

Authoring software and JClic – Part I

Giorgio Musilli

Abstract *The close and open educational applications. The characteristics of the learning objects and the authoring softwares. JClic: description, authors, links, strengths and weaknesses, configurations.*

Key words *authoring software, jclic, learning objects, educational software*

Sommario *I programmi didattici aperti e chiusi. Le caratteristiche degli oggetti di apprendimento e i software autore. Jclic: descrizione, autori, collegamenti, pregi e difetti, configurazioni, gestione e distribuzione dei progetti.*

Parole chiave *software autore, jclic, oggetti di apprendimento, software didattici*

Introduzione

In questo contributo si tenterà anzitutto di operare una classificazione dei software didattici seguendo il criterio della loro personalizzazione. In secondo luogo saranno descritte le caratteristiche degli oggetti di apprendimento e i programmi per la loro realizzazione. Successivamente saranno evidenziate le caratteristiche dei programmi autore e in particolare dei software autore veri e propri. Infine per il programma JClic verranno indicati autori, collegamenti, distribuzioni, pregi e difetti e saranno mostrate le possibilità di configurazione del software e di distribuzione dei lavori prodotti.

Software chiusi, semiaperti ed aperti

Un comune criterio di classificazione dei programmi didattici riguarda le possibilità di *personalizzazione dei software* e soprattutto di *realizzazione di oggetti riutilizzabili*. Il problema può essere visto da diverse angolazioni e molto dipende dagli scopi per cui utilizziamo i programmi. È evidente però che un minimo di *flessibilità* è richiesto ad ogni applicazione che voglia trovare spazio nella scuola, soprattutto a causa della varietà delle esigenze che in essa emergono continuamente.

Una prima rozza (e forse poco chiara) distinzione considera due categorie di software, quelli “*aperti*” e quelli “*chiusi*”: secondo un’interpretazione molto diffusa sarebbero “*aperti*” i programmi modificabili da qualsiasi programmatore, purché sia disponibile alla diffusione del proprio codice sorgente e alla sua ulteriore modifica da parte di altri programmatori; sarebbero invece “*chiusi*” le applicazioni compilate non modificabili, ma solo usabili dall’utente: le stesse possibilità (limitate) di modifica (opzioni, scelte, funzioni) sarebbero spesso presenti solo perché previste nello stesso codice sorgente e facenti parte del progetto complessivo del programmatore.

Secondo la nostra prospettiva (di applicazioni dei software nei contesti scolastici) possiamo parlare di software “*aperti*” e “*chiusi*” nel senso più specifico di realizzazione di obiettivi educativi, produzione di oggetti didattici e personalizzazione delle caratteristiche, per cui la nostra azione sarà guidata da alcune domande base: “posso usare il software con tutti i bambini?”; “è possibile adattare il software alle caratteristiche di ogni alunno?”; “le funzioni e le opzioni del programma rispondono alle nostre esigenze (di realizzazione degli obiettivi, verifica degli apprendimenti, miglioramento delle prestazioni)?”; “posso produrre oggetti significativi (testi, animazioni, esercizi, album fotografici, compilation di brani musicali, giornalini) usando il software?”; “posso ampliare le possibilità del software inserendo esercizi, testi, grafici, disegni, video, realizzati in classe?”. Proprio considerando

tutte queste domande, ci rendiamo conto che non possiamo parlare semplicemente di software “aperto” e “chiuso”, ma sarà necessario specificare “quanto” un programma debba essere “aperto” perché sia funzionale ai nostri scopi. In particolare, per ogni applicazione dovremo verificare tre aspetti importanti:

- 1) la presenza di funzioni e opzioni, il loro numero, la loro significatività;
- 2) la modificabilità e integrabilità dei dati;
- 3) la possibilità di produrre “*learning objects*” attraverso il software (ovviamente per questo terzo aspetto sono rilevanti la complessità del prodotto realizzato e la sua riusabilità).

