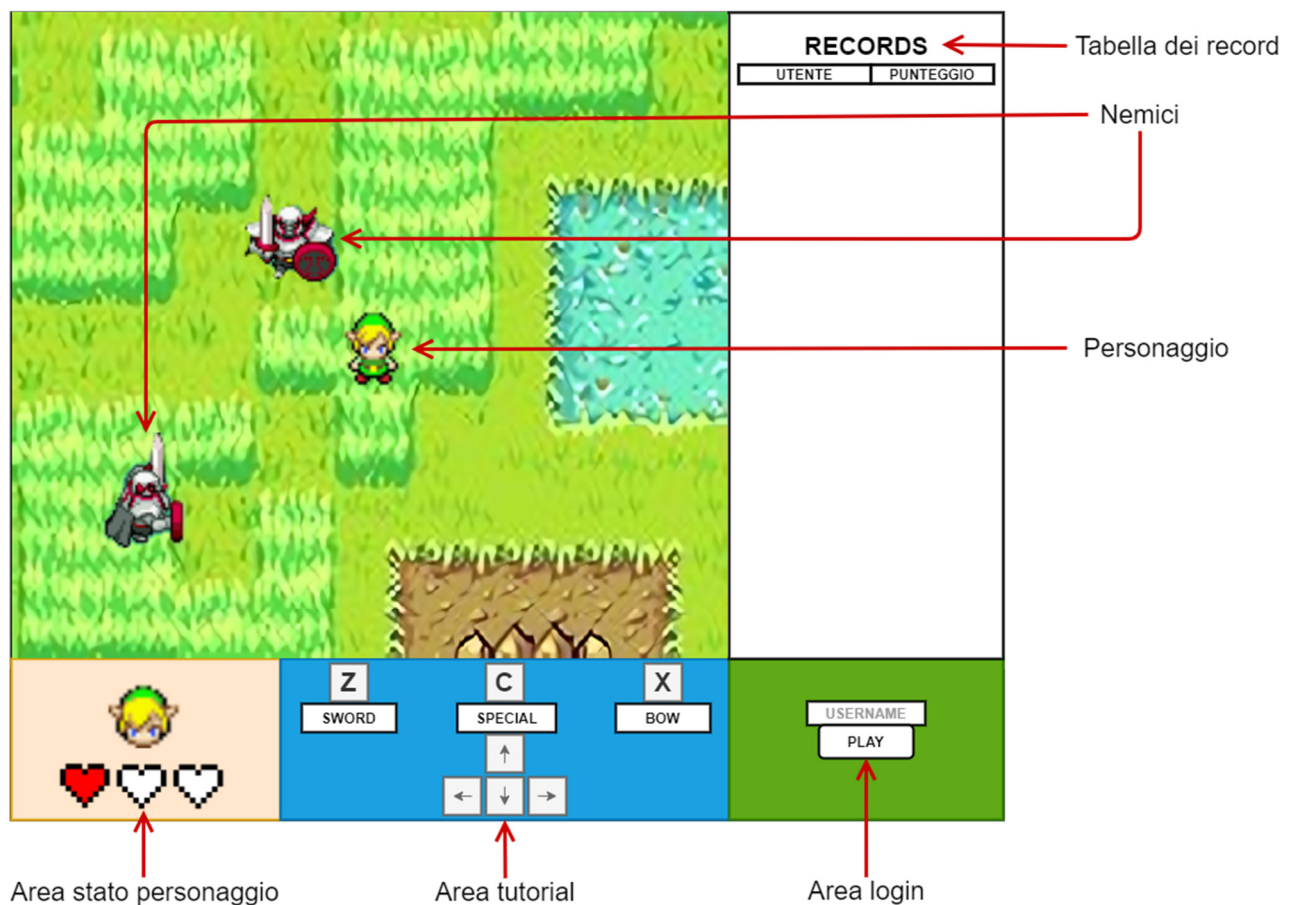


Analisi



Casi d'uso:

1. All'apertura dell'applicazione, l'utente inserisce il proprio username nell'area di testo "username"
2. L'utente preme il tasto "PLAY"
3. Avvio di una partita
 - a. Vengono mostrati a video Link (il personaggio del giocatore) ed il primo nemico
 - b. Viene aggiunta all'area di testo "BOW" il numero di frecce disponibili, nel seguente formato "BOW: " + numero frecce
 - c. Vengono riempiti i 3 cuori rappresentanti la vita del giocatore
4. Durante una partita, l'utente preme un tasto di movimento
 - a. Link eseguirà un'animazione di movimento verso la casella adiacente puntata
5. IF Durante una partita, l'utente preme il tasto SWORD
 - a. Link eseguirà un'animazione di attacco nella direzione in cui si trova, eliminando il nemico presente nella casella adiacente in quella direzione
6. IF Durante una partita, l'utente preme il tasto SPECIAL
 - a. Link eseguirà l'animazione di un attacco rotante, eliminando i nemici presenti in tutte le caselle adiacenti a quella di Link
 - b. L'attacco SPECIAL non sarà più disponibile per un determinato numero di turni, chiamato "Cooldown"
 - c. L'area di testo "SPECIAL" viene colorata di rosso, mentre viene aggiunto il numero di turni rimanenti per il prossimo utilizzo, nel seguente formato "SPECIAL: " + cooldown
7. IF Durante una partita, l'utente preme il tasto BOW

- a. Link eseguirà l'animazione di un tiro con l'arco nella direzione in cui si trova, eliminando il primo nemico presente nella semiretta puntata
 - b. Viene decrementata di un'unità in numero di frecce disponibili e l'area di testo "BOW" viene modificata, aggiornando il numero di frecce disponibili
 - c. IF numero di frecce = 0
 - i. L'area di testo "BOW" viene colorata di rosso
 - ii. Non è più possibile utilizzare questo attacco
8. Durante una partita, nel turno dell'utente
 - a. L'utente ha a disposizione un determinato numero di input
 - b. IF contatore input = numero input permessi
 - i. Passa il turno agli avversari
9. Durante una partita, nel turno degli avversari
 - a. FOR EACH Nemico
 - i. Esegui due azioni, tra movimento verso Link o attacco
10. Durante una partita, quando un attacco colpisce Link
 - a. Decrementa i cuori di uno, mostrando un cuore vuoto al posto di uno rosso
 - b. IF cuori = 0
 - i. Termina la partita
11. Termina la partita
 - a. Viene caricato il Record all'interno del database e mostrato nella tabella "Records" al lato

FILE DI CONFIGURAZIONE LOCALE IN XML

All'avvio il Gioco legge dal file di configurazione, modificabile dall'utente, i seguenti dati:

- Tasti associati ai comandi di gioco, ossia movimento (su, giù, destra, sinistra), Sword, Special, Bow
- Indirizzo e porta del server di log
- Username, password, Indirizzo e porta del database
- Numero di righe visibili nella tabella Records

All'avvio dell'applicazione, viene caricato il file e utilizzate le impostazioni che contiene.

CACHE LOCALE DEGLI INPUT

Se il Gioco viene chiuso durante una partita non terminata, salva su file binario lo stato del gioco in quel momento, caratterizzato da:

- Posizione e direzione del personaggio
- Posizione e direzione dei nemici
- Vite del personaggio
- Numero di frecce disponibili
- Disponibilità dell'attacco special e cooldown attuale (se la special è disponibile cooldown = 0)
- Punteggio attuale
- Appartenenza del turno
- Numero di input usati all'interno del turno corrente

All'avvio il Gioco carica dal file binario i suddetti dati, e li utilizza per riprendere la partita nello stato in cui è stata lasciata. Nel caso in cui la chiusura dell'applicazione avviene durante il turno nemico, la partita riprenderà passando il turno al giocatore.

SPECIFICHE DATABASE

Il Gioco archivia i seguenti dati all'interno del Database:

- Username, Score

FILE DI LOG REMOTO IN XML

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

- Avvio dell'applicazione
- Chiusura dell'applicazione
- Pressione del pulsante di Login
 - IF Area di testo "USERNAME" riempito ("Username inserito")
- Avvio della partita
- Termine della partita
- Caricamento delle associazioni dei tasti dal file di impostazione, includendo nel messaggio le associazioni scelte dall'utente. Viene registrato l'evento nel log solo se le associazioni sono diverse da quelle default.
- Pressione tasto associato ad un comando

La riga di log contiene:

- Nome dell'applicazione
- Indirizzo IP del client
- Timestamp dell'evento
- Descrizione dell'evento

Il server di log, ricevuto un messaggio, ne valida il contenuto tramite schema xsd e se la validazione ha successo aggiunge il messaggio all'apposito file di log.

Progetto

Documentazione delle classi

Classi di logica

Zelda

Classe applicazione. Istanza l'interfaccia, carica le impostazioni e istanzia di conseguenza gli oggetti di modello che costituiscono l'applicazione oltre ad inizializzare le componenti per il log e le interazioni con il database.

