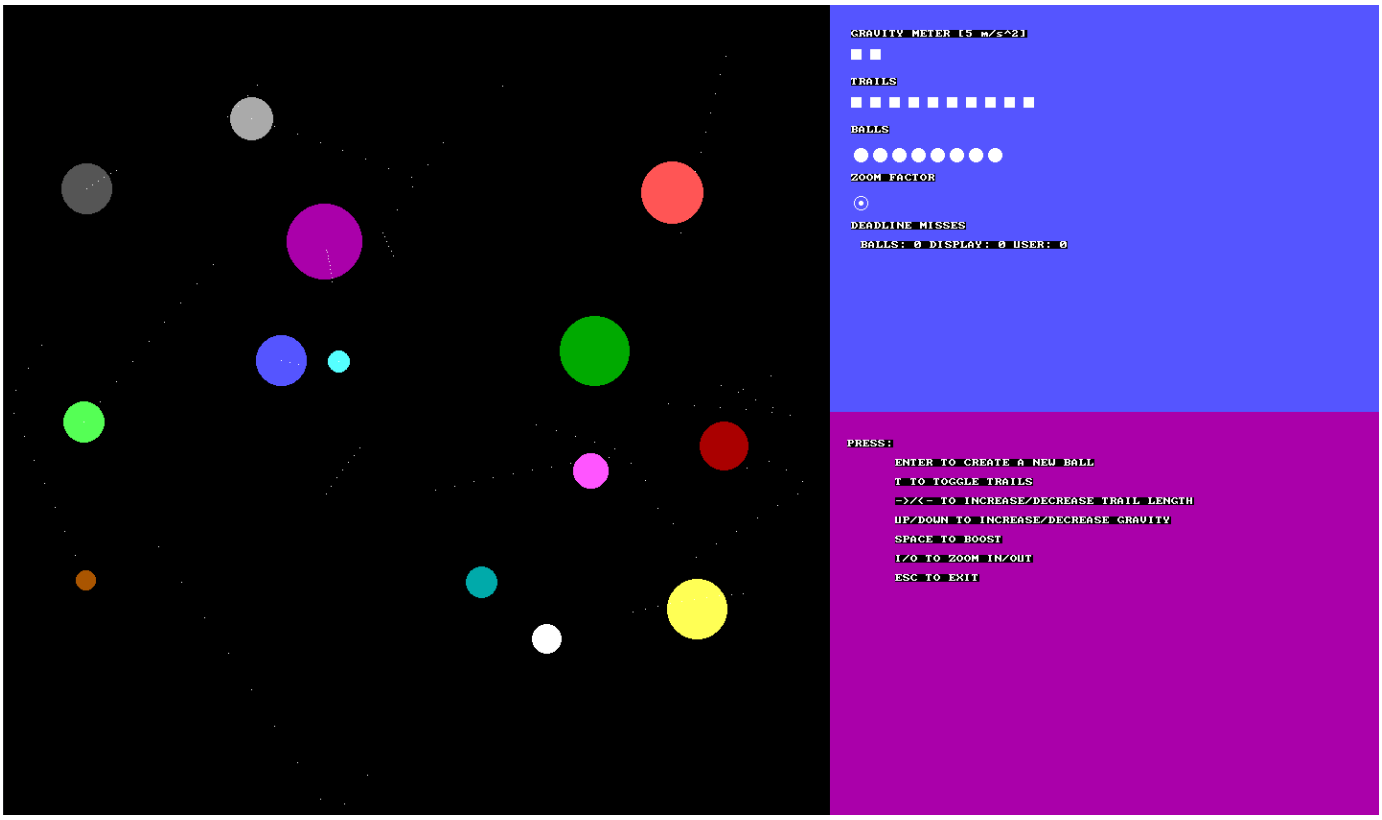
Real Time Systems

Jumping Balls with Collisions

Luca Lombardo
6055884
l.lombardo10@studenti.unipi.it

Gennaio 2022



Contents

1 Introduzione

Il progetto consiste nella realizzazione di un programma che simuli le interazioni fisiche di una quantità variabile di sfere, vincolate all'interno di una scatola, capaci di collidere tra di loro. Le palline sono dotate di massa e hanno un aspetto variabile e l'utente vi può interagire in vari modi tramite la tastiera. Il programma è stato scritto in linguaggio C su sistema Linux, utilizzando le librerie pthread per la gestione dei task e Allegro per la parte grafica.

