Claudio Penna

SCRATCH A SCUOLA

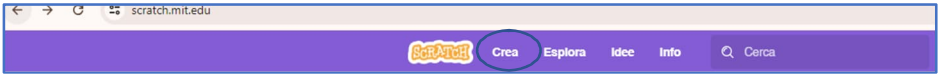
MATEMATICA PER LA SCUOLA PRIMARIA

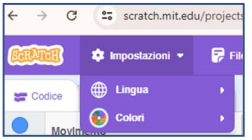
L'AMBIENTE SCRATCH

Prima di utilizzare Scratch per i nostri progetti, quache informazione riguardante Scratch.

Scratch online

Si può lavorare con Scratch direttamente online, senza scaricare nulla sul proprio computer/tablet andando sul sito ufficiale:
https://scratch.mit.edu/





Entrati nel sito, occorre cliccare sul pulsante del menu Crea per poter iniziare a creare i progetti.
Entrati nell'editor, cliccando sul pulsante delle Impostazioni si potrà scegliere la lingua di utilizzo fra le numerose lingue disponibili.

Scratch offline

Entrando in https://scratch.mit.edu/download si potrà scaricare l'App gratuita di Scratch Desktop per computer/tablet che abbiano come sistemi operativi *Windows*, *macOS*, *ChomeOS*, *Android* (per quanto riguarda il sistema *Android*, l'App funziona sui tablet, ma non sugli smartphone); inoltre al momento l'App di Scratch non è ancora supportata da *Linux*.

CHE COS'E' SCRATCH

Scratch è un ambiente visuale pensato per i ragazzi, perché possano imparare a programmare, ma divertendosi. Detto così forse non dice nulla! È vero, non si tratta di realizzare programmi per ufficio o per aziende, si intende! Né sarebbe possibile farlo con Scratch, è vero che la programmazione a blocchi, tipo i puzzle, utilizzati da Scratch, facilita un

po' il tutto, ma d'altra parte limita: i blocchi hanno comandi e parametri che non possono essere flessibili come in una programmazione testuale. Comincerete a muovere i primi passi nel complesso e affascinante mondo qual è quello della programmazione informatica.
Attraverso l'uso divertente, Scratch aiuta i ragazzi a utilizzare il pensiero computazionale; cioè a pensare a delle soluzioni di un problema così come farebbe un vero informatico, con strategie, algoritmi, la suddivisione di un problema complesso in sotto-problemi e cercando una soluzione che, forse, sarà utile quando si svolgeranno altri problemi dello stesso genere. Ci si dà un obiettivo e si studiano le strategie per arrivare alla soluzione o alle soluzioni. Non bisogna dimenticare mai che fantasia è il più importante ingrediente per qualunque tipo di programmazione!
Il ragionamento computazionale, la collaborazione e la creatività, sono capacità essenziali per chi fa parte della generazione attuale. Proprio il pensiero creativo, infatti, è alla base di Scratch. Non solo.
Si impara anche a collaborare e a ragionare in modo sistematico.
Si potrebbero condividere i progetti Scratch con tutti, in ogni parte del mondo: basti pensare che oggi Scratch è tradotto già in più di 40 lingue!
Con Scratch quindi si programma. Oggi, una delle parole chiave della Scuola è *coding*. Questa parolina magica "coding" significa, appunto, programmazione.

Scratch è uno degli strumenti più divertenti per iniziare a entrare nel mondo del *coding*, ma senza scrivere una sola riga di codice: con Scratch si programma utilizzando dei blocchi, incastrandoli tra di loro, proprio come facevamo da bambini con i giochi della Lego: in base alla forma che hanno sarà possibile incastrarli in mille modi e ottenere migliaia di combinazioni diverse.
Non lo sapevamo, ma stavamo già programmando!
Sono relativamente pochi i blocchi che si possono utilizzare; ma anche le note musicali sono davvero poche: soltanto sette. Ma quante melodie sono state scritte e si continua a scrivere (anche da parte di persone che non conoscono nulla di teoria musicale) con i diversi "incastri" di quelle 7 note!

