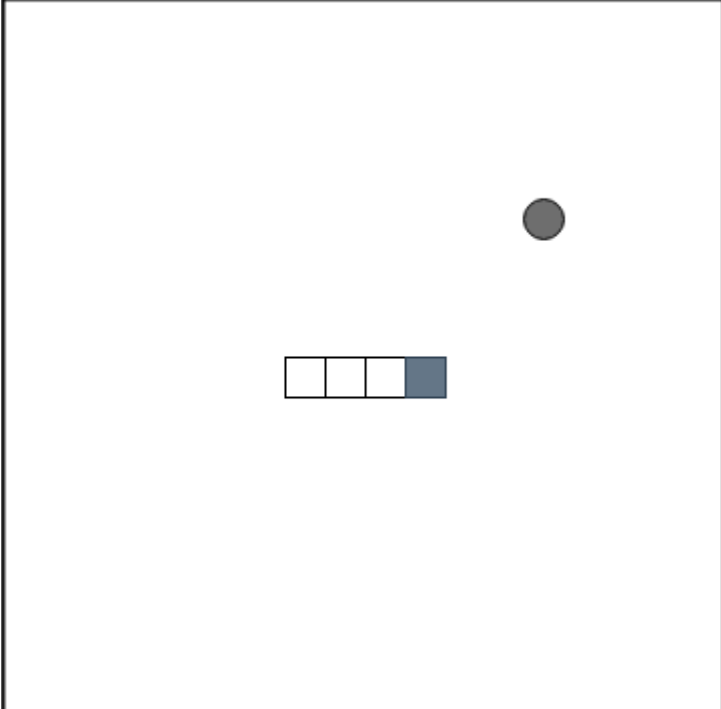


Snake

Documento di Analisi

Vista statica dell'interfaccia

	<table border="1"><thead><tr><th colspan="2">Leaderboard</th></tr></thead><tbody><tr><td>1. Valerio</td><td>120</td></tr><tr><td>2. Francesco</td><td>80</td></tr><tr><td>3. Nicola</td><td>65</td></tr></tbody></table>	Leaderboard		1. Valerio	120	2. Francesco	80	3. Nicola	65
	Leaderboard								
1. Valerio	120								
2. Francesco	80								
3. Nicola	65								
	<p>Score: 0</p> <p>Player ID: <input type="text" value="Lorenzo"/></p> <p><input type="button" value="Start"/></p>								

Caso d'uso: nuova partita

1. l'Utente inserisce il Player ID
2. l'Utente preme Start
3. il Sistema muove costantemente Snake lungo la scia tracciata dalla testa
4. l'Utente sceglie la prossima direzione di Snake tramite tastiera (default: tasti freccia)
5. IF Snake passa sopra Apple
 - 5.1. il Sistema aumenta Score di 5
 - 5.2. il Sistema aumenta la lunghezza di Snake
 - 5.3. il Sistema cambia la posizione di Apple
6. IF Snake sbatte contro un ostacolo
 - 6.1. il Sistema interrompe Snake
 - 6.2. il Sistema aggiorna la Leaderboard

File di configurazione locale in XML

All'avvio il Sistema legge dal file di configurazione i seguenti dati:

- Difficolta' del gioco
- Tasti associati al controllo del serpente
- Colori del serpente
- Font, dimensioni, colore del background
- Numero massimo di elementi della tabella dei record
- Periodo (espresso in giorni) mostrato dalla tabella
- Indirizzo IP del client
- Indirizzo IP e porta del server di log
- Indirizzo IP e porta del database
- User e password del database

Cache Locale degli input

Alla chiusura il Sistema salva su file lo stato corrente del gioco, il Player ID e lo Score.

All'avvio il Sistema carica da file suddetti dati.

