

# GEOGEBRA E CODING

di Giuseppe Alessandri

**PAROLE CHIAVE:**

DGS, CODING, TIC, GEOMETRIA DINAMICA, DRAGGING

Quando è permesso il coding può essere gestito in ambienti ibridi, cioè non specifici per la costruzione di programmi, qualora si debbano sviluppare attività per le quali l'ambiente non ha adeguati strumenti per gestirle, quando si vogliono ottenere soluzioni migliori o perché unico strumento per articolare determinate esperienze. Nell'articolo verrà presentato GeoGebra, un software per la didattica della geometria, e si vedrà come lavorare con il coding in esso.

## 1. GeoGebra

GeoGebra<sup>1</sup> fa parte dei *Dynamic Geometry Software (DGS)*<sup>2</sup>, software specifici per la Geometria ma non solo. Inseriti nella didattica assumono una significativa funzione di supporto rendendo evidente il rapporto tra i grafici e gli oggetti teorici da essi rappresentati. La produzione e la manipolazione di forme geometriche attraverso i classici strumenti da disegno come riga, squadra e compasso, sono sostituite da analoghi virtuali che introducono anche la dimensione del movimento, attraverso la funzione di dragging (trascinamento), rendendo dinamica la geometria. Selezionando un elemento (vertice, lato, angolo) e trascinandolo con il mouse, viene coinvolta l'intera figura che, tuttavia, mantiene le sue caratteristiche geometriche<sup>3</sup>. Così lo studente può comprendere e dimostrare le proprietà delle figure elaborate e verificare le regolarità, proprio per mezzo della permanenza delle caratteristiche invariante, nonostante le operazioni di ingrandimento/riduzione, deformazione, rotazione, ...<sup>4</sup>.

I DGS possono essere utilizzati anche da utenti inesperti ma che diventano via via sempre più auto-

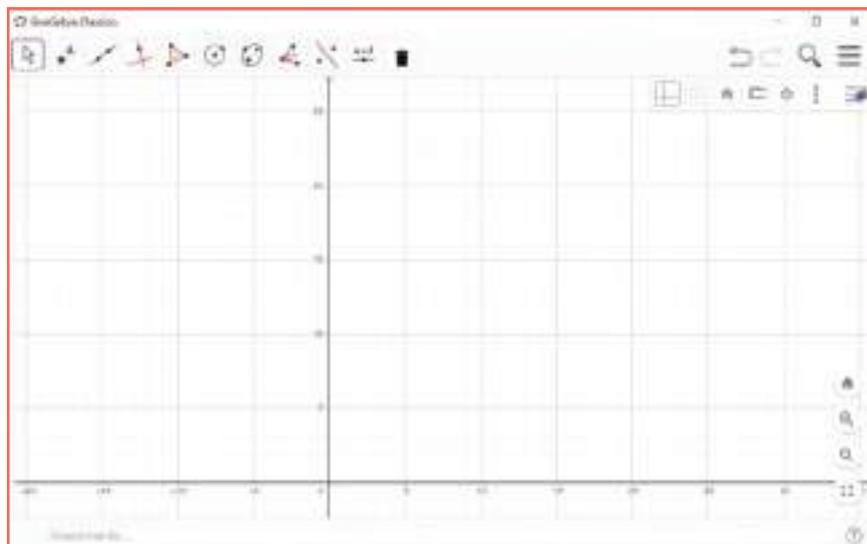


Fig. 1 – Interfaccia del software GeoGebra

nomi e in grado di sperimentare personali intuizioni. Gli studenti sono incentivati a costruire le loro conoscenze geometriche in un ambiente che mette a disposizione dispositivi che esprimono il linguaggio della geometria e che possono essere attivati attraverso meccanismi coerenti con le regole di quel dominio. Non si limitano a riprodurre graficamente figure geometriche, a sperimentare senza avere conoscenze, ma procedono attraverso la geometria per costruire la propria geometria<sup>5</sup>.

