

ESCAPE GAME

Gianvito Peragine 738753

Sandro Palermo 738749

Federico Mondelli 737322

Indice

1) Avventura grafica	3
1.1) Introduzione e trama del gioco	3
1.2) Mappa del gioco	3
2) Documentazione tecnica	4
2.1) Architettura del sistema	4
2.2) Strutture dati (Caratteristiche tecniche)	4
2.3) Diagrammi delle classi	5
2.4) Dettagli di implementazione	6
2.5) Strutture dati: Lista	6
2.5.1) Specifica Sintattica	6
2.5.2) Specifica Algebrica	7
2.5.3) Specifica Semantica	7
3) Manuale dell'utente	8
3.1) Tasto 'Gioca'	8
3.2) Tasto 'Comandi'	10
3.3) Tasto 'Storico'	11
3.4) Tasto 'Chi siamo'	11
4) Informazioni utili al completamento del gioco	12

1) Avventura grafica

1.1) Introduzione e trama del gioco

Lo stile di gioco utilizzato è un'avventura testuale. L'avventura è svolta in prima persona e l'interazione tra il gioco e l'utente è gestita esclusivamente da linea di comando. Il menù è gestito interamente con un'interfaccia grafica che permette al giocatore di:

- Giocare, previo inserimento dei propri dati
- Visualizzare l'elenco dei comandi fornito dall'helper del gioco
- Visualizzare lo storico dei giocatori
- Visualizzare la presentazione del team di sviluppo

Il gioco ha una trama apparentemente semplice: il personaggio è stato rapito e trasportato in una casa abbandonata. L'obiettivo principale del giocatore è quello di fuggire dalla casa abbandonata prima che arrivino i suoi rapinatori.

Al protagonista è permesso potersi spostare tra i vari ambienti, a meno che non ci sia un muro tra di essi, digitando la direzione in cui si vuole andare (nord, sud, est, ovest). Inoltre è permesso aprire, prendere e utilizzare determinati oggetti.

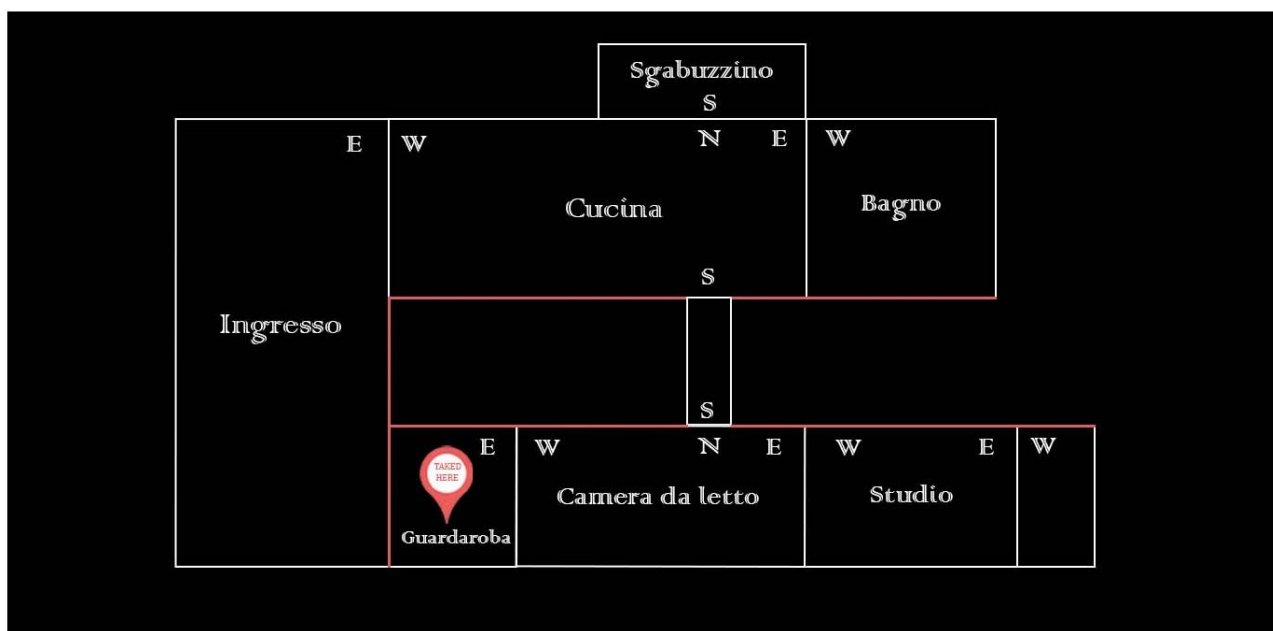
All'interno della casa possono essere presenti degli oggetti che possono aiutare alla fuga, degli indizi e degli enigmi.

