DIDATTICA INFORMATICA 2018 Best Paper Award

codOWood - un nuovo modo di programmare

Alberto Ferrari, Giulio Angiani, and Michele Tomaiuolo

Dipartimento di Ingegneria e Architettura, Università di Parma
{alberto.ferrari,giulio.angiani,michele.tomaiuolo}@unipr.it

Abstract. Il progetto codOWood è stato sviluppato a livello prototipale dal laboratorio di ricerca SoWIDE [1] dell'Università di Parma. Si tratta di una proposta didattica pensata per l'introduzione al coding in ambito di scuola primaria basata su un ambiente di programmazione tangible. Il progetto utilizza tecnologie a basso costo ed è strutturato in modo da favorire il lavoro collaborativo.Le varie funzionalità sono state testate a livello prototipale fornendo risultati più che soddisfacenti. Il progetto di Tangible Computer Programming prevede blocchi passivi di materiale povero (legno) che vengono individuati mediante image recognition e "trasformati" in blocchi logici per applicazioni di Block Programming basate su Google Blockly. La sperimentazione portata avanti in tre classi di scuola primaria ha mostrato chiaramente che questa metodologia avvicina i bambini all'arte della programmazione più di quanto lo faccia quella solamente digitale. In più ha evidenziato degli ottimi risultati in termini di aumento della collaborazione nel lavoro di gruppo, della condivisione delle idee e dell'inclusione dei componenti del gruppo stesso.

Keywords: Coding; block programming; tangible computer programming; image recognition; education; computational thinking.

1 Introduzione

I classici linguaggi di programmazione testuali sono in genere pensati per programmatori professionali e sono costituiti da termini e simboli spesso difficili da comprendere per i bambini. Le difficoltà di natura sintattica spesso impegnano i giovani programmatori che invece dovrebbero concentrarsi soprattutto sul procedimento di soluzione dei problemi.
A differenza di questi linguaggi in cui i programmi sono costituiti da sequenze di linee di testo, nel block programming le istruzioni sono rappresentate da blocchi che vengono collegati fra loro a formare un puzzle che rappresenta il programma. Il Block Programming è il modo più semplice e immediato per avvicinarsi al mondo del coding e infatti è alla base della maggior parte dei progetti indirizzati ad un pubblico di giovani e giovanissimi. Dal punto di vista didattico, la programmazione a blocchi consente di apprendere le basi del coding, aiuta a sviluppare la logica, stimola la creatività ed educa al pensiero computazionale, a ragionare sui problemi e sul modo migliore per risolverli.

[1] http://sowide.ce.unipr.it/

Scratch o Scratch Jr. per i più piccoli sono un esempio di strumenti divertenti e lo stesso si può dire dei problemi presenti nel sito code.org. Sono giochi, ma sono anche sfide a risolvere problemi più o meno complessi la cui soluzione è un vero e proprio programma anche se non sono visibili le righe di codice ma un insieme di blocchi colorati e di varia forma collegati fra loro a formare un puzzle. Si parla in questo caso di Graphical User Interfaces (GUI) poiché i semplici ambienti di sviluppo educativi si basano sul drag & drop di blocchi virtuali che vengono selezionati e disposti in modo da formare il puzzle-programma.
Se i blocchi-istruzione non sono più virtuali ma diventano oggetti fisici la programmazione avviene manipolandoli, spostandoli e incastrandoli in un ambiente reale, in questo caso si parla allora di Tangible User Interfaces e di Tangible Computer Programming.
Il nostro progetto si colloca proprio in questo segmento facendo utilizzare elementi fisici di legno ai bambini che si accostano per la prima volta alla programmazione.

