

Documento di Analisi

- Studente: Tommaso Giorgi
- Matricola: 546921

Mockup:



Scenario d'uso

1. Il Giocatore inserisce un nickname qualsiasi nel campo di testo caratterizzato dal label Nickname
2. Il Giocatore seleziona la difficoltà cliccando sul pulsante "DIFFICOLTA"
 - 2.1 Il Giocatore seleziona come difficoltà "FACILE"
 - 2.2 Il Giocatore seleziona come difficoltà "INTERMEDIO"
 - 2.3 Il Giocatore seleziona come difficoltà "DIFFICILE"
3. Il giocatore clicca il pulsante "START" e il gioco ha inizio
4. Il sistema visualizza l'interfaccia di gioco
 - 4.1 Il sistema visualizza in alto a sinistra il punteggio attuale inizialmente posto a "00000000"
 - 4.2 Il sistema visualizza in alto al centro il tempo rimanente
 - 4.2.1 Il sistema visualizza un tempo rimanente pari a 150 se la difficoltà scelta è "FACILE"
 - 4.2.2 Il sistema visualizza un tempo rimanente pari a 120 se la difficoltà scelta è "INTERMEDIO"
 - 4.2.3 Il sistema visualizza un tempo rimanente pari a 90 se la difficoltà scelta è "DIFFICILE"

- 4.3 Il sistema visualizza in alto a destra il numero del livello ("lvl:01")
- 4.4 Il sistema visualizza l'immagine frontale di Bomberman nella casella 1-1
- 4.5 Il sistema visualizza randomicamente (ad eccezione delle caselle 1-1 | 1-2 | 2-1 e di quelle occupare da "rocce") un numero n di "massi" (in una di queste sarà presente la porta segreta)
 - 4.5.1 n vale 20 se la difficoltà scelta è "FACILE"
 - 4.5.2 n vale 30 se la difficoltà scelta è "INTERMEDIO"
 - 4.5.3 n vale 40 se la difficoltà scelta è "DIFFICILE"
- 4.6 Il sistema visualizza m nemici che iniziano a muoversi randomicamente nella mappa di gioco
 - 4.6.1 m vale 3 se la difficoltà scelta è "FACILE"
 - 4.6.2 m vale 5 se la difficoltà scelta è "INTERMEDIO"
 - 4.6.3 m vale 7 se la difficoltà scelta è "DIFFICILE"
- 4.7 Il sistema disabilita il bottone "START" e il campo di testo con label Nickname

5. FOR EACH tick

- 5.1 IF è trascorso 1 secondo, il sistema visualizza un tempo decrementato di 1
 - 5.1.1 IF il sistema visualizza un tempo pari a 0s, il sistema visualizza un messaggio di Game Over e l'interfaccia viene resettata (torno al punto 3)
- 5.2 IF Il giocatore preme il tasto della tastiera "D", il sistema visualizza Bomberman in una posizione spostata a destra rispetto alla precedente di qualche pixel (se possibile)
- 5.3 IF Il giocatore preme il tasto della tastiera "A", il sistema visualizza Bomberman in una posizione spostata a sinistra rispetto alla precedente di qualche pixel (se possibile)
- 5.4 IF Il giocatore preme il tasto della tastiera "S", il sistema visualizza Bomberman in una posizione spostata verso il basso rispetto alla precedente di qualche pixel (se possibile)
- 5.5 IF Il giocatore preme il tasto della tastiera "W", il sistema visualizza Bomberman in una posizione spostata verso l'alto rispetto alla precedente di qualche pixel (se possibile)
- 5.6 IF il giocatore preme il tasto della tastiera "SHIFT", il sistema visualizza una bomba nella posizione corrente di Bomberman
- 5.7 Il sistema visualizza i nemici in una posizione spostata rispetto alla precedente di qualche pixel in una delle quattro direzioni adiacenti possibili (basso-alto-destra-sinistra) randomicamente
 - 5.7.1 IF un nemico entra in contatto con Bomberman, Bomberman muore (viene rimosso)
 - 5.7.1.1 Il sistema visualizza l'immagine di una lapide nella posizione corrente di Bomberman
 - 5.7.1.2 Il sistema visualizza un messaggio di Game Over e l'interfaccia viene resettata (torno al punto 3)
- 5.8 IF il sistema visualizza una bomba e sono passati 1.6s la bomba esplode
 - 5.8.1 IF Bomberman è nel raggio di azione della bomba, Bomberman muore (viene rimosso)
 - 5.8.1.1 Il sistema visualizza l'immagine di una lapide nella posizione corrente di Bomberman
 - 5.8.1.2 Il Sistema visualizza un immagine di Game Over e l'interfaccia viene resettata (torno al punto 3)
 - 5.8.2 IF un nemico è nel raggio di azione della bomba, il nemico muore
 - 5.8.2.1 Il sistema non visualizza più il nemico
 - 5.8.2.2 Il sistema visualizza un punteggio incrementato di 2000
 - 5.8.3 IF un "masso" è nel raggio di azione della bomba, il "masso viene distrutto"
 - 5.8.3.1 Il sistema visualizza un punteggio incrementato di 100