Scratch è un progetto del Lifelong Kindergarten Group dei Media Lab del MIT, ossia dell'Università di ricerca privata di Cambridge, nel Massachusetts.
Inoltre Scratch è stato reso completamente libero!

Non occorre nessuna licenza per usare Scratch a casa, a scuola o in qualunque altro luogo.
Mentre scrivo questa guida, l'ultima versione di Scratch è la 3.29.1 (è quella che ho usato in questo libro per i diversi progetti che propongo), si tratta dello sviluppo della 3.0 lanciata a gennaio del 2019. Le precedenti versioni sono la 2.0 e la 1.4.

Scratch è così libero che si possono condividere sul web i propri progetti e qualunque membro della comunità di Scratch lo potrà modificare; anche tu potrai modificare i progetti degli altri: il nuovo progetto modificato è chiamato *remix*. Quando si fa il *remix* di un progetto, occorre indicare chiaramente il creatore originale del progetto e tutti coloro che, in qualche modo, hanno realizzato qualche *remix* importante per quel progetto.
È utile remixare i progetti; del resto si impara a programmare anche osservando i progetti realizzati da altri, osservando le soluzioni che hanno adottato altri scratcher per raggiugere un certo obiettivo.

I progetti condivisi in rete sono sotto licenza Creative Commons Share Alike: significa che chiunque è libero di modificare qualunque progetto condiviso nel sito di Scratch!
I progetti non condivisi, non possono essere remixati! Quindi, se diventerai un bravo programmatore e vorrai vendere i tuoi progetti, potrai farlo. In questo caso, però, non devi condividere i tuoi progetti sul sito di Scratch.

Scratch è Open Source, cioè il codice scritto dai creatori di Scratch è libero, è aperto e chiunque può vederlo, studiarlo e modificarlo, creando così un programma derivato da Scratch (come mBlock, giunto ormai alla versione 5).
Tale nuova versione derivata può tranquillamente essere distribuita, a condizione che si rispettino determinati vincoli di legge, uno è sicuramente il riconoscimento dell'opera originale e dei suoi autori.

