

EDU-K.O.

Il Simulatore di Sopravvivenza Scolastica

Capolavoro per l'Esame di Stato 2025

Nome Studente: Alessandro Privitera

Classe: 5^a Informatica

Istituto Tecnico Industriale "Galileo Ferraris"

Anno Scolastico 2024-2025

Indice

1	Introduzione	2
1.1	Motivazione e Obiettivi	2
1.2	Competenze Dimostrate	2
2	Analisi e Progettazione	3
2.1	Analisi del Problema	3
2.2	Metodologia di Sviluppo	3
2.3	Strumenti Utilizzati	3
3	Implementazione	3
3.1	Struttura del Programma	3
3.2	Strutture Dati	4
3.3	Flusso di Gioco	4
3.4	Meccaniche di Gioco	4
3.5	Interfaccia Utente	5
4	Codice Sorgente	5
4.1	Header e Definizioni	5
4.2	Strutture Dati e Array	5
4.3	Funzione Main	6
4.4	Funzioni di Gioco	7
5	Riflessioni e Considerazioni Tecniche	8
5.1	Scelte Implementative	8
5.2	Difficoltà Incontrate	8
5.3	Soluzioni Adottate	9
6	Valore Educativo e Sociale	9
6.1	Critica Costruttiva	9
6.2	Possibili Sviluppi	9
7	Conclusioni	10
8	Bibliografia e Riferimenti	10
A	Codice Completo	10
B	Istruzioni per la Compilazione ed Esecuzione	18

1 Introduzione

Questo progetto rappresenta una riflessione critica e provocatoria sul sistema scolastico italiano, tradotta in un videogioco testuale sviluppato in linguaggio C. *EDU-K.O. - Il Simulatore di Sopravvivenza Scolastica* è un'opera che, attraverso la programmazione e l'ironia, mette in luce le contraddizioni e le difficoltà che gli studenti affrontano quotidianamente.

Il programma simula una giornata tipica di uno studente italiano, costretto a navigare tra verifiche a sorpresa, aspettative irrealistiche, e un sistema che spesso sembra progettato per mettere alla prova non solo le conoscenze, ma anche la resistenza psicologica.

1.1 Motivazione e Obiettivi

La motivazione alla base di questo progetto nasce dalla mia esperienza personale e da quella condivisa con altri studenti. L'obiettivo non è semplicemente criticare, ma piuttosto:

- Stimolare una riflessione critica sul sistema educativo attuale
- Evidenziare, attraverso la gamification e l'ironia, problematiche reali che meritano attenzione
- Dimostrare come l'informatica possa essere uno strumento di espressione e critica sociale
- Proporre implicitamente un sistema educativo più umano ed empatico

1.2 Competenze Dimostrate

Questo progetto mi ha permesso di mettere in pratica e sviluppare diverse competenze acquisite durante il percorso di studi:

- **Sviluppo Software:** Progettazione e implementazione di un'applicazione interattiva in C
- **Algoritmi e Strutture Dati:** Utilizzo di struct per la gestione dei dati del giocatore
- **Interfacce Utente:** Creazione di un'interfaccia testuale intuitiva
- **Problem Solving:** Gestione delle diverse situazioni di gioco e dei relativi esiti
- **Pensiero Critico:** Analisi del sistema educativo e traduzione delle sue problematiche in meccaniche di gioco
- **Documentazione:** Capacità di presentare e documentare adeguatamente un progetto software

2 Analisi e Progettazione

2.1 Analisi del Problema

Il problema affrontato da questo progetto è duplice:

1. Come rappresentare efficacemente, attraverso un programma, le contraddizioni del sistema scolastico
2. Come creare un'esperienza interattiva che sia al contempo provocatoria ma anche coinvolgente

2.2 Metodologia di Sviluppo

Per lo sviluppo di questo progetto ho adottato un approccio iterativo:

1. Raccolta di esperienze e situazioni tipiche della vita scolastica
2. Classificazione di queste situazioni in categorie (mattino, lezioni, verifiche, ecc.)
3. Progettazione della struttura dati per rappresentare lo stato del giocatore
4. Implementazione delle singole situazioni di gioco
5. Testing e bilanciamento delle meccaniche di gioco
6. Rifinitura dell'interfaccia utente e dei messaggi

2.3 Strumenti Utilizzati

- **Linguaggio di Programmazione:** C
- **Ambiente di Sviluppo:** Visual Studio Code
- **Compilatore:** GCC (GNU Compiler Collection)
- **Controllo Versione:** Git
- **Documentazione:** \LaTeX

