

Programmare giocando con ScratchJr nella scuola dell'infanzia.

La corrispondenza quantità-numero

Playful programming with ScratchJr in Kindergarten. The quantity-number correspondence

Annarosa Serpe

Dipartimento di Matematica e Informatica – Università della Calabria, Italia

Sunto / L'articolo si colloca nel dibattito inerente la pratica del coding e pone l'accento sull'uso di un linguaggio di programmazione visuale, come ScratchJr, nella scuola dell'infanzia. A partire dai riferimenti normativi attuali emanati dal MIUR¹, l'articolo, dopo aver introdotto il quadro di riferimento teorico, presenta le peculiarità e le potenzialità educative dell'App ScratchJr. Al crocevia tra l'esame di alcune riflessioni teoriche e quello di alcune pratiche didattiche, l'articolo mostra una modalità di lavoro in sezione inerente la corrispondenza quantità-numero. La pratica didattica esposta è stata estrapolata da un percorso di ricerca-azione finalizzato al controllo scientifico delle condizioni e delle procedure di utilizzo di ScratchJr nel campo d'esperienza "La conoscenza del mondo".

Parole chiave: coding; scuola dell'infanzia; numero; gioco; quantità; ScratchJr.

Abstract / This article is part of the debate surrounding the practice of coding with special attention on the use of a visual programming language, such as ScratchJr, in Kindergarten. Starting from the current National Guidelines issued by the Italian Ministry for Education and Research (MIUR), the article, after introducing the theoretical framework, presents the peculiarities and educational potential of the ScratchJr App. Situated at the crossroad between the examination of theoretical reflections and of didactic practices, the article shows an approach to the quantity-number correspondence in classroom. The discussed teaching practice has been taken from an action research project aimed to provide validation of the requirements and procedures for introducing ScratchJr in the field of experience concerning "Knowledge of the world".

Keyword: coding; kindergarten; number; play; quantity; ScratchJr.

1 Premessa

Il Piano Nazionale Scuola Digitale (PNSD) emanato dal MIUR (Legge 107 del 15 luglio 2015 - La Buona Scuola) traccia i contorni di una nuova visione della scuola, cioè una scuola digitale basata su competenze, ricca di strumenti di co-creazione di contenuti; una scuola, dunque, che richiede una ridefinizione dei ruoli di chi insegna e di chi apprende.

Il PNSD – pilastro fondamentale della "Buona Scuola" – riserva particolare attenzione alla comprensione della tecnologia e della logica della rete sin dai primi livelli di istruzione. L'obiettivo dell'Azione#17 del PNSD, coerentemente declinato all'interno

1. Ministero dell'Istruzione, dell'Università e della Ricerca italiano.

del progetto “Programma il Futuro”,² è quello di introdurre e sostenere il pensiero computazionale nelle scuole, in particolare nella scuola primaria,³ garantendo a ciascun alunno la possibilità di avvicinarsi ai principi di base dell'informatica (*coding*) in un contesto di gioco.

Tale introduzione strumentale – denominata “Ora del Codice” – è articolata su due livelli di attività: di base e avanzato.

Nello specifico, nella C.M. del MIUR 08/10/2015 si legge:

«(...) un'appropriata educazione al “pensiero computazionale”, (...) è infatti essenziale affinché le nuove generazioni siano in grado di affrontare la società del futuro non da consumatori passivi ed ignari di tecnologie e servizi, ma da soggetti consapevoli di tutti gli aspetti in gioco e come attori attivamente partecipi del loro sviluppo».⁴

(C.M. del MIUR 08/10/2015)

In questo quadro istituzionale – ancora una volta – la scuola dell'infanzia (SdI) non viene esplicitamente menzionata, anche se è proprio in rapporto alle tecnologie che questo segmento scolastico ha dato il meglio di sé partecipando ad esperienze innovative, a reti di scuole e a progetti specifici finanziati da enti, associazioni e agenzie nazionali e internazionali (Costabile & Serpe, 2010, 2011; Mantovani & Ferri, 2006, 2008; Sbaragli, 2011).

Su questi percorsi polivalenti sono state condotte specifiche ricerche che hanno mostrato il carattere innovativo di queste realtà didattiche non solo per la formazione, ma anche per la costruzione di nuove forme di conoscenza di cui i bambini sono i principali artefici (Calvani, 2007).

A distanza di anni, questa serie di esperienze “tecnologiche” richiede una riconfigurazione e un'estensione che possa consentire di rispondere ai nuovi bisogni degli allievi e al contempo di sviluppare appieno le potenzialità della tecnologia digitale nell'ambito della formazione, integrando le istanze della personalizzazione dei nuovi processi formativi che la tecnologia abilita.

Allo stato attuale, la SdI ritaglia spazi sempre maggiori alla pratica del *coding*, cioè un nuovo tipo di alfabetizzazione che permette il raggiungimento di due obiettivi: apprendere attraverso la programmazione e sviluppare l'abitudine a risolvere problemi più o meno complessi.

In questa prospettiva, assume particolare rilevanza la formazione iniziale dei futuri insegnanti all'impiego delle tecnologie in sé e soprattutto all'utilizzo di nuovi modelli di interazione didattica con questi strumenti.

In tale direzione, il Corso di laurea in Scienze della Formazione Primaria dell'Università della Calabria ha promosso, all'interno di alcuni insegnamenti, l'uso di linguaggi di programmazione elementari. In particolare, all'interno dell'insegnamento di Didattica della matematica - Modulo 1, nell'ottica suddetta, si vuole fornire agli studenti⁵ l'opportunità di acquisire conoscenze e competenze atte a garantire un utilizzo critico del *coding* nell'insegnamento-apprendimento della matematica.

2. <https://programmailfuturo.it/>

3. In Canton Ticino questo livello scolastico è detto scuola elementare, in Italia è chiamato scuola primaria.

4. <https://www.programmailfuturo.it/media/docs/Circolare-Programma-il-Futuro-2016.pdf>

5. L'insegnamento di Didattica della matematica è disposto al III anno del corso di laurea; gli studenti sono stati alfabetizzati dal punto di vista informatico avendo già sostenuto nel I anno il Laboratorio di Tecnologie Didattiche.

In questo scenario è stato effettuato un percorso biennale (AA.AA. 2015-2017) di ricerca-azione (R.-A.) finalizzato alla progettazione, sperimentazione e controllo scientifico di pratiche didattiche sul *coding* mediante adeguati ambienti di programmazione. Questo articolo rende conto di una piccola parte del percorso che mette al centro il *coding* come una delle possibili risorse utilizzabili per favorire e sviluppare il pensiero computazionale fin dalla SdI. A partire dal quadro teorico di riferimento, l'articolo mette in evidenza le potenzialità educative del linguaggio di programmazione visuale ScratchJr. Di seguito, al crocevia tra l'esame di alcune riflessioni teoriche e alcune pratiche educative in sezione, l'articolo pone l'accento su alcuni aspetti utili e praticabili in merito a condizioni e procedure di utilizzo di ScratchJr nel campo d'esperienza "La conoscenza del mondo".

2 Matematica, infanzia e tecnologia

La matematica, o meglio la "prima matematica" nella SdI, si configura non come un sistema di conoscenze disciplinarmente organizzate, ma come un sistema simbolico che sostiene i piccoli allievi nei processi di decodificazione della vita reale, nella soluzione dei problemi, nella revisione ed integrazione delle ipotesi, nell'attribuzione dei significati (Serpe, 2008).