L’Ambiente dell’App di Scratch

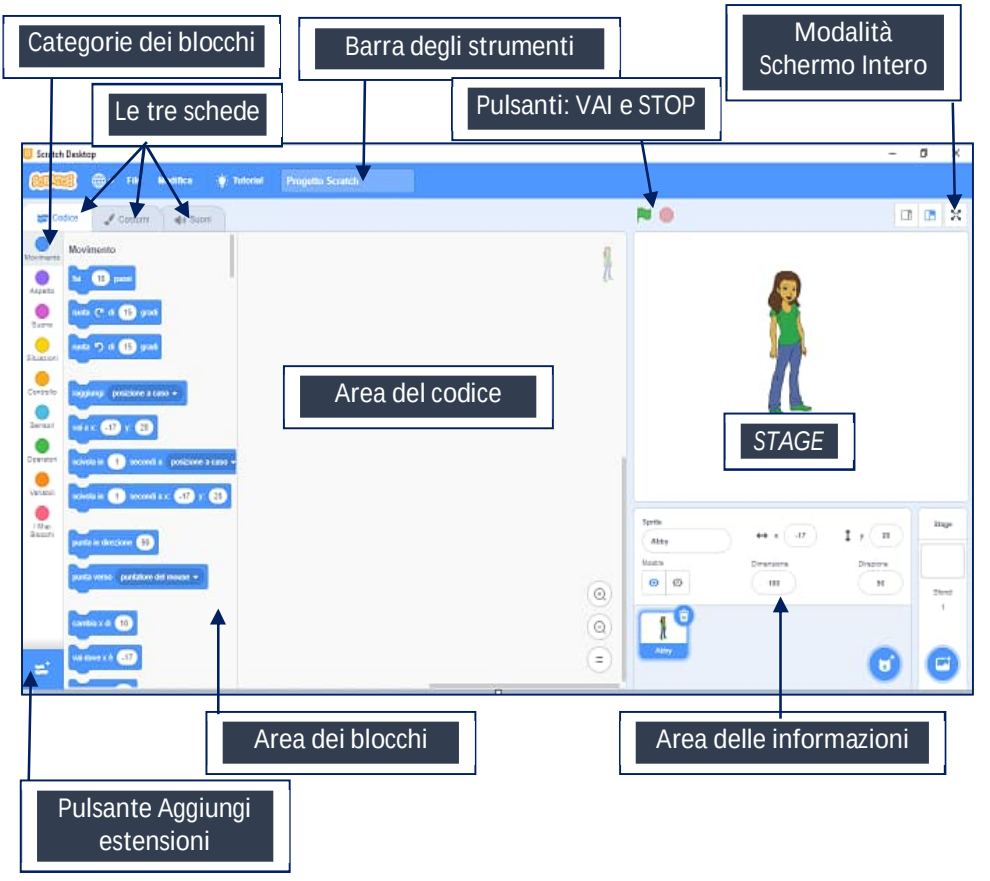


Fig. 1 Ambiente dell’App Scratch Desktop

Questa in alto è l’App Desktop di Scratch con i suoi componenti principali; nel testo ci riferiremo a questa nomenclatura durante la presentazione dei diversi progetti.

LAVORARE CON SCRATCH

Per lavorare con l'App di Scratch, e con Scratch in generale, non occorre scrivere righe di programmazione, ma tanta fantasia e pazienza!
I comandi sono impartiti utilizzando i diversi blocchi presenti nell'Area dei blocchi e suddivisi per Categoria (9 sono le categorie di default, ma se ne possono aggiungere altre utilizzando il pulsante Aggiungi estensioni).

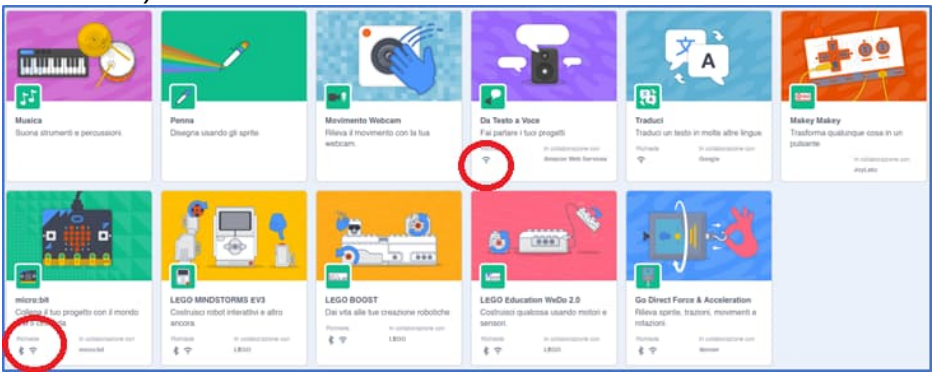


Fig. 2 La pagine delle Estensioni

Alcune estensioni funzionano solo se si è connessi a Internet, altre necessitano della connessione *Bluetooth*; quando comunque si apre la finestra delle estensioni, un piccolo logo indica chiaramente se è necessaria una connessione a Internet e/o il *Bluetooth*.

I Blocchi

I blocchi contengono i comandi, le istruzioni che agiscono sui personaggi (gli *sprite*) ma, nello stesso tempo, permettono agli stessi di agire. E' un po' come se gli *sprite* fossero degli attori che si muovono sul palco (lo *stage*) in base ai comandi scritti su un copione (il codice formato dalle istruzioni dei vari blocchi). Ovviamente anche il nostro palco (lo stage), come tutti i palchi può avere il suo sfondo.
blocchi hanno forme differenti: queste forme permettono di impilarli uno sull'altro o di incastrarli uno nell'altro.

Alcuni blocchi possono stare solo in cima; altri, pochissimi in realtà, devono stare per forza in basso, come blocchi di chiusura.

