TEMA

La programmazione “from Scratch”

Alberto Barbero

I.I.S. “G.Vallauri”

alberto.barbero@vallauri.edu

Introduzione

Con l’avvio, nel 2010, del riordino scolastico nelle secondarie superiori di tutto il paese sono state introdotte molte novità nei programmi scolastici e tra queste di sicuro spicca la proposizione di una nuova disciplina chiamata Tecnologie informatiche in tutte le prime degli istituti tecnici (ma lo stesso dicasi per la disciplina Informatica nei licei scientifici opzione scienze applicate) che nelle linee guida ministeriali prevede l’obiettivo prioritario di far acquisire allo studente le seguenti competenze di base:

- individuare le strategie appropriate per la soluzione di problemi;
- analizzare dati e interpretarli sviluppando deduzioni e ragionamenti sugli stessi anche con l’ausilio di rappresentazioni grafiche, usando consapevolmente gli strumenti di calcolo e le potenzialità offerte da applicazioni specifiche di tipo informatico;
- essere consapevole delle potenzialità e dei limiti delle tecnologie nel contesto culturale e sociale in cui vengono applicate.

Tali competenze vengono poi articolare in conoscenze ed abilità, per fornire ai docenti della materia indicazioni di massima sul curriculum da progettare in dipartimento e da sviluppare in classe. In particolare, per la prima competenza che fa specificatamente riferimento al tema del problem solving, conoscenze ed abilità vengono dettagliate nel seguente modo:

Conoscenze
- Concetto di algoritmo.
- Fasi risolutive di un problema e loro rappresentazione.
- Fondamenti di programmazione.

Abilità
- Raccogliere, organizzare e rappresentare informazioni.
- Impostare e risolvere problemi utilizzando un linguaggio di programmazione

Viene quindi esplicitamente richiesto al docente di introdurre già nella classe prima concetti di logica algoritmica e di programmazione di base, correlati ad attività di laboratorio in cui si affrontino le problematiche legate all'utilizzo di un linguaggio di programmazione per la risoluzione di problemi ed il trattamento dei relativi dati. Quale migliore occasione per dare finalmente dignità ad una disciplina quale l'Informatica, il più delle volte non valorizzata o relegata ad un insegnamento puramente addestrativo all'uso del computer o, ancora, trasformata in ore di altra disciplina a seconda del docente incaricato all'insegnamento?

Il problema del curriculum

Compito dei docenti in questi anni è stato quindi quello di progettare e sperimentare un curriculo che, partendo dai contenuti esplicitati nelle linee guida, fosse in grado di coniugare le conoscenze ed abilità indicate in un percorso che non risultasse troppo ostico per degli studenti in età adolescenziale e che non banalizzasse concetti basilari per lo sviluppo di capacità di *computational thinking* utili in tutti i contesti di studio ed oltre. Dalle varie esperienze raccolte in giro per le scuole italiane e dall'analisi dei libri di testo pubblicati negli ultimi anni, sembrerebbe che il problema del curriculo sia stato risolto anticipando parte di quelli che erano i contenuti della disciplina Informatica per la classe terza dell'in-

dirizzo informatico, estrapolandone i concetti teorici di partenza che costituiscono i fondamenti della programmazione, tralasciando formalismi e ragionamenti più complessi che potrebbero dimostrarsi troppo ostici e non appropriati al momento.

Pertanto, un possibile percorso di un modulo didattico dal titolo "Problemi ed algoritmi" della disciplina Tecnologie informatiche per la classe prima potrebbe prevedere i seguenti contenuti:

- concetto di informazione e sua rappresentazione;
- analisi dettagliata di un problema;
- definizione dei passi fondamentali da compiere per giungere alla sua soluzione;
- definizione di algoritmo e sue caratteristiche;
- rappresentazione formale degli algoritmi;
- strutture di controllo fondamentali: sequenziali, condizionali, iterative;
- realizzazione di semplici programmi per mezzo di un linguaggio di programmazione.

Dalla teoria alla pratica

In questo nuovo scenario diventa di fondamentale importanza, per il pieno raggiungimento degli obiettivi disciplinari, utilizzare strumenti che siano in grado di coniugare le conoscenze e le abilità indicate in un percorso che non risulti troppo ostico per studenti in età adolescenziale e che allo stesso tempo non banalizzi concetti basilari per lo sviluppo di capacità di *problem solving* utili in tutti i contesti di studio del quinquennio ed oltre.