Gestisce gli eventi quali la pressione del pulsante di login, l'utilizzo del campo username e la pressione dei tasti di gioco, in collaborazione con le altre classi.

InterfaceBuilder

Classe dedicata alla costruzione e stilizzazione dell'interfaccia grafica

ServerLogXml

Classe main del server di log. Riceve dal client di gioco gli eventi per il log in formato xml, li valida e li aggiunge al file di log.

GameModel

Classe che costituisce il modello del gioco, riceve i comandi da eseguire da Zelda (ovvero dal giocatore) e li applica tramite la classe GameCharacter. Mantiene lo stato della partita in termini di informazioni riguardo il posizionamento di nemici e personaggio, e di informazioni riguardo l'andamento della stessa, come il punteggio e le vite del giocatore. Gli elementi grafici di gioco invece vengono acceduti tramite GameView.

GameCharacter

Classe che definisce le informazioni e le possibilità di movimento e di attacco del personaggio controllato dal giocatore.

GameEnemy

Classe che definisce le informazioni e le possibilità di movimento e di attacco di un nemico, così come una elementare intelligenza artificiale per poter scegliere le migliori opzioni di movimento per raggiungere il giocatore e attaccare quando possibile.

GameTile

Classe che gestisce le informazioni riguardo un elemento della matrice costituente il campo di gioco.

TurnHandler

Tread dedicato alla gestione dei turni, e al passaggio da un nemico all'altro per la rispettiva azione automatica.

GameView

Classe che gestisce le componenti visuali del gioco. Istanza e posiziona gli elementi grafici del gioco, fornisce metodi per accedervi o indicarne posizioni specifiche. Fornisce metodi per avviare le animazioni, e aggiorna lo stato delle stesse, per indicare se sono terminate o meno, per il controllo del proseguimento del gioco.

GraficTile

Elemento grafico che rappresenta una casella della matrice costituente il campo di gioco visibile.

Espone metodi per l'aggiornamento grafico in base alle informazioni date dal rispettivo GameTile contenuto nel GameModel.

DBManager

Classe contenente i metodi per accedere al database, in particolare per le operazioni di inserimento, lettura ed eliminazione dei Record presenti nello schema Ranking.

Classi statiche

LocalCacheOperations

Classe statica che contiene metodi per salvare e recuperare la cache locale.

Accede alle classi applicative per salvare e ripristinare lo stato della partita.

EventLoggerXML

Classe statica che contiene i metodi per creare un oggetto EventLogXML a partire dalla descrizione dell'evento, serializzarlo e inviarlo via socket al server di log.

Valida il messaggio XML tramite XSD prima dell'invio, inviando solo i messaggi che superano la validazione.

ReaderSettingsXML

Classe statica che contiene i metodi per leggere il file di impostazione xml, validarne il contenuto e ottenere l'oggetto SettingsXML da utilizzare nell'applicazione. Nel caso la validazione del contenuto fallisce, vengono restituite impostazioni di default mentre il file xml viene sovrascritto con queste.

GameUtils

Classe statica contenente costanti e metodi di utilità per le classi di logica dedicate alla gestione della partita.

Strutture dati

Record

Struttura dati dedicata all'archiviazione di un Record conseguito dal giocatore corrente.

Viene utilizzato come modello per l'estrazione o registrazione di dati presenti nel Database

EventLogXML

Classe che contiene le informazioni di log da inviare via socket al server di log.

Richiede come parametro la descrizione dell'evento, mentre ricava le altre informazioni (indirizzo IP e timestamp) alla creazione.

Viene creato, serializzato e inviato via socket dalla classe EventLoggerXML.

CacheData, GameCacheData, ElementCacheData

Strutture dati serializzabili utilizzate da LocalCacheOperations per salvare e caricare lo stato dell'applicazione via file binario.

SettingsXML, KeyAssociation, ServerAddress

Strutture dati utilizzate per la deserializzazione delle impostazioni xml, e per serializzazione in caso di scrittura delle impostazioni default.