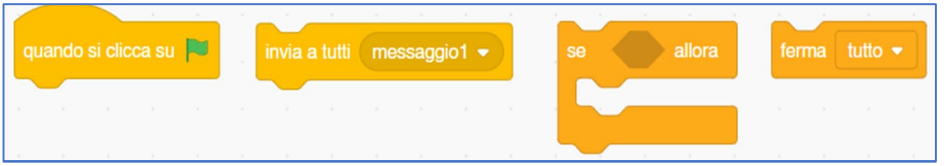


Fig. 3 Tipi di blocchi

In figura3 abbiamo degli esempi di blocchi con forme diverse: il primo dei quattro tipi di blocchi è quello solitamente utilizzato per far partire un progetto, è un *hat block*, cioè un blocco-cappello perché può stare, appunto, solo in cima agli altri blocchi. Il secondo è il classico blocco che si può impilare ovunque, può anche essere posta alla fine di un codice; il terzo è un blocco che richiede la presenza di altri blocchi al suo interno; il quarto è un blocco che può stare solo in coda, alla fine di un codice.
Ci sono dei blocchi che hanno una forma particolare: sono ovali. Questi sono blocchi valore: contengono delle informazioni che possono essere utilizzati come dei valori.

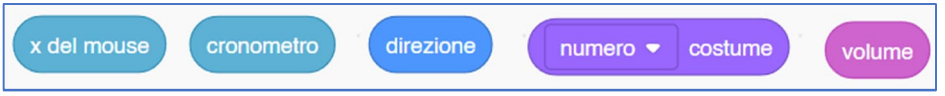


Fig. 4 I blocchi "valore"

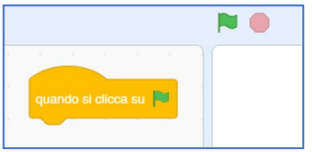
Data la forma che hanno non possono essere utilizzati da soli, ma inseriti in qualunque altro blocco che abbia uno spazio ovale (lo spazio ovale può avere dimensioni differenti, in ogni caso si adeguerà alla dimensione del blocco valore). Il primo blocco valore in figura contiene il valore della coordinata x della posizione del puntatore del mouse in quell'istante. Non dimentichiamo che Scratch utilizza una coppia di coordinate x,y per indicare la posizione di qualsiasi oggetto sullo stage.
La coordinata x indica la posizone orizzontale, mentre la coordinata y indica la posizione verticale.

Fig. 5 Blocchi esagonali

Infine, le categorie Sensori e Operatori hanno alcuni blocchi a forma esagonale e, ovviamente, potranno essere inseriti solo all'interno di quei blocchi che hanno uno spazio esagonale (che anche in questo caso si adatterà alle dimensioni dell'esagono). Questi blocchi per operare hanno bisogno dei blocchi valore che, vedremo, sono davvero tanti.

Tutta la programmazione con Scratch consiste nell'impilare i vari blocchi, utilizzando vari tipi di blocchi insieme ai valori. Solo la forma dei blocchi limita il loro utilizzo, ma le combinazioni dei blocchi, con le loro diverse forme, permettono di ottenere migliaia di soluzioni e di programmare progetti semplici ma anche molto complessi.
La programmazione testuale contiene spesso decine e decine di righe di codice, poiché ogni blocco di Scratch è una sola istruzione, sarà normale avere lunghe colonne di blocchi Scratch quando il progetto diventa complesso.

Normalmente ogni *sprite* ha un codice che esegue; anche lo sfondo può avere un suo codice. Per far partire un codice il blocco che più si utilizza è l'*hat block* con la bandierina verde.



Quando nell'App di Sceatch si cliccherà sulla bandierina verde il programma verrà avviato; se si volesse bloccare il programma, occorre cliccare sul pulsante rosso ottagonale, sulla destra della bandierina verde. Più *sprite* contemporaneamente possono avviare il proprio codice se hanno come blocco di avvio la bandierina verde.

Per far interagire gli *sprite* fra di loro, o per passare il "controllo" del programma da uno *sprite* all'altro, normalmente si utilizza il blocco di Situazione Invia a tutti messaggio1 che è semppre abbianato al blocco quando ricevo messaggio1. Immaginiamo di avere tre *sprite* nel nostro progetto.

