri. Le due ragazze hanno spontaneamente usato due differenti convenzioni per rappresentare la velocità, entrambe della forma "tempo impiegato a percorrere un specifica distanza". Le due convenzioni sono divenute un argomento di discussione con l'insegnante, così come quello di convertirle in una rappresentazione convenzionale della velocità ("invertire le frazioni").

Un ragazzo nella stessa classe ha prodotto un impressionante gioco d'avventura sui grafici che presentava vari livelli di difficoltà nell'identificare grafici che corrispondessero a vari moti incontrati in una storia fantastica complessa, prodotta in gran parte da un altro studente. Il gioco d'avventura era corredato da spiegazioni sulle convenzioni grafiche e istruzioni su come giocare.

Nel liceo molti dei progetti erano giochi di moto. Uno si chiamava "lunar lander" dove il giocatore cerca di far atterrare una navicella spaziale sulla luna posizionando ed accendendo i suoi razzi di controllo. Un altro era

un difficile gioco di ormeggio di un'imbarcazione nel mezzo di correnti vorticose. Parecchi progetti hanno usato i vettori.

Un gruppo di studenti ha condotto uno studio sulle concezioni errate del moto intervistando compagni e professori. Come molti ricercatori prima di loro, hanno scoperto che le persone sembrano "conoscere" molte cose sul moto che non sono vere. Il messaggio di quest'ultimo esempio è che consideriamo un *computational medium* una risorsa da usare quando realmente necessario.

## CONCLUSIONI

Abbiamo descritto i molti modi in cui un *computational medium*, **Boxer**, è stato usato in due corsi sulla matematica del moto. Abbiamo sviluppato micromondi per gli studenti che favoriscono l'incontro con idee fondamentali e lasciano spazio all'iniziativa di insegnanti e studenti nel concettualizzare e mettere in pratica queste idee. Abbiamo anche sviluppato tutoriali ed esercizi per gli studenti che contengono rappresentazioni dinamiche ed interattive che loro stessi possono modificare. Gli studenti hanno usato la programmazione sia come uno strumento generale di modellizzazione per esplorare il moto, sia come rappresentazione formale e precisa di definizioni e costrutti di base. È stata usata anche come un mezzo per la progettazione collaborativa e come strumento espressivo per migliorare la collaborazione in genere. Lo sviluppo di strumenti generali, come i vettori di Boxer, non solo ha migliorato la nostra capacità di sviluppare materiali didattici di ogni tipo, ma questi stessi strumenti sono stati usati dagli studenti in molti modi. Per esempio sono serviti da contesto di lavoro in cui le idee (come i vettori) lavoravano in modo trasparente facendo muovere oggetti in modo comprensibile. Strumenti e programmabilità ci hanno anche permesso di incorporare facilmente in questi corsi sensori, altri dati ed analisi di esperimenti. Infine abbiamo sottolineato come un *computational medium* fornisca un eccellente supporto per il lavoro indipendente degli studenti.

Attraverso questi esempi abbiamo cercato di illustrare gli aspetti generali dell'uso di un *computational medium* in un processo istruzionale. Innanzitutto mentre ammettiamo che apprendere un *computational medium* non è compito facile, speriamo di aver dimo-

strato, almeno in parte, la forza pratica ed intellettuale che studenti ed insegnanti possono ottenere impadronendosi gradualmente di una nuova estensione espressiva del pensiero coadiuvato da strumenti esterni. Forse l'affermazione più provocatoria implicita in queste descrizioni è che, con un *computational medium*, siamo riusciti a insegnare alcuni aspetti importanti della matematica del moto in modo non convenzionale ma convincente a studenti di un livello scolare (prima media) lontano da quello in cui in genere queste idee vengono introdotte.

Usare la programmazione e le capacità generali di boxer in una esperienza di apprendimento, come un tutoriale o un micromondo, amplifica le capacità dello studente di usarle in altri futuri incontri con Boxer. Qualsiasi strumento prodotto può facilmente essere incluso in qualsiasi produzione futura, oppure, parti di esso possono essere cannibalizzate ed estese a situazioni molto diverse.

Una ultima considerazione, ma non meno importante, riguarda il nostro tentativo di documentare come un *computational medium* possa essere tanto aperto e flessibile da coinvolgere studenti e insegnanti nel ruolo di veri co-costruttori del medium e della conoscenza che speriamo possa scaturirne. Que-

sta è sia una realizzazione democratica, cioè gli "esperti sviluppatori di curricoli" non hanno tutte le risposte, che un riconoscimento del fatto che un coinvolgimento personale genuino è un potente elemento del processo di apprendimento. Se noi creiamo strumenti comprensibili, modificabili ed estendibili - e condividiamo con studenti e insegnanti quelle competenze che permettono di costruire e modificare - la scuola può diventare, al tempo stesso, più efficace e più personalmente significativa.

## RINGRAZIAMENTI

Quest'articolo è una versione ridotta del capitolo intitolato "The Many Faces of a Computational Medium: Teaching the Mathematics of Motion." pubblicato nel libro a cura di A. diSessa, C. Hoyles, R. Noss, with L. Edwards *Computers and Exploratory Learning*, Springer. Permesso di pubblicazione ottenuto sia dall'autore che dall'editore. Traduzione di Augusto Chiocciariello. Il lavoro descritto è stato, in parte, supportato dalla National Science Foundation, contratti numero NSF-MDR-88-50363 e NSF-RED-92-52725. Le opinioni espresse sono quelle dell'autore e non necessariamente quelle della National Science Foundation.

## Riferimenti Bibliografici

- Adams, S. and diSessa, A. A. (1991). Learning by cheating: Children's inventive use of a Boxer microworld. *Journal of Mathematical Behavior*, 10(1), 79-89.
- diSessa, A. A. (1989). A child's science of motion: Overview and first results. In U. Leron and N. Krumholtz (Eds.), *Proceedings of the Fourth International Conference for Logo and Mathematics Education*. Haifa, Israel: Israeli Logo Center, Technion - Israel Institute of Technology, 211-231.
- diSessa, A. A. (1990). Social niches for future software. In M. Gardner, J. Greeno, F. Reif, A. Schoenfeld, A. diSessa & E. Stage (Eds.), *Toward a Scientific Practice of Science Education*, Hillsdale, NJ: Lawrence Erlbaum, 301-322.
- diSessa, A. A. (1995). Collaborating via Boxer. In L. Burton and B. Jaworski (Eds.), *Technology - A Bridge between Teaching and Learning Mathematics*. Bromley, Kent, UK: Chartwell-Bratt, 69-94.
- diSessa, A. A. (1995). Designing Newton's laws: Patterns of social and representational feedback in a learning task. In R.-J. Beun & M. Reiner (Eds.), *Natural Dialog and Interactive Student Modeling*. Heidelberg: Springer-Verlag.
- diSessa, A. A., Hammer, D., Sherin, B. & Kolpakowski, T. (1991). Inventing graphing: Meta-representational expertise in children. *Journal of Mathematical Behavior*, 10(2), 117-160.
- Metz, K. E., & Hammer, D. M. (1993). Learning physics in a computer microworld: In what sense world? *Interactive Learning Environments*, 3(1), 55-76.
- Sherin, B., diSessa, A. A., & Hammer, D. M. (1992). Programming as a language for learning physics. Paper presented at the annual meeting of the American Educational Research Association.
- Sherin, B., diSessa, A. A., & Hammer, D. M. (1993). Dynaturtle revisited: Learning physics via collaborative design of a computer model. *Interactive Learning Environments*, 3(2), 91-118.