

## DOCUMENTAZIONE DEL GIOCO:

"Murder in the castle" creato dal gruppo **BlackFriendly**, composto da:

- Ramkalawon Alessia
- Rigante Chiara
- Scalzo Andrea

Il gioco in questione è un'avventura testuale appartenente al genere giallo.

Il gioco inizia con una breve introduzione in cui si illustra il suo scopo, ovvero risolvere un omicidio.

Successivamente viene chiesto il nome del giocatore, per far sì che il gioco non resti una parte a se stante rispetto al giocatore.

Inoltre durante il gioco ci sono frasi ironiche sempre per lo stesso motivo.

Il gioco tratta di un omicidio avvenuto all'interno di un castello, in cui ci vive la famiglia De Santis, una famiglia nobilissima. Non si sa né il motivo, né chi, né cosa è successo, il giocatore è chiamato in qualità di detective a risolvere lo strano caso, girando per la casa, osservando ed eventualmente raccogliendo o premendo oggetti.

Essenzialmente, quindi l'utente potrà muoversi all'interno del castello, cercando indizi per far sì che il caso venga risolto.

A livello di programmazione, il gioco è stato suddiviso in diversi package, definiti esattamente come uno strumento per raggruppare tipi in qualche modo legati fra di loro (dove con "tipi" si intendono classi, interfacce, enumerazioni ed annotazioni).

E' stato suddiviso in 4 package:

- Adventure: contiene il caricamento dell'avventura.
- Games: fondamentale per la buona riuscita del gioco in quanto all'interno ci sono le implementazioni dei comandi, delle direzioni, delle stanze e oggetti.
- Parser: è costituito dalle classi di gestione del parser.
- Type: comprende le entità base del gioco.

Di seguito, troviamo la composizione di ciascun package.

### 1. Package Adventure

Suddiviso in 3 classi

- Art: sono implementate alcune illustrazioni di oggetti presenti nell'avventura, utilizzate nella classe MurderInTheCastle.
- Engine: contiene il main e il metodo run che serve a gestire il caricamento dell'entità del gioco. Contiene anche un metodo per chiudere il gioco. Si occupa della gestione delle risposte ai comandi inseriti dall'utente.
- GameDescription: è una classe astratta che contiene i metodi astratti per la gestione dei comandi e le strutture dati che contengono le stanze, i comandi e l'inventario.

## 2. Package Games

Contiene un'unica classe:

- `MurderInTheCastle`: contiene l'implementazione dei comandi e delle direzioni possibili all'utente. E' la classe che implementa la logica di gioco e come interagiscono i comandi dell'utente con il gioco.

Estende la classe `GameDescription` e ne implementa i suoi metodi astratti.

## 3. Package Parser

- `Parser`: si occupa di determinare la struttura e la correttezza dell'input analizzando i comandi e i potenziali oggetti inseriti nella frase eliminando le informazioni superflue.

E' implementato in modo indipendente dalla lingua, in quanto implementato per riconoscere frasi semplici senza l'utilizzo di articoli o preposizioni.

Verifica la correttezza dell'input inserito dall'utente, costituito da uno o più parole, riferendosi ad un comando, oppure comando e oggetto.

- `ParserOutput`: la classe `parserOutput` contiene l'implementazione per gestire una struttura dati che contiene i dati dell'input utente tokenizzati e strutturati. Non contiene altro che l'output della classe `Parser`.

## 4. Package Type

- `AdvObject`: classe contenente gli attributi che costituiscono gli oggetti del gioco: nome, descrizione, alias, indicazioni riguardanti la possibilità di essere raccolto e/o premuto.
- `AdvObject Container`: classe che estende la classe `AdvObject`, in quanto questi oggetti hanno la possibilità di contenere altri oggetti della classe `AdvObject`. Ha come attributi nome, descrizione, alias e possibilità di essere aperto.
- `Command`: ha come attributi il nome, il tipo di comando e un insieme di alias per eseguirlo utilizzando anche dei sinonimi. In particolare, in questa classe sono stati implementati anche i metodi `equals` e `hashCode`.
- `Command Type`: è una classe enumerativa che contiene un numero finito di comandi: rappresentano i comandi che troviamo nel gioco.
- `Rooms`: troviamo gli attributi che costituiscono le stanze del gioco: nome, descrizione, collegamenti possibili con altre stanze, lista di oggetti contenuti all'interno e visibilità della stanza
- `Inventory`: è la classe che contiene la struttura dati di tipo lista, ovvero collezione di oggetti, utilizzata per registrare gli oggetti posseduti nell'inventario.