Lo *sprite1* esegue il suo codice, poi deve far intervenire lo *sprite3*. Allora lo *sprite1* inserisce il blocco invia a tutti messaggio1; nel codice dello *sprite*3 inseriremo il blocco quando ricevo messaggio1, in questo modo il controllo passerà allo *sprite3*. In realtà il blocco quando ricevo

messaggio1 possiamo inserirlo a tutti quelli *sprite* che vogliamo che interagiscano, non è obbligatorio metterlo nel codice di un solo *sprite*.

Immaginate un'altra situazione del genere: si invia a tutti il messaggio nasconditi, a 5 *sprite* inseriamo il blocco quando ricevo nasconditi e gli aggiungiamo blocco di Aspetto nascondi: i 5 *sprite* saranno nascosti dallo *stage* tutti contemporaneamente!

N.B. Cliccando sulla freccetta a triangolo, possiamo modificare il testo messaggio1, scrivendone uno a nostra piacere.

Blocchi particolari

Il blocco Se… allora… altrimenti (*If… then… else*) è un blocco di Controllo e permette di gestire delle alternative al verificarsi di certe condizioni.
Se una certa condizione è vera verrà rilasciato *true*, allora si eseguiranno i blocchi inseriti all'interno del primo spazio; altrimenti, se la condizione ha rilascito *false* verranno eseguiti i blocchi presenti nel secondo spazio.
Se… allora può essere utilizzato anche senza l'altrimenti.
La condizione viene verificata normalmente utilizzando i blocchi Operatore.

Nella pagina precedente ci sono due esempi del blocco di Controllo Se… allora… altrimenti.
A sinistra il blocco controlla se una certa variabile x è maggiore di 50; se il controllo rilascia *true*, cioè se è vero che x è maggiore di 50 allora apparirà un fumetto, per 2 secondi, con la scritta "*E' vero*". Se il controllo

rilascia *false* (non è vero che x è maggiore di 50) apparirà un fumetto, sempre per 2 secondi, con la frase "*E' falso, x non è maggiore di 50*".

Invece, nell'esempio a destra viene sempre controllata la variabile x, se è maggiore di 50, allora lo *sprite* a cui è abbinato quel blocco andrà nella posizione al centro dello *stage* (x:0 y:0), altrimenti lo *sprite* raggiungerà una posizione a caso.

Le variabili

Chi fa matematica è abituato a usare delle incognite come *x, y, z.*
Anche Scratch, come tutti i linguaggi di programmazione, utilizza delle incognite o, meglio, delle variabili, ossia degli scatoloni in cui possiamo conservare qualcosa da riutilizzare più volte: lo scatolone lo chiameremo per esempio "x" ma al suo interno ci potrà essere di tutto, come un valore numerico, una stringa, una frase, ecc...

La Categoria Variabili ha molti blocchi per gestire le variabili: per crearle, assegnarle un valore, cancellarle, ecc… Vedremo che esistono variabili particolari, chiamate liste (*array*) che potremme considerare delle super-variabili in grado di contenere molti elementi al loro interno, non uno solo come succede per le "normali" variaibili.

Il blocco Chiedi… e attendi

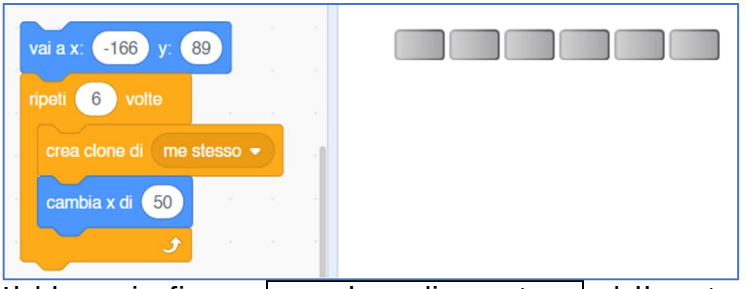
La figura accanto mostra il blocco Chiedi… e attendi in azione. Questo è un blocco di *input*, dà un valore di ritorno; permette di inserire qualcosa attraverso la tastiera. Il blocco lavora in tandem con il blocco risposta: infatti ciò che sarà inserito con la tastiera sarà salvato nella variabile di sistema di Scratch risposta. Poiché ogni

volta che si inserisce qualcosa con il blocco Chiedi, il blocco risposta assume l'ultimo valore inserito, in genere conviene scaricare il valore del blocco risposta in una nostra variabile, nell'esempio in figura è stata creata la variabile *x* che conterrà il valore di risposta: ciò che è stato inserito attraverso la tastiera sarà salvato automaticamente in risposta, ma subito riversato in *x*.