Archivio

Il Sistema archivia i seguenti dati:

- Player ID del giocatore
- record personale del giocatore
- data ed ora del nuovo record

File di log remoto in XML

Il Sistema invia una riga di log ad ogni evento di seguito:

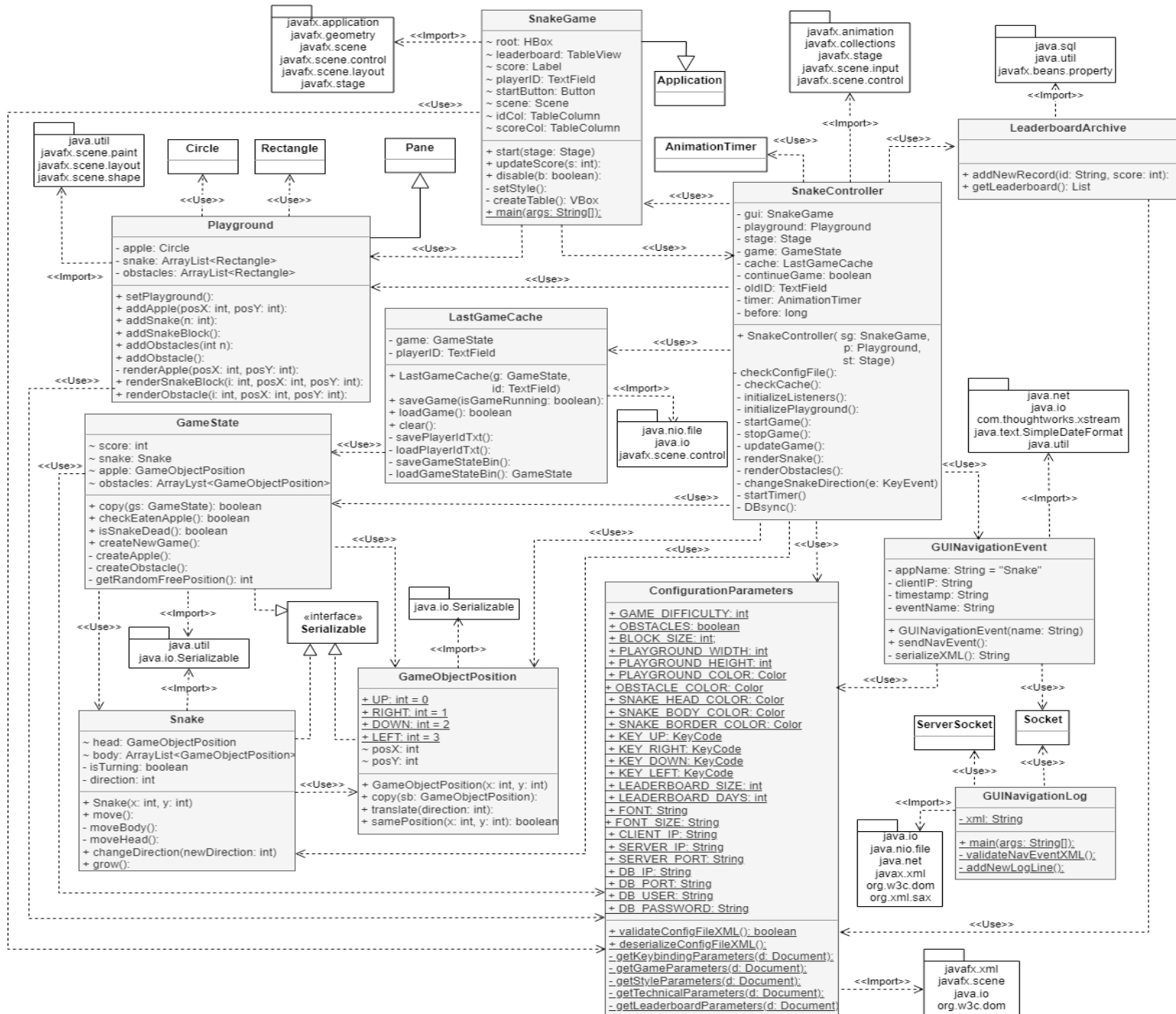
- Avvio dell'applicazione (“AVVIO”)
- Pressione del pulsante “START”
- Termine dell'applicazione (“TERMINE”)

La riga di log contiene:

- nome dell'applicazione
- indirizzo IP del client
- data-ora corrente
- etichetta associata all'evento