5.8.3.2 IF il “masso” conteneva la porta segreta, il sistema la visualizza sostituendo la roccia con la porta stessa

5.8.3.3 IF il masso non conteneva la porta segreta, la roccia non viene più visualizzata

5.9 IF Bomberman è visualizzato sulla porta segreta

5.9.1 Il sistema visualizza un messaggio di vittoria con il tempo rimanente e il punteggio

5.9.2 Il sistema visualizza nella tabella “Leaderboard” il nickname con il quale è stato superato il livello, il punteggio ottenuto e la difficoltà scelta del livello.

5.9.3 L'interfaccia viene resettata (torno al punto 3)

File di Configurazione Locale XML:

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

- username, password, porta Dbms
- numero di righe da visualizzare nella tabella Leaderboard
- font, dimensioni, colore del background
- dimensioni (righe e colonne) della griglia di gioco
- l'indirizzo IP del client, l'indirizzo IP e la porta del server log

Cache locale degli Input:

Alla chiusura, il Sistema salva su file lo stato del gioco al momento della chiusura ovvero:

- la difficoltà attualmente selezionata
- il nickname attualmente inserito nel campo di testo con label Nickname

Archivio:

Il Sistema archivia i dati presenti nella Leaderboard ovvero:

- Nickname
- il valore dello Score a livello completato
- Difficoltà del livello completato

File di LOG Remoto in XML:

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

- Avvio dell'applicazione (“Apertura Applicazione”)
- Pressione del pulsante START (“Click Bottone Start”)
- Una bomba è stata posizionata (“Bomba Piazzata”)
- Un nemico è stato ucciso (“Nemico Ucciso”)
- Una porta segreta è stata rivelata (“Porta Segreta Trovata!”)
- Termine dell'applicazione (“Chiusura Applicazione”)

La riga di log contiene: nome dell'evento, indirizzo IP del client, data-ora corrente associata all'evento

Documento di Progetto

- Studente: Tommaso Giorgi

- Matricola: 546921

Classi front-end

- Classe **InterfacciaStatica**
 - Realizza l'interfaccia grafica statica
 - Posiziona i massi e i nemici casualmente ad inizio game
 - Resetta l'interfaccia a fine gioco
- Classe **InterfacciaDinamica**
 - Realizza l'interfaccia grafica dinamica
 - Contiene una AnchorPane che contiene tutti gli elementi dell'interfaccia
- Classe **Leaderboard**
 - Mostra i record del database ordinati per punteggio decrescente
- Classe **Masso**
 - Crea un oggetto di tipo Masso
- Classe **Roccia**
 - Crea un oggetto di tipo Roccia
- Classe **Bomb**
 - Crea un oggetto di tipo Bomb
- Classe **SecretDoor**
 - Crea un oggetto di tipo SecretDoor (Porta Segreta fine livello)
- Classe **Character**
 - Crea un oggetto di tipo Character
 - Contiene le funzioni per il movimento del personaggio
 - Gestisce le bombe tramite una Timeline usata anche per l'animazione
- Classe **Enemy**
 - Crea un oggetto di tipo Enemy
 - Contiene le funzioni per il movimento dei nemici
- Classe **FileConfigurazioneXML**
 - Contiene i parametri di configurazione
 - Serializzata/Deserializzata tramite XMLStream
 - Validata tramite XML Schema

