

Esercitazione W6D1 - Pratica 2

Linguaggio C - Gioco Domande

Fabio Benevento - 28/11/2023

Scopo

Scopo dell'esercitazione è quello di riprendere il codice dell'esercitazione precedente relativo al gioco di domande e modificarlo al fine di ottimizzare il codice e soprattutto gestire eventuali casistiche non contemplate che possono dare luogo ad errori. Spesso sono proprio questi errori logici o di mancata gestione di situazioni non standard che possono essere sfruttate da un attaccante per prendere controllo dell'esecuzione del programma ed eseguire codice malevolo.

Implementazione

Partendo dal codice dell'esercitazione precedente l'ho ottimizzato aggiungendo alcune funzionalità come l'estrazione randomica delle domande (funzione `estrai_domande()`) in maniera da non avere più sempre le domande nello stesso ordine.

Ho inoltre aggiunto una serie di controlli nel codice per gestire alcune casistiche non standard.

In particolar modo:

- Nel menu di scelta iniziale l'utente potrebbe inserire un carattere diverso da A (Inizia partita) o B (Esci dal gioco) e, data l'implementazione precedente, proseguirebbe comunque con il programma. Ho quindi aggiunto un controllo al fine di permettere l'input solo dei caratteri A o B e di stampare un messaggio di errore in caso contrario al fine di invitare l'utente a procedere con una delle scelte possibili.
- La dimensione dell'array dove viene salvato il nome utente (tramite la funzione `scanf`) è di 15 caratteri. Se l'utente ne inserisce di più, si avrebbe un buffer-overflow con possibili comportamenti imprevedibili del programma fino alla sua chiusura improvvisa. Ho quindi sostituito la funzione `scanf` con la funzione `fgets` che ha già da sé un controllo sul numero di caratteri massimo da inserire nel buffer. Per un suo corretto utilizzo, ho

dovuto assicurarmi di ripulire lo standard input prima della sua chiamata e successivamente di rimuovere il carattere '\n' che viene comunque immagazzinato nel buffer dalla funzione fgets. Ho quindi raccolto il tutto in una specifica funzione leggi_stringa per poterla eventualmente riutilizzare in più parti del programma.

- Le possibili risposte alle domande sono 4 (A, B, C, D). L'utente in precedenza poteva rispondere con un carattere diverso e la risposta veniva considerata sbagliata. Ho aggiunto un controllo in maniera tale che se l'utente inserisce un carattere diverso da quelli attesi, venga stampato un messaggio di errore ed invitato a riprovare a rispondere alla domanda.

Di seguito è riportato il codice sorgente dell'applicazione con le modifiche descritte.

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <stdbool.h>
#include <string.h>
#include <time.h>

#define NUM_DOMANDE 10
#define MAX_RISPOSTE 4

#define MAX_LEN_NOME 15

// Array domande
char* domande[NUM_DOMANDE] = {
    "Qual è la capitale della Cina?",
    "Chi è l'autore del libro 'Il Nome della Rosa'?",
    "Chi ha dipinto la 'Notte Stellata'?",
    "Qual è il più grande oceano sulla Terra?",
    "Qual è l'organo responsabile della produzione dei globuli rossi nel corpo umano?",
    "Chi ha scritto 'Viaggio al centro della Terra'?",
    "Quale dei seguenti animali è un mammifero marino?",
    "Quale delle seguenti lingue non utilizza l'alfabeto latino?",
    "Quale grande fiume attraversa l'Antico Egitto?",
    "Qual è il metallo più pesante?",
};

// Array risposte
char* risposte[NUM_DOMANDE][MAX_RISPOSTE] = {
    {"Pechino", "Hong Kong", "Bangkok", "Shanghai"},
    {"Italo Calvino", "Giovanni Verga", "Umberto Eco", "Luigi Pirandello"},
    {"Vincent van Gogh", "Leonardo da Vinci", "Pablo Picasso", "Michelangelo Buonarroti"},
    {"Oceano Indiano", "Oceano Atlantico", "Oceano Artico", "Oceano Pacifico"},
    {"Milza", "Rene", "Midollo osseo", "Fegato"},
    {"Jules Verne", "Antoine de Saint-Exupéry", "Mark Twain", "Victor Hugo"},
    {"Squalo", "Balena", "Murena", "Anguilla"},
    {"Spagnolo", "Tedesco", "Greco", "Arabo"},
    {"Tigri", "Nilo", "Eufrate", "Gange"},
    {"Oro", "Piombo", "Uranio", "Platino"},
};

