Relazione finale Nathan’s Adventure

Gruppo Amilton
Autori: Valentino Federico, Volpicella Savino Davide, Sequenza Antonio Roberto

Sommario

Introduzione

Questo documento consiste nella relazione tecnica finale del gioco testuale realizzato dal gruppo Amilton per il corso di Metodi Avanzati di Programmazione.

Trama del gioco

Il gioco 'Nathan's Adventure' narra le vicende di un giovane ricercatore universitario di nome Nathan Wilson. Il sogno di Nathan è sempre stato quello di vivere avventure simili a quelle del suo antenato, un famoso esploratore che non si è mai fatto mancare nulla nella vita. Nathan si ritrova però in casa e, immerso nella noia della quotidianità, decide di rovistare tra vecchie foto in cerca di qualche stimolo. Tra le tante foto una in particolare cattura l'attenzione di Nathan, ovvero quella raffigurante il suo antenato John Wilson. Lo sguardo determinato e pieno di vita di John suscita in Nathan una forte voglia di avventura. Il protagonista decide quindi di iniziare la sua avventura e si dirige verso il tempio di Santir, nonché ultimo luogo esplorato dal suo antenato prima della sua scomparsa.

Mappa

Di seguito viene riportata la mappa principale del gioco:

Struttura dei package e delle classi

Tra le scelte progettuali che sono state adottate per la creazione del gioco, quella più importante è stata sicuramente l'applicazione dei principi del paradigma orientato ad oggetti, ovvero information hiding, incapsulamento, polimorfismo ed ereditarietà fra classi. Questo ovviamente per far sì che tutto sia più gestibile e soprattutto per evitare delle ridondanze a livello di codice.
Altra caratteristica importante è la suddivisione in package per far si che ogni gruppo di classi abbia una funzione ben precisa. Di seguito viene mostrata l'organizzazione dei suddetti Package:

Package battle:

- *AttaccoNemico:* contiene i metodi per gestire l'attacco del nemico;
- *Combattimento*: contiene i metodi che permettono di gestire i turni del combattimento. Inoltre viene utilizzato il framework Swing per poter creare l'interfaccia dei combattimenti;
- *Help*: classe in cui viene realizzato il pannello help;
- *Inventario:* classe che crea un pannello in cui viene mostrato l'elenco delle armi raccolte durante il gioco;
- *WarningExit:* pannello che viene visualizzato nel caso in cui un utente vuole chiudere il pannello di combattimento. Qui viene chiesto all'utente se è sicuro di voler terminare l'applicazione.

Package entity:

- *AdvObject:* classe che rappresenta gli oggetti del gioco;
- *Command:* classe che rappresente i comandi del gioco;
- *CommandType:* classe gli enumerativi che rappresentano le tipologie dei comandi;
- *Container:* classe che rappresenta gli oggetti contenitori;
- *MultipleChioceDialog:* classe contente i metodi che gestiscono i dialoghi a scelta multipla del gioco;
- *Room*: classe che rappresenta le stanze del gioco;
- *User:* classe che rappresenta gli utenti del gioco;

Package game:

- *NathansAdventure*: classe che contiene i metodi che consentono l'inizializzazione del gioco, più precisamente vengono inizializzati:

  - Oggetti
  - Stanze
  - Comandi
  - Dialoghi a scelta multipla

  Inoltre questa classe presenta i metodi che implementano le funizionalità dei comandi.

Package gameEngine:

- *Engine*: classe contente il motore del gioco;
- *GameDescription:* classe astratta da cui ogni possibile gioco, che sarà rappresentata da una classe, la estenderà.

Package loginAndRegister:

- *LoginForm*: classe in cui viene realizzata l'interfaccia swing di login;
- *RegisterForm*: classe in cui viene realizzata l'interfaccia swing di registrazione;
- *StartForm*: classe in cui viene realizzato il pannello di avvio del gioco.

Package parser:

- *Parser*: classe che implementa le funzionalità principale del parser;
- *ParserOuput*: classe che implementa tutti i comandi che possono essere accettati.