ATTENZIONE: Per l'apertura o l'utilizzo di alcuni oggetti, è necessario possederne altri.

Non è detto che il giocatore riesca a terminare il gioco in quanto sono presenti anche degli esplosivi con un proprio timer, che limiterà il restante tempo in cui il giocatore deve trovare necessariamente la via di fuga.

1.2) Mappa del gioco

La mappa della casa in cui si è stati abbandonati non è disponibile dall'inizio del gioco ma è stata nascosta all'interno della casa stessa. Una volta recuperata, è possibile visualizzarla digitando il comando "apri mappa".



2) Documentazione tecnica

2.1) Architettura del sistema

L'avventura è stata realizzata con il linguaggio di programmazione Java e presenta un'architettura di sistema suddivisa in diversi gruppi (package)

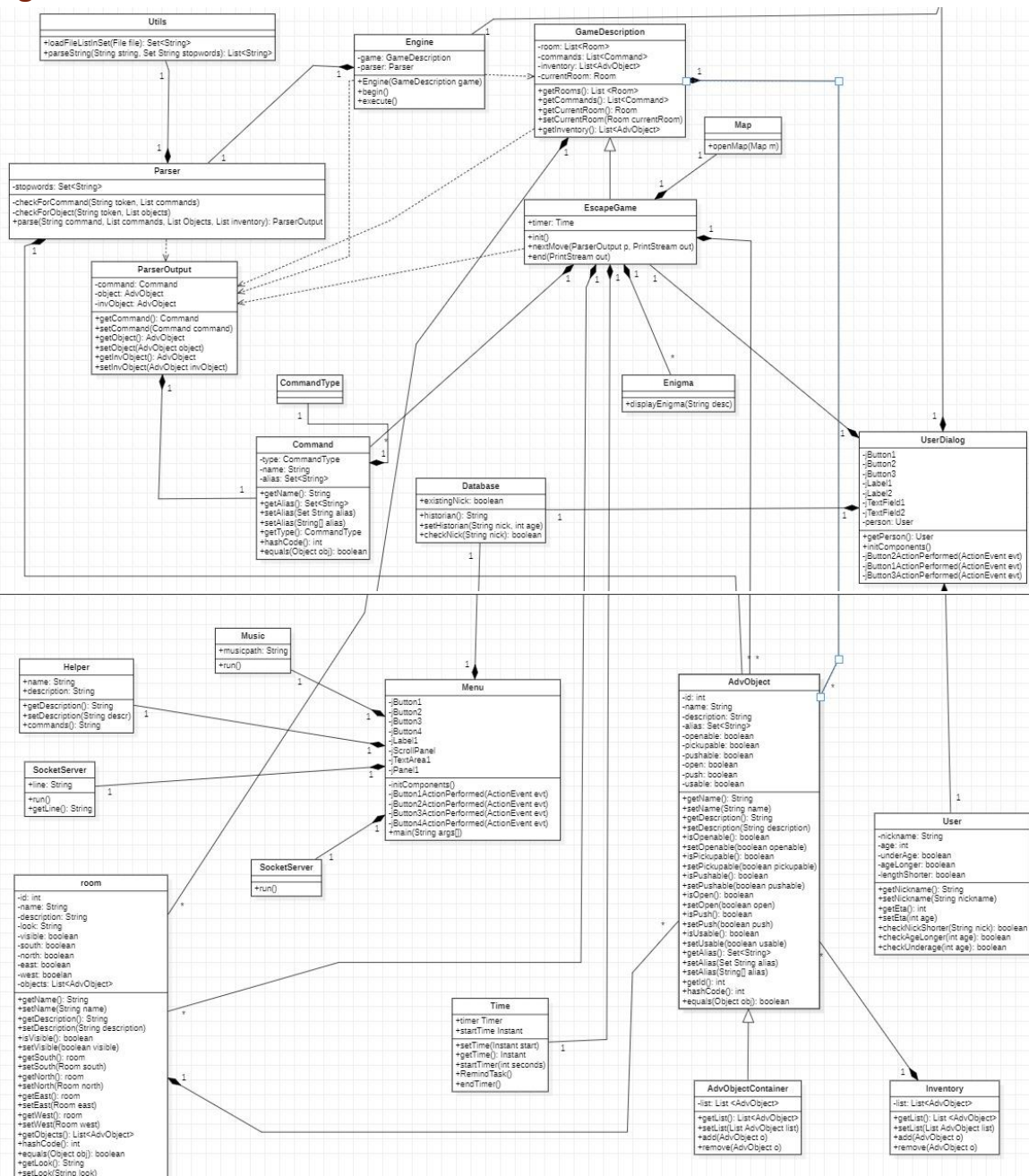
- **Adventure:** contiene il motore del progetto.
- **Database:** contiene tutto ciò che riguarda l'implementazione dei database
- **Games:** contiene le meccaniche di gioco.
- **Graphic:** contiene tutto ciò che riguarda l'interfaccia grafica.
- **Music:** contiene il necessario per l'implementazione di una colonna sonora.
- **Parser:** contiene tutto ciò che riguarda il parser.
- **Socket:** contiene il necessario per l'implementazione della programmazione in rete
- **Type:** contiene tutte le strutture dati necessarie.