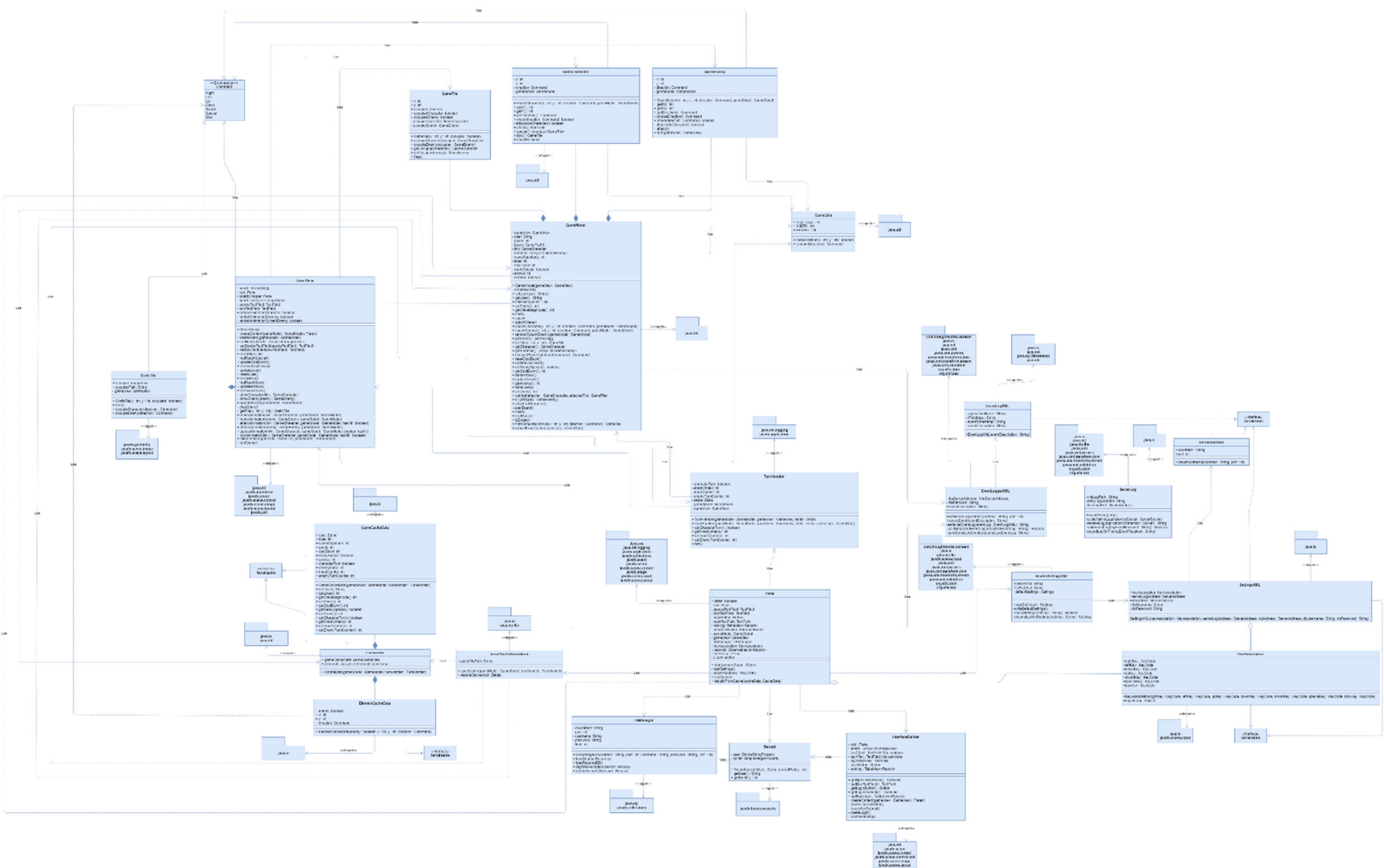
All'avvio Zelda ottiene l'oggetto SettingsXML da ReaderSettingsXML, e con questo oggetto vengono configurati i vari componenti dell'applicazione.

Enumerazioni

Command

Enumerazione che indica il comando di gioco inserito dall'utente.

Schema UML



Manuale Utente

Impostazioni su file xml

All'avvio dell'applicazione vengono caricate le impostazioni dal file "settings.xml".

In assenza del file, o se questo ha una sintassi scorretta, l'applicazione genera e utilizza un file di impostazione di default.