2 Modello fisico

Le palline sono state modellate come corpi rigidi omogenei bidimensionali che producono urti elastici scontrandosi tra loro

$$v_{b1}^{x+} = \frac{(m_1 - m_2)v_{b1}^{x} + 2m_2 v_{b2}^{x}}{m_1 + m_2}$$

$$v_{b1}^{y+} = \frac{(m_1 - m_2)v_{b1}^{y} + 2m_2 v_{b2}^{y}}{m_1 + m_2}$$

e anelastici con l'ambiente, portando a una perdita di energia cinetica legata a una certa costante di smorzamento d.

$$v_{b1}^{x+} = -dv_{b1}^{x+}$$

$$v_{b1}^{y+} = -dv_{b1}^{y+}$$

Nella legge del moto compare anche il contributo dell'accelerazione di gravità:

$$x_{b1}(t) = v_{b1}^{x} t$$

$$y_{b1}(t) = (v_{b1}^{y} - gt)t$$

3 Struttura del programma

Il programma è composto da un $main()$ che lancia una funzione di inizializzazione necessaria a per la rappresentazione degli elementi statici dello schermo, come il tutorial, e per l'attivazione delle funzioni necessarie di Allegro, e più task che compiono ognuno una sola funzione.

3.1 Ball Task

Ogni Ball Task gestisce la fisica di una pallina, aggiornando la sua velocità in base alle collisioni effettuate nell'ultimo periodo e conseguentemente la sua posizione. Inoltre aggiorna il buffer circolare dedicato alla memorizzazione delle

sue posizioni precedenti per mostrare eventualmente la traccia.
Le collisioni sono rilevate dalle funzioni *overlapping*() e *contact*() che misurano la distanze dei centri di tutte le palline confrontandole con i rispettivi raggi: se troppo vicine, è avvenuto un contatto tra due palline e vengono aggiornate le loro velocità e resettate le posizioni proporzionalmente alla compenetrazione.

3.2 Display Task

Il Display Task cancella ogni pallina per ridisegnarla nella posizione aggiornata insieme alla traccia (se l'opzione è attiva), tenendo anche conto del grado di zoom, ritagliando la preview della scatola prima di proiettarla sullo schermo. Inoltre aggiorna le informazioni a schermo.

3.3 User Task

Lo User Task traduce tramite uno *switch* i comandi dell'utente modificando le variabili globali e aggiorna la parte corrispondente dell'interfaccia.

3.4 Strutture Dati e Variabili Condivise

- Status: contiene tutte le informazioni di una pallina, intrinseche (colore, raggio, massa) e sullo stato attuale (posizione e velocità).
- Cbuf: è un buffer circolare che memorizza le ultime posizioni di una pallina per essere utilizzata nella funzione di tracciamento della scia.
- g: accelerazione di gravità [0-135].
- tflag: stato di attivazione delle scie.
- tlen: lunghezza delle scie [1-27].
- zoomLvl: livello di zoom [1-3].
- nab: numero di palline attive [1-14].
- end: etichetta di chiusura del programma.