Documento di Progetto

Diagramma delle classi



Responsabilita' delle classi

Classe **ConfigurationParameters**: legge il file di configurazione XML; invoca la validazione del file di configurazione XML, deserializza il contenuto XML; mette a disposizione di qualsiasi classe i vari parametri

Classe **LastGameCache**: contiene su file di testo l'ultimo PlayerID inserito, su file binario lo stato di gioco dell'ultima partita; preleva PlayerID dal campo testuale dell'interfaccia; preleva lo Score e le posizioni di Snake e Apple dalla classe **GameState**; ripristina i vecchi valori di PlayerID e Score nei campi dell'interfaccia; invoca il ricalcolo delle posizioni di Snake e Apple

Classe **LeaderboardArchive**: inserisce un nuovo record in archivio; restituisce i migliori record globali di un determinato periodo (periodo e numero di record mostrati sono configurabili)

Classe **GUINavigationEvent**: contiene le informazioni di un evento del log; si serializza in XML; invia l'evento di log in XML al server **GUINavigationLog**

Classe **GUINavigationLog**: (server) riceve un evento di log XML; invoca la validazione dell'evento di log tramite XML Schema, aggiunge la riga XML al file di log in modo incrementale

Classe **SnakeGame**: inizializza la GUI; crea un oggetto **Playground**; crea un oggetto **SnakeController**; aggiorna la GUI;

Classe **SnakeController**: controller applicativo, risponde agli eventi invocando le opportune classi;

Classe **Playground**: rappresenta graficamente l'area di gioco; mostra a schermo gli oggetti di gioco basandosi sulle posizioni logiche passate come parametri dal controller applicativo;

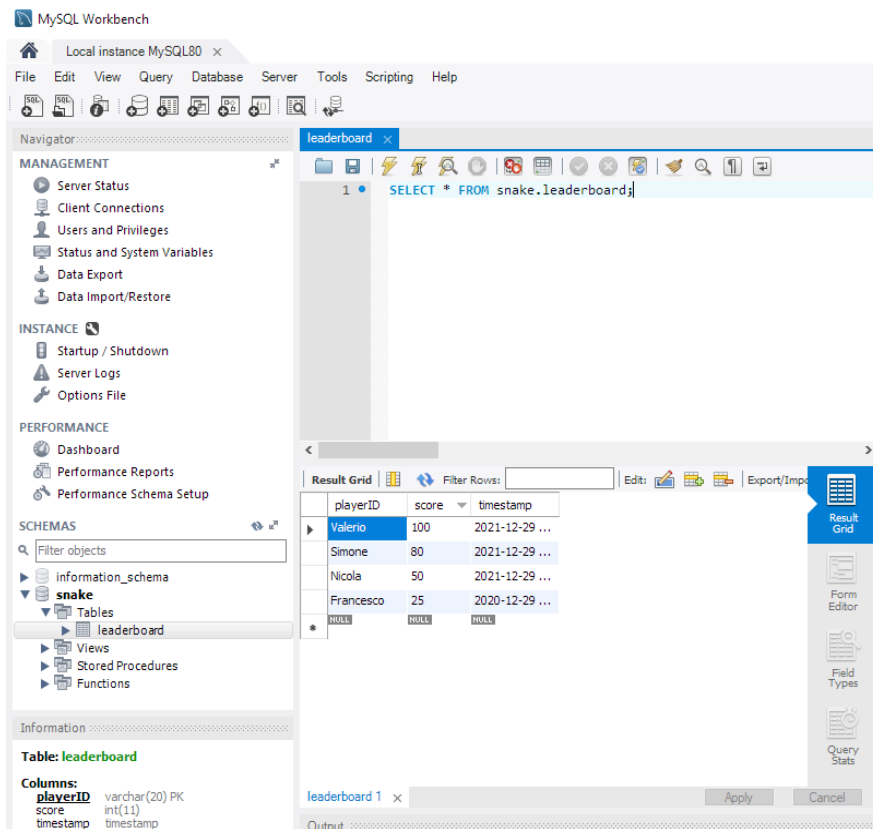
Classe **GameState**: contiene lo stato di gioco; controlla gli eventi “mela mangiata” e “gameover”; crea gli oggetti di gioco **Snake**, apple e obstacles;