3 Implementazione

3.1 Struttura del Programma

Il programma è strutturato in diverse sezioni:

- Definizione delle strutture dati e costanti
- Funzioni di gestione dell'interfaccia utente
- Funzioni che simulano le diverse situazioni scolastiche
- Funzione principale che coordina il flusso di gioco

3.2 Strutture Dati

La principale struttura dati utilizzata è la struct **Studente**, che mantiene lo stato del giocatore:

```
1 typedef struct {  
2     char nome[NOME_LENGTH];  
3     int stress;  
4     float media;  
5     int ore_sonno;  
6     int assenze;  
7     int note_disciplinari;  
8     int voglia_vivere;  
9 } Studente;
```

Listing 1: Struttura dati Studente

Questa struttura permette di tracciare diversi parametri che rappresentano lo stato psicofisico e il rendimento scolastico del giocatore.

3.3 Flusso di Gioco

Il gioco si sviluppa attraverso una serie di situazioni che rappresentano momenti tipici della giornata scolastica:

1. Risveglio e preparazione mattutina
2. Lezioni in classe
3. Verifiche a sorpresa
4. Colloquio con i docenti
5. Ritorno a casa e gestione dei compiti
6. Valutazione finale della giornata

Per ogni situazione, il giocatore deve scegliere tra diverse opzioni, ciascuna con conseguenze diverse sui parametri del personaggio.

3.4 Meccaniche di Gioco

Le principali meccaniche di gioco includono:

- **Sistema di scelte:** Ogni situazione presenta al giocatore 3 opzioni
- **Parametri dinamici:** I parametri del giocatore (stress, media, ore di sonno, ecc.) vengono modificati in base alle scelte
- **Conseguenze a catena:** Le scelte fatte in una situazione influenzano le situazioni successive
- **Feedback immediato:** Dopo ogni scelta, il giocatore riceve un feedback sui risultati delle sue azioni
- **Visualizzazione stato:** Dopo ogni situazione, viene mostrato lo stato aggiornato dei parametri

3.5 Interfaccia Utente

L'interfaccia è minimalista e basata su testo, una scelta deliberata per enfatizzare il contenuto e il messaggio:

- Rappresentazione grafica dello stress e della voglia di vivere tramite barre di progresso testuali
- Menu di scelta numerici
- Messaggi ironici e provocatori che accompagnano ogni situazione

4 Codice Sorgente

4.1 Header e Definizioni

```
1 /*
2  * EDU-K.O. - Il Simulatore di Sopravvivenza Scolastica
3  * Un viaggio interattivo nell'assurda realtà del sistema educativo
4   *   italiano
5  *
6  * "Nel codice, come nella vita: lo studente perde sempre"
7  */
8
9 #include <stdio.h>
10 #include <stdlib.h>
11 #include <string.h>
12 #include <time.h>
13
14 // Configurazione del gioco
15 #define MAX_STRESS 100
16 #define MAX_VOTO 10
17 #define NOME_LENGTH 50
```

Listing 2: Header e definizioni

4.2 Strutture Dati e Array

```
1 // Struttura per lo studente virtuale
2 typedef struct {
3     char nome[NOME_LENGTH];
4     int stress;
5     float media;
6     int ore_sonno;
7     int assenze;
8     int note_disciplinari;
9     int voglia_vivere;
10 } Studente;
11
12 // Frasi ironiche del sistema
13 char* frasi_prof[] = {
14     "\"Non dovevi studiare solo l'ultima settimana\"",
15     "\"Io alla tua età studiavo 12 ore al giorno\"",
16     "\"Non ho tempo per spiegazioni individuali\"",
```

```
17     "\"La creatività non serve, servono le nozioni\"",
18     "\"Non è colpa mia se non capisci\"",
19     "\"Questo lo avete già fatto in precedenza\"",
20     "\"Nel mondo del lavoro non ti passano nulla\"",
21     "\"Dovete imparare a essere autonomi\""
22 };
23
24 char* situazioni_assurde[] = {
25     "Ti hanno assegnato 8 materie con verifica questa settimana",
26     "Il WiFi non funziona ma pretendono che usi il registro elettronico"
27 ,
28     "Ti chiedono di essere creativo ma poi ti bocciano se esci dal
programma",
29     "Devi fare 30 ore di alternanza scuola-lavoro non retribuite",
30     "Ti fanno studiare poesie del 1200 ma non come fare le tasse",
31     "La prof cambia le regole del compito durante il compito",
32     "Ti chiedono di avere 'spirito critico' ma solo se concordi con loro
"
33 };
```