Considerata la natura stessa della matematica, l'accesso agli oggetti matematici non può avvenire che in maniera indiretta, tramite la mediazione di rappresentazioni semiotiche (D'Amore, 2003; Duval, 2000). Per mobilitare questo processo l'insegnante deve promuovere le motivazioni intrinseche della curiosità, della lucidità e del piacere della scoperta, che sono fattori insostituibili per capire e interpretare la realtà (Co-stabile & Serpe, 2011).

In termini di metodologia, quindi, le proposte didattiche devono collocarsi all'interno di una prospettiva che vede strettamente correlati i fenomeni socio-affettivi e le attività di apprendimento, nonché essere trasversali a tutte le aree disciplinari in modo da sostenere i diversi piani rappresentativi, manipolativo-iconico-simbolico. Di conseguenza, le attività in sezione devono privilegiare il gioco in modo da promuovere e favorire il coinvolgimento della sfera cognitiva, relazionale e affettivo-emotiva del bambino. Il gioco è costituito da diverse tappe: la definizione della consegna, la situazione d'azione, la situazione di formulazione e la situazione di validazione. Quest'ultima è molto importante sia perché sottende l'ottica cooperativa dell'apprendimento, sia perché promuove lo sviluppo dell'attitudine all'argomentazione arrivando a intuire il concetto matematico come punto di arrivo e non di partenza. Il gioco traduce così il processo dell'apprendere "facendo" in una divertente attività intellettuale e cognitiva. È mediante questi processi di costruzione dell'apprendimento individuale e di gruppo, nelle fasi di "gioco" e "dopo gioco", che emergeranno i valori formativi della personalità.

La struttura compositiva delle Indicazioni Nazionali per campi di esperienza mette al centro dell'apprendimento l'operare del bambino, la sua corporeità, le sue azioni, le sue percezioni; sotteso a ogni campo troviamo uno o più sistemi simbolico-culturali,⁶ ma anche le differenze. In ogni campo – connotato dai segni della cultura – il

6

6. Un substrato di conoscenze, linguaggi e abilità.

bambino trova il contesto per diventare via via più consapevole delle sue esperienze, perché le rielabora, le rievoca, le ricostruisce proprio grazie ai mediatori (immagini, parole, strumenti, informazioni), messi a disposizione dal campo. In particolare, il campo d'esperienza "La conoscenza del mondo" è quello dove i bambini pongono le basi per l'elaborazione di concetti scientifici e matematici che verranno proposti nel segmento scolastico successivo.

Tale campo, considerata la sua peculiarità, richiede percorsi metodologici dinamici-costruttivi atti a favorire attività cooperative capaci di mantenere vivi l'interesse e la curiosità dei bambini, nonché a stimolarne l'organizzazione, la progettazione, la ricerca e la scoperta.

L'evoluzione delle pratiche pedagogiche permette di diversificare le situazioni d'apprendimento e al contempo le tradizionali attività in sezione possono essere supportate dalla tecnologia in modo da rendere più efficace l'azione educativa quotidiana mediante strumenti interattivi che traducono l'esperienza in sistemi più efficienti di classificazione.

L'introduzione di strumenti tecnologici deve essere effettuata in maniera progressiva privilegiando la dimensione ludico-esperienziale e avere connotati di continuità con l'esperienza quotidiana (Costabile & Serpe, 2010).

Tale passaggio, parafrasando l'autorevole pensiero di Seymour Papert (1993), non deve essere lineare e "sequenziale", ma deve integrarsi in maniera graduale nelle diverse fasi dello sviluppo. Un'integrazione di natura non lineare tra l'esperienza frutto della dimensione ludica e le attività riflessive e metacognitive, secondo Donald Norman (1993, 2004), dovrebbe integrare *experimental learning* e *reflexive learning*. Pertanto, il substrato e il riferimento teorico non possono che essere trovati sia nella pedagogia post-piagetiana sia nelle teorizzazioni, ispirate da Vygotskij (2010), di quelle che sono state definite la "pedagogia della complessità" di Howard Gardner (1993, 2011) e la "psicologia culturale" di Jerome Bruner (1990, 2009).

Lo strumento tecnologico si deve, quindi, armonizzare con le differenti fasi dello sviluppo psicologico e senso-motorio del bambino e con il rispetto dei tempi e della soggettività; inoltre, si deve sfruttare appieno la strutturale multimedialità⁷ mediante la progettazione di percorsi didattici e *Mind Tools* (Jonassen, 2000) in modo da valorizzare e accrescere e valorizzare in senso critico tutte le possibili abilità e forme d'intelligenza, in primis quella relazionale e comunicativa. In ultimo, ma non meno importante, l'uso della tecnologia deve essere fruibile nei tre aspetti rilevanti della finalizzazione, del carattere situato e della co-progettazione (Ferri & Mantovani, 2006). Nel campo d'esperienza "La conoscenza del mondo" è previsto che il bambino possa manipolare, giocare con qualsiasi artefatto tecnologico che fa parte della sua esperienza, tutto questo si coniuga molto bene con l'attività matematica. Infatti, nella SdI, l'area della matematica si presenta come ambito privilegiato per lo sviluppo di capacità logiche dunque, come linguaggio utile per la conquista delle capacità di pensare e ragionare in modo scientifico. L'artefatto tecnologico può, dunque, fungere da ponte giocoso per attivare capacità di comunicazione e di elaborazione di significati in modo che il bambino possa, con gradualità, passare dalle azioni alle operazioni, dal concreto all'astratto, attraversando i piani dell'organizzazione spazio-tem-

7. Intreccio di codici comunicativi ed espressivi diversi (scritto, sonoro, immagini e video) utilizzati in maniera integrata per realizzare un unico "oggetto comunicativo" (ad esempio le "emotion" destinate a rappresentare azioni o stati d'animo o ad abbreviare forme testuali complesse realizzabili sia usando segni di punteggiatura sia in forma grafica).

porale, delle classificazioni, delle relazioni e delle trasformazioni. Ad esempio, l'uso del *tablet* mediante *educational app*, se ben supportato dal lato didattico, può consentire ai bambini di fare esperienze di apprendimento significative e socialmente interattive, in quanto le dimensioni del gioco li abitua – fin da piccoli – a cogliere il senso delle cose (Riva, 2014)⁸.

8

In questa prospettiva, l'insegnante deve appropriarsi dell'attuale tecnologia per potersi orientare in una marea di *app* destinate a più piccoli, familiarizzare con quelle scelte in modo da acquisire consapevolezza in merito a differenti usi, potenzialità e limiti. Un percorso lungo e non facile che richiede un processo formativo, senza il quale si rischia un utilizzo molto circoscritto di *app* incentrato solo sulle classiche abilità di base come il riconoscimento di lettere e numeri, piuttosto che sulla creazione di contenuti o abilità di pensiero di ordine superiore.

A questo proposito, esistono degli ambienti di programmazione visuale, che possono essere usati tramite *app*, che aiutano a sviluppare il pensiero computazionale attraverso ambienti di programmazione adeguati ai bambini più piccoli.

