

L'apprendimento di conoscenze geometriche e abilità visuo-spaziali attraverso il coding

Learning geometry and visuo-spatial abilities through coding

Martina Sabatini^a

^a *Università degli Studi di Perugia*, martyna.sabatini@gmail.com

Abstract

Il presente lavoro affronta la tematica del coding focalizzandosi in particolar modo sulla sua introduzione in uno specifico curriculum di scuola primaria. Viene presentato il progetto di ricerca di dottorato “L'apprendimento di conoscenze geometriche e abilità visuo-spaziali attraverso il coding”. Il progetto, non ancora concluso, intende riflettere sulla possibilità di introdurre attività di coding all'interno dei curricoli di geometria delle classi terze di scuola primaria, al fine di studiare se e come questo tipo di attività influisca sull'acquisizione di apprendimenti geometrici e abilità visuo-spaziali. Per farlo, verranno confrontati, sia in termini quantitativi che qualitativi, i processi di insegnamento-apprendimento di classi sperimentali (quelle in cui avvengono attività di coding) e di classi di controllo (quelle in cui non avvengono attività di coding). Le classi sono state scelte secondo un criterio di omogeneità relativamente alle situazioni di partenza delle conoscenze geometriche e abilità visuo-spaziali.

Parole chiave: coding; abilità visuo-spaziali; geometria.

Abstract

This paper deals with the theme of coding, focusing particularly on its introduction in a specific primary school curriculum. The work presents the PhD research project “Learning geometry and visuo-spatial abilities through coding”. The project, which is not yet concluded, aims to reflect on the possibility of introducing coding activities within the primary school curricula of geometry of third graders, in order to study if and how these activities affect learning geometry and visuo-spatial abilities. To do so, the processes of teaching and learning of experimental classes (those in which coding activities take place) and control classes (those in which coding activities do not take place) will be compared, both in terms of quantity and quality. The classes has been chosen according to a uniformity requirement with respect to the starting points of geometrical knowledge and visuo-spatial abilities.

Keywords: coding; visuo-spatial abilities; geometry.

1. Introduzione

La parola coding sta acquistando una crescente notorietà sia in ambito scolastico che extrascolastico, in Italia e all'estero, grazie al fiorire di iniziative e progetti di largo successo ideati in ambiti diversi. In realtà, però, quella che ai non addetti al settore può sembrare come "l'ultima moda del momento", ha radici lontane. Innanzitutto, quando si parla di coding si intende il processo di trasformazione di una serie di azioni, volte alla risoluzione di un problema, in un linguaggio simbolico. In altre parole, di fronte ad un problema, compiere un'azione di coding significa progettare un algoritmo che risolva il problema per poi trasformarlo in un linguaggio simbolico scelto. A differenza di quanto solitamente si pensi, è possibile fare coding anche in assenza di dispositivi elettronici quali computer, tablet, LIM o telefonini, in questo caso si dice che viene fatto in modalità *unplugged*¹. Quando, invece, si utilizzano dispositivi elettronici, è più opportuno parlare di programmazione per chiarire che il linguaggio simbolico in cui viene tradotta la serie di azioni utili a risolvere un problema può essere compreso ed eseguito da un computer. In sostanza, la programmazione è un tipo particolare di coding e fa riferimento all'utilizzo di dispositivi elettronici.

L'idea di proporre all'interno del contesto scolastico attività simili, in un momento, come quello attuale, in cui tanti vorrebbero fare un passo indietro rispetto alla presenza della tecnologia a scuola a seguito del recente rapporto OECD (2015) interpretato e tradotto da troppi con l'avvertimento "le tecnologie peggiorano l'apprendimento", sembra una scommessa persa in partenza. In realtà, si ritiene opportuno precisare in riguardo ai risultati del rapporto OECD che l'affermazione "le tecnologie peggiorano l'apprendimento" denota una visione molto superficiale della questione perché non dà conto della specifica fascia d'età presa in esame (15-16 anni), del tipo di tecnologie e dei contesti interessati. Inoltre, le criticità emerse da questo dibattito non sono nuove nell'ambito educativo e in particolare scolastico, in quanto, fin dal suo arrivo, ci si è interrogati ad esempio sul perché inserire ed utilizzare il computer in classe. La questione è datata e credo mal posta per almeno due motivi. Innanzitutto, la proposta non è di utilizzare il computer, che denoterebbe una visione esclusiva, ma piuttosto di utilizzare *anche* il computer nell'ottica di una cornice teorica inclusiva in cui la varietà degli strumenti è vista come occasione di offrire opportunità diverse di apprendere. Inoltre, verrebbe da chiedersi piuttosto perché non utilizzare il computer? Perché non cogliere la sfida e assumersi la responsabilità educativa? Per rispondere con precisione è doveroso ripercorrere storicamente la discussione in merito e definire in modo opportuno la cornice teorica di riferimento che fa da sfondo a questa idea. In particolare, si cercherà di comprendere come è nata l'idea di inserire il computer in aula e qual è il senso di proporre attività di coding a scuola.

Di seguito viene presentato il progetto di ricerca "L'apprendimento di conoscenze geometriche e abilità visuo-spaziali attraverso il coding", attualmente in fase di svolgimento.