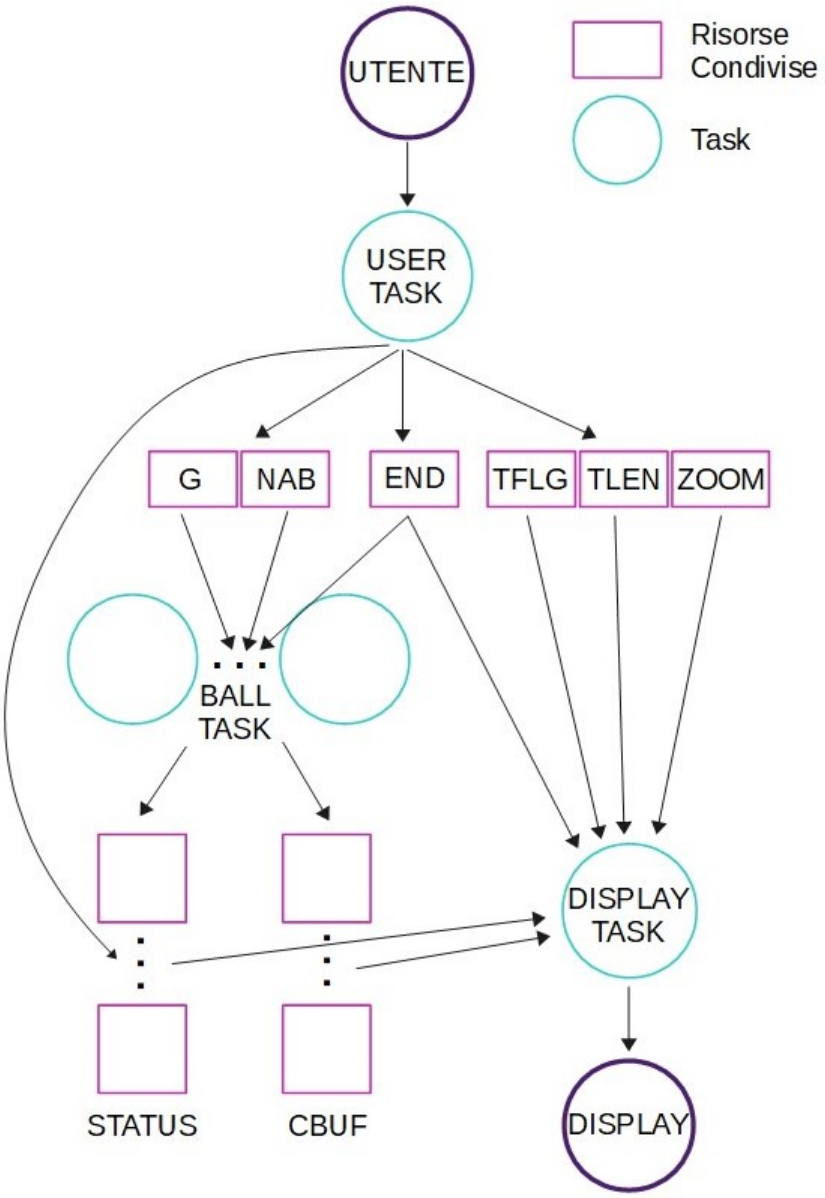


Figure 1: Diagramma di accesso alle risorse.

4 Interfaccia

L'utente può conoscere lo stato attuale del programma e i comandi a sua disposizione dalla sezione destra della schermata.

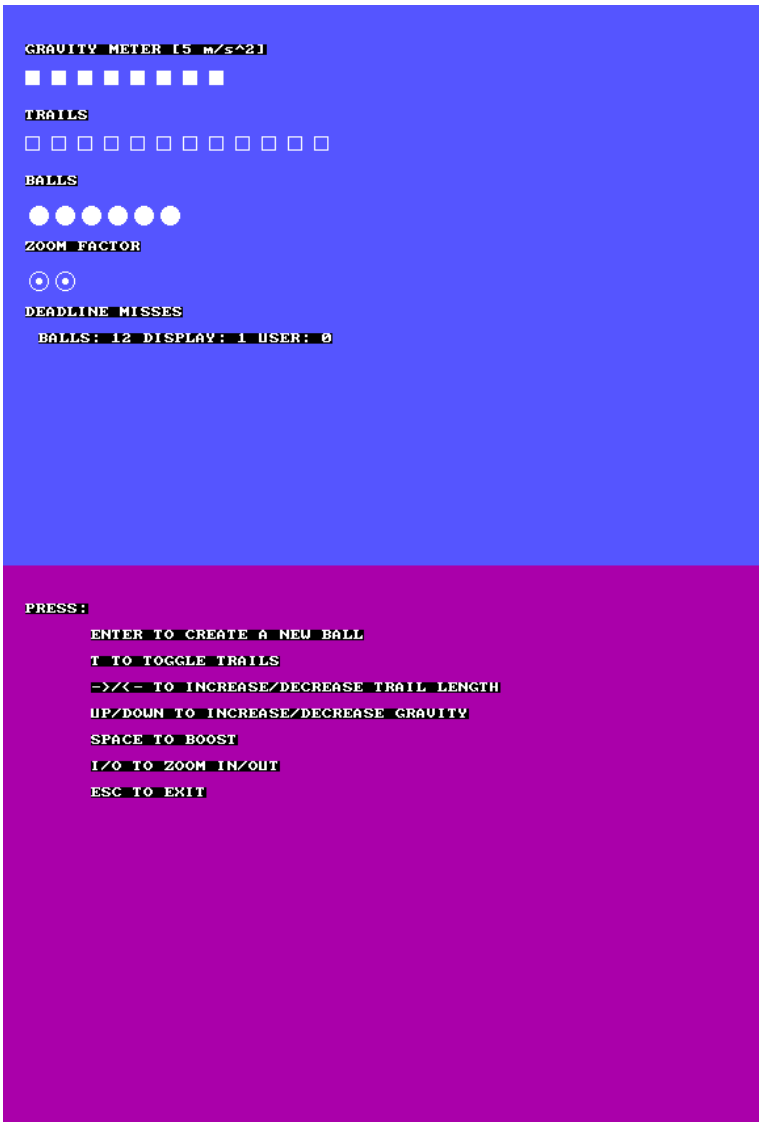


Figure 2: Interfaccia grafica.

