

**DIP. DI MATEMATICA E INFORMATICA**

**CORSO DI LAUREA IN INFORMATICA**



## Interfacce Grafiche e Programmazione ad eventi

Aiello Alessandra 169988

Monetti Salvatore 162222

Russo Chiara 163538

## Eating Cells

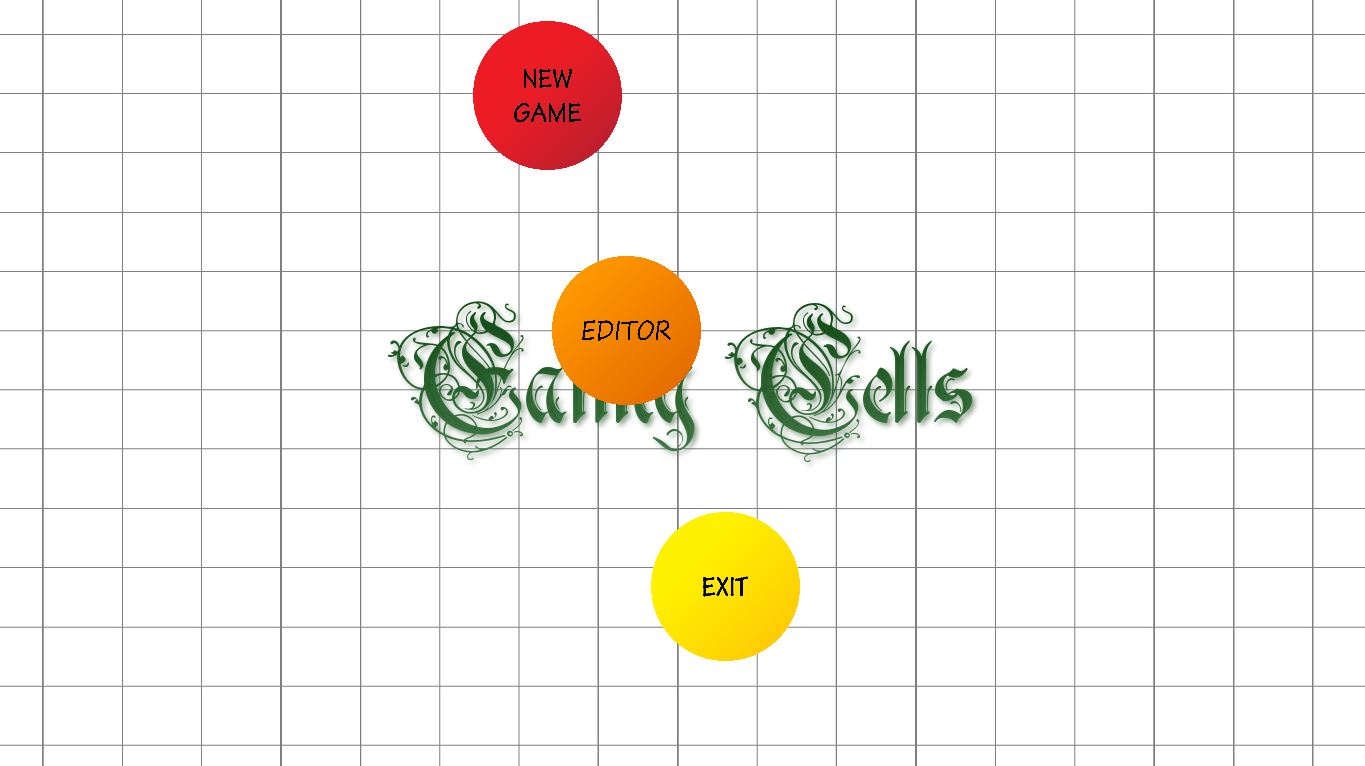
Eating Cells è un gioco che consiste nel “mangiare” le altre entità di dimensione inferiore a quella del giocatore, ed evitare di essere mangiato a sua volta. Lo scopo principale del gioco è quello di sopravvivere.

