

Campo Minato in JavaScript

Confronto tra sviluppo manuale e sviluppo
con supporto di Intelligenza Artificiale

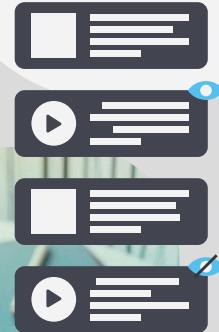


OBIETTIVO DEL PROGETTO

Realizzare il gioco Campo Minato in JavaScript senza usare librerie esterne.

Confrontare due metodi di sviluppo:

- Sviluppo manuale
- Sviluppo con supporto di Intelligenza Artificiale



ORGANIZZAZIONE DEL LAVORO

Ogni gioco è stato sviluppato da una coppia di studenti



Studente 1

Sviluppo completo del gioco scrivendo tutto il codice a mano, senza utilizzo dell'intelligenza artificiale nemmeno per cercare funzioni o selettori

Studente 2

Sviluppo dello stesso gioco con supporto di un modello di Intelligenza Artificiale per la scrittura di funzioni



METODO 1: SVILUPPO MANUALE

Il gioco è stato progettato e programmato interamente a mano

Tutta la logica è stata scritta senza aiuti automatici:

- Creazione della griglia
- Gestione delle mine
- Calcolo dei numeri
- Gestione del click e del tasto destro
- Controllo vittoria e sconfitta



METODO 2: SVILUPPO CON SUPPORTO DELL'AI

Lo studente utilizza l'IA come:

- Aiuto per scrivere funzioni
- Supporto per correggere errori
- Spiegazione di parti di codice difficili

Il codice non viene copiato alla cieca, ma capito e adattato





Analisi comparativa dei due progetti

Progetto AI

Struttura dati

- Utilizza un array monodimensionale
- Ogni cella è un oggetto con proprietà:
 - Bomba
 - Scoperta
 - Bandiera
- Approccio più compatto

Algoritmi

- mine: posizionamento casuale
- Calcolo del vicinato: funzione dedicata che gestisce automaticamente i bordi
- Rivelazione ricorsiva: le celle vuote rivelano automaticamente le adiacenti
- Condizione di vittoria: confronto tra celle scoperte e celle totali non-mina

UI / UX

- Layout realizzato con CSS Grid
- Overlay finale per vittoria o sconfitta
- 3 livelli di difficoltà
- Uso di emoji Unicode (🚩) per una grafica semplice e immediata



Analisi comparativa dei due progetti

Progetto senza AI

Struttura dati

- Uso di array bidimensionali separati, uno per ogni aspetto:
- grid[][] → numeri delle celle
- minesGrid[][] → posizione delle mine
- revealed[][] → celle scoperte
- flagged[][] → bandiere
- Approccio procedurale e classico

Algoritmi

- Prima mossa sicura: le mine vengono piazzate solo dopo il primo click
- Generazione mine con esclusione della cella iniziale
- Rivelazione con controlli sui limiti
- Vittoria quando tutte le celle non-mina sono rivelate

UI / UX

- Struttura basata su tabella HTML
- Celle ridimensionabili (20px / 30px)
- 4 livelli di difficoltà, inclusa modalità personalizzata
- Risorse grafiche PNG (bombe, bandiere, sfondo)
- Overlay post-game avanzato
- Titolo nascosto automaticamente per griglie molto grandi

Differenze principali nei metodi

	AI	Non AI
Struttura dati	Array 1D con oggetti	Più array 2D separati
Prima mossa	Può essere una bomba	Sempre sicura
Rendering	CSS Grid	Tabella HTML
Risorse grafiche	Emoji Unicode	Immagini PNG
Configurazione	3 livelli fissi	4 livelli + personalizzato
Post-game	Overlay semplice	Modalità “rivedi partita”
Codice	Più compatto (~180 righe)	Più esteso (~340 righe)



Differenze principali nei metodi

	Vantaggi	Svantaggi
Sviluppo manuale	<ul style="list-style-type: none">○ Comprensione profonda del funzionamento del programma○ Migliora il ragionamento logico○ Aiuta a organizzare meglio il codice○ Rafforza l'apprendimento della programmazione○ Codice originale scritto dallo studente	<ul style="list-style-type: none">○ Richiede molto tempo rispetto a quello fatto con l'aiuto dell'AI○ Facile bloccarsi su errori complessi○ Necessita di una buona preparazione iniziale○ Debugging lungo e complicato
Uso dell'Intelligenza Artificiale	<ul style="list-style-type: none">○ Riduce i tempi di sviluppo○ Aiuta a trovare e correggere errori○ Migliora e semplifica il codice○ Supporta la scrittura di parti complesse○ Utile come assistente, tutor e supporto allo studio	<ul style="list-style-type: none">○ Rischio di non comprendere realmente il codice○ Possibile uso di soluzioni non spiegabili○ Difficoltà nel verificare la correttezza senza basi solide○ Possibile generazione di codice non ottimizzato



Conclusioni

Progetto AI

- Più moderno ed elegante
- Codice compatto e leggibile
- Ideale per dimostrare uso di CSS Grid e strutture dati semplificate

Progetto non AI

- Più ricco di funzionalità
- Maggiore controllo sul flusso di gioco
- Offre personalizzazione e strumenti di analisi post-partita

Entrambi implementano correttamente le meccaniche fondamentali del Campo Minato, ma:

- AI privilegia semplicità e pulizia del codice
- Not_AI punta su completezza e flessibilità per l'utente

