

# 02 – Elaborazione - Iterazione 2

## 1 Introduzione

Durante questa seconda iterazione ci si concentrerà su:

- Implementazione dello scenario principale di successo del caso d'uso *UC2: Scoprire una carta dallo stock*, in modo da permettere al giocatore di scartare, una alla volta, le carte dal mazzo stock.

## 2 Presentazione del caso d'uso UC2

Andiamo nuovamente a presentare lo schema del caso d'uso *UC2: Scoprire una carta dallo stock*.

### *UC2: Scoprire una carta dallo stock*

<b>Nome del caso d'uso</b>	<b>UC2: <i>Scoprire una carta dallo stock</i></b>
<b>Portata</b>	Applicazione gioco Solitario
<b>Livello</b>	obiettivo utente
<b>Attore primario</b>	Giocatore del Sistema
<b>Parti interessate e Interessi</b>	<ul style="list-style-type: none"><li>▪ <b>Giocatore:</b> vuole giocare al gioco del Solitario, nel modo più fluido possibile. Vuole una visualizzazione chiara degli elementi presenti sul tavolo da gioco.</li></ul>
<b>Precondizioni</b>	Il mazzo deve contenere delle carte da poter scoprire.
<b>Garanzia di successo (post-condizioni)</b>	La carta viene estratta dal mazzo (stock) e visualizzata accanto allo stesso per poter essere poi eventualmente spostata dal giocatore.
<b>Scenario principale di successo</b>	<ol style="list-style-type: none"><li>1. Il giocatore clicca sull'immagine del mazzo (stock).</li><li>2. Il programma dispone sul tavolo, accanto allo stock, la carta estratta dal mazzo.</li></ol> <p><i>I passi 1 e 2 vengono ripetuti finché serve e fino a quando sono presenti carte nello stock.</i></p>
<b>Estensioni (o flussi alternativi)</b>	<p>*a. In qualsiasi momento, l'applicazione termina in modo anomalo:</p> <p>per consentire il ripristino, bisogna garantire che il sistema possa essere ripristinato, a partire da qualsiasi passo dello scenario.</p>

	<ol style="list-style-type: none"> <li>1. Il giocatore riavvia l'applicazione e richiede il ripristino dello stato precedente.</li> <li>2. L'applicazione ricostruisce lo stato precedente. <ol style="list-style-type: none"> <li>2a. L'applicazione rileva delle anomalie che impediscono il ripristino: <ol style="list-style-type: none"> <li>1. L'applicazione segnala un errore al giocatore, registra l'errore, e passa in uno stato pulito.</li> <li>2. Il giocatore inizia una nuova partita.</li> </ol> </li> </ol> </li> </ol> <p>1a – Non sono più presenti carte nel mazzo (stock):</p> <ol style="list-style-type: none"> <li>1. L'applicazione segnala l'errore e rifiuta l'operazione da parte del giocatore, mostrando a quest'ultimo un messaggio a schermo.</li> </ol> <p>2a – Il giocatore ha scoperto tutte le possibili carte nel mazzo (stock):</p> <ol style="list-style-type: none"> <li>1. L'applicazione prevede che l'icona del mazzo coperto venga sostituita con l'immagine di un cerchio verde. Sopra questo cerchio viene mostrato un messaggio in cui si chiede al giocatore se si vuole arrendere, in quanto non può fare più alcuna mossa utilizzando le carte del mazzo. <ol style="list-style-type: none"> <li>a. Se il giocatore clicca nuovamente sul cerchio, l'applicazione provvederà a far iniziare una nuova partita resettando tutte le mosse fatte fino a quel momento dal giocatore.</li> <li>b. Il giocatore può cercare di continuare al gioco utilizzando le carte a disposizione sul tavolo, se quelle nel mazzo non consentono di effettuare nessuna mossa.</li> <li>c. Il giocatore può decidere di uscire dall'applicazione o di iniziare una nuova partita cliccando sul relativo pulsante nella parte superiore della finestra.</li> </ol> </li> </ol>
<b>Requisiti speciali</b>	
<b>Elenco delle varianti tecnologiche e dei dati</b>	

<b>Frequenza di ripetizione</b>	strettamente legata alla quantità di carte presenti nel mazzo.
<b>Varie</b>	

### 3 Analisi Orientata agli Oggetti

Al fine di descrivere il dominio da un punto di vista ad oggetti e gestire ulteriori requisiti, saranno utilizzati nuovamente gli stessi strumenti dell'iterazione precedente (Modello di Dominio, SSD Sequence System Diagram e Contratti delle operazioni). In particolare, i paragrafi seguenti permettono di evidenziare i cambiamenti che tali elaborati hanno subito rispetto alla fase precedente.

