“Ingegneria del Software”
2020-2021

Docente: Prof. Angelo Furfaro

Snakes and Ladders

Data	14/07/2021
Documento	Documento Finale – D3

Team Members		
Nome e Cognome	**Matricola**	**E-mail address**
Alessandra Paone	200667	pnalsn00b58i874r@studenti.unical.it

Sommario

List of Challenging/Risky Requirements or Tasks

Challenging Task	Date the task is identified	Date the challenge is resolved	Explanation on how the challenge has been managed
Scegliere un formato appropriato per il salvataggio del file di configurazione di gioco e capire in che modo ripristinare correttamente tutte le variabili di gioco.	25/06/2021	26/06/2021	Poichè le strutture dati utilizzate per il salvataggio delle variabili di gioco sono piuttosto semplici, si è pensato di utilizzare JSON, evitando quindi l'utilizzo di un database. Per ripristinare le informazioni contenute nel file JSON, è stata utilizzata la classe Jackson ObjectMapper, che consente la costruzione di una mappa a partire dall'analisi del file. Inoltre, utilizzando la funzionalità Java Reflection, è possibile automatizzare il processo di valorizzazione degli attributi di interesse in quanto la reflection permette di ottenere informazioni su classi ed interfacce direttamente a run-time.
Dato il gran numero di icone ed elementi grafici utilizzati nell'interfaccia grafica, il problema che si pone riguarda l'ingente costo di istanziazione di tutti gli elementi richiesti al momento dell'invocazione del metodo paintBoard(). Ciò che caratterizza la problematica è la necessità di istanziare questi oggetti "costosi" contemporaneamente poiché questi dovranno essere	29/06/2021	01/07/2021	L'intuizione principale che ha portato alla risoluzione del problema è nata dalla necessità di dover caricare tutte le immagini prima di disegnare la game board. Il pattern utilizzato, Object Pool, risolve questa problematica ottimizzando la performance dell'applicazione e l'utilizzo della memoria riutilizzando gli oggetti salvati in un pool fisso anziché allocarli e deallocarli individualmente.

immediatamente visualizzabili sulla game board.			
Come rappresentare graficamente scale e serpenti tenendo a mente che la griglia di gioco è un JPanel che ha come layout manager un GridBagLayout.	02/07/2021	07/07/2021	Per quanto concerne i serpenti, è stato definito un set di JLabel della stessa dimensione delle celle che rappresentano i pezzi del corpo di un generico serpente. Il problema non risiede nel posizionamento di testa e coda del serpente, ma nel disegno del resto del corpo; si è quindi deciso di utilizzare un algoritmo ricorsivo che tiene conto della posizione di testa e coda ed effettua le opportune scelte sul tipo di icona "connettore" da inserire. Per quanto riguarda, invece, la rappresentazione grafica delle scale, si è deciso di creare un JPanel (laddersPan) avente lo stesso ruolo del pannello su cui è disegnata la game board (boardPan): quando una nuova configurazione viene inizializzata, si disegna all'interno del pannello di conseguenza. Tuttavia, a differenza di boardPan, laddersPan posiziona un pannello più piccolo avente la stessa dimensione della game board, le cui dimensioni sono specificate dall'utente ad ogni nuova configurazione o caricamento di una configurazione salvata precedentemente. Di conseguenza, è eliminato il vincolo di avere una rappresentazione "a celle" come accade per il disegno dei serpenti.

A. Stato dell'Arte

La gran parte di applicazioni esistenti relative al gioco "Scale e Serpenti" sono per lo più web based e forniscono un minimo margine di personalizzazione all'utente per quanto riguarda le impostazioni di gioco. Di seguito sono analizzati gli aspetti principali di uno dei migliori sistemi reperibili sul web.

L'applicazione permette all'utente di scegliere tra le modalità di gioco "human vs human", ideata nel caso in cui ci sia più di un utilizzatore, e "human vs computer", nel caso in cui l'utente sia uno solo. In entrambe le configurazioni, l'utente o gli utenti possono selezionare il numero di giocatori. Nel primo caso, ciascun utente ha la possibilità di scegliere il colore della propria pedina e potrà lanciare manualmente i dadi, mentre l'avanzamento sul tabellone di gioco avviene automaticamente; nel secondo caso, solo il giocatore umano avrà la possibilità di scegliere il colore della propria pedina e lanciare manualmente i dadi.

L’utente o gli utenti non hanno la possibilità di scegliere le dimensioni del tabellone di gioco in quanto è utilizzata solo la configurazione standard (tabellone 10x10), né possono scegliere in che posizione inserire scale e serpenti (la posizione e il numero di scale e serpenti sono unici per ogni partita, indipendentemente dalla modalità di gioco) o il numero di dadi utilizzati (in questo caso si utilizza un unico dado). In entrambi i casi è applicata la regola “Extra Dice”, per cui un giocatore ha la possibilità di lanciare nuovamente i dadi dopo aver ottenuto un sei col tiro precedente.

B. Raffinamento dei Requisiti

I servizi offerti dal sistema

L'applicativo Snakes and Ladders offre i seguenti servizi:

- L'utente può impostare una nuova partita e salvarne la configurazione sovrascrivendo un file esistente o creandone uno nuovo.
- L'utente può caricare e giocare una partita precedentemente impostata e salvata.
- L'utente può scegliere le dimensioni del tabellone di gioco.
- L'utente può posizionare a piacimento scale e serpenti sul tabellone, comunque rispettando i seguenti vincoli:

    - Sia scale che serpenti collegano due caselle poste in righe diverse.
    - Per definizione di scala, la cima della scala deve trovarsi in una casella posta più in alto della casella in cui si trovano i piedi.
    - Per definizione di serpente, la coda del serpente deve trovarsi in una casella posta più in basso della casella in cui si trova la testa.
    - In nessuna casella è possibile che inizino o finiscano contemporaneamente più scale o serpenti.
    - Non è presente la testa di un serpente né la cima di una scala sulla casella finale, così come i piedi di una scala o la coda di un serpente non sono presenti sulla cella iniziale.

- L'utente può selezionare il numero di giocatori da simulare.
- L'utente può selezionare il numero di dadi da utilizzare (uno o due). In relazione a tale scelta, se il numero di dadi è maggiore di uno, allora l'utente può scegliere di selezionare le seguenti regole:

    - Lancio di un solo dado: normalmente il giocatore lancia entrambi i dadi. Tuttavia, se si trova in una delle caselle da N-6 ad N-1 (nel caso generale in cui il percorso sia composto da N caselle), allora tira un solo dado anziché entrambi i dadi.
    - Doppio sei: se un giocatore lanciando i dadi ottiene un doppio sei allora, dopo aver raggiunto la sua casella ed essere stato eventualmente spostato a causa di scale o serpenti, lancia ancora i dadi e si muove di nuovo.

- L'utente può scegliere se posizionare delle caselle speciali sul tabellone. In caso positivo, può scegliere anche la tipologia di celle tra le seguenti:

  - Caselle sosta: se un giocatore arriva su una casella sosta, allora rimarrà fermo un numero di turni che dipende dalla particolare casella sosta. In particolare, una casella "panchina" indica una sosta di 1 turno, mentre una casella "locanda" indica una sosta di 3 turni. In fase di configurazione, l'utente può scegliere se inserire una cella "panchina" o una cella "locanda".
  - Caselle premio: se un giocatore arriva su una casella premio, allora godrà del beneficio della particolare casella premio. In particolare, una casella "dadi" indica di lanciare di dadi e muoversi di nuovo, mentre una casella "molla" indica di avanzare ancora del punteggio ottenuto con l'ultimo lancio di dadi. In fase di configurazione, l'utente può scegliere se inserire una cella "molla" o una cella "dadi".
  - Caselle "pesca una carta": se un giocatore arriva su una casella "pesca una carta", allora deve pescare una carta da un apposito mazzo e comportarsi di conseguenza. In particolare, le carte "panchina", "locanda", "dadi" o "molla" provocano lo stesso effetto delle corrispondenti caselle sosta o premio. Una carta di questo tipo viene immediatamente consumata dal giocatore e rimessa in fondo al mazzo.

- In relazione all'utilizzo delle caselle "pesca una carta", l'utente può scegliere di inserire all'interno del mazzo di carte anche la carta "divieto di sosta" che, diversamente dalle altre, non viene consumata immediatamente dal giocatore, ma messa da parte. Se poi il giocatore arriva su una casella sosta oppure se arriva su una casella "pesca una carta" e pesca una carta sosta, allora può utilizzare la carta "divieto di sosta" precedentemente messa da parte per evitare di effettuare la sosta. Quando la carta viene effettivamente utilizzata dal giocatore, va poi subito rimessa in fondo al mazzo.
- L'utente può selezionare il numero di carte che dovranno essere inserite nel mazzo.
- L'utente può scegliere la modalità di esecuzione della partita selezionando l'opzione "automatica" o l'opzione "manuale".
- Quando lo desidera, l'utente può mettere in pausa la partita.
- Se la modalità manuale è selezionata, l'utente deve poter effettuare il lancio dei dadi al posto dei giocatori simulati.

Altri requisiti funzionali sono relativi alle regole del gioco. In particolare:

- Quando un giocatore arriva su una casella posta "ai piedi" di una scala, allora il giocatore sale la scala e si sposta nella casella "in cima" alla scala.
- Quando un giocatore arriva su una casella in cui è posta la "testa" di un serpente, allora il giocatore scivola sul serpente e si sposta nella casella in cui è collocata l'estremità della "coda" del serpente.

- Le caselle “speciali” descritte nel seguito non sono mai collocate ove iniziano o finiscono scale o serpenti.
- L’ultima casella deve essere raggiunta con un lancio di dadi esatto. Eventuali punti in eccesso porterebbero il giocatore a raggiungere la casella finale per poi retrocedere dei punti residui.

A.1 Servizi (con prioritizzazione)

ID	Nome Servizio	Importanza	Complessità	Descrizione
REQF1	Caricamento configurazione di gioco	alta	media	Deve essere possibile caricare da file system o DBMS una configurazione di gioco precedentemente salvata.
REQF2	Nuova configurazione di gioco	alta	media	Deve essere possibile definire una nuova configurazione di gioco impostando le preferenze in un’opportuna sezione.
REQF3	Salvataggio della configurazione di gioco	alta	bassa	Deve essere possibile salvare su file o DBMS una configurazione di gioco impostata dall’utente.
REQF4	Impostazione del numero di giocatori	alta	bassa	Deve essere possibile specificare nell’apposita sezione “impostazioni” il numero di giocatori da simulare e i loro nomi.
REQF5	Impostazione dimensioni del tabellone	alta	media	Deve essere possibile specificare le dimensioni del tabellone di gioco

				secondo le preferenze dell'utente.
REQF6	Impostazione scale e serpenti	alta	media	Deve essere possibile specificare il numero e la posizione delle scale e dei serpenti che dovranno essere posizionati sul tabellone di gioco.
REQNF1	Visualizzazione di scale e serpenti sul tabellone	alta	alta	Scale e serpenti dovranno essere visualizzati correttamente dall'utente sul tabellone di gioco, in modo tale da rendere sempre visibile il numero della casella e i giocatori posizionati sopra di essa. **Relative problematiche**: - utilizzare una rappresentazione grafica compatibile con il layout a griglia del tabellone di gioco. - utilizzare una rappresentazione grafica compatibile con la composizione ad "s" del tabellone di gioco (le caselle disposte sulle righe dispari sono numerate in ordine crescente da sinistra a destra, mentre le caselle disposta sulle righe pari sono numerate in ordine crescente da destra verso sinistra). - definire un algoritmo per il disegno dei

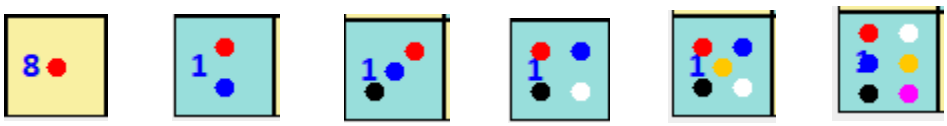
				serpenti sul tabellone di gioco.
REQF7	Impostazione celle speciali	alta	media	Deve essere possibile specificare il tipo e la posizione delle celle speciali da inserire sul tabellone di gioco.
REQNF2	Visualizzazione di celle speciali sul tabellone	alta	bassa	Celle speciali dovranno essere visualizzate correttamente sul tabellone indipendentemente dalla tipologia, in modo tale da rendere sempre visibile il numero della casella e i giocatori posizionati sopra di essa.
REQF8	Impostazione mazzo di carte	alta	media	Se ne è richiesto l'utilizzo, deve essere possibile specificare il numero di carte da inserire nel mazzo.
REQNF3	Visualizzazione della carta estratta sulla finestra di gioco	media	bassa	Ad ogni pescata, la carta estratta dovrà essere correttamente visualizzata sulla finestra di gioco, nello spazio in cui è inserita l'icona del mazzo di carte.
REQF9	Impostazione dadi e regole associate	alta	bassa	Deve essere possibile specificare il numero di dadi da utilizzare durante la partita. L'utente deve avere la possibilità di decidere se e quali varianti di gioco associate utilizzare, se tali varianti sono compatibili con il

				numero di dadi utilizzato.
REQF10	Impostazione modalità di gioco	alta	media	Deve essere possibile selezionare la modalità di gioco desiderata (automatica o manuale) anche durante una partita in corso.
REQNF4	Memorizzazione dei dati in fase di configurazione	alta	bassa	Deve essere possibile memorizzare in una struttura dati le informazioni immesse dall'utente in fase di configurazione nonostante queste non siano ancora considerate persistenti e applicabili.
REQF11	Modalità manuale	alta	bassa	Se la modalità manuale è selezionata, l'utente deve avere la possibilità di effettuare il lancio dei dadi al posto del giocatore corrente.

A.2 Requisiti non Funzionali

- Generalmente, nei giochi da tavolo, si pone un vincolo sul numero massimo di giocatori in una stessa partita. Anche in questo caso, l'utente può inserire un massimo di sei giocatori. Le motivazioni riguardano **l'usabilità** del software: l'utente deve poter vedere come sono posizionati i giocatori sul tabellone di gioco senza che le pedine si sovrappongano l'un l'altra. Nello specifico, a seconda del

numero di giocatori localizzati su una stessa cella, si utilizzano diversi pattern per la visualizzazione della loro posizione.



- Sempre per questioni di usabilità nel contesto della grafica, l'utente può selezionare come massima dimensione del tabellone del gioco una configurazione 15x15. La minima dimensione selezionabile è invece 4x3, determinata dal fatto che, se l'utente sceglie di utilizzare due dadi, un qualsiasi giocatore può arrivare alla casella finale con un unico tiro di dadi.
- L'utente è tenuto a posizionare la testa di un serpente almeno sulla terza riga del tabellone. La motivazione risiede, anche in questo caso, nel contesto della rappresentazione grafica e si vuole garantire un comportamento accettabile anche in circostanze non previste dalle specifiche. Per come sono disegnate le icone "testa" e "coda" del serpente, se si posiziona la testa una riga al di sopra della coda e in seguito l'utente modifica le dimensioni del tabellone di gioco (modificando, indirettamente, anche la posizione del serpente), può verificarsi una situazione in cui non è possibile connettere le due parti in quanto lo spazio di disegno non è sufficiente (più nello specifico, non si deve verificare la situazione in cui la testa e la coda, pur trovandosi su due righe differenti, non si trovano sulla stessa colonna).

Testa del serpente

Coda del serpente

Situazione da evitare

- Se l'utente sceglie di utilizzare le celle speciali "pesca una carta" e quindi il mazzo di carte associato, allora è tenuto a selezionare un minimo di cinque carte, uno per ogni tipo di effetto. La motivazione è legata all'intento di rendere più realistico l'utilizzo del mazzo di carte sacrificando, in parte, la casualità. Al momento della creazione del mazzo, viene garantito l'inserimento di una carta di ciascun tipo (dadi, molla, panchina, locanda e divieto di sosta, se la relativa opzione è selezionata). Se, a questo punto, il numero di carte selezionato dall'utente risulta essere maggiore di 5, allora le ulteriori carte sono inserite in maniera casuale a prescindere dall'effetto.
- Per questioni di consistenza nel funzionamento dell'applicativo, l'utente ha la possibilità di cambiare modalità di gioco (da automatico a manuale e viceversa) e mettere in pausa la partita solo quando permesso, ovvero quando nel riquadro delle

informazioni sulla partita appare il testo “Waiting for the next move…” e i bottoni sottostanti saranno abilitati; se così non fosse, le informazioni relative al turno di gioco corrente sarebbero “troncate”. In alternativa, per evitare tale situazione, si avrebbe dovuto ideare una logica di funzionamento più complessa dell’automa e del relativo gestore, responsabile dei cambiamenti di stato.