L’analisi delle combinazioni tra i vari aspetti ci può dare indicazioni significative sulla validità di un software educativo e ci può guidare nella scelta dei prodotti più efficaci per le nostre scuole:

- 1) a un livello massimo avremo un’applicazione che presenta tante e importanti funzioni e opzioni, consente la modifica integrale dei dati, produce un “learning object” completo, complesso e riusabile;
- 2) a un livello minimo avremo un software completamente chiuso, con pochissime opzioni e senza la possibilità di modificare e produrre dati. Si noti che non sempre è opportuno e necessario avere programmi personalizzabili: in diverse situazioni scolastiche anche software molto “chiusi” possono rivelarsi preziosi, soprattutto per obiettivi molto specifici della matematica e dell’italiano. Così per un programma sulle tabelline saranno molto utili eventuali opzioni e funzioni, ma non ci sarà bisogno di modificare gli archivi contenuti e per l’output basterà prevedere un report (rapporto) dell’attività svolta.

Senza entrare nell’analisi specifica del variegato mondo dei programmi educativi, dobbiamo notare che i tre aspetti precedentemente indicati (funzioni e opzioni, modifica dei dati e produzione di oggetti) quasi sempre si intrecciano e combinano in modi sempre differenti. Appare quindi opportuno ricorrere a una classificazione che abbia un carattere pratico, senza essere troppo semplicistica o arbitraria; relativamente alla possibilità di modificare e aggiungere dati ai programmi, possiamo individuare tre diverse categorie:

- 1) software “chiusi”: nulla può essere modificato nei dati;
- 2) software “semiaperti”: si prevede la possibilità di inserire nuovi dati accanto a quelli già presenti;
- 3) software “aperti”: i programmi sono pensati come strumenti flessibili per la creazione di esercizi, test, giochi; come esempi, si possono qui indicare Crucimaster, Puzzle Wizard, Hot Potatoes, EdiLIM, Cuadernia, Didapages, JClic, ma naturalmente la lista potrebbe essere molto più lunga.



Fig.1 – Software aperti

La modifica dei programmi semiaperti

In generale per la modifica di un programma “semiaperto” sono necessari semplici strumenti già disponibili nel sistema: un editor di testi (in formato ASCII Windows), un programma di grafica per le immagini (quasi sempre in formato .bmp, .jpg, .gif), software vari per le animazioni, i filmati e i suoni. Ovviamente librerie di testi, immagini e suoni potranno risultare molto utili. Nei pochi casi in cui sono necessari strumenti particolari (es. Flash, Visual Basic, Delphi) saranno gli stessi programmatori a indicare le soluzioni agli interessati e a fornire i necessari sorgenti. In ogni caso i dati sono sempre forniti come files esterni facilmente modificabili. A questo proposito un gestore di files più efficiente di quello fornito da Windows (es. Q.Dir) potrebbe risultare molto utile.