Il blocco ripeti e ripeti fino a quando

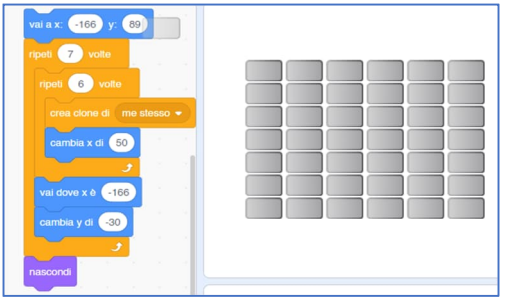
I blocchi ripeti e ripeti fino a quando permetteno di ripetere dei comandi un numero definito di volte (nel caso del blocco ripeti) o fino a quando si avvera una certa condizione (nel caso del blocco ripeti fino a quando). Nell'esempio della figura accanto, viene chiesto di "Scrivere un numero", finché il numero inserito non sarà maggiore di 50, ci verrà posta sempre la stessa domanda; solo se il numero inserito sarà maggiore di 50 verrà inserito nella variabile risposta e, poi, nella variabile x.

Il blocco crea clone di me stesso. Le righe e le colonne



Il blocco in figura, crea clone di me stesso, della categoria Controllo, permette di replicare una o più copie dello *sprite* a cui è associato il blocco. In questo caso allo *sprite* grigio Button ho assegnato il codice che si vide in figura: inizialmente il blocco si posiziona secondo le coordinate x:-166, y: 69 (in alto a sinistra), poi per 6 volte ripete due comandi: crea un clone di se stesso e si sposta a destra di 50 passi: questo, come si vede in figura, lo farà per 6 volte consecutivamente, ottenendo il risultato che si vede in figura, ossia la creazione di una riga dello stesso *sprite*.

E' importante saper utilizzare questo blocco di Controllo perché nel libro sarà usata molto spesso questa tecnica.
Ora, se ci occorressero altre 6 righe uguali, per ottenere una griglia di 6x7, potrei copiare tutto quel codice per altre 6 volte, ottenendo così una colonna di ben 21 blocchi più altri 6 per il comando dello spostamento in verticale dopo ogni riga.



L'altra soluzione è ripetere solo quelle tre righe di codice per 7 volte di seguito, spostandoci in basso dopo ogni ciclo delle 6 ripetizioni, otterrei questo che vedete in figura: delle righe e delle colonne dello stesso *sprite* ottenute solo da un solo *sprite* clonato più volte, dopo aver calibrato le coordinate x,y in base alle dimensioni dello *sprite*.

Il blocco finale Nascondi, serve a nascondere il blocco originale, altrimenti si posizionerebbe in basso a sinistra alla fine di tutti gli altri blocchi.
Vai dove x è -166 serve a indicare la posizione orizzontale da cui deve ripartire ogni ripetizione e quindi ogni nuova riga.

Ricordo che *x* indica la posizione orizzontale, mentre *y* la posizione verticale; *x* può assumere valori che vanno da *x: -220* (estrema sinistra) *x:0* (centro) fno a *x:220* (estrema destra); la coordinata *y* assume valori da *y:180* (in alto) *y:0* (al centro), fino a *y:-180* (in basso).
Dobbiamo pensare allo stage come a un rettangolo suddiviso in 4 parti:

x:-220 y:180 *x:0* alto sinistra *x:-220 y:0* *x:0*	*y:180* *x:220 y:180* alto destra *x:200 y:0*
basso sinistra *x:220 y:-180* *x:0*	*y:0* basso destra *y:-180* *x:220 y:-180*

In base a queste coordinate possiamo renderci esattamente conto della posizione di un certo elemento rispetto ad un altro o rispetto al centro dello stage.