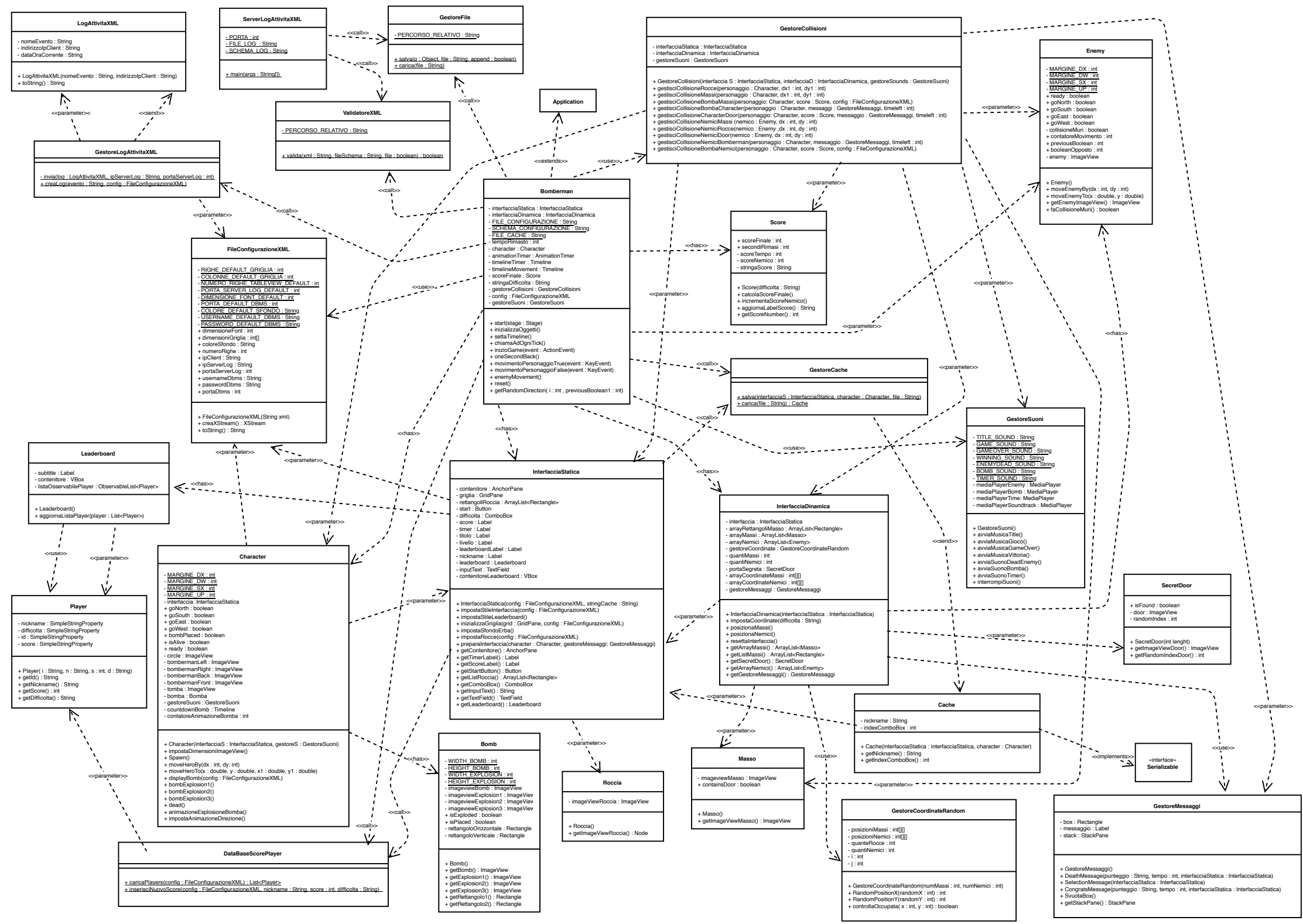
Classi middleware

- Classe **Bomberman**
 - Estende *Application*
 - Realizza il metodo start
 - Gestisce l'AnimationTimer e gli eventi click/tastiera
- Classe **GestoreCollisioni**
 - Gestisce le collisioni tra nemici/personaggio e massi/rocce
 - Gestisce le collisioni tra il personaggio e i nemici
 - Gestisce le collisioni tra bomba e massi/nemici
 - Gestisce la collisione tra la porta segreta e il personaggio

