**SILENT HOUSE**

**MEMBRO:**

**-**Palumbo Samuele, matricola 676988

**DESCRIZIONE GENERALE:**

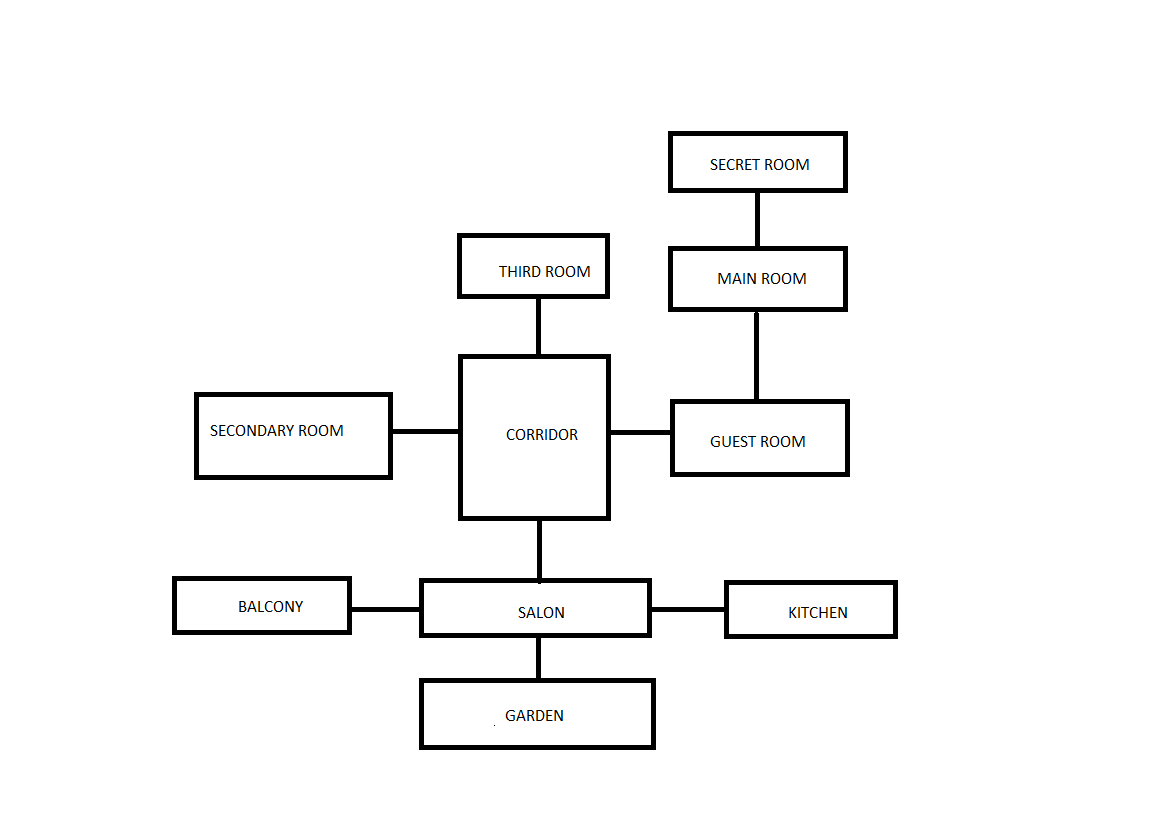
il gioco realizzato è un’avventura testuale che tratta di un detective che viene chiamato per risolvere un caso, per questo giunge in questa casa dove si svolgerà il gioco.

Il detective dovrà addentrarsi nella casa per scoprire il mistero che si cela dietro.

La casa è stata divisa in diverse stanze in cui il giocatore può muoversi liberamente, per poter esplorare le varie stanze e prendere eventuali oggetti.

Per prendere gli oggetti nella mappa si usa il comando prendi, parla per parlare con qualcuno, inserisci per inserire un oggetto, usa per usare un oggetto

SPOILER🡪 Per risolvere l’enigma bisogna inserire calvino per passare avanti oppure usa coltello per aprire lo scrigno

**MAPPA:**

**SALVATAGGIO E CARICAMENTO**

Durante la partita è possibile salvare in qualsiasi momento attraverso l’uso del pulsante salva che si trova nella home di gioco.

È stata creata una classe FileHandler che si occupa di tutte le operazioni di salvataggio e caricamento.

**PRINCIPALI CLASSI**

Il Parser è in grado di riconoscere il seguente tipo di frasi:

• <comando>

• <comando> <articolo/preposizione> <oggetto>

• <comando> <articolo/preposizione> <oggetto inventario>

• <comando> <oggetto>

Il progetto è stato diviso in diversi package per rendere più comodo riconoscere le funzionalità di ogni classe e poterle modificare all’occorrenza.

**COMANDI E PARSER**

I comandi inseriti andranno inseriti alla seconda persona singolare. Quando l’utente digita il comando l’azione verrà processata dal parser che andrà a confrontare il comando con una lista di comandi, se il confronto va a buon fine andrà a controllare se esiste una corrispondenza tra l’oggetto cercato e gli oggetti presenti nella stanza, se non esiste nella stanza verrà controllato l’inventario.

**INTERFACCIA**

Per la parte grafica sono state create due classi, UI e VisibilityManager che utilizzano le java swing per gestire la parte dell’interfaccia.

Abbiamo una finestra in cui sono stati inseriti tutte le componenti che formano l’interfaccia in base a quello che descrivono, mentre la classe VisibilityManager utilizza i componenti istanziati in UI per definire le schermate di gioco, tipo il menu principale, oppure tipo la conferma dell’uscita dal gioco.