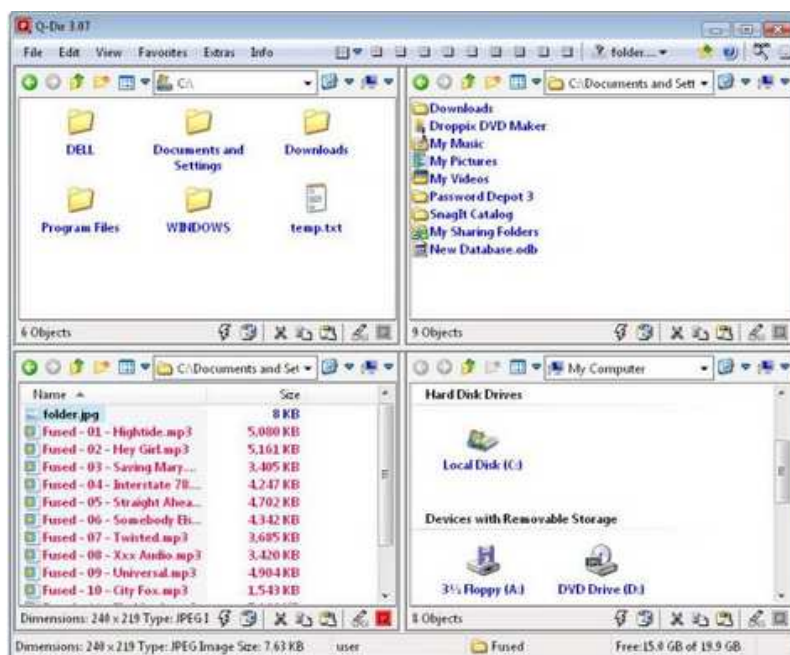


Fig. 2 – Una schermata a 4 finestre di Q-Dir

I programmi aperti ed i software autore veri e propri

Per i programmi “aperti” sono valide le indicazioni fornite per la modifica dei programmi “semiaperti”, con l’aggiunta che in questi software particolari sono sempre presenti opzioni per la creazione diretta dei giochi e degli esercizi. In quest’ambito possiamo indicare come “programmi autore veri e propri” o semplicemente come “programmi autore” quelle applicazioni che:

- 1) sono destinate a un uso prevalentemente didattico;
- 2) prevedono la possibilità di preparare un numero sufficientemente vario di esercizi;
- 3) permettono l’inserimento di elementi multimediali (immagini, suoni, animazioni);
- 4) comprendono un adeguato e amichevole editor interno;
- 5) hanno determinate caratteristiche di flessibilità;
- 6) consentono una gestione chiara degli archivi preparati e la loro condivisione in Internet;
- 7) non presentano una licenza legata strettamente all’acquisto di una particolare Lavagna Interattiva Multimediale.

In output i software autori veri e propri generano oggetti di apprendimento con specifiche caratteristiche; in particolare i “learning objects” (LO) prodotti sono granulari, riusabili e reperibili, cioè:

- 1) contengono una piccola porzione di conoscenze, da combinare in modo vario, opportuno e creativo con altre piccole porzioni;
- 2) sono riutilizzabili (con o senza modifiche e adattamenti);
- 3) sono archiviati e descritti tramite “metadati” (parole chiave).

Il software autore JClic: le ragioni di una scelta (pregi e difetti)

Il software JClic, open source distribuito con licenza GNU GPL, è stato scelto da chi scrive come mezzo privilegiato per la creazione di oggetti di apprendimento, con il solo limite di non supportare in modo efficiente filmati e animazioni. A JClic possono comunque essere affiancati tranquillamente altri prodotti molto validi, e cioè: 1) i migliori software per realizzare risorse testuali, grafiche, audio e video; 2) alcuni generatori di presentazioni (ad es. Didapages, CamStudio); 3) altri programmi autore veri e propri (eduLIM, eXeLearning e Hot Potatoes in primis). La scelta di JClic come strumento principe per la realizzazione di “learning objects” non è esclusiva, cioè non esclude altre possibilità e altri software: anzi, l’uso combinato di più applicazioni diverse molto spesso permette di raggiungere risultati di eccellenza, inarrivabili utilizzando singoli programmi.

