

3 – Elaborazione – Iterazione 2

3.1 Introduzione

Questa seconda iterazione si concentra sull'implementazione del nucleo della logica di gioco: il lancio dei dadi (UC2) e il movimento della pedina (UC3), basandosi sulla configurazione partita realizzata nell'iterazione 1.

3.2 Aggiornamento casi d'uso UC2 e UC3

Riproponiamo la descrizione formalizzata dei casi d'uso UC2 (già presentata nella fase di ideazione, verrà solamente riportata) e UC3 (qui verrà estesa) per riflettere i dettagli emersi durante l'analisi.

3.2.1 UC2: Tirare i dadi

Nome del caso d'uso	UC2: Tirare i dadi
Portata	Applicazione DigiGoose!
Livello	Obiettivo utente
Attore primario	Giocatore
Parti interessate e interessi	Giocatore: Vuole lanciare i dadi per determinare il movimento della sua pedina. Il gioco deve essere equo e seguire le regole.
Precondizioni	Una partita è in corso ed è il turno del giocatore.
Garanzia di successo	Il sistema genera un risultato valido (2-12) del lancio dei dadi e lo visualizza al giocatore.
Scenario principale di successo	<ol style="list-style-type: none">1. Il giocatore seleziona l'azione "Lancia i dadi".2. Il sistema genera due numeri casuali tra 1 e 6.3. Il sistema visualizza il risultato della somma dei due numeri.4. Il sistema entra nel caso d'uso UC3 e viene mossa automaticamente la pedina del giocatore del numero di caselle corrispondente al risultato del lancio dei dadi. Il sistema controlla se la pedina del giocatore si trova nella casella 63, se sì, avvia UC6.5. Il sistema entra nel caso d'uso UC4 e applica l'effetto della casella speciale corrispondente (se speciale).6. Il sistema entra nel caso d'uso UC5 e passa il turno al prossimo giocatore.
Estensioni	<ol style="list-style-type: none">a) L'applicazione non risponde al comando di tirare i dadi:<ol style="list-style-type: none">1. Il giocatore riavvia l'applicazione.
Requisiti speciali	Animazione per simulare il lancio dei dadi.

Elenco delle varianti tecnologiche e dei dati	Dati: Risultato del lancio dei dadi (due numeri interi tra 1 e 6).
Frequenza delle ripetizioni	Ad ogni turno di un giocatore.
Varie	

3.2.2 UC3: Muovere la pedina

Nome del caso d'uso	UC3: Muovere la pedina
Portata	Applicazione DigiGoose!
Livello	Obiettivo utente
Attore primario	Sistema
Parti interessate e interessi	<p>Giocatore: Vuole che la sua pedina venga mossa correttamente secondo il risultato dei dadi e le regole del gioco, e vedere la sua nuova posizione sul tabellone.</p> <p>Sistema: Deve calcolare la posizione corretta, aggiornare lo stato del gioco e gestire le conseguenze del movimento (caselle speciali, vittoria).</p>
Precondizioni	Una partita è in corso ed è il turno del giocatore corrente. I dadi sono stati lanciati (UC2 completato con successo) e il numero di passi da muovere è noto.
Garanzia di successo	La pedina del giocatore corrente è stata spostata alla posizione corretta sul tabellone, in base al risultato dei dadi e alle regole (incluso il rimbalzo dalla casella 63). Lo stato del gioco è aggiornato. Il sistema ha verificato se la casella di destinazione è speciale o la casella di vittoria, e procede di conseguenza (attivando UC4, UC5 o UC6).
Scenario principale di successo	<ol style="list-style-type: none"> 1. Il sistema riceve il numero di passi (risultato del lancio dei dadi da UC2). 2. Il sistema identifica il giocatore corrente e la sua pedina. 3. Il sistema calcola la potenziale nuova posizione sommando la posizione corrente della pedina e il numero di passi. 4. Il sistema controlla se la potenziale nuova posizione supera la casella 63. 5. Se la posizione supera la casella 63, il sistema calcola la posizione di "rimbalzo" (63 meno la differenza tra la potenziale posizione e 63). Altrimenti, la posizione finale è la potenziale nuova posizione. 6. Il sistema sposta la pedina del giocatore alla posizione finale calcolata.