Package utils:

- *Classifica*: classe che contiene i metodi che permettono di calcolare il punteggio finale del giocatore e l'inserimento di questi in un database;
- *ClassificaForm*: interfaccia swing che realizza la classifica di gioco;
- *HelpForm*: interfaccia swing contentente il pannello help;
- *SavingLoading:* classe che contiene metodi statici che permettono di salvare e caricare la partita.
- *SerializedBattle:* classe che permette di salvare lo stato di un combattimento e di serializzare i suoi attributi;
- *Utils*: classe contentente i metodi che permettono di individuare i token di un comando e l'eliminazione delle stopwords.

Tecnologie utilizzate

Il progetto prevede una serie di argomenti inseriti nel gioco che sono stati spiegati durante il corso:

Programmazione OO

Sono stati applicati i principi della programmazione OO quali: information hiding, incapsulamento ed ereditarietà.

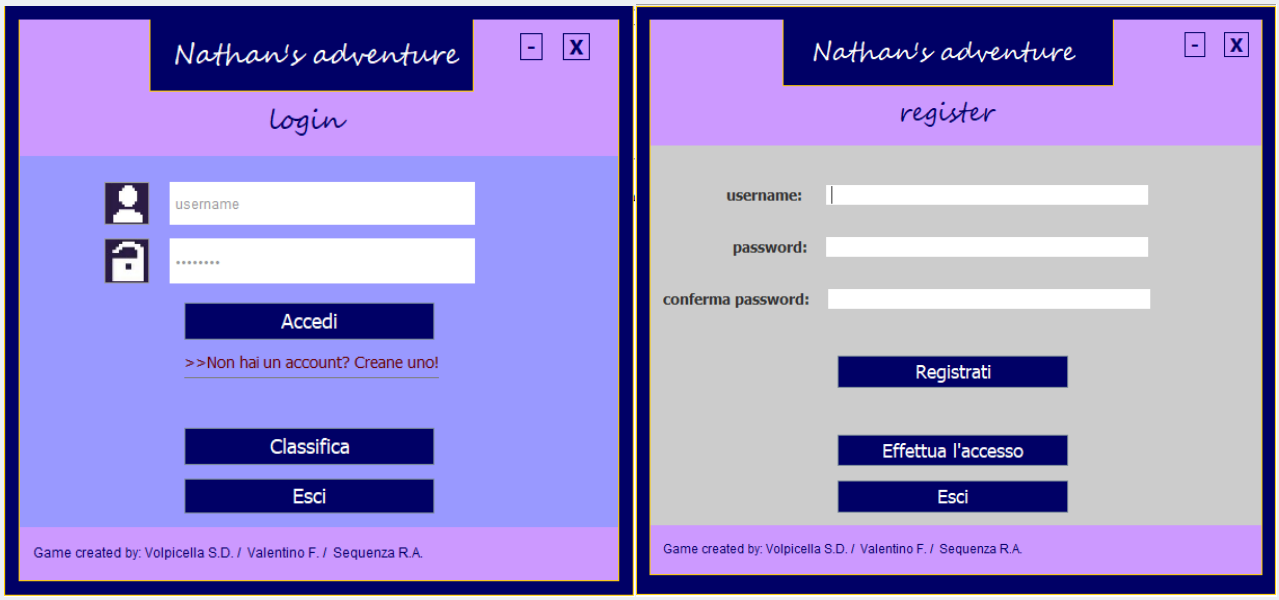
File

Sono stati utilizzati i file per permettere l'inizializzazione degli elementi principali del gioco, quali: stanze, comandi, dialoghi, oggetti e stopwords in riferimento al parser.