¹ Si vedano ad esempio le attività proposte da "Programma il Futuro", "Code.org", "CS Unplugged", "Teaching London Computing", "Computer Science Education Week", "Samsung Smart Coding", solo per citarne alcune.

2. Il coding: una sfida da cogliere?

Negli anni Sessanta del secolo scorso nasce l'idea di inserire il computer all'interno dell'ambito scolastico e quella di offrire ai bambini l'opportunità di imparare un linguaggio attraverso cui comunicare con un computer. L'idea è del celebre matematico, informatico e pedagogista Seymour Papert (1993), il quale aveva intenzione di instaurare un cambiamento culturale nel rapporto delle persone con le tecnologie. Questo cambiamento, secondo il professore del Massachusetts Institute of Technology (MIT), doveva avvenire in termini di un cambiamento delle persone dallo status di consumatori delle tecnologie in quello di produttori, e quindi doveva fare in modo che il rapporto con le macchine non fosse di tipo passivo ma anzi consapevole e critico. Affinché ciò potesse avvenire, secondo Papert era necessario che tali persone fossero in grado di programmare un computer. L'idea di Papert, però, incontrò sin da subito le resistenze degli ambienti educativi che si caratterizzavano, allora come oggi, come poco flessibili di fronte alla possibilità di cambiamento in merito a contenuti e metodologia. Scrive, infatti, Papert che quello che vorrebbero gli insegnanti è semplicemente utilizzare “nuovi gadget per insegnare le stesse cose in modi solo leggermente diversi rispetto al solito”² (Papert & Solomon, 1971, p. 2) e che la stessa comunità che vorrebbe i computer in ambito educativo “sembra piuttosto riluttante a utilizzare i computer per scopi diversi da quelli di somigliare il più possibile a quanto insegnato nelle scuole nei secoli passati”³ (ivi, p. 1). Per rispondere alle reticenze, Papert cercò di dimostrare la validità della sua idea facendo riferimento alle modalità di apprendimento delle persone e in particolare dei bambini, che era riuscito a cogliere grazie a molti studi. E così, sebbene di primo impatto l'idea di imparare a programmare sembri qualcosa di molto complesso, in termini di apprendimento, per i bambini e molto distante dalle loro capacità, in realtà Papert affermò che non solo questo era possibile ma anzi era un modo di avvicinarsi al loro stile di apprendimento. Riflettendo sui modi in cui le persone affrontano una situazione problematica, il professore del MIT sostenne infatti che di fronte alla richiesta di risolvere un problema esistono sostanzialmente due modi prevalenti di agire: allestire un piano, è il caso delle persone che tendono a non mettersi subito al lavoro operativamente ma anzi investono la parte più consistente del compito a riflettere su come svolgerlo (i planners); o procedere per tentativi ed errori, è il caso di chi, al contrario, si butta subito nella parte operativa del compito per poi strutturare infine il piano (i bricoleurs). Il modo di apprendere dei bricoleurs è molto simile a quello dei bambini ed è caratterizzato da un procedere per gradi, da un avvicinarsi gradualmente alla meta risolvendo i sotto-problemi del problema e poi rimettendo tutto insieme. L'idea di imparare a programmare nasce, allora, in quanto la modalità di agire e apprendere dei bambini è simile a quella dei programmatori di computer, che procedono impostando inizialmente solo alcune linee che poi, per fasi successive, vengono modificate e arricchite.

Tenendo in considerazione quanto detto, al fine di introdurre la programmazione all'interno dell'ambito scolastico, Papert ideò e realizzò il linguaggio LOGO. Tale linguaggio, di tipo *interprete*⁴ (caratteristica importante in quanto aiuta ad individuare i possibili errori di un programma, indicando la linea problematica, durante la stesura del programma e non a

² “Bright new gadgets to teach the same old stuff in thinly disguised versions of the same old way” (trad. mia).

³ “Seems remarkably reluctant to use the computers for any purpose that fails to look very much like something that has been thought in schools for the past centuries” (trad. mia).

⁴ Interpreta le istruzioni impartite alla macchina una alla volta e ogni volta traduce un'istruzione, verifica la sua correttezza sintattica, la traduce e la esegue.

lavoro terminato) e basato sulla cosiddetta “geometria della tartaruga” (Papert, 1993), spinge in effetti ad un modo di lavorare molto simile a quello del bricolage. La forza di LOGO sta nel fatto che l’allievo riesce a “vedere” come egli pensa in quanto vede sia i suoi tentativi che poi l’intero programma, e quindi il computer diventa un potente mezzo per aiutare i bambini a pensare e a riflettere sul proprio modo di pensare. Nel corso degli anni Settanta il LOGO ha cominciato a diffondersi, soprattutto negli Stati Uniti, e un decennio dopo sono state realizzate molte esperienze anche in Italia. Tuttavia, quella rivoluzione culturale che il professore sudafricano auspicava non è arrivata, e in particolare ciò non si è verificato nelle scuole, dove il computer e i suoi utilizzi sono stati relegati a specifici momenti distanti dalle altre discipline e fuori della classe. A scuola l’introduzione del computer è stata interpretata come la necessità di costruire una didattica *dell’informatica*, piuttosto che una didattica *con l’informatica*.