Studi di ricerca hanno dimostrato come i bambini possono trarre beneficio nell'apprendimento attraverso la pratica della programmazione. Consentire di esplorare i concetti alla base dei linguaggi di programmazione nelle prime fasi della vita pone le basi per il successo nell'apprendimento del codice negli anni successivi (Bers, 2007, 2008; Cejka, Rogers, & Portsmore, 2006; Rogers & Portsmore, 2004). Inoltre, i primi studi condotti con LOGO⁹ hanno dimostrato che la programmazione al computer, se introdotta in modo strutturato, può aiutare i bambini piccoli a sviluppare differenti abilità cognitive, incluso il senso base del numero (Clements, 1999). Nell'ultimo trentennio gli studi di ricerca si sono concentrati molto sulla rimozione degli ostacoli fisici (manipolatori) e concettuali incontrati dai programmatori principianti tenendo presente, in particolar modo, i bisogni di sviluppo e capacità dei bambini (Rader, Brand, & Lewis, 1997; Ioannidou, Repenning, Lewis, Cherry, & Rader, 2003).

9

La maggior parte di questi ostacoli sono stati superati e gli attuali *Visual Programming Language* (VPL) consentono di programmare trascinando e collegando le icone sullo schermo; in questo modo anche i più piccoli hanno la possibilità di rendere visibili e "concreti" i loro processi di pensiero astratto. Tutto ciò porta a una maggiore facilità di manipolazione e controllo; inoltre, la visualizzazione sullo schermo aiuta gli insegnanti ad avere informazioni su cosa e come pensa un bambino.

La natura aperta di questi VPL, combinati con la struttura intrinseca della programmazione informatica, promuove in maniera efficace l'immaginazione e la creatività (Resnick, 2007); ciò offre ai bambini l'opportunità di apprendere le competenze essenziali per la programmazione dei computer in tenera età, al contempo consente agli insegnanti l'integrazione della tecnologia in modo significativo.

I programmi di *coding* per bambini hanno raggiunto una notevole popolarità con l'introduzione di *tablet* e altri dispositivi digitali poiché questi ultimi rendono l'interazione con lo strumento più appetibile per le mani piccole, il tutto in un contesto ludico capace di calamitare l'attenzione. Introdotti in sezione, tali programmi promuovono nuove opportunità di apprendimento sia in termini di conoscenze fon-

8. Il significato che ha la tecnologia è diverso da quello del gruppo sociale che la usa.

9. Primo linguaggio di programmazione pensato per scopi eminentemente educativi, orientato all'infanzia. Progettato nel 1967 da W. Feurzeig, Seymour Papert e C. Solomon, vide la prima implementazione della sua caratteristica interfaccia grafica, la tartaruga, nel 1969. Il LOGO trova origine negli studi di Papert con Jean Piaget e nella successiva fondazione, con Marvin Minsky, del laboratorio di intelligenza artificiale del MIT. Il suo fondamento filosofico è costruttivista.

damentali sia in termini di conoscenza specifica. Inoltre incoraggiano interazioni e collaborazioni tra pari nonché occasioni di dialogo con gli adulti; tutto questo favorisce il miglioramento delle capacità di linguaggio, l'arricchimento del vocabolario e la costruzione delle competenze di base (come ordinare, stabilire corrispondenze, classificare ecc.).

Di seguito si presentano le peculiarità e le potenzialità educative di ScratchJr, un VPL basato su Scratch e ridisegnato appositamente per la SdI.

3 Scratch Junior

Scratch è un ambiente di programmazione che usa un *Visual Programming Language* (VPL) che consente di realizzare contenuti digitali interattivi: unisce un linguaggio di programmazione semplice e intuitivo a un sistema di *authoring*¹⁰.

10

È costituito da quattro elementi (Figura 1):

- un linguaggio visuale che permette di costruire programmi, assemblando blocchi come in un gioco di costruzioni;
- un ambiente di programmazione grafico (*Integrated Development Environment* IDE) che consente di creare, modificare ed eseguire programmi scritti nel linguaggio di Scratch;
- un sito web e una piattaforma *cloud*¹¹ dove creare e salvare i progetti e interagire con altri utenti che utilizzano Scratch;
- una community¹² che crea, condivide e scambia idee e progetti.

11

12



Figura 1
Elementi che compongono il sistema Scratch (Beri & Boscaini, 2016, p. 2).

Ideato e sviluppato dallo Scratch Team presso il *Lifelong Kindergarten Group* dei MIT Media Lab di Boston,¹³ Scratch è un software libero¹⁴ il cui ambiente è utilizzabile sia via web sia tramite installazione sul proprio computer, *tablet* e *smartphone*, in

13 – 14

10. Strumento che consente lo sviluppo rapido di software multimediali grazie alla presenza di elementi grafici e sonori pronti all'uso.

11. <https://scratch.mit.edu>

12. "A creative learning community".

13. Massachusetts Institute of Technology (MIT), Cambridge - Massachusetts.

14. Un software a codice aperto (Open source); il codice sorgente ha la licenza GPL (General Public License) che ne garantisce la libertà di uso, studio, modifica e ridistribuzione.

quanto le varie versioni sono compatibili con i sistemi operativi Windows, Mac OS X e con le distribuzioni di GNU/Linux.

L'idea di base è di combinare contenuti multimediali, immagini, animazioni, suoni e testi per produrre programmi informatici utilizzabili anche da bambini o persone inesperte di linguaggi di programmazione; in altre parole consente di programmare in maniera semplice e divertente puntando molto sullo spirito di condivisione e collaborazione. Infatti, Scratch è basato sulla metafora del teatro, o del cinema: l'azione si svolge sul palcoscenico (*stage*), dove gli attori, o i personaggi (*sprite*), eseguono quanto è scritto nel loro copione (*script*).

Nello specifico, uno *sprite* può avere più *script*; ogni *script* è composto da una serie di istruzioni impilate l'una sull'altra e ciascuna istruzione è costituita da un blocco colorato. Gli *sprite* hanno un proprio guardaroba, con uno o più costumi, che possono essere indossati in corso d'opera, uno solo per volta. Anche lo *stage*, che è uno *sprite* un po' particolare, ha *script*, sfondi e suoni.

Con Scratch – proprio come a teatro – c'è chi lavora dietro le quinte ed esegue durante lo spettacolo i compiti che gli sono stati assegnati.¹⁵

15

Un VPL come Scratch offre nuove strade nei processi di insegnamento e d'apprendimento, le sue peculiarità danno possibilità a ciascun bambino di esprimere e condividere le proprie idee con gli altri in molti modi (Ackermann, 2002).

La programmazione visuale non è altro che un metodo di rappresentazione che permette di esprimere un procedimento come concatenazione di blocchi colorati che ne rappresentano i passi elementari, o le istruzioni che li descrivono.

I bambini in qualità di programmatori "in erba" possono ricoprire molti ruoli: autore, regista e sceneggiatore. Ad esempio, scegliere di fare lo sceneggiatore vuol dire decidere gli sfondi per le scenografie, scrivere i copioni per gli attori e preparare le loro raccolte di costumi e suoni. Tutto questo significa pianificare quello che accade nelle varie situazioni (definire chi c'è sulla scena, in quale posizione del palco si trova ecc.), e in seguito provare le scene per vedere cosa succede effettivamente: se c'è qualcosa che non funziona bisogna effettuare delle modifiche, correzioni.

Di questo VPL esiste anche la versione Junior (ScratchJr)¹⁶, cioè la versione semplificata di Scratch, fruibile dai bambini aventi età compresa tra 4-8 anni.