- Per rendere l’applicativo il più possibile “user-friendly” anche in fase di configurazione, è importante ridurre al minimo lo sforzo dell’utente. Poiché le finestre di configurazioni sono due, i dati temporanei devono essere inseriti in opportune strutture dati in modo tale da consentire all’utente di passare da una pagina di configurazione all’altra senza perdere le informazioni precedentemente inserite prima di cliccare sul bottone “Apply”.
- Per garantire la correttezza dei parametri inseriti dall’utente in fase di configurazione, è necessario effettuare dei controlli sui valori/testi inseriti in input.

A.3 Scenari d’uso dettagliati

Configurazione di una partita / caricamento di una configurazione di gioco

All’avvio dell’applicazione, l’utente ha la possibilità di scegliere se impostare una nuova configurazione di gioco cliccando sulla voce “New game” nella barra dei menu e poi sulla voce “Configure”. A questo punto, si aprirà una finestra di configurazione attraverso la quale l’utente può impostare i parametri di gioco secondo le proprie preferenze. La finestra di configurazione è scomposta da due finestre. Nella prima finestra è possibile impostare il numero di giocatori da simulare e relativi nomi, il numero di dadi da utilizzare e regole associate ed è infine possibile scegliere se utilizzare o meno il mazzo di carte qualora sia possibile inserire sul tabellone di gioco la casella per pescare una carta dal suddetto mazzo. È inoltre possibile scegliere se includere o meno la carta “divieto di sosta” all’interno del mazzo. La seconda finestra di configurazione riguarda la struttura del tabellone di gioco: è possibile, in questa finestra, impostare le dimensioni del tabellone di gioco, scegliere se inserire o meno celle speciali e selezionarne la tipologia, e, ovviamente, impostare la posizione di serpenti e scale. Le informazioni immesse dall’utente esistono solo “temporaneamente” finché quest’ultimo non clicca sul bottone “Apply”; dopodiché, la finestra di configurazione sarà chiusa e le informazioni immesse saranno utilizzate per poter costruire e disegnare l’intero tabellone di gioco e inizializzare l’interfaccia grafica. Se dopo questa operazione l’utente desidera modificare i parametri immessi, sarà sufficiente cliccare sulla voce “Settings” nella barra dei menu e in seguito sulla voce “Set

options". Una volta iniziata la partita, non sarà più possibile modificare le impostazioni di gioco fino al termine della stessa.
Nota: nonostante le informazioni immesse dall'utente esistano come dati temporanei in fase di configurazione, l'utente può cliccare sui bottoni "Next" e "Back" per passare da una finestra di configurazione all'altra senza che i dati inseriti vadano persi.

Simulazione della partita

Una volta impostati tutti i parametri correttamente, l'utente può avviare la partita secondo la modalità desiderata cliccando sugli appositi bottoni inseriti al di sotto del tabellone di gioco. Tutte le informazioni relative agli eventi che avvengono durante la partita saranno visualizzate sempre sul pannello sottostante il tabellone di gioco, mentre è possibile vedere il nome e la pedina del giocatore corrente sul pannello laterale, ed eventualmente anche la carta pescata grazie all'effetto della relativa cella speciale. Inoltre, quando l'utente lo desidera e se tale azione è permessa dal sistema, è possibile mettere in pausa la partita e riprenderla in un secondo momento. L'interfaccia grafica svolge un ruolo fondamentale all'interno del sistema in quanto, per garantire una buona esperienza utente, è necessario associare alle informazioni testuali le relative animazioni sulla finestra di gioco.

A.4 Excluded Requirements

Dei servizi richiesti nelle specifiche di progetto, nessuno è stato escluso.

Il seguente servizio non è stato fornito: la possibilità di modificare o eliminare la posizione di celle speciali, scale e serpenti in una stessa configurazione di gioco. Questa scelta porta ad una riduzione dell'usabilità del sistema da parte dell'utente; tuttavia, si è preferito affinare l'implementazione dei requisiti richiesti al fine di ottimizzare la robustezza dell'applicazione, gestendo quindi, in modo opportuno, il comportamento del sistema software anche in circostanze non previste dalle specifiche.

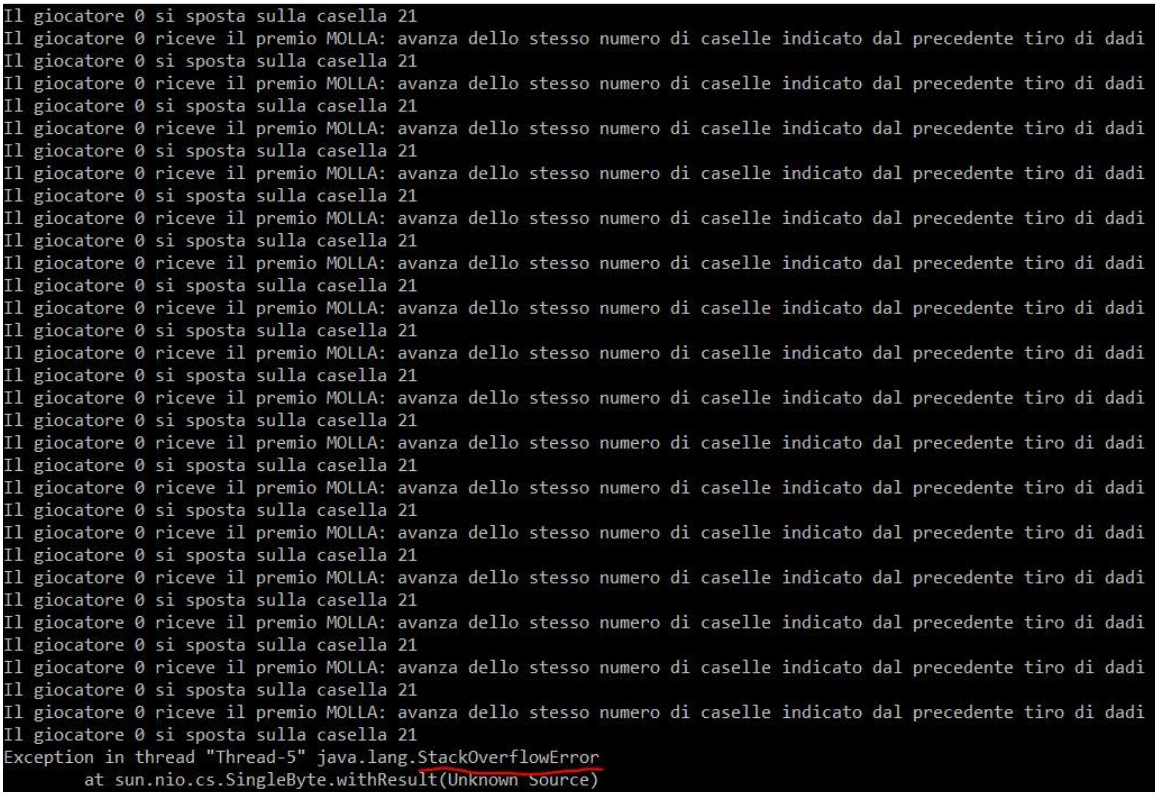
A.5 Assunzioni

Come discusso precedentemente nella sezione A.2 (Requisiti non funzionali), sono state prese importanti decisioni sia nel contesto della configurazione di una partita che nel contesto della simulazione della stessa. Nello specifico sono state poste le seguenti limitazioni per l'impostazione dei parametri di gioco:

- È possibile selezionare fino ad un massimo di sei giocatori.
- Le dimensioni minime del tabellone di gioco sono di 4x3 (righe per colonne), mentre le massime dimensioni sono di 15x15.
- Quando l'opzione per l'utilizzo del deck è selezionata, il minimo numero di carte selezionabili è 5.
- La testa di un serpente deve essere inserita almeno sulla terza riga del tabellone di gioco.

Un'ulteriore decisione presa nel contesto della configurazione di una nuova partita è la seguente: nel caso in cui, durante una partita, si decida di impostare una nuova configurazione di gioco accettando le relative conseguenze (espresse nel warning message), non sarà più possibile la partita interrotta ma occorrerà ricaricare una configurazione precedentemente salvata o crearne una nuova.
Per quanto riguarda il contesto di simulazione di una partita, nella sezione A.2 si è discusso anche il motivo per cui l'utente può mettere in pausa la partita o modificare la modalità di gioco solo quando permesso dal sistema (ovvero quando gli opportuni pulsanti sono abilitati).

Di particolare importanza è la seguente decisione presa nel contesto della logica di gioco: è possibile inserire le caselle premio di tipo "molla" al più 12 caselle (esattamente il massimo valore che si potrebbe ottenere lanciando due dadi) prima della casella finale. La motivazione risiede nel fatto che, se una casella "molla" si dovesse trovare nelle vicinanze della casella finale, allora potrebbe verificarsi la seguente situazione:
si supponga che una casella di tipo "molla" sia posta tre caselle prima di quella finale. Se il giocatore è posizionato sul tabellone sei caselle prima della molla, con un tiro di dadi pari a sei riesce a raggiungerla. A questo punto si attiverà l'effetto "molla", per cui il giocatore avanza ulteriormente di un numero di caselle pari al precedente tiro di dadi. Tuttavia, secondo le regole del gioco, è necessario raggiungere la casella finale con un tiro di dadi esatto; di conseguenza, una volta raggiunta la casella finale, il giocatore dovrà retrocedere di tre caselle, esattamente dove si trova la molla. In questo modo si innescherà un loop infinito per cui il giocatore sarà costretto ad avanzare e retrocedere in continuazione sul tabellone. Questo non sarebbe un problema nella vita reale (ovvero se la partita si giocasse su un tabellone fisico), ma nel caso di un sistema software questo loop porta alla generazione di uno StackOverflowError.

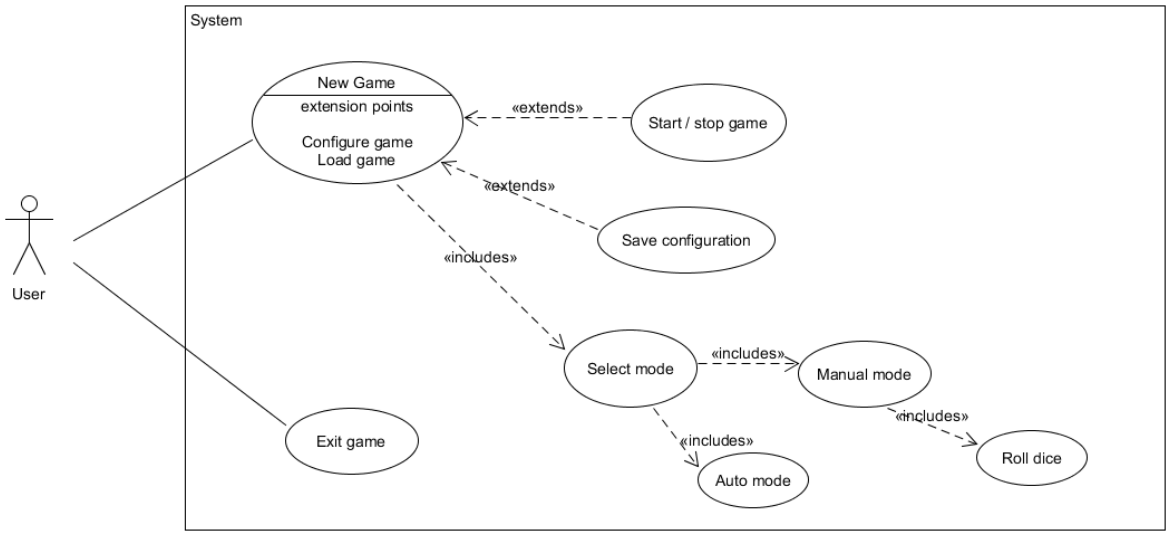


```
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Il giocatore 0 riceve il premio MOLLA: avanza dello stesso numero di caselle indicato dal precedente tiro di dadi
Il giocatore 0 si sposta sulla casella 21
Exception in thread "Thread-5" java.lang.StackOverflowError
        at sun.nio.cs.SingleByte.withResult(Unknown Source)
```

StackOverflowError causato dal "bug molla"

A.6 Use Case Diagrams

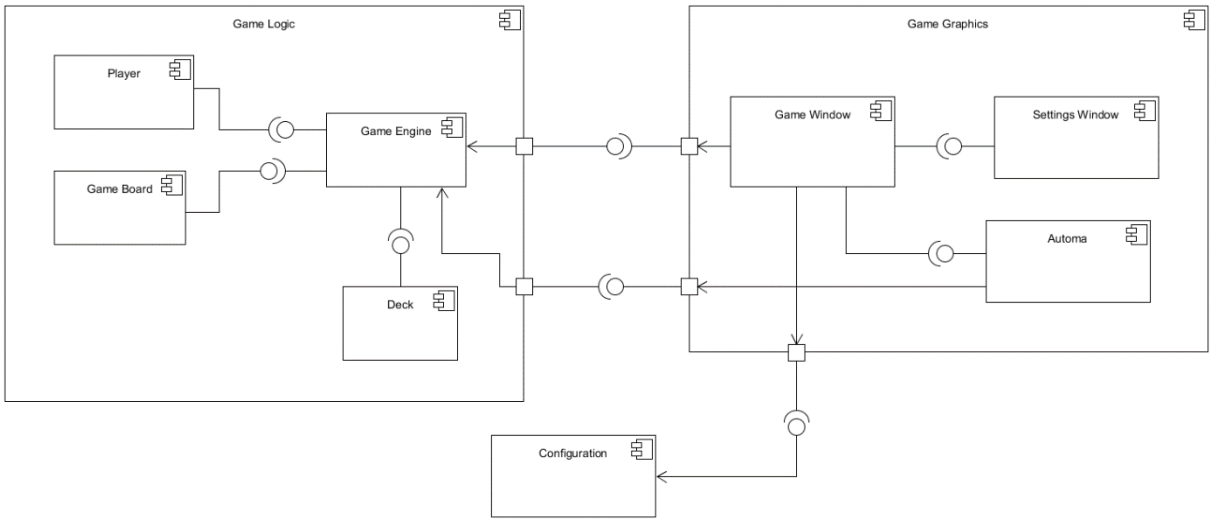
L'unico caso d'uso significativo nel sistema software "Snakes and Ladders" è la simulazione di una partita. L'utente è l'osservatore di una simulazione, il quale imposta la configurazione di gioco desiderata o carica da file una configurazione precedentemente salvata e rimane ad osservare il sistema che presenta una traccia delle attività durante i turni di gioco simulati, fino a quando uno dei giocatori non raggiunge la casella finale terminando la partita. Inoltre, subito prima di iniziare la partita e durante la simulazione, l'utente ha la possibilità di cambiare a piacimento la modalità di gioco (auto/manual mode) e/o mettere in pausa la partita.