Classe **Snake**: composto da un array di posizioni **GameObjectPosition**, rappresenta logicamente il serpente; trasla la testa nella direzione indicata, assegna le nuove posizioni del corpo seguendo la scia tracciata dalla testa; aumenta la propria lunghezza;

Classe **GameObjectPosition**: contiene le informazioni sulla posizione logica di un oggetto di gioco

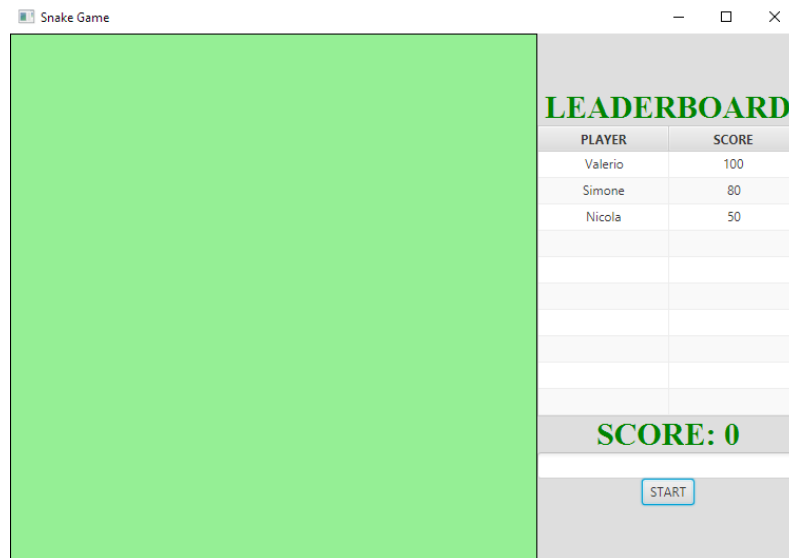
Manuale Utente

Cache locale e log remoto vuoti. Stato iniziale del database e del file di configurazione:

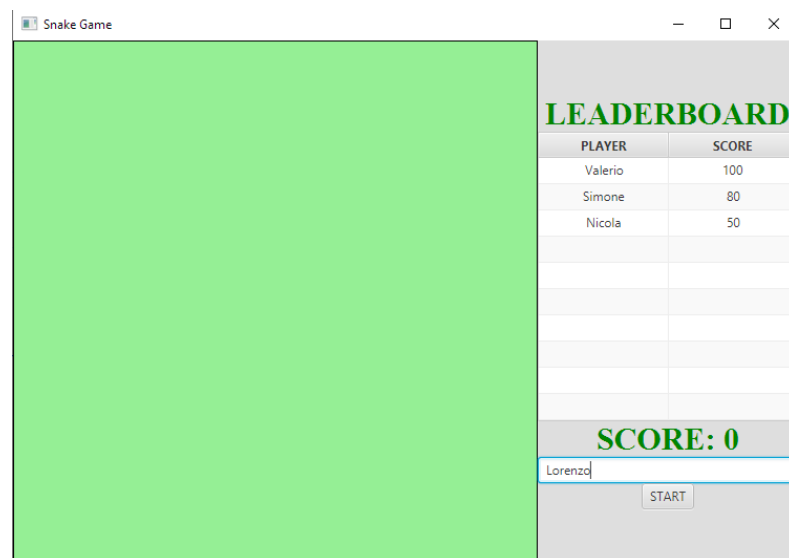


```
<?xml version="1.0" encoding="UTF-8"?>
<!-- config.xml -->
<ConfigurationParameters>
  <game>
    <GAME_DIFFICULTY>3</GAME_DIFFICULTY>
    <OBSTACLES>false</OBSTACLES>
    <PLAYGROUND_WIDTH>25</PLAYGROUND_WIDTH>
    <PLAYGROUND_HEIGHT>25</PLAYGROUND_HEIGHT>
  </game>
  <style>
    <BLOCK_SIZE>20</BLOCK_SIZE>
    <PLAYGROUND_COLOR>lightgreen</PLAYGROUND_COLOR>
    <OBSTACLE_COLOR>gray</OBSTACLE_COLOR>
    <SNAKE_BORDER_COLOR>blue</SNAKE_BORDER_COLOR>
    <SNAKE_HEAD_COLOR>cornflowerblue</SNAKE_HEAD_COLOR>
    <SNAKE_BODY_COLOR>skyblue</SNAKE_BODY_COLOR>
    <FONT>Serif</FONT>
    <FONT_SIZE unit="px">30</FONT_SIZE>
  </style>
  <keybinding>
    <KEY_UP></KEY_UP>
    <KEY_RIGHT></KEY_RIGHT>
    <KEY_DOWN></KEY_DOWN>
    <KEY_LEFT></KEY_LEFT>
  </keybinding>
  <leaderboard>
    <LEADERBOARD_SIZE>10</LEADERBOARD_SIZE>
    <LEADERBOARD_DAYS>7</LEADERBOARD_DAYS>
  </leaderboard>
  <technical>
    <CLIENT_IP>localhost</CLIENT_IP>
    <SERVER_IP>localhost</SERVER_IP>
    <SERVER_PORT>8080</SERVER_PORT>
    <DB_IP>localhost</DB_IP>
    <DB_PORT>3306</DB_PORT>
    <DB_USER>root</DB_USER>
    <DB_PASSWORD></DB_PASSWORD>
  </technical>
</ConfigurationParameters>
```