Il lavoro di Papert, ad ogni modo, è stato continuato dallo stesso MIT, e in particolare da Mitchel Resnick, il quale, con i colleghi del gruppo di ricerca “Lifelong Kindergarten” del dipartimento di Media Laboratory del MIT, ha creato il linguaggio di programmazione Scratch, un vero e proprio linguaggio informatico ideato nello specifico per essere comprensibile e utilizzabile anche dai bambini (Resnick, 2013; Resnik et al., 2009). L’idea di fondo dei creatori di Scratch è che la programmazione favorisca importanti benefici quali: espansione del campo di cose che possono essere create e dei modi in cui possono essere create; possibilità di riflettere sulle strategie di problem solving; possibilità di suddividere la complessità in sotto-problemi più semplici; apprendimento per scoperta e per errori; collaborazione e, poiché la programmazione coinvolge la creazione di rappresentazioni esterne dei propri processi di problem solving, essa permette anche di riflettere sul proprio modo di pensare. La cornice teorica che fa da sfondo alla creazione di questo strumento, quindi, è quella del costruzionismo. La grammatica di Scratch è basata su una serie di blocchi di programmazione colorati che i bambini possono connettere per creare dei programmi più o meno complessi (dal semplice far muovere un personaggio alla creazione di storie o videogame) e sono facilitati in questo da connettori che suggeriscono come collegare i blocchi. I bambini possono sin da subito mettere insieme questi blocchi per vedere cosa succede, procedendo con un apprendimento per scoperta. La tartaruga pensata da Papert per LOGO come un oggetto attraverso cui pensare, a cui insegnare nuove cose, si trasforma su Scratch in *sprite*. Lo *sprite*, come prima lo era la tartaruga, è un oggetto digitale programmabile cui possiamo attribuire comportamenti e compiti diversi (Colombi, 2010), cui possiamo far assumere costumi e mettere in relazione con altri *sprite*; di diverso dalla tartaruga di LOGO ha certamente l’aspetto più accattivante e moderno. Come è evidente, la novità introdotta da Scratch non è da intendersi in senso assoluto (ibidem) ma certamente l’evidente ispirazione al modello dei mattoncini Lego, uno dei giochi più amati da generazioni di bambini, la sua intuitività e interattività, data dal fatto che cliccando su ogni blocco di programmazione si ha da subito la possibilità di vedere che cosa accadrà, e gli ottimi dati derivati dall’analisi dell’utilizzo sin qui fatto di tale linguaggio (Kafai & Burke, 2014; Peppler, Santo, Gresalfi, Tekinbas & Sweeney, 2014), porta a credere che il suo utilizzo nei processi di insegnamento-apprendimento possa costituire un’occasione importante.

Oltre alle possibilità fornite da Scratch e dalla comunità online ad esso connessa, che vanta poco meno di 10 milioni di utenti registrati e circa 13 milioni di progetti condivisi (<https://scratch.mit.edu/statistics/>), ad ogni modo, nell’ultimo anno in particolare si è registrato un notevole incremento di proposte relative a percorsi che offrano l’opportunità di svolgere attività di coding e/o di programmazione da poter utilizzare sia nell’ambito scolastico che extrascolastico. Alla campagna “The hour of code”, lanciata già nel 2013

per celebrare la settimana della “Computer Science Education” con l’intento di introdurre 10 milioni di studenti a un’ora di informatica dall’associazione pubblica “Code.org”, resa famosa anche e soprattutto dalla partecipazione del Presidente degli Stati Uniti Barack Obama (Mechaber, 2014), si sono aggiunte numerose altre iniziative nazionali e internazionali, proposte da organizzazioni, team di studiosi o ricercatori, etc., come ad esempio: “Programma il Futuro”, “Hello Processing!”, “Code Academy”, “Cody Roby”, “Hello Ruby”, “Code Avengers”, “Smart Coding”, “Tynker”, le attività proposte dai CoderDojo e moltissime altre.