// Array risposte esatte
char risposte_esatte[NUM_DOMANDE] = {'A', 'C', 'A', 'D', 'C', 'A', 'B', 'D', 'B', 'C'};

void intestazione() {
    printf("*****\n");
    printf("    Benvenuto a    *\n");
    printf("  Chi vuol essere poveraccio *\n");
    printf("*****\n\n");
}

void presentazione() {
    printf("Scegli cosa vuoi fare:\n");
    printf("A) Iniziare una nuova partita\n");
    printf("B) Uscire dal gioco\n");
}

int esci() {
    printf("Grazie per aver giocato!\n");
    return 0;
}
```

```

void prepara_domanda(int indice_domanda) {
    if(indice_domanda < 0 || indice_domanda >= NUM_DOMANDE)
    {
        printf("Si è verificato un errore nella generazione della domanda\n");
        return;
    }

    char lettera_iniziale = 'A';

    printf("%s\n", domande[indice_domanda]);

    for(int i=0; i < MAX_RISPOSTE; i++) {
        printf("%c ", lettera_iniziale + i);
        printf("%s\n", risposte[indice_domanda][i]);
    }
}

void leggi_stringa(char* input)
{
    getchar(); //pulisce il buffer
    fgets(input, MAX_LEN_NOME, stdin); //legge stringa
    input[strlen(input) - 1] = '\0'; //rimuove lo '\n' finale letto da fgets
}

int contains(const int *array, int size, int target) {
    for (int i = 0; i < size; i++) {
        if (array[i] == target) {
            return 1; // Restituisce 1 se il numero è presente nell'array
        }
    }
    return 0; // Restituisce 0 se il numero non è presente nell'array
}

int estratti[NUM_DOMANDE];

void reinit_estratti()
{
    for(int i=0; i<NUM_DOMANDE; i++)
    {
        estratti[i] = 0;
    }
}

int estrai_domanda(int indice) {
    srand(time(NULL));
    int numero_casuale = 0;

    do
    {
        numero_casuale = rand() % NUM_DOMANDE + 1;
    }while(contains(estratti, sizeof(estratti)/sizeof(estratti[0]), numero_casuale));

    estratti[indice] = numero_casuale;

    return numero_casuale;
}

int main() {
    char scelta = '\0';
    bool continua = true;

    intestazione();

    while(continua) {
        int punteggio = 0;
        reinit_estratti();
        presentazione();
        scanf(" %c", &scelta);

        switch(toupper(scelta))

```

```

    estratti[indice] = numero_casuale;

    return numero_casuale;
}

int main() {
    char scelta = '\0';
    bool continua = true;

    intestazione();

    while(continua) {
        int punteggio = 0;
        reinit_estratti();
        presentazione();
        scanf(" %c", &scelta);

        switch(toupper(scelta))
        {
            case 'A':
                break;

            case 'B':
                continua = false;
                break;

            default:
                printf("Riposta non valida: Digita A o B\n\n");
                continue;
        }

        char nome[MAX_LEN_NOME] = {'\0'};
        printf("Inserisci il tuo nome: \n");
        leggi_stringa(nome);
        printf("Ciao, %s! Cominciamo la partita.\n\n", nome);

        int indice_prox_domanda = 0;

        for(int i=0; i < NUM_DOMANDE; i++) {
            printf("Estrai domanda %d\n", i);
            indice_prox_domanda = estrai_domanda(i);
            printf("Prossima domanda: %d\n", indice_prox_domanda);
            prepara_domanda(indice_prox_domanda - 1);
            char risposta = '\0';
            bool richiedi_risposta = false;
            do{
                richiedi_risposta = false;
                scanf(" %c", &risposta);
                if(toupper(risposta) < 'A' || toupper(risposta) > 'D')
                {
                    printf("Riposta non valida: Digita A, B, C o D\n\n");
                    richiedi_risposta = true;
                }
            }while(richiedi_risposta);

            // Valuta la risposta
            if (toupper(risposta) == risposte_esatte[i]) {
                punteggio++;
                printf("Risposta corretta!\n\n");
            } else {
                printf("Risposta sbagliata!\n\n");
            }
        }

        printf("Il tuo punteggio finale è: %d\n\n", punteggio);
    };

    return esci();
}

```