- Classe **GestoreCoordinateRandom**
 - Imposta coordinate randomiche per ogni masso e nemico da posizionare
 - Impedisce il posizionamento di nemici/massi sulle stesse caselle
- Classe **GestoreMessaggi**
 - Gestisce i messaggi di Vittoria/GameOver da mostrare al giocatore
 - Li inserisce dinamicamente nell'interfaccia statica
- Classe **GestoreSuoni**
 - Gestisce Soundtrack/Suoni dell'applicazione
- Classe **GestoreCache**
 - Salva la cache su file binario alla chiusura dell'applicazione
 - Importa la cache da file binario all'avvio dell'applicazione
- Classe **Cache**
 - Implementa *Serializable*
 - Contiene i dati di input su file binario
- Classe **Score**
 - Gestisce il punteggio del livello
- Classe **GestoreFile**
 - Scrive e legge files

Classi back-end

- Classe **Player**
 - Classe bean
- Classe **DatabaseScorePlayers**
 - Carica su List i record con punteggio più alto dal database
 - Inserisce un nuovo record nel database
- Classe **GestoreLogAttivitaXML**
 - Crea log XML e li invia al server log
- Classe **LogAttivitaXML**
 - Contiene le informazioni di un log
 - Serializzata tramite XMLStream
 - Inviata alla classe *ServerLogAttivitaXML*
 - Validata tramite XMLSchema
- Classe **ServerLogAttivitaXML**
 - Mostra a video i singoli log e li salva su file xml, dopo averli validati singolarmente
- Classe **ValidatoreXML**
 - Valida stringhe o file XML secondo dato uno schema



Manuale d'uso

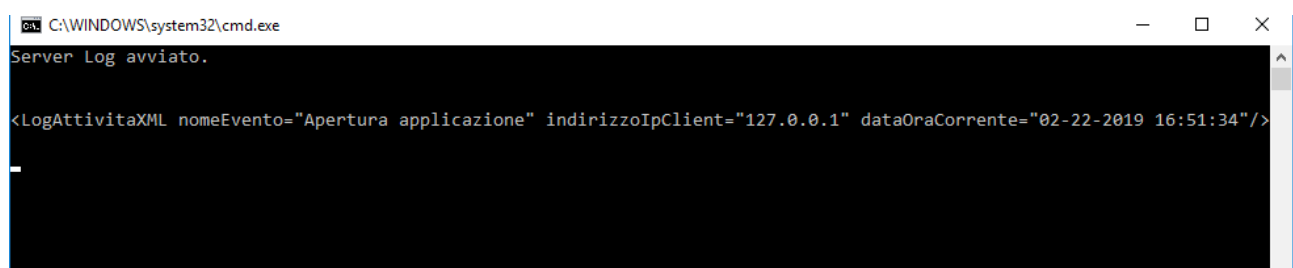
- Studente: Tommaso Giorgi
- Matricola: 546921

Avvio dell'applicazione

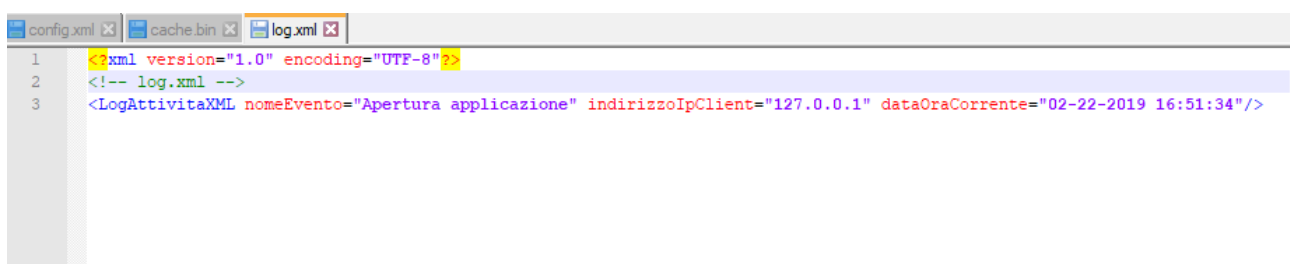
All'avvio dell'applicazione all'utente viene mostrata l'interfaccia grafica, in caso non sia il primo avvio dell'applicazione verrà caricata la cache che imposterà la difficoltà e inserirà un nickname all'interno del campo di testo dedicato. In caso di primo avvio, l'interfaccia sarà la seguente:



Verrà inviato un log relativo all'avvio dell'applicazione al server log, che lo stamperà a video:

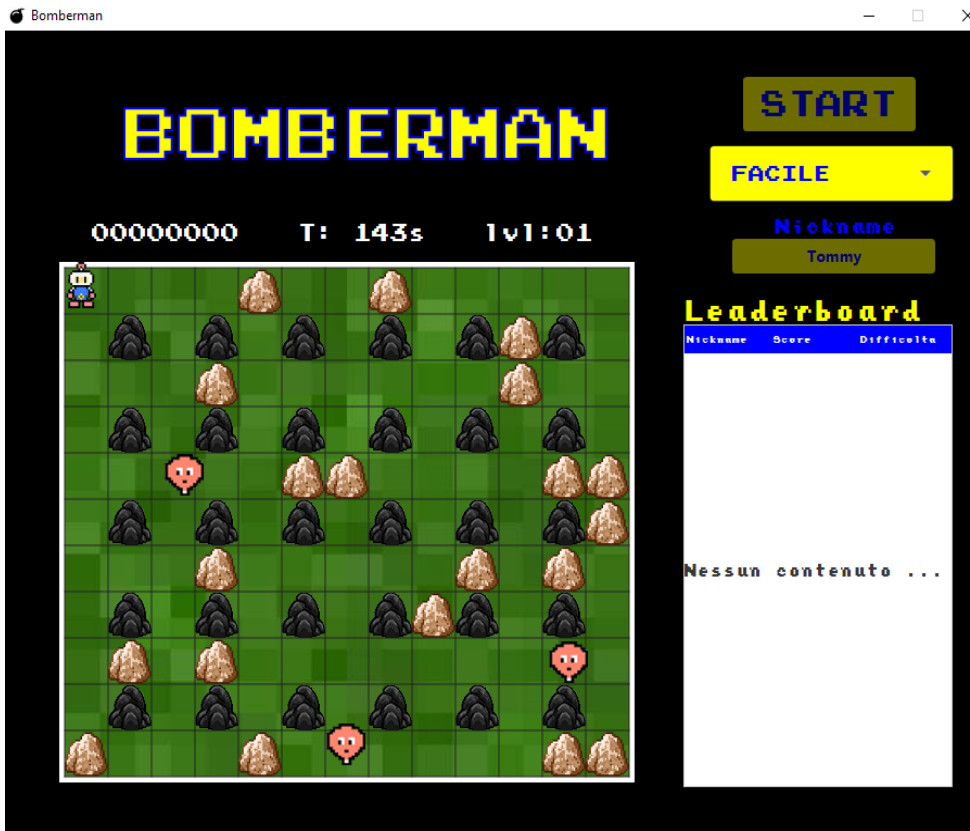


e lo salverà nel file log.xml:



Avvio della partita

Se l'utente clicca sul pulsante "START" ha inizio il gioco. Verranno posizionati un numero variabile di massi all'interno dell'area di gioco e di nemici a seconda della difficoltà selezionata e questi ultimi inizieranno a muoversi randomicamente per la mappa. Il pulsante "START" e il campo di testo verranno disabilitati mentre il gioco è in corso.