In particolare sono mostrati:

- valore attuale della costante gravitazionale

- lunghezza delle scie lasciate dalle palline (quadratini pieni indicano che le scie sono attive)
- il numero attuale di palline nella scatola
- il valore di zoom (1, 2 o 3)
- il numero di deadline miss per ogni task

I comandi a disposizione sono:

- INVIO per generare una nuova pallina, con raggio, massa, colore e velocità iniziali casuali
- T per attivare o disattivare le scie
- FRECCIA DX/SX per aumentare o diminuire la lunghezza delle scie
- FRECCIA SU/GIÙ per aumentare o diminuire la gravità
- SPAZIO per resettare a un valore casuale la velocità di tutte le palline (iniettare energia)
- I/O per aumentare o diminuire il valore dello zoom
- ESC per uscire dal programma

5 Parametri e performance

Le priorità dei task sono state scelte in base all'importanza della loro esecuzione. Al primo posto si trovano i ball task, fra di loro schedulati secondo una politica *RoundRobin*, al fine di mantenere la simulazione fisica quanto più realistica possibile; di seguito il display task e infine lo user task, avendo notato che l'interazione con l'utente rimane ottima anche senza una priorità alta. I periodi sono frutto di trial and error, avendo raggiunto un compromesso tra qualità estetica della simulazione e soddisfazione dei vincoli delle deadline. L'aspetto della rappresentazione è infatti correlato al refresh rate dell'immagine, coincidente con l'attivazione del display task. Inoltre si è notato che è possibile diminuire l'utilizzazione del processore attribuendo un periodo più lungo allo user task essendo impercettibile per l'utente.

	Ball Task	Display Task	User Task
Periodo	15	20	40
Priorità	3	2	1

Figure 3: Tabella riassuntiva dei parametri dei task.

6 Conclusioni

La scelta dei parametri risulta adeguata poiché anche nelle combinazioni più critiche (massimo numero di palline, attivazione di scie e zoom e ripetuta iniezione di energia) il numero di deadline miss rimane zero, a meno di fluttuazioni dovute a rallentamenti esterni del processore.

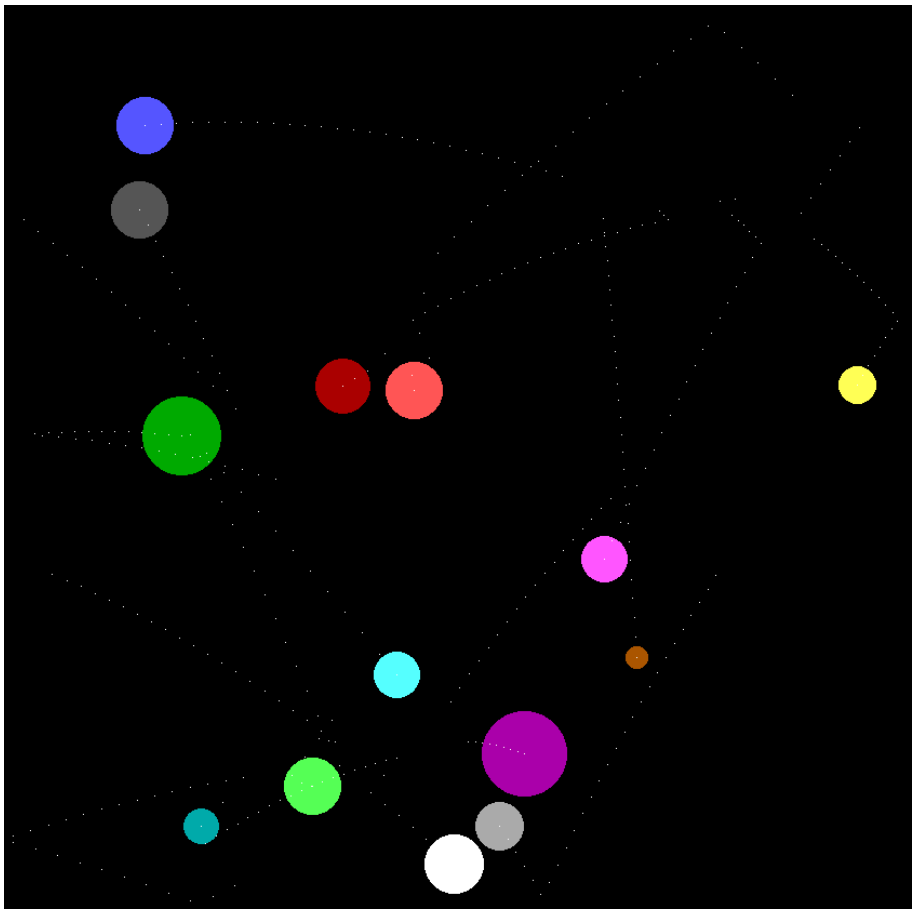


Figure 4: Schermata di gioco.