

Relazione progetto: simulazione del comportamento di stormi

Pini Carlotta
Savasta Arcangelo
Trotta Luigi

Agosto 2024

1 Scelte progettuali e implementative

Si è scelto per l'elaborazione del progetto l'utilizzo della libreria grafica (SFML). SFML è una libreria C++ che facilita la gestione di grafica 2d, audio e input. Nell'implementazione del progetto si è scelto Visual Studio Code come editor.

Il codice è organizzato in quattro translation units, incluso il main, con una chiara separazione tra le leggi di volo (`flight_laws`), la statistica (`statistics`) e la logica di generazione e movimento degli uccelli (`boids`).

Gli uccelli sono stati rappresentati come una coppia di vettori: il primo indica la sua posizione nel piano rispetto all'origine (fissata nell'angolo in alto a sinistra della finestra) e il secondo la sua velocità (per la quale si intende la variazione di posizione per frame). Ciascun uccello orienta la propria direzione in base alle componenti cartesiane del vettore velocità e possiede una texture, caricata da un file png.

Inizialmente, gli uccelli sono posizionati casualmente sullo schermo con una velocità random (le componenti cartesiane per ciascuna posizione e velocità sono estratte da una distribuzione uniforme); tale velocità è limitata a un valore massimo ed influenzata dalle leggi di volo per lo stormo: legge della separazione, legge dell'allineamento e legge della coesione.

Ciascuna legge produce un vettore che si somma alla velocità del boid ed è calcolato attraverso le posizioni e le velocità dei boids considerati vicini. Per la legge della separazione, sono considerati vicini i boids che si trovano a una distanza al massimo d_s rispetto al boid preso in esame; mentre per le altre due, si considera come distanza massima d . In ambedue i casi si è scelto di raffigurare due cerchi, centrati in ciascun uccello e di raggio d_s e d , con lo scopo di far notare all'utente quando entrano in gioco le velocità dovute alle leggi elencate precedentemente.

Tutti gli uccelli sono raggruppati in una struttura chiamata "Flock" (stormo), che contiene i parametri relativi alle tre leggi del volo.

Il comportamento degli uccelli ai bordi della finestra è stato gestito nella seguente maniera: il boid che tenta di lasciare lo schermo viene teletrasportato al lato opposto (conservando la propria velocità).

2 Istruzioni per compilazione

La compilazione del programma si esegue tramite CMake. Sono riportate le righe di codice da scrivere su terminale per compilare (in Debug o in Release mode), eseguire e testare correttamente il programma:

- Compilazione in Debug mode: `cmake -S . -B build -DCMAKE_BUILD_TYPE=Debug.`
- Compilazione in Release mode: `cmake -S . -B build -DCMAKE_BUILD_TYPE=Release.`

- Esecuzione del programma: `./build/boids`.
- Verifica dei test: `cmake --build build --target test`.

Per utenti che utilizzano un computer con Windows Subsystem for Linux con Ubuntu, dopo l'esecuzione potrebbero comparire due indirect leaks da una libreria esterna (libLLVM).

3 Descrizione input e output

Il programma richiede input esterno interattivo durante l'esecuzione, dove l'utente è chiamato a inserire dei parametri caratterizzanti la simulazione. Nonostante gli ampi intervalli di cui l'utente si può servire, ci permettiamo di consigliare alcuni valori che permettono di apprezzare al meglio gli stormi che si vengono a creare.

Generalmente, si consigliano: $20 \leq n \leq 50$, $80 \leq d \leq 120$, $20 \leq d_s \leq 50$.

Per utenti che utilizzano un computer con Windows Subsystem for Linux con Ubuntu, consigliamo: $maxSpeed = 5$, $s = 0.01$, $a = 0.005$, $c = 0.00001$.

Per utenti che utilizzano un computer con sistema operativo MacOS, consigliamo: $0.4 \leq maxSpeed \leq 0.6$, $s = 0.1$, $a = 0.05$, $c = 0.000001$.

In output, l'utente potrà visualizzare in una finestra di simulazione il volo degli uccelli su un piano, in altre due finestre le posizioni medie occupate dagli uccelli in ascisse e ordinate (tramite istogrammi a frequenza, rispettivamente in blu e in verde) e infine in una finestra la distanza media tra i boids e la loro velocità media, ciascuna con la rispettiva deviazione standard.

I dati relativi alle posizioni, distanze e velocità degli uccelli sono calcolate e mandate a schermo con una frequenza di valore varia in base al sistema operativo adoperato: per una corretta visualizzazione, si consiglia di impostare nel file `main.cpp` in riga 209, previa compilazione, `const size_t FRAME_FREQUENCY` con un valore di 100 frame per utenti con Windows Subsystem for Linux con Ubuntu e un valore di 1000 frame per utenti con sistema operativo MacOS.

4 Interpretazione dei risultati

I grafici e le statistiche forniscono una panoramica del comportamento degli uccelli. Le posizioni medie e le velocità aiutano a comprendere come i boids si aggregano e si muovono nel sistema; in particolare, si nota con l'avanzare dei frame, una diminuzione della distanza media tra i boids e della velocità media e un accentrimento delle posizioni dei boids rispetto alla finestra. La variabilità del comportamento è ulteriormente confermata dalle deviazioni standard sulle velocità e distanza medie.

5 Strategia di test

Per garantire un programma esaustivo ed esente da errori, si è scelto di verificare le funzioni usate con dei test di funzionalità, che controllano che vengano eseguiti correttamente tutti i componenti grafici e non e che le finestre si aggiornino come previsto. A tal fine, si è scelto di utilizzare la libreria di testing Doctest, progettata per facilitare il processo di scrittura e gestione dei test attraverso un approccio semplice e diretto.