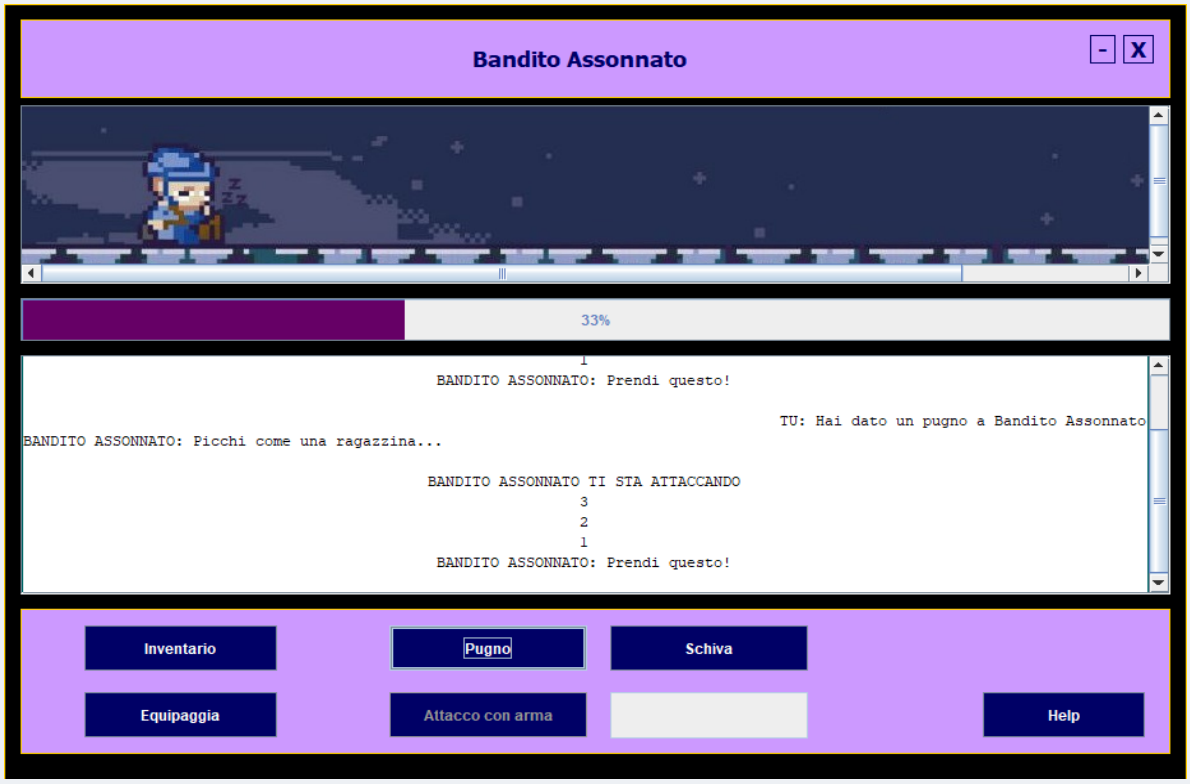
Swing

È stato utilizzato il framework SWING per poter realizzare le schermate di avvio del gioco, quali : login, start e register.

La schermata di registrazione permette a un utente non ancora presente nel sistema di registrarsi.
La schermata del login permette all'utente di effettuare l'accesso.
La schermata di start permette all'utente collegato di iniziare una nuova partita oppure di caricare una partita già esistente.