Listing 3: Strutture dati e array

4.3 Funzione Main

```
1 int main() {
2     srand(time(NULL));
3
4     printf("\033[2J\033[1;1H"); // Pulisce lo schermo
5
6     intro();
7
8     Studente studente;
9     inizializza_studente(&studente);
10
11     printf("\n\n== EDU-K.O. - Il Simulatore di Sopravvivenza Scolastica
==\n");
12     printf("        \"Benvenuto nell'incubo quotidiano\"           \n\n
");
13
14     // Varie situazioni di una giornata scolastica
15     situazione_mattino(&studente);
16     situazione_lezione(&studente);
17     situazione_verifica(&studente);
18     situazione_colloquio(&studente);
19     situazione_casa(&studente);
20
21     // Risultato finale
22     risultato_finale(&studente);
23
24     return 0;
25 }
```

Listing 4: Funzione main

4.4 Funzioni di Gioco

Per motivi di spazio, riporto solo alcune delle funzioni principali. Il codice completo è disponibile nell'appendice.

```
1 void inizializza_studente(Studente *s) {
2     printf("\nInserisci il tuo nome: ");
3     fgets(s->nome, NOME_LENGTH, stdin);
4     s->nome[strcspn(s->nome, "\n")] = 0; // Rimuove newline
5
6     s->stress = 50;
7     s->media = 6.0;
8     s->ore_sonno = 5; // Realistico per uno studente italiano
9     s->assenze = 0;
10    s->note_disciplinari = 0;
11    s->voglia_vivere = 50;
12
13    printf("\nBene %s, benvenuto nella macchina tritacarne.\n", s->nome)
14    ;
15    pausa();
16 }
```

Listing 5: Funzione di inizializzazione

```
1 void mostra_status(Studente *s) {
2     printf("\n--- Status di %s ---\n", s->nome);
3     printf("Stress: %d/100 ", s->stress);
4     for(int i = 0; i < s->stress/10; i++) printf("\n");
5     printf("\n");
6
7     printf("Media scolastica: %.1f\n", s->media);
8     printf("Ore di sonno: %d\n", s->ore_sonno);
9     printf("Assenze: %d\n", s->assenze);
10    printf("Note disciplinari: %d\n", s->note_disciplinari);
11    printf("Voglia di vivere: %d% ", s->voglia_vivere);
12    for(int i = 0; i < s->voglia_vivere/20; i++) printf("♥");
13    printf("\n-----\n");
14 }
```

Listing 6: Funzione di visualizzazione stato

```
1 void situazione_verifica(Studente *s) {
2     printf("\n\n=== ORE 10:15 - VERIFICA A SORPRESA DI LATINO ===\n");
3     printf("'Ragazzi, compito in classe! Spero abbiate ripassato.'\n");
4     printf("Nessuno aveva annunciato nulla. Il panico si diffonde.\n");
5
6     printf("\nCome affronti la situazione?\n");
7     printf("1. Tenti di copiare dal tuo compagno\n");
8     printf("2. Scrivi qualsiasi cosa sperando nella piet \n");
9     printf("3. Consegni in bianco con dignit \n");
10
11    int scelta = scelta_utente(1, 3);
12
13    switch(scelta) {
14        case 1:
15            printf("\nLa prof ti becca dopo 2 minuti.\n");
16            printf("'%s! VERGOGNATI! Voti annullati per entrambi!'\n", s
17            ->nome);
18            printf("Il tuo compagno ti odia. Hai perso un amico.\n");
19            s->media = (s->media * 4 + 2) / 5;
20    }
```



```
19         s->stress += 35;
20         s->voglia_vivere -= 20;
21         s->note_disciplinari++;
22         break;
23
24     case 2:
25         printf("\nScrivi frasi con errori grammaticali italiani
26 nella versione latina.\n");
27         printf("La prof ti corregge con inchiostro rosso dappertutto
28 .\n");
29         printf("Voto: 4. 'Almeno ci hai provato' dice
30 sarcasticamente.\n");
31         s->media = (s->media * 4 + 4) / 5;
32         s->stress += 25;
33         s->voglia_vivere -= 10;
34         break;
35
36     case 3:
37         printf("\nLa prof apprezza l'onestà ma ti mette 2.\n");
38         printf("'Non puoi non sapere nulla dopo un anno di latino!'\n");
39         printf("(Come se un anno bastasse per imparare una lingua
40 morta)\n");
41         s->media = (s->media * 4 + 2) / 5;
42         s->stress += 30;
43         s->voglia_vivere -= 15;
44         break;
45     }
```

```
46 mostra_status(s);
47 pausa();
48 }
```