2 L'ambiente didattico di codOWood

2.1 Aule aumentate

L'ambiente didattico proposto per l'utilizzo di codOWood è caratterizzato da un'aula aumentata dalle tecnologie in cui è presente una postazione centralizzata di condivisione, e una serie di spazi in cui operano gli alunni suddivisi in piccoli gruppi (figura 1).
La struttura e le modalità di interazione fra i gruppi favoriscono l'apprendimento laboratoriale pur non necessitando di costose apparecchiature elettroniche e di strutture particolari. L'insieme delle tecnologie proposte è di basso costo e facilmente reperibile e il software è semplice da utilizzare; questi fattori sono importanti perché rendono il sistema sostenibile per la maggior parte delle realtà didattiche. La postazione di condivisione è gestita dal docente che propone alla

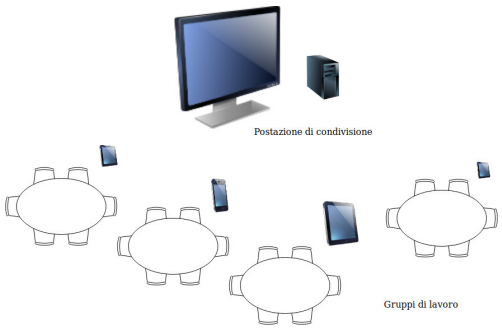

Fig. 1. Ambiente didattico.

classe un problema scelto fra un insieme di temi in base al livello precedentemente raggiunto e alle competenze logiche degli studenti. Il problema consiste nel definire la sequenza di azioni che devono compiere personaggi che si muovono all'interno di un ambiente virtuale, in figura 2 vediamo un classico esempio di labirinto. L'ambiente e i personaggi sono visualizzati su uno schermo condiviso (LIM o videoproiettore) della postazione docente. Sempre sullo schermo condiviso verranno poi visualizzate le soluzioni proposte dai vari gruppi di studenti.
Le postazioni per gli alunni prevedono una superficie su cui disporre e collegare fra loro i blocchi fisici e un dispositivo per acquisire l'immagine del puzzle-programma e per comunicarla alla postazione di condivisione; in figura 3 vediamo un esempio di algoritmo formato da blocchi concatenati.

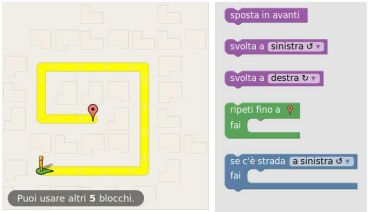

Fig. 2. Un semplice esempio di problema.

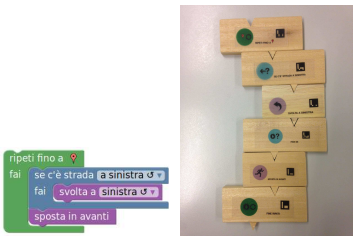

Fig. 3. Un esempio di algoritmo.

2.2 Postazione di condivisione

Uno degli obiettivi che ci siamo posti già in fase di progettazione è stato il contenimento dei costi e la possibilità di utilizzare strumentazione spesso già presente negli ambienti scolastici. Per questa ragione l'implementazione di tutte le funzionalità della postazione di condivisione è stata effettuata su un single-board computer Raspberry Pi del costo complessivo di poche decine di Euro.

Questa postazione ha le funzionalità di server per l'intero sistema e si occupa della configurazione di una intranet per permettere la condivisione di informazioni con i dispositivi client utilizzati dagli alunni.
L'utilizzo di Raspberry Pi tuttavia non è necessario e il software server codOWood può essere installato su un qualsiasi computer eventualmente già presente in aula.
Per la visualizzazione dei problemi e delle varie proposte di soluzione è necessario uno schermo comune che può essere un qualsiasi monitor collegato al server, un videopriettore o una lavagna interattiva multimediale (LIM).