2.2) Tipologie di dati e caratteristiche tecniche

Per implementare le diverse funzionalità del motore, sono state definite le seguenti strutture dati:

- **AdvObject:** Definisce tutti gli oggetti del gioco.
- **AdvObjectContainer:** Definisce tutti i contenitori di oggetti del gioco. Un esempio di questi è la cassaforte.
- **Helper:** Definisce le caratteristiche dell'helper e l'elenco dei comandi.
- **User:** Definisce le caratteristiche dell'utente.
- **Command:** Definisce i comandi di gioco
- **Inventory:** Definisce la composizione dell'inventario del personaggio principale
- **Time:** Definisce le caratteristiche del timer, utile quando viene azionata la bomba.
- **Room:** Definisce le caratteristiche delle stanze

2.3) Diagrammi delle classi



2.4) Dettagli di implementazione

L'interfaccia grafica dell'avventura è stata implementata utilizzando gli elementi di "javaw.swing", un noto framework di Java che permette la realizzazione di interfacce grafiche (GUI).

La memorizzazione delle informazioni dell'utente (id, nickname ed età) è stata implementata con l'utilizzo del **DBMS H2**. E' stato, inoltre, implementato l'utilizzo dei **Socket** quando l'utente vorrà conoscere la storia del team di sviluppo. Viene effettuata una richiesta al server che restituirà la descrizione richiesta.

Il Socket Server e il Socket Client generalmente non risiedono sulla stessa macchina ma è stato deciso di implementarli sulla stessa macchina poiché non è stato possibile testare il gioco su due macchine diverse.

I socket, così come l'implementazione della colonna sonora e del timer, sono stati implementati utilizzando i **Thread**. Questi ultimi sono necessari per implementare la programmazione concorrente ed eseguire più task temporaneamente condividendo le stesse risorse di memoria e di Input/Output.

2.5) Strutture dati: Lista

Una lista è una sequenza finita, anche vuota, di elementi dello stesso tipo. Nella lista uno stesso elemento può comparire più volte in posizioni diverse. Essa è un esempio di accesso per scansione, ovvero consente l'accesso all'elemento generico solo dopo aver scandito gli elementi che lo precedono. A ciascun elemento della lista viene associata una posizione ($pos(i)$) e un valore ($a(i)$). La lista è una struttura dati dinamica perchè è possibile inserire e cancellare elementi; poiché la lista è a dimensione variabile, queste operazioni che alterano la dimensione sono fondamentali.