Le entità si dividono in due categorie:

* quelle mobili, ovvero i giocatori;
* quelle fisse che possono essere ostacoli oppure bonus (questi ultimi sono inizialmente mobili);

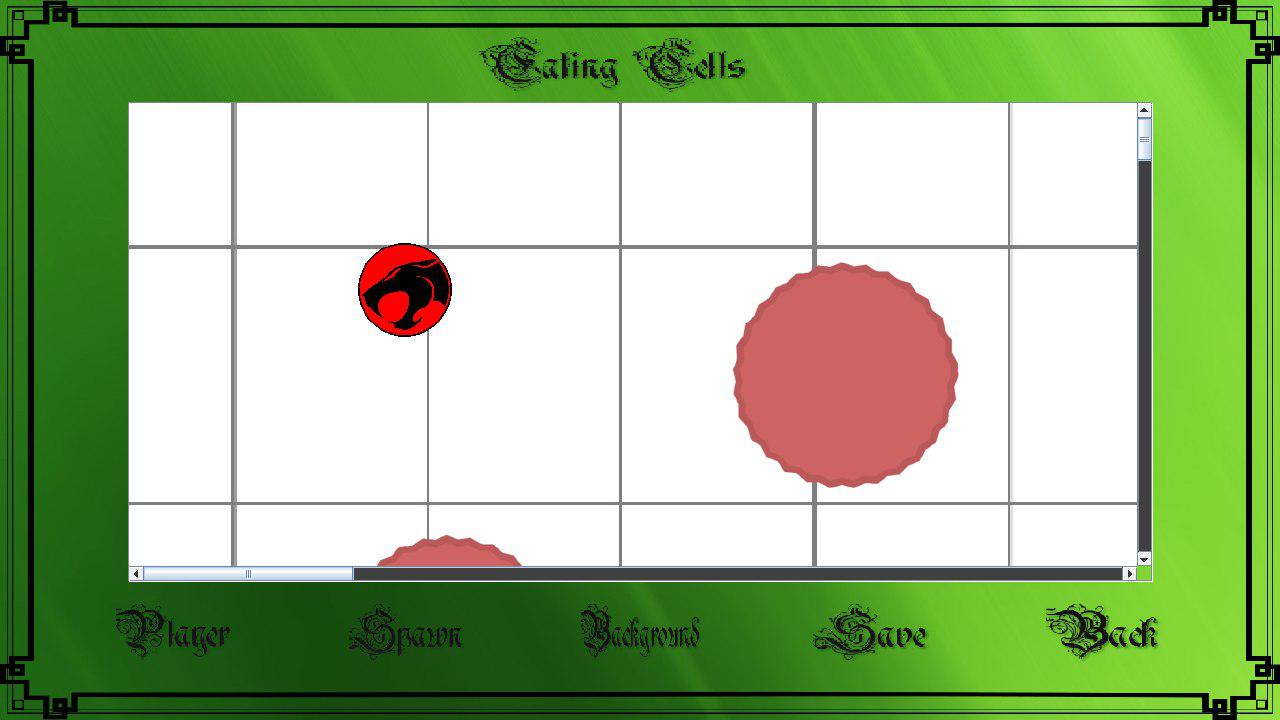
Tra le difficoltà del gioco troviamo la velocità la quale diminuisce all’aumentare della dimensione.

Durante la partita potrebbero comparire alcuni bonus temporanei come velocità ed espansione, che aumenteranno, rispettivamente, la velocità o la dimensione di chi le mangia.