L’attenzione verso questo tipo di attività ha destato l’interesse anche dei Governi che hanno deciso o stanno pensando di inserire il coding a scuola, nei curricoli come disciplina a se stante o come attività parascolastica. Ha scelto la prima opzione l’Inghilterra che dal settembre del 2014 ha inserito il coding come insegnamento obbligatorio per i bambini da 5 a 14 anni (Garotta, 2014) e la Finlandia che nelle Linee Guide che entreranno in vigore dal primo agosto 2016 introduce l’insegnamento delle basi di programmazione (Bertinetto, 2015). La Francia si sta rifacendo molto, invece, alla soluzione ideata in Italia dal MIUR, pensando di introdurre una iniziazione del coding come momento parascolastico facoltativo (Garotta, 2014). La volontà del MIUR di inserire all’interno delle scuole italiane questo tipo di attività nasce nel 2014 e viene ufficializzata prima nel documento “La buona scuola” (MIUR, 2014a) che recitava testualmente “serve quindi un piano nazionale che consenta di introdurre il coding (la programmazione) nella scuola italiana” e poi nella Legge 107/2015, all’art.1 comma 7, rimarcando l’importanza dello sviluppo del pensiero computazionale, che è fondante nelle attività di coding: “le istituzioni scolastiche [...] individuano il fabbisogno di posti dell’organico dell’autonomia [...] per il raggiungimento degli obiettivi formativi individuati come prioritari tra i seguenti: [...] h) sviluppo delle competenze digitali degli studenti, con particolare riguardo al pensiero computazionale, all’utilizzo critico e consapevole dei social network e dei media nonché alla produzione e ai legami con il mondo del lavoro”. In questo interesse manifestato dai Paesi europei evidente è l’influenza delle richieste mosse dal Parlamento Europeo (2006) agli Stati membri quando si inseriva la competenza digitale⁵ tra quelle competenze chiave per le persone nell’ambito delle strategie di apprendimento permanente. Il Piano Nazionale citato nel documento de “La buona scuola” si è concretizzato con la creazione del progetto triennale “Programma il Futuro” (MIUR, 2014b) che vede la collaborazione del MIUR con il Consorzio Interuniversitario Nazionale per l’Informatica (CINI) e con Code.org, cui “Programma il Futuro” rimanda per le proposte di attività. Il progetto nasce con “l’obiettivo di fornire alle scuole una serie di strumenti semplici, divertenti e facilmente accessibili per formare gli studenti ai concetti di base dell’informatica” (<http://www.programmailfuturo.it/>) e si pone come obiettivo finale quello di “coinvolgere il 40% delle scuole primarie (circa 6.100); completare il percorso base “L’Ora del Codice” in almeno il 25% delle classi delle scuole primarie; completare il percorso avanzato, o nella modalità interattiva o nella modalità ‘senza rete’, in almeno il 9% delle classi delle scuole primarie (circa 1380)” (ibidem). Le motivazioni che accompagnano l’ideazione del progetto fanno riferimento sia all’importanza dell’informatica e all’apporto che lo studio di questa disciplina potrebbe portare in una società, come quella attuale, “in cui la

⁵ “La competenza digitale consiste nel saper utilizzare con dimestichezza e spirito critico le tecnologie della società dell’informazione per il lavoro, il tempo libero e la comunicazione. Essa implica abilità di base nelle tecnologie dell’informazione e della comunicazione (TIC): l’uso del computer per reperire, valutare, conservare, produrre, presentare e scambiare informazioni nonché per comunicare e partecipare a reti collaborative tramite Internet.” (D.M. n. 254/2012)

tecnologia dipende in misura fondamentale dall'informatica e la presenza dei calcolatori (computer) diventa pervasiva", sia all'importanza del pensiero computazionale (Wing, 2006): "i benefici del 'pensiero computazionale' si estendono a tutte le professioni. Avvocati, insegnanti, dirigenti di azienda, psicologi, architetti, medici, funzionari di amministrazioni [...] ogni giorno devono affrontare problemi complessi; ipotizzare soluzioni che prevedono più fasi e la collaborazione con altri colleghi o collaboratori; formulare una descrizione chiara di cosa fare e quando farlo" (<http://www.programmailfuturo.it/>). Ad un anno e mezzo dall'iniziativa, i dati forniti dai tre monitoraggi del progetto sin qui fatti (MIUR & CINI, 2014; 2015a; 2015b) mostrano un'ottima partecipazione delle scuole (2.066), classi (16.336), insegnanti (4.966) e studenti (304.761), confermati dalle parole del Ministro dell'Istruzione Giannini che commenta i risultati definendoli come "una grande risposta che ci motiva ad andare avanti" (MIUR, 2015) e dai dati della partecipazione alla terza edizione della "Europe Code Week"⁶ che ha visto l'Italia in testa alla classifica dei Paesi più attivi nell'organizzazione di eventi (<http://codeweek.it/risultati-codeweek-2015/>).

3. La ricerca

Il progetto di ricerca "L'apprendimento di conoscenze geometriche e abilità visuo-spaziali attraverso il coding" nasce con l'intento di approfondire l'aspetto relativo all'apprendimento che le attività di coding possono o meno favorire poiché, sebbene siano state compiute osservazioni durante lo svolgimento di queste attività, esse erano relative all'ambito extrascolastico (le Computer Club House statunitensi) (Kafai & Burke, 2014; Papert, 1993; 1996; Peppler, et al., 2014), e sono pochi gli studi che si situano all'interno dell'ambito scolastico, sia all'estero che in Italia. In effetti, ad esempio, la proposta del MIUR e di "Programma il Futuro" ha fornito interessanti dati quantitativi rispetto a numero di partecipanti, scuole, genere, discipline coinvolte ma non ha dato conto né di descrizioni qualitative (come sono state introdotte queste attività? In che contesto: ricreazione, attività libera, ...? Quale ruolo hanno assunto insegnanti e bambini? etc.) né di riflessioni concernenti gli ambiti disciplinari presenti nelle scuole interessate (sono state strutturate come attività a parte? O inserite all'interno di un curriculum disciplinare? In caso, ci sono state modifiche interessanti rispetto all'insegnamento "tradizionale"? etc.).