Nei primi anni di attuazione della riforma, alcuni strumenti paiono particolarmente adatti per il raggiungimento degli obiettivi precedentemente enunciati con modalità vicine alla sensibilità della generazione dei nativi digitali. Uno di questi è sicuramente rappresentato dal linguaggi a blocchi *Scratch* (vedi Fig. 1), sviluppato nel 2007 dai ricercatori del *Lifelong Kindergarten Group* dell'M.I.T. MediaLab di Boston guidati dal prof. Mitchel Resnick.

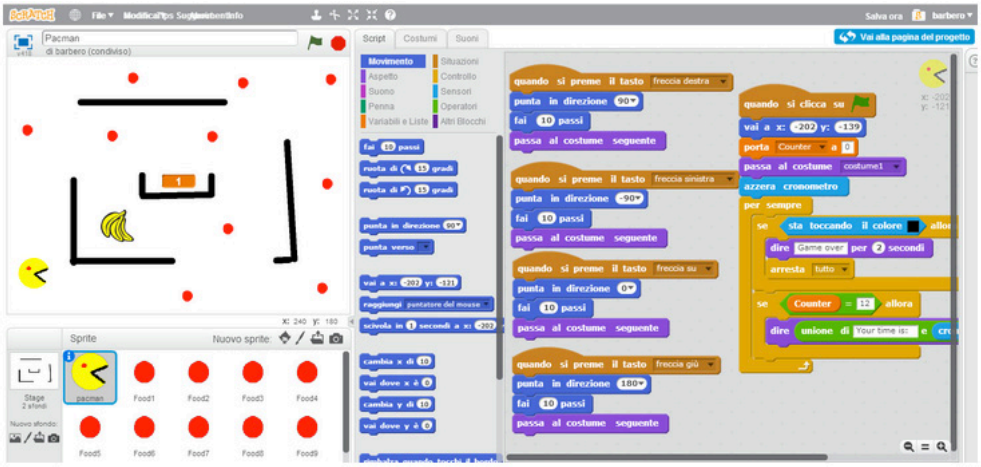

Figura 1 – L'ambiente di lavoro di Scratch.

Le caratteristiche principali di funzionamento sembrano essere tutte quelle che fanno al nostro caso. Si tratta infatti di un linguaggio creato proprio con l'obiettivo di introdurre a studenti nella fascia dell'obbligo scolastico i concetti di base della programmazione e del *computational thinking* attraverso uno strumento che ad un primo approccio sa molto di ludico ma che in realtà esercita i discenti alla logica e al ragionamento. Infatti permette di elaborare variabili e liste di valori, offre controlli per la selezione e l'iterazione, permette di realizzare animazioni via via più complesse, offre la possibilità di far eseguire più processi contemporaneamente e di farli interagire, permette l'implementazione del paradigma ad eventi, permette la gestione della messaggistica tra processi e molto altro ancora.

La codifica del programma avviene impilando blocchi di forma e colore diverso, a seconda della funzione e della categoria di appartenenza, che vanno ad incastrarsi (vedi Fig. 2) come nel gioco del Lego. Scratch (giunto alla versione 2.0) inoltre è un prodotto *free*, e quindi scaricabile gratuitamente dal sito http://scratch.mit.edu, oppure è utilizzabile attraverso la sua I.D.E. web based, stabile, potente (ogni giorno gli studenti scoprono qualche nuova funzionalità), duttile e ideale per lo sviluppo di applicazioni ludiche, animazioni grafiche, ipertesti ma anche per l'implementazione delle classiche applicazioni che si sviluppano quando si impara la programmazione da zero.

Il sito di Scratch, in pieno stile web 2.0, è diventato il punto di riferi-

mento di una vera e propria comunità virtuale che può trovare e condividere manuali gratuiti, gallerie di progetti, materiali informativi, video esplicativi, forum di discussione e permette di scaricare più di 5 milioni e mezzo (al 1 giugno 2014) di progetti completamente gratuiti con licenza *Creative Commons* o di caricare i propri progetti condividendoli con gli altri utenti sparsi per il mondo.