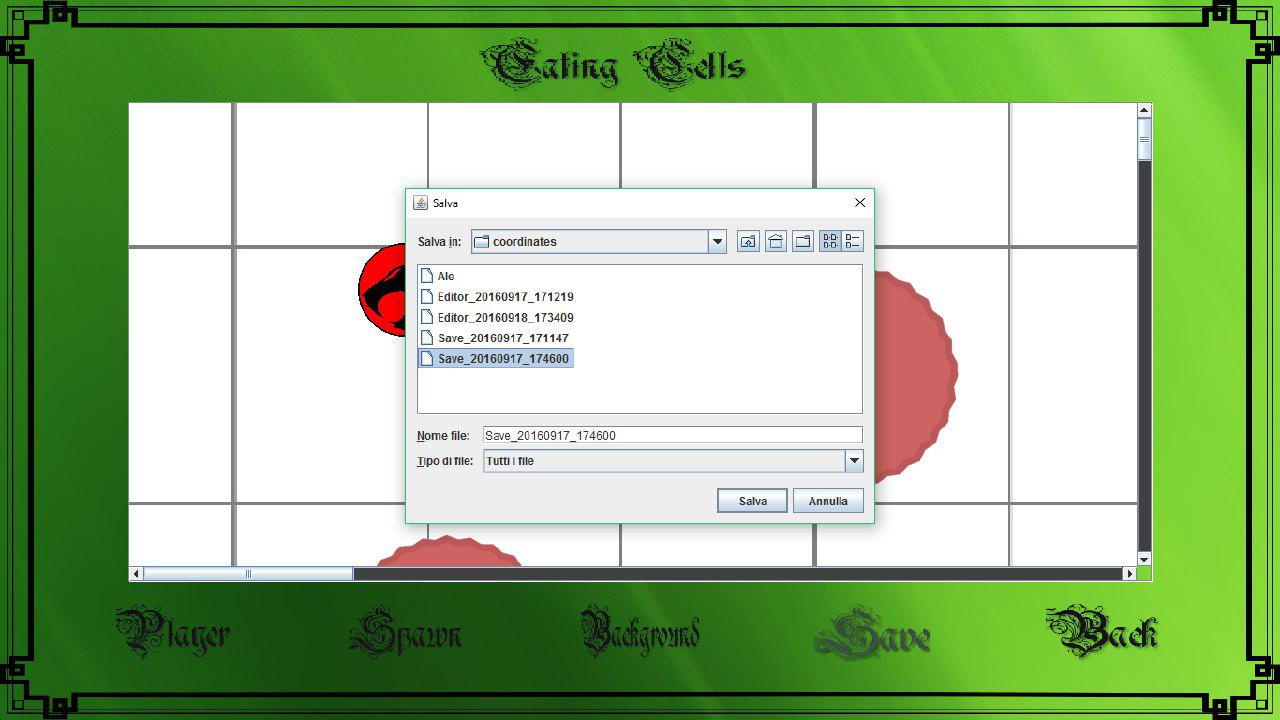


Il gioco si apre con una schermata con tre tasti, il tasto “New Game” per iniziare una nuova partita, il tasto “Exit” per uscire e il tasto per accedere all’editor del gioco.

In particolare quest’ultima funzione permette di creare una propria partita personalizzata.



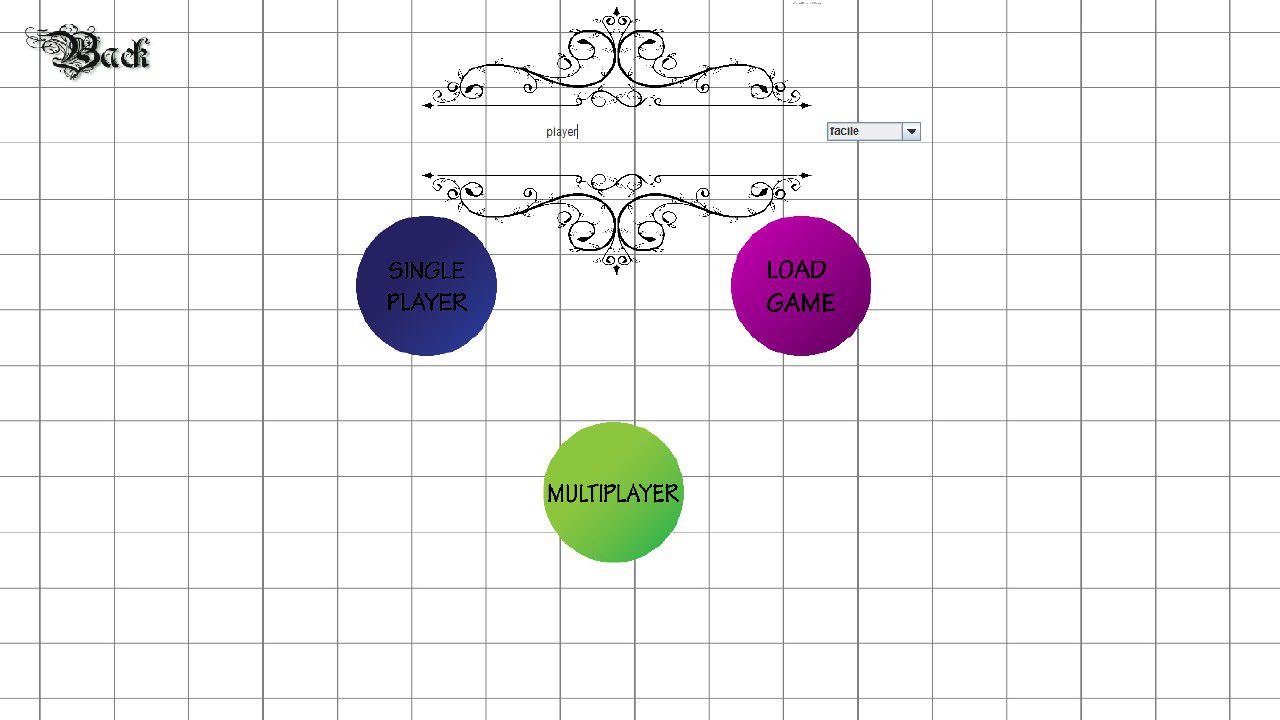
L’Editor permette di scegliere l’immagine che rappresenterà il giocatore durante la partita e la posizione dalla quale desidera partire. Oltre la scelta dei personaggi vi è la possibilità di editare la mappa, aggiungendo ostacoli, modificando la posizione dei personaggi e spostandoli a proprio piacimento.