La scelta di inserire attività di coding all'interno del curriculum di geometria è dovuta a più motivi. Innanzitutto, al fatto che per molto tempo questa disciplina ha rappresentato uno dei campi del sapere più importanti della matematica, al punto che i matematici stessi amavano definirsi geometri, ma al giorno d'oggi essa risulta piuttosto trascurata nella didattica e, a dimostrazione di questo fatto, occupa solo una piccola sezione dei libri di testo di matematica (Mammarella, Todeschini, Englaro, Lucangeli & Cornoldi, 2012). In seconda istanza, vi è la constatazione che oltre al poco spazio riservato nella didattica, esso è anche caratterizzato da approcci che tendono a presentare sin dai primi anni di scuola concetti astratti, distanti dall'esperienza dei bambini, che risultano pertanto poco significativi.

I pochi studi che testimoniano l'introduzione di attività di coding nei curricula di geometria (Calder, 2010; Smith & Neumann, 2014) mostrano come l'utilizzo di Scratch

⁶ Iniziativa promossa da Neelie Kroes' Young Advisors con il supporto di DG Connect alla Commissione Europea (<http://codeweek.eu/about/#resources>).

combinato anche con la messa in scena, la rappresentazione fisica da parte degli alunni, di determinati problemi concernenti concetti geometrici (movimenti, angoli e coordinate), e quindi la rappresentazione sia digitale che fisica, abbia favorito importanti riflessioni sui concetti stessi e di conseguenza anche una più profonda conoscenza degli stessi da un punto di vista cognitivo. La proposta di attività di coding vuole quindi sia fornire un contributo ai pochi studi presenti che offrire una sorta di riscoperta della geometria, in modo che possa essere affrontata avvalendosi di una vasta gamma di strumenti differenti e attraverso una metodologia che connetta il più possibile il vissuto reale dei bambini ai concetti astratti che sono soliti affrontare con una didattica più tradizionale, e dedicare una doverosa attenzione alle abilità visuo-spaziali. L'interesse per queste abilità, non definibili con un concetto unitario, ma distinguibili in diversi fattori (Cornoldi & Vecchi, 2003), è dovuto all'ovvia relazione che esse hanno con la geometria: "se pensiamo al modo in cui opera la nostra mente per risolvere un problema di geometria, ci rendiamo conto che è necessario rappresentarsi correttamente la figura di cui si sta parlando e, alle volte, compiere su di essa alcune trasformazioni per giungere al risultato corretto" (Mammarella et al., 2012). La scuola, troppo spesso, favorisce quasi unicamente l'apprendimento di abilità linguistiche, mentre tende a sottovalutare quelle non verbali, in particolare le visuo-spaziali, non tenendo conto di importanti ricerche (Hannafin, 2004; Hannafin, Vermillion, Truxau & Liu, 2008) che mostrano proprio il ruolo cruciale delle abilità visuo-spaziali nell'apprendimento, in particolare della geometria.

Nel progetto di ricerca, si cercherà quindi di capire se inserire attività di coding e programmazione all'interno del percorso di insegnamento-apprendimento di geometria di una specifica classe di scuola primaria, la terza, favorisca l'apprendimento di conoscenze geometriche e abilità visuo-spaziali ritenute adeguate al grado di scuola e classe, facendo riferimento alle Indicazioni Nazionali per il Curricolo (MIUR, 2012). I contenuti delle attività di coding/geometria si articoleranno intorno a quattro tematiche:

- orientamento spaziale: posizioni, coordinate, movimenti e relazioni spaziali;
- percorsi: costruire percorsi, dare e seguire istruzioni in un labirinto in modi differenti;
- figure geometriche: riconoscimento, descrizione e rappresentazione di figure geometriche;
- composizione/decomposizione delle figure.

Nello strutturare le attività, elemento centrale sarà l'integrazione di strumenti e linguaggi eterogenei: percorsi fisici da seguire o strutturare, dando istruzioni ad un compagno o ad un robot (BeeBot); programmazione su carta a quadretti; Tangram; attività su piattaforme online (Code.org), costruzione di progetti con Scratch; utilizzo di linguaggio verbale, non verbale e paraverbale. La didattica proposta sarà per problemi e laboratoriale, ovvero una didattica che vede gli insegnanti assumere il ruolo di tutor in quanto i veri protagonisti delle attività sono i bambini, chiamati a lavorare individualmente, in coppie e in gruppi più numerosi alla costruzione di progetti sia digitali che non.

Il progetto cercherà quindi di rispondere alla domanda: il coding favorisce l'apprendimento di conoscenze geometriche e abilità visuo-spaziali? Nel farlo, oltre a dar conto in termini quantitativi del quesito posto, verranno osservati proprio quei processi che vedono come protagonisti bambini e insegnanti.