Listing 7: Esempio di situazione di gioco

5 Riflessioni e Considerazioni Tecniche

5.1 Scelte Implementative

Nella realizzazione del progetto, ho fatto diverse scelte implementative:

- **Utilizzo di struct:** Per mantenere i dati del giocatore in modo organizzato
- **Passaggio per riferimento:** Per modificare i parametri del giocatore nelle varie funzioni
- **Randomizzazione:** Per rendere ogni partita leggermente diversa (frasi dei professori, situazioni)
- **Interfaccia basata su terminale:** Per mantenere la semplicità e focalizzarsi sul contenuto

5.2 Difficoltà Incontrate

Durante lo sviluppo del progetto, ho affrontato alcune difficoltà:

- Bilanciamento delle meccaniche: trovare il giusto equilibrio tra difficoltà e giocabilità
- Gestione dell'input utente: prevenire input non validi e comportamenti inaspettati
- Aspetti narrativi: creare situazioni realistiche ma anche provocatorie

5.3 Soluzioni Adottate

Per risolvere queste difficoltà ho:

- Implementato una funzione robusta per la gestione dell'input utente
- Testato il programma su diverse piattaforme (Windows, Linux)
- Raccolto feedback da altri studenti per calibrare le situazioni di gioco
- Utilizzato array di stringhe per gestire facilmente le frasi e le situazioni random

6 Valore Educativo e Sociale

6.1 Critica Costruttiva

Sebbene il progetto adotti un tono ironico e provocatorio, non intende essere una semplice critica fine a se stessa. L'obiettivo è stimolare una riflessione su:

- L'impatto psicologico che il sistema attuale può avere sugli studenti
- La discrepanza tra gli obiettivi dichiarati dell'educazione e le pratiche quotidiane
- La mancanza di attenzione verso il benessere psicologico degli studenti
- L'importanza di un approccio più umano e comprensivo nell'educazione

6.2 Possibili Sviluppi

Questo progetto potrebbe evolversi in diversi modi:

- Implementazione di una versione grafica con interfaccia più elaborata
- Espansione delle situazioni di gioco per coprire l'intero anno scolastico
- Creazione di una versione "utopica" che simuli un sistema educativo ideale
- Traduzione in formato web per renderlo accessibile a più persone
- Raccolta di dati anonimi per analizzare le scelte più comuni e le situazioni più stressanti

7 Conclusioni

EDU-K.O. - Il Simulatore di Sopravvivenza Scolastica rappresenta un esempio di come la programmazione possa essere utilizzata non solo per creare applicazioni funzionali, ma anche per esprimere idee, stimolare riflessioni e proporre critiche costruttive.

Attraverso questo progetto, ho voluto dimostrare come le competenze tecniche acquisite durante il mio percorso di studi possano essere applicate in modo creativo e significativo. La programmazione diventa così non solo uno strumento tecnico, ma anche un mezzo di espressione e comunicazione.

In un'epoca in cui il sistema educativo è in continua evoluzione, ritengo che sia importante che noi studenti contribuiamo al dibattito con le nostre prospettive e i nostri talenti. Questo progetto è il mio piccolo contributo a questa importante conversazione.

8 Bibliografia e Riferimenti

1. Kernighan, B. W., & Ritchie, D. M. (1988). *The C Programming Language*. Prentice Hall.
2. Ministero dell'Istruzione (2024). *Nota MIM 7557/2024 - Piattaforma UNICA*.
3. Adams, S. (1996). *The Dilbert Principle*. Harper Business.
4. Prensky, M. (2001). *Digital Game-Based Learning*. McGraw-Hill.
5. Robinson, K. (2015). *Creative Schools: The Grassroots Revolution That's Transforming Education*. Viking.