16

ScratchJr è una *app* per il *coding* gratuita che si può scaricare e installare su *tablet* Android e su iPad.

La logica di ScratchJr è analoga a quella di Scratch, così da permettere anche ai bambini più piccoli di imparare a scrivere delle storie scegliendo fra una serie di personaggi, realizzare giochi e animazioni senza scrivere una sola riga di codice e senza conoscere la programmazione: è sufficiente spostare dei blocchi creando delle sequenze di comportamenti per i singoli personaggi.

ScratchJr è stato ridisegnato e la sua interfaccia (Figura 2) adattata ai più piccoli: risulta molto semplificata nei comandi perché basata sull'iconografia e su note vocali. Il codice viene creato trascinando blocchi in un'area di codifica e agganciandoli insieme. Tutti i blocchi sono completamente *icon-based* (nessun testo): i bambini possono utilizzare questo linguaggio prima di poter leggere. Inoltre, i blocchi sono collegati da sinistra a destra, come le parole.

15. Manovrare le luci, cambiare la scenografia, far partire musiche e suoni di sottofondo.

16. La versione iniziale è stata lanciata nel mese di luglio 2014 per iPad, la versione Android nel 2015; poi nel 2016 è stata lanciata un'applicazione Chromebook.

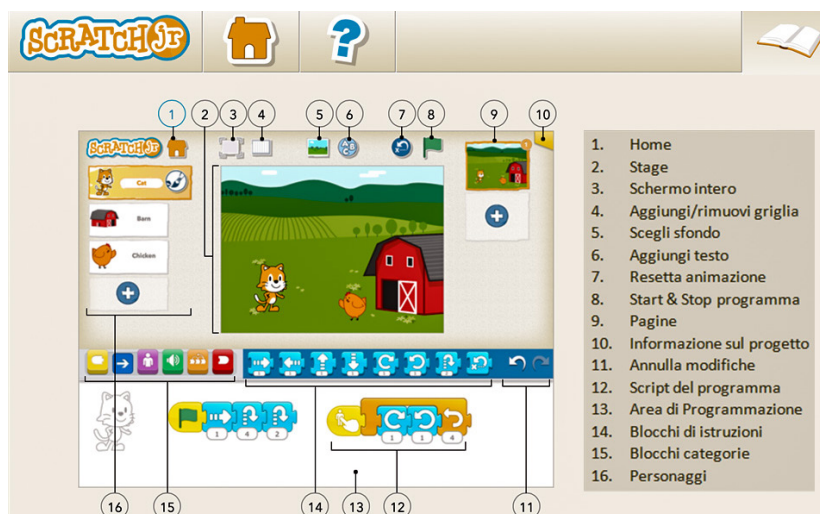


Figura 2
Interfaccia di ScratchJr.

La semplicità dell'interfaccia utente implica una categoria di blocchi di programmazione esigua (Figura 3).

CATEGORIA	NOTE	
Attivazione	Avvia <i>script</i> e invia messaggi ad altri <i>script</i>	
Movimento	Sposta <i>sprite</i> e cambia gli angoli	
Aspetto	Controlli di visibilità, i costumi, e la sintesi vocale bolla	
Suono	Riproduce un " <i>pop</i> " suono o un suono registrato	
Controllo	Ripete una parte dello scritto un determinato numero di volte	
Fine	Finisce, ripete infinitamente, e va alla pagina specifica del progetto	

Figura 3
Categorie blocchi di ScratchJr.

Oltre agli *sprite*, i bambini possono aggiungere sfondi ai progetti, per creare differenti ambienti e atmosfera. Ogni sfondo viene trattato come una pagina in un libro e ha una propria serie di *sprite*.

Un progetto può avere un numero massimo di quattro sfondi.

Attraverso una serie di giochi ed esercizi interattivi basati su un'interfaccia visuale, il bambino può determinare le azioni di uno o più personaggi spostando blocchi o oggetti grafici su un monitor. A ciascun blocco corrisponde un'azione, una linea di codice che non ha bisogno quindi di essere digitato. Basta muovere o assemblare i mattoncini tra di loro nell'ordine necessario a raggiungere un certo obiettivo, e il gioco è fatto.

In definitiva, ScratchJr è molto semplice perché dotato di un set ridotto di istruzioni e, soprattutto caratterizzato da un approccio visuale "clicca e trascina" che rende l'esperienza di programmazione simile ad un gioco di costruzioni ad incastro.

A ogni mattoncino corrisponde un comando, un'istruzione che non ha bisogno di essere digitata ma solo "incastrata" al blocco precedente.

La programmazione visuale è un modo intuitivo e divertente per esprimere un procedimento; introdotta in sezione aiuta a trasformare in modo creativo l'approccio alla conoscenza matematica relativa alle procedure. Fare *coding* con ScratchJr aiuta i più

piccoli a pensare meglio e in modo creativo, stimola la loro curiosità attraverso quello che apparentemente può sembrare solo un gioco; consente di imparare le basi della programmazione informatica, insegna a “dialogare” con il computer, a impartire alla macchina comandi in modo semplice e intuitivo. Il segreto sta tutto nel metodo: poca teoria e tanta pratica.

4 ScratchJr in sezione

Sulla base del quadro teorico menzionato, gli obiettivi pedagogici del percorso di R.–A. sono:

- sperimentazione e controllo scientifico di situazioni didattiche inerenti l'utilizzo di ScratchJr nel campo di esperienza “La conoscenza del mondo”;
- favorire la pratica del *coding*, in sezione, come palestra per educare al pensiero computazionale (i bambini mentre giocano esercitano e accrescono le capacità logico-creative-simboliche e le capacità di socializzazione);
- sostenere percorsi d'apprendimento che colleghino l'uso di dispositivi digitali a vissuti esperienziali plurisensoriali.

Lo scaffolding metodologico-didattico, strutturato nel pieno rispetto delle peculiarità biopsichiche, sociali e culturali dei piccoli allievi nonché dei loro stili e ritmi di apprendimento, ha posto le sue fondamenta su una serie di metodi (storytelling, role playing, learning by doing, problem solving) che includono lavoro individuale, lavoro di gruppo e istruzione diretta da parte dell'insegnante, in modo da offrire ai piccoli allievi opportunità di esplorare, conoscere e pianificare per sviluppare le proprie abilità matematiche.

In quest'ottica, è stata privilegiata la metodologia del *Math talk*¹⁷ in quanto aiuta molto il bambino nel processo di acquisizione relativo al senso del numero. Infatti, come affermano D'Amore e Fandiño Pinilla (2015) il *Math talk* è fondamentale perché costituisce un momento di negoziazione di significati e di concetti tra insegnanti e allievi.

Le pratiche didattiche, che afferiscono alle tre aree (quantification, counting and representing number) di dominio del senso del numero (Moomaw, 2011), sono state esperite prima in maniera tradizionale (senza l'impiego della tecnologia) e poi con il *tablet*, utilizzando l'App ScratchJr in base al contesto e all'intenzionalità didattica. Il paradigma educativo-formativo seguito è rappresentato in Figura 4.

17

¹⁷. Conversazione matematica che si concretizza nella discussione e nel dialogo con gli allievi. L'insegnante crea un ambiente finalizzato ad incoraggiare una discussione costruttiva dove l'allievo può descrivere i propri metodi ad un'altra persona chiarendo il proprio modo di pensare e la questione per gli altri. È una conversazione incentrata sullo sviluppo della comprensione per tutti i bambini della sezione.