System
New Game
extension points
Configure game
Load game
«extends»
Start / stop game
«extends»
Save configuration
«includes»
User
Select mode
«includes»
Manual mode
«includes»
Roll dice
«includes»
Auto mode
Exit game

C. Architettura Software

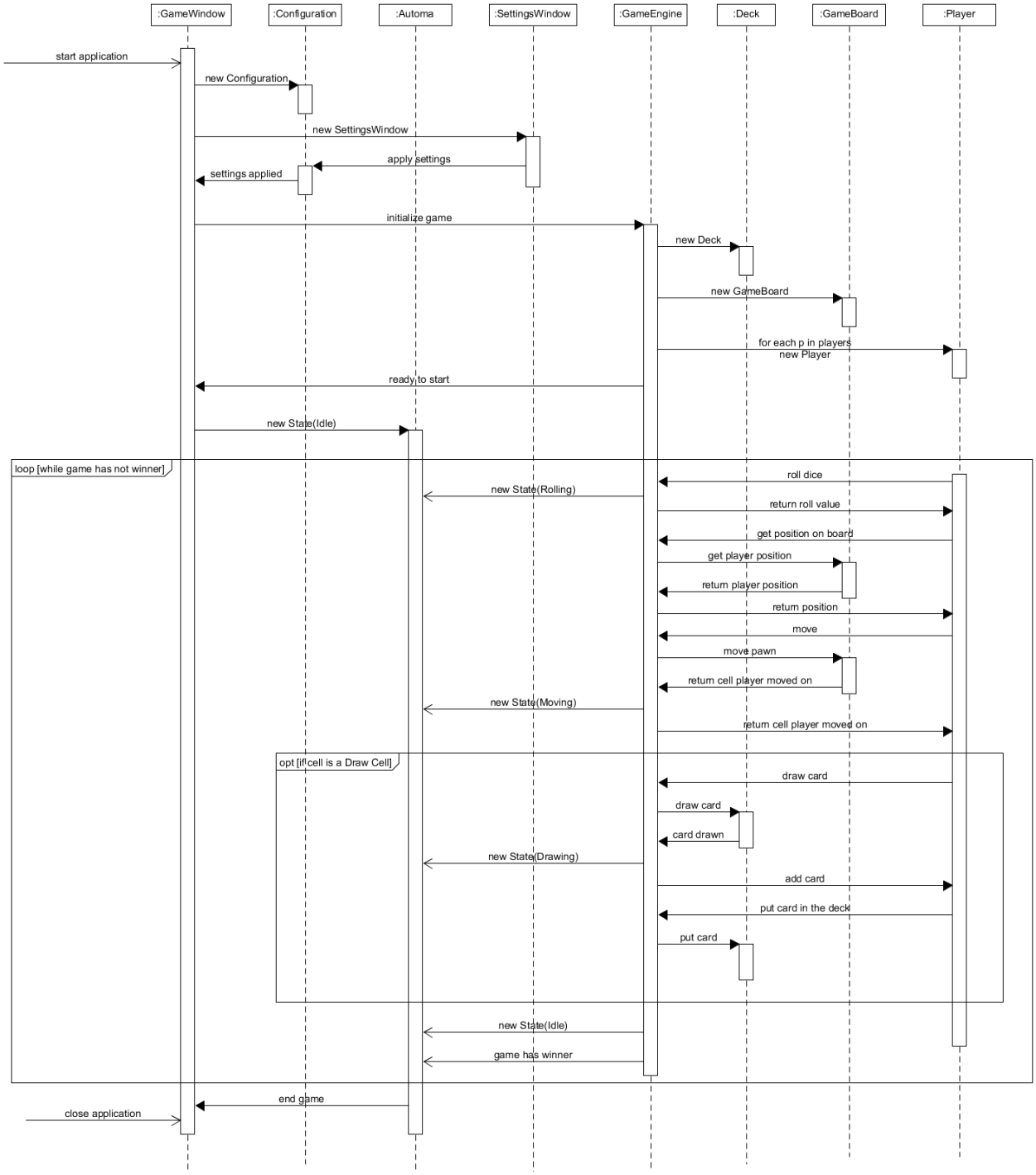
Di seguito è rappresentato il comportamento del sistema software dal punto di vista statico mediante Component Diagram e dal punto di vista dinamico mediante il relativo Sequence diagram. Si noti che il comportamento presentato è quello relativo allo scenario in cui l'utente sceglie di posizionare, come variante di gioco, delle celle speciali sul tabellone di gioco e in particolare la cella speciale che permette di pescare carte dal deck. Di conseguenza, nel Sequence diagram sarà mostrato il caso specifico in cui, durante il turno di gioco, il player si posiziona sulla cella sopracitata e attiverà l'effetto della stessa.

C.1 The static view of the system: Component Diagram



È possibile individuare, all'interno del sistema, due macro-componenti principali: la componente logica e la componente grafica del gioco. Nella componente logica, il ruolo principale è ricoperto dal Game Engine, che inoltra i servizi richiesti dal componente Player ai componenti Game Board e Deck, agendo da intermediario. Il collegamento tra i due macro-componenti è realizzato nel momento in cui la Game Window, componente principale della grafica del gioco, richiede al Game Engine di inizializzare il gioco stesso secondo i parametri specificati dall'utente, forniti dal componente Configuration. Un altro componente di rilievo risulta essere anche l'Automa, che fornisce servizi sia al Game Engine che alla Game Window al fine di fornire all'utente informazioni in tempo reale sull'andamento della partita.

C.2 The dynamic view of the software architecture: Sequence Diagram



D. Dati e loro modellazione (se il sistema si interfaccia con un DBMS)

Il sistema software implementato non si interfaccia con un DBMS. Gli unici dati utilizzati sono quelli forniti dall'utente al momento dell'impostazione di una configurazione di gioco. Poiché tali dati possono essere facilmente salvati in strutture dati elementari, non si è ritenuto necessario ricorrere all'utilizzo di un DBMS.

E. Scelte Progettuali (Design Decisions)

Utilizzo dell'automa per scandire le fasi di gioco

L'automa a stati finiti, utilizzato per scandire le fasi di gioco di ciascun turno, rende più dinamica e coinvolgente l'interfaccia grafica, contribuendo anche a garantire una buona User Experience. Grazie all'automa è possibile apprezzare graficamente il risultato di tutte le parti che compongono il generico turno (lancio dei dadi, spostamento, accesso al mazzo di carte); altrimenti, ciò che verrebbe mostrato di ciascun turno si limiterebbe al risultato finale ottenuto dopo che tutte le parti hanno contribuito. Si consideri, come esempio, un turno in cui un giocatore finisce, dopo un lancio di dadi, su una casella che contiene la testa di un serpente: l'automa permette di osservare il risultato del lancio dei dadi, la pedina del giocatore spostarsi dalla cella di partenza sulla testa del serpente e, infine, spostarsi sulla cella della coda; senza di esso, si potrebbe osservare la pedina del giocatore spostarsi dalla cella di partenza sulla coda del serpente, rischiando di generare confusione nello spettatore. Maggiori dettagli sull'implementazione sono forniti nella sezione F (progettazione di basso livello – parte grafica).

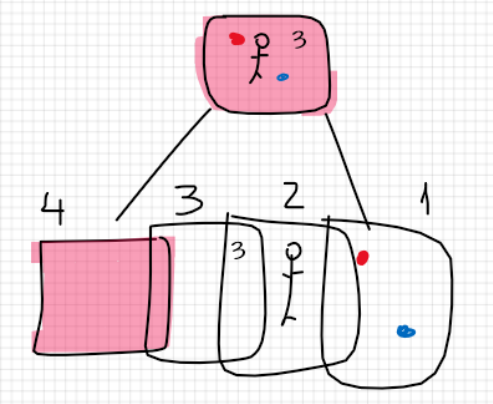
Separazione della parte logica dalla parte grafica

Al fine di rendere il sistema riutilizzabile, è vantaggioso ideare parte logica e parte fisica del gioco come due realtà distinte, come si può osservare nella sezione C.1 (Architettura del software). In questo modo non si è vincolati ad utilizzare lo stesso tipo di interfaccia grafica, ma sarà possibile, eventualmente, modificarla o ridefinirla completamente in futuro.

La logica "a strati" della game board

Ogni casella della game board è progettata come un pannello di dimensioni 50x50 pixels a cui, eventualmente, possono essere aggiunti degli strati. In particolare, ogni cella possiede di base un pannello cui è impostato il colore di background. Un secondo strato è aggiunto appositamente per inserire la label del numero della casella. A questo punto, se la cella contiene anche icone (è il caso di celle speciali o celle contenenti parti di scale o serpenti), un ulteriore pannello sarà aggiunto per disegnarle. Infine, l'ultimo pannello situato al di sopra di tutti gli altri, sarà aggiunto al fine di disegnare i giocatori posizionati

sull'opportuna casella. Di seguito è inserito un esempio illustrativo sull'ordine in cui devono essere posizionati i pannelli sopracitati.



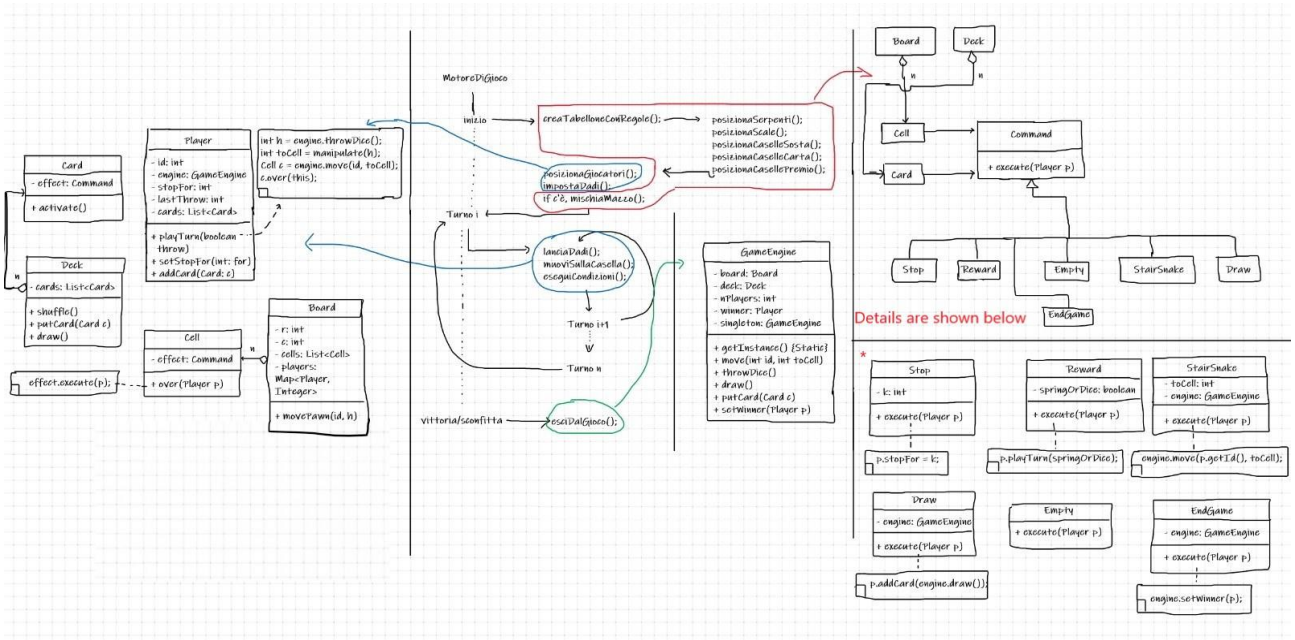
Progettazione della UX (User Experience)

La User Experience riguarda gli aspetti esperienziali e affettivi legati al possesso di un prodotto o all'interazione con esso, le percezioni personali riguardo l'utilità, la semplicità d'utilizzo, l'efficienza e l'estetica del sistema. Questa è una dimensione della progettazione che pone al centro le caratteristiche e i bisogni degli utenti e si focalizza sul loro contesto d'uso. L'intera fase di design della grafica è stata attuata ponendo gran parte dell'attenzione su questo aspetto e lavorando di conseguenza. Effetti grafici e sonori conformi alla tipologia di gioco ideata principalmente per bambini, nonché una corretta visualizzazione di tutti gli elementi principali, soprattutto scale e serpenti, sono fondamentali per garantire all'utente una discreta user experience.

Salvataggio dei dati in formato JSON

Poiché le strutture dati utilizzate per memorizzare le variabili di gioco sono piuttosto semplici, si è scelto di salvare tali dati su file in formato JSON, evitando quindi l'utilizzo di un database. Salvare i parametri di una data configurazione di gioco in formato JSON rende, inoltre, le operazioni di ripristino più semplici da gestire. Per ripristinare le informazioni contenute nel file JSON, è stata utilizzata la classe Jackson ObjectMapper, che consente la costruzione di una mappa a partire dall'analisi del file. Inoltre, utilizzando la funzionalità Java Reflection, è stato possibile automatizzare il processo di valorizzazione degli attributi di interesse in quanto la reflection permette di ottenere informazioni su classi ed interfacce direttamente a run-time.

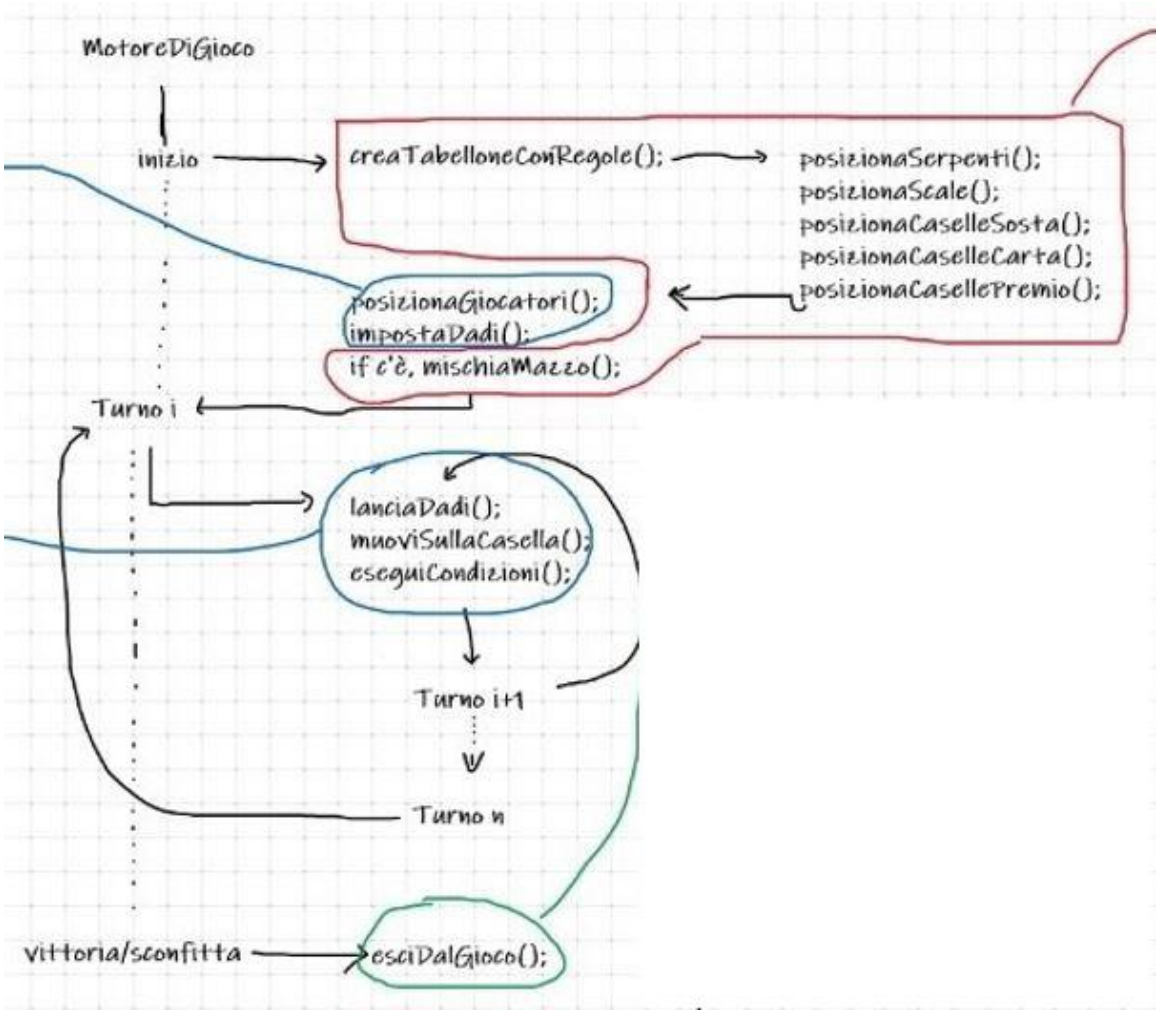
F. Progettazione di Basso Livello



Bozza della logica di gioco e delle relazioni tra le varie classi

Logica di base del gioco

L'intera logica di base di "Snakes and Ladders" è implementata nel componente che rappresenta il motore di gioco (gameEngine). Nel sistema software presentato, il game engine svolge il ruolo di mediatore tra i principali componenti del sistema stesso, al fine di garantirne il controllo e la coordinazione. In particolare, il game engine ha come suoi *colleagues* rispettivamente i componenti Player e Board e Player e Deck. Tale scelta è stata effettuata in quanto il comportamento del sistema è basato sulle modalità di interazione dei vari componenti; tuttavia, nonostante essi comunichino in modo ben definito, la proliferazione di interconnessioni potrebbe inficiare sulla riusabilità del sistema. Per tale motivo, si è optato per l'utilizzo di un Mediator che incapsulasse le modalità di interazioni tra gli oggetti di interesse favorendo, di conseguenza, un basso accoppiamento tra quest'ultimi. Inoltre, poiché è richiesta un'unica istanza del game engine e deve essere fornito un punto di accesso globale a tale istanza, esso è stato implementato come Singleton.



