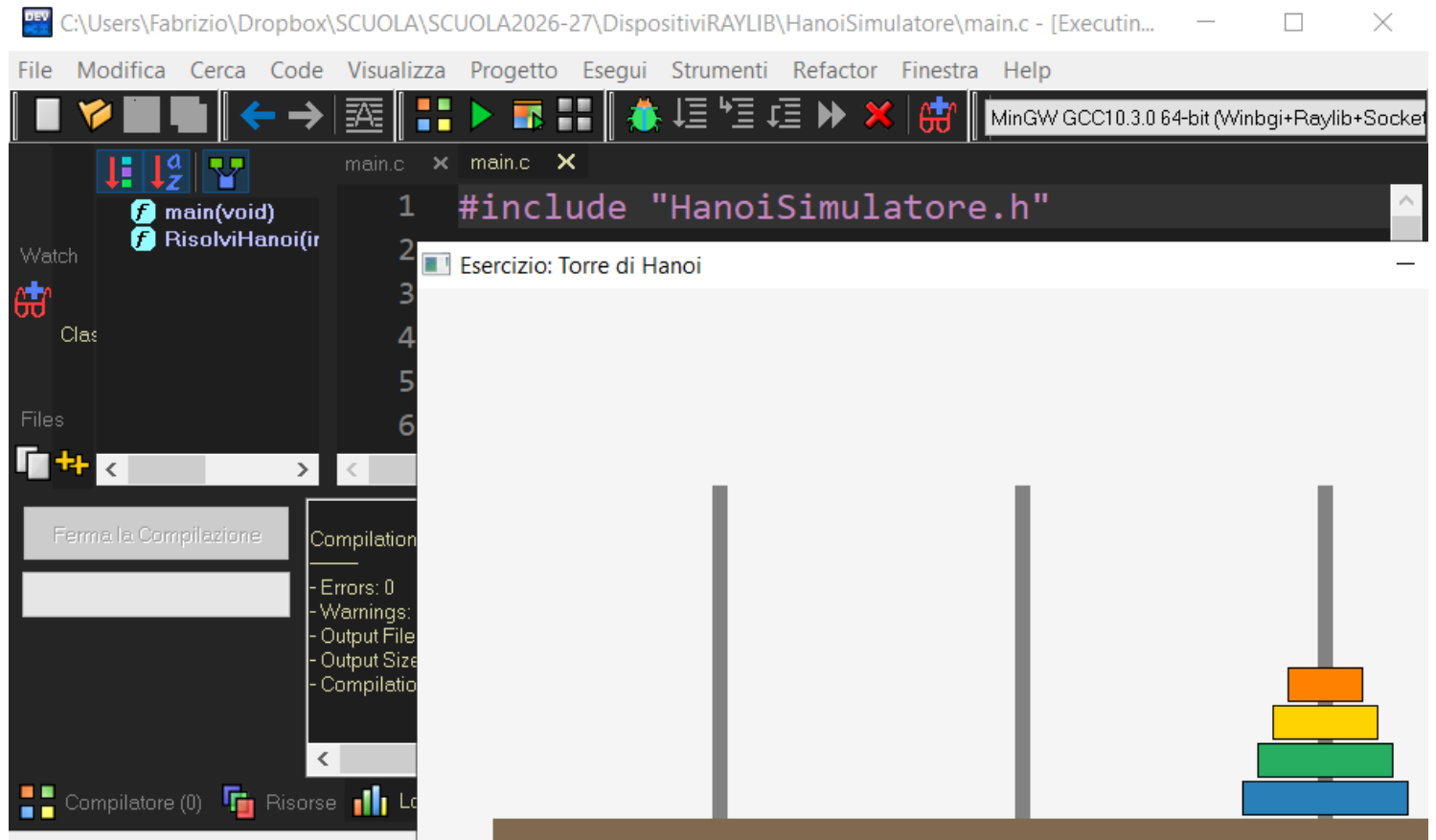


hanoi_sim.h



`hanoi_sim.h` è una libreria **header-only** che disegna e anima la **Torre di Hanoi** usando Raylib. Tu scrivi solo le mosse (`MuoviDisco(...)`), mentre la libreria si occupa di:

- disegnare base, pioli e dischi,
- controllare che le mosse rispettino le regole,
- animare il disco che si sposta,
- mostrare eventuali messaggi di errore a schermo.

Va inclusa in **un solo file** `.c` (di solito `main.c`).