2.3 Postazioni alunni

Gli alunni sono suddivisi in piccoli gruppi che operano indipendentemente ma in stretto contatto con il docente. Ogni gruppo ha ha disposizione un insieme di blocchi di legno che può disporre su uno spazio di lavoro (una qualunque superficie piana) in modo da realizzare un puzzle che rappresenta il flusso operativo e l'insieme dei comandi che portano alla soluzione del problema.
Ogni gruppo utilizza uno smartphone o tablet sul quale è installata l'app client di codOWood che riproduce l'ambiente virtuale del problema e implementa varie funzionalità:

- interazione con il server per la registrazione del gruppo di lavoro
- acquisizione dell'immagine del puzzle realizzato
- riconoscimento della struttura complessiva del puzzle e di ogni singolo blocco mediante una procedura di image recognition partendo dall'immagine acquisita e successiva conversione in codice operativo codOWood
- interazione con in server per l'invio della proposta di soluzione

3 Motivazioni alla base del progetto

3.1 Coding nella scuola primaria

Il grande sforzo oggi in atto per far sì che tutti i bambini abbiano la possibilità di imparare la programmazione nasce da concetto formulato da Seymour Papert oltre 50 anni fa: *i bambini sono gli insegnanti e le macchine diventano gli allievi* [5, 4]. Papert [6] e Resnick [8] hanno dimostrato che imparare i concetti fondamentali di programmazione comporta un cambiamento nel modo di pensare delle persone.
L'obiettivo di questi primi approcci al coding non è quello di creare professionisti della programmazione ma di far acquisire ai giovani un nuovo modo di pensare e di vedere il mondo, di padroneggiare la tecnologia e non esserne soltanto dei consumatori. La programmazione non è soltanto un job skill ma una literacy fondamentale che ognuno deve imparare nel 21 secolo [7, 2, 1].
L'apprendimento dei concetti di base della programmazione è al posto al centro di un progetto più ampio che abbatte le barriere dell'informatica e stimola un approccio indirizzato alla risoluzione dei problemi. Con il coding bambini e ragazzi sviluppano il pensiero computazionale e l'attitudine a risolvere problemi più o meno complessi. Il coding favorisce il pensiero creativo e stimola la

curiosità, attraverso il coding si "dialoga" con il computer, l'obiettivo non è formare una generazione di futuri programmatori, ma educare i più piccoli al pensiero computazionale, che è la capacità di risolvere problemi applicando la logica e ragionando sulla strategia migliore per giungere alla soluzione.
Il nostro progetto è quindi pensato per bambini e bambine della scuola primaria anche se può facilmente essere rimodellato nella tipologia dei problemi e nella semantica dei blocchi per adattarlo a studenti dei primi anni di scuola secondaria.

3.2 Sostegno all'interazione di gruppo: cooperative learning

Gli ambienti di sviluppo per i linguaggi di programmazione tangible permettono a piccoli gruppi di studenti di progettare e realizzare programmi in modo collaborativo. Questo risulta più difficile con i classici ambienti di programmazione basati sostanzialmente sull'utilizzo di mouse e/o tastiera, strumenti che possono essere utilizzati da un solo programmatore alla volta.
La possibilità di interagire a più mani sui blocchi disposti su un piano di lavoro favorisce quindi il lavoro di gruppo e l'interazione faccia a faccia fra gli studenti, favorisce inoltre lo sviluppo di una leadership distribuita all'interno di ogni gruppo. Tutti questi sono aspetti importanti che ritroviamo nel concetto di cooperative learning: ogni componente del gruppo può selezionare e posizionare i vari blocchi ma l'obiettivo, il puzzle finale, è comune e quindi risulta importante la relazione e l'interazione tra studente e studente.
L'obiettivo primario è ovviamente quello di introdurre i concetti del pensiero computazionale, ma questo modo di operare in gruppo educa anche ai comportamenti sociali e alle capacità comunicative, finalità certo di non minore importanza.
È importante inoltre sottolineare la figura del docente che ha un ruolo di guida che esercita monitorando e intervenendo nei lavori dei piccoli gruppi e nel gestire la presentazione e la discussione delle varie soluzioni proposte a tutta la classe.

4 Implementazione

4.1 Server

La postazione condivisa gestita dal docente è responsabile della gestione della comunicazione con le altre componenti del sistema. Come stabilito in fase di progetto deve rispondere a una serie di requisiti che riteniamo fondamentali per una effettiva spendibilità dell'intero sistema:

- utilizzo di hardware già presente nella sede didattica o in alternativa operabilità su hardware di basso costo,
- facilità di installazione e utilizzo sia in presenza di una rete preesistente sia in assenza di questa,
- possibilità di collegamento con ogni tipo di periferica di visualizzazione.