L'obiettivo della ricerca è quello di mettere in relazione i due aspetti problematici individuati: competenza digitale, in particolare il saper utilizzare con dimestichezza e spirito critico le tecnologie, calato in questo contesto come il saperle programmare, e apprendimento di conoscenze geometriche, in particolare: conoscenza del significato di termini specifici (lessico) e proprietà delle figure. Nello specifico, attraverso questa ricerca,

si vorrebbe osservare se nelle classi in cui sono state introdotte attività di coding nei curricoli di geometria si verifichi un miglioramento in termini di conoscenze geometriche (lessico e proprietà delle figure) e abilità visuo-spaziali (composizione/scomposizione di figure, sviluppo di figure solide, stima del volume, figure nascoste e intersezione di figure) rispetto alle classi in cui questo tipo di attività non sono state inserite e se, ed eventualmente come, nelle classi in cui vengono condotte le attività di coding si modificano il setting di classe (prevalenza di situazioni in cui i banchi sono disposti singolarmente per il lavoro individuale? Oppure a isole/coppie, per favorire il lavoro di gruppo? Cattedra/lavagna di fronte ai banchi disposti in file?) e i processi che in essa avvengono (posizioni e ruoli assunti da insegnanti e studenti a confronto di quelli di classi in cui non vengono svolte attività di coding). Si vorrebbe, inoltre, analizzare, attraverso i commenti liberi dei bambini, il gradimento delle attività di coding proposte e ricevere commenti e suggerimenti degli insegnanti.

3.1. Impianto della ricerca

Fare ricerca in educazione significa “designare una riflessione sul fatto educativo condotta con appropriato metodo scientifico” (Trincherò, 2002), pertanto un progetto di ricerca che si pone l’obiettivo di essere rigoroso deve senza alcun dubbio essere caratterizzato da una buona progettazione che non può permettersi di trascurare alcun aspetto e dalla volontà di vedere nella miglior chiarezza possibile il fenomeno, fatto, che si intende descrivere (Mortari, 2010). La ricerca progettata è una ricerca per esperimento in cui vengono impostati come fattori dipendenti le conoscenze geometriche (lessico e proprietà delle figure) e le abilità visuo-spaziali di bambini di classe terza, mentre come fattore indipendente l’attività di coding. Il coding fungerà quindi da stimolo sperimentale (o fattore sperimentale). Nella strutturazione del piano sperimentale è stato scelto di modificare, invertendo le prime due fasi, il classico schema del piano sperimentale a due gruppi, ovvero si è deciso di scegliere il campione (composto da gruppo sperimentale e gruppo di controllo) secondo un criterio di omogeneità relativamente ai valori del fattore dipendente prima dello stimolo sperimentale, e quindi si è scelto di formare il campione in base ai risultati di prove pre-test inerenti a conoscenze e abilità visuo-spaziali. Questa scelta, a differenza di quella prevista dalla struttura classica del piano sperimentale a due gruppi, permette di mettere in relazione non solo i valori del fattore dipendente prima e dopo lo stimolo sperimentale, sia nel gruppo sperimentale che in quello di controllo, ma anche, e soprattutto, confrontare i valori del fattore dipendente dopo lo stimolo sperimentale del gruppo sperimentale e del gruppo di controllo. Partendo dallo stesso livello, infatti, l’intento è proprio quello di capire se e come il coding abbia fatto la differenza.

La ricerca coinvolge studenti di classe terza di scuola primaria in quanto per quell’anno sono fissati i traguardi di sviluppo delle competenze e poiché esiste uno strumento, “Geometria Test” (Mammarella et al., 2012), che offre prove standardizzate per la valutazione delle conoscenze geometriche e delle abilità visuo-spaziali per le classi seconda e terza della scuola primaria.

Per quanto concerne i metodi di indagine, la scelta di utilizzare sia metodi quantitativi che qualitativi è dovuta alla convinzione che, non esistendo una realtà oggettiva, per “avere un quadro di indizi più ampio possibile” (Trincherò, 2002, p. 30) è necessario superare il dualismo tra questi due tipi di metodi solitamente associati prettamente ad approcci nomotetici o idiografici rispettivamente. La realtà viene vista in modo ecologico come un insieme strutturato di relazioni, in cui l’essenza di ciascun ente è data dalle relazioni dinamiche nelle quali esso è implicato e che contribuisce contestualmente a strutturare

(Mortari, 2010). Attraverso i diversi approcci epistemici, si cercherà, quindi, di analizzare in modo sia quantitativo che qualitativo le relazioni tra i fattori presi in esame. Il progetto ipotizzato, quindi, può essere definito “multimetodo” (Trincherò, 2002). Nello specifico, per raccogliere i dati quantitativi è stato scelto “Geometria Test” (Mammarella et al., 2012). Le prove di valutazione dell’apprendimento della geometria fanno riferimento a tre aree che vanno a costituire tre rispettivi subtest: conoscenze geometriche, problemi di geometria e abilità visuo-spaziali (ibidem). Per il pre-test e il post-test saranno somministrati i subtest relativi alle conoscenze geometriche e alle conoscenze visuo-spaziali. I dati qualitativi, invece, verranno raccolti da:

- valutazioni del percorso di formazione attraverso domande stimolo aperte rivolte agli insegnanti;
- focus group con gli insegnanti prima, durante e dopo la sperimentazione;
- valutazioni dagli studenti espresse sotto forma di commenti liberi riguardanti gradimento e soddisfazione delle attività proposte;
- monitoraggio delle attività tramite diario di bordo tenuto dagli insegnanti;
- commenti e suggerimenti liberi degli insegnanti.