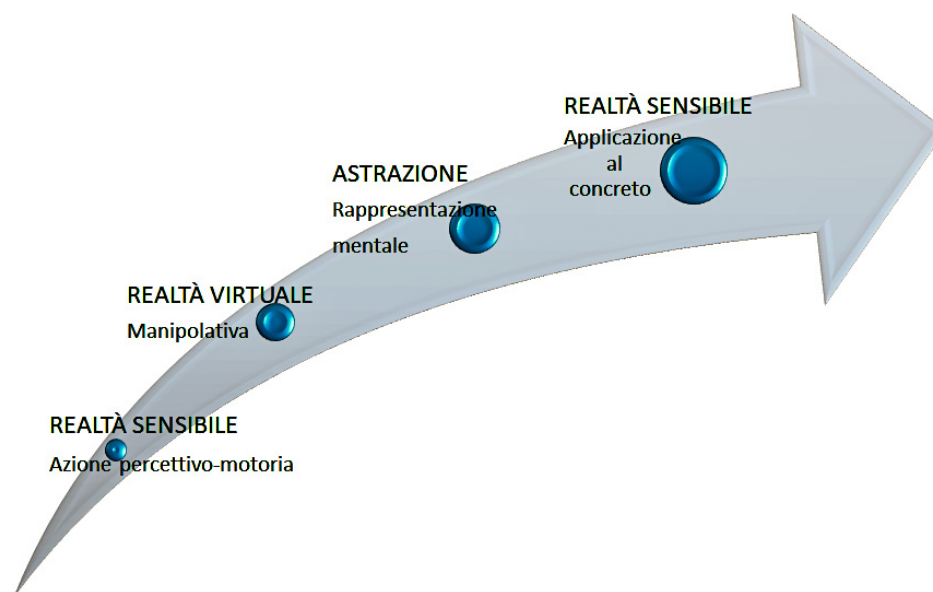


Figura 4
Paradigma metodologico.

A titolo di esempio, si riporta di seguito un estratto del laboratorio, articolato in quattro attività, ciascuna della durata di due ore, finalizzato alla corrispondenza quantità-numero.

4.1 Attività 1 – “Aiutare lo scoiattolo Tom”

L'attività, realizzata con 22 bambini (fascia d'età 5 anni), inizia con il racconto di uno scoiattolo smemorato che ha perso la sua provvista di ghiande e non riesce a trovarla. La storia introduce il gioco finalizzato ad aiutare lo scoiattolo.

L'insegnante favorisce la verbalizzazione per quanto riguarda sia gli aspetti del gioco che quelli aritmetici allo stato embrionale. La struttura del gioco è stata articolata in modo da favorire l'esplorazione spaziale “micro e macro”, si avvale pertanto di un percorso a ostacoli che i bambini devono affrontare in coppia con ruoli differenti: un componente della coppia viene bendato e l'altro componente deve aiutarlo, tramite opportune indicazioni, nel percorso ad ostacoli. La coppia deve affrontare il percorso per arrivare al traguardo rappresentato da un cerchio contenente le ghiande perse dallo scoiattolo. Dopodiché, il bambino non bendato deve individuare, su comando verbale del suo compagno, il cartoncino recante il numero corretto e togliere dal cerchio la quantità corrispondente di ghiande e riporle in un apposito cestino. Quattro bambini hanno il compito di monitorare tutti gli steps del gioco in modo da far emergere, in maniera costruttiva, eventuali errori; mentre altri dieci bambini rappresentano le cifre da 0 a 9 tenendo nelle mani – bene in vista – il cartoncino sul quale è scritto il numero. Infine, i rimanenti bambini, sono responsabili del cestino dove vengono riposte le ghiande.

L'insegnante lascia ai bambini libertà nella scelta dei ruoli, successivamente li invita a provare in coppia il percorso ad ostacoli. In definitiva, tutti i bambini ricoprono i diversi ruoli ad eccezione di quello inerente il componente bendato della coppia: durante la sperimentazione alcuni bambini non hanno voluto essere bendati e l'insegnante ha rispettato il loro volere. La tipologia di gioco proposta affina l'orientamento spaziale, il controllo e la coordinazione visuo-motoria, la flessibilità, l'attenzione e la concentrazione; al contempo armonizza, in una relazione di responsabilità reci-

proca, i differenti ruoli ricoperti, in particolar modo per i componenti della coppia: le istruzioni fornite dal bambino che deve guidare il suo compagno bendato fino al traguardo e viceversa il comando verbale dato dal bambino bendato al suo compagno il quale deve individuare il cartoncino che indica il numero (Figura 5) e inserire, quindi, la corrispondente quantità di ghiande nel cestino (Figura 6).

Di seguito, si riportano due momenti tratti dal diario dell'attività:

Bambina non bendata: «Devi fare un salto, poi lo zig-zag e salire sui mattoncini facendo due passi in avanti».



Figura 5
Una coppia nel percorso ad ostacoli.

Bambina bendata: «Devi prendere tre ghiande e inserirle dentro il cestino».



Figura 6
Il riconoscimento del cartoncino con il numero tre.

Il gioco preceduto dalla narrazione esalta gli aspetti emotivi e apre la strada a quelli aritmetici privilegiando le relazioni spaziali, la discriminazione percettiva, le associazioni, le sequenze logiche e i primi numeri. In particolare, il gioco consente di fare esperienza con il concetto di quantità che riveste un ruolo importante per una corretta educazione in campo aritmetico.

4.2 Attività 2 – “Giochiamo con ScratchJr”

Ai bambini viene chiesto di creare un progetto/programma sulla base dell'esperienza reale vissuta in precedenza; si tratta di rappresentare degli *stages* relativi alla corrispondenza quantità-numero. Di conseguenza, vengono discussi gli elementi di ScratchJr che è necessario mettere in gioco quando si pensa ad un programma come risultato di un processo di organizzazione e strutturazione logica e come strumento di comunicazione.

In quest'attività, i bambini non si trovano più ad agire con tutto il corpo in quanto viene privilegiato il canale percettivo-visivo e l'uso della coordinazione oculo-manuale: devono inventare una serie di istruzioni in sequenza per raggiungere l'obiettivo. In questo processo di progettazione/programmazione interattiva i bambini sono impegnati in una attività di problem-solving che richiede:

- identificazione degli obiettivi;
- formulazione di un piano;
- sviluppo di un primo tentativo per raggiungere l'obiettivo;
- sviluppo del metodo per valutare il successo;
- testare e valutare;
- revisione e apportare modifiche;
- sviluppo del prossimo tentativo.

In altre parole, i bambini costruiscono, risolvono, ragionano e astraggono sviluppando il pensiero computazionale. Questo compito in forma di gioco richiede la specificazione dei ruoli o l'onere di ricoprire più ruoli all'interno dei gruppi.

I bambini, dunque, diventano dei veri e propri programmatori che analizzano i problemi, pensano le soluzioni e utilizzano il codice per realizzarle.

Tutto ciò implica anche vestire i panni dell'investigatore: scoprire come funziona un programma scritto da altri o trovare l'errore che non fa funzionare il programma. Per le ragioni suddette, inizialmente l'insegnante presenta alla LIM il programma “I pesci nel mare” (Figura 7) in modo da stimolare la discussione non solo sui concetti base della programmazione, ma anche sull'individuazione dei blocchi di ScratchJr utilizzati e la relativa sequenza dei passi necessari.

