

Laboratorio 8 Navigazione di un robot

Discussione

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

- What is a template?
- How can you make a class abstract?
- What is a virtual function and how does it differ from a non-virtual function?
- What is a base class?
- What makes a class derived?
- How does a pure virtual function differ from other virtual functions?
- What does overriding mean?

Esercizio

Avete il compito di programmare robot che deve trovare l'uscita di un labirinto. 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;
- il carattere E rappresenta un'uscita – più uscite possono essere presenti;
- il carattere S rappresenta il punto in cui si trova il robot all'inizio della navigazione (una sola posizione presente per ciascun labirinto).

Un esempio di labirinto è il seguente:

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

Per uscire dal labirinto, il robot ha a disposizione due politiche di movimento:

- RandomRobot: un robot che effettua movimenti casuali tra le 8 caselle vicine alla posizione corrente;
- RightHandRuleRobot: 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.

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 in funzione dello specifico tipo di robot, gestito tramite le seguenti classi derivate:
 - classe `RandomRobot`: derivata di `Robot`, gestisce la relativa politica di movimento;
 - classe `RightHandRuleRobot`: derivata di `Robot`, gestisce la relativa politica di movimento.

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

Oltre a quanto richiesto, devono essere implementate le operazioni essenziali descritte a lezione.

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?