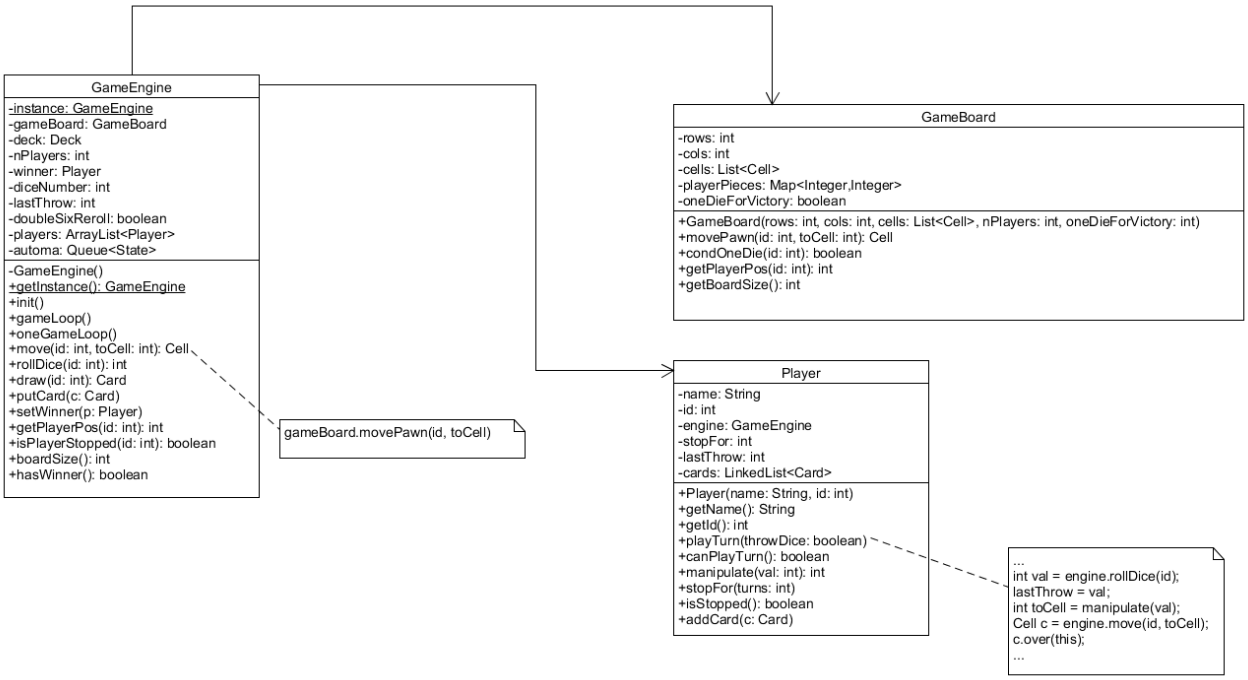
Interazione con Player e Board

Il turno di gioco di un Player include le seguenti fasi:

- Lancio dei dadi
- Calcolo della cella su cui occorre posizionarsi
- Effettivo spostamento sulla cella
- Applicazione di eventuali effetti (ad esempio applicazione dell'effetto della cella speciale o spostamento su un'altra cella nel caso di scale e serpenti)

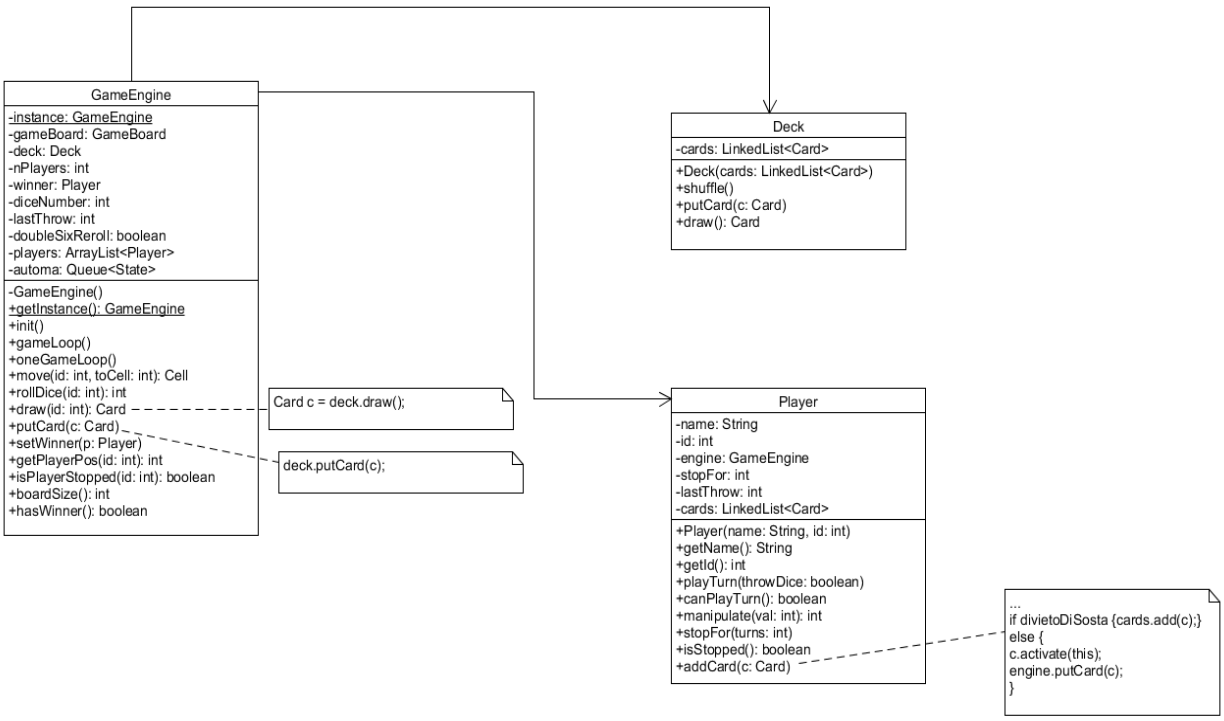
Ogniqualvolta un giocatore desideri effettuare il proprio turno, deve richiedere al Game Engine di gestire il lancio dei dadi e lo spostamento sull'apposita cella, in quanto il Player non comunica direttamente con la Board e quindi non sa in che modo sono posizionate le celle né quali siano i loro effetti. Il Game Engine, che invece ha un riferimento alla Board e conosce la sua composizione, avrà la responsabilità di richiedere a quest'ultima di spostare la pedina del giocatore sul tabellone, recuperare le informazioni necessarie sullo

spostamento effettuato e fornirle al Player, che attiverà eventualmente l'effetto della cella su cui è posizionato.



Interazione con Player e Deck

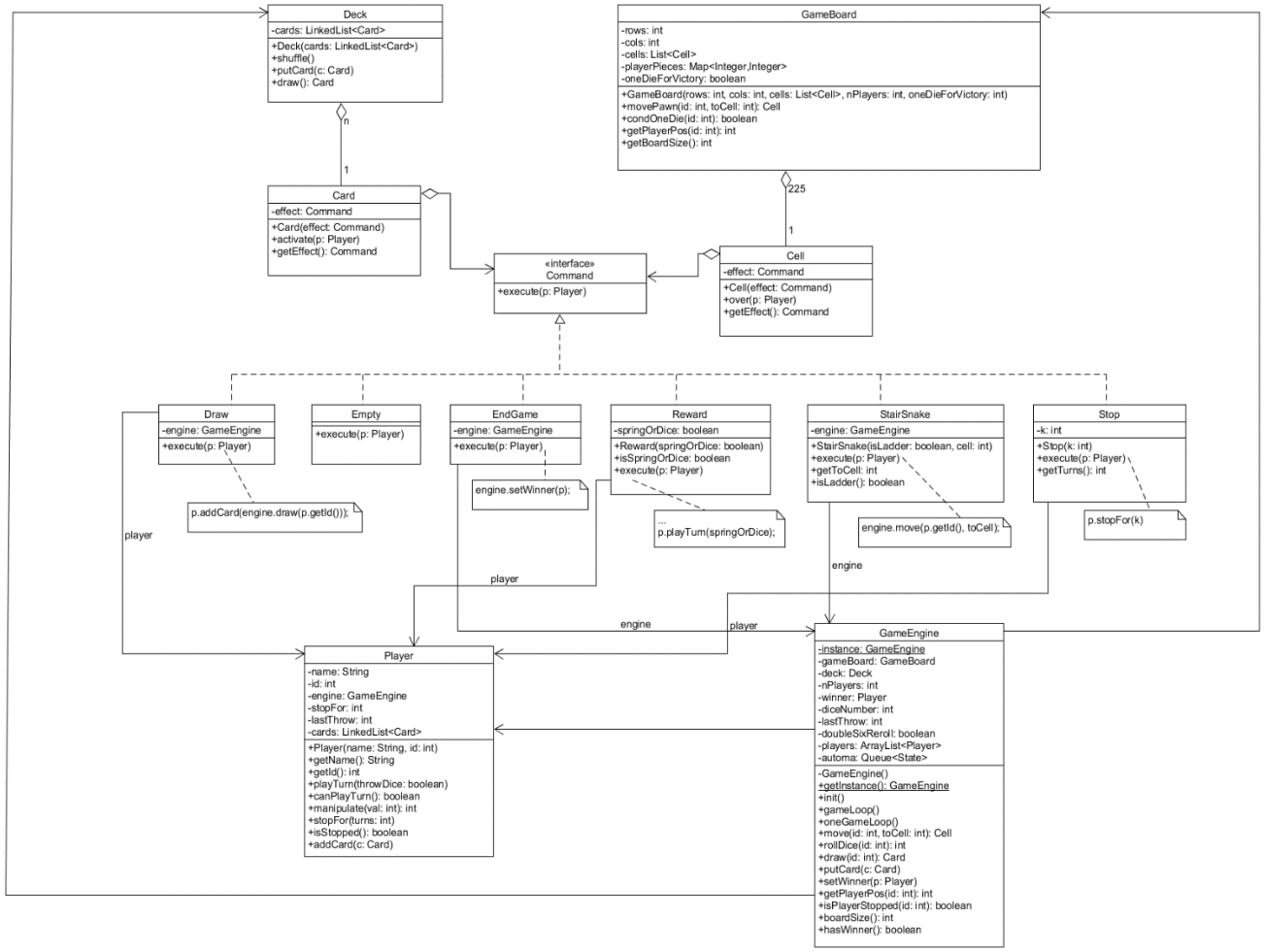
Nel caso in cui l'utente scelga di inserire nel tabellone celle speciali che permettono ai giocatori di pescare una carta, il Game Engine sarà responsabile di gestirne l'interazione con il Deck. Quando, durante la partita, il Player si trova su una cella speciale "Pesca una carta", dovrà pescare una carta e attivare il suo effetto o, nel caso della carta "Divieto di Sosta", inserirla nella sua mano e utilizzarla in un secondo momento. Il giocatore invierà quindi al Game Engine la sua richiesta e quest'ultimo la dovrà inoltrare al Deck. Una volta utilizzata la carta, il giocatore dovrà inserirla nuovamente in fondo al mazzo di carte. La richiesta di reinserire la carta nel deck sarà soddisfatta, chiaramente, con le stesse modalità.



Effetti di carte e celle speciali

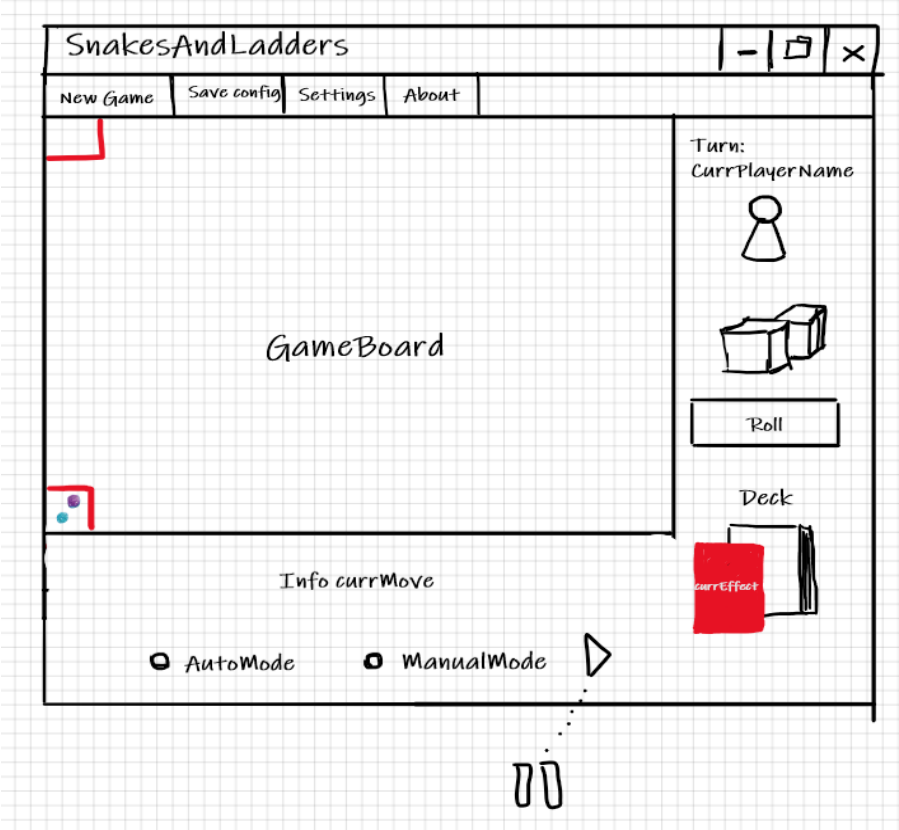
Quando un giocatore si posiziona su una cella speciale o quando pesca una carta (conseguenza del posizionamento sulla cella “Pesca una carta”), si deve assicurare l’attivazione del relativo effetto. Tuttavia, né celle né carte conoscono in che modo portare a termine l’operazione richiesta e non sanno chi dovrà soddisfarla. Per tale motivo, si è deciso di ricorrere all’implementazione del design pattern Command. L’interfaccia Command presenta, appunto, un’interfaccia per le esecuzioni di operazioni ed include il metodo astratto *execute()*. Ogni classe concreta che implementa tale interfaccia, specifica una particolare azione su un destinatario mantenendo, ove necessario, il riferimento al destinatario come attributo e implementando il metodo execute() in modo tale che invochi la richiesta desiderata. In questo caso specifico, gli Invoker che chiedono a Command di portare a termine la richiesta sono le classi Card e Cell, mentre i ConcreteCommand sono rappresentati dalle classi Draw, Empty, EndGame, Reward, StairSnake, Stop. Le classi Draw, Reward e Stop hanno come Receiver la classe Player, mentre il Receiver di EndGame e StairSnake è la classe GameEngine.

Class diagram della parte logica di Snakes and Ladders

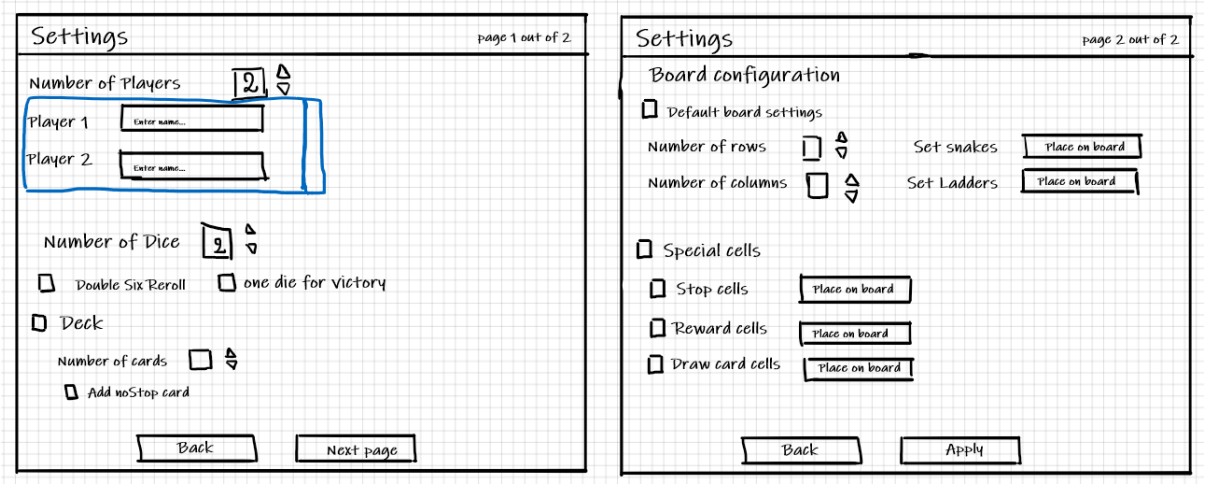


Progettazione della parte grafica

Il primo passo da effettuare per ideare una buona interfaccia grafica è capire come rendere l'esperienza di gioco dell'utente piacevole, conciliando quindi UI Design (User Interface) e UX Design (User Experience). In questi casi porsi dal lato utente è fondamentale al fine di curare l'estetica del gioco coerentemente con la tipologia. Nonostante chiunque possa giocare a Snakes and Ladders, questo gioco è nato principalmente come gioco da tavolo per bambini; di conseguenza, si è reso opportuno concentrarsi sulla realizzazione di un'interfaccia grafica non troppo elaborata ma allo stesso tempo stimolante, utilizzando icone semplici e colori vivaci.