Le fasi di lavoro pensate sono le seguenti:

1. somministrazione delle prove di pre-test: la ricerca è stata presentata alle scuole primarie dell’Umbria e le prove pre-test sono state effettuate su un campione iniziale di 47 classi terze interessate a prendere parte al progetto di ricerca;
2. selezione del campione costituito da gruppo sperimentale e gruppo di controllo: delle 47 classi, sono state individuate 20 classi secondo un criterio di omogeneità relativamente a conoscenze geometriche e abilità visuo-spaziali. Delle 20 classi componenti il campione, 10 sono andate a formare il gruppo di controllo e 10 quello sperimentale. Nella scelta dei gruppi, determinanti sono state le preferenze degli insegnanti coinvolti;
3. formazione degli insegnanti del gruppo sperimentale: la formazione degli insegnanti è relativa allo sviluppo del pensiero computazionale attraverso il coding e alla programmazione in generale e poi nello specifico ad alcuni strumenti attraverso cui proporre le attività, sia con dispositivi elettronici che in modalità unplugged. Riguarda inoltre i contenuti delle attività e i vari strumenti designati per monitorare le attività, con suggerimenti sulla costruzione delle lezioni e la cornice teorica di riferimento;
4. inizio della sperimentazione nel gruppo sperimentale e monitoraggio delle attività;
5. somministrazione delle prove di post-test sia alle classi sperimentali che di controllo;
6. confronto tra i risultati.

4. Conclusioni

La ricerca è ancora in corso, la fase di formazione degli insegnanti è appena terminata mentre quella sperimentale inizierà a giorni. Al momento, pertanto, non è possibile fornire risultati relativi all’apprendimento dei bambini o al monitoraggio delle attività nelle classi. Nella fase di formazione, gli insegnanti hanno lavorato in prima persona sul pensiero computazionale grazie ad attività di coding e programmazione, utilizzando Scratch, i corsi di Code.org, il BeeBot ma anche su carta e allo stesso tempo hanno avuto modo di ripensare il curriculum di geometria alla luce di queste nuove attività. Gli insegnanti partecipanti hanno

costruito una rete, grazie anche alla creazione di un corso su Edmodo, in cui condividere risorse, esperienze e riflessioni, in cui lavorare individualmente e in gruppo.

La proposta di formazione è stata accolta in modo positivo dagli insegnanti che, alla domanda sugli aspetti del corso che ritenevano più significativi, hanno messo in luce come gli spunti pratici e la possibilità di essere stati coinvolti in prima persona abbia permesso loro di poter riproporre in classe le attività. Gli aspetti che riterrebbero utili approfondire risultano essere collegati alla volontà di raggiungere una maggiore padronanza di Scratch.

Alcuni insegnanti hanno riferito di aver già iniziato a proporre alcune attività di base agli studenti che hanno risposto alla proposta con molto interesse.

Bibliografia

- Bertinetto, C. (2015). Finlandia: ecco le nuove linee guida per la scuola. *Aula di Scienze*. Zanichelli. <http://aulascienze.scuola.zanichelli.it/come-te-lo-spiego/2015/02/09/finlandia-ecco-le-nuove-linee-guida-per-la-scuola/> (ver. 15.04.2016).
- Calder, N. (2010). Using Scratch: an integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, 15(4), 9–14.
- Code.org. <https://code.org/> (ver. 15.04.2016).
- Code Academy. <https://www.codecademy.com/> (ver. 15.04.2016).
- Code Avengers. <https://www.codeavengers.com/> (ver. 15.04.2016).
- Cody Roby. <http://codeweek.it/cody-roby/> (ver. 15.04.2016).
- Colombi, A.E. (2010). *Immagina, programma e condividi con Scratch. Con CD-ROM*. Trento: Erickson.
- Computer Science Education Week. <https://csedweek.org/> (ver. 15.04.2016).
- Cornoldi, C., & Vecchi, T. (2003). *Visuo-spatial working memory and individual differences*. Hove and New York, NY: Psychology Press.
- CS Unplugged. <http://csunplugged.org/> (ver. 15.04.2016).
- Decreto Ministeriale 16 novembre 2012, n. 254. *Regolamento recante indicazioni nazionali per il curriculum della scuola dell'infanzia e del primo ciclo d'istruzione, a norma dell'articolo 1, comma 4, del decreto del Presidente della Repubblica 20 marzo 2009, n. 89*.
- Edmodo. <https://www.edmodo.com/?language=it> (ver. 15.04.2016).
- Garotta, G. (2014). Coding e produzione digitale. *Scuola dell'Infanzia*. <http://www.giuntiscuola.it/scuoladellinfanzia/magazine/opinioni/piano-scuola-discutiamone-qui/coding-e-produzione-digitale/> (ver. 15.04.2016).
- Hannafin, R.D. (2004). Achievement differences in structured versus unstructured instructional geometry programs. *Educational Technology Research & Development*, 52, 19–32.