Una volta personalizzata la mappa di gioco, anche attraverso la scelta dello sfondo, la si può salvare ed utilizzarla quando si vuole. O si può semplicemente tornare al menù iniziale attraverso il tasto “Back”.

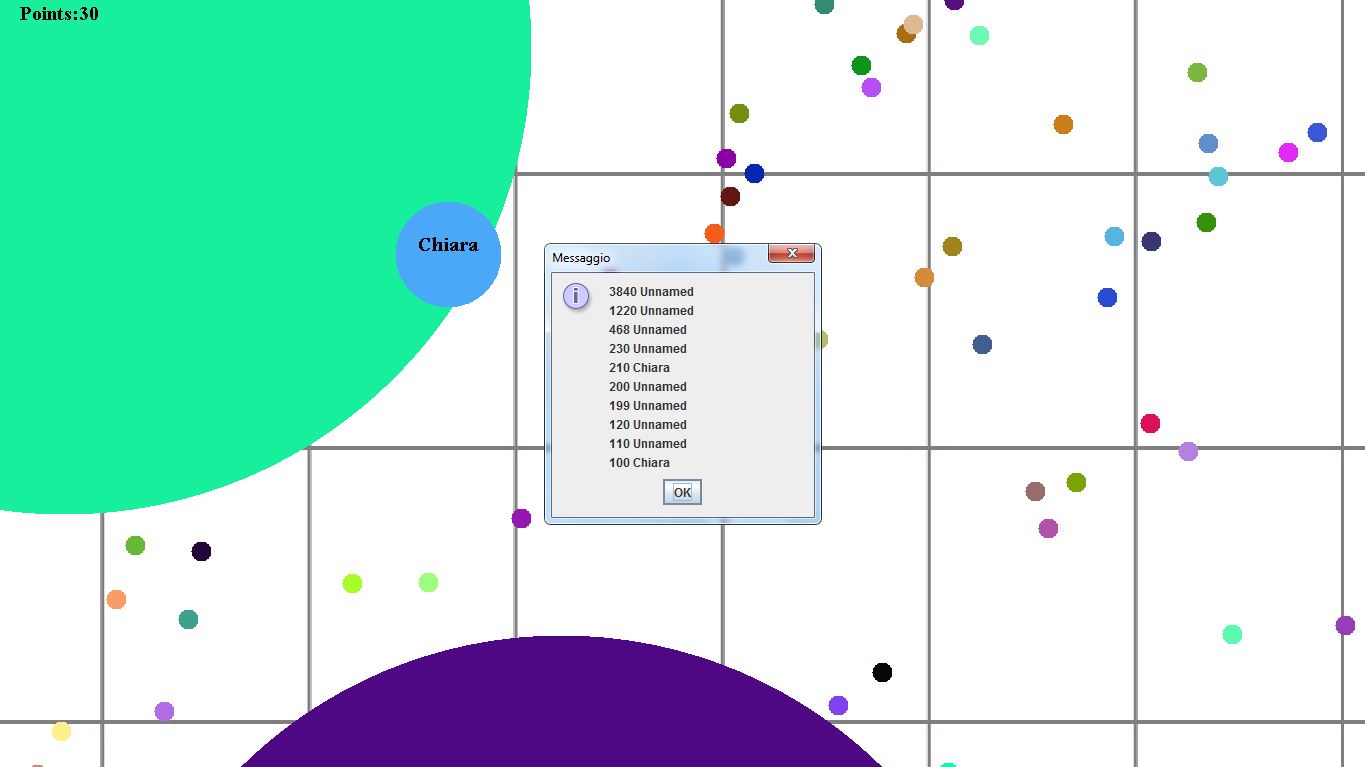
All’interno del gioco si ha la possibilità di settare il proprio nickname, il quale viene inserito prima di iniziare la partita e accompagnerà il giocatore per tutta la durata del gioco.