2.5.1) Specifica Sintattica

OPERATORI	DESCRIZIONE
creaLista() -> lista	crea una lista vuota
listaVuota(lista) -> boolean	controlla se la lista è vuota
leggiLista(posizione, lista) -> tipoelem	legge un elemento della lista
scriviLista(tipoelem, posizione, lista) -> lista	aggiunge un elemento nella lista
primoLista(lista) -> posizione	restituisce il primo elemento della lista
fineLista(posizione, lista) -> boolean	controlla se l'elemento è l'ultimo della lista
succLista(posizione, lista) -> posizione	passa all'elemento successivo della lista
predLista(posizione, lista) -> posizione	passa all'elemento precedente della lista
insLista(tipoelem, posizione, lista) -> lista	aggiunge un altro campo alla lista
canclista(posizione, lista) -> lista	cancella un campo dalla lista

2.5.2) Specifica Algebrica

Supponendo che L sia una lista, e sia un tipo elem, p una posizione e b un boolean:

OSSERVAZIONI		COSTRUTTORI
	creaLista()	inserisciLista(e,p,L)
listaVuota(L')	true	false
scriviLista(e',p',L')	Error	inserisciLista(e',p',cancellaLista(p',L'))
cancellaLista(p',L')	Error	If(p'=p) then L else inserisciLista (e, p, cancellaLista(p',L))
leggiLista(p',L')	Error	If(p'=p) then L else leggiLista(p',L')

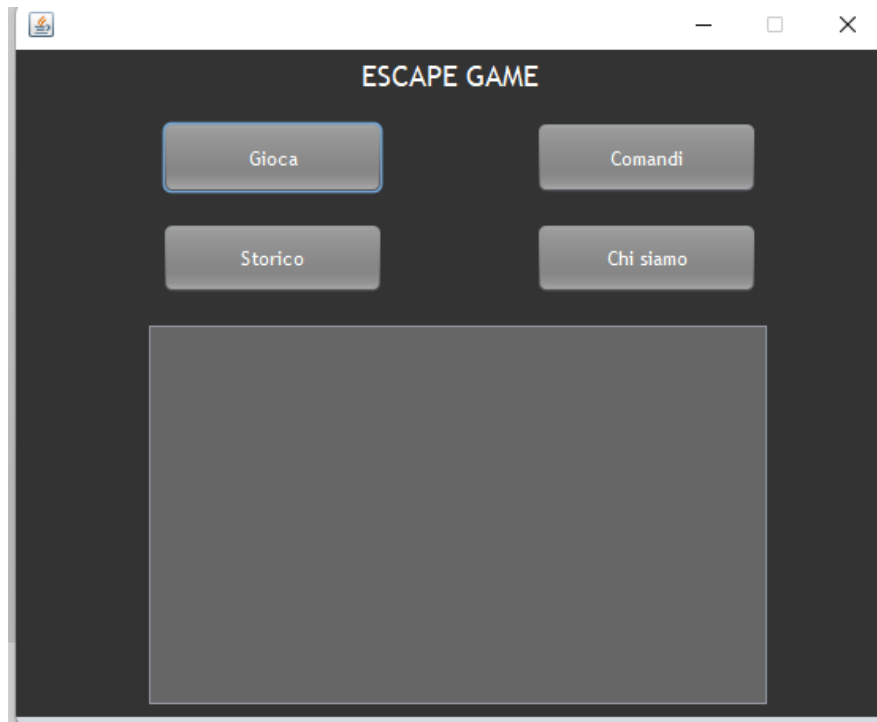
2.5.3) Specifica Semantica

Supponendo che L sia una lista, e sia un tipo elem, b un boolean:

OPERATORI	
	<i>Studenti</i>
listaVuota(creaLista)	True
listaVuota(scriviLista(e,p,L))	False
leggiLista(p,scriviLista(e,p,L))	If listaVuota(L) then e else leggiLista(p,L)
cancellaLista(p,scriviLista(e,p,L))	If listaVuota(L) then creaLista() else scriviLista(e,p,cancellaLista(p,L))

3) Manuale dell'utente

All'avvio dell'applicazione verrà mostrato all'utente un menù da cui sarà possibile scegliere diverse opzioni.



Saranno presenti, per questa versione, quattro tasti: *'Gioca'*, *'Comandi'*, *'Storico'*, *'Chi siamo'*. Selezionando la prima opzione verrà eseguita l'avventura di default pre-caricata nel menu principale.