Nell'interfaccia del software GeoGebra (Fig. 1) convivono una *toolbar* (in alto), che rende disponibili gli elementi della geometria insieme ad altre utilità, due "viste", cioè zone, quella algebrica e quella grafica. Nella vista grafica è possibile costruire figure geometriche, se-

lezionandole dalla *toolbar*, e sulla vista algebrica vengono riportate le coordinate dei punti, la lunghezza dei segmenti, le equazioni delle figure, l'ampiezza degli angoli, delle aree: le formule algebriche che qui "prendono vita" aiutano lo studente a comprendere il parallelismo e il rapporto di causazione tra un grafico e la sua formulazione. È altresì presente un campo per l'inserimento dei comandi (in basso), in forma testuale da tastiera, che generano risultati sulle due viste.

L'ambiente ha un ruolo di supporto e di facilitazione perché, agevolando o sostituendo le azioni che potrebbero essere svolte con carta e matita, consente un più efficiente svolgimento del compito; inoltre attraverso il *dragging* promuove l'esplorazione e l'analisi<sup>6</sup>.

<sup>1</sup> Dal sito [www.geogebra.org](http://www.geogebra.org) è possibile scaricare il software e prendere visione di diversi tutorial; ultima visualizzazione: 10/10/2020.

<sup>2</sup> All'url <http://www.didafor.com/geogebra/index.htm> è disponibile un ebook con esempi relativi a quanto presentato nell'articolo; ultima visualizzazione: 10/10/2020.

<sup>3</sup> Ad esempio, una volta costruite le tre bisettrici di un triangolo, si può verificare che si incontrano in un punto (incentro); trascinando, uno alla volta, i tre vertici l'incentro resta sempre interno al triangolo e gli angoli ai vertici sono sempre divisi in due angoli uguali fra loro.

<sup>4</sup> Cfr. Laborde e al. 2006, pp.275-303.

<sup>5</sup> Cfr. Costabile 2013.

<sup>6</sup> Spunti del paragrafo tratti da C. Mezzabotta in Alessandri 2015, pp. 239–242.

## 2. Il coding in GeoGebra

In GeoGebra è possibile disegnare grafici anche costruendo insiemi di comandi, chiamati script. GeoGebra supporta due linguaggi di scripting: *GGBScript* e *JavaScript*. Il primo permette uno sviluppo sequenziale di comandi, il secondo introduce la possibilità di controllare il flusso dello svolgimento utilizzando le classiche strutture di programmazione: azioni elementari, di confronto, di ripetizione. In entrambi i casi si parla di codice *embedded*.

## 3. GGBScript

Se si volesse disegnare un grafico che visualizzi una retta intersecante la funzione  $\sin(x)$  in due punti, si potrebbe operare secondo una delle tre seguenti modalità.

La prima: inserire il comando  $f(x) = \sin(x)$  nel campo "Inserimento" (nella vista grafica viene visualizzata la funzione trigonometrica "seno"); selezionare, nella *toolbar*, l'icona "Punto" e cliccare su un punto della funzione  $\sin(x)$ ; ripetere l'operazione precedente per il secondo punto; selezionare, nella *toolbar*, il pulsante "Retta", puntando su un punto e raggiungendo l'altro, viene disegnata la retta fra i due punti.

In alternativa si può operare solamente nel campo "Inserimento", inserendo i seguenti comandi, ciascuno seguito da invio:

- $f(x) = \sin(x)$  (viene visualizzata la funzione trigonometrica "seno"),
- $a = 2$  (alla variabile "a" viene assegnato il valore 2),
- $A = (a, f(a))$  (è creato e visualizzato il punto "A" di coordinate  $(2, \sin(2))$ ),
- $b = 4$  (alla variabile "b" viene assegnato il valore 4),
- $B = (b, f(b))$  (è creato e visualizzato il punto "B" di coordinate  $(4, \sin(4))$ ),
- Retta(A, B) (viene tracciata la retta fra i due punti A e B).