Indice

L'AMBIENTE SCRATCH 2
CHE COS'E' SCRATCH 2
L'Ambiente dell'App di Scratch 5
LAVORARE CON SCRATCH 6
I Blocchi 6
Blocchi particolari 9
Le variabili 10
Il blocco Chiedi… e attendi 10
Il blocco ripeti e ripeti fino a quando 11
Il blocco crea clone di me stesso. Le righe e le colonne 11
I PROGETTI - PRIMO CICLO PRIMARIA .....Errore. Il segnalibro non è definito.
Progetto: IL QUADRATO DI DIECI .....Errore. Il segnalibro non è definito.
Progetto: CONTA LE PALLINE COLORATE...Errore. Il segnalibro non è definito.
Progetto: SOPRA/SOTTO – DESTRA/SINISTRAErrore. Il segnalibro non è definito.
Progetto: LA SCALA DEI NUMRI DA 1 A 10.Errore. Il segnalibro non è definito.
Progetto: SOMMA SULLA LINEA.....Errore. Il segnalibro non è definito.
Progetto: L'ABACO.....Errore. Il segnalibro non è definito.
Progetto: LA SCOMPOSIZIONE .....Errore. Il segnalibro non è definito.
Progetto: DUELLO: CONFRONTO FRA NUMERIErrore. Il segnalibro non è definito.
Progetto: MAGGIORE – MINORE – UGUALEErrore. Il segnalibro non è definito.
Progetto: COLPISCI SOLO I NUMERI DISPARI!Errore. Il segnalibro non è definito.
Progetto: CENTINAIA, DECINE e UNITA' ....Errore. Il segnalibro non è definito.
Progetto: LA MACCHINA DEL MINIMO .....Errore. Il segnalibro non è definito.
PROGETTI – SECONDO CICLO PRIMARIA ...Errore. Il segnalibro non è definito.
Progetto: LE FRAZIONI /1.....Errore. Il segnalibro non è definito.
Progetto: LE FRAZIONI /2.....Errore. Il segnalibro non è definito.
Progetto: LE FRAZIONI /3.....Errore. Il segnalibro non è definito.
Progetto: SOMMA e SOTTRAZIONE.....Errore. Il segnalibro non è definito.
Progetto: LE TABELLINE /1 .....Errore. Il segnalibro non è definito.
Progetto: LE TABELLINE /2 .....Errore. Il segnalibro non è definito.

Progetto: LE POTENZE..... Errore. Il segnalibro non è definito.
Progetto: ORE, MINUTI e SECONDI..... Errore. Il segnalibro non è definito.
Progetto: LE EQUIVALENZE ..... Errore. Il segnalibro non è definito.
Progetto: PESO NETTO, LORDO e TARA .... Errore. Il segnalibro non è definito.
GEOMETRIA..... Errore. Il segnalibro non è definito.
Progetto: GLI ANGOLI ..... Errore. Il segnalibro non è definito.
Progetto: LA ROTAZIONE ..... Errore. Il segnalibro non è definito.
Progetto: SIMMETRIA ORIZZONTALE..... Errore. Il segnalibro non è definito.
Progetto: SIMMETRIA VERTICALE..... Errore. Il segnalibro non è definito.
Progetto: LA TRASLAZIONE ..... Errore. Il segnalibro non è definito.
Progetto: IL RAGGIO E IL CERCHIO ..... Errore. Il segnalibro non è definito.
Progetto: LA MOLTIPLICAZIONE..... Errore. Il segnalibro non è definito.
Progetto: IL QUADRATO DELLA SOMMA .. Errore. Il segnalibro non è definito.
Progetto: IL CALCOLO DELLA PROBABILITA'Errore. Il segnalibro non è definito.
..... Errore. Il segnalibro non è definito.
Progetto: BIT, BYTE E CALCOLO BINARIO.. Errore. Il segnalibro non è definito.
Progetto: LA GESTIONE DI UN QUIZ ..... Errore. Il segnalibro non è definito.
Indice 14
Altre guide dell'Autore riguardanti ScratchErrore. Il segnalibro non è definito.