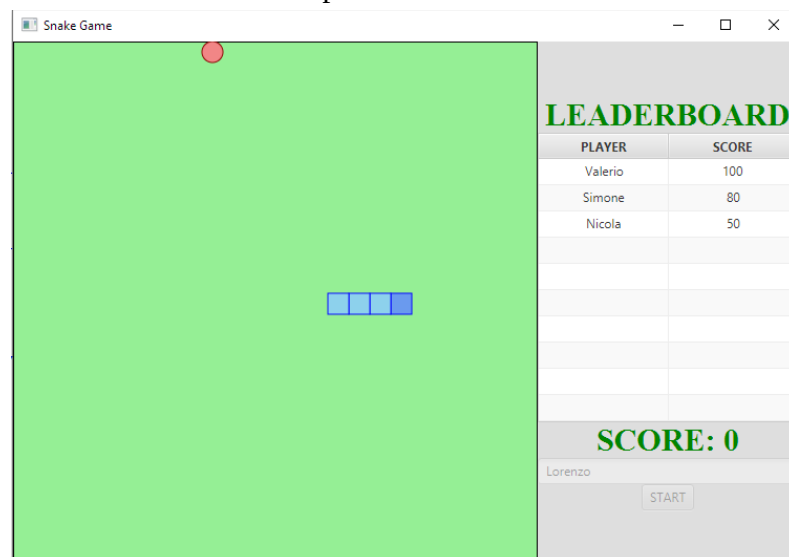
- 1) Una volta avviata l'applicazione, si presenterà la seguente interfaccia:



- 2) Inserire nome del giocatore nel campo testuale

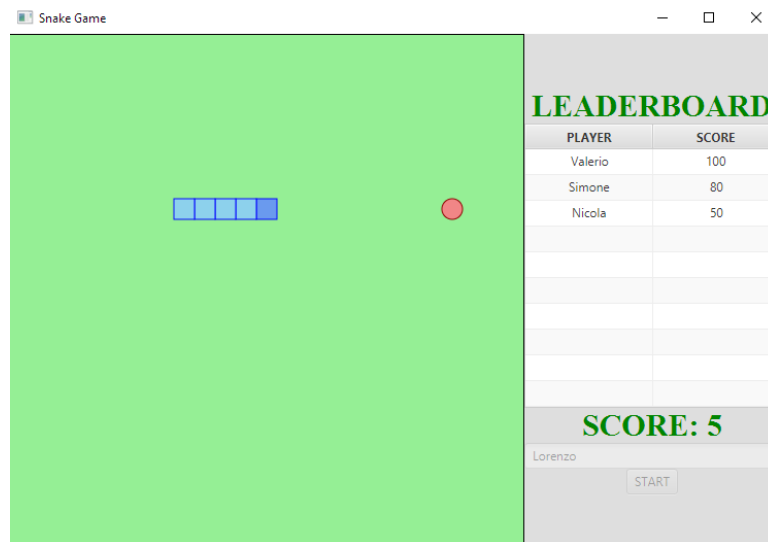


- 3) Premere START: il bottone e il campo testuale si disattiveranno e inizierà una nuova partita