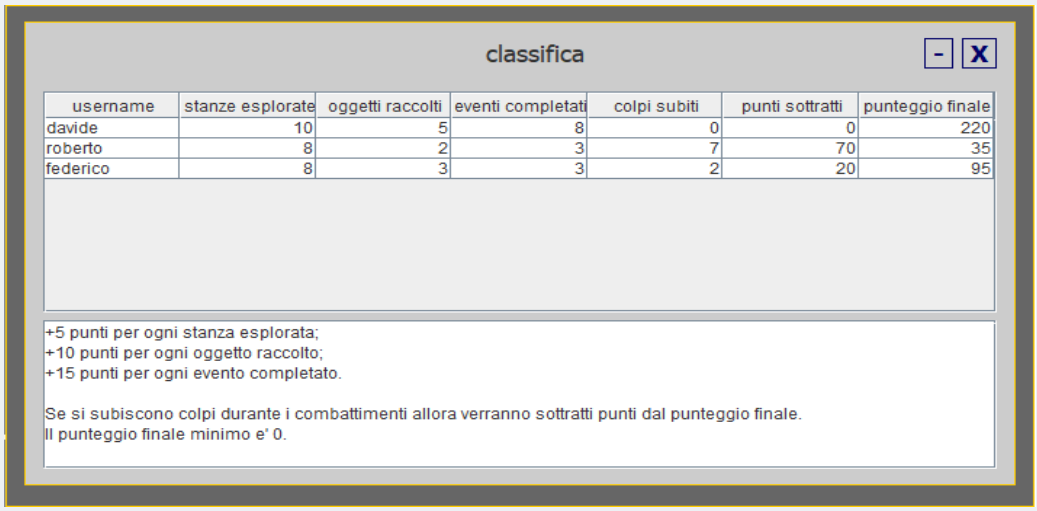


Inoltre l'uso del framework è servito per realizzare i combattimenti del gioco in modo da rendere l'avventura non completamente testuale, inserendo così delle componenti grafiche:



Il combattimento consiste in uno scontro in turni alternati. In ogni combattimento il nemico è il primo ad attaccare. Il giocatore potrà attaccare il nemico con un pugno o con un arma raccolta durante il gioco e schivare gli attacchi del nemico . Ogni nemico ha una propria barra della salute e lo scontro termina quando la vita del nemico si azzera. Infine ogni colpo subito dal nemico andrà a ridurre il punteggio finale del giocatore ottenuto a fine partita.

Infine è stato utilizzato per visualizzare la classifica, la quale mostra i punteggi dei vari giocatori:



Ogni qual volta che un giocatore arriva alla fine del gioco viene inserito nella classifica il suo punteggio. Il punteggio si basa su: oggetti raccolti durante l'avventura, eventi completati, stanze esplorate e colpi subiti. Inoltre il punteggio finale viene diminuito nel caso in cui il giocatore ha subito dei danni durante i combattimenti.

Database

Viene utilizzato un database gestito dal DBMS H2, per permettere le seguenti funzionalità:

- Salvataggio di un utente nel sistema, in particolare di username e password;
- Salvataggio e caricamento della partita di un utente;
- Salvataggio del punteggio ottenuto a fine partita;

Thread

Vengono utilizzati i thread per risolvere una problematica di multitasking riguardante i combattimenti, ovvero stampare un conto alla rovescia prima dell'attacco del nemico e contemporaneamente cercare di schivare l'attacco prima che il suddetto conto termini.

Diagramma delle classi

Di seguito viene presentato il diagramma delle classi della struttura principale del gioco:

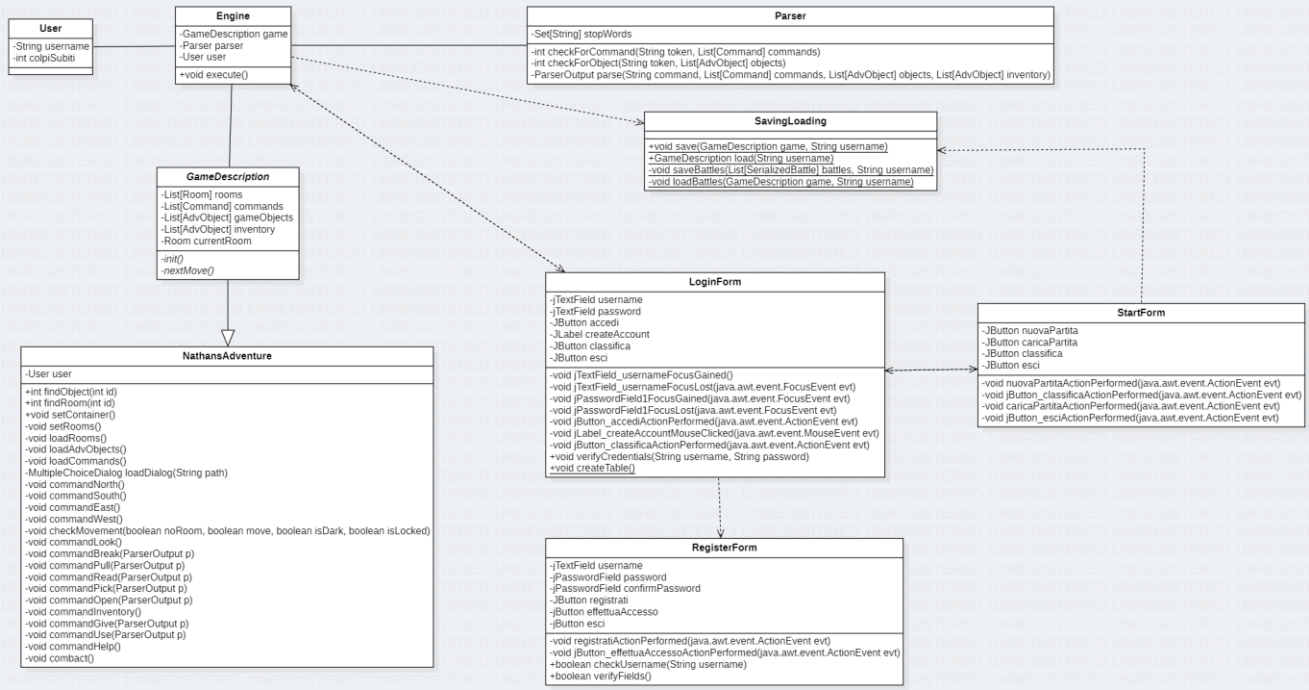
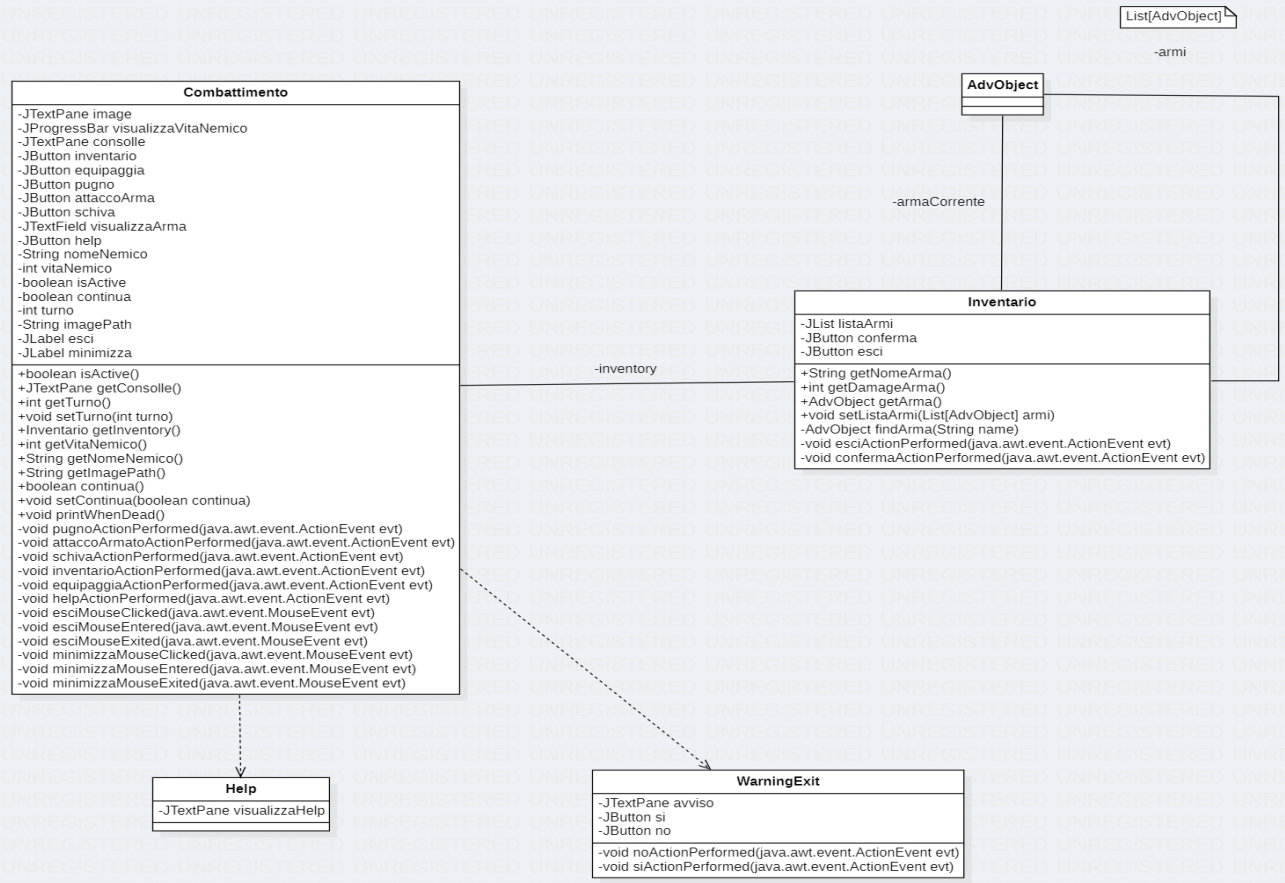