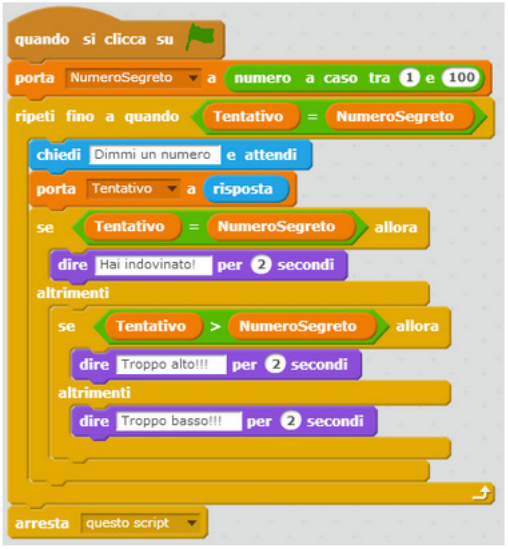

Figura 2 – Il programma "Indovina il numero segreto".

Il gatto e l'arte della programmazione

L'elemento base di Scratch è lo *sprite*, un oggetto che viene collocato all'interno di un'area denominata *stage* all'interno della quale può interagire con altri sprite e può eseguire azioni descritte in uno o più *script*, il vero e proprio codice associato ad ogni sprite formato da *blocchi* impilati.

Uno script viene mandato in esecuzione in seguito al verificarsi di un determinato evento (ad es. *Quando si clicca sulla bandiera verde, Quando si preme un tasto, Quando si riceve un messaggio*); i blocchi grafici che compongono lo script sono "mattoncini" di colore e forma diversi a seconda delle azioni che eseguono e che si incastrano a formare suggestive pile colorate. Il collegamento tra blocchi può avvenire solo secondo modalità predeterminate: ad esempio, sopra un blocco della categoria "Cappello", alla quale appartiene *Quando si clicca sulla bandiera verde*,

non è possibile incastrare nessun altro blocco. In questo modo la scrittura dei programmi risulta semplificata, perché vengono completamente eliminati gli errori di tipo sintattico.

Il livello di partenza di Scratch è facilmente accessibile anche a ragazzi con poche nozioni di informatica, ma è anche possibile creare progetti di notevole complessità e, soprattutto, da un punto di vista didattico è spesso possibile coniugare l'aspetto ludico al rigore tecnico–scientifico. Ad esempio (vedi Fig. 3) il funzionamento di un ciclo può essere spiegato facendo muovere lo sprite della stella marina due passi alla volta fino a quando non tocca lo sprite della vongola: a quel punto la stella marina saluta.

Figura 3 – Implementazione di un ciclo.

Nell'esempio seguente, il problema del gioco "Indovina il numero segreto" (già visto precedentemente) può essere implementato (vedi Fig. 4) facendo interagire due processi associati a due diversi sprite, lo sprite rappresentato dal gatto che chiede di indovinare il numero e lo sprite rappresentato dal cane che prova ad indovinare, attraverso lo scambio di messaggi. I due sprite interagiscono mediante i comandi *Invia a tutti e attendi* e *Quando ricevo*, rendendo l'esercitazione più interessante e introducendo il tema della comunicazione tra processi.

Nello script associato al primo sprite viene scelto in modo casuale il numero da indovinare e attraverso il blocco *Invia a tutti e attendi* viene sollecitato il secondo sprite a rispondere con un tentativo; nello script del secondo sprite la ricerca del numero da indovinare viene implementata con una tecnica dicotomica; quando il secondo sprite termina l'esecu-

zione dello script associato alla ricezione del messaggio, il primo sprite confronta il valore da indovinare con il tentativo e visualizza l'esito della comparazione.

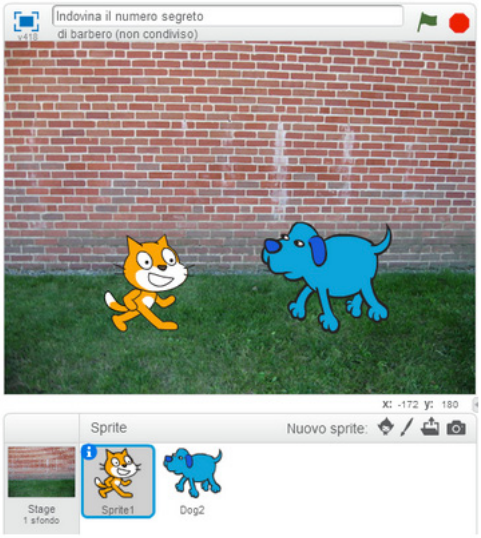

Figura 4a – Gli sprite del programma "Indovina il numero segreto".

