

Corso di Laboratorio di Programmazione

Esercitazione 8 Navigazione di un robot e smart pointer 13/12/2021

Discussione

Rispondete alle seguenti domande (Review, cap. 19&14):

1. What is a template?
2. How can you make a class abstract?
3. What is a virtual function and how does it differ from a non-virtual function?
4. What is a base class?
5. What makes a class derived?
6. How does a pure virtual function differ from other virtual functions?
7. What does overriding mean?

Esercizi

1. Completare il laboratorio della settimana scorsa

Avete il compito di programmare robot con vari comportamenti che cerchino di uscire da un labirinto di questo tipo:

```
*E*****
*          *
*   ***   *
* * *     *
* * ***   *
*   *     *
**  S  *  *
*          *
*****
```

Il labirinto è rappresentato da una matrice 9x9 che deve essere letta da file; si adotta la seguente codifica:

- gli asterischi rappresentano una posizione dove il robot non può andare;
- E rappresenta un'uscita – più uscite possono essere presenti;
- S rappresenta il punto in cui si trova il robot all'inizio della navigazione (una sola posizione presente per ciascun labirinto).

Il progetto è composto dalle seguenti classi:

- classe Maze: rappresenta il labirinto, gestisce la lettura da file e fornisce funzioni opportune per la navigazione (da progettare);
- classe Robot: rappresenta un robot; implementa la funzione virtuale move che accetta un argomento di tipo Maze& (da valutare se const o meno) che gestisce il movimento del robot in funzione dello specifico tipo di robot, gestito tramite le seguenti classi derivate;

- classe RandomRobot: derivata di Robot, gestisce un robot che effettua movimenti casuali tra le 8 caselle vicine alla posizione corrente;
- classe RightHandRuleRobot: derivata di Robot, gestisce un robot che si muove in modo che la sua mano destra sia sempre in contatto con una parete. Se la posizione iniziale non è a contatto con nessuna parete, si sceglie una direzione iniziale casuale che determina tutti i successivi spostamenti finché il robot non entra in contatto con una parete.

La funzione main gestisce l'interazione tra robot e labirinto.

Oltre a quanto richiesto, devono essere implementate le operazioni essenziali (lezione 9).

2. Smart pointer

Implementare un semplice unique_ptr che supporta solo:

- costruttore
- distruttore
- →
- *
- release()

Non implementare altre funzionalità (es., assegnamento, costruttore di copia, ecc.).

3. make_unique()

Durante il modulo sugli smart pointer abbiamo visto che una debolezza che non può essere gestita dal compilatore è l'inizializzazione di due unique_ptr con lo stesso puntatore (slide 15). Per risolvere questo problema è possibile usare make_unique(), che restituisce uno unique_ptr senza rendere accessibile il relativo puntatore. make_unique è una funzione template in cui il template specificato corrisponde al tipo puntato dallo unique_ptr, mentre gli argomenti sono utilizzati per inizializzare l'oggetto puntato. Leggere la documentazione:

https://docs.w3cub.com/cpp/memory/unique_ptr/make_unique

(con particolare attenzione alla sezione "example" e a come sono creati unique_ptr a oggetti di classe Vec3).

Successivamente:

- creare con make_unique un unique_ptr a uno std::vector<int> di 10 elementi (la dimensione deve essere passata in fase di costruzione);
- inizializzare tutti gli elementi con un ciclo;
- aggiungere altri 4 elementi con push_back;
- stampare tutti i valori del vettore.

4. Aggiungere un terzo tipo di robot

Sulla base della vostra esperienza, create una nuova politica di movimento e implementatela in una nuova classe, che può essere derivata di Robot (come le altre) oppure derivata di RandomRobot o RightHandRuleRobot. Scegliete un nome appropriato per la nuova classe. Descrivete **bene** la politica scelta nei commenti.

Note

Il progetto deve essere correttamente diviso in più file – ricordando le **include guards** per gli header; la compilazione deve avvenire **usando CMake**.

L'interfaccia tra Robot e Maze è solo parzialmente specificata nel testo. È possibile definire a piacere quanto non esplicitamente indicato.

È possibile implementare ulteriori tipi di robot creando nuove classi derivate.

La funzione move di Robot deve essere virtuale pura?