



Welcome to
HouseBreak

Che cos'è HouseBreak?

HouseBreak è un'avventura testuale ambientata in una casa, dove il giocatore sarà un **ex agente Federale** rapito durante la spesa al supermarket, perché conosce **informazioni segrete** e potrebbe essere una minaccia per il governo.

Il giocatore dovrà utilizzare **l'astuzia** per aggirare i nemici che incontrerà, ma dovrà stare **attento**, perché sono armati fino ai denti.

Che cos'è HouseBreak?

Il giocatore si risveglierà in una **stanza buia e fredda**, dovrà cercare a tutti i costi di liberarsi e **scappare**.

Buona fortuna!

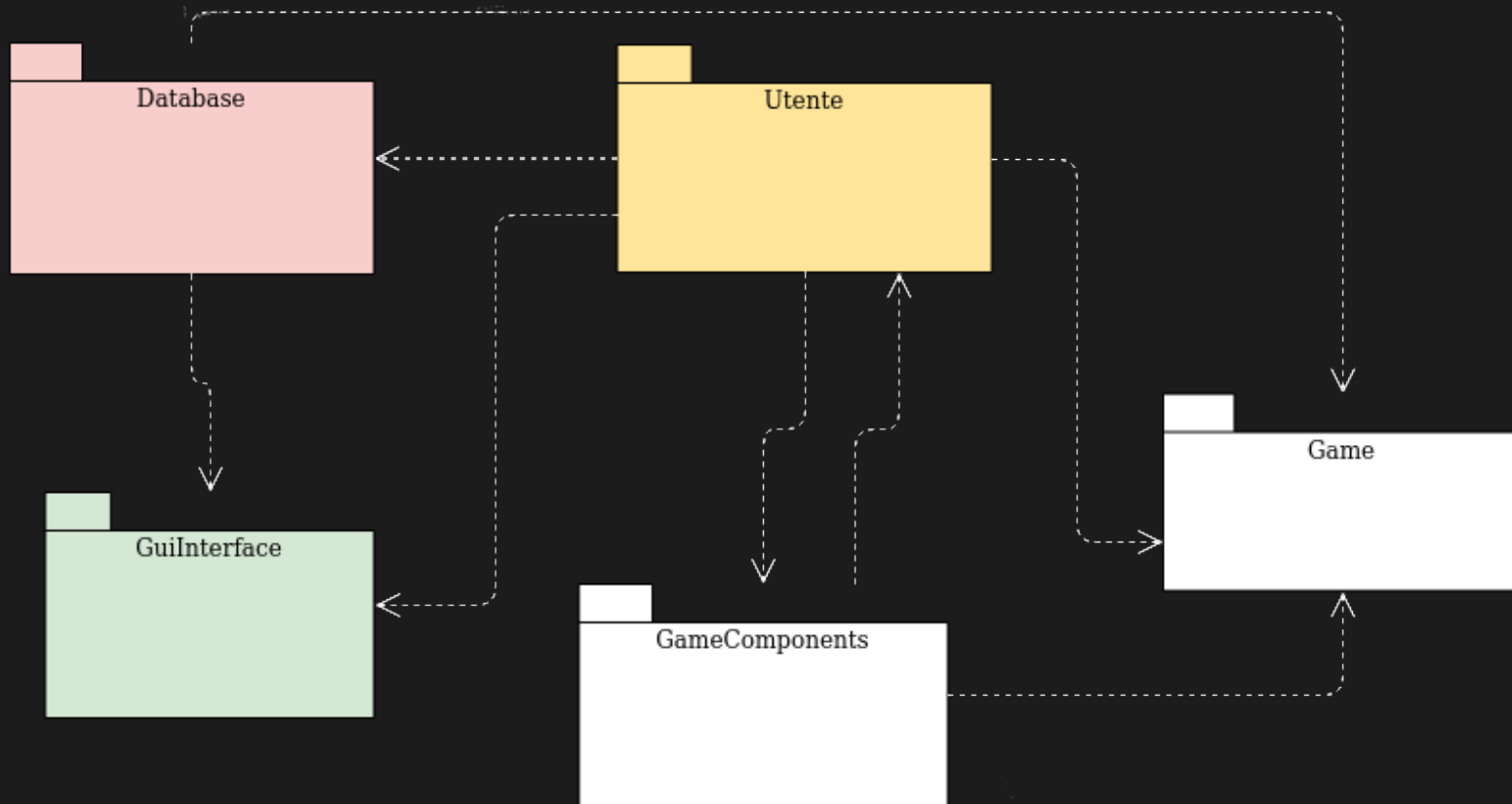
Servizi offerti

Il gioco permette di giocare in **modalità GUEST** senza avere un **account** registrato **online**, ma durante il corso dell'avventura **non potrà salvare** la partita.

Il giocatore che decide di **registrarsi** online potrà usufruire dei **seguenti servizi**:

- *Salvataggi illimitati all'interno del database*
- *Possibilità di giocare online con un amico (prossimo rilascio)*

