

S.P.Q.Robot

Relazione prima unità didattica

*METHODS IN COMPUTER SCIENCE
EDUCATION: ANALYSIS*

Andrea Rosati
1595834

Federica Spini
1588482

Revision History

Revision	Date	Author(s)	Description
1.0	30.04.19	AR, FS	Prima stesura documento

Indice

1	Introduzione	3
2	A chi ci rivolgiamo per questa lezione	4
2.1	Pre-requisiti didattici	5
2.2	Pre-requisiti informatici	5
2.3	Obiettivi didattici	5
2.4	Tempo richiesto per l'attività	6
3	Materiale	6
3.1	Lego Ev3	6
3.2	Limitazioni Lego Ev3	7
3.2.1	Limitazioni per il display e l'audio	7
3.2.2	Il problema di avere un solo sensore del colore	7
3.2.3	Affidabilità del sensore di riconoscimento dei colori	7
3.2.4	Il problema delle combinazioni per i pattern e come riconoscere le città	8
4	La lezione	9
4.1	Fase introduttiva	9
4.2	Software già pronto per i ragazzi	10
4.3	Cosa i ragazzi devono implementare	10
4.3.1	Task 1	10
4.3.2	Task 2	11
4.3.3	Task 3	11
4.3.4	Task 4	11
4.3.5	Task5	11
4.4	Valutazione	12
5	Altre possibilità per riadattare la lezione	12
5.1	Riadattare l'attività ad altri eventi storici o materie.	12
5.2	Provare a far fare il pattern recognition o altre parti a ragazzi più grandi.	13

Introduzione

S.P.Q.Robot consiste nel far camminare il robottino EV3 su una cartina, seguendo dei percorsi colorati in nero. Lungo i percorsi disegnati possono essere messe delle sequenze di colori (pattern) che rappresentano degli eventi in base ai quali il robottino, riconoscendo la sequenza, può avere varie reazioni. Quest'idea ci è piaciuta per diversi motivi:

Punti forza

- Far muovere un robot è un modo coinvolgente di approcciare la programmazione
- Si tratta di un progetto che permette di interagire davvero molto bene con diversi ambiti disciplinari, come la storia, la geografia o la letteratura
- Il linguaggio a blocchi messo a disposizione sul sito lego è più sintetico rispetto a Scratch o ad altri linguaggi simili, perché progettato appositamente per l'EV3 e il minor numero di comandi può essere vantaggioso con le fasce di età più giovani di studenti, perché meno confusionario
- Nel caso di un progetto collaborativo tra più gruppi di studenti, essendo il linguaggio visibile o a blocchi o come programma Javascript, è possibile fare un copia incolla del codice per unire i codici dei ragazzi

L'ispirazione ci è venuta pensando agli ozobot: si tratta di robottini utilizzati in ambito educativo, che hanno la capacità di seguire una linea nera e riconoscere dei pattern colorati. Nel loro caso le azioni che possono eseguire una volta riconosciuto un pattern sono già prestabilite e l'attività per i ragazzi sta nel disegnare percorsi con pattern (di fatto il target degli ozobot è tra i 4 e i 7 anni d'età). Nel nostro caso abbiamo voluto invertire questa logica e quindi i ragazzi troveranno già pronto il percorso con i pattern, ma dovranno programmare loro cosa far fare al robot una volta che questo avrà riconosciuto il pattern.