A Codice Completo

Il codice completo del progetto è disponibile qua sotto ed è possibile eseguire il programma compilando il seguente file:

```
1 /*
2  * EDU-K.O. - Il Simulatore di Sopravvivenza Scolastica
3  * Un viaggio interattivo nell'assurda realtà del sistema educativo
4   *   italiano
5  *
6  * "Nel codice, come nella vita: lo studente perde sempre"
7  */
8
9 #include <stdio.h>
10 #include <stdlib.h>
11 #include <string.h>
12 #include <time.h>
13
14 // Configurazione del gioco
15 #define MAX_STRESS 100
16 #define MAX_VOTO 10
17 #define NOME_LENGTH 50
18
19 // Struttura per lo studente virtuale
20 typedef struct {
```

```
20     char nome[NOME_LENGTH];
21     int stress;
22     float media;
23     int ore_sonno;
24     int assenze;
25     int note_disciplinari;
26     int voglia_vivere;
27 } Studente;
28
29 // Frasi ironiche del sistema
30 char* frasi_prof[] = {
31     "\"Non dovevi studiare solo l'ultima settimana\"",
32     "\"Io alla tua età studiavo 12 ore al giorno\"",
33     "\"Non ho tempo per spiegazioni individuali\"",
34     "\"La creatività non serve, servono le nozioni\"",
35     "\"Non è colpa mia se non capisci\"",
36     "\"Questo lo avete già fatto in precedenza\"",
37     "\"Nel mondo del lavoro non ti passano nulla\"",
38     "\"Dovete imparare a essere autonomi\""
39 };
40
41 char* situazioni_assurde[] = {
42     "Ti hanno assegnato 8 materie con verifica questa settimana",
43     "Il WiFi non funziona ma pretendono che usi il registro elettronico"
44     ,
45     "Ti chiedono di essere creativo ma poi ti bocciano se esci dal programma",
46     "Devi fare 30 ore di alternanza scuola-lavoro non retribuite",
47     "Ti fanno studiare poesie del 1200 ma non come fare le tasse",
48     "La prof cambia le regole del compito durante il compito",
49     "Ti chiedono di avere 'spirito critico' ma solo se concordi con loro"
50     "
51 };
52
53 // Prototipi funzioni
54 void inizializza_studente(Studente *s);
55 void mostra_status(Studente *s);
56 void situazione_mattino(Studente *s);
57 void situazione_lezione(Studente *s);
58 void situazione_verifica(Studente *s);
59 void situazione_colloquio(Studente *s);
60 void situazione_casa(Studente *s);
61 void risultato_finale(Studente *s);
62 void intro();
63 void pausa();
64 int scelta_utente(int min, int max);
65
66 int main() {
67     srand(time(NULL));
68
69     printf("\033[2J\033[1;1H"); // Pulisce lo schermo
70
71     intro();
72
73     Studente studente;
74     inizializza_studente(&studente);
```

```
74     printf("\n\n=== EDU-K.O. - Il Simulatore di Sopravvivenza Scolastica\n\n");
75     printf("        \"Benvenuto nell'incubo quotidiano\"        \n\n");
76
77     // Varie situazioni di una giornata scolastica
78     situazione_mattino(&studente);
79     situazione_lezione(&studente);
80     situazione_verifica(&studente);
81     situazione_colloquio(&studente);
82     situazione_casa(&studente);
83
84     // Risultato finale
85     risultato_finale(&studente);
86
87     return 0;
88 }
89
90 void inizializza_studente(Studente *s) {
91     printf("\nInserisci il tuo nome: ");
92     fgets(s->nome, NOME_LENGTH, stdin);
93     s->nome[strcspn(s->nome, "\n")] = 0; // Rimuove newline
94
95     s->stress = 50;
96     s->media = 6.0;
97     s->ore_sonno = 5; // Realistico per uno studente italiano
98     s->assenze = 0;
99     s->note_disciplinari = 0;
100    s->voglia_vivere = 50;
101
102    printf("\nBene %s, benvenuto nella macchina tritacarne.\n", s->nome);
103    ;
104    pausa();
105 }
106
107 void mostra_status(Studente *s) {
108     printf("\n--- Status di %s ---\n", s->nome);
109     printf("Stress: %d/100 ", s->stress);
110     for(int i = 0; i < s->stress/10; i++) printf("\n");
111     printf("\n");
112
113     printf("Media scolastica: %.1f\n", s->media);
114     printf("Ore di sonno: %d\n", s->ore_sonno);
115     printf("Assenze: %d\n", s->assenze);
116     printf("Note disciplinari: %d\n", s->note_disciplinari);