Bozza dell'aspetto della Game Window



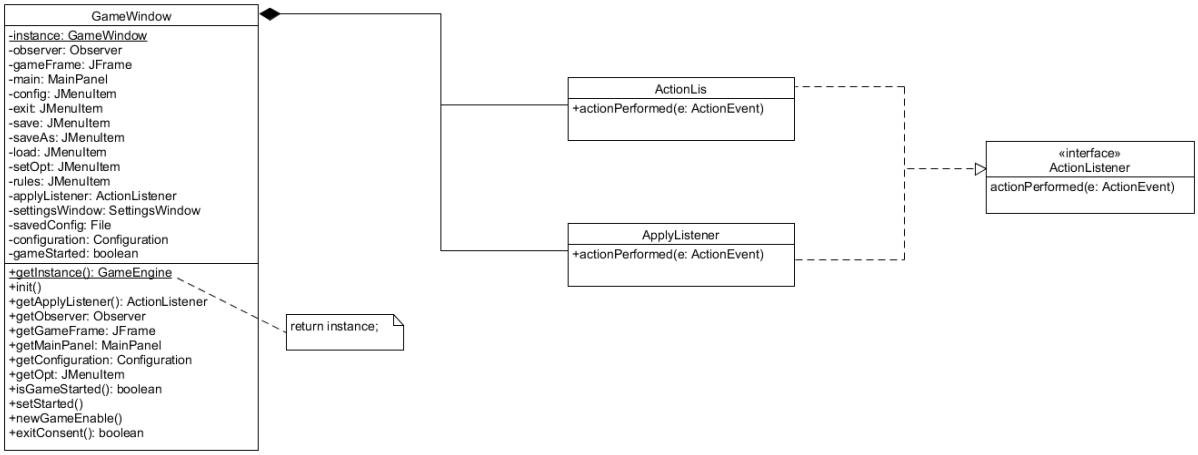
Bozza dell'aspetto della Settings Window (pagina 1 e 2)

Come discusso nel punto C. Architettura del Software, la Game Window rappresenta l'elemento principale all'interno del contesto grafico e mantiene un riferimento alla parte logica del gioco mediante il Game Engine. Strutturalmente, essa è costituita da un pannello

principale, il Main Panel, suddiviso a sua volta in tre pannelli separati: il Side Panel, il Bottom Panel e il Board Panel. Sul Side Panel, l'utente può facilmente osservare, durante la partita, l'icona e il nome del giocatore che sta giocando il proprio turno in un dato istante, l'icona dei dadi e il bottone sottostante che eventualmente può utilizzare per lanciarli e, se incluso, il mazzo di carte che, al momento opportuno, mostra la carta pescata dal giocatore. Sul Bottom Panel sono invece riportate le informazioni relative alla partita (es. quale giocatore ha lanciato i dadi, chi si sta muovendo sul tabellone, chi pesca una carta), il bottone di start/stop utilizzato per avviare o mettere in pausa la partita e i radio buttons che permettono all'utente di cambiare modalità di gioco (automatica o manuale) quando permesso. Tutti gli elementi grafici sono inseriti in base ai parametri specifici richiesti dall'utente in fase di configurazione attraverso la Settings Window.

GameWindow come Singleton

Nel contesto della progettazione della parte grafica del gioco, essendo la Game Window unica, deve essere garantita l'esistenza di un'unica istanza appartenente a tale classe cui tutti gli utilizzatori sono liberi di accedere.



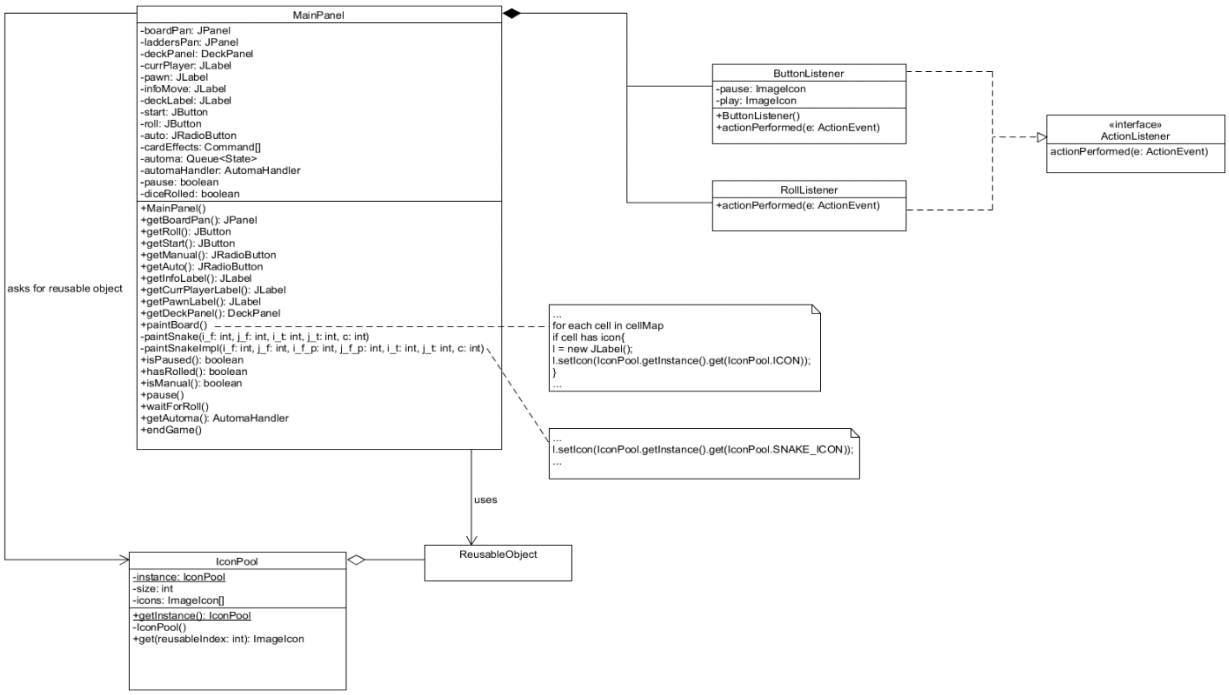
Ottimizzazione delle performance con l'utilizzo di Icon Pool

Gli elementi grafici presenti nell'interfaccia dovrebbero essere creati e istanziati su richiesta dell'utente durante il disegno della game board. In particolare, l'utente specifica tipologia e quantità delle icone che devono essere visualizzate una volta premuto il pulsante Apply attraverso la scelta e il posizionamento di celle speciali, scale e serpenti. Ciò significa che non è possibile rinviare il costo della creazione e inizializzazione di ciascun oggetto in quanto, nonostante essi siano effettivamente creati su richiesta, devono poter essere visualizzabili contemporaneamente. Inoltre, occorre tenere a mente che i costi di creazione e istanziazione di tutti gli elementi grafici sono pagati ogniqualvolta il tabellone di gioco viene ridisegnato in seguito al caricamento di una nuova configurazione di gioco

o modifica delle impostazioni. Si potrebbe pensare allora di implementare il pattern Flyweight per supportare in modo efficiente un gran numero di oggetti che altrimenti utilizzerebbero una quantità notevole di memoria e introdurrebbero un carico inaccettabile durante l'esecuzione dell'applicazione. Tuttavia, i benefici derivanti dall'utilizzo di tale pattern avrebbero avuto un peso minore rispetto alla complessità di progettazione dello stesso. Pertanto, si è presa in considerazione la possibilità di creare e istanziare anticipatamente tutti gli elementi grafici e posizionarli in una classe pool, da cui verranno prelevati in seguito per poter essere riutilizzati. Questo è lo scopo del design pattern creazionale Object Pool, ampiamente utilizzato nei videogiochi per l'utilizzo di entità di gioco o effetti visivi. Il pattern Object Pool è utilizzato nei seguenti casi:

- Quando è necessario creare e distruggere frequentemente oggetti;
- Gli oggetti hanno simili dimensioni;
- L'allocazione di oggetti nell'heap è lenta o potrebbe causare la frammentazione della memoria;
- Ogni oggetto incapsula una risorsa come un database o una connessione di rete che è costosa da acquisire e potrebbe essere riutilizzata.

Quando il pool viene inizializzato, crea l'intera raccolta di oggetti in anticipo, di solito in un'unica allocazione contigua. In questo modo si evita che gli oggetti vengano continuamente creati e distrutti ad ogni invocazione di paintBoard() causando una eventuale frammentazione della memoria. Inoltre, anche l'Icon Pool è implementata come Singleton poichè, come accade per Game Window e Game Engine, deve essere garantita l'esistenza di un'unica pool nonché l'accesso globale a coloro i quali lo richiedono.



Questo design pattern è molto simile a Flyweight: entrambi mantengono una collezione di oggetti riusabili, ma la differenza sta in come questi oggetti sono usati. Gli oggetti Flyweight vengono riutilizzati condividendo la stessa istanza fra più clients e utilizzandola simultaneamente in più contesti. Gli oggetti di un pool possono, invece, essere utilizzati simultaneamente da un singolo client e non vi è alcuna separazione tra stato intrinseco ed estrinseco come accade per gli oggetti Flyweight.

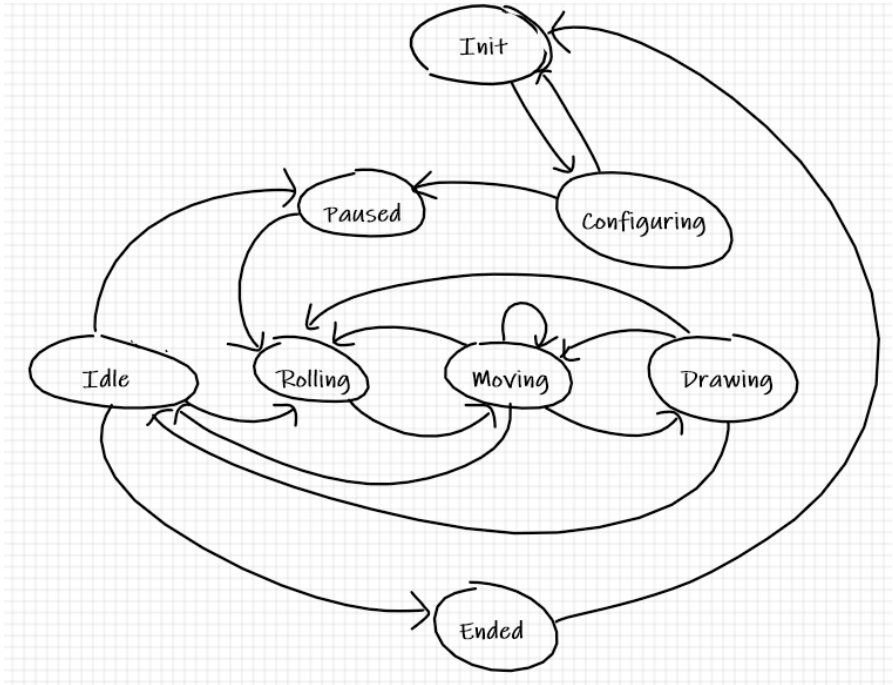
Traccia degli eventi durante la partita: i pattern State e Observer

Al fine di fornire un'esperienza utente completa, le limitazioni grafiche devono essere compensate da informazioni testuali visibili sulla Game Window. Nonostante la "logica a strati" adoperata per la progettazione delle celle della game board, utile a fornire una corretta visualizzazione di tutti gli elementi grafici anche quando i giocatori si muovono sul tabellone di gioco, non è sempre facile tenere traccia degli eventi durante la partita. Per questo motivo, si è reso necessario pianificare un modo per risolvere tale problematica. L'incedere delle fasi del gioco è scandito nell'interfaccia grafica dal design pattern State. Si individuano, quindi, cinque stati principali:

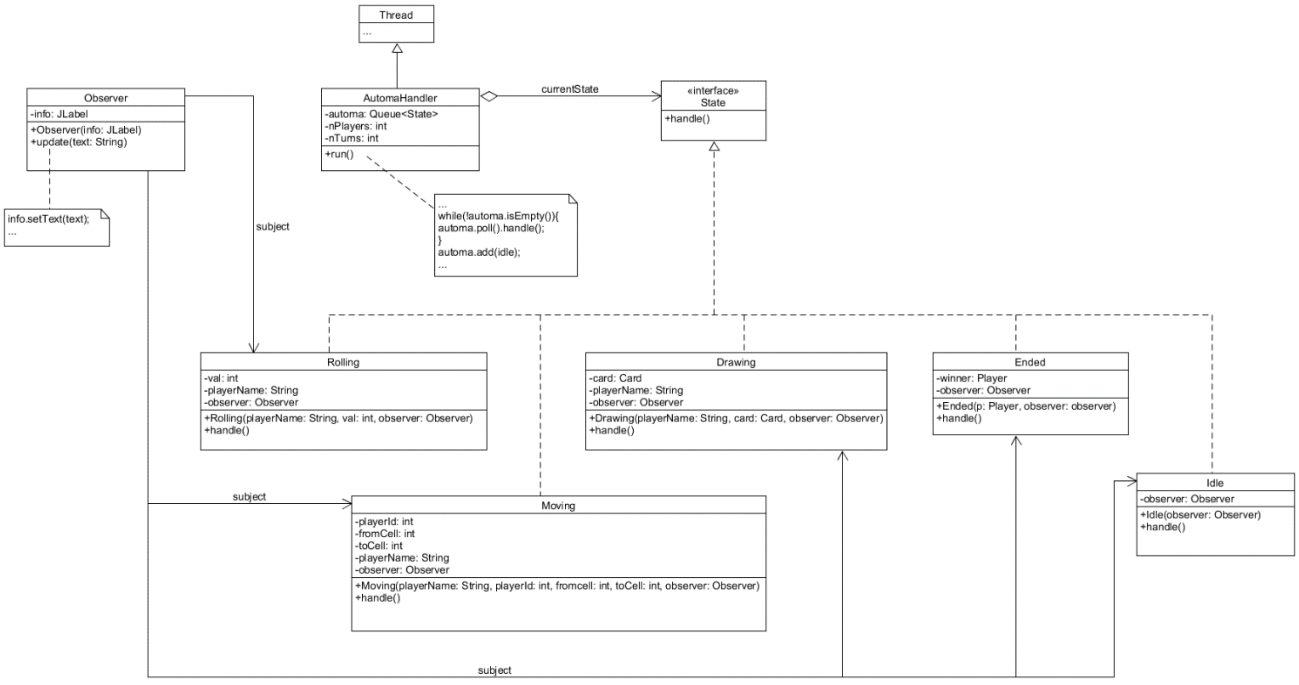
- Idle: come suggerisce il nome, questo stato "inattivo" rappresenta una fase di transizione. Esso è stato inserito per permettere all'utente di mettere in pausa il gioco o cambiare la modalità di gioco da automatico a manuale e viceversa. La motivazione risiede nel fatto che, in questo modo, è più facile gestire l'avanzamento delle fasi di gioco in quanto si evita di interrompere il flusso di azioni necessarie per il compimento di ciascun turno di gioco.
- Rolling: il giocatore corrente ha lanciato i dadi (modalità automatica) oppure l'utente ha premuto il pulsante "Roll Dice". In questa fase si calcola il valore totale ottenuto dal lancio dei dadi/del dado.
- Moving: è stata calcolata la cella su cui il giocatore corrente deve posizionarsi; tale informazione verrà visualizzata dall'utente nel momento in cui il giocatore si muove sul tabellone.
- Drawing: caso particolare riguarda la pescata di una carta da parte del giocatore. Nel momento in cui il giocatore si posiziona su una cella che ha come effetto la pesca di una carta dal mazzo, l'utente sarà avvisato di questo avvenimento e potrà visualizzare la carta pescata sul pannello laterale.
- Ended: questo stato è raggiunto nel momento in cui un giocatore si trova sulla cella finale; in questo caso, verrà visualizzato il nome del vincitore e la partita terminerà.