Infine se si volesse la visualizzazione in base a un'azione dell'utente,

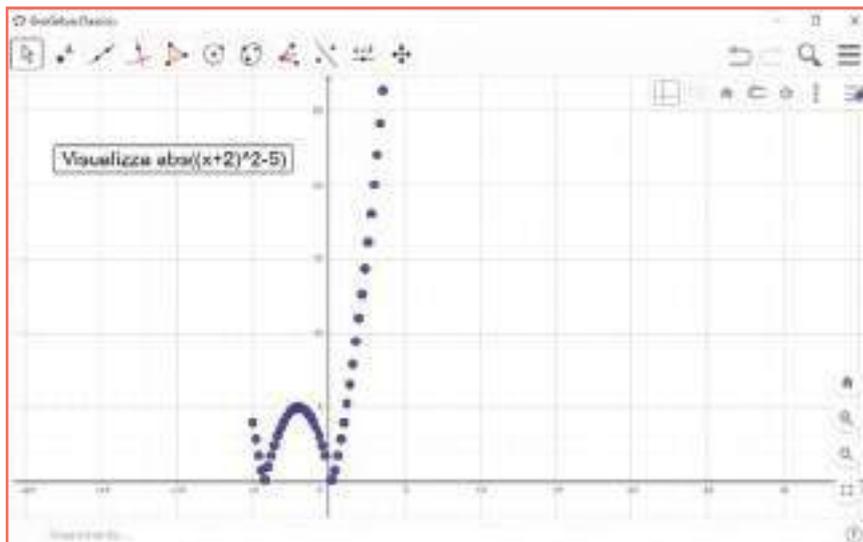


Fig. 2 – Andamento di  $\text{abs}((x+2)^2-5)$  attraverso JavaScript

ad esempio con un click su una determinata immagine o pulsante, si dovrebbe inserire la sequenza dei comandi in uno script associandolo, in un apposito ambiente di scripting, a quell'oggetto e prevedere di mandarlo in esecuzione quando si clicca sullo stesso oggetto<sup>7</sup>.

## 4. JavaScript

Diversamente dagli script *GGBScript*, in questo caso, è possibile articolare procedimenti più complessi che impieghino costrutti di programmazione.

Supponiamo di voler visualizzare l'andamento della funzione  $f(x) = \text{abs}((x+2)^2-5)$ <sup>8</sup> attraverso uno script che ne tracci "punto punto" il grafico nella relativa vista. Si può associare a un pulsante il seguente script di tracciamento<sup>9</sup>:

```
for(var i = -5; i<5; i = i + 0.2)
{
  ggbApplet.evalCommand("("+"i+",
  abs("("+"i+"+"2)^2-5)");
}
```

Il risultato è visibile in Fig. 2.

Si può notare la complicatezza della scrittura del codice<sup>10</sup> – tuttavia, a un adeguato livello di scolarità non è tale – dovuta all'*embedding* di JavaScript nell'ambiente di Ge-

oGebra. Lo script contiene la conosciuta istruzione di ripetizione *for* che fa ripetere l'esecuzione del metodo *evalCommand*, dell'oggetto *ggbApplet*, con il quale si esegue un comando come se questo fosse digitato direttamente nella barra di inserimento di GeoGebra. In questo caso, a ogni ciclo dell'iterazione, viene eseguito il comando per la visualizzazione di un punto della funzione nella vista grafica. La verifica della correttezza può essere ottenuta digitando, nel campo per l'inserimento dei comandi, la forma algebrica della funzione e cioè:  $f(x) = \text{abs}((x+2)^2-5)$  e verificare se la curva ottenuta si sovrappone alla precedente.

## 5. Conclusioni

Dal punto di vista didattico può rivelarsi esperienza utile poter lavorare in ambienti ibridi nei quali occorre tener in debito conto le caratteristiche di ciascuno di quelli che li generano.

<sup>7</sup> Spunti del paragrafo tratti da C. Mezzabotta in Alessandri 2015, pp. 239–242.

<sup>8</sup> abs: valore assoluto.

<sup>9</sup> Cfr. nota 2.

<sup>10</sup> Riconducibile alla sola sintassi del codice, altrimenti lo *script* è di facile interpretazione.