Oltre l’username, è possibile anche impostare la difficoltà di gioco divisa in “facile” e “difficile”.



Premendo sul “Single Player” si inizia una nuova partita. Durante questa modalità di gioco è inoltre possibile salvare la partita e caricarla in seguito.

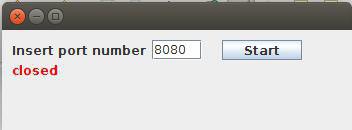
Il gioco terminerà quando il giocatore sarà mangiato o quando il player avrà mangiato tutti i suoi nemici.



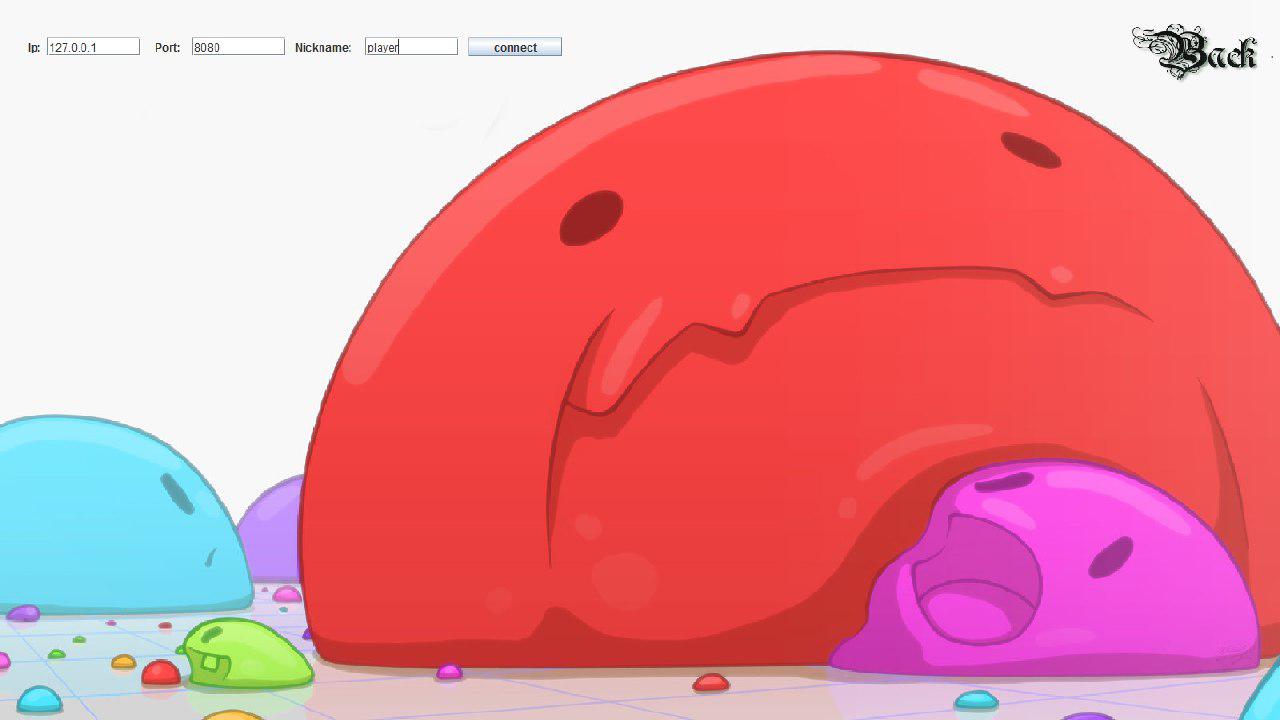
Subito dopo comparirà il punteggio del giocatore e dei partecipanti alla partita.