Al di là di queste considerazioni, numerose ragioni ci spingono a preferire senz’altro JClic rispetto ai concorrenti software autore; in particolare è possibile: disporre di una versione “portatile” molto stabile completamente in italiano del programma; organizzare i progetti in sequenze ordinate di attività; importare e gestire in un’unica libreria multimediale le risorse grafiche, audio e video; compilare numerosi descrittori; inserire diversi “skin” (integrabili) per l’interfaccia in esecuzione; distribuire i progetti in pacchetti per Windows (anche disciplinari), senza bisogno di collegamento Internet; rendere eseguibile un progetto, senza avere Java installato, su qualsiasi sistema Windows (98/ME/XP/Vista/7/8/10); ottenere rapporti e resoconti sulle attività svolte dagli alunni; usare, durante lo svolgimento degli esercizi, 3 tipi di contatori (cronometro, tentativi, punteggio), attivare gli aiuti, visualizzare/nascondere le frecce di navigazione, scegliere tra due diverse modalità di interazione (due clic o trascinamento); definire in modo distinto lo stile della finestra principale, della finestra di lavoro e dei pannelli delle attività; importare, esportare, copiare, incollare, duplicare, eliminare, spostare e modificare le attività inserite; salvare i progetti, con tutte le risorse necessarie, in un unico file .jclic.zip; definire e gestire i messaggi iniziali, finali e d’errore, utilizzando anche immagini e suoni già pronti; lavorare direttamente sull’albero del documento; creare una pagina web (.html) per accogliere un progetto creato (in formato .jclic.zip); visualizzare immediatamente l’anteprima del progetto; modificare direttamente (all’interno dell’archivio .jclic.zip) il file .jclic in formato XML creato dal programma; realizzare attività vivaci e molto diverse tra loro (associazioni complesse e semplici, memory, esplorazioni, identificazioni, schermate informative, puzzle doppi, a scambio e a buchi, scrittura di risposte, cruciverba, crucipuzzle, attività testuali di completamento, riempimento, identificazione e ordinamento) in modo semplice e veloce, usando comode griglie e spazi già predisposti; convertire attività compatibili (es. puzzle a scambio e puzzle doppi); usufruire di una funzione “ritaglio” particolarmente utile per la preparazione di attività interattive complesse e gradevoli; utilizzare in modo flessibile le griglie e le varie funzioni del programma per ideare esercizi nuovi; prelevare e modificare migliaia di progetti già pronti, molti dei quali in lingua italiana, distribuiti con la stessa licenza open source del programma.

I limiti sono relativamente pochi e si possono individuare nei seguenti punti: la versione originale con installer talora va in blocco (ma la versione “portatile” è stabilissima); manca di un generatore automatico di schemi per crucipuzzle e cruciverba; i files .avi e .swf non sono effettivamente supportati, anche se viene dichiarato il contrario, e per inserire animazioni bisogna ricorrere a .gif animate; manca uno strumento pensato specificatamente per la realizzazione di test a scelta multipla (anche se si possono realizzare ugualmente, ad es. con l’attività “Identificare celle”).

In linea generale e operando con una certa approssimazione, possiamo dire che JClic è rivolto maggiormente agli alunni della scuola dell’infanzia, della scuola primaria e della scuola secondaria di primo grado e privilegia le attività più visuali (puzzle, memory, abbinamenti immagini-parole, cruciverba e crucipuzzle con immagini), rispetto a quelle prettamente linguistiche (cloze, ordinamenti,

identificazione di parole). In JClic il continuo uso di immagini e suoni (nei messaggi, nelle finestre e nei pannelli) è previsto e incoraggiato e numerose opzioni regolano l'inserimento, la posizione e il comportamento di tali risorse all'interno di tutte le attività previste dal programma.



Fig.3 – Il software autore JClic

JClic: descrizione, riferimenti, links, distribuzioni, struttura, installazione

JClic (clic.xtec.cat/en/jclic/download.htm), pacchetto *open source* distribuito con licenza GNU GPL e basato su standard aperti (come l'XML), è stato realizzato da *Francesc Busquets*, con la collaborazione di Mònica Grau, Fernando Cuartero e Albert Llastarri, per conto del Ministero dell'educazione del governo autonomo della Catalogna (*Departament d'Educació de la Generalitat de Catalunya*). Il programma, disponibile attualmente nella versione 0.2.3.4, rappresenta l'evoluzione del programma *Clic 3.0*, creato sempre da Francesc Busquets nel 1992. L'utilizzo intensivo di questi strumenti da parte degli insegnanti in Spagna, ma anche in Francia, Germania e Gran Bretagna, ha permesso a Clic 3.0 prima e JClic dopo di essere al centro di un progetto collaborativo vivo e produttivo. Sono state realizzate migliaia di unità didattiche soprattutto in lingua spagnola (castigliano), catalano, basco e gallego, ma anche in francese, inglese, italiano e tedesco. Se già nella versione 0.2.1.0 era stato previsto un inserimento più semplice e diretto delle risorse multimediali all'interno di tutte le attività e una facile importazione dei files preparati con Clic 3.0, con l'ultimo aggiornamento (29/10/2013), il programma ha raggiunto la sua maturità, correggendo numerosi bug.