Ad esempio, nell'istruzione `ggbApplet.evalCommand("(“+i+”, abs(“(“+i+”+2)^2-5))”)` esiste un copioso utilizzo di parentesi; alcune dipendono proprio dalla sintassi della formula che esprime la funzione matematica, ma altre dalla modalità di scrittura dell'istruzione in questo ambiente. Sistemandole in modo corretto dal punto di vista della sintassi dell'istruzione, ma errato concettualmente da un punto di vista matematico, si può ottenere comunque un risultato nella vista grafica, però quell'andamento non è quello giusto. Entrano in gioco, quindi, conoscenze matematiche e di codifica che si intersecano fra loro e competenza nell'usarle in modo opportuno. In linea generale, oltre a questi aspetti, la realizzazione di esperienze in questi ambienti e, in particolare, nella costruzione degli script impone che lo studente rifletta continuamente sul rapporto fra il modello concreto che sta costruendo e quello su cui si basa la soluzione del problema che, tuttavia, egli non possiede; ecco allora «che la costruzione del programma si configura come la costruzione di una conoscenza»<sup>11</sup>.

La costruzione di script in ambienti ibridi, qui brevemente analizzata, ha una duplice valenza che, tuttavia, non deve portare a una semplificazione educativa: seppur apre a un uso del *coding* in ambienti software che fanno direttamente riferimento alle discipline, permettendo lo svilup-

**“ gli studenti sanno leggere e scrivere testi in diverse lingue, ma raramente sanno leggere i linguaggi di programmazione ”**

po di compiti maggiormente orientati alla realtà, tuttavia occorre ricordare che lo studente non si limita a costruire delle conoscenze e acquisire competenze in ordine alla disciplina di riferimento (in questo caso la matematica), ma vive anche in quella che Gee<sup>12</sup> definisce «situazione di apprendimento senza contenuti». L'autore si riferisce all'insieme di atteggiamenti, strategie, modalità che una persona mette in campo quando videogioca, nel nostro caso quando costruisce programmi.

Concludendo, gli studenti sanno leggere e scrivere testi in diverse lin-

gue, ma raramente sanno leggere i linguaggi di programmazione o sanno scriverci o capirli. Sono esperti nella gestione delle interfacce dei tablet, dei computer ma difficilmente sanno operare a un livello approfondito con essi. Per anni la scuola non ha arginato tale andamento con variegate analisi e da parte dei docenti si è sviluppata una affannosa ricerca di tecnologie che risultassero il più possibile trasparenti rispetto alla didattica. Tuttavia attraverso ambienti ibridi (come GeoGebra) che sfruttano coerenze con ambiti disciplinari è possibile condurre gli studenti verso proficue modalità di apprendimento avvicinando il *coding*, inteso non solamente come costruzione programmi, ma come pensiero computazionale che lo sostiene.

<sup>11</sup> Cfr. Costabile 2013, p. 65.

<sup>12</sup> Cfr. Gee 2014.

### Bibliografia

- Alessandri G., *Didattica per il coding. Didattica, pensiero computazionale, corporeità*, Aracne Editrice, Roma 2015.
- Costabile F.A., *Aspetti pedagogici e cognitivi dell'apprendimento della matematica con le tecnologie. La programmazione come metodologia. Il software MatCos*, Carocci Editore, Roma 2013.
- Gee J.P., *What Video Games Have to Teach Us about Learning and Literacy*, St. Martin's Press, LLC, 2007; trad. it., Rivoltella P.C., Carenzio A., *Come un videogioco. Insegnare e apprendere nella scuola digitale*, Raffaello Cortina Editore, Milano 2013.
- Laborde C., Kynigos C., Hollebrands K., Strässer R., *Teaching and learning geometry with technology*, pp.275-303 in: Angel Gutiérrez, Paolo Boero (eds), *Handbook of Research on the Psychology of Mathematics Education. Past, Present and Future*, Sense Publishers, Rotterdam 2006.

### GIUSEPPE ALESSANDRI



È docente universitario in quiescenza. Ha insegnato Didattica generale, Tecnologie didattiche e Fondamenti di Informatica presso il Dipartimento di Scienze della Formazione, dei Beni Culturali e del Turismo dell'Università degli Studi di Macerata. Si è occupato di Tecnologie e di Informatica nel loro rapporto con la Didattica. Ha pubblicato diversi testi e articoli in questi settori.