

DilemmaMH
Simulazione del dilemma di Monty Hall
Documentazione

Sandro Bellone

03/11/2015

Versione 0.03 - Novembre 2015

Scritto in linguaggio C con Dev C++ ver. 5.11

Compilato con TDM-GCC 4.9.2 32-bit e 64-bit

Copyright (C) 2015 Sandro Bellone.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Indice

1	Il dilemma di Monty Hall	2
2	Impiego del software	2
3	Compilazione dei sorgenti con ambiente di sviluppo	3
4	Compilazione manuale dei sorgenti	3

1 Il dilemma di Monty Hall

Il dilemma di Monty Hall ¹ è legato ad un quiz a premi statunitense, in cui il presentatore è noto con lo pseudonimo di “Monty Hall”. Il giocatore deve scegliere tra tre buste, di cui solo una comporta la vincita. Effettuata la scelta, il presentatore (che conosce qual’è la busta vincente) apre una delle buste perdenti (differente da quella scelta dal giocatore), e chiede al giocatore se intende cambiare busta o confermare quella scelta.

Convieni cambiare busta o confermare? Questo programma si propone di rispondere a questa domanda.

2 Impiego del software

Questo software effettua un certo numero di iterazioni, simulando lo svolgimento del gioco. Per ogni iterazione si genera un numero intero in maniera casuale da 1 a 3, che si riferisce alla busta vincente. Successivamente si genera un altro numero (scelta del giocatore). A questo punto, se è previsto il cambio di scelta, viene generato un terzo numero (busta perdente) e la scelta del giocatore viene spostata sul numero-busta rimanente. Si registrano i risultati (vincita o perdita, con o senza scambio) e si passa all’iterazione successiva.

E’ possibile scegliere, agendo sugli opportuni controlli grafici, il numero di iterazioni da effettuare, e quattro differenti politiche di cambio scelta da parte del giocatore:

- conferma la scelta randomicamente
- conferma sempre la scelta
- cambia sempre la scelta
- cambia la scelta la metà delle volte, alternativamente

E’ presente il bottone Reset, che riporta il programma alla configurazione iniziale (1000 iterazioni - conferma la scelta randomicamente). Nella cartella eseguibili sono presenti due differenti versioni del software già compilato, a 32 bit e 64 bit.

¹ https://it.wikipedia.org/wiki/Problema_di_Monty_Hall

3 Compilazione dei sorgenti con ambiente di sviluppo

Sono distribuiti i file:

1. dilemmamh.dev (progetto), creato utilizzando l'ambiente di sviluppo Dev C++ ver. 5.11 (<http://sourceforge.net/projects/orwelldvcpp>).
2. dilemmamh.c - codice sorgente
3. dilemmamh.rc - file resource
4. resource.h - macro di C per riferimenti ad elementi grafici

Per compilare il programma è necessario aprire il file dilemmamh.dev con Dev C++ e avviare la compilazione.

Per utilizzare un altro ambiente di sviluppo (ad esempio Code::Blocks), è necessario creare un nuovo progetto e aggiungere il file sorgente (.c) e il file resource (.rc). Non è necessario importare il file resource.h in quanto è presente già nella direttiva `#include` del file sorgente e del file resource.

Questo programma utilizza le librerie Win32 API per l'ambiente grafico, che normalmente sono incluse nei compilatori per Windows.

4 Compilazione manuale dei sorgenti

Si possono compilare i sorgenti senza l'ausilio di un ambiente di sviluppo, ma solo di un compilatore, come TDM-GCC o MinGW. Di seguito riporto le istruzioni necessarie.

La prima istruzione è la normale compilazione del codice sorgente, con generazione del file oggetto (.o).

```
gcc -c dilemmamh.c -o dilemmamh.o
```

Si compila quindi il file resource mediante un "resource compiler" (windres, normalmente incluso nelle suit di compilazione per Windows).

```
windres -i dilemmamh.rc -o dilemmamh.res -O coff
```

Infine l'ultima istruzione rappresenta il linker, con cui si inserisce il file resource compilato (.res) nel file eseguibile (.exe).

```
gcc dilemmamh.o dilemmamh.res -o dilemmamh.exe
```