- Hannafin, R.D., Vermillion, J.R., Truxau, M.P., & Liu, Y. (2008). Effects of spatial ability and instructional program on geometry achievement. *The Journal of Educational Research*, 101, 148–156.
- Hello Processing. <http://hello.processing.org/> (ver. 15.04.2016).
- Hello Ruby. <http://www.helloruby.com/> (ver. 15.04.2016).
- Kafai, Y.B., & Burke, Q. (2014). *Connected code: why children need to learn programming*. MIT Press.
- Legge 13 luglio 2015, n. 107. *Riforma del sistema nazionale di istruzione e formazione e delega per il riordino delle disposizioni legislative vigenti*.
- Mammarella, I.C., Todeschini, M., Englaro, G., Lucangeli, D., & Cornoldi, C. (2012). *Geometriatest. Prove di valutazione per la scuola primaria e secondaria di primo grado. Con DVD*. Trento: Erickson.
- Mechaber, E. (10 dicembre 2014). *President Obama is the first President to write a line of code*. Blog. <https://www.whitehouse.gov/blog/2014/12/10/president-obama-first-president-write-line-code> (ver. 15.04.2016).
- MIUR. Ministero dell'Istruzione, dell'Università e della Ricerca (2012). Indicazioni nazionali per il curriculum della scuola dell'infanzia e del primo ciclo d'istruzione. *Annali della Pubblica Istruzione*. No. Speciale.
- MIUR. Ministero dell'Istruzione, dell'Università e della Ricerca (2014a). *La buona scuola. La consultazione*. http://www.istruzione.it/allegati/2014/focus151214_all1.pdf (ver. 15.04.2016).
- MIUR. Ministero dell'Istruzione, dell'Università e della Ricerca (2014b). *Nota Prot. 2937: Iniziativa "Programma Il Futuro"*. http://www.istruzione.it/allegati/2014/iniziativa_programma_il_futuro.pdf (ver. 15.04.2016).
- MIUR. Ministero dell'Istruzione, dell'Università e della Ricerca (2015). *"Programma il Futuro", oltre 300.000 gli studenti protagonisti del #coding*. <http://hubmiur.pubblica.istruzione.it/web/ministero/cs260515> (ver. 15.04.2016).
- MIUR, & CINI. (2014). *Programma il Futuro*. Cartella Stampa http://www.programmailfuturo.it/media/docs/lancio/PRESS%20KIT_2014_programmailfuturo.pdf (ver. 15.04.2016).
- MIUR, & CINI. (2015a). *Programma il Futuro. Il bilancio del primo anno di sperimentazione del progetto*. http://www.istruzione.it/allegati/2015/25052015_PROGRAMMA_IL_FUTURO.pdf (ver. 15.04.2016).
- MIUR, & CINI. (2015b). *Programma il Futuro. Monitoraggio dell'andamento del progetto*. <http://www.programmailfuturo.it/media/docs/Rapporto-monitoraggio-settembre-2014-gennaio-2015.pdf> (ver. 15.04.2016).
- Mortari, L. (2010). Cercare il rigore metodologico per una ricerca pedagogica scientificamente fondata. *Education Sciences & Society*, 1(1), 143–156.
- OECD. Organization for Economic Co-operation and Development (2015). *Students, computers and learning: making the connection*. PISA, OECD Publishing.

- Raccomandazione 2006/962/CE del Parlamento Europeo e del Consiglio, 18 dicembre 2006. *Competenze chiave per l'apprendimento permanente*. <http://eur-lex.europa.eu/legal-content/IT/TXT/PDF/?uri=CELEX:32006H0962&from=IT> (ver. 15.04.2016).
- Papert, S. (1993). *Mindstorms: Children, Computers, And Powerful Ideas*. New York, NY: Basic Books Inc.
- Papert, S. (1996). *The connected family: bridging the digital generation gap*. Longstreet Press.
- Papert, S., & Solomon, C. (1971). *Twenty things to do with a computer*. Cambridge, MA: Massachusetts Institute of Technology, A.I. Laboratory.
- Peppler, K., Santo, R., Gresalfi, M., Tekinbas, K.S., & Sweeney, L.B. (2014). *Script changers: digital storytelling with Scratch*. MIT Press.
- Programma il Futuro. <http://www.programmailfuturo.it/> (ver. 15.04.2016).
- Resnick, M. (2013). *Learn to code, code to learn*. <http://web.media.mit.edu/~mres/papers/L2CC2L-handout.pdf> (ver. 15.04.2016).
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60–67. <http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf> (ver. 15.04.2016).
- Smart Coding. <http://www.smart-coding.it/> (ver. 15.04.2016).
- Smith, C.P., & Neumann, M.D. (2014). Scratch it out! Enhancing geometrical understanding. *Teaching Children Mathematics*, 21(3), 185–188. <http://doi.org/10.5951/teachilmath.21.3.0185> (ver. 15.04.2016).
- Teaching London Computing. <https://teachinglondoncomputing.org/> (ver. 15.04.2016).
- Trincherò, R. (2002). *Manuale di ricerca educativa*. Milano: Franco Angeli.
- Tynker. <https://www.tynker.com/> (ver. 15.04.2016).
- Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <http://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf> (ver. 15.04.2016).