Attualmente JClic è distribuito in diversi modi:

- 1) con installer per Windows (clic.xtec.cat/dist/jclic/jclic-0.2.3.4.exe);
- 2) in formato Java per tutti i sistemi operativi (clic.xtec.cat/dist/jclic/jclic-0.2.3.4.zip);
- 3) come "portatile" per Windows 98/Xp/Vista/7/8/10 creato dagli esperti di Winpenpack e perfezionato da chi scrive (www.didattica.org/ccount/click.php?id=257). Quest'ultima versione è in italiano (traduzione dall'inglese ad opera dello scrivente e di Roberto Marcolin), è assolutamente stabile e si avvia anche in sistemi Windows senza Java; l'archivio deve essere scompattato in una cartella vuota e poi si deve avviare X-JClicAuthor.exe nella directory principale.

Questa versione "portable" si trova nel sito [didattica.org](http://www.didattica.org) (www.didattica.org/clic.htm), insieme a numerose risorse riguardanti JClic, mentre le altre due versioni del pacchetto sono prelevabili dal sito di clicZone (clic.xtec.net/en/jclic/index.htm), disponibile attualmente in spagnolo, catalano e inglese, e composto dalle seguenti sezioni:

- 1) JClic (caratteristiche, installazione, progetto);
- 2) Clic 3.0 (caratteristiche, scaricamento e installazione);
- 3) attività (ricerca, collaborazioni, informazioni);
- 4) comunità (mailing lists, collegamenti, mailbox);
- 5) documenti (articoli, corsi);
- 6) supporto (FAQ, segnalazione di bugs);
- 7) strumenti (utilities, altro);

- 8) ricerche (mappa del sito, attività);
- 9) selezione lingua (catalano, spagnolo, inglese). Le parti più interessanti sono ovviamente l'archivio con oltre 1300 progetti già pronti (clic.xtec.cat/db/listact_en.jsp) e la sezione della documentazione (clic.xtec.cat/en/documents/index.htm).

Abbiamo parlato di “pacchetto” perché JClic è effettivamente composto da quattro moduli:

- 1) JClic Author, il software autore per la creazione, modifica e pubblicazione dei progetti JClic;
- 2) JClic Player, per eseguire un progetto di JClic off-line;
- 3) JClic Applet, per utilizzare i nostri lavori in una pagina web;
- 4) JClic Reports, per la raccolta e la memorizzazione delle informazioni sui risultati degli utenti.

Si noti che:

- 1) JClic Reports non è necessario per il funzionamento degli altri moduli;
- 2) le pagine web create da JClic Author fanno riferimento all'Applet JClic che si trova sui server di clicZone (clic.xtec.cat/db/jclicApplet.jsp);
- 3) JClic Player non sarà analizzato in modo specifico, dato che è stato inserito all'interno di interfacce apposite per la distribuzione dei progetti JClic, le quali invece saranno descritte in seguito nei particolari.

JClic non veniva aggiornato da gennaio del 2010; del resto non erano emersi "bugs" gravi e per un programma autore destinato agli insegnanti è positivo che l'aspetto e le procedure siano stabili nel tempo. Le ultime (recenti) versioni si sono limitate a correggere diversi piccoli bug, senza modificare le caratteristiche, le funzioni e l'interfaccia del programma. All'interno del pacchetto un'importanza particolare riveste ovviamente il modulo JClic Author. Se usiamo la versione “portatile” (www.didattica.org/ccount/click.php?id=257), consigliabile per la sua solidità, sarà opportuno creare sul desktop un collegamento a questo programma (X-JClicAuthor.exe presente nella directory principale), che è l'unico veramente necessario per la produzione di oggetti di apprendimento. Per il resto non dobbiamo fare altre operazioni, ma probabilmente è utile sapere che:

- 1) i files .exe presenti nella directory principale (X-JClicAuthor.exe, X-JClic.exe e X-JClicReports.exe) richiamano i corrispondenti files .jar (Java) posti nella sottodirectory "Bin\JClic";
- 2) nella sottodirectory “Lib” la cartella “7za” contiene il compressore in .jclic.zip usato dal programma, mentre la cartella “Java” presenta una versione Java per Windows 98 (ma funzionante senza conflitti o problemi anche con Windows Xp/Vista/7);
- 3) la sottodirectory “ReadMe” raccoglie alcuni files informativi in formato .txt;
- 4) ultima, ma non meno importante, la sottodirectory “User\JClic” contiene tutti i nostri progetti (in cartelle separate) e piccole immagini .gif e .jpg (elaborate e distribuite dallo scrivente) che possiamo usare nei messaggi (iniziali, finali, di errore), ma anche nelle caselle di qualsiasi esercizio (inserendo uno sfondo bianco per avere una migliore resa visiva). Se si usa la versione standard di JClic (clic.xtec.cat/dist/jclic/jclic-0.2.3.4.exe) o quella Java (clic.xtec.cat/dist/jclic/jclic-0.2.3.4.zip), le immagini per i messaggi possono essere aggiunte scaricandole dall'indirizzo www.didattica.org/ccount/click.php?id=230 e scompattandole nella sottocartella “User\JClic”.



Fig.4 – Il sito di riferimento di JClic (versione inglese)

Gestione e distribuzione dei progetti JClic

Per la gestione dei lavori, è buona norma (e il programma lo suggerisce) assegnare una cartella a ogni progetto: anche quando si modifica un file .jcllic.zip già pronto è opportuno creare per esso una cartella di lavoro apposita. Un altro accorgimento importante è l'assegnazione di nomi significativi ai progetti, adoperando caratteri solo minuscoli e il segno “_” (sottolineato) per separare le varie parole (es. italia_fisica). Infine è sempre consigliabile salvare spesso il proprio lavoro, ma soprattutto realizzare copie di sicurezza del file .jcllic.zip in altre posizioni dell'hard disk o su una pen-disk. Ma perché questa seconda soluzione sarebbe migliore? Le esperienze in laboratorio hanno dimostrato che la causa più frequente di danneggiamento dei files di dati è la mancanza di un file di risorse proprio durante la fase di salvataggio. L'unico modo quindi per salvaguardare con certezza il proprio lavoro è realizzare copie di sicurezza con una certa frequenza: in caso di crash del programma (e conseguente danneggiamento del file di lavoro) avremo una copia recente del progetto da cui ripartire.

L'interfaccia di JClic, pulita e funzionale, comprende una barra dei menu (in alto) e quattro distinte aree di lavoro, Progetto (in primo piano in avvio), Libreria Multimediale, Attività e Sequenze. Ma, al di là della descrizione particolareggiata del funzionamento del programma e delle attività/sequenze realizzabili, argomenti che potranno essere oggetto di successivi articoli, possiamo segnalare in questa occasione la possibilità di modificare direttamente sia il file .jcllic sia l'archivio .jcllic.zip di ogni progetto. Operare su questi files ci permette in determinate situazioni di risparmiare tempi notevoli. Ad esempio, se convertiamo tutti i files .gif di un progetto in formato .jpg, non è necessario importarli tutti nella libreria multimediale e quindi cambiare uno per uno ogni collegamento all'interno delle varie attività; sarà molto più veloce:

- 1) eliminare tutti i files .gif dall'archivio .jcllic.zip e sostituirli con i corrispondenti files .jpg;
- 2) estrarre dallo stesso archivio il file .jcllic e modificarlo in un editor ASCII, sostituendo tutte le stringhe di testo “.gif” con la stringa “.jpg”;
- 3) salvare il file .jcllic modificato e sostituire il file originale presente nell'archivio .jcllic.zip.