Diagramma delle classi utilizzate per implementare il combattimento del gioco:



Specifica algebrica

E' stata sviluppata la specifica algebrica di Set:

Specifica sintattica:

Sorts: Set, tipoelem, integer, boolean

Operations: new(Set) -> Set
add(Set, tipoelem) -> Set
remove(Set, tipoelem) -> Set
size(Set) -> integer
addAll(Set, Set) -> Set
isEmpty(Set) -> Boolean
contains(Set, tipoelem) -> Boolean
equals(Set, Set) -> boolean

Matrice con costruttori e osservazioni:

Osservazioni	Costruttori	
	new()	add(S, e)
remove(S', o)	S'	If o=e then S else remove (S, o)
size(S')	error	If contains(S, e) then size(S) else size(S) +1
addAll(S', T)	true	add(addAll(S, true), e)
isEmpty(S')	true	false
contains(S', o)	false	If o=e then true else contains (S, o)

equals(Set, Set) -> boolean

Costruttori T'	new()	add(S, e)
new()	true	false
add(true, o)	true	If(e=0 then equals(S, true) else false

Specifica semantica:
Declare S: Set, o, e: tipoelem
remove(new) = S'
remove(add(S, e)) = If o=e then S else remove (S, o)
size(add(S, e) = If contains(S, e) then size(S) else size(S) +1
addAll(new) = true
addAll(add(S, e)) = add(addAll(S, true), e)
isEmpy(new) = true
isEmpty(add(S, e)) = false
contains(new) = false
contains(add(S, e)) = If o=e then true else contains (S, o)

new(new) = true
new(add(S, e)) = false
add(new) = true
add(add(S, e)) = If(e=0 then equals(S, true) else false

Specifica di restrizione:
restrictions:
size(new) = error

Manuale utente

All'apertura della schermata di gioco l'utente potrà:

- Effettuare il login
- Registrarsi
- Uscire dal gioco
- Visualizzare la classifica

Una volta effettuato l'accesso, l'utente potrà svolgere le seguenti azioni:

- Iniziare nuova partita
- Caricare una partita già esistente
- Visualizzare la classifica
- Uscire dal gioco

A partita avviata i comandi validi all'interno del gioco sono i seguenti:

- Nord, per andare a nord
- Sud, per andare a sud

- Est, per andare ad est
- Ovest, per andare ad ovest
- Rompi, per rompere un oggetto valido
- Osserva, per osservare una stanza
- Tira, per tirare un oggetto
- Leggi, per leggere un oggetto
- Prendi, per raccogliere un oggetto
- Apri, per aprire un oggetto
- Inventario, per osservare la lista degli oggetti contenuti nell'inventario
- Dai, per dare un oggetto
- Usa, per usare un oggetto

In particolare, i comandi di sistema dell'applicazione sono i seguenti:

- Help per visualizzare i comandi principali
- Salva, per salvare la partita corrente
- Esci, per uscire dal gioco

Soluzione del gioco

I passaggi per arrivare alla soluzione del gioco sono i seguenti:

- ✓ Dopo il primo dialogo di gioco utilizzare il comando '**rompi le sbarre**' per uscire dalla cella
- ✓ Andare a **ovest** e poi a **nord** fino ad arrivare alla stanza delle celle dei prigionieri
- ✓ Dopo il dialogo con un prigioniero dirigersi a **ovest** nell'alloggio delle guardie e dirigersi nuovamente ad **ovest** nell'archivio
- ✓ Affrontare la guardia nell'archivio e poi dirigersi a **sud** verso l'ingresso della miniera
- ✓ '**Prendi la fiaccola**' nell'ingresso della miniera e usare il comando '**usa la fiaccola**' in modo tale da poter accedere nelle profondità della miniera ad **ovest**
- ✓ Dunque dirigersi nelle profondità della miniera ed affrontare il boss per ottenere la chiave utilizzando il comando '**prendi la chiave**'. La chiave servirà per accedere alle catacombe
- ✓ Dirigersi quindi a **est** ed utilizzare la chiave appena raccolta utilizzando il comando '**usa la chiave**' per aprire la porta. Una volta aperta dirigersi a **est**
- ✓ Affrontare il boss finale, leggere il diario utilizzando il comando '**leggi il diario**' e scegliere se raccogliere o distruggere l'amuleto.

Soluzione per completare il gioco al 100%

I passaggi per arrivare alla soluzione completa del gioco sono i seguenti:

- ✓ Dopo il primo dialogo di gioco utilizzare il comando ‘**rompi le sbarre**’ per uscire dalla cella
- ✓ Andare a **ovest** e poi a **nord** fino ad arrivare alla stanza delle celle dei prigionieri
- ✓ Da qui è possibile procedere a **sud** dove è situata la cucina. Una volta arrivati nella cucina digitare **‘apri la dispensa’** e **‘prendi i croccantini’**
- ✓ Dirigersi ad **ovest** e dai da mangiare ai cani tramite i comandi **‘dai i croccantini’,** adesso sarà possibile aprire il baule con **‘apri il baule’** e prendere il pugnale con **‘prendi il pugnale’**
- ✓ Dirigersi dunque ad **est** e successivamente a **nord** per due volte sino alla stanza dei prigionieri
- ✓ Dopo il dialogo con un prigioniero dirigersi a **ovest** nell’alloggio delle guardie. Qui si hanno due possibilita: la prima consiste nell’aprire il comodino con **‘apri il comodino’** e prendere successivamente i tappi per le orecchie con **‘prendi i tappi per le orecchie’** ed utilizzarli sulla guardia presente nella stanza con **‘usa tappi per le orecchie’**. Da qui sarà possibile tirare la leva con **‘tira la leva’** senza essere scoperti dalla guardia. La seconda possibilità invece consiste nel tirare direttamente la leva con **‘tira la leva’** ed affrontare successivamenete la guardia. Una volta tirata la leva le celle si apriranno ed i prigionieri saranno liberi. Si potrà dunque ritornare ad **est** e leggere la nota lasciata dal prigioniero con **‘leggi la nota’** e prendere il martello con il comando **‘prendi il martello’**.
- ✓ Dirigersi due volte ad **ovest** sino all’archivio ed affrontare il bandito
- ✓ Dirigersi successivamente a **sud** verso l’ingresso della miniera
- ✓ ‘**Prendi la fiaccola**‘ nell’ingresso della miniera e usare il comando ‘**usa la fiaccola**’ in modo tale da poter accedere nelle profondità della miniera ad **ovest**
- ✓ Dunque dirigersi nelle profondità della miniera ed affrontare il boss per ottenere la chiave utilizzando il comando ‘**prendi la chiave**’. La chiave servirà per accedere alle catacombe
- ✓ Dirigersi quindi a **est** ed utilizzare la chiave appena raccolta utilizzando il comando ‘**usa la chiave**’ per aprire la porta. Una volta aperta dirigersi a **est**
- ✓ Affrontare il boss finale, leggere il diario utilizzando il comando ‘**leggi il diario**’ e scegliere se raccogliere o distruggere l’amuleto.