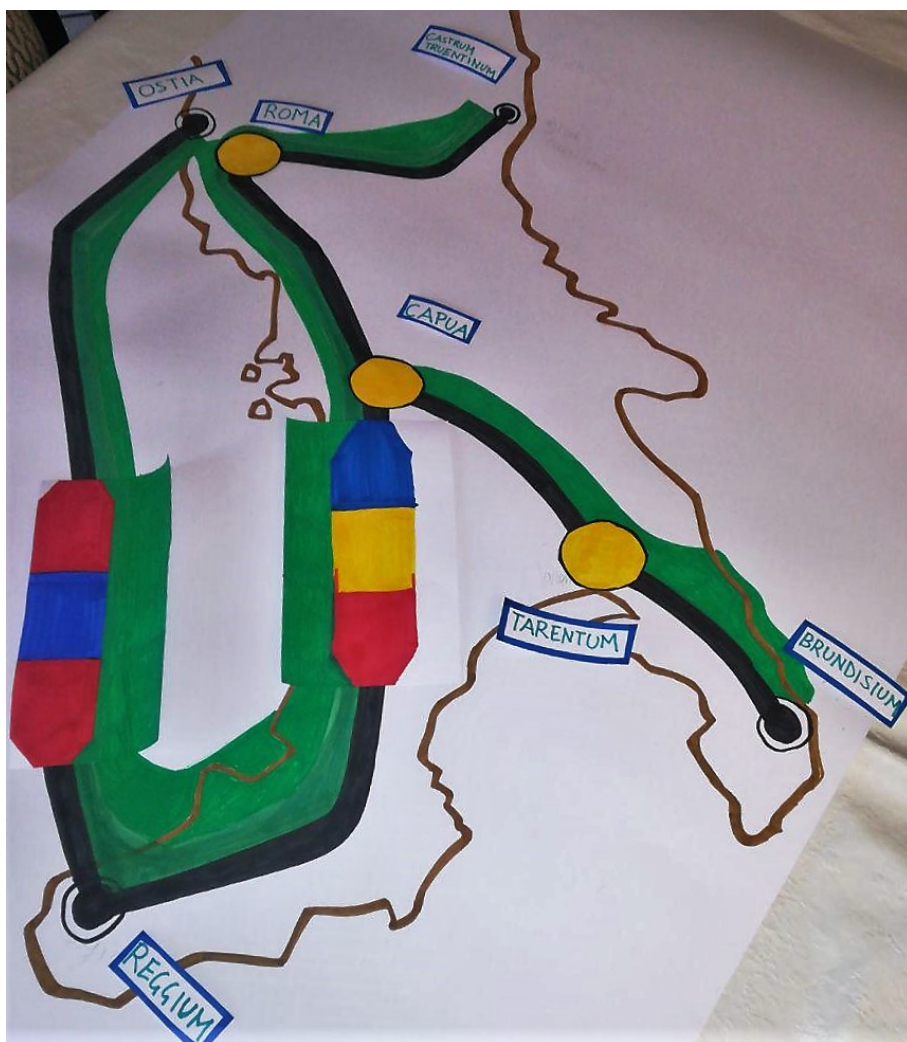


Figura 1: Il disegno della cartina che abbiamo realizzato

2

A chi ci rivolgiamo per questa lezione

Il progetto che abbiamo realizzato è pensato per bambini che vanno dalla terza alla quinta elementare, anche se la quinta è perfetta perché è l'anno in cui i bambini studiano l'Italia in geografia e l'antica Roma in storia. Anche la struttura con cui i bambini devono usare il robot è pensata per una fascia d'età bassa: abbiamo scritto noi delle funzioni che loro devono usare, insieme ad altri comandi base per programmare il robottino.

2.1 Pre-requisiti didattici

I prerequisiti non sono necessari per svolgere l'attività, i concetti che servono possono essere "raccontati" sul momento. Nel caso in cui però i bambini abbiano delle conoscenze è bello che queste siano valorizzate ed è anche per questo che abbiamo deciso di fare la cartina muta. Nozioni che possono essere apprezzate sono:

- Essere in grado di riconoscere elementi base della geografia italiana (i mari, le montagne, le regioni e le province)
- Conoscere i concetti di punti cardinali e distanza in scala
- Conoscere il contesto storico della Roma imperiale
- Riconoscere strade romane come strade attuali (magari spiegando che se sono stati nei luoghi indicati sicuramente le hanno percorse in macchina con i genitori) ed essere a conoscenza dell'importanza delle infrastrutture che caratterizza la civiltà romana. Dei cartellini con i nomi propri delle varie cose da riconoscere sono posti sulla cartina prima di iniziare l'attività di programmazione.

