Progetto di Programmazione - A.A. 2025/2026

Dipartimento di Informatica "Giovanni Degli Antoni"
Università degli Studi di Milano
Studente: Marco Brembi
Matricola: 41901A
Nome del Progetto: Shoot Your Shot

17 Giugno 2025

---

1 Informazioni Generali

Il progetto Shoot Your Shot nasce con l'obiettivo di combinare elementi di fisica e logica in un contesto ludico e interattivo. Il gioco sfida l'utente a colpire un bersaglio virtuale attraverso il calcolo di angoli e traiettorie, simulando il lancio di un proiettile. Ogni livello rappresenta una crescente difficoltà, con un numero limitato di tentativi e un raggio d'azione progressivamente ridotto. L'interfaccia si basa su input numerici forniti dall'utente, elaborati per determinare il punto di impatto. Il progetto è pensato per testare capacità di calcolo, ragionamento spaziale e precisione, offrendo un'esperienza coinvolgente e formativa.

Gli obiettivi del progetto sono:
newline

- Simulare un gioco a livelli tramite terminale.
- Applicare concetti fisici (moto parabolico) per calcolare la traiettoria di un colpo.
- Gestire l'interazione con l'utente attraverso input da tastiera.
- Scrivere i risultati su file di output.
- Sperimentare l'uso di:
  - Strutture (`struct`)
  - Condizioni e cicli
  - File I/O
  - Librerie esterne

2 Struttura del progetto e file utilizzati

2.1 File principali

File	Descrizione
`index.html`	Pagina iniziale del progetto (Home)
`main.c`	Contiene il codice principale del gioco
`lib.h`	Contiene dichiarazioni, costanti e la struct `Punto`
`output.txt`	File di log del gioco
*File `.txt`	File grafici del gioco: bomba, razzo, vittoria, istruzioni

Table 1: Elenco completo dei file utilizzati nel progetto

3 Informazioni Generali

3.1 Flusso generale

1. Introduzione iniziale (con possibilità di leggere le istruzioni).
2. Inizio del primo livello.
3. Generazione casuale del bersaglio (`punto()`).
4. Inserimento da parte dell'utente di:
   (a) Angolo di alzo (0°–70°)
   (b) Angolo orizzontale (- 45° – + 45°)
5. Calcolo delle coordinate di impatto (`spara()`).
6. Verifica colpo (`colpito()`):
   (a) Se colpito → livello successivo (`lvlsuccessivo()`).
   (b) Se sbagliato → decremento colpi.
7. Se colpi esauriti → `termina()` + scelta se ricominciare.
8. Se raggiunto livello 6 → `fine()` (vittoria).

4 Strutture e Funzioni Principali

4.1 Struct

```
typedef struct {
    float x;
    float y;
} Punto;
```

4.2 Funzioni principali

Funzione	**Scopo**
`main()`	Controlla il ciclo generale del gioco
`spara()`	Richiede input e calcola le coordinate del colpo
`punto()`	Genera il bersaglio con coordinate casuali
`colpito()`	Calcola distanza e verifica impatto
`lvlsuccessivo()`	Incrementa livello e rigenera bersaglio
`termina()`	Stampa messaggio di Game Over e chiude partita
`fine()`	Messaggio di vittoria
`intro()`	Introduzione e chiamata istruzioni
`istruzioni()`	Legge il file `istruzioni.txt`
`lettura()`	Stampa a terminale i file testuali
`bomb()`, `razzo()`, `vittoria()`	Visualizza grafica testuale dei rispettivi eventi

Table 2: Elenco delle principali funzioni del programma

5 Aspetti Tecnici

- Input: Tastiera (convalidato).
- Output: Schermo + file `output.txt`.
- Costanti fisiche: g = gravità, v = velocità iniziale.
- Angoli: convertiti da gradi a radianti.
- Uso della funzione `sleep()` per temporizzazioni.
- Pulizia terminale: con `system("cls")` (Windows).

6 Esempio di formula usata

Gittata del proiettile:

$$\text{Gittata} = \frac{v^2}{g} \cdot \sin(2 \cdot \theta)$$

Coordinate finali:

$$\begin{cases} x = \text{gittata} \cdot \cos(\phi) \\ y = \text{gittata} \cdot \sin(\phi) \end{cases}$$

7 Considerazioni finali

In questo progetto ho utilizzato il linguaggio **C in versione base**, senza ricorrere a tecniche avanzate o librerie complesse, con l'obiettivo di mantenere il codice **semplice e accessibile**. Ho cercato tuttavia di **curare ogni dettaglio del gioco**, dalla gestione accurata dell'input dell'utente alla simulazione realistica del moto parabolico, passando per la scrittura dei risultati su file di log.

Questo approccio mi ha permesso di **consolidare le basi della programmazione strutturata**, sperimentando con strutture dati, condizioni, cicli e operazioni di input/output su file. Inoltre, ho posto particolare attenzione all'esperienza dell'utente, assicurando **messaggi chiari** e un **flusso di gioco fluido e coinvolgente**.

Il progetto rappresenta quindi un **equilibrio tra semplicità tecnica e cura nel design**, offrendo una solida base da cui poter eventualmente sviluppare versioni più complesse e articolate in futuro.