Il modo in cui questi stati si alternano l'un l'altro durante la partita è rappresentato da un automa a stati finiti, implementato come una coda di State. L'AutomaHandler ha la responsabilità di prelevare dall'automa ogni stato e gestirlo in modo appropriato. Infine, le informazioni visualizzate dall'utente nel box posizionato sul Bottom Panel saranno effettivamente aggiornate da un oggetto observer, il quale riceve una notifica

dall'AutomaHandler ogniqualvolta avviene un cambiamento di stato all'interno dell'automa, il soggetto osservato.



Bozza dell'automa a stati finiti



G. Spiegare come il progetto soddisfa i requisiti funzionali (FRs) e quelli non funzionali (NFRs)

Nelle sezioni precedenti si è ampiamente discusso delle modalità con cui l'utente può creare una nuova configurazione impostando opportunamente i parametri presenti nella finestra delle impostazioni e salvare su file per poi ripristinarla in seguito. Si è inoltre discusso dell'implementazione della logica di gioco grazie al controllo centralizzato operato dal Game Engine e di come la Game Window svolga un ruolo principale nella parte grafica. Risulta interessante capire, invece, come siano soddisfatti i requisiti di usabilità del sistema garantendo una corretta visualizzazione degli elementi grafici. Come introdotto nella sezione E, la Game Board utilizza un GridBagLayout come Layout Manager, quindi tutte le icone richieste all'Icon Pool devono essere posizionate sulla griglia di gioco secondo questo layout e implementando la logica "a strati" discussa. Certamente, diversa è la situazione nel caso di serpenti e scale.

Rappresentazione dei serpenti

Il disegno dei serpenti sulla game board è effettuato esattamente nel momento in cui essa viene disegnata. Dato il vincolo che impone una rappresentazione a celle di qualsiasi immagine presente sulla board, si è reso necessario pianificare un algoritmo che, dati in input le posizioni di testa e coda, disegna l'intero corpo del serpente a partire dalla testa. Ovviamente, per non rendere l'algoritmo più complesso del necessario, non sono state prese in considerazione tutte le casistiche possibili, ma è stata inserito comunque un minimo di casualità per rendere quantomeno realistico il tipico andamento sinuoso del serpente.

Algoritmo di disegno ricorsivo (paintSnake)

se riga_testa == riga_coda → coda raggiunta
<u>se il pezzo corrente da inserire si trova sotto quello precedente</u>:
    se il pezzo corrente si trova a sinistra della coda:
        se riga_coda == riga_pezzo_corrente -2:

            scegli casualmente tra e
            paintSnake(…)

se il pezzo corrente si trova a destra della coda:
    se riga_coda == riga_pezzo_corrente -2:

            scegli casualmente tra e
            paintSnake(…)

se il pezzo corrente si trova esattamente sopra la coda:

    usa
    paintSnake(…)

<u>se il pezzo corrente da inserire si trova sulla stessa riga di quello precedente:</u>
    se il pezzo precedente è a sinistra di quello corrente:
        se la coda si trova sulla stessa colonna:

            usa
            paintSnake(…)

            altrimenti:
                se riga_corrente == riga_coda +1:

usa
paintSnake(…)

altrimenti:


scegli casualmente tra e
paintSnake(…)

se il pezzo precedente è a destra di quello corrente:
se colonna_corrente == colonna_coda:

usa
paintSnake(…)

altrimenti:
se riga_corrente == riga_coda+1:

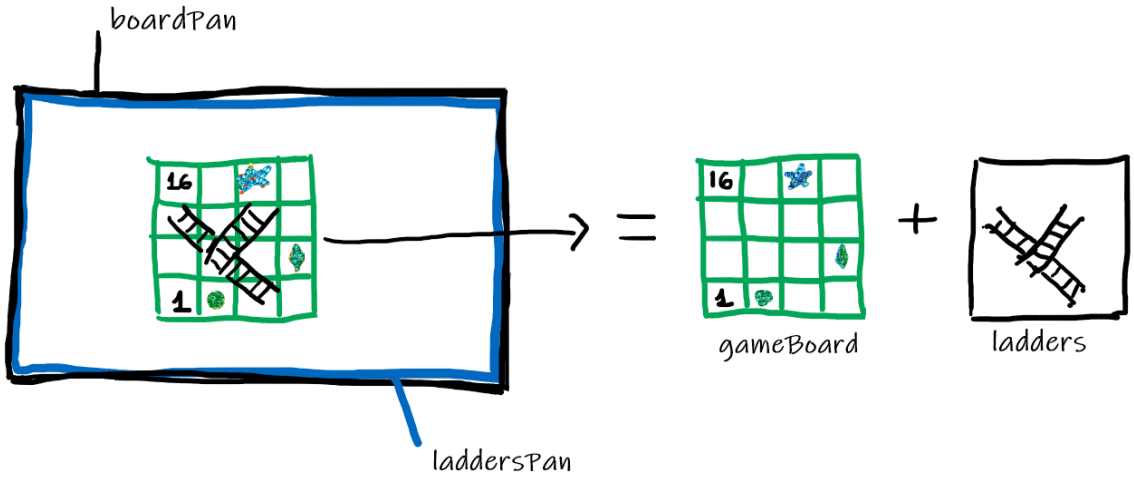
usa
paintSnake(…)

altrimenti:


scegli tra e
paintSnake(…)

Rappresentazione delle scale

Lo stesso tipo di rappresentazione dei serpenti non è applicabile alle scale sapendo che queste, a seconda della loro posizione, possono avere un gran numero di inclinazioni differenti. Per questo motivo, si è deciso di non utilizzare alcun tipo di icona prefabbricata né ideare un algoritmo come fatto per i serpenti. La soluzione migliore consiste nell'ispirarsi al comportamento del pannello contenente la game board e riprodurlo creando un secondo pannello dedicato al disegno delle scale. Ogniqualvolta venga invocato il metodo paintBoard(), sul pannello boardPan è inserita la game board impostata dall'utente, composta da celle che sono a loro volta piccoli pannelli sui quali sono presenti tutti gli elementi grafici. Se paintBoard() viene nuovamente invocato, la game board sarà prima rimossa dal board panel e poi ridisegnata in base alla nuova configurazione. Lo stesso comportamento è riprodotto per il disegno delle scale: sul board panel su cui è disegnata la griglia di gioco, è posto un secondo pannello (laddersPan), su cui sarà posizionato il pannello contenente le scale. Nel momento in cui la game board viene disegnata, anche le scale saranno disegnate sull'apposito pannello posto sulla game board avente le sue stesse dimensioni. Quando richiesto, la griglia di gioco e il pannello posto sopra di essa, contenente le scale, saranno cancellati, mentre i pannelli boardPan e laddersPan rimarranno comunque posizionati uno sopra l'altro.



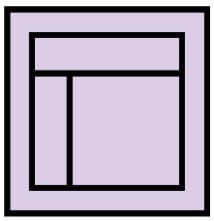
Visualizzazione della carta pescata

Oltre a tutti i benefici derivanti dall'utilizzo di un automa, discussi nella sezione E, vi è anche la facilità di gestione della visualizzazione delle carte pescate. Difatti, quando il giocatore simulato si posiziona su una casella "pesca una carta" attivandone l'effetto, l'automa si ritrova nello stato di Drawing, in cui si imposta la corretta icona (richiesta all'Icon Pool) da visualizzare sopra il mazzo di carte, sul pannello laterale.

Visualizzazione del nome e della pedina del giocatore corrente

In questo caso, il responsabile dell'aggiornamento del colore della pedina e del nome del giocatore corrente (elementi posti sulla parte superiore del pannello laterale) è il thread gestore dell'automa, che richiede le informazioni necessarie alla Game Window e le utilizza in fase di Rolling.

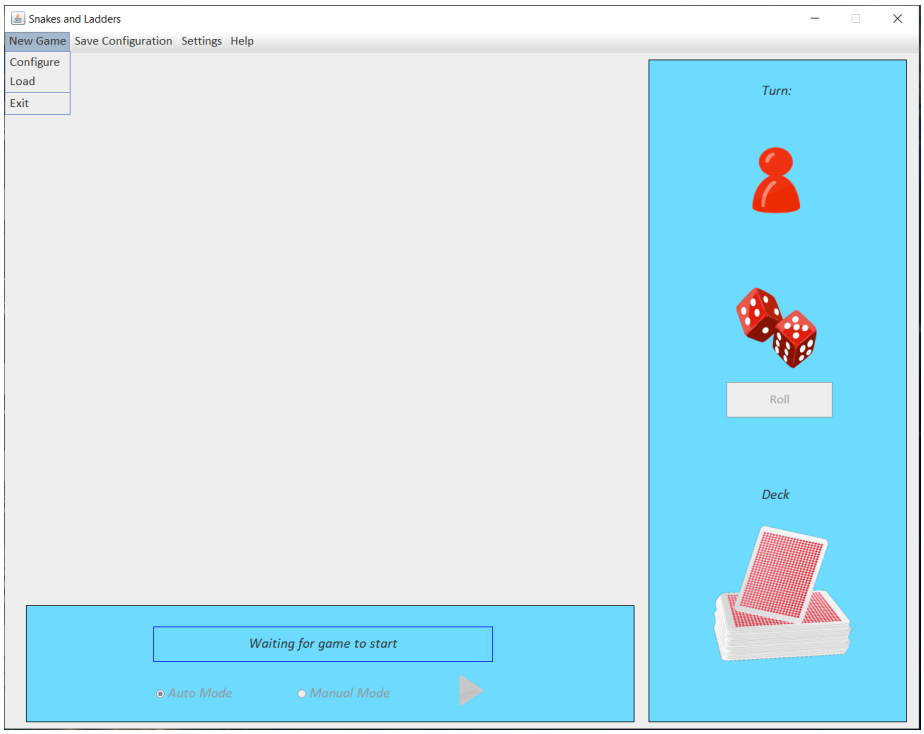
Appendix. Prototype



In questa sezione sono riportati gli esempi grafici di come l'applicativo soddisfa requisiti funzionali e non funzionali secondo le modalità descritte nella presente relazione.

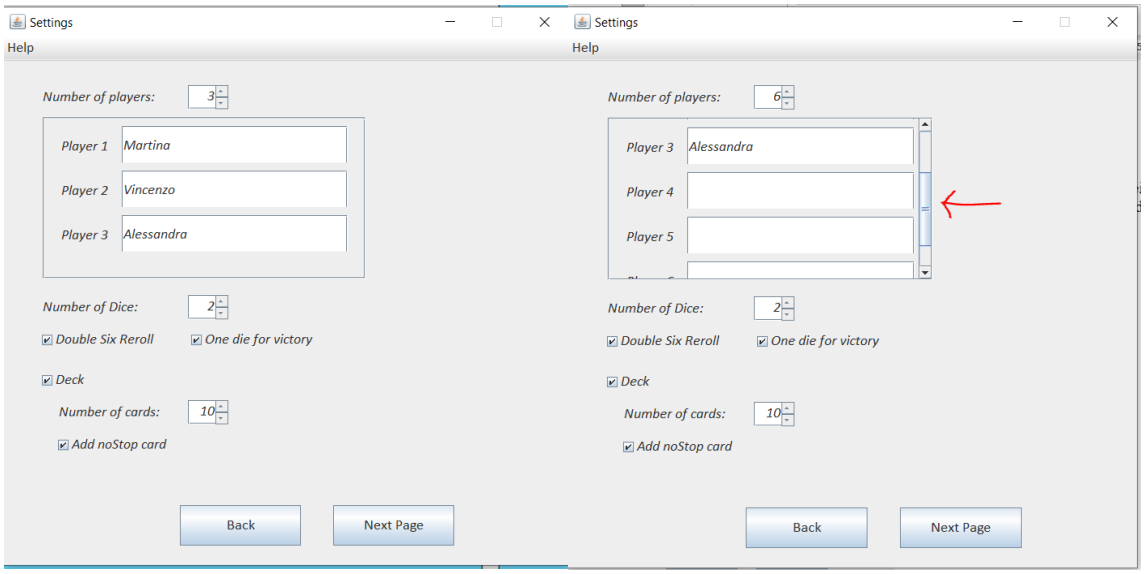
Configurazione/Caricamento partita

Per creare o caricare una nuova configurazione di gioco, è necessario cliccare sulla voce "New Game" nella barra dei menu e selezionare l'opzione desiderata.

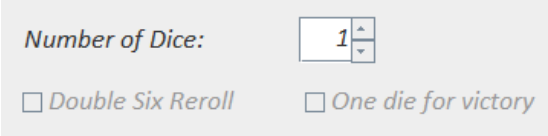


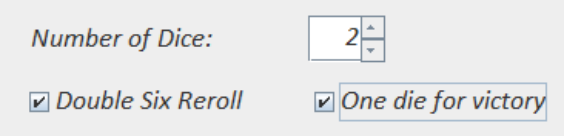
Impostazioni di gioco

Numero di giocatori: il minimo numero di giocatori selezionabili è due, il massimo è sei. Se si aggiungono più di tre giocatori, una barra di scorrimento laterale permetterà di visualizzare l'intero contenuto del riquadro.

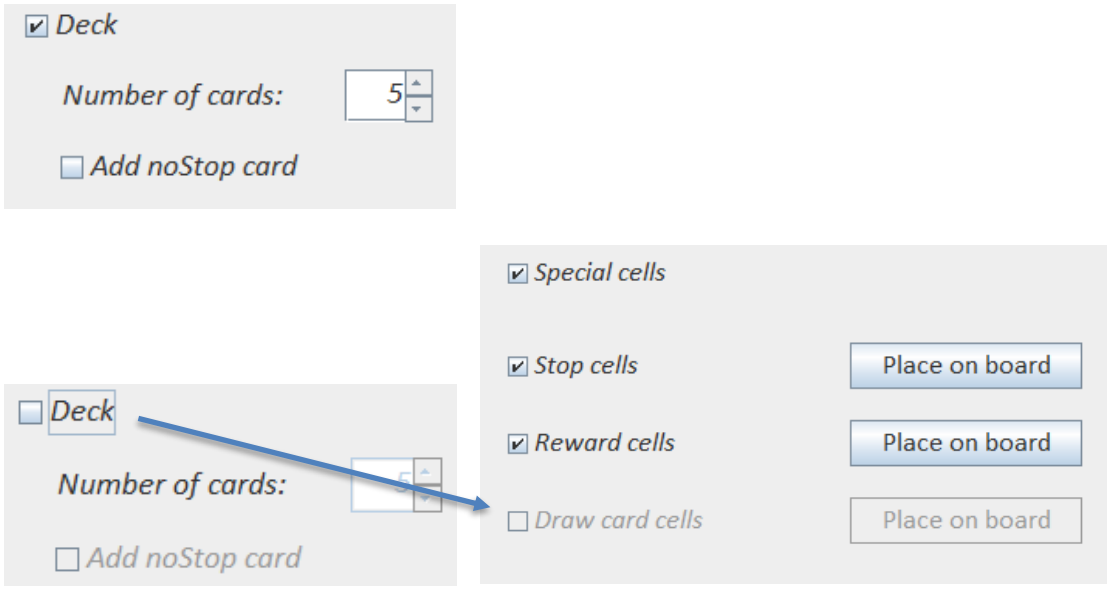


Numero di dadi: è possibile scegliere di utilizzare uno o due dadi. Selezionando due dadi, è anche possibile scegliere se utilizzare le regole associate.