2.2 Pre-requisiti informatici

Questa learning unit non richiede particolari conoscenze informatiche e potrebbe anche essere utilizzata come prima lezione. Ovviamente una conoscenza base di un linguaggio a blocchi come Scratch renderebbe molto più rapido lo svolgimento dell'attività e ci permetterebbe di concentrarsi di più sui compiti più avanzati che sono previsti, mentre per una fascia di età molto bassa non è garantito di arrivare a realizzare i compiti più complessi (riconoscere un pattern e programmare una "reazione" all'evento associato).

2.3 Obiettivi didattici

La cartina che abbiamo disegnato per il progetto rappresenta l'Italia centro-meridionale. I bambini studiano la storia e la civiltà dell'antica Roma, che viene associata alle grandi infrastrutture che collegavano città e province all'interno del territorio di Roma. Al contempo, considerando che i bambini di quinta elementare studiano anche l'Italia con regioni e province, è carino far vedere che molte strade e province romane sono le stesse che esistono tutt'oggi. Non è così necessario che i bambini siano ferrati nella parte dei prerequisiti, è però importante che dei concetti culturali possano restare impressi per quanto riguarda l'ambito storico-geografico.

Dal punto di vista informatico possiamo considerare questa learning-unit come una lezione introduttiva ai linguaggi a blocchi: i ragazzi trovano vuote e devono riempirle per fare reagire nel modo più appropriato il robot per il specifico evento. Possiamo quindi elencare gli obiettivi principali in questo elenco:

Obiettivi didattici informatici

- Capire il concetto di istruzione e di programma come sequenza di istruzioni
- Capire il concetto di funzione
- Far eseguire delle istruzioni base al robot, sia del linguaggio a blocchi (come mostrare una scritta o un'immagine sul display dell'EV3), sia create da noi (come partire e percorrere la linea nera) riempiendo una funzione
- Utilizzare l'if

2.4 Tempo richiesto per l'attività

Secondo noi, senza prerequisiti ci vogliono in media 4/5 ore di lavoro per fare in modo che i bambini completino tutto. Potrebbero essere divise in 3 lezioni da 2 ore, dove sei ore diventano necessarie anche per ricordare ogni volta i concetti appresi nelle lezioni precedenti, oppure mezza giornata di scuola dedicata a questo. Ovviamente se i bambini hanno già confidenza con i linguaggi a blocchi il numero di ore si ridurrebbe a 3.

3

Materiale

Per questa learning unit c'è bisogno di un robottino programmabile che abbia uno o due sensori di riconoscimento dei colori e di una mappa con disegnati i percorsi e gli eventi da far programmare ai ragazzi. Ovviamente, per programmare il robottino, serve un dispositivo in grado di supportarne il linguaggio ovvero che possa andare su internet. Il linguaggio infatti è disponibile solamente online, quindi un'altra grande necessità per la corretta realizzazione dell'unità didattica è la presenza di connessione internet. Tramite il dispositivo utilizzato per programmare inoltre deve essere possibile caricare il codice sul robot, quindi deve essere munito di Bluetooth o di una porta USB.

3.1 Lego Ev3

Il lego Ev3 della serie mindstorm rientra perfettamente nelle nostre specifiche poiché è programmabile tramite il sistema microsoft makecode(c'è sia un'interfaccia a blocchi che uno in codice Javascript) e soprattutto l'hardware è modulare e tra i vari moduli disponibili c'è il sensore di riconoscimento dei colori e quindi si possono aggiungere uno o due sensori di questi.