117     printf("Voglia di vivere: %d%% ", s->voglia_vivere);
118     for(int i = 0; i < s->voglia_vivere/20; i++) printf("♥");
119     printf("\n-----\n");
120 }
121
122 void situazione_mattino(Studente *s) {
123     printf("\n\n=== ORE 06:30 - LA SVEGLIA SUONA ===\n");
124     printf("Hai dormito %d ore. Ti senti distrutto.\n", s->ore_sonno);
125
126     // Situazione assurda random
127     printf("\n%s\n", situazioni_assurde[rand() % 7]);
128
129     printf("\nCosa fai?\n");
```

```
129 printf("1. Vai a scuola da zombie\n");
130 printf("2. Fingi di essere malato\n");
131 printf("3. Piangi nel cuscino per 10 minuti e poi vai\n");
132
133 int scelta = scelta_utente(1, 3);
134
135 switch(scelta) {
136     case 1:
137         printf("\nArrivi in classe con le occhiaie fino ai piedi.\n"
138 );
139         printf("La prof ti guarda e dice: %s\n", frasi_prof[rand() %
140 8]);
141         s->stress += 15;
142         s->voglia_vivere -= 10;
143         break;
144     case 2:
145         printf("\nTua madre non ci crede e ti obbliga ad andare.\n")
146 ;
147         printf("Arrivi in ritardo e prendi una nota.\n");
148         s->note_disciplinari++;
149         s->stress += 20;
150         s->assenze++; // Contano come assenza anche se vai
151         break;
152     case 3:
153         printf("\nArrivi con gli occhi rossi. Il bidello ti chiede
154 se va tutto bene.\n");
155         printf("È l'unica persona che si preoccupa per te oggi.\n");
156         s->stress += 10;
157         s->voglia_vivere -= 5;
158         break;
159 }
160
161 mostra_status(s);
162 pausa();
163 }
164
165 void situazione_lezione(Studiante *s) {
166     printf("\n\n== ORE 08:15 - PRIMA ORA: MATEMATICA ==\n");
167     printf("La prof entra e dice: 'Spero abbiate studiato il capitolo
168 7'.\n");
169     printf("Tu ovviamente hai studiato il capitolo 6.\n");
170
171     printf("\nCosa fai?\n");
172     printf("1. Confessi di aver studiato il capitolo sbagliato\n");
173     printf("2. Fingi di sapere e speri di non essere interrogato\n");
174     printf("3. Fingi un malore e vai in infermeria\n");
175
176     int scelta = scelta_utente(1, 3);
177
178     switch(scelta) {
179         case 1:
180             printf("\nLa prof: 'COME È POSSIBILE? È SUL REGISTRO DA DUE
181 SETTIMANE!'\n");
182             printf("Vieni interrogato comunque e prendi 3.\n");
183             s->media = (s->media * 4 + 3) / 5;
184             s->stress += 25;
```

```
181         s->voglia_vivere -= 15;
182         break;
183
184     case 2:
185         printf("\nViene chiamato il tuo compagno di banco.\n");
186         printf("Mentre lui balbetta le risposte, la prof nota che
187 stai leggendo sotto il banco.\n");
188         printf("Nota disciplinare per 'comportamento scorretto'.\n")
189 ;
190         s->note_disciplinari++;
191         s->stress += 20;
192         break;
193
194     case 3:
195         printf("\nIn infermeria non c'è nessuno (come al solito).\n"
196 );
197         printf("Ti siedi e apri il libro di matematica, ma è tardi
198 per recuperare.\n");
199         s->stress += 15;
200         s->ore_sonno--; // Lo stress ti sta consumando
201         break;
202     }
203
204     mostra_status(s);
205     pausa();
206 }
207
208 void situazione_verifica(Studiante *s) {
209     printf("\n\n=== ORE 10:15 - VERIFICA A SORPRESA DI LATINO ===\n");
210     printf("'Ragazzi, compito in classe! Spero abbiate ripassato.'\n");
211     printf("Nessuno aveva annunciato nulla. Il panico si diffonde.\n");
212
213     printf("\nCome affronti la situazione?\n");
214     printf("1. Tenti di copiare dal tuo compagno\n");
215     printf("2. Scrivi qualsiasi cosa sperando nella pietà\n");
216     printf("3. Consegni in bianco con dignità\n");
217
218     int scelta = scelta_utente(1, 3);
219
220     switch(scelta) {
221     case 1:
222         printf("\nLa prof ti becca dopo 2 minuti.\n");
223         printf("'%s! VERGOGNATI! Voti annullati per entrambi!'\n", s