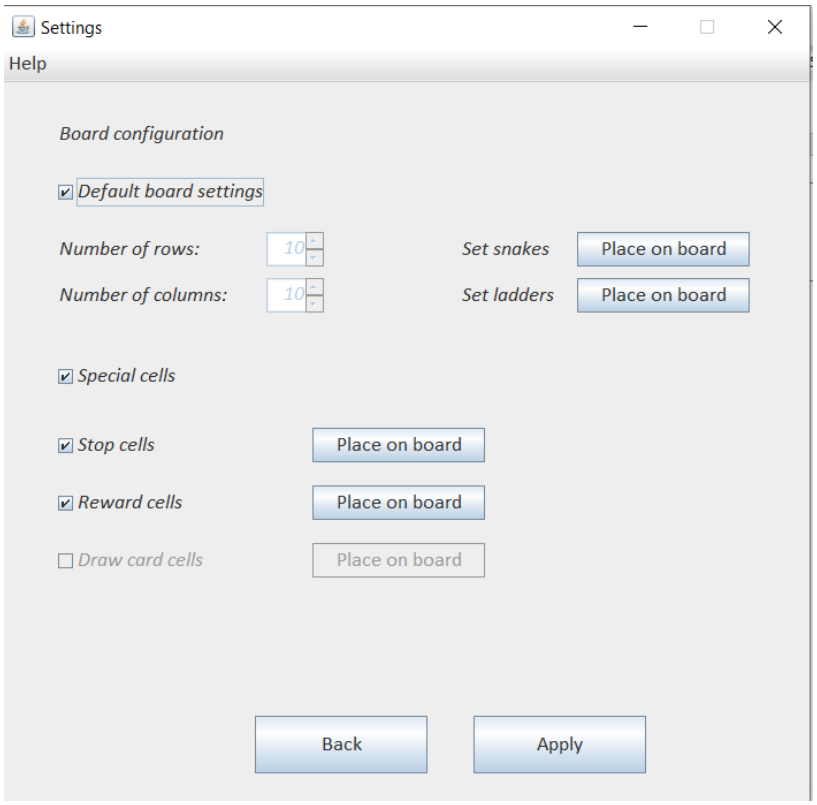


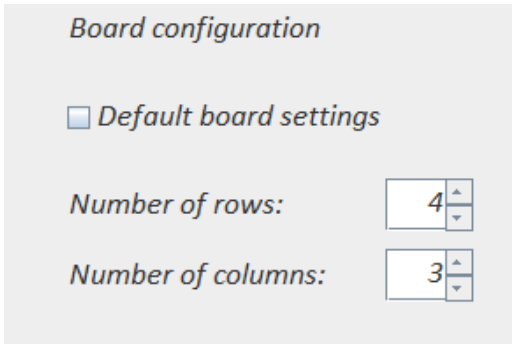


Selezione opzione deck: spuntando la casella Deck si sceglie di utilizzare un mazzo di carte, da cui si estrarrà una carta ogni volta che un giocatore si posiziona su una cella "pesca una carta". Se questa opzione non è selezionata, non è possibile aggiungere sul tabellone di gioco le caselle speciali associate. Il minimo numero di carte da inserire all'interno di un deck è cinque (una per ogni tipo di effetto), mentre non vi è un limite superiore. In questa sezione è anche possibile selezionare o meno l'opzione per aggiungere al mazzo la carta "divieto di sosta", qui nominata "noStop card".

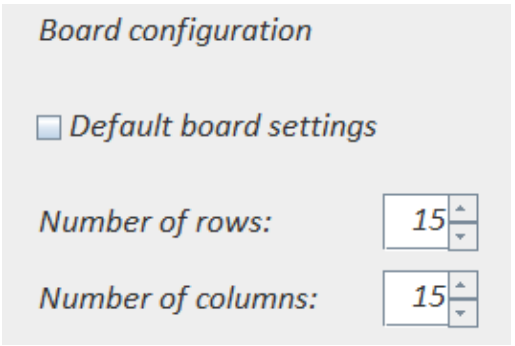


Impostazione dimensioni della game board: in questa finestra, l'utente può scegliere di mantenere le impostazioni di default (scegliendo quindi di utilizzare un tabellone 10x10) o modificare le dimensioni a proprio piacimento.



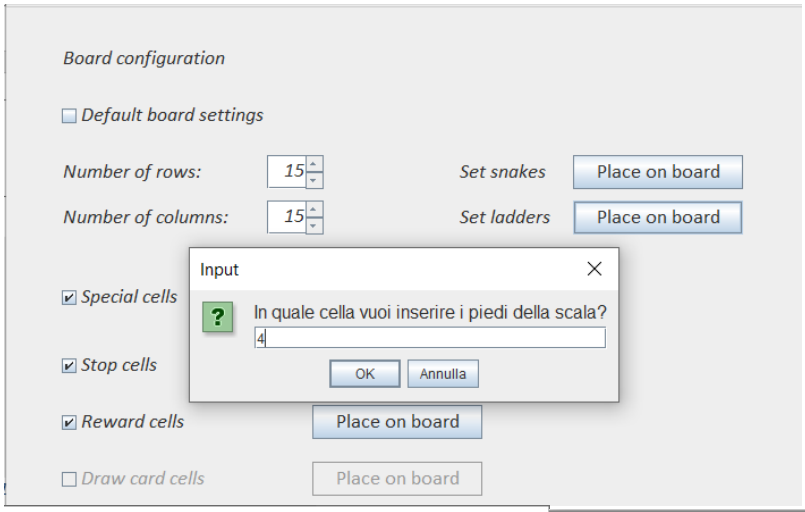


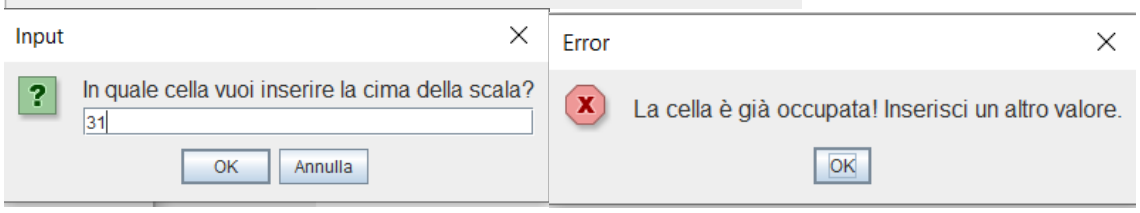
Dimensioni minime



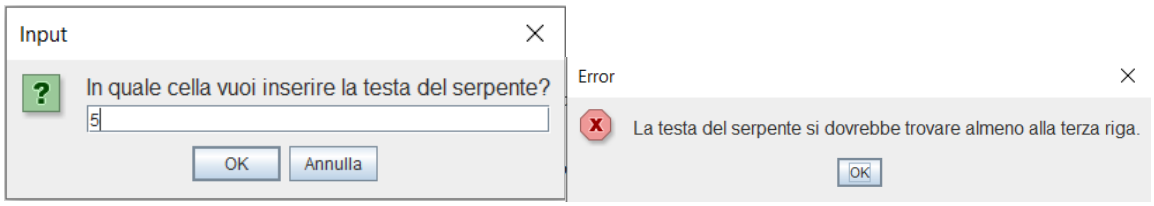
Dimensioni massime

Inserimento scale: l'utente può inserire le scale sul tabellone di gioco secondo le regole specificate cliccando sul pulsante "place on board". Nel caso in cui i valori inseriti non siano validi, verranno visualizzati dei messaggi di errore/warning.

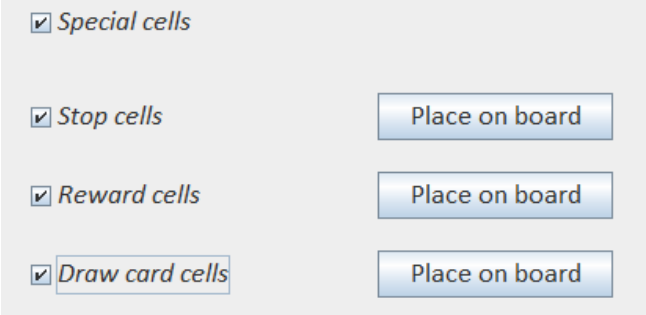


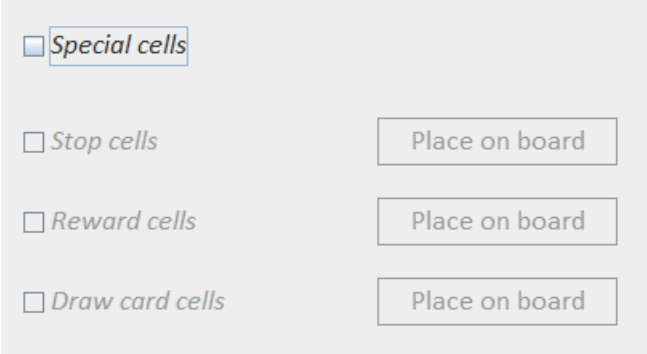


Inserimento serpenti: allo stesso modo, sarà possibile inserire i serpenti sul tabellone di gioco.

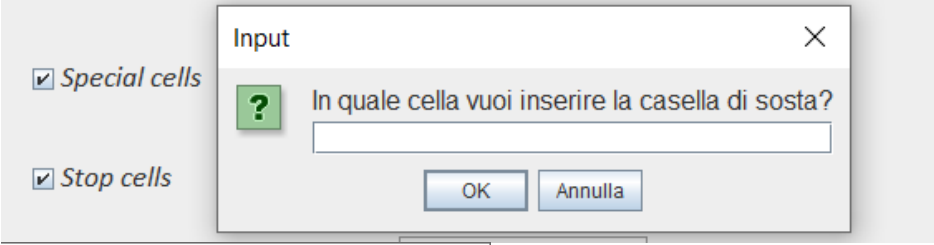


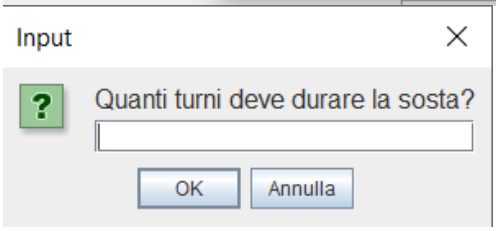
Inserimento celle speciali: con le stesse modalità, è possibile inserire le celle speciali sul tabellone di gioco.



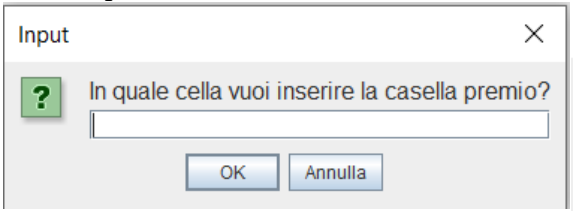


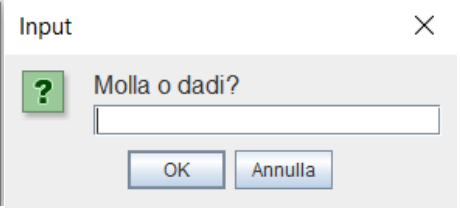
Caselle di sosta

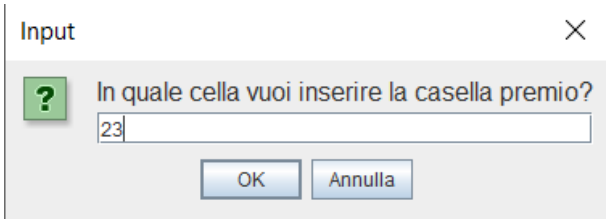


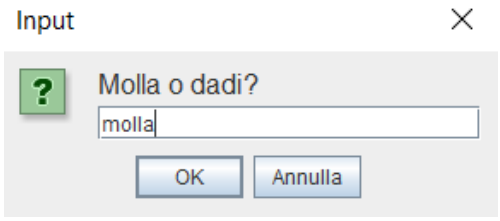


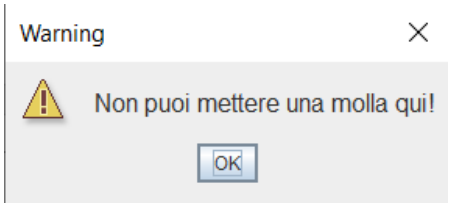
Caselle premio





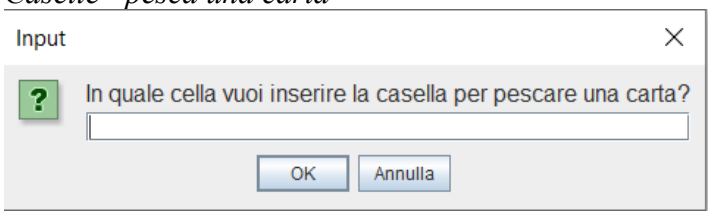




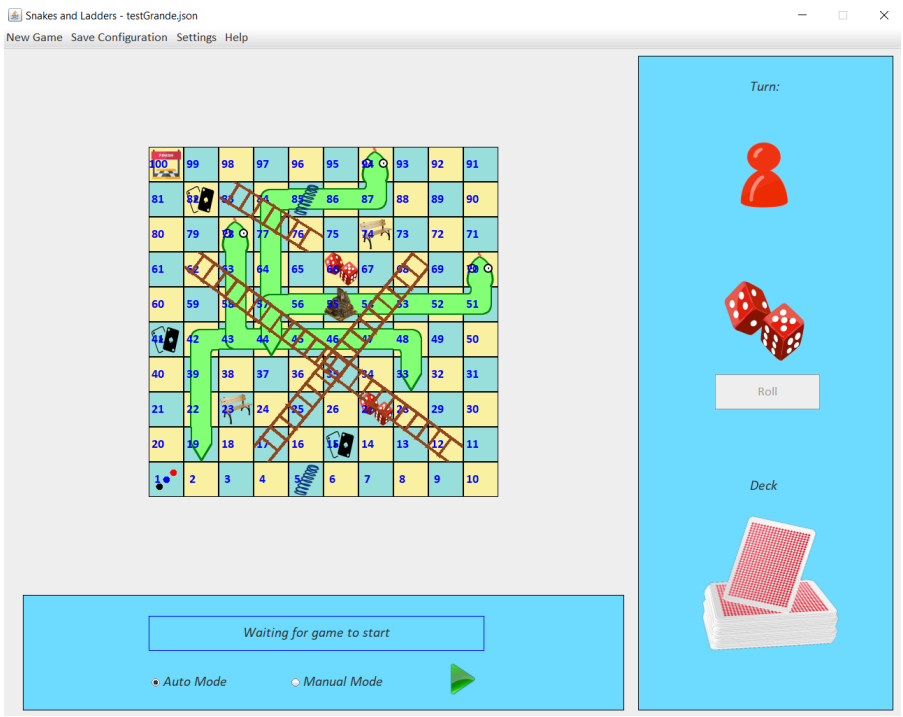


Come evitare il bug "molla"

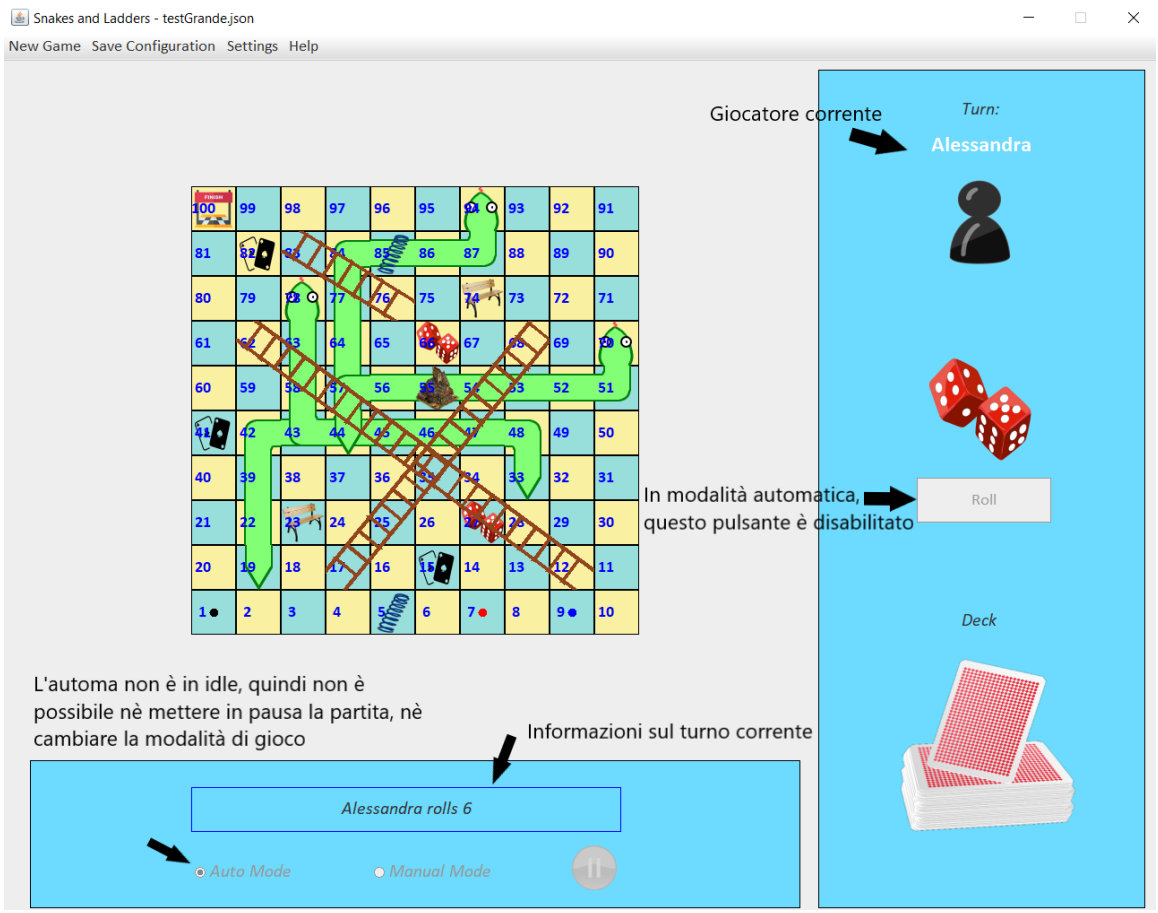
Caselle "pesca una carta"