3.2 Limitazioni Lego Ev3

3.2.1 Limitazioni per il display e l'audio

Il Lego EV3 tra le tante funzionalità possiede sia un display lcd e sia un altoparlante. Nel display LCD non soltanto è possibile scriverci valori e stringhe ma si può far stampare a schermo dei disegni, purtroppo però non è possibile aggiungere disegni alla libreria preimpostata dei disegni. Lo stesso discorso vale per l'altoparlante che può eseguire effetti sonori (versi di animale, voci che parlano ecc...) senza poterne aggiungere di propri. Queste limitazioni hanno limitato la nostra creatività nell'inventare eventi poichè effetti sonori come il classico saluto romano *AVE CESARE* sarebbero sicuramente tornati utili.

3.2.2 Il problema di avere un solo sensore del colore

Se si dispone soltanto di un sensore di riconoscimento dei colori far seguire una linea al robot non è banale: infatti il robot avendo un unico "occhio" non può capire se per rientrare in carreggiata (magari perché va leggermente storto o la strada curva) deve girare verso destra o sinistra. Per non fare le curve in un unico senso avremmo dovuto farlo oscillare per ritrovare il nero, una strategia che esteticamente sarebbe stata poco funzionale e poco accattivante. Per ovviare a questo problema abbiamo aggiunto una riga verde attaccata alla riga nera, in modo così da permettere al robot di capire se sta uscendo dal lato destro o sinistro della linea, e quindi di applicare la giusta correzione. Nel nostro codice offriamo una funzione che permette di impostare se il verde e il bianco devono essere rispettivamente sulla destra o sulla sinistra del percorso, così che le strade possono essere percorse in entrambe le direzioni. Ovviamente con due sensori del colore questo problema non ci sarebbe stato con una strada sufficientemente larga: se il nero fosse scomparso da uno solo dei due, sarebbe stato sufficiente svoltare verso la direzione del nero e ci sarebbero stati meno vincoli anche nel disegno della cartina.

3.2.3 Affidabilità del sensore di riconoscimento dei colori

Abbiamo riscontrato che il sensore di riconoscimento dei colori non sempre funziona come dovrebbe e capita che restituisca come risultato di una scansione un colore diverso da quello aspettato. Dopo aver eseguito vari test[2] abbiamo scoperto che questi errori sono sia dovuti all'illuminazione ambientale della stanza, ma anche dalla scelta di tonalità di colore e che alcuni di essi sono più sensibili all'errore come il verde che viene spesso confuso con il blu. Il test ci ha quindi permesso di scegliere le tonalità di colore meno inclini all'error



Figura 2: Alcuni dei test dei colori

3.2.4 Il problema delle combinazioni per i pattern e come riconoscere le città

Per semplicità abbiamo scelto di combinare tre colori (rosso, giallo e blu) in sequenze che sono sempre di tre colori. Il codice può essere ampliato e migliorato per permettere combinazioni di meno colori, ma il numero che ne avevamo a disposizione era sufficiente. Tuttavia bisogna notare che combinazioni come rosso-rosso-blu devono essere evitate: mettere un separatore tra i colori renderebbe i pattern troppo lunghi, quindi è il cambio di colore a far capire che siamo in presenza del prossimo colore di un pattern. I pattern iniziano e finiscono, ovviamente, con il nero della strada.

Per riconoscere quando il robot raggiunge una città abbiamo scartato l'ipotesi di utilizzare i pattern poiché per rendere il percorso percorribile in ogni verso il robot ritroverebbe il pattern della città riflesso non appena esce fuori dalla città (alcuni pattern riflessi sono identici a loro stessi) [3]. Abbiamo deciso quindi di rendere le città gialle in modo così da evitare questa problematica, questa soluzione però implica un nuovo vincolo sui pattern e cioè che non possono iniziare con il colore giallo.