## Modalita’ Versus (Online)

L’applicazione permette di sfidare online altri giocatori. Vi è un’applicazione esterna che fa da server, sulla quale è possibile scegliere la porta su cui collegare il server.



Cliccando sul “Multi Player”, dopo aver inserito il Nickname, si inserisce L’IP e il numero della porta e si joina sulla modalità online.



Il vincitore sarà il giocatore che avrà ottenuto il punteggio più alto.

Casi d’uso

<extends>

Multi Player

Giocare



<extends>

Caricamento

User

Creare una mappa

Principali attività

Caso d’uso n°1

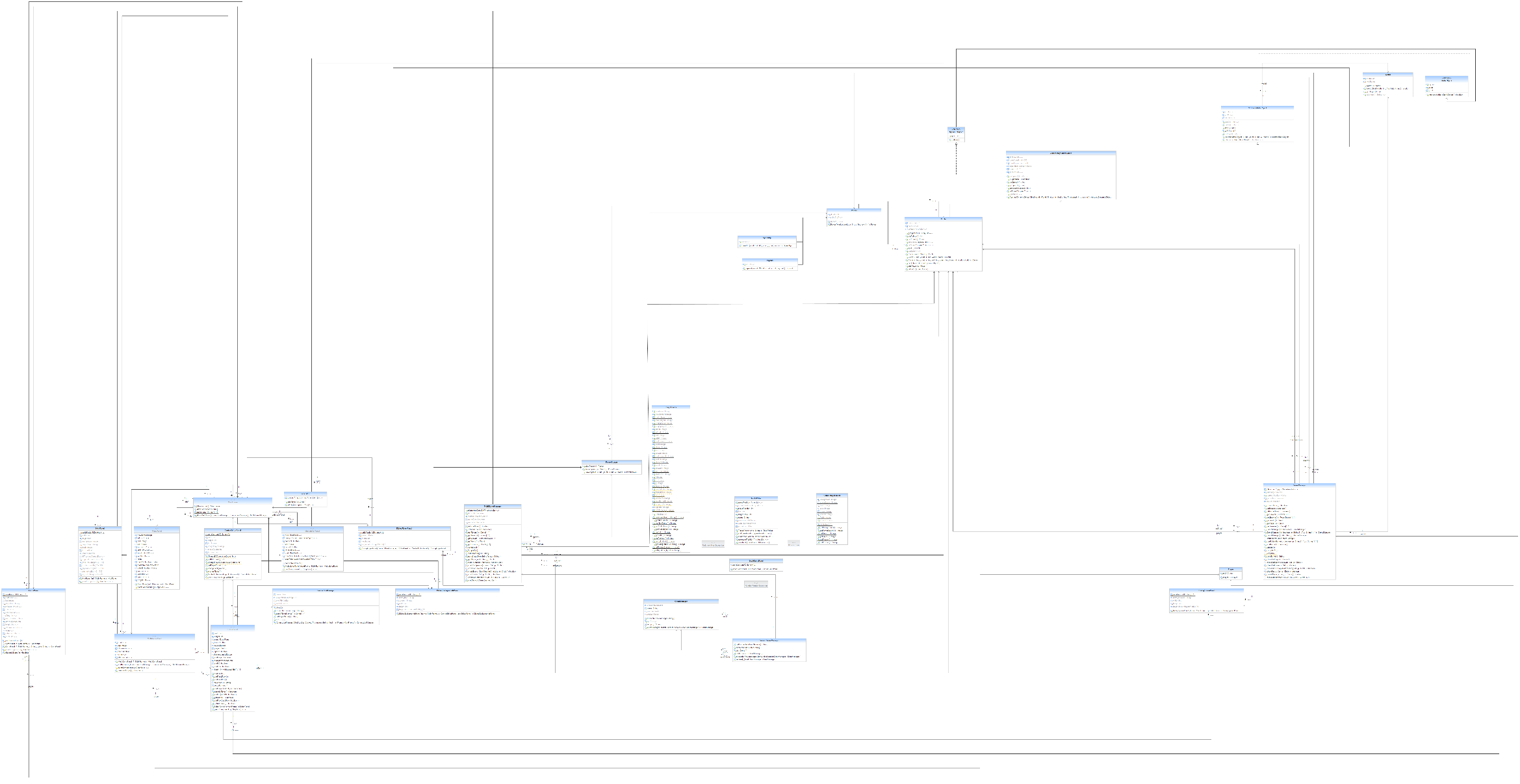
|  |  |
| --- | --- |
| Nome Avviare una partita | |
| *Attore primario* | **Utente** |
| *Altri attori* | Il **Sistema** che interfaccia con il gioco |
| *Pre-condizioni* | L’utente ha avviato l’applicazione |
| *Garanzia di successo (Post-condizioni)* | Visione del punteggio ottenuto alla fine di una partita |
| *Scenario principale (Scenario di successo)* | 1. L’utente avvia l’applicazione 2. L’utente visualizzerà il menù e le opzioni che potrà scegliere 3. L’utente sceglie di effettuare una partita 4. L’utente inserisce il suo Nickname che lo contraddistinguerà nella classifica 5. L’utente decide di effettuare una nuova partita 6. Si avvia la partita 7. L’utente muove il mouse per far muovere il proprio personaggio 8. L’utente viene mangiato 9. Esce la schermata della classifica con il punteggio ottenuto e la relativa posizione 10. L’utente torna al menu principale 11. Il sistema torna al passo 2. |
|  | |
| *Scenari alternativi*  *Frequenza di ripetizione* | 3a. L’utente decide di editare una partita  3a.1 vedi caso d’uso n°2  4a. L’utente non inserisce il proprio nickname  4a.1 il caso d’uso va al punto 5  5a. L’utente decide di effettuare il caricamento.  5a.1 sceglie di caricare una partita  precedentemente giocata  5a.1.1 sceglie di caricare una partita creata da  editor  5a.2 il caso d’uso va al punto 6  5b. L’ utente sceglie la modalità versus online  5b.1 vedi caso d’uso n° 3  3b. L’utente decide di voler chiudere l’applicazione  3b. il caso d’uso termina |
| *Frequenza di ripetizione* | Ogni qual volta l’utente ha voglia di giocare |