Difficoltà: FACILE

Tempo → 150s

N° Nemici → 3

N° Massi → 20

Difficoltà: INTERMEDIO

Tempo → 120s

N° Nemici → 5

N° Massi → 30

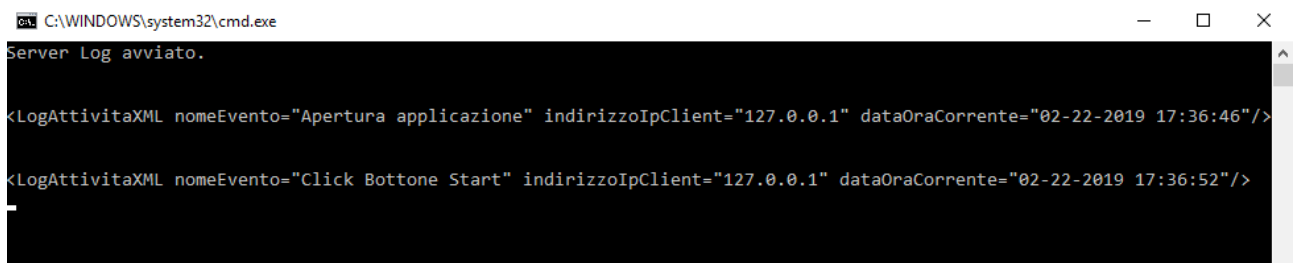
Difficoltà → DIFFICILE

Tempo → 90s

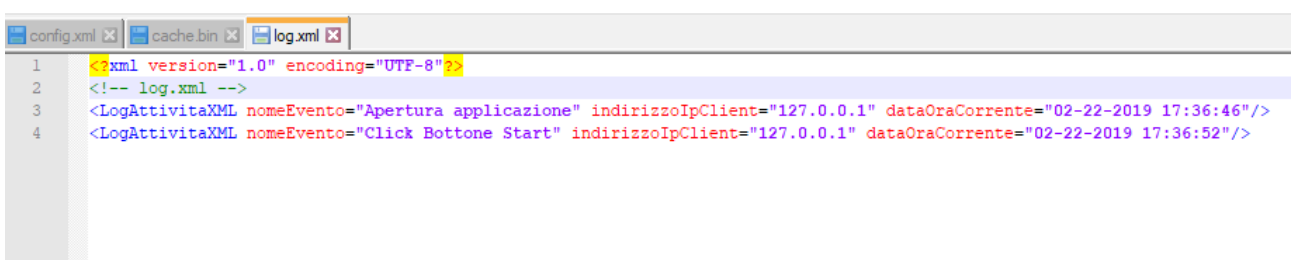
N° Nemici → 7

N° Massi → 40

Come per l'avvio, verrà inviato un log relativo al click del pulsante "START" al server di log, che lo stamperà a video:



E lo salverà nel file log.xml.



Movimento

L'utente potrà a questo punto muoversi nella mappa di gioco usando i tasti **W,A,S,D** per il movimento e potrà piazzare bombe con il tasto **SHIFT**.



Viene premuto D → Movimento verso destra



Viene premuto A → Movimento verso sinistra



Viene premuto S → Movimento verso il basso

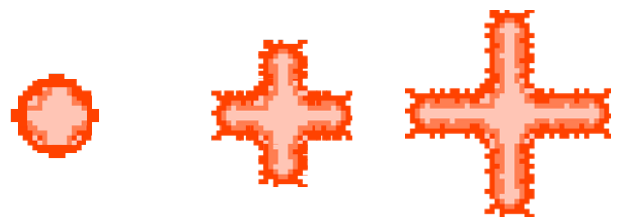


Viene premuto W → Movimento verso l'alto

Gestione Bombe



Viene Premuto SHIFT → La bomba viene piazzata



Animazione Bomba

Ogni volta che una bomba viene piazzata, verrà inviato un log relativo al piazzamento della bomba al server di log, che lo stamperà a video:

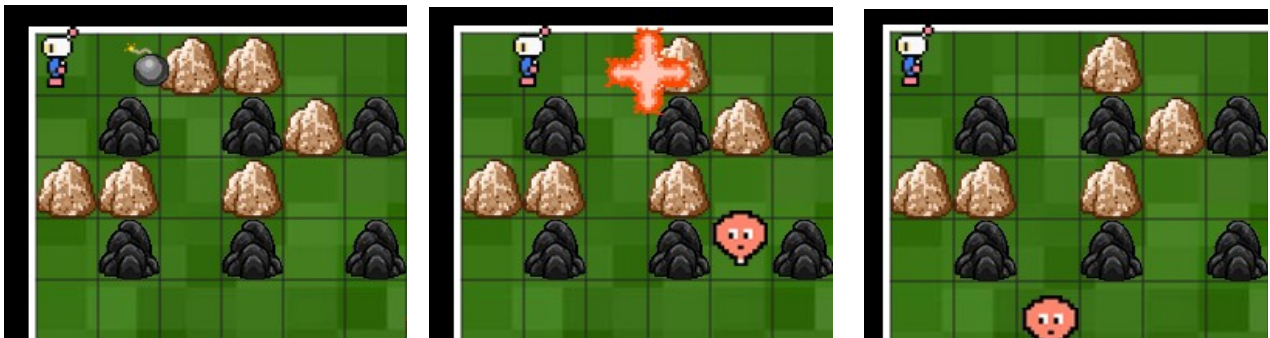
```
C:\WINDOWS\system32\cmd.exe
Server Log avviato.
<LogAttivitaXML nomeEvento="Apertura applicazione" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 18:08:24"/>
<LogAttivitaXML nomeEvento="Click Bottone Start" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 18:08:32"/>
<LogAttivitaXML nomeEvento="Bomba Piazzata" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 18:08:34"/>
```

E lo salverà nel file log.xml:

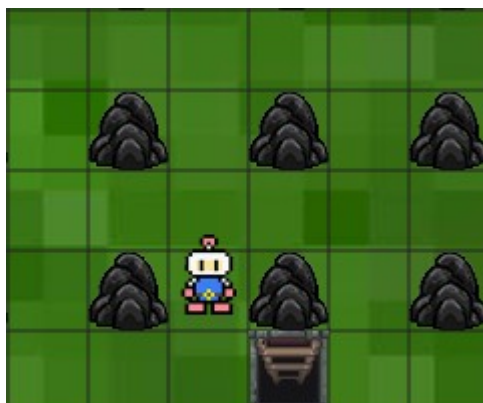
```
config.xml  cache.bin  log.xml
1  <?xml version="1.0" encoding="UTF-8">
2  <!-- log.xml -->
3  <LogAttivitaXML nomeEvento="Apertura applicazione" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 18:08:24"/>
4  <LogAttivitaXML nomeEvento="Click Bottone Start" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 18:08:32"/>
5  <LogAttivitaXML nomeEvento="Bomba Piazzata" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 18:08:34"/>
```

La bomba esplode dopo 1.6 secondi e non posso piazzare un'altra bomba se la precedente non è esplosa. Vediamo alcuni scenari d'esplosione possibili:

Distruzione "massi":



Se si piazza una bomba vicino ad uno o più massi, la bomba una volta esplosa li distruggerà. Uno dei massi nella mappa di gioco una volta distrutto potrebbe rivelare una botola per completare il livello:



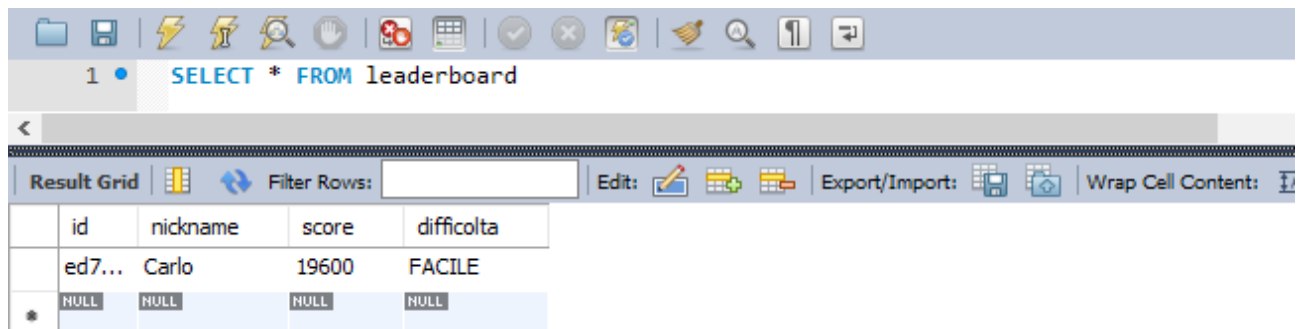
Appena viene rivelata la botola, viene inviato l'evento al server di log che lo stamperà a video e lo salverà sul file log.xml.

Se il personaggio viene spostato sopra la botola il livello termina con una **vittoria**.
Viene mostrato a video il seguente messaggio:



Il punteggio finale non coincide con il punteggio visualizzato sopra la mappa di gioco, infatti viene calcolato attribuendo un punteggio per ogni secondo rimasto dall'inizio del livello

Vengono inseriti ID | Nickname | Difficoltà | Punteggio della partita corrente all'interno della tabella *leaderboard* nel database. (ID viene generato automaticamente in modo che sia univoco)



I dati nella tabella vengono immediatamente mostrati nella tabella Leaderboard dell'interfaccia e i pulsanti vengono riattivati e pronti per una nuova partita:

Leaderboard		
Nickname	Score	Difficoltà
Carlo	19600	FACILE



Uccisione nemici:

Se la bomba esplode nelle vicinanze di un nemico esso verrà rimosso dal gioco. (Questo evento sarà inviato al server di log che lo stamperà a video e lo salverà su log.xml)