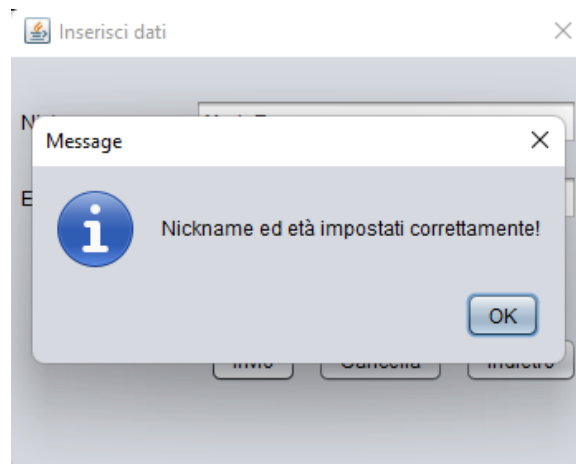
3.1) Tasto 'Gioca'

A screenshot of a dialog box titled "Inserisci dati". The dialog box has a light gray background and a close button (X) in the top right corner. It contains two input fields: "Nickname:" with the text "MarioR" and "Eta:" with the text "21". Below the input fields are three buttons: "Invio", "Cancella", and "Indietro".

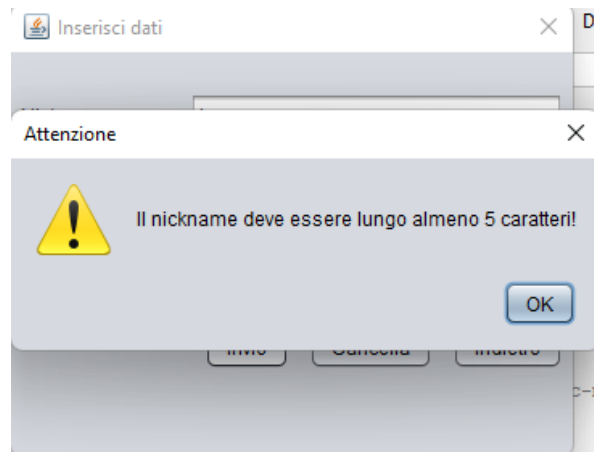
Il sistema chiederà all'utente dei dati personali solamente ai fini del gioco.

E' possibile riscontrare degli errori durante il caricamento dei dati:

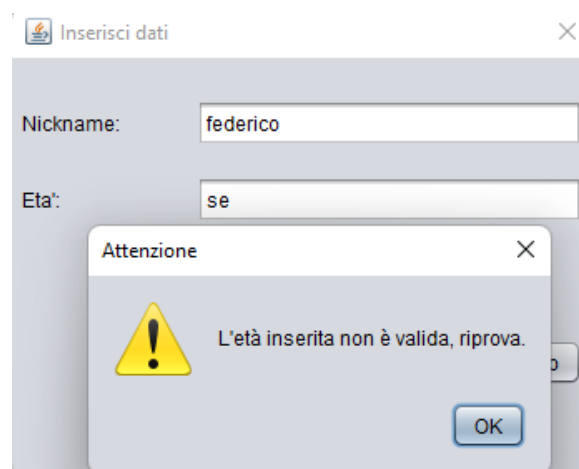
- Inserimento valido:



- Nickname non accettato dal sistema:



- Età non accettata dal sistema



- Ulteriori controlli per:
 - verificare che il nickname non sia già presente all'interno del database
 - verificare che l'utente abbia un'età compresa tra i 10 e i 100 anni