Sul server opera una web application scritta in Python che utilizza il framework open source Pyramid. L'applicazione configura automaticamente un rete locale wireless aperta alla comunicazione con i vari dispositivi client.

Nel caso sia presente nell'aula un computer per il docente, il software è di facile installazione e non presenta particolari requisiti hardware o di sistema software. Nel caso invece in cui non sia già presente un computer o non si ritenga opportuno utilizzarlo, l'applicazione è stata installata e testata in tutte le sue funzionalità su un single-board computer Raspberry Pi con sistema operativo Raspbian. Questo hardware è facilmente reperibile e acquistabile a un costo complessivo di poche decine di euro. La scelta di Raspberry Pi è dettata, oltre che per il basso costo, dal fatto che questo sistema è stato concepito proprio con l'intento di realizzare un dispositivo economico e utile per stimolare l'insegnamento di base dell'informatica e della programmazione nelle scuole. Tramite l'uscita per la periferica video del Raspberry è possibile poi collegare un dispositivo per la visualizzazione dei problemi e delle soluzioni proposte in modo da permettere la condivisione fra il docente e i vari gruppi di studenti.

4.2 Client

Ogni gruppo di lavoro dispone di un dispositivo che deve essere in grado di ottenere un'immagine del puzzle di blocchi, di convertirla in codice codOWood e di gestire la comunicazione con il server. Anche in questo caso abbiamo stabilito, in fase di progettazione, una serie di requisiti che hanno condizionato la scelta dell'hardware e del software:

- possibilità di utilizzo di hardware già disponibile o in ogni caso di basso costo,
- estrema semplicità di utilizzo da parte degli alunni.

Lo strumento ideale per visualizzare il puzzle proposto, catturare l'immagine del puzzle-programma ed eleborarla mediante algoritmi di image recognition è probabilmente un tablet. Purtroppo questo tipo di attrezzatura spesso non è presente nelle scuole.
I requisiti hardware richiesti per la nostra applicazione (risoluzione della fotocamera e potenza del processore) sono minimi quindi sarebbe sufficiente un tablet entry level, ma visto che ogni gruppo dovrebbe utilizzare un diverso dispositivo (la condivisione di pochi dispositivi renderebbe meno interattiva l'esperienza di programmazione) il costo complessivo supererebbe i limiti che ci siamo imposti. L'applicazione è stata quindi testata e risulta funzionante su smartphone dotati di sistema operativo Android, al momento il più diffuso.
Gli smartphone non fanno normalmente parte della dotazione tecnologica delle scuole ma ormai quasi ogni famiglia (e spesso anche gli stessi alunni) dispone di dispositivi di questo tipo. Questo potrebbe essere quindi un momento per favorire l'utilizzo e l'integrazione con l'attività didattica dei propri dispositivi elettronici personali seguendo le politiche di BYOD (Bring Your Own Device) che oggi vengono proposte dai ministeri e dai vari enti che si occupano di didattica.

> "La scuola digitale, in collaborazione con le famiglie e gli enti locali, deve aprirsi al cosiddetto BYOD (Bring Your Own Device), ossia a politiche per cui l'utilizzo di dispositivi elettronici personali durante le attività didattiche sia possibile ed efficientemente integrato." [3]

5 Tecniche di riconoscimento

La scelta di utilizzare blocchi non dotati di elettronica interna rende necessario definire le procedure che permettono di riconoscere i vari tipi di blocchi e la loro collocazione all'interno del puzzle-programma.
Abbiamo testato varie soluzioni tutte basate su image recognition, il riconoscimento di oggetti (nel nostro caso i vari tipi di blocco) in base ad alcune caratteristiche quali forma, colore o la presenza di un marker simbolico. Il riconoscimento avviene su un'immagine del puzzle catturata dalla fotocamera del dispositivo e successivamente elaborata.