#### 3.1 Modello di dominio

Dall'analisi del caso d'uso UC2, non emergono nuove classi concettuali. Di conseguenza, il modello di dominio qui sotto riportato sarà uguale a quello dell'iterazione precedente.

**Player:** Rappresenta l'attore primario, che interagisce col sistema per eseguire le operazioni

**SolitaireGame:** Rappresenta l'applicazione del gioco Solitario

**Deck:** Rappresenta un mazzo di carte francesi costituito da 52 carte

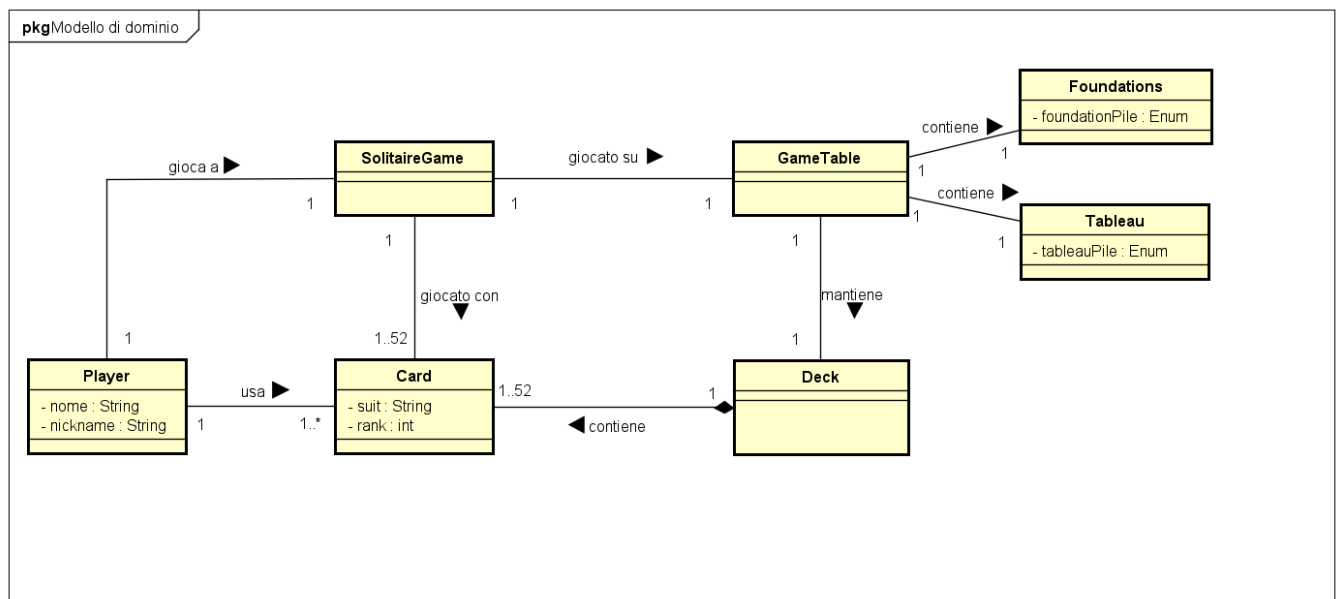
**Card:** Rappresenta il concetto di carta da gioco, caratterizzata da un valore e da un seme

**GameTable:** Rappresenta il luogo in cui viene svolto il gioco

**Foundations:** Rappresenta la zona del tavolo da gioco in cui sono presenti quattro pile di carte inizialmente vuote (basi)