L'obiettivo non è solo quello di riprendere i termini (avanti/dietro, destra/sinistra, su/giù, andare avanti di tot. numero di passi ecc.), ma anche di evidenziare l'importanza dell'ordine quando si danno delle istruzioni. D'altro canto, i bambini sono esperti nell'uso di ScratchJr, quindi l'insegnante si avvale di una sorta di *flipped* allo stato embrionale per capire se i bambini riescono ad orientarsi in un progetto già definito.



Figura 7
Editor di progetto
"I pesci nel mare".

Qui, di seguito, si riporta un breve estratto del protocollo di discussione, corredato da alcuni output:

- F.: «Hai preso lo sfondo del mare e poi abbiamo messo i pesci».
- M.: «Nooo! Prima devi scegliere i personaggi e poi metti tutti i personaggi che vuoi».
- M.: «Abbiamo aumentato i personaggi. Ora devi dire quanti sono i pesci che vedi e li devi associare al numero giusto come abbiamo fatto con il gioco delle ghiande».
- P.: «Prima devi aprire l'app sul tablet e poi crei lo stage».
- T.: «Maestra dobbiamo andare in ordine! Secondo me Paolo ha detto una cosa giusta e un'altra sbagliata».
- Ins.: «Spiegati meglio».
- T.: «Dopo che hai aperto l'app nella pagina devi cliccare sul segno "più" perché vuoi iniziare a lavorare. Cliccando sulla "più" fai un nuovo progetto scegliendo il mare» (Figura 8).

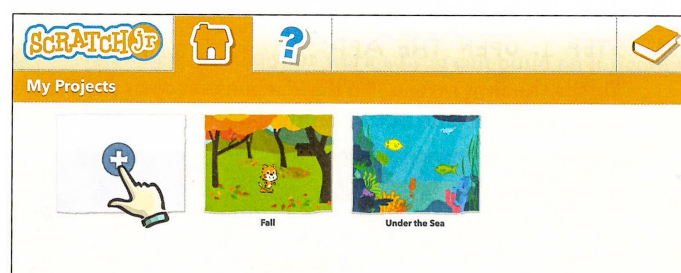


Figura 8
Editor per iniziare
un nuovo progetto.

- F. (interrompe T.): «(...) e poi la maestra ha dovuto creare i personaggi che sono i pesci e i numeri. Questo è lo script!».

- D.: «Noi questo lo sappiamo fare! La maestra vuole sapere quanti sono i pesci e devi cliccare sul numero (indicando il numero 3 sulla LIM). Dopo puoi vedere più pesci e devi fare la stessa cosa».
- M.: «Basta guardare l'ordine dei blocchi».
- A.: «Infatti la maestra ha utilizzato il blocco giallo che avvia lo script quando tocchi il personaggio e poi quello che ingrandisce il personaggio» (Figura 9).



Figura 9
Il blocco giallo degli eventi e il blocco fucsia degli aspetti.

L'insegnante, successivamente, divide la classe in gruppi eterogenei di 3 o 4 componenti sulla base dei livelli di partenza, stili cognitivi, aspetto emotivo e socio-affettivo, abilità e capacità. Distribuisce, poi, un *tablet* a ogni gruppo in modo che i bambini possano lavorare con ScratchJr. Infatti, ogni gruppo deve realizzare degli stage relativi alla corrispondenza quantità-numero.

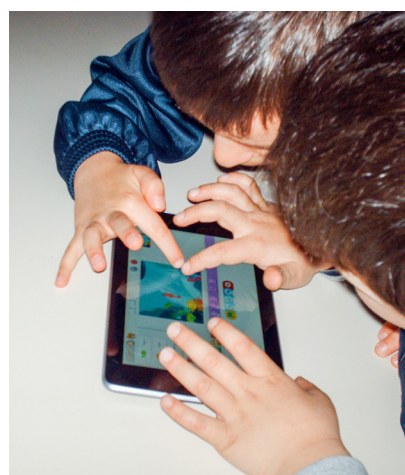
I passaggi effettuati sono stati i seguenti:

1. creazione dello stage e scrittura del numero all'interno di un quadrato;
2. scelta degli *sprite* (a piacere) dal comando apposito;
3. associazione degli *sprite* al simbolo numerico;
4. utilizzo dei blocchi di azione e di registrazione;
5. realizzazione di nuovi stage.

In figura 10 sono riportati alcuni momenti dell'attività.



Figura 10
Gruppi di bambini in azione con ScratchJr.



A titolo di esempio, si riportano due programmi realizzati dai bambini (Figura 11 e 12).



Figura 11
Le mele.

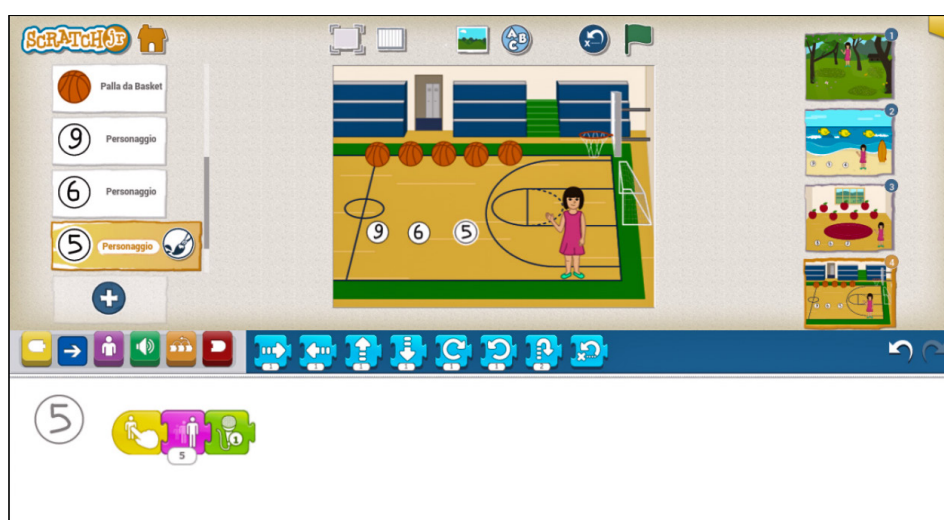


Figura 12
Palle da Basket.

Il laboratorio è stato completato con altre due attività, la prima impegnava singolarmente i bambini nella rappresentazione spontanea "quantità-numero" (Figura 13); nella seconda, invece, i bambini disposti in "circle time" dovevano inventare una storia utilizzando le parole *molti*, *pochi*, *nessuno*, *tanti*, *ogni* e i numeri da 0 a 9 e rappresentarla poi su un cartellone da collocare in sezione.



Figura 13
Rappresentazioni
spontanee
"quantità-numero".

Queste due attività supportano e completano le precedenti nell'ottica del paradigma educativo-formativo (Figura 5). Infatti, il bambino inizia a razionalizzare le due tipologie di esperienze (reale e virtuale) potenziando così il processo cognitivo inerente ai meccanismi semantici che regolano la comprensione della quantità, i meccanismi lessicali che regolano il nome del numero (ad esempio tre, quattro ecc.) nonché l'irrelevanza dell'ordine che non modifica la cardinalità e l'aspetto sintattico (distinguere una unità dall'insieme di elementi che la costituiscono).

