Unity’s Pac-Man

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.3 Use case 7

2.4 Pianificazione 8

2.5 Analisi dei mezzi 10

2.5.1 Software 10

2.5.2 Hardware 10

3 Progettazione 11

3.1 Design dell’architettura del sistema 11

3.2 Design delle interfacce 12

3.2.1 Interfaccia pagina home 12

Bottoni per iniziare la partita, andare alla pagina delle opzioni od uscire dal gioco. 12

3.2.2 Interfaccia pagina Opzioni 12

3.2.3 Interfaccia pagina di gioco 13

3.3 Design procedurale 14

3.3.1 Iniziale 14

3.3.2 Finale 15

4 Implementazione 11

5 Test 11

5.1 Protocollo di test 11

5.2 Risultati test 18

5.3 Mancanze/limitazioni conosciute 20

6 Consuntivo 21

7 Conclusioni 21

7.1 Sviluppi futuri 21

7.2 Considerazioni personali 21

8 Glossario 21

9 Bibliografia 22

9.1 Sitografia 22

# Introduzione

## Informazioni sul progetto

Allievi: Michael Dobeson, Luca Fumasoli

Classe: I3AC

Docente responsabile: Guido Montalbetti

Data inizio: 09.09.2022

Data fine: 23.12.2022

## Abstract

*Pac-Man is a well-known videogame which has stood the test of time and cemented itself has a classic.*

*In this game the player has to control Pac-Man in a maze that has a pill on every cell and the players goal is to eat all the pills. In the meantime, there are 4 ghosts following the player and if they catch you, you lose a life. If the player loses al his lives the game is over.*

*There are also super-pills in the maze which make Pac-Man invincible and capable of eating the ghosts.*

*In our variation of Pac-Man there is only one ghost and the maze is randomly generated each game.*

## Scopo

Lo scopo di questo progetto è di avere una variazione di Pac-Man con 1 fantasma e dei labirinti randomizzati per continuare a rendere le partite diverse ed eliminare la familiarità che il giocatore potrebbe avere con la mappa forzandolo ad improvvisare ogni partita.

# Analisi

## Analisi del dominio

Il gioco dovrebbe essere il più semplice possibile da utilizzare dall’utente, esattamente come il gioco originale. Gli utenti dovrebbero essere già familiari con il gioco originale e giocando alla nostra versione dovrebbero capire velocemente come funziona.

## Analisi e specifica dei requisiti

Priorità 2 = opzionale

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Generazione Mappa |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si deve usare l'algoritmo Manhattan Mapper per generare una mappa. |
| **Sotto requisiti** | |
| **001** | Conoscenza base di Unity. |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Movimento Pac-Man |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il giocatore può usare la tastiera per muoversi dentro la mappa. |
| **Sotto requisiti** | |
| **001** | Una matrice su cui può muoversi. |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Piazzamento Pillole |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Vengono piazzate le pillole e Super-Pillole, che aumentano il punteggio, in modo casuale dentro la mappa. |
| **Sotto requisiti** | |
| **001** | Una matrice su cui piazzarle. |
| **ID: REQ-04** | |
| **Nome** | AI Fantasma |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Usando l'algoritmo Manhattan Mapper l'AI viene piazzato sulla mappa per inseguire il giocatore. |
| **Sotto requisiti** | |
| **001** | Una matrice su cui può muoversi. |
| **002** | La posizione del giocatore. |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Super-Pillole |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Quando Pac-Man consuma una Super-Pillola il fantasma scappa (Manhattan inverso) e se viene raggiunto inizia a fuggire più velocemente per un tempo parametrizzabile per poi rinascere e ritornare ad inseguire Pac-Man. |
| **Sotto requisiti** | |
| **001** | Le pillole sono piazzate. |
| **002** | Il giocatore riesce a consumarle. |
| **003** | Un'AI che segue il giocatore. |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Grafica |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Un'interfaccia con i bottoni per cominciare, uscire e ricominciare. Dettagli del gioco come le vite, il punteggio e l'Highscore che viene salvato in una variabile globale. |
| **Sotto requisiti** | |
| **001** | Le pillole sono piazzate. |
| **002** | Il giocatore riesce a consumarle. |
| **003** | Un'AI che segue il giocatore. |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Vittoria/Perdita/Ricomincia |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | I diversi casi in cui una nuova mappa viene generata ed i personaggi vengono ripristinati:   * Quando tutte le pillole e Super-Pillole sono consumate (Vittoria). * Quando perdi tutte le tue vite per colpa del fantasma che ti colpisce (Perdita). * Quando metti in pausa il gioco e clicchi ricomincia. |
| **Sotto requisiti** | |
| **001** | Le pillole sono piazzate. |
| **002** | Il giocatore riesce a consumarle. |
| **003** | Un'AI che segue il giocatore. |
| **004** | Pulsanti per ricominciare la partita. |