224 ->nome);
225         printf("Il tuo compagno ti odia. Hai perso un amico.\n");
226         s->media = (s->media * 4 + 2) / 5;
227         s->stress += 35;
228         s->voglia_vivere -= 20;
229         s->note_disciplinari++;
230         break;
231
232     case 2:
233         printf("\nScrivi frasi con errori grammaticali italiani
234 nella versione latina.\n");
235         printf("La prof ti corregge con inchiostro rosso dappertutto
236 .\n");
237         printf("Voto: 4. 'Almeno ci hai provato' dice
238 sarcasticamente.\n");
```

```
231     s->media = (s->media * 4 + 4) / 5;
232     s->stress += 25;
233     s->voglia_vivere -= 10;
234     break;
235
236     case 3:
237         printf("\nLa prof apprezza l'onestà ma ti mette 2.\n");
238         printf("'Non puoi non sapere nulla dopo un anno di latino!'\n");
239         printf("(Come se un anno bastasse per imparare una lingua
240 morta)\n");
241         s->media = (s->media * 4 + 2) / 5;
242         s->stress += 30;
243         s->voglia_vivere -= 15;
244         break;
245     }
246     mostra_status(s);
247     pausa();
248 }
249
250 void situazione_colloquio(Studente *s) {
251     printf("\n\n=== ORE 13:00 - COLLOQUIO CON LA COORDINATRICE ===\n");
252     printf("'%s, vieni un momento. Dobbiamo parlare.'\n", s->nome);
253     printf("Il cuore ti si ferma. Cosa ho fatto adesso?\n");
254
255     printf("\nCome ti presenti?\n");
256     printf("1. Umile e pentito (anche se non sai di cosa)\n");
257     printf("2. Sicuro di te e pronto al confronto\n");
258     printf("3. Rassegnato al destino\n");
259
260     int scelta = scelta_utente(1, 3);
261
262     switch(scelta) {
263         case 1:
264             printf("\nProf: 'Ho notato un calo nel tuo rendimento.'\n");
265             printf("Tu: 'Ha ragione prof, mi impegnerò di più.'\n");
266             printf("Prof: 'Bene, voglio vedere miglioramenti o chiamo i
267 tuoi genitori.'\n");
268             s->stress += 20;
269             s->voglia_vivere -= 10;
270             break;
271
272         case 2:
273             printf("\nTu: 'Se c'è un problema, parliamone apertamente.'\n");
274             printf("Prof: 'Il problema è il tuo atteggiamento! Non ti
275 rendi conto della gravità!'\n");
276             printf("Escalation. Nota disciplinare. I tuoi genitori
277 vengono avvisati.\n");
278             s->note_disciplinari++;
279             s->stress += 35;
280             s->voglia_vivere -= 20;
281             break;
282
283         case 3:
284             printf("\nTu: 'Lo so, sono un fallimento.'\n");
```



```
282         printf("Prof: 'Non dire così... ma effettivamente devi
reagire.'\n");
283         printf("Ti senti peggio di prima. Almeno non hai preso una
nota.\n");
284         s->stress += 15;
285         s->voglia_vivere -= 15;
286         break;
287     }
288
289     mostra_status(s);
290     pausa();
291 }
292
293 void situazione_casa(Studente *s) {
294     printf("\n\n=== ORE 15:30 - FINALMENTE A CASA ===\n");
295     printf("Guardi l'agenda: 3 compiti scritti, 4 capitoli da studiare,
ricerca di gruppo.\n");
296     printf("Domani interrogazioni in 2 materie.\n");
297
298     printf("\nCome organizzi il pomeriggio?\n");
299     printf("1. Studio 6 ore filate per recuperare\n");
300     printf("2. Fai una pausa e poi studi (rischi di non finire)\n");
301     printf("3. Ti arrendi e guardi Netflix\n");
302
303     int scelta = scelta_utente(1, 3);
304
305     switch(scelta) {
306         case 1:
307             printf("\nStudi fino alle 21:30. Riesci a fare quasi tutto.\n
n");
308             printf("Ma hai mal di testa e non hai cenato bene.\n");
309             printf("Ti addormenti alle 01:00. Domani saranno 5 ore di
sonno.\n");
310             s->stress += 25;
311             s->ore_sonno = 5;
312             s->voglia_vivere -= 15;
313             s->media += 0.2; // Piccolo miglioramento nella media
314             break;
315