Fig.5 – La pagina dedicata a JClic nel sito didattica.org

Naturalmente:

- 1) queste operazioni possono essere effettuate se il file .jcllic.zip non è aperto correntemente da JClic Author;
- 2) prima di modificare .jcllic.zip crearne una copia di salvataggio. Si noti che il file .jcllic è in formato “aperto” XML e ogni sua parte può essere modificata.

Infine l'Interfaccia vuota per JClic (www.didattica.org/ccount/click.php?id=234), preparata dallo scrivente in linguaggio Delphi, può essere utile per preparare pacchetti personalizzati (come i sette corposi pacchetti disciplinari presenti in www.didattica.org/clic.htm) e per distribuire i propri progetti JClic. Basta mettere i progetti (nel formato .jcllic.zip) nella directory “jcllic”. Per il resto è possibile anche inserire i files .jcllic.zip in una pagina web e fare riferimento all'applet JClic come indicato in precedenza. Questa possibilità è importante quando si utilizzano a scuola sistemi diversi da Windows.

Conclusione

Il software, relativamente poco conosciuto in Italia, rappresenta uno strumento eccezionalmente flessibile e veramente semplice da usare. Nei prossimi contributi, che saranno disponibili negli articoli successivi della rivista, saranno illustrate le funzioni di JClic e le attività che è possibile costruire. Un ultimo contributo illustrerà le configurazioni e procedure avanzate che è possibile utilizzare all'interno del software.

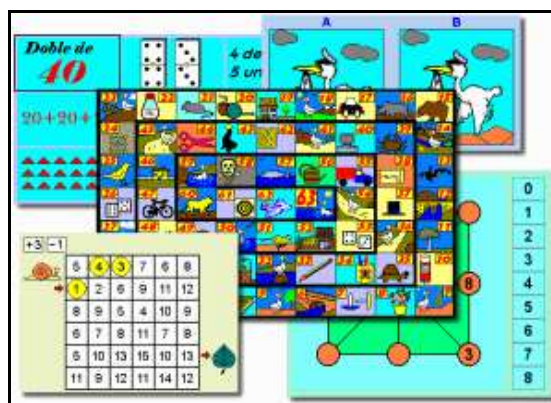


Fig.6 – Un gioco matematico realizzato con JClic

Deposito dei materiali dell'attività

Al seguente link sono depositati eventuali materiali inerenti questo l'articolo. Questi materiali nel tempo potranno essere modificati e arricchiti seguendo l'evoluzione delle idee sottostanti o/e future sperimentazioni svolte dall'autore dell'articolo.

<http://www.edimast.it/J/20150101/01230130MU/>



Giorgio Musilli

Istituto Comprensivo di Marina di Cerveteri – Cerveteri (RM)
Via Tarquinio Prisco, 52, 00052 Cerveteri (RM)
giomu2@yahoo.com
Italy

Insegnante di scuola primaria a tempo indeterminato. Programmatore Basic, Visual Basic, Pascal, Delphi e Flash. Esperto di programmi didattici ed in particolare di software autore. Ha realizzato numerosi software didattici e si occupa della raccolta e distribuzione di progetti JClic e Didapages.

Cura corsi di aggiornamento per scuole e docenti ed è stato relatore in diverse manifestazioni in tutta Italia (Crema, Cuneo, Bologna, Roma, Napoli, ecc.). È autore dei testi "I software autore per la didattica – Percorsi creativi nella scuola primaria" e "Grande cassetta degli attrezzi – I software freeware per la didattica".

Website: www.didattica.org

Pagina Facebook: <https://www.facebook.com/giorgio.musilli>

Received June 14, 2015; revised July 23, 2015; accepted July 27, 2015

Open Access This paper is distributed under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0)