|  |  |
| --- | --- |
| **ID: REQ-08** | |
| **Nome** | Variabili del Gioco |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Prima di iniziare la partita, si potrà modificare delle variabili di gioco (Vite, tempo Super-Pillole, grandezza delle mappe, velocità del fantasma, ecc..). |
| **Sotto requisiti** | |
| **001** | Il gioco finito e funzionante. |

|  |  |
| --- | --- |
| **ID: REQ-09** | |
| **Nome** | Difficoltà |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Si potrà scegliere il livello di difficoltà prima di iniziare la partita. |
| **Sotto requisiti** | |
| **001** | Il gioco finito e funzionante. |
| **002** | Si può cambiare le variabili del gioco facilmente. |

|  |  |
| --- | --- |
| **ID: REQ-10** | |
| **Nome** | Gaming controller |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Si potrà navigare il menu e giocare al gioco usando solo il controller. |
| **Sotto requisiti** | |
| **001** | Il gioco finito e funzionante. |
| **002** | Il menu finito e funzionante. |

## Use case

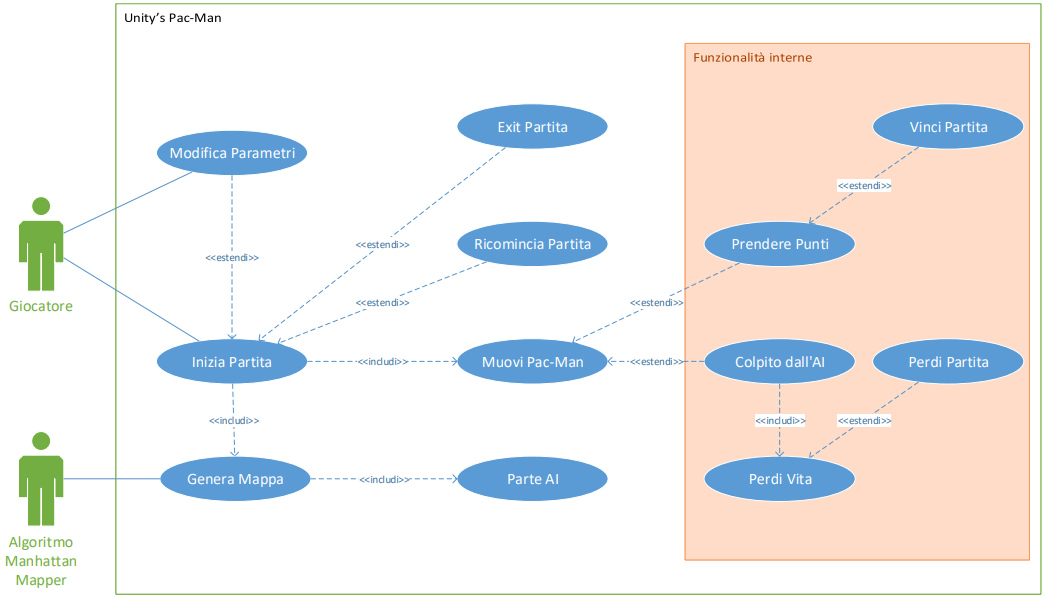


Figura 1 - Use Case

Ci sono 2 tipi di utente:

1. **Giocatore:** Il giocatore può modificare i parametri del gioco a suo piacimento prima di iniziare la partita. Poi quando avvia la partita muove il Pac-Man per il labirinto. Mangiando le pillole nel labirinto guadagna punti, se li mangia tutti vince la partita. Se viene colpito dal fantasma perde una vita e alla perdita di tutte le sue vite perde la partita.  
   Può anche mettere in pausa la partita e ricominciarla o uscire.
2. **Algoritmo Manhattan Mapper:** Si occupa di generare la mappa randomica e muovere l’AI, usando l’algoritmo trova la strada più corta per arrivare al giocatore e la segue, se invece il giocatore ha consumato una super-pillola trova la strada più efficace per scappare dal giocatore.