Caso d’uso n°2

|  |  |
| --- | --- |
| Nome Editare una mappa | |
| *Attore primario* | **Utente** |
| *Altri attori* | Il **Sistema** che interfaccia con il gioco |
| *Pre-condizioni* | L’utente ha avviato l’applicazione |
| *Garanzia di successo (Post-condizioni)* | Visione della mappa personalizzata all’interno del gioco |
| *Scenario principale (Scenario di successo)* | 1. L’utente avvia l’applicazione 2. L’utente visualizzerà il menu e ciò che potrà scegliere 3. L’utente sceglie di accedere all’editor del gioco 4. Sceglie l’immagine del giocatore e la posiziona sulla mappa 5. Posiziona gli spawn sulla mappa 6. L’utente sceglie l’immagine di sfondo 7. Salva le modifiche effettuate 8. Il caso d’uso termina |
|  | |
| *Scenari alternativi*  *Frequenza di ripetizione* | 7a. L’utente sceglie di tornare indietro al menu principale |
| *Frequenza di ripetizione* | Ogni qual volta l’utente ha voglia di giocare |

Caso d’uso n°3

|  |  |
| --- | --- |
| Nome Avviare una partita online | |
| *Attore primario* | **Utente** |
| *Altri attori* | Il **Sistema** che interfaccia con il gioco |
| *Pre-condizioni* | L’utente ha avviato l’applicazione |
| *Garanzia di successo (Post-condizioni)* | Visione della partita con i giocatori che volevano connettersi ad essa |
| *Scenario principale (Scenario di successo)* | 1. L’utente avvia l’applicazione 2. L’utente visualizzerà il menu e ciò che potrà scegliere 3. L’utente sceglie di effettuare una partita 4. L’utente inserisce il nickname 5. Sceglie di effettuare la modalità online 6. L’utente inserisce i dati per effettuare una partita online, quali, IP e numero della porta del server 7. Attende la connessione dell’altro giocatore 8. La partita inizia 9. Il caso d’uso termina |
|  | |
| *Scenari alternativi*  *Frequenza di ripetizione* | 5a. L’utente non inserisce il nickname  5a.1 Esce una finestra che invita l’utente ad inserire  il nome  5a.2 inserisce il nickname  5a.2 il caso d’uso va al passo 5  6a. Il server non è acceso  6a.1 si accende il server  6a.2 il caso d’uso torna al passo 6 |
| *Frequenza di ripetizione* | Ogni qual volta l’utente ha voglia di giocare |



UML

Descrizione delle classi

**Game Manager:**

Questa classe gestisce l’intero mondo durante una partita offline. I metodi che la compongono si occupano di inizializzare il mondo e riempirlo con gli oggetti del gioco.