	<ol style="list-style-type: none"> Il sistema aggiorna la posizione della pedina e lo stato del gioco. Il sistema identifica la casella corrispondente alla posizione finale. Il sistema controlla se la casella finale è la casella 63. Se sì, il sistema avvia UC6 (Determinare il vincitore). Se la casella finale non è la casella 63, il sistema controlla se è una casella speciale. Se sì, il sistema avvia UC4 (Gestire le caselle speciali). Se UC6 non è stato avviato e UC4 non è stato avviato (o dopo il completamento di UC4), il sistema avvia UC5 (Gestire i turni) per passare al giocatore successivo.
Estensioni	<ol style="list-style-type: none"> L'applicazione si blocca durante il movimento della pedina <ol style="list-style-type: none"> Il giocatore riavvia l'applicazione.
Requisiti speciali	<ol style="list-style-type: none"> Il sistema deve calcolare la posizione di rimbalzo dalla casella 63 in modo accurato. Il sistema deve identificare correttamente il tipo di casella di destinazione (normale, speciale, vittoria). Il sistema deve attivare i casi d'uso UC4, UC6 o UC5 in base alla casella di destinazione. L'interfaccia utente deve visualizzare l'aggiornamento della posizione della pedina.
Elenco delle varianti tecnologiche e dei dati	<p>Dati: Posizione corrente della pedina, numero di passi dai dadi, struttura del tabellone (caselle e proprietà).</p> <p>Metodi: Calcolo posizione, verifica limiti tabellone, verifica casella speciale/vittoria, aggiornamento posizione pedina, attivazione altri UC.</p>
Frequenza delle ripetizioni	Una volta per turno di gioco, dopo il lancio dei dadi.
Varie	

2.3 Contratti delle operazioni (UC2 e UC3)

Adesso vengono definiti i contratti, per definire le principali operazioni di sistema al fine di gestire quanto visto nell'SSD:

Contratto CO13: tiraDadi

Operazione:	tiraDadi(): int
Riferimenti:	Caso d'uso: UC2: Tirare i dadi
Pre-condizioni:	Una partita è in corso ed è il turno del giocatore corrente. Il giocatore ha selezionato l'azione "Lancia i dadi".
Post-condizioni:	<ul style="list-style-type: none">• Sono stati generati due numeri casuali tra 1 e 6 per i dadi.• La somma dei risultati dei dadi è stata calcolata.• Il sistema è pronto per passare al caso d'uso UC3 (Muovere la pedina) utilizzando la somma ottenuta.• Il risultato (somma) è disponibile per essere visualizzato dall'interfaccia utente.

Contratto CO14: muoviPedina

Operazione:	muoviPedina(passi: int): Casella
Riferimenti:	Caso d'uso: UC3: Muovere la pedina
Pre-condizioni:	Una partita è in corso, è il turno del giocatore specificato, e il numero di passi da muovere è noto (risultato del lancio dei dadi).
Post-condizioni:	<ul style="list-style-type: none">• La nuova posizione della pedina del giocatore corrente è stata calcolata applicando le regole di movimento (incluso il "rimbalzo" dalla casella 63).• La posizione della pedina del giocatore corrente è stata aggiornata sul tabellone.• Lo stato del gioco è stato aggiornato.

- Il sistema ha identificato la casella di destinazione.
- La casella di destinazione (Casella object) è restituita per consentire la gestione degli effetti (UC4, UC6) o il passaggio del turno (UC5).
- L'interfaccia utente è pronta per visualizzare la pedina nella nuova posizione.

2.4 Diagramma di sequenza aggiornato

Il diagramma delle classi generale, che segue l'architettura MVC, rimane quello definito nell'Iterazione 1, poiché le classi e le relazioni necessarie per la logica di UC2 e UC3 (come Dadi, Pedina, Tabellone, Casella e i metodi correlati nei controller e nel modello) erano già state identificate.

Per quanto riguarda il diagramma di sequenza, in aggiunta a quelli che mostrano l'inizializzazione della partita (ereditati dall'Iterazione 1 e anch'essi forniti nel codice UML per completezza), presentiamo il diagramma di sequenza SSD che illustra il flusso principale degli eventi per i casi d'uso UC2 ("Tirare i dadi") e UC3 ("Muovere la pedina"), mostrando l'interazione tra l'utente, l'interfaccia e il sistema per queste funzionalità centrali del gioco.