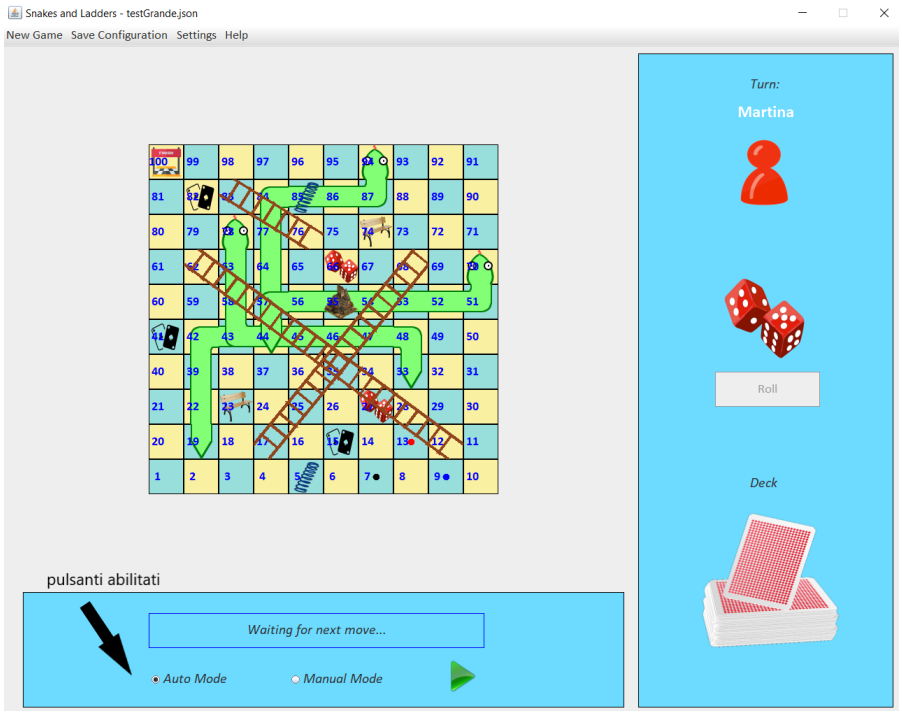
Di seguito è riportato l'aspetto dell'interfaccia grafica una volta applicate le impostazioni definite dall'utente.



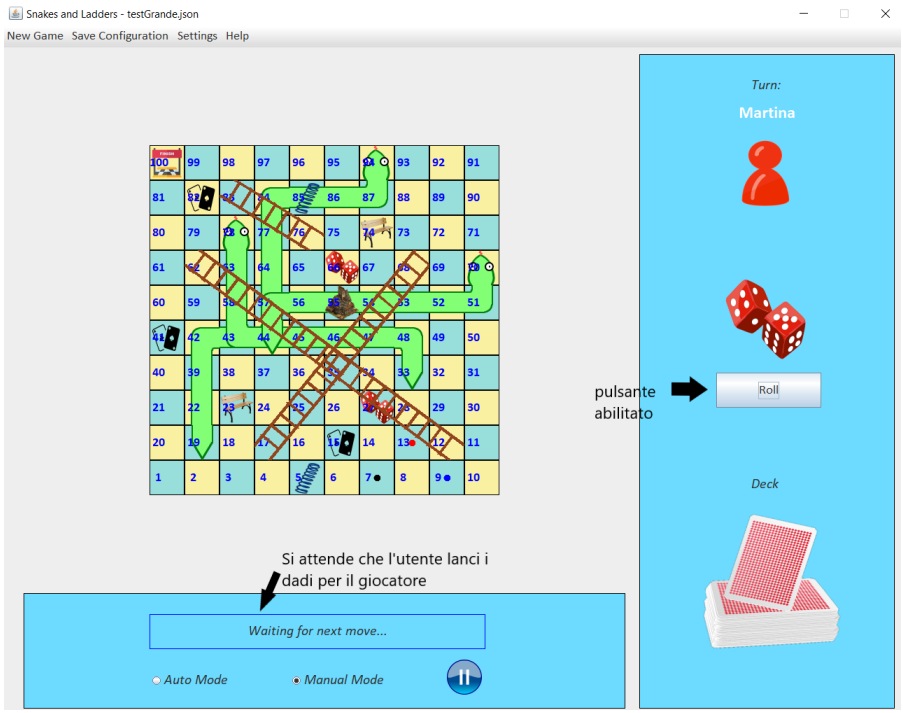
Partita in esecuzione



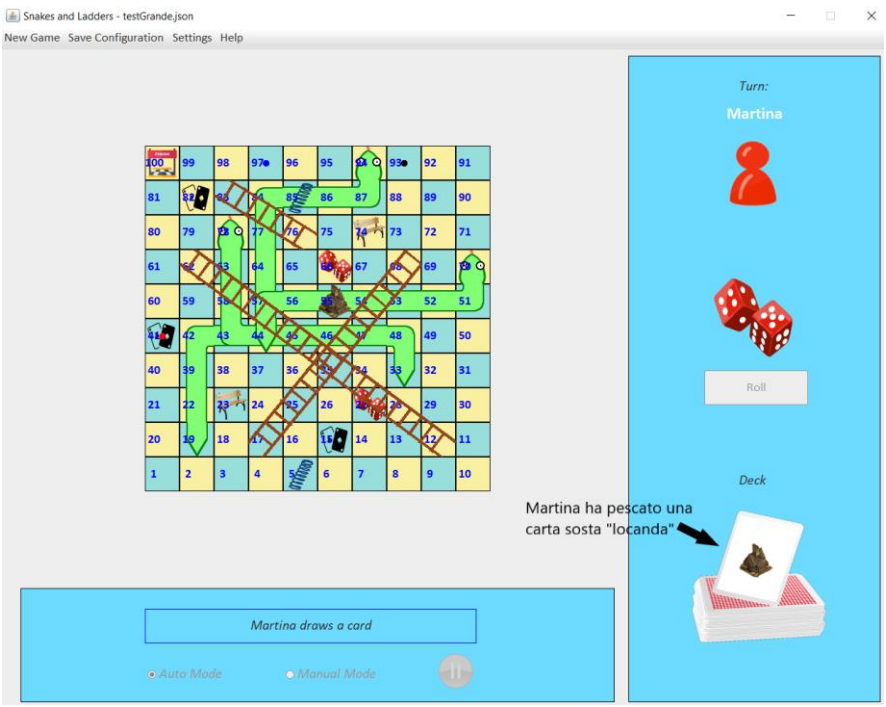
Partita in pausa



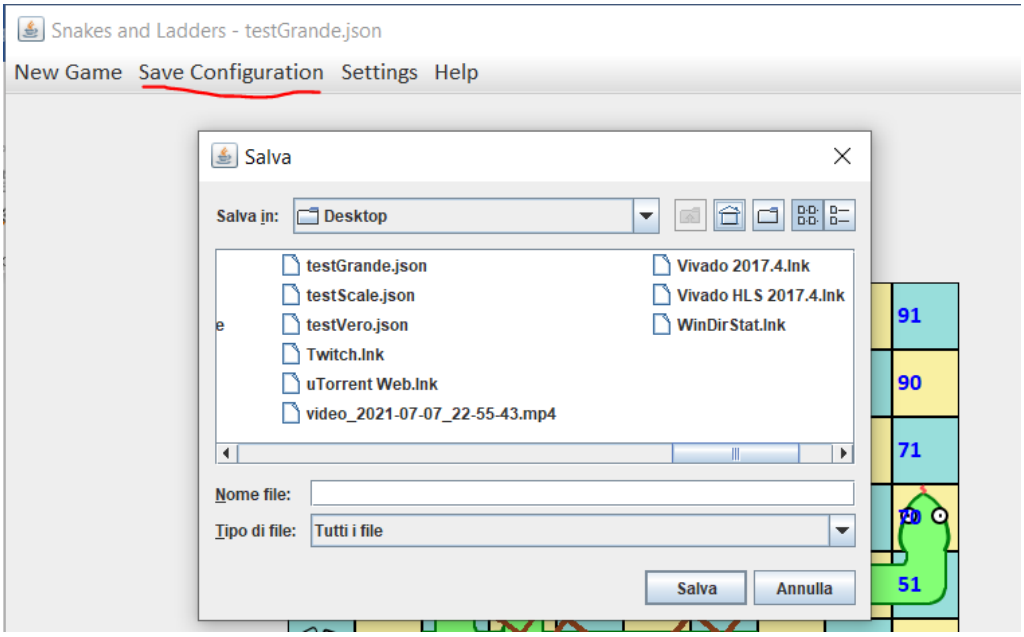
Modalità manuale attiva



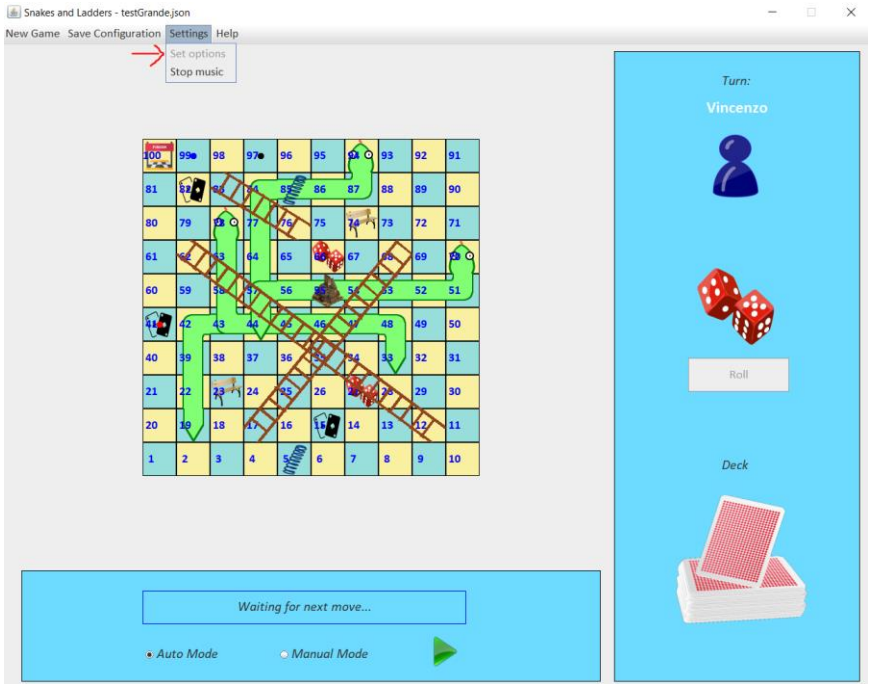
Estrazione di una carta



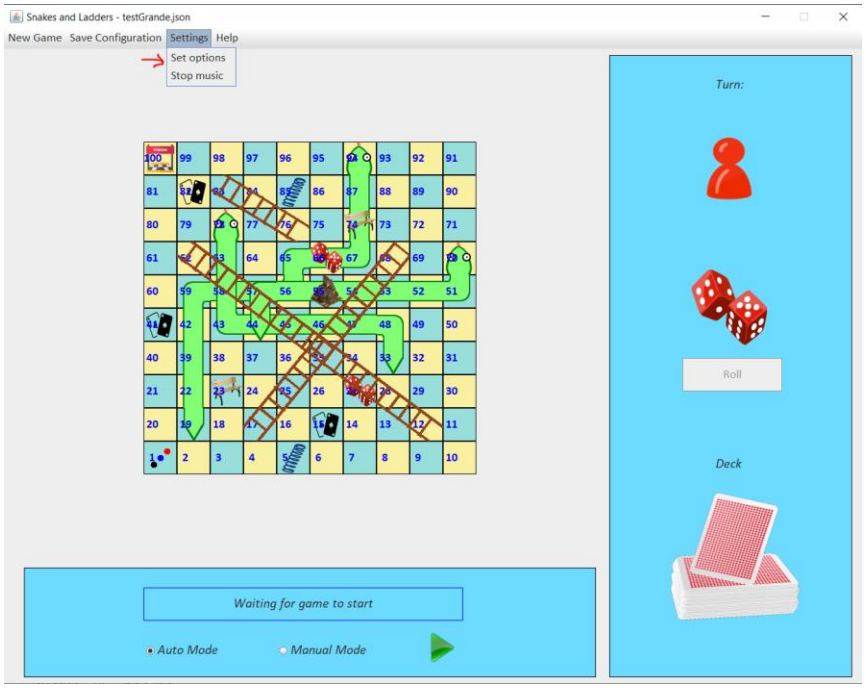
Salvataggio della configurazione corrente



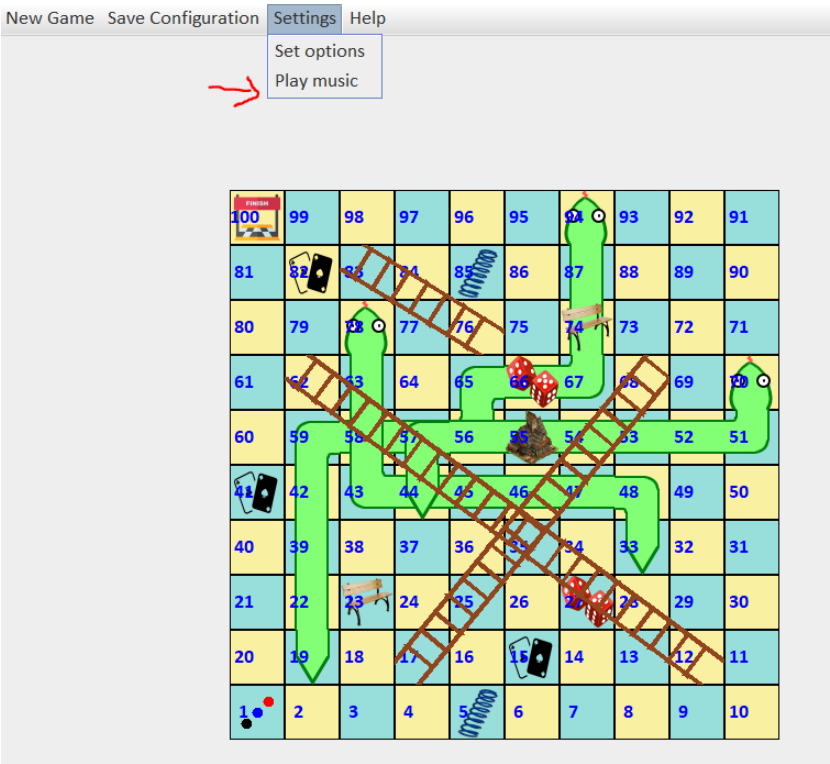
Modifica di una configurazione appena impostata/caricata e disattivazione dell'audio di gioco



"Set options" è disabilitato perché la partita è in corso



"Set options" è abilitato perchè la partita ancora deve iniziare



Se l'audio è già disattivato, invece di "Stop music" sarà visualizzato "Play music"