5 Osservazione

Per il monitoraggio delle attività implicate nei processi di innovazione e degli effetti prodotti nell'ambiente educativo, sono state strutturate e gestite dai ricercatori delle schede di protocollo osservativo che consentissero di evidenziare conoscenze, abilità e competenze afferenti sia al campo di esperienza "La conoscenza del mondo", sia ad altri campi menzionati nelle Indicazioni Nazionali italiane (Allegato 1).

Questa modalità ha consentito di valorizzare l'uso di ScratchJr in sezione: una *app* non fine a se stessa, ma funzionale all'attivazione dei processi cognitivi vagliati nella fase di progettazione. Qui di seguito si riporta il quadro sinottico (Tabella 1) di alcuni risultati espressi in percentuale riferiti agli item più interessanti e prevalenti per questo tipo di attività ottenuti dal campione dei 22 piccoli allievi. I risultati ottenuti dai ricercatori sono stati in seguito condivisi e discussi con gli insegnanti di sezione.

VALUTAZIONE OBIETTIVI FORMATIVI	MAI	A VOLTE	PIÙ DELLE VOLTE	SEMPRE
Confronto quantitativi: tanti-pochi	3%	/	31%	66%
Cogliere l'equipotenza: tanti-quant	/	/	30%	70%
Confronto quantitativi: uno-tanti-pochi	18%	5%	25%	52%
Corrispondenza quantità-numero (da 1 a 9)	10%	5%	30%	55%
Collocare le azioni nel tempo: prima/dopo	15%	2%	25%	58%
Sperimentare in situazione il concetto di avanti/indietro	7%	/	18%	75%
Muoversi nella spazio con consapevolezza topologica	12%	5%	38%	45%
Coordinazione visivo-motoria	28%	8%	28%	36%
Controllo motricità fine	28%	18%	29%	25%

Tabella 1
Sinottico risultati
delle due attività.

Le fasi operative e valutative costituiscono un tentativo pilota di documentare traiettorie di apprendimento relative all'interfaccia di ScratchJr e i primi concetti di programmazione come supporto per la costruzione di conoscenze fondamentali e di conoscenze specifiche nell'ambito logico-matematico.

L'uso delle differenti categorie dei blocchi di programmazione favorisce, nei bam-

bini, lo sviluppo di abilità in merito alla stima (“quanti” o “quanto lontano”), alla previsione (cosa accade quando eseguo il programma interamente) e alla riflessione sui cambiamenti effettuati (i cambiamenti si traducono nel risultato previsto? Sì/no e perché).

Questo processo facilita l'apprendimento esplorativo delle istruzioni e supporta la percezione dei bambini sul ruolo che ogni azione ha in un programma.

Le azioni di programmazione devono avere non solo un output visibile, ma anche un tempo per essere eseguite; tale consapevolezza è maturata nel tempo consentendo la sperimentazione di nuove situazioni e, quindi, la creazione di programmi sempre più strutturati. Infatti, all'inizio l'interfaccia amichevole, gioiosa ed invitante ha indotto molti bambini a riempire gli schermi con molti personaggi mediante un “clic ripetuto” del pulsante *sprite* per poi non costruire alcun comportamento di questi personaggi, ma unicamente divertendosi nell'ingrandirne/diminuirne la dimensione. A corredo della **Tabella 1**, si riportano alcuni dati valutativi inerenti l'utilizzo di ScratchJr in sezione (**Tabella 2**).

	MAI	A VOLTE	PIÙ DELLE VOLTE	SEMPRE
Dà istruzioni in sequenza per ottenere semplici risultati	/	3%	31%	66%
Trova diverse categorie di blocchi	/	/	20%	80%
Sposta i blocchi nell'area scripting	/	13%	37%	50%
Utilizza i blocchi nell'area scripting come pulsanti	/	/	46%	54%
Seleziona caratteri e impostazioni	5%	15%	21%	59%
Usa e crea sfondi per il progetto	/	/	10%	90%
Modifica o disegna nuovi personaggi e impostazioni	5%	10%	25%	60%
Aggiunge nuove pagine	5%	8%	25%	62%
Gioca/esegue i programmi usando il pulsante della bandiera verde	/	/	20%	80%
Combina i diversi blocchi di movimento in sequenze programmate	5%	10%	35%	50%
Usa i blocchi di ripetizione e “ripetizione per sempre” per ottenere risultati specifici	8%	12%	52%	28%
Utilizza i numeri sui blocchi di movimento per ridurre il numero dei blocchi necessari	5%	10%	32%	53%
Utilizza il blocco finale per indicare la fine di un programma	/	/	20%	80%
Salva il proprio lavoro e apre progetti nuovi ed esistenti	5%	10%	26%	59%
Trova blocchi su istruzione verbale dei compagni	8%	12%	22%	58%

Tabella 2
Estratto di alcuni dati
relativi all'utilizzo
di ScratchJr in sezione.

Si sottolinea che tutte le attività condotte all'interno del laboratorio hanno consentito a ciascun bambino di mettere a disposizione del gruppo “ciò che sapeva fare”,

assumendo, nel rispetto delle regole condivise, un ruolo ed un compito; di conseguenza si è instaurato un ambiente sereno che ha favorito la modalità di apprendimento cooperativo.

In merito agli obiettivi prefissati si può realisticamente affermare che la dimensione euristica di ScratchJr ha avuto ricadute significative sulle abilità attentive e di memoria di lavoro visuo-spaziale. Inoltre, l'opportunità di creare storie impegnandosi nella struttura della sceneggiatura (Figura 14 e 15) ha consentito ai bambini di sperimentare in maniera esplicita il passaggio da una forma di ragionamento concreto sugli oggetti e sull'ambiente reale ad un pensiero più astratto elicitato dall'utilizzo degli oggetti rappresentati nel mondo virtuale.

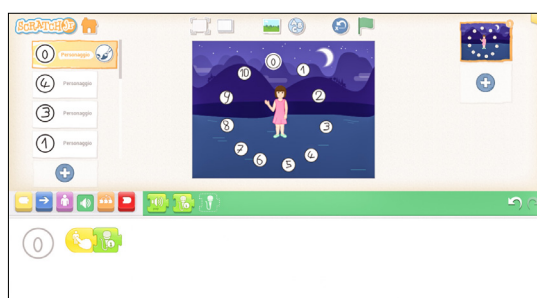


Figura 14
Filastrocca dei numeri.

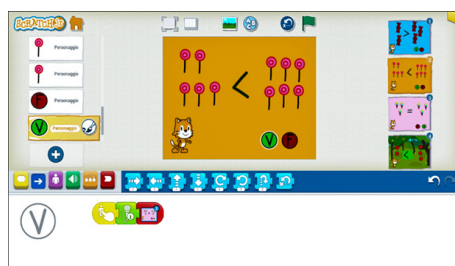


Figura 15
Confronto di quantità.



Tutto ciò rientra nella competenza digitale che si presenta e sviluppa non come un'area a sé stante, ma come una competenza che entra in relazione e potenzia le altre aree di competenza. Dunque, non un impiego acritico e passivo di contenuti digitali, ma produzione di esperienze digitali fruite in termini di efficacia e rilevanza apprenditiva.

6 Conclusione

L'obiettivo di questo tipo di laboratori non è formare una generazione di futuri programmatori, ma educare i più piccoli al pensiero computazionale che coinvolge la capacità di risolvere problemi – anche complessi – applicando la logica, ragionando passo-passo sulla strategia migliore per arrivare alla soluzione.

