



By Luca Bertiato

# Campo Minato





# Campo Minato, tra Logica e Intuizione

Ho deciso di ricreare lo storico Videogioco Campo Minato

È un classico rompicapo in cui devi evitare le mine nascoste nel campo. Scopri le celle sicure e piazza le bandierine per segnalare le posizioni delle mine. Sii strategico e cerca di completare il campo senza detonare le mine!

---



# Come giocare?

Campo Minato è un gioco di logica in cui l'obiettivo è scoprire e liberare tutte le celle senza mine di un campo rettangolare. Ogni cella può nascondere una mina o indicare il numero di mine nelle celle adiacenti. I giocatori usano queste indicazioni per evitare le mine, liberare spazio sicuro, e, quando necessario, posizionare bandierine per segnalare le mine sospette. La partita si conclude con successo quando tutte le celle sicure sono state scoperte senza attivare alcuna mina.

---

# Campo di gioco

---

Il campo di gioco si presenta come nella seguente immagine.

Puoi selezionare tra tre livelli di difficoltà:

- livello facile, 10 mine;
- livello medio, 15 mine;
- livello difficile, 20 mine.



# Campo di gioco

## Campo Minato

difficoltà:

mine presenti: 13					reset	partita in corso...				
					1					
					2			1	1	
					2		1	3		
		1	1	1	1		1			
		1			1	1	2	2	2	
		2			2		3	1	1	
		1			2					
		2	1		1	1	4			
	2		1				2			
	2	1	1				1			

**Congratulazioni, hai vinto!**

Bravo! Hai superato la sfida con successo.

[Gioca di nuovo](#)

## Campo Minato

difficoltà:

mine presenti: 6					reset	partita in corso...				
	1		2		1					
1	2	1	2	1	1		1	1	1	
	1	1	1	1		1	2		1	
1	2	2		1	1	2		2	1	
			2	1	1		2	1		
			2	1	2	1	1			
					1	1	1	1		
				3	2	1		1		
					2	1	1	1		
					2					

**Peccato, hai perso!**

Non scoraggiarti, riprova e potresti vincere la prossima volta.

[Gioca di nuovo](#)

# Tecnologie utilizzate

Questo gioco di Campo Minato è stato realizzato utilizzando:

- HTML per la struttura;
- CSS per lo stile;
- JavaScript per la logica di gioco;
- jQuery per semplificare la manipolazione del DOM e gestire gli eventi.



```

1 //genera coordinate casuali per le mine
2 generaPosizioneMine() {
3   let c, r;
4   //per quante sono le mine da mettere
5   for (let i = 0; i < this.numeroMine; i++) {
6     //genera coordinate
7     c = this.generaRandom(this.colonne);
8     r = this.generaRandom(this.righe);
9     //se la cella con coordinate r e c è già una cella si deve rifare
10    //finchè non ne trova una libera
11    while (this.matriceCelle[r][c].cellaIsMina == true) {
12      c = this.generaRandom(this.colonne);
13      r = this.generaRandom(this.righe);
14    }
15    //segnalo la presenza di una mina
16    this.matriceCelle[r][c].cellaIsMina = true;
17    //vicino alla mina segno le altre celle adiacenti comprese quelle in obliquo
18    //partendo alto a sinistra
19    for (let contr = -1; contr <= 1; contr++) {
20      for (let contC = -1; contC <= 1; contC++) {
21        let posR = r + contr;
22        let posC = c + contC;
23        //se non si è fuori dal campo
24        if (posR >= 0 && posR < this.righe && posC >= 0 && posC < this.colonne){
25          if(this.matriceCelle[posR][posC].cellaIsMina != null)
26            this.matriceCelle[posR][posC].numMineVicine++;
27        }
28      }
29    }
30  }
31 }

```

```

1 class myCampo {
2   constructor(numRighe, numColonne, numMine) {
3     this.colonne = numColonne;
4     this.righe = numRighe;
5     this.numeroMine = numMine;
6     this.matriceCelle = [];
7     this.mineTrovate = 0;
8     this.isGiocoInCorso = true;
9     this.bandierine = 0;
10    this.aperte = 0;
11    this.totCelle = this.righe * this.colonne;
12  }
13 }

```

```

1 class myCella{
2   constructor(){
3     this.numMineVicine = 0;
4     this.cellaIsMina = false;
5     this.stato = "chiusa";
6   }
7 }

```

# Un po' di codice...

Puoi trovare tutto il codice completo qui:

<https://github.com/lucabertiato/lucabertiato.github.io.git>

Grazie per aver provato  
questo piccolo gioco!