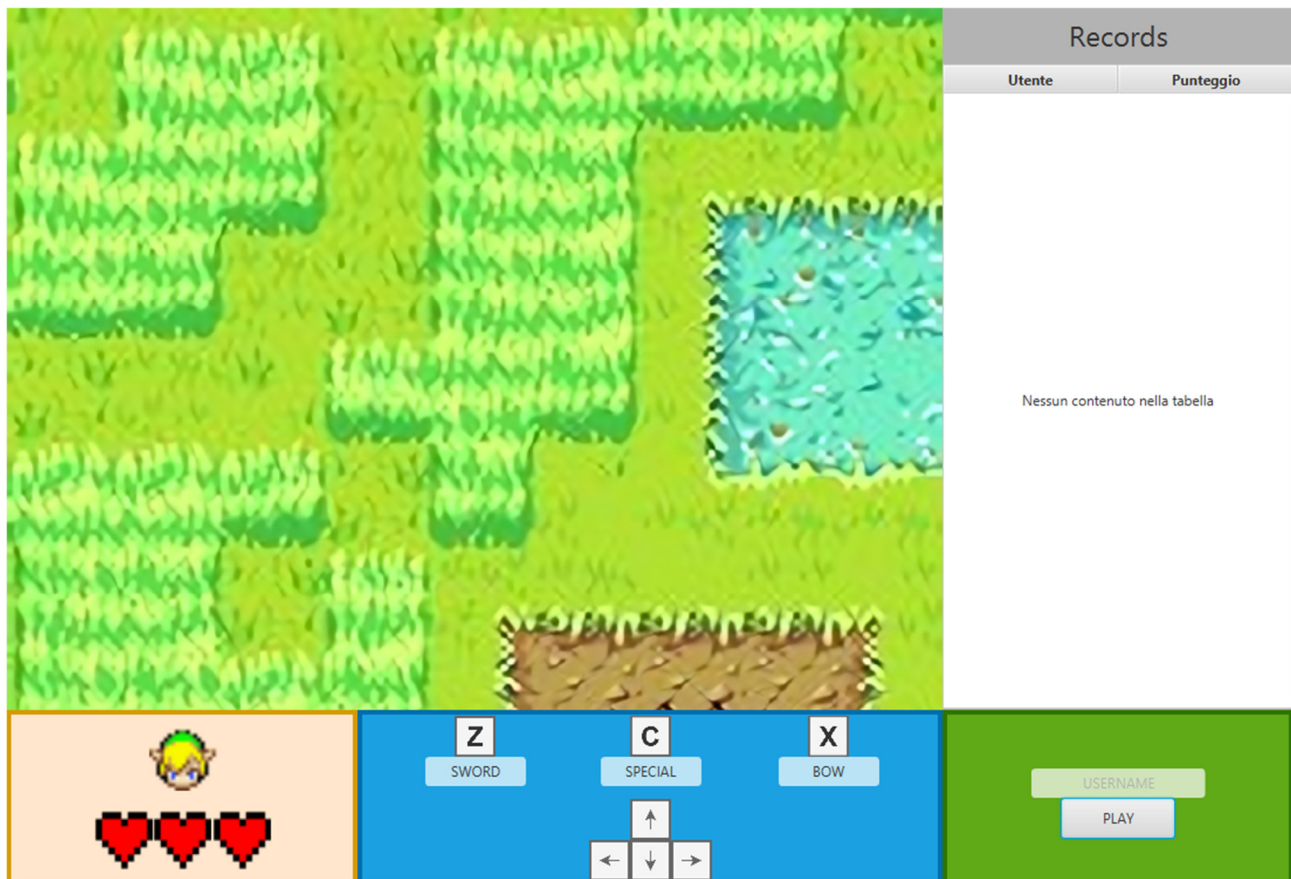
Le impostazioni sono:

- Associazioni dei tasti di gioco (right, left, up, down, sword, special, bow)
- Indirizzo IP e porta del server di log
- Indirizzo IP, porta, username e password del database
- Numero di righe visibili nella tabella Records

```
1 <zelda.SettingsXML>
2   <keyAssociation>
3     <rightKey>RIGHT</rightKey>
4     <leftKey>LEFT</leftKey>
5     <upKey>UP</upKey>
6     <downKey>DOWN</downKey>
7     <swordKey>Z</swordKey>
8     <specialKey>C</specialKey>
9     <bowKey>X</bowKey>
10  </keyAssociation>
11  <serverLogAddress>
12    <IPAddress>localhost</IPAddress>
13    <port>9200</port>
14  </serverLogAddress>
15  <dbAddress>
16    <IPAddress>localhost</IPAddress>
17    <port>3306</port>
18  </dbAddress>
19  <dbUsername>root</dbUsername>
20  <dbPassword></dbPassword>
21  <dbLimit>5</dbLimit>
22 </zelda.SettingsXML>
```

Avvio dell'applicazione

Una volta avviata l'applicazione la schermata che viene mostrata all'Utente è la seguente:

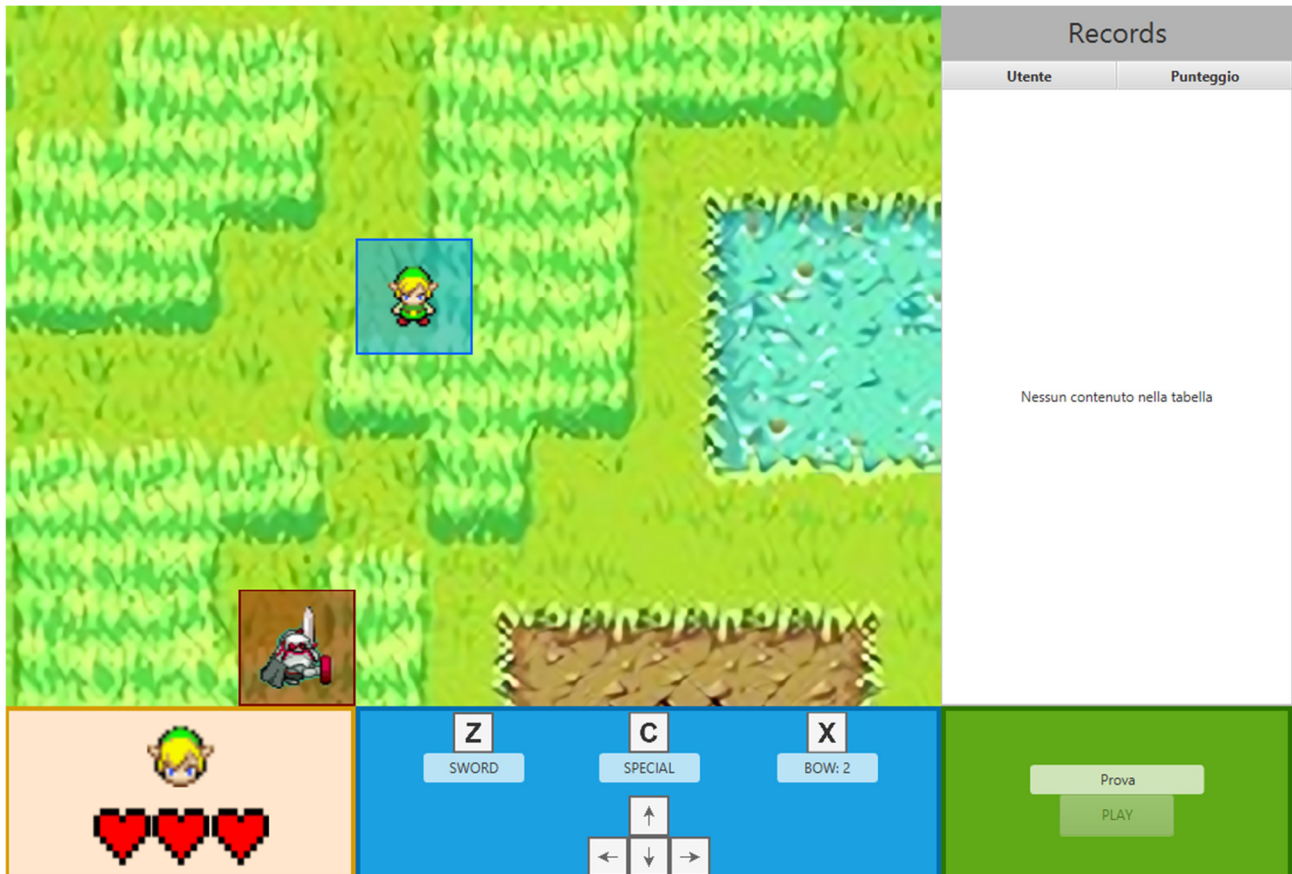


L'interfaccia è suddivisa in 5 sezioni:

1. In alto a sinistra è collocata l'area di gioco.
2. In alto a destra vi è una tabella dei Record, caricati dal Database.
3. In basso a sinistra vi è lo stato del giocatore, che mostra la vita sotto forma di Cuori pieni.
4. In basso al centro vengono indicati i tasti default per le azioni che può compiere il giocatore, e la disponibilità dell'attacco speciale e delle frecce dell'arco durante la partita.
5. In basso a destra vi è il campo d'inserimento per lo username ed il bottone per l'avvio della partita.