Package Diagram



Database

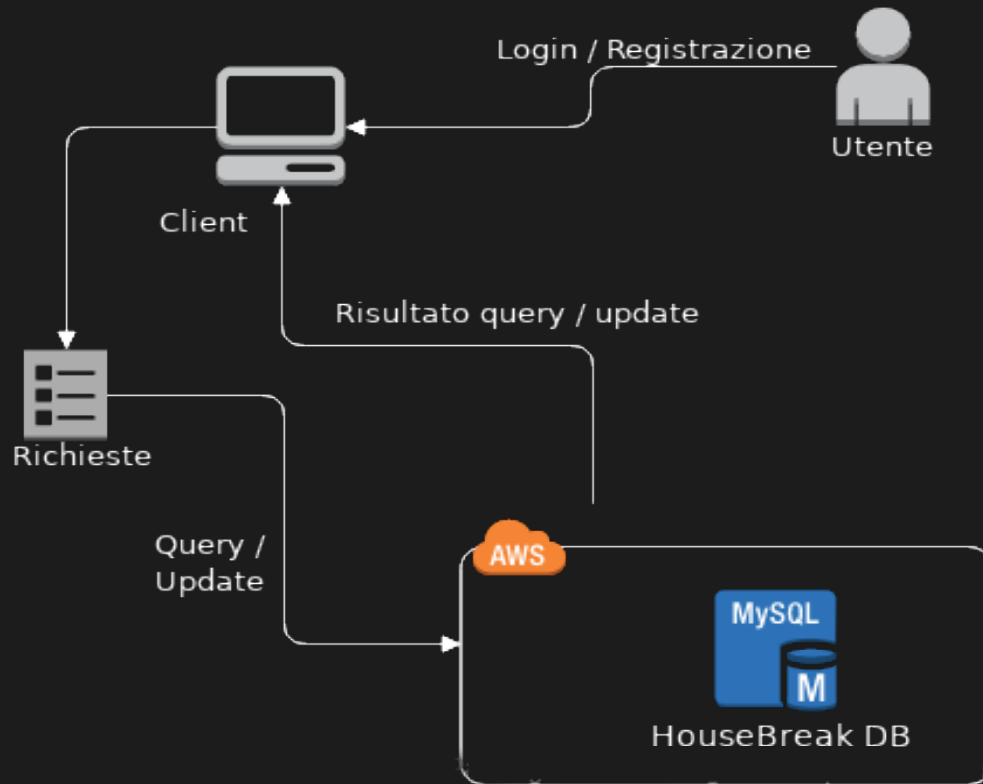
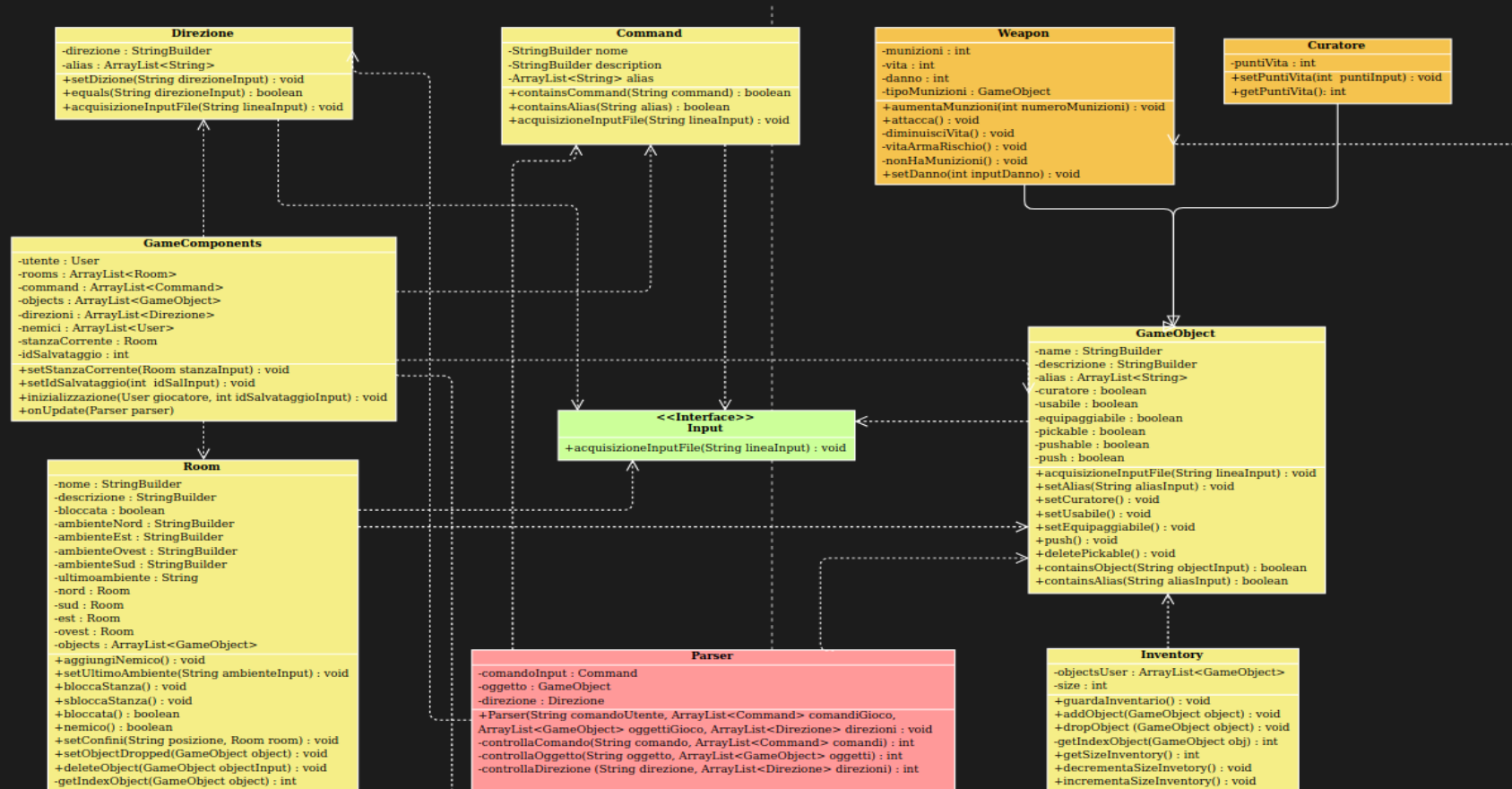
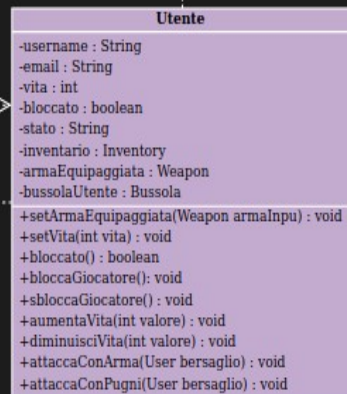
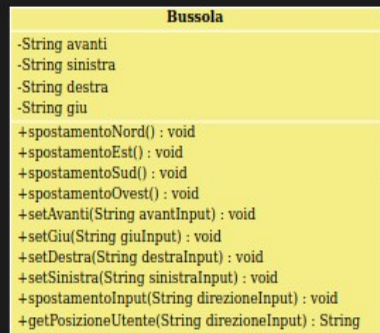