316         case 2:
317             printf("\nTi riposi 1 ora, poi cominci a studiare.\n");
318             printf("Alle 23:00 ti accorgi che hai fatto solo metà.\n");
319             printf("Panico. Studi fino alle 02:00 in ansia.\n");
320             s->stress += 35;
321             s->ore_sonno = 4;
322             s->voglia_vivere -= 20;
323             break;
324
325         case 3:
326             printf("\nGuardi una serie fino a tardi. Ti senti in colpa.\n
n");
327             printf("Domani sarà un disastro, ma almeno stasera respiri.\n
n");
328             printf("Ansia da domani: già pensi alle scuse per i prof.\n"
);
329             s->stress += 40;
330             s->ore_sonno = 6;
331             s->voglia_vivere -= 10;
```

```
332         s->media -= 0.3; // Peggioramento della media
333         break;
334     }
335
336     mostra_status(s);
337     pausa();
338 }
339
340 void risultato_finale(Studiante *s) {
341     printf("\n\n=== FINE GIORNATA - RIEPILOGO ESISTENZIALE ===\n");
342
343     printf("\n STATISTICHE DI %s:\n", s->nome);
344     printf("-----\n");
345     mostra_status(s);
346
347     printf("\n ANALISI PSICOLOGICA:\n");
348     if(s->stress >= 90) {
349         printf("Sei vicino al burnout. Il sistema ha quasi vinto.\n");
350     } else if(s->stress >= 70) {
351         printf("Il peso del sistema ti sta schiacciando.\n");
352     } else if(s->stress >= 50) {
353         printf("Sopravvivi, ma a quale prezzo?\n");
354     } else {
355         printf("Hai resistito, ma domani si ricomincia.\n");
356     }
357
358     printf("\n REPORT GIORNALIERO:\n");
359     printf("- Media scolastica: %.1f (", s->media);
360     if(s->media >= 6.0) printf("Sufficiente, ma a che costo?\n");
361     else printf("Insufficiente, come prevedibile)\n");
362
363     printf("- Ore di sonno: %d (", s->ore_sonno);
364     if(s->ore_sonno >= 8) printf("Miracolo!\n");
365     else if(s->ore_sonno >= 6) printf("Almeno qualcosa)\n");
366     else printf("Zombie mode attivo)\n");
367
368     printf("- Stress accumulato: %d/100\n", s->stress);
369     printf("- Voglia di vivere: %d%%\n", s->voglia_vivere);
370
371     printf("\n CONCLUSIONE:\n");
372     printf("-----\n");
373     printf("Un'altra giornata di sopravvivenza completata.\n");
374     printf("La scuola ti prepara alla vita... \n");
375     printf("...ti prepara a soffrire.\n\n");
376
377     printf("Il sistema non è fatto per lo studente,\n");
378     printf("lo studente è fatto per il sistema.\n\n");
379
380     printf("\n\"Nel mondo del lavoro sarà uguale\" dicono.\n");
381     printf("E se fosse il momento di cambiare tutto?\n\n");
382
383     printf("Grazie per aver giocato a EDU-K.O.\n");
384     printf("Domani, stessa ora, stessa sofferenza.\n");
385     printf("-----\n");
386 }
387
388 void intro() {
389     printf("\n");
```

```
390     printf("\n");
391     printf("                Il Simulatore di Sopravvivenza Scolastica\n");
392     printf("                \"Benvenuto nel sistema che vuole distruggerti\"\n\n");
393 }
394
395 void pausa() {
396     printf("\nPremi INVIO per continuare...");
397     getchar();
398 }
399
400 int scelta_utente(int min, int max) {
401     int scelta;
402     char buffer[100];
403
404     while(1) {
405         printf("\nScelta (inserisci il numero): ");
406         if(fgets(buffer, sizeof(buffer), stdin) != NULL) {
407             scelta = atoi(buffer);
408             if(scelta >= min && scelta <= max) {
409                 return scelta;
410             }
411         }
412         printf("Scelta non valida. Riprova.\n");
413     }
414 }
```

Listing 8: Codice completo (file Capolavoro2.c)

B Istruzioni per la Compilazione ed Esecuzione

Per compilare ed eseguire il programma, seguire questi passaggi:

1. Aprire un terminale
2. Navigare nella directory contenente il file sorgente
3. Eseguire il comando di compilazione: `gcc -o edu-ko Capolavoro2.c`
4. Eseguire il programma: `./edu-ko`