Figura 3: Riconoscere città

4

La lezione

4.1 Fase introduttiva

Per iniziare viene spiegato ai ragazzi, in modo interattivo e facendo loro domande su cosa pensano che stiano per fare, su cos'è che vedono sulla cartina: si tenta di far indovinare quali siano le città presenti (Roma, Ostia, Brindisi e Reggio Calabria) e gli si spiega che sono presenti anche la città di Capua e di Castrum Truentinum. Gli si racconta che le strade che vedono si chiamano Appia, Salaria e Popilia, con cenni storici minimi si racconta perché erano così importanti insieme ad altre strade. Dopo questa introduzione si prende il robot che in corrispondenza del sensore per rilevare i colori ha il disegno di un carro e si racconta come questo carro debba attraversare le strade capendo dove si trova in base ai colori.

4.2 Software già pronto per i ragazzi

Una volta fatto collegare il robot con il computer si apre il programma con il linguaggio a blocchi e lo si fa vedere ai ragazzi. Gli si spiega che:

- Ogni blocco è un'istruzione e che ogni istruzione può far fare qualcosa al robot.
- Spiegare che anche le funzioni sono sequenze di istruzioni e quando si chiama la funzione (ovvero l'istruzione è la funzione) bisogna eseguire tutte le istruzioni al suo interno
- Ci sono diversi blocchi vuoti da riempire che corrispondono alle sequenze di pattern, al viaggio e all'arrivo in una città, il loro compito è mettere le istruzioni che hanno a disposizione all'interno di queste funzioni per riuscire a svolgere i compiti richiesti

Si specifica molteplici volte, essendo piccoli, che loro possono esclusivamente aggiungere blocchi nelle funzioni "vuote" e che se cambiano qualcosa di diverso perderanno punti nella valutazione.

Si presentano le varie istruzioni per avere la soluzione e i blocchi già pronti, descritti nella Tabella 2.

Tabella 2: Blocchi già pronti

Nome blocco	Descrizione
Scappa	Raddoppia la velocità del robot per 2 secondi
Arrivo	Ferma il robot e suona la fanfara
CambiaDirezione	il robot inverte la propria direzione invertendo cosa fare quando trova il verde o il bianco
Accendi fuoco	Fissa il led sopra il robot al colore rosso e disegna un fuoco sul display

Oltre a questi blocchi i ragazzi trovano anche una variabile *città-visitate* che mantiene il conteggio delle città che il robot ha visitato.

4.3 Cosa i ragazzi devono implementare

I ragazzi avranno diversi task da svolgere di difficoltà crescente, tutti contestualizzati nell'ambito dell'antica Roma e legati ad aneddoti storici realistici, che saranno maggiori rispetto alla breve descrizione qui fornita.

4.3.1 Task 1

Il compito è consegnare un carico di sale da Castrum Truentinum fino a Roma. La via Salaria ha questo nome proprio perché vi si trasportava il sale.

I bambini dovranno usare correttamente la funzione partenza ed arrivo, posizionando il robot e il pallino giallo che rappresenta le città nel modo giusto, dimostrando così di aver capito dov'è il rilevatore dei colori.

4.3.2 Task 2

Ora che il mercante romano ha ricevuto il carico di sale, vuole riinvviare delle merci fenicie come pagamento a Castrum Truentinum.

I bambini dovranno fare la stessa funzione di prima, ma utilizzando la funzione che inverte il verde e il bianco come riferimenti che indicano se la strada si sta perdendo da destra o da sinistra.

4.3.3 Task 3

Adesso da Brundisium, uno dei porti più grandi dell'antica Roma, che commerciava con la Grecia e con l'Oriente, bisogna percorrere la via Appia, la "Regina viarum" fino a Roma.

I bambini devono dimostrare di saper leggere il contenuto della variabile *città-visitate* e quindi scrivere il corpo della funzione *sei_in_città* considerando i casi di ogni città. La terza città è l'obiettivo, alla seconda saranno in presenza di un incrocio e dovranno invertire correttamente il bianco e il verde come destra e sinistra (utilizzo dell'if).

4.3.4 Task 4

Stavolta il viaggio verrà fatto sia di andata che di ritorno: bisogna partire da Ostia, uno dei più antichi porti romani, arrivando via mare a Regium. Lì si percorrerà la via Popilia fino a Capua e poi si arriverà a Roma.