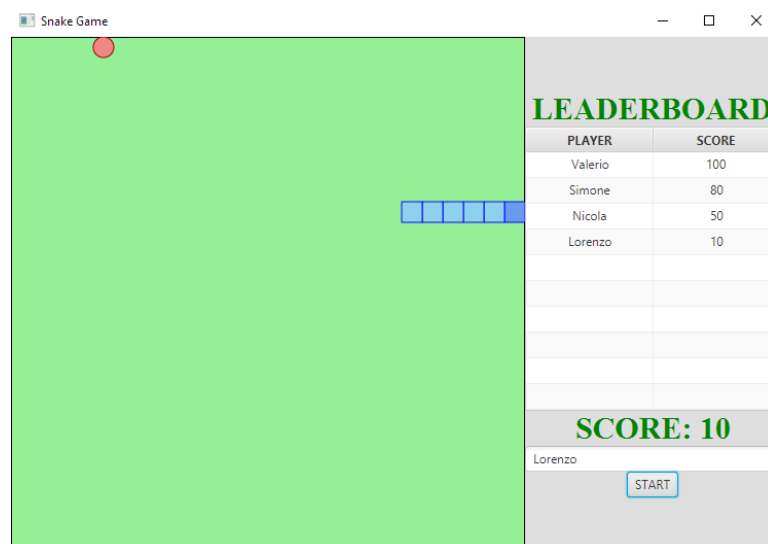


- 4) Direzionare il serpente con i tasti freccia della tastiera cercando di mangiare quante più mele possibili senza sbattere contro gli ostacoli.

Di seguito, presentato un possibile scenario di gioco:



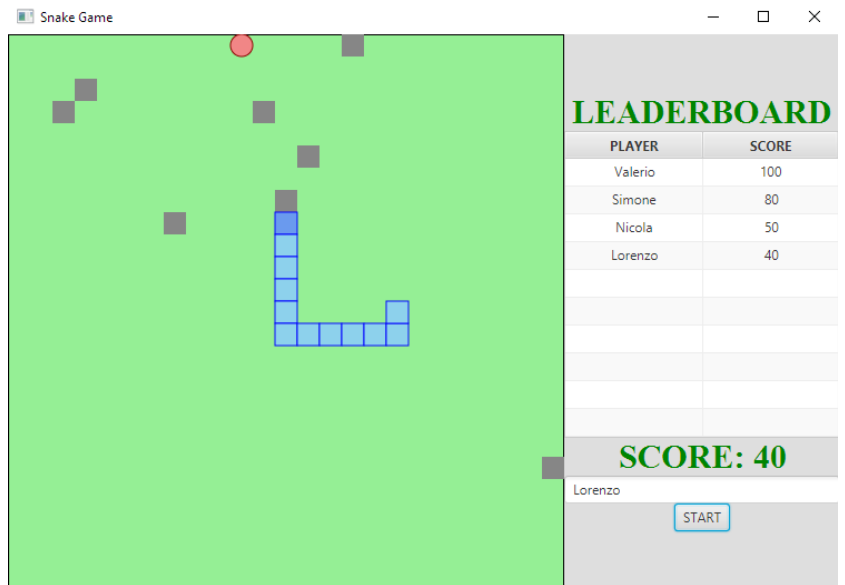
prima mela mangiata:
il serpente si allunga, lo score aumenta di 5 ed appare una nuova mela



dopo aver mangiato la seconda mela, il serpente sbatte contro un ostacolo:
il gioco termina e la leaderboard viene aggiornata