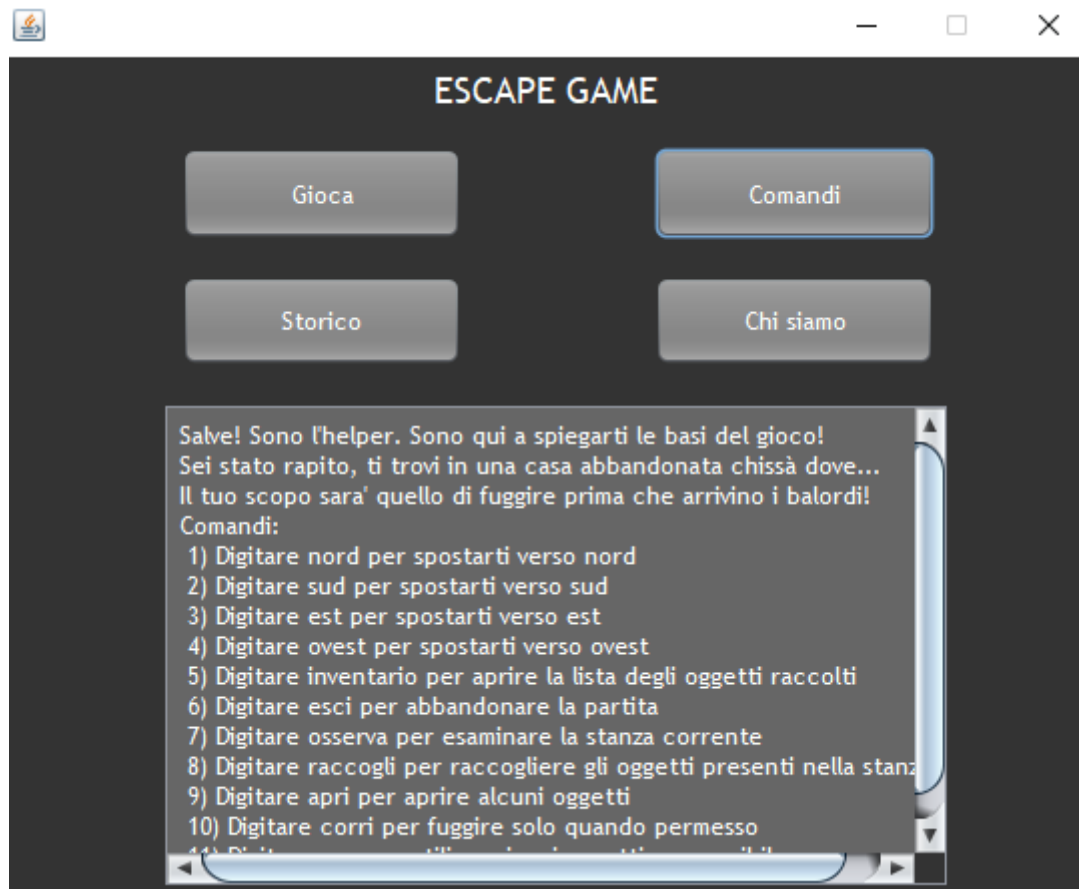
Dopo aver inserito correttamente i dati è possibile iniziare l'avventura e il sistema mostra la seguente schermata di gioco:

```
=====
* Escape Game *
Sei appena stato rapito e ti trovi in una casa abbandonata.
L'entrata è bloccata, per uscirne vivo dovresti trovare un'uscita segreta...
=====
Ingresso

Sei nell'ingresso, sei circondato da polvere e animali morti, si può andare in una sola direzione ... potresti trovare qualcosa qui!
est
```

3.2) Tasto 'Comandi'

All'utente sarà mostrato l'elenco di tutti i comandi accettati dall'avventura



3.3) Tasto 'Storico'

All'utente sarà mostrato l'elenco di tutti gli utenti che hanno giocato



3.4) Tasto 'Chi siamo'

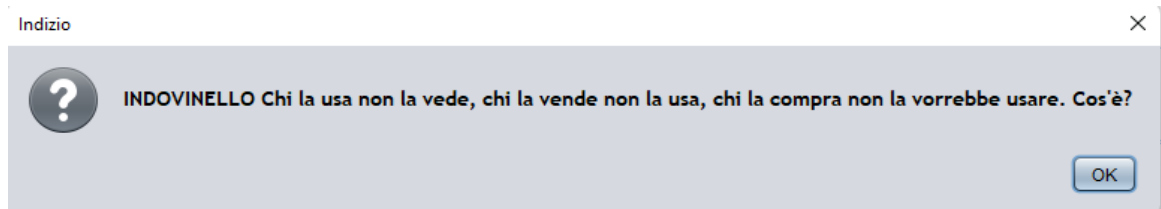
Il sistema fornirà una breve descrizione del team di sviluppo