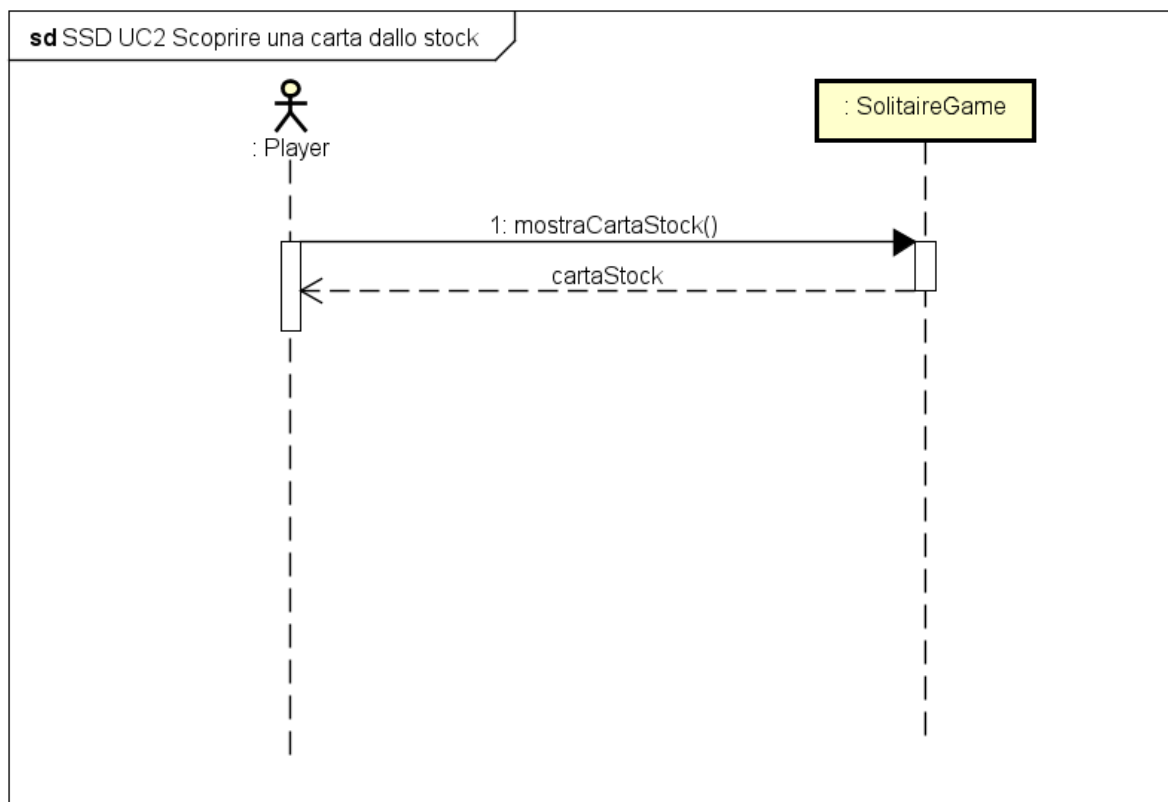
**Tableau:** Rappresenta la zona del tavolo da gioco in cui sono presenti sette colonne di carte. Su queste verranno distribuite le carte in modo che ogni colonna ne abbia un numero uguale alla sua posizione (la prima colonna una, la seconda due, ecc..), per un totale di 28 carte presenti nel Tableau all'inizio del gioco

Tenendo conto di associazioni e attributi tra queste classi il modello di dominio che ne viene fuori è il seguente:



## 3.2 Diagramma di sequenza di sistema

Procediamo ora con il secondo step dell'analisi orientata agli oggetti, con la creazione del diagramma di sequenza di sistema (SSD), al fine di illustrare il corso degli eventi di input e output costituenti il caso d'uso in analisi UC2, e nello specifico come già detto lo scenario principale di successo. Avremo allora:



### 3.3 Contratti delle operazioni

Il prossimo passo è quello della descrizione delle operazioni individuate all'interno del SSD tramite i contratti delle operazioni.