Metodi di rilievo:

* startLoadedGame. Questo metodo permette il caricamento di una partita precedentemente giocata e salvata. Crea quindi i personaggi e li posiziona nel punto in cui si trovavano al momento del salavataggio. E la partita ricomincia da qui.
* startEditedGame. Questo metodo dà la possibilità di caricare una mappa editata e salvata precedentemente. Posiziona i personaggi dove il giocatore li ha posizionati al momento del salvataggio.
* startGame. Questo metodo permette di iniziare una nuova partita. Infatti si occupa di creare in modo casuale la posizione del player nella mappa, genera casualmente la posizione e il raggio dei nemici e genera in modo random le posizioni degli spawn e crea un abbondante numero di personaggi che danno la possibilità di far aumentare il volume dei giocatori. Ed inoltre attribuisce alla partita il livello di difficoltà scelto precedentemente dal giocatore.
* update. Attraverso questo metodo si ha un costante aggiornamento dell’andamento della partita. Si occupa di aggiornare il numero delle entità fisse presenti e crearne quindi di nuove ogni quando vengono mangiate. Crea i poteri che compariranno durante la partita e riferisce se il gioco è terminato.

**Editor Panel:**

I metodi che compongono questa classe danno la possibilità di personalizzare una mappa di gioco attraverso la scelta di una skin che rappresenterà il giocatore, permette di posizionare gli spawn che rappresentano degli ostacoli e scegliere un’immagine di sfondo per la mappa.

Metodi di rilievo:

* saveMap. Questo metodo trascrive su file le posizioni dei personaggi, e quindi permette di salvare la mappa.
* paintComponent. Il seguente metodo si occupa di stampare tutte le skin del giocatore e le immagini dei vari componenti del gioco.
* setupButtons. Questo metodo associa a ciascun tasto l’azione corrispettiva che si deve scaturire al momento del click.

**Game Panel:**

Questa è tra le classi più importanti dell’applicazione. Il Game Manager si occupa di inizializzare una partita, e attraverso questa classe il gioco prende vita. Permette di far interagire l’utente con il gioco attraverso i comandi della tastiera e del mouse.

Metodi di rilievo:

-addKeyListener. Attraverso la tastiera permette di visualizzare la mini mappa del gioco, mettere in pausa la partita e salvare una partita che si sta giocando.

- run. Questo metodo si occupa di gestire la partita fino alla fine del gioco, e, a tal punto, manda il giocatore al menu iniziale.

- printLeaderBoard. Questo metodo a fine partita si occupa di mostrare una finestra a fine partita, sulla quale è possibile leggere la classifica con i nomi dei partecipanti e i relativi punteggi.

**MultiGame Manager:**

Questa classe gestisce la partita in modalità online, in particolare si occupa del lato client.

Metodi di rilievo:

* parseStatusFromString. Questo metodo prende la stringa di stato mandatagli dal server e la divide in maniera tale da poter disegnare lo stato del gioco.

**Client Manager:**

Questa classe si occupa di trasferire i dati sul socket .

Metodi di rilievo:

* dispatch. In particolare questo è il metodo che permette ciò.

**Connection Manager:**

Questa classe si occupa di gestire la connessione tra client e server.

Metodi di rilievo:

* run. Questo buffer resta sempre in ascolto, in attesa che il server gli comunichi qualcosa e restituisce al Multigame Manager lo stato del gioco.

**Server – Server Game Manager**

Questa classe si occupa di accettare e aggiungere o rimuovere i client che si devono connettere in rete per giocare e inviare le informazioni dei client connessi a tutti i client.

Metodi di rilievo:

* dispatch. Questo metodo trasferisce al client lo stato corrente del gioco.
* Received. Questo metodo riceve gli aggiornamenti dal client e aggiorna lo stato del gioco.
* parseStatusToString. Questo metodo converte in stringa da spedire, lo stato del gioco.