Diagramma delle classi

Gianmarco Moresi





Interfaccia 'Input'

L'interfaccia **Input**, all'interno del progetto permette di effettuare inserimenti di **oggetti, stanza e comandi**, direttamente dai file inseriti all'interno del progetto. Ogni classe, ha una propria implementazione perché la struttura del file, cambia in base alle informazioni richieste.

Esempio: in **oggetti** ci potrebbe essere solo informazioni come **nome, descrizione e alias** all'interno del file.

Al **contrario**, in **stanza**, ci potrebbero essere più informazioni come: **nome stanza, descrizione, ambiente, stanze confinanti, ecc...**

Il **metodo utilizzato**, prende in input la riga prelevata dal file, e viene letta, **inserendo** le informazioni all'interno dei campi interessati, e **tramite dei caratteri** precedentemente inseriti nella riga prelevata, viene identificato **la fine di un determinato campo**, per poi passare al campo successivo.

Specifica algebrica 'Inventario'

Specifica sintattica

Sorts:

- int , GameObject , Inventario, bool

Operazioni:

- **Inventory()** → inventario
- **getObjects(Inventario)** → GameObject
- **guardaInventario(Inventario)** → GameObject
- **addObject(Inventario, oggetto)** → inventario
- **dropObject(Inventario, oggetto)** → inventario
- **containsObject(Inventario ,GameObject)** → bool
- **getIndexObject(GameObject)** → int
- **getSizeInventory(Inventario)** → int
- **decrementaSizeInventory(Inventario)** → inventario
- **incrementaSizeInventory(Inventario)** → inventario

Specifica algebrica 'Inventario'

Specifica semantica

Declare:

- **inv**: Inventario, **oggetto**: GameObject

Operazioni:

- **guardaInventario**(inv) = oggetto
- **guardaInventario**(Inventory()) = null
- **guardaInventario**(addObject(inv, oggetto)) = oggetto
- **addObject**(inv, oggetto) = inv
- **addObject**(addObject(inv, oggetto2), oggetto) = inv2
- **addObject**(dropObject(inv, oggetto), oggetto) = inv
- **dropObject**(inv, oggetto) = inv
- **dropObject**(addObject(inv, oggetto), oggetto) = inv
- **dropObject**(addObject(inv, oggetto2), oggetto) = inv2

Specifica algebrica 'Inventario'

Specifica semantica

Declare:

- **inv**: Inventario, **oggetto**: GameObject

Operazioni:

- **containsObject**(addObject(inv, oggetto), oggetto) = true
- **containsObject**(dropObject(inv, oggetto), oggetto) = false
- **getIndexObject**(addObject(inv, oggetto), oggetto) = int
- **getIndexObject**(dropObject(inv, oggetto), oggetto) = -1
- **getIndexObject**(Inventory(), oggetto) = -1
- **getSizeInventory**(inv) = int
- **getSizeInventory**(addObject(Inventory(), oggetto) = 1
- **getSizeInventory**(dropObject(inv, oggetto) = int

Specifica algebrica 'Inventario'

Specifica di restrizione

Restrictions:

- `dropObject(Inventory(), oggetto) = error`