#### Contratto CO1: Mostra Carta Stock

**Operazione:** mostraCartaStock()

**Riferimenti:** Caso d'uso: Scoprire una carta dallo stock

**Pre-condizioni:**

- è presente almeno una carta nel mazzo stock che può essere scartata;

**Post-condizioni:**

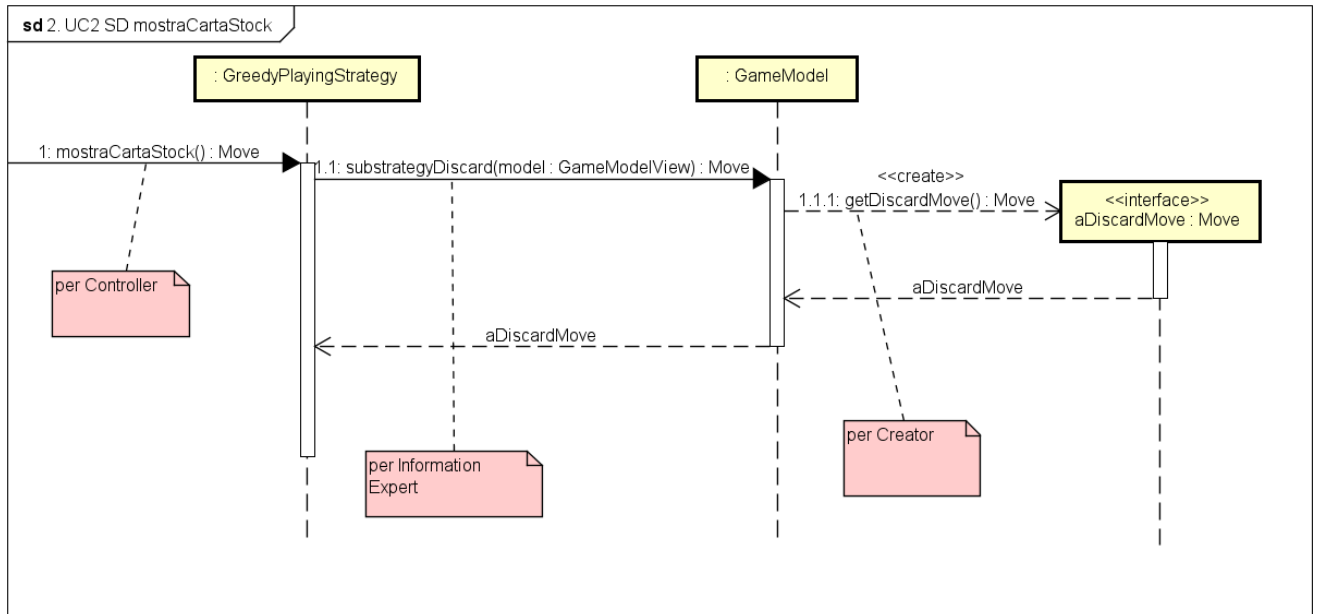
- è stata selezionata un'istanza *c* di Card dal mazzo Deck;
- è stata creata un'istanza *m* dell'interfaccia Move che rimuove la carta selezionata *c* dalle carte presenti in Deck e la inserisce nella pila delle carte scartate (Discard Pile) ;

## 4 Progettazione

La progettazione orientata agli oggetti è la disciplina di UP interessata alla definizione degli oggetti software, delle loro responsabilità e a come questi collaborano per soddisfare i requisiti individuati nei passi precedenti. L'elaborato principale di questa fase che è stato preso in considerazione è il **Modello di Progetto**, ovvero l'insieme dei diagrammi che descrivono la progettazione logica sia da un punto di vista dinamico (Diagrammi di Interazione) che da un punto di vista statico (Diagramma delle Classi). Seguono dunque i diagrammi di Interazione più significativi e il diagramma delle Classi relativi al caso d'uso UC2, determinati a seguito di un attento studio degli elaborati scritti in precedenza.

## 4.1 Diagrammi di sequenza

### ➤ Mostra una Carta dallo Stock



## 4.2 Diagramma delle classi

Al fine di migliorare la leggibilità, il Diagramma delle Classi è riportato anche nella cartella immagini presente in questa iterazione. Il nome dell'immagine è "DCD.png".