Rispetto al percorso precedente i bambini troveranno i pattern. Non dovranno più riempire solo la funzione "viaggia", ma anche le funzioni "RossoBlu-Rosso" e "RossoGialloBlu". Dovranno chiamare la funzione "scappa" quando incontreranno il pattern che rappresenta i pirati, dovranno chiamare la reazione "accendi il fuoco" quando incontreranno il pattern che rappresenta le bestie feroci per superare il Task (sarà superato anche se il robot avrà altre reazioni coerenti con l'evento).

4.3.5 Task5

Questo task è libero e verrà effettuato solo con bambini che riescono a completare correttamente i compiti precedenti: gli si dà la possibilità di far viaggiare il robottino a piacere modificando più liberamente le funzioni "partenza", "arrivo", "scappa", "accendi il fuoco" e offrendo anche la possibilità di definire nuove funzioni. , Potranno utilizzare suoni, luci, scritte ed immagini o altre loro richieste realizzabili in poco tempo.

4.4 Valutazione

Trattandosi di bambini abbiamo deciso di non usare voti, ma giudizi: per ciascuno dei seguenti punti verrà dato un voto tra Insufficiente, Sufficiente, Buono, Distinto, Ottimo ed Eccellente (Insufficiente ed Eccellente devono essere molto rari). I punti di valutazione sono:

Punti di valutazione

- Raggiungimento degli obiettivi dei task
- Comprensione dei concetti di logica ed informatica
- Coinvolgimento e creatività

5

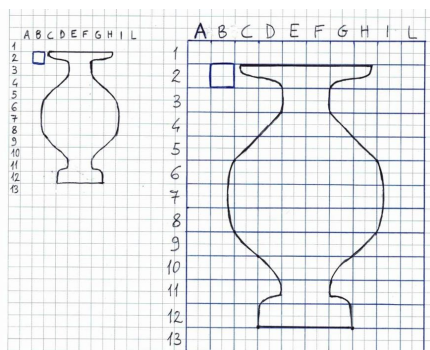
Altre possibilità per riadattare la lezione

5.1 Riadattare l'attività ad altri eventi storici o materie.

A noi sono venuti in mente molto argomenti da affrontare con lo stesso, di diverse materie: viaggi di Colombo, Magellano e Vespucci, viaggi di Ulisse, Marco Polo, itinerari di guerre, come quelle Puniche o Napoleoniche, o persino moti di pianeti entrando nell'ambito scientifico.

Non è difficile pensare alla possibilità di creare attività in cui, divisi in gruppi, ai bambini venga affidato il compito di disegnare le cartine; viene fatto fare infatti, in ambito di arte, sia alle

elementari che alle medie, le scale di ingrandimento tramite la quadrettatura. Individuando tanti itinerari, uno per ogni gruppo di bambini, potrebbero fare loro stessi delle ricerche al riguardo e ideare ognuno un percorso da far percorrere al robot. Quindi come lavoro è davvero adatto ad essere la base per progetti più ampi ed interdisciplinari, che ovviamente sarebbero possibili solo con sufficiente tempo da dedicare e la collaborazione di diversi insegnanti.



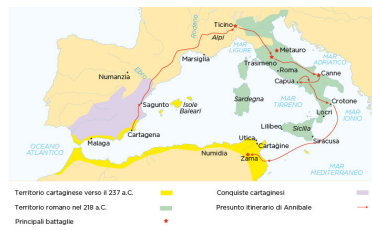


Figura 4: Possibili percorsi

5.2 Provare a far fare il pattern recognition o altre parti a ragazzi più grandi.

Lo stesso progetto potrebbe essere realizzato da ragazzi delle scuole medie in cui non devono più utilizzare librerie preparate che semplifichino la logica dei programmi, ma dovranno scrivere loro il codice, ovviamente in un crescendo di difficoltà delle richieste, pari a quello fatto per i bambini.