ScratchJr è un ambiente di sviluppo pedagogico integrato proprio perché è un VPL che sostiene – senza tralasciare l'aspetto ludico – alcune significative esperienze a livello affettivo, cognitivo, metacognitivo e relazionali che costituiscono il filo con-

duttore della SdI.

Consente un approccio alla matematica nuovo, stimolante perché favorisce uno sviluppo poliedrico delle capacità mentali ed una formazione non solo matematica, ma anche socio-culturale.

In questo modo, i bambini non imparano soltanto a programmare ma programmano per imparare.

Per ciò che concerne, invece, i futuri insegnanti, è importante non solo sviluppare dal punto di vista teorico consapevolezza in merito alle potenzialità di ambienti di programmazione visuale, ma anche acquisire una serie di competenze metodologiche riguardo l'utilizzo in sezione.

Infatti, nell'intero percorso i contenuti digitali non sono stati concepiti come aggiuntivi a quelli tradizionali, ma come parte integrante di una attenta progettazione implementata attraverso una didattica per competenze, come previsto dalle Indicazioni Nazionali. Come si legge nella parte "Spazi e ambienti per l'apprendimento" del PNSD:

«(...) L'educazione nell'era digitale non deve porre al centro la tecnologia, ma i nuovi modelli di interazione didattica che la utilizzano».¹⁸

18

Bibliografia

Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference. *Future of learning group publication*, 5(3), 438.

Beri, M., & Boscaini, M. (2016). *Imparare a programmare con Scratch. Il manuale per programmatori dai 9 anni in su*. Milano: Apogeo.

Bers, M. U., & Horn, M. S. (2010). Tangible programming in early childhood. *High-tech tots: Childhood in a digital world*, 49, 49-70.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.

Bruner, J. S. (1990). *Acts of meaning* (Vol. 3). Harvard: Harvard University Press.

Bruner, J. S. (2009). *Actual minds, possible worlds*. Harvard: Harvard University Press.

Calvani, A., (A cura di). (2007). *Tecnologia, scuola, processi cognitivi. Per una tecnologia dell'apprendere*. Milano: Franco Angeli.

Cejka, E., Rogers, C., & Portsmore, M. (2006). Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school. *International Journal of Engineering Education*, 22(4), 711.

Clements, D. H. (1999). The future of educational computing research: The case of computer programming. *Information Technology in Childhood Education Annual*, 1, 147-179.

Costabile, F. A., & Serpe, A. (2010). The computer in nursery schools with the INF@ 0.1 software. An action-research experience. In *EDULEARN10 Proceedings* (pp. 3256-3262). IATED.

18. http://www.istruzione.it/scuola_digitale/allegati/Materiali/pnsd-layout-30.10-WEB.pdf (p. 28).

- Costabile, F.A., & Serpe, A. (2011). La "prima matematica" con INF@0.1: un'esperienza monitorata nell'anno scolastico 2007/08. In D'Amore B., Sbaragli S. (Eds.), *Matematica ed esperienze didattiche* (pp. 80-81). Bologna: Pitagora.
- D'Amore, B. (2003). *Le basi filosofiche, pedagogiche, epistemologiche e concettuali della Didattica della Matematica*. Bologna: Pitagora.
- D'Amore, B., & Fandiño Pinilla, M. I. (2015). *Matematica: come farla amare: Miti, illusioni, sogni e realtà*. Firenze: Giunti Scuola.
- Donaldson, M. (2009). *Come ragionano i bambini*. Springer Science & Business Media.
- Duval, R. (2000). Basic issues for research in mathematics education. In T. Nakahara, M. Koyama (Eds.), *Proceedings of the 24th Conference of the International Group for the Psychology of Mathematics Education* Vol 1 (pp. 55-69). Hiroshima: Hiroshima University.
- Ferri, P., & Mantovani, S. (2006). *Bambini e computer. Alla scoperta delle nuove tecnologie a scuola e in famiglia*. ETAS RCS.
- Ferri, P., & Mantovani, S. (2008). *Digital kids. Come i bambini usano il computer e come potrebbero usarlo genitori e insegnanti*. ETAS RCS.
- Gardner, H. (1993). *Multiple Intelligences: The Theory in Practice*. A Reader.
- Gardner, H. (2011). *The Unschooled Mind: How Children Think and How Schools Should Teach*. Basic Books.
- Ioannidou, A., Repenning, A., Lewis, C., Cherry, G., & Rader, C. (2003). *Making constructionism work in the classroom*. *International Journal of Computers for Mathematical Learning*, 8(1), 63-108.
- Kazakoff, E. & Bers, M. (2012). Programming in a Robotics Context in the Kindergarten Classroom: The Impact on Sequencing Skills. *Journal of Educational Multimedia and Hypermedia*, 21(4), 371-391.
- Jonassen, D. H. (2000). *Computers as mindtools for schools: Engaging critical thinking*. Prentice Hall, New York.
- Littleton, K., & Mercer, N. (2013). *Interthinking: Putting talk to work*. Routledge.
- Moomaw, S. (2011). *Teaching Mathematics in Early Childhood*. Brookes Publishing Company. PO Box 10624, Baltimore, MD 21285.
- Norman, D. A. (1993). *Things that make us smart*. Addison Wesley, Reading (Mass.).
- Norman, D. A. (2004). *Emotional design: Why we love (or hate) everyday things*. New York: Basic Civitas Books.
- Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. Basic Books, 10 East 53rd St., New York, NY 10022-5299.
- Pira F., Marrali V. (2007), *Infanzia, media e nuove tecnologie*, Milano: Franco Angeli.
- Rader, C., Brand, C., & Lewis, C. (1997). Degrees of comprehension: children's understanding of a visual programming environment. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (pp. 351-358). New York: ACM.

- Resnick, M. (2007). Sowing the seeds for a more creative society. *Learning and Leading with Technology*, 35(4), 18-22.
- Riva, G. (2014). *Nativi digitali: crescere e apprendere nel mondo dei nuovi media*. Bologna: Il Mulino.
- Rogers, C., & Portsmore, M. (2004). Bringing engineering to elementary school. *Journal of STEM Education: innovations and research*, 5(3/4), 17.
- Sbaragli, S. (A cura di). (2011). *Buone pratiche d'aula in matematica. Percorsi didattici in continuità tra scuola dell'infanzia e secondaria di secondo grado*. Bologna: Pitagora.
- Serpe, A. (2008). KidSmart: an essential tool for mathematical education in nursery schools. In *Learning to Live in the Knowledge Society* (pp. 313-320). Boston: Springer, MA.
- Vygotskij, L. S. (2010). *Storia dello sviluppo delle funzioni psichiche superiori*. Firenze: Giunti.

Sitografia

<http://www.indicazioninazionali.it/>
<http://www.istruzione.it/allegati/2015/prot2187.pdf>
http://www.istruzione.it/scuola_digitale/allegati/Materiali/pnsd-layout-30.10-WEB.pdf
<https://www.programmailfuturo.it/media/docs/Circolare-Programma-il-Futuro-2016.pdf>
<https://programmailfuturo.it/>
<https://scratch.mit.edu>

Autore/Annarosa Serpe

Dipartimento di Matematica e Informatica – Università della Calabria, Italia
annarosa.serpe@unical.it