Avvio della partita

Per avviare la partita è necessario inserire il proprio username nel campo apposito e premere il bottone “PLAY”. Se il campo non risulterà vuoto al momento della pressione, il bottone verrà disabilitato e comincerà la partita, con l’apparizione del personaggio controllato dal giocatore, in una casella predefinita, e del primo nemico, in una casella libera casuale.



Durante la partita è possibile controllare il personaggio utilizzando i tasti indicati nel file di settings.xml (che sono quelli indicati nella sezione blu dell’interfaccia se non sono stati cambiati).

Il gioco alterna turni dedicati alle azioni del giocatore a turni dedicati ai comportamenti automatici dei nemici. Nel turno del giocatore gli input concessi, tra movimento e attacchi è pari a 3, nel turno dei nemici invece, ogni avversario eseguirà due azioni, tra il movimento verso il giocatore e, se possibile, l’attacco diretto da una casella adiacente.



Scopo del gioco è sconfiggere più nemici possibile, che compaiono tramite ondate di dimensione crescente, finché non si perdono tutte le vite a disposizione per via dei colpi subiti.

Movimento



Premendo uno dei tasti di movimento il personaggio si muoverà verso la casella adiacente nella direzione indicata, mantenendo poi tale direzione da fermo.



Attacco



Premendo il tasto di attacco con la spada il personaggio farà un'animazione ed eliminerà, se presente, il nemico sulla casella adiacente puntata.

Special



Premendo il tasto per l'attacco speciale il personaggio farà un'animazione ed eliminerà, se presenti, i nemici su tutte le caselle adiacenti.

Dopo l'esecuzione della mossa speciale, non sarà disponibile per un determinato numero di turni, chiamato "cooldown". Il tempo di attesa rimanente per poterlo riutilizzare viene mostrato ed aggiornato nella sezione blu, sul campo "SPECIAL". Il colore rosso del campo indica che la mossa non è disponibile.



Bow



Premendo il tasto per l'attacco con l'arco il personaggio farà un'animazione ed eliminerà, se presente, il primo nemico presente sull'asse puntato.

Termine della partita

La partita termina quando le vite di un giocatore sono a 0, ovvero quando tutti i cuori della sezione arancione sono vuoti.

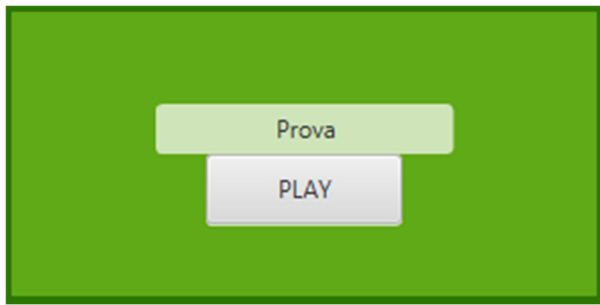


Viene quindi inserito il nuovo record all'interno del database ed aggiornata la tabella.

| | user | points |
|---|-------|--------|
| ▶ | Prova | 70 |

| Records | |
|---------|-----------|
| Utente | Punteggio |
| Prova | 70 |
| | |
| | |
| | |

Poi viene riabilitato il bottone ed il campo Username per poterlo modificare, ed avviare una nuova partita.



Interruzione e recupero di una partita

Se durante una partita l'applicazione viene chiusa, questo salva su un file di cache locale lo stato della partita.

Quando l'applicazione viene aperta, se il file di cache locale non risulta vuoto, avvia la procedura di recupero della partita

Log dell'uso dell'applicazione

Il client invia dei log riguardo l'uso che l'utente fa dell'interfaccia e delle funzioni del gioco, allo scopo di raccolta di dati di utilizzo che possono aiutare a migliorare l'applicazione. I log sono raccolti da un server il cui indirizzo è indicato nelle impostazioni, che ha il compito di validare i log ricevuti e se validi aggiungerli ad un file di log cumulativo.