Requisiti generali e caratteristiche di un blocco:
- collegabilità con altri blocchi,
- possibilità di rappresentare il flusso di esecuzione del programma,
- identificabilità da parte dell'alunno,
- identificabilità da parte del SW da immagine fotografica,
- facilità di realizzazione e costo contenuto.

Presentiamo qui alcuni esempi di blocchi che abbiamo realizzato per l'esperimento.
Dopo le procedure di eventuale ridimensionamento dell'immagine questa viene

Fig. 4. Blocco con QRCode.

convertita in Bitmap Binario.
Una prima scansione restituisce una lista di valori risultanti della decodifica dei QRCode rilevati. Per ciascun marker oltre al suo codice si ottengono le coordinate di 3 angoli, partendo da quello in basso a sinistra e procedendo poi in senso orario. Questi valori sono utilizzati per ritagliare la porzione significativa dell'immagine e per riallinearla. Individuata la posizione dei QR Code

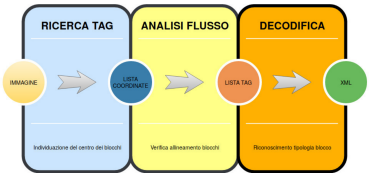

Fig. 5. Flusso di decodifica.

all'interno dell'immagine si deduce il flusso sequenziale dell'algoritmo identificando l'indentazione dei blocchi. Nel caso di indentazione corretta (chiusura di ogni blocco di selezione o iterazione) si procede alla decodifica del QR Code e alla traduzione di ogni blocco in codice XML per codOWood (figura 5).

6 Tipi di blocchi - Il linguaggio codOWood

Per le prime attività di coding, seguendo un modello presente nella maggior parte dei corsi introduttivi, abbiamo pensato a una serie di problemi a difficoltà crescente. Nel prototipo di codOWood abbiamo seguito la sequenza di problemi presente nel modulo *maze* di Blockly Games [2]. Si tratta di una serie di situazioni in cui un personaggio deve eseguire le istruzioni che gli vengono fornite per raggiungere l'uscita da un labirinto.
Per risolvere i primi livelli sono sufficienti i comandi di avanzamento e di svolta che in codOWood corrispondono ai blocchi presenti in figura 6. Nel risolvere questo primi livelli gli alunni apprendono quindi il concetto di sequenza proponendo programmi composti da una sequenza di blocchi operativi. Il passaggio

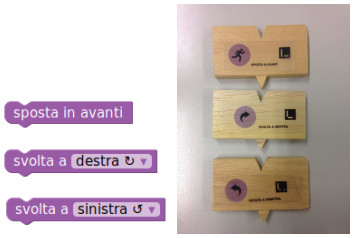

Fig. 6. Blocchi operativi di Blockly e di codOWood.

successivo è l'introduzione del concetto di ciclo tramite un blocco che definisce la ripetizione di una serie di comandi. Blockly Games propone un unico blocco di questo tipo (fig. 7) per ripetere una serie di azioni fino al raggiungimento dell'obiettivo finale. I blocchi virtuali hanno la possibilità di variare il loro aspetto e, in questo caso, il blocco si espande per poter contenere un numero variabile di altri blocchi (i blocchi interni al ciclo). Questo aspetto "mutante" non può però essere mantenuto nel caso di blocchi fisici. Il problema è stato quindi risolto introducendo un blocco di chiusura. Per rendere evidente la struttura complessiva del ciclo, oltre a colorare diversamente le icone dei vari tipi di blocchi, varia anche il punto di connessione tra un blocco e l'altro in modo da realizzare puzzle che ripropongono l'idea di indentazione comune a tutti i linguaggi di programmazione testuali (fig. 7).
Un problema analogo relativo alla "rigidità" di struttura dei blocchi fisici lo abbiamo riscontrato in relazione alle strutture di selezione del flusso di esecuzione

[2] https://blockly-games.appspot.com/maze

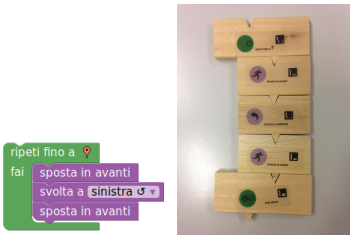

Fig. 7. Ripetizione di azioni in Blockly e in codOWood.