1. Compilazione

```
gcc main.c -o hanoi -lraylib -lm
```

Se la tua installazione di Raylib richiede più librerie:

```
gcc main.c -o hanoi -lraylib -lGL -lm -lpthread -ldl -lrt -lX11
```

2. Programma minimo

```
#include "hanoi_sim.h"

int main(void)
{
    InizializzaSimulazione(3); /* 3 dischi sul primo piolo */

    MuoviDisco(1, 3, 50);      /* da piolo 1 a piolo 3, velocità media */
    MuoviDisco(1, 2, 50);
    MuoviDisco(3, 2, 50);
    MuoviDisco(1, 3, 50);
    MuoviDisco(2, 1, 50);
    MuoviDisco(2, 3, 50);
    MuoviDisco(1, 3, 50);

    ChiudiSimulazione(); /* mostra la finestra finché non la chiudi */
    return 0;
}
```

Le mosse vengono eseguite **in sequenza**: ogni chiamata a `MuoviDisco` anima un disco e blocca il programma finché l'animazione non è finita.

3. API pubblica

Funzione

```
void InizializzaSimulazione(int
numDischi)
void MuoviDisco(int da, int a, int
velocita)

void ChiudiSimulazione(void)
```

Descrizione

Crea la finestra e mette `numDischi` dischi sul piolo **1**.
Valori validi: **1-8** (fuori range vengono clampati).
Sposta il disco in cima al piolo da (**1, 2 o 3**) sul piolo a (**1, 2 o 3**). `velocita` va da **1** (lentissimo) a **100** (istantaneo).
Mantiene aperta la finestra finché l'utente non la chiude, poi libera tutto.

Parametro `velocita`

- 1 → animazione da **1,2 s**
 - 50 → circa **0,6 s**
 - 98, 99, 100 → **istantaneo** (nessun frame intermedio)
 - valori fuori range vengono clampati tra 1 e 100.
-

4. Cosa vedi nella finestra

- Finestra 800×450, sfondo bianco, 60 FPS.
 - Tre pioli grigi su una base marrone.
 - I dischi sono colorati in base alla loro dimensione (8 colori ciclici).
 - Il disco in movimento fa un percorso a **3 fasi**: sale, si sposta orizzontalmente, scende.
 - Se una mossa è **illegale**, la simulazione si ferma e compare un messaggio rosso in alto a sinistra.
-

5. Regole controllate automaticamente

MuoviDisco verifica le regole classiche della Torre di Hanoi e in caso di violazione **ferma la simulazione** mostrando un messaggio:

- indici dei pioli fuori range (< 1 o > 3),
- tentativo di muovere un disco da un piolo vuoto,
- tentativo di mettere un disco **più grande** sopra uno **più piccolo**.

In tutti questi casi la finestra resta aperta (in attesa che l'utente la chiuda) ma non accetta più mosse.

6. Esempio: risolvere Hanoi ricorsivamente

```
#include "hanoi_sim.h"
```

```
void hanoi(int n, int da, int a, int via, int velocita)
{
    // da codificare
}
```

```
int main(void)
{
    InizializzaSimulazione(4);
    hanoi(4, 1, 3, 2, 80); /* 4 dischi, piolo 1 → 3, mediamente veloce */
    ChiudiSimulazione();
    return 0;
}
```

Per 4 dischi servono $2^4 - 1 = 15$ mosse. Metti `vel = 100` se vuoi vederle tutte in un lampo, `vel = 20` se vuoi gustarti l'animazione.

7. Note e limiti

- Massimo **8 dischi** (`MAX_DISCHI`).
 - La libreria è **header-only** ma non `static inline`: includila in un unico `.c` per evitare simboli duplicati al linking.
 - `InizializzaSimulazione` imposta `SetTraceLogLevel(LOG_NONE)` per silenziare i messaggi di Raylib in console.
 - `ChiudiSimulazione` va chiamata **dopo** l'ultima mossa, altrimenti la finestra si chiude subito e non vedi niente.
 - Non è pensato per input interattivo dell'utente: le mosse sono decise dal tuo codice.
-

8. Schema mentale

main.c

- | — InizializzaSimulazione(n) → apre la finestra, sistema i dischi
- | — MuoviDisco(da, a, vel) → controlla regole + anima + aggiorna stato
| (ripetuto quante volte vuoi)
- | — ChiudiSimulazione() → aspetta chiusura finestra, poi esce

Buon divertimento con la Torre di Hanoi!