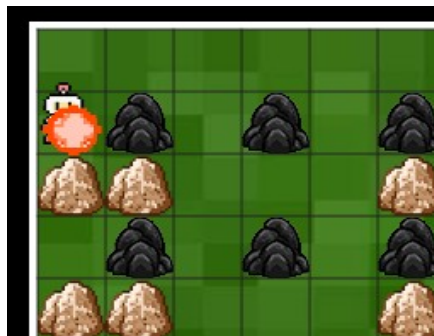
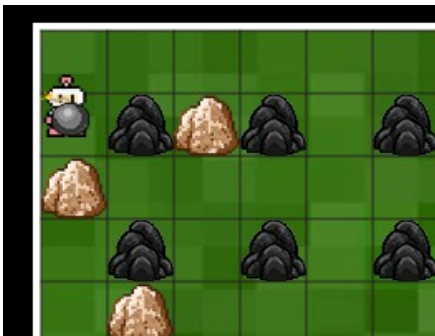


Game Over

Il Game Over può avvenire in tre circostanze:

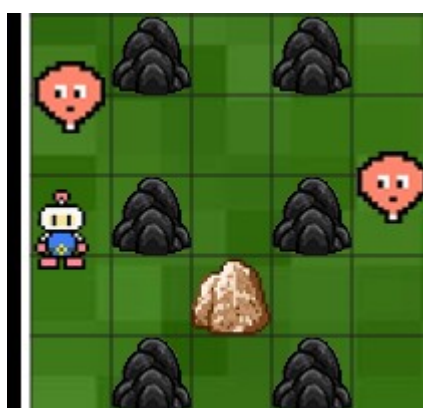
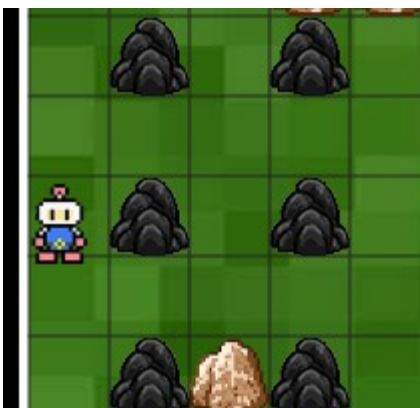
Suicidio con una propria bomba:

Se il personaggio si trova nelle vicinanze della bomba al momento dell'esplosione, verrà colpito e ciò causerà la sua morte.



Ucciso da un nemico

Se il personaggio entra in contatto con uno dei nemici nella mappa di gioco, verrà colpito e ciò causerà la sua morte.



Timeout:

Se il tempo a disposizione per completare il livello termina, avviene il Game Over.



In tutti questi scenari verrà visualizzato un messaggio identico di Game Over e verrà resettata l'interfaccia per poter incominciare una nuova partita:



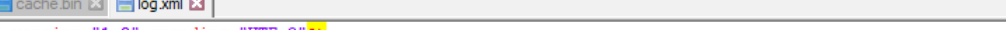
Chiusura Applicazione

Alla chiusura dell'applicazione, verrà inviato tale evento al server di log che lo stamperà a video:

```
C:\WINDOWS\system32\cmd.exe
Server Log avviato.

<LogAttivitaXML nomeEvento="Apertura applicazione" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 20:00:30"/>
<LogAttivitaXML nomeEvento="Click Bottone Start" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 20:00:35"/>
<LogAttivitaXML nomeEvento="Bomba Piazzata" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 20:00:38"/>
<LogAttivitaXML nomeEvento="Chiusura applicazione" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 20:00:41"/>
```


E lo salverà sul file log.xml:



The screenshot shows a code editor with three tabs: 'config.xml', 'cache.bin', and 'log.xml'. The 'log.xml' tab is active, displaying the following XML content:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- log.xml -->
3 <LogAttivitaXML nomeEvento="Apertura applicazione" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 20:00:30"/>
4 <LogAttivitaXML nomeEvento="Click Bottone Start" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 20:00:35"/>
5 <LogAttivitaXML nomeEvento="Bomba Piazzata" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 20:00:38"/>
6 <LogAttivitaXML nomeEvento="Chiusura applicazione" indirizzoIpClient="127.0.0.1" dataOraCorrente="02-22-2019 20:00:41"/>

```

Verrà poi memorizzato sul file binario cache.bin:

- La difficoltà selezionata prima della chiusura dell'applicazione
- La stringa di testo inserita nel campo di testo con label Nickname

```
1 -i NUL:ENCsrNUL:ENCCacheP\áACKACK-Ie STXNULSTXI NUL
2 indexComboBoxL NUL:BSnicknamemet NUL:DC2Ljava/lang/String;xp NUL:NULNUL:NULt NUL:ENC Tommy
```