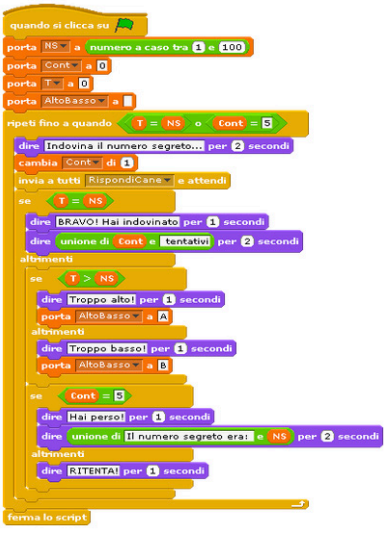

Figura 4b – Lo script associato allo sprite del gatto.

Figura 4c – Lo script associato allo sprite del cane.

Fantasia in cattedra

Tutto risolto? Sembrerebbe di sì. Ma come spesso capita, "avere una bici" non significa per forza "essere capaci a pedalare". Infatti, uno dei principali problemi su cui si sta ragionando è quello di inquadrare la didattica del modulo che abbiamo chiamato "Problemi ed algoritmi" in un'ottica di rinnovamento ed innovazione, partendo dal presupposto che non è possibile introdurre la programmazione nelle classi prime con le stesse modalità che viene seguita nelle classi terze dell'indirizzo informatico, anche per il ridotto monte ore a disposizione.

Un aiuto ci viene come sempre scartabellando il materiale informativo che si trova sul sito di Scratch e scaricando i numerosi progetti condivisi sulla rete, dove è possibile trovare suggerimenti ed idee per affrontare temi in modo non sempre del tutto canonico, come già visto negli esempi precedenti. Ad esempio, è possibile introdurre il concetto di contatore (vedi Fig. 5) attraverso la progettazione e realizzazione di un programma per la gestione di un semplice quiz dove ad inizio elaborazione il contatore viene posto a zero e ad ogni risposta corretta viene incrementato di un'unità.

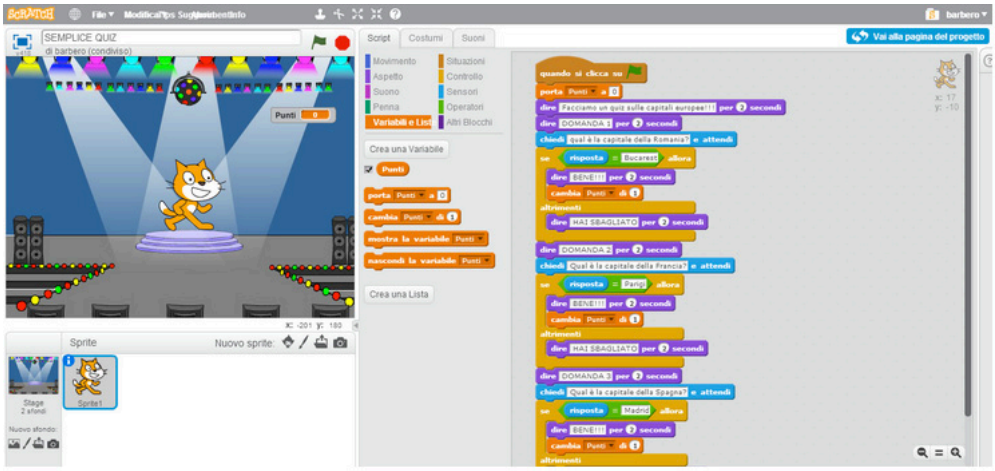

Figura 5 – Il programma "Semplice quiz".

Da queste idee possono nascerne altre per modulare la didattica coniugando, quando possibile, aspetti ludici al rigore scientifico. Lo sforzo del docente deve essere quindi quello di affiancare alle classiche esercitazioni iniziali la sperimentazione di nuove metodologie di proposizione di problemi che sfruttino le potenzialità offerte da Scratch per introdurre concetti anche non banali senza perdere l'interesse della classe.

Una scommessa non facile che richiede al docente di rimettersi in gioco per appassionare una platea non facile e per non buttare via un'occasione per dimostrare capacità e professionalità.