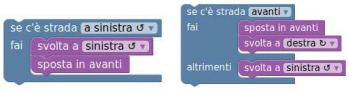

Fig. 8. Blocchi virtuali per la selezione.

che sono necessarie per risolvere gli ultimi livelli e che introducono il concetto di selezione.
In figura 8 un esempio di algoritmo con blocchi di selezione in Blockly e in figura 9 l'equivalente in codOWood.
Possiamo quindi riconoscere in codOWood tre tipologie di blocchi:

- blocchi operativi
- blocchi di ripetizione
- blocchi di selezione

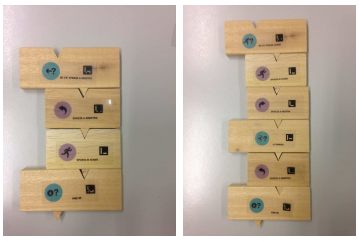

Fig. 9. Blocchi fisici per la selezione.

7 Sperimentazione

Seppur a livello prototipale codOWood è pienamente funzionante ed è in atto una sperimentazione con tre classi di scuola elementare per confrontare il comportamento di gruppi di alunni che utilizzano blocchi virtuali con altri che utilizzano i blocchi tangibili per risolvere lo stesso problema.

In fig. 10, sono mostrate alcune immagini della sperimentazione effettuata in una classe quinta di una scuola Primaria. I bambini hanno evidenziato come l'uso del tangible abbia facilitato il lavoro di gruppo e la condivisione e si siano divertiti molto più che utilizzando i dispositivi elettronici a loro disposizione. Anche dalle insegnanti è stato restituito un feedback molto positivo in merito soprattutto all'aspetto collaborativo e inclusivo dell'esperienza.

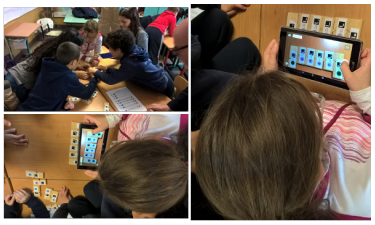

Fig. 10. Sperimentazione.

8 Sviluppi futuri

Dal lato tecnologico del riconoscimento ottico dei blocchi stiamo lavorando nella direzione di blocchi senza marker di nessun tipo in cui il riconoscimento avviene tramite l'individuazione delle forme e dei colori delle icone. Questo permetterebbe di ottenere oggetti maggiormente user friendly ma è da verificare la qualità di riconoscimento mantenendo inalterati vincoli che ci siamo imposto relativi alla qualità dell'hardware e degli aspetti logisitici (luminosità non ottimale, inquadratura non perfettamente ortogonale e allineata al puzzle).

Riferimenti bibliografici

1. A. Ferrari, G. Lombardo, M. Mordonini, A. Poggi, and M. Tomaiuolo. OOPP: Tame the Design of Simple Object-Oriented Applications with Graphical Blocks. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, Volume 233, pp. 279-288, 2018.
2. A. Ferrari, A. Poggi, and M. Tomaiuolo. Object oriented puzzle programming. *Mondo Digitale*, 15(64), 2016.
3. MIUR. Piano nazionale scuola digitale. *Ministero dell'Istruzione dell'Università e della Ricerca. Direttiva del 27 ottobre 2015, n. 851*, 2015.
4. S. Papert. Learn think to children. *L. A. relation*, 1972.
5. S. Papert. Teaching children thinking. *Programmed Learning and Educational Technology*, 9(5):245–255, 1972.
6. S. Papert. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., 1980.
7. S. Papert and C. Solomon. Twenty things to do with a computer. 1971.
8. M. Resnick, F. Martin, R. Sargent, and B. Silverman. Programmable bricks: Toys to think with. *IBM Systems journal*, 35(3.4):443–452, 1996.