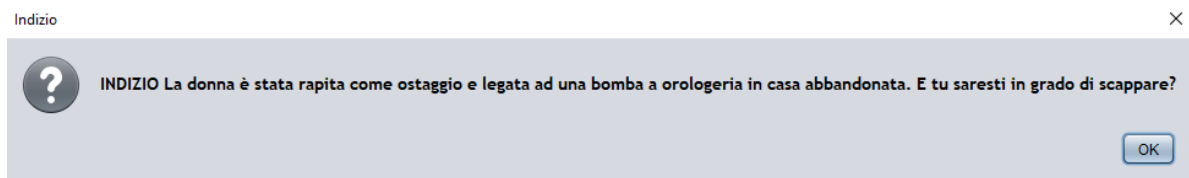
4) Informazioni utili al completamento del gioco

L'applicazione, come già citato al [punto 1.1](#), simula il rapimento di una persona e quest'ultima deve cercare a tutti i costi di scappare e sopravvivere, utilizzando buona parte degli oggetti disponibili all'interno dell'edificio. Per aumentare la difficoltà e rendere il tutto più originale, l'utente dovrà tenere a mente alcune risposte/ipotesi circa i seguenti indizi dislocati in stanze diverse:

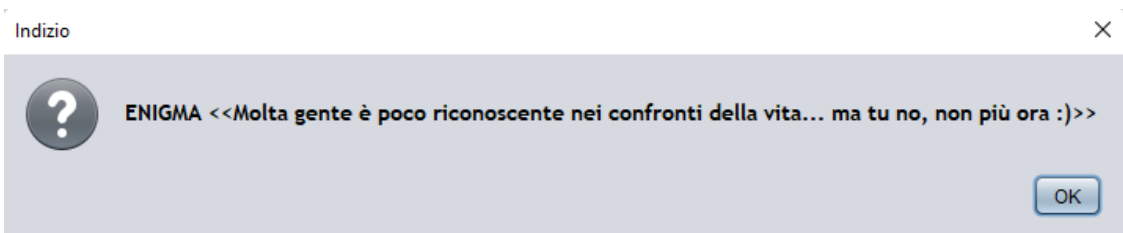
- *Primo indizio:*



- *Secondo indizio:*



- *Terzo indizio:*



Spoiler: Le risposte / ipotesi ai vari indizi non sono necessarie/i ai fini del gioco, cioè il team ha deciso di aiutare l'utente nella sua impresa dando dei piccoli aiuti. La risposta al primo indovinello è molto semplice e, di conseguenza, fa capire all'utente che se non riesce a scappare morirà all'istante. Il secondo indizio allude alla capacità dell'utente di mantenere la calma e cercare tutte le soluzioni per andare via dall'edificio. Infine, il terzo e ultimo indizio, sprona il giocatore a trovare velocemente una via d'uscita.

Inoltre, dopo aver raggiunto la stanza più importante del gioco (guardaroba) e, dopo aver raccolto gli indizi, l'utente ha a disposizione [la mappa](#) per poter uscire dall'edificio e salvarsi.

Però, dal momento in cui l'utente prende la mappa, ha a disposizione solo 30 secondi per scappare e vedere la luce del sole; a complicare la situazione ci sono due vincoli:

- dopo lo studio c'è una stanza segreta visibile solo dopo aver aperto la mappa;
- questa stanza segreta presenta una porta e per aprirla l'utente dovrà possedere nel suo inventario una chiave particolare.

Momento in cui l'utente prende la mappa e inizia il countdown:

```
prendi mappa
Hai raccolto: Ecco la mappa! Ora puoi scappare da questo posto!
OH NO! E' scattato l'allarme di diversi C4.. hai 30 secondi per trovare la via d'uscita e scappare da qui vivo, buona fortuna!
```

Cosa succede se il giocatore non ha trovato la via d'uscita?

```
Tempo scaduto! Si sente la casa tremare, sta per esplodere qualcosa...
BOOM! La casa cade a pezzi, tutto si riduce in rovina poichè numerosi C4 erano stati piazzati nella casa!
Mi dispiace che non hai salutato la tua famiglia in tempo! Adesso li puoi rivedere dal paradiso... addio!
```

Invece, cosa succede se il giocatore ha trovato la via d'uscita?

```
usa chiave
Hai utilizzato: chiave

apri porta
Hai aperto la porta, adesso puoi fuggire da qui!
La porta e' stata aperta, niente ti divide dal mondo esterno. Goditi la pace e il riposo dopo questa lunga fuga!
Sei riuscito a scappare in tempo prima che le numerose bombe all'interno della casa ti facessero saltare la testa :)
.
```