- 5) Se si è in cerca di sfida, è possibile attivare la modalità ad ostacoli scrivendo “true” nel campo OBSTACLES del file di configurazione

```
config - Blocco note di Windows
File Modifica Formato Visualizza ?
<?xml version="1.0" encoding="UTF-8"?>
<!-- config.xml -->
<ConfigurationParameters>
  <game>
    <GAME_DIFFICULTY>3</GAME_DIFFICULTY>
    <OBSTACLES>true</OBSTACLES>
    <PLAYGROUND_WIDTH>25</PLAYGROUND_WIDTH>
    <PLAYGROUND_HEIGHT>25</PLAYGROUND_HEIGHT>
  </game>
  <style>
    <BLOCK_SIZE>20</BLOCK_SIZE>
    <PLAYGROUND_COLOR>lightgreen</PLAYGROUND_COLOR>
    <OBSTACLE_COLOR>gray</OBSTACLE_COLOR>
    <SNAKE_BORDER_COLOR>blue</SNAKE_BORDER_COLOR>
    <SNAKE_HEAD_COLOR>cornflowerblue</SNAKE_HEAD_COLOR>
    <SNAKE_BODY_COLOR>skyblue</SNAKE_BODY_COLOR>
    <FONT>Serif</FONT>
    <FONT_SIZE unit="px">30</FONT_SIZE>
  </style>
  <keybinding>
    <KEY_UP></KEY_UP>
    <KEY_RIGHT></KEY_RIGHT>
    <KEY_DOWN></KEY_DOWN>
    <KEY_LEFT></KEY_LEFT>
  </keybinding>
  <leaderboard>
    <LEADERBOARD_SIZE>10</LEADERBOARD_SIZE>
    <LEADERBOARD_DAYS>7</LEADERBOARD_DAYS>
  </leaderboard>
  <technical>
    <CLIENT_IP>localhost</CLIENT_IP>
    <SERVER_IP>localhost</SERVER_IP>
    <SERVER_PORT>8080</SERVER_PORT>
    <DB_IP>localhost</DB_IP>
    <DB_PORT>3306</DB_PORT>
    <DB_USER>root</DB_USER>
    <DB_PASSWORD></DB_PASSWORD>
  </technical>
</ConfigurationParameters>
```



Riportate di seguito le modifiche che il sistema apporta al log remoto e al database al gameover:

```
Output X
SnakeServer (run) X SnakeGame (jfxsa-run) X
<GUINavigationEvent appName="Snake">
  <clientIP>localhost</clientIP>
  <timestamp>2022-01-02T12:49:07</timestamp>
  <eventName>AVVIO</eventName>
</GUINavigationEvent>
<GUINavigationEvent appName="Snake">
  <clientIP>localhost</clientIP>
  <timestamp>2022-01-02T12:49:11</timestamp>
  <eventName>START</eventName>
</GUINavigationEvent>
<GUINavigationEvent appName="Snake">
  <clientIP>localhost</clientIP>
  <timestamp>2022-01-02T12:53:52</timestamp>
  <eventName>TERMINE</eventName>
</GUINavigationEvent>
```

navLog - Blocco note di Windows

File Modifica Formato Visualizza ?

```
<GUINavigationEvent appName="Snake">
  <clientIP>localhost</clientIP>
  <timestamp>2022-01-02T12:49:07</timestamp>
  <eventName>AVVIO</eventName>
</GUINavigationEvent>
<GUINavigationEvent appName="Snake">
  <clientIP>localhost</clientIP>
  <timestamp>2022-01-02T12:49:11</timestamp>
  <eventName>START</eventName>
</GUINavigationEvent>
<GUINavigationEvent appName="Snake">
  <clientIP>localhost</clientIP>
  <timestamp>2022-01-02T12:53:52</timestamp>
  <eventName>TERMINE</eventName>
</GUINavigationEvent>
```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

- information_schema
- snake
 - Tables
 - leaderboard
 - Views
 - Stored Procedures
 - Functions

Information

Table: leaderboard

Columns:

- playerID varchar(20) PK
- score int(11)
- timestamp timestamp

leaderboard x

SELECT * FROM snake.leaderboard;

playerID	score	timestamp
Valerio	100	2021-12-29 ...
Simone	80	2021-12-29 ...
Nicola	50	2021-12-29 ...
Francesco	25	2020-12-29 ...
Lorenzo	10	2022-01-02 ...
NULL	NULL	NULL

Result Grid

Form Editor

Field Types

Query Stats

Context Help Snippets