I log vengono registrati per:

- Avvio dell'applicazione
- Chiusura dell'applicazione
- Pressione del pulsante di Login
 - IF Area di testo "USERNAME" riempito ("Username inserito")
- Avvio della partita
- Termine della partita
- Caricamento delle associazioni dei tasti dal file di impostazione, includendo nel messaggio le associazioni scelte dall'utente. Viene registrato l'evento nel log solo se le associazioni sono diverse da quelle default.
- Pressione tasto associato ad un comando

Un esempio di contenuto del file di log è il seguente:

```

1 <zelda.EventLogXML>
2 <applicationName>Zelda</applicationName>
3 <IPAddress>192.168.1.5</IPAddress>
4 <eventTimestamp>2021-09-10 23:41:04.662</eventTimestamp>
5 <eventDescription>Applicazione avviata</eventDescription>
6 </zelda.EventLogXML>
7
8 <zelda.EventLogXML>
9 <applicationName>Zelda</applicationName>
10 <IPAddress>192.168.1.5</IPAddress>
11 <eventTimestamp>2021-09-10 23:41:04.674</eventTimestamp>
12 <eventDescription>Caricate associazioni tasti personalizzate. Destra: Right Sinistra: Left Su: Up Giù: Down Sword: A Special: S Bow: D</eventDescription>
13 </zelda.EventLogXML>
14
15 <zelda.EventLogXML>
16 <applicationName>Zelda</applicationName>
17 <IPAddress>192.168.1.5</IPAddress>
18 <eventTimestamp>2021-09-10 23:41:12.203</eventTimestamp>
19 <eventDescription>Pulsante premuto</eventDescription>
20 </zelda.EventLogXML>
21
22 <zelda.EventLogXML>
23 <applicationName>Zelda</applicationName>
24 <IPAddress>192.168.1.5</IPAddress>
25 <eventTimestamp>2021-09-10 23:41:12.206</eventTimestamp>
26 <eventDescription>Username inserito</eventDescription>
27 </zelda.EventLogXML>
28
29 <zelda.EventLogXML>
30 <applicationName>Zelda</applicationName>
31 <IPAddress>192.168.1.5</IPAddress>
32 <eventTimestamp>2021-09-10 23:41:12.212</eventTimestamp>
33 <eventDescription>Partita avviata</eventDescription>
34 </zelda.EventLogXML>
35
36 <zelda.EventLogXML>
37 <applicationName>Zelda</applicationName>
38 <IPAddress>192.168.1.5</IPAddress>
39 <eventTimestamp>2021-09-10 23:41:19.056</eventTimestamp>
40 <eventDescription>Premuto tasto ok: Left</eventDescription>
41 </zelda.EventLogXML>
42
43 <zelda.EventLogXML>
44 <applicationName>Zelda</applicationName>
45 <IPAddress>192.168.1.5</IPAddress>
46 <eventTimestamp>2021-09-10 23:41:24.979</eventTimestamp>
47 <eventDescription>Premuto tasto ok: S</eventDescription>
48 </zelda.EventLogXML>
49
50 <zelda.EventLogXML>
51 <applicationName>Zelda</applicationName>
52 <IPAddress>192.168.1.5</IPAddress>
53 <eventTimestamp>2021-09-10 23:41:26.412</eventTimestamp>
54 <eventDescription>Premuto tasto ok: D</eventDescription>
55 </zelda.EventLogXML>
56
57 <zelda.EventLogXML>
58 <applicationName>Zelda</applicationName>
59 <IPAddress>192.168.1.5</IPAddress>
60 <eventTimestamp>2021-09-10 23:41:30.052</eventTimestamp>
61 <eventDescription>Premuto tasto ok: Up</eventDescription>
62 </zelda.EventLogXML>
63
64 <zelda.EventLogXML>
65 <applicationName>Zelda</applicationName>
66 <IPAddress>192.168.1.5</IPAddress>
67 <eventTimestamp>2021-09-10 23:41:30.779</eventTimestamp>
68 <eventDescription>Premuto tasto ok: A</eventDescription>
69 </zelda.EventLogXML>
70
71 <zelda.EventLogXML>
72 <applicationName>Zelda</applicationName>
73 <IPAddress>192.168.1.5</IPAddress>
74 <eventTimestamp>2021-09-10 23:41:32.056</eventTimestamp>
75 <eventDescription>Premuto tasto ok: Left</eventDescription>
76 </zelda.EventLogXML>
77
78 <zelda.EventLogXML>
79 <applicationName>Zelda</applicationName>
80 <IPAddress>192.168.1.5</IPAddress>
81 <eventTimestamp>2021-09-10 23:41:33.257</eventTimestamp>
82 <eventDescription>Partita terminata</eventDescription>
83 </zelda.EventLogXML>
84
85 <zelda.EventLogXML>
86 <applicationName>Zelda</applicationName>
87 <IPAddress>192.168.1.5</IPAddress>
88 <eventTimestamp>2021-09-10 23:41:36.744</eventTimestamp>
89 <eventDescription>Applicazione chiusa</eventDescription>
90 </zelda.EventLogXML>

```