## Pianificazione

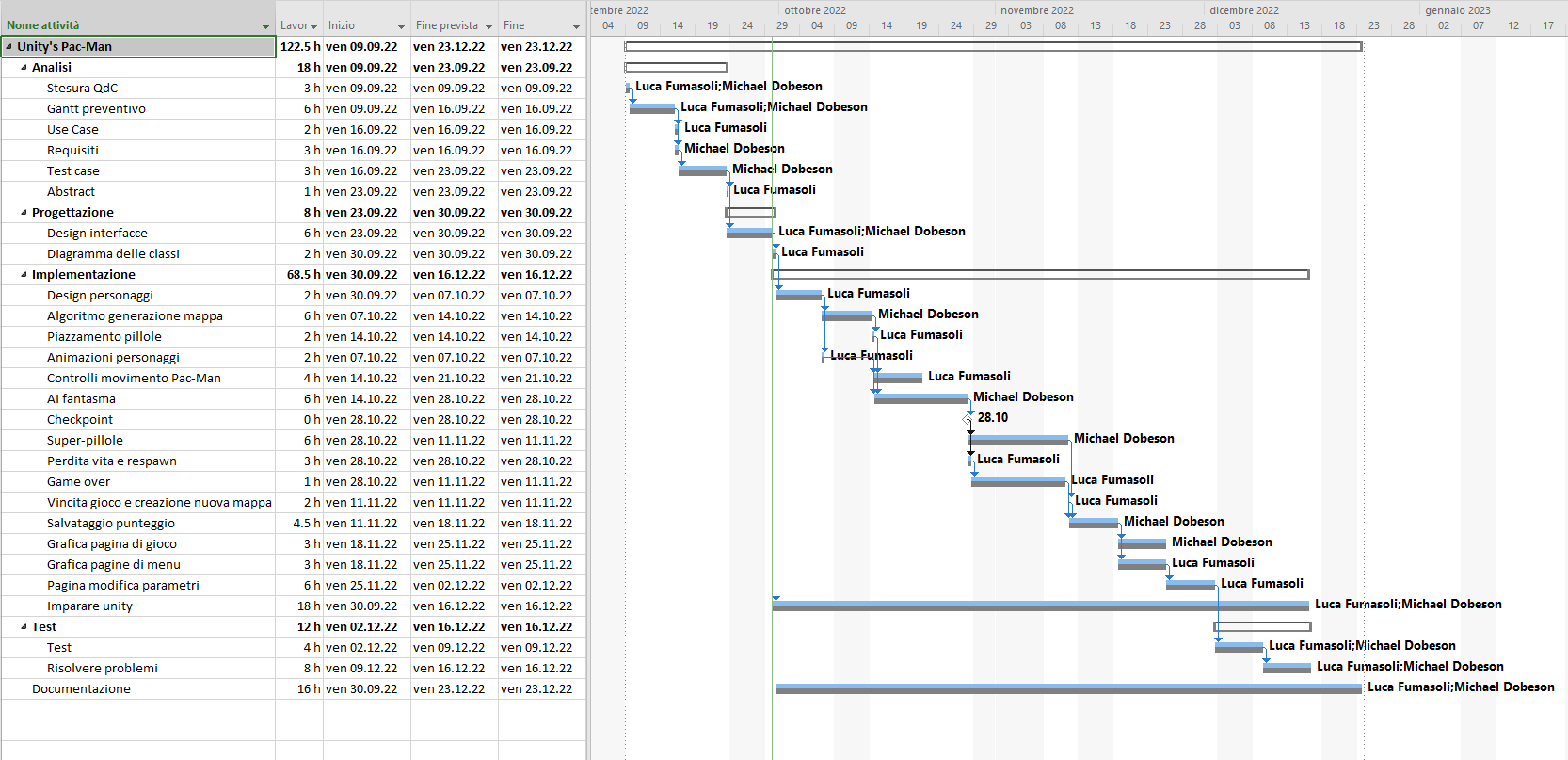


Figura 2 - Gantt

## Analisi dei mezzi

### Software

I software che abbiamo usato sono:

* Microsoft Office Professional Plus 2019
* Microsoft Word: per fare la documentazione
* Microsoft Visio: per fare lo schema del sito e dei database
* Microsoft Project: per fare il diagramma di Gantt
* PlantUML: Per fare il diagramma di flusso e di classi
* Paint versione 10.0: per il design delle interfacce
* Unity 2022.1.1f1

### Hardware

Come hardware abbiamo usato i computer scolastici:

* Processore Intel Core i7-9700 CPU @ 3.00GHz
* RAM: 32GB
* Scheda video: Intel(R) UHD Graphics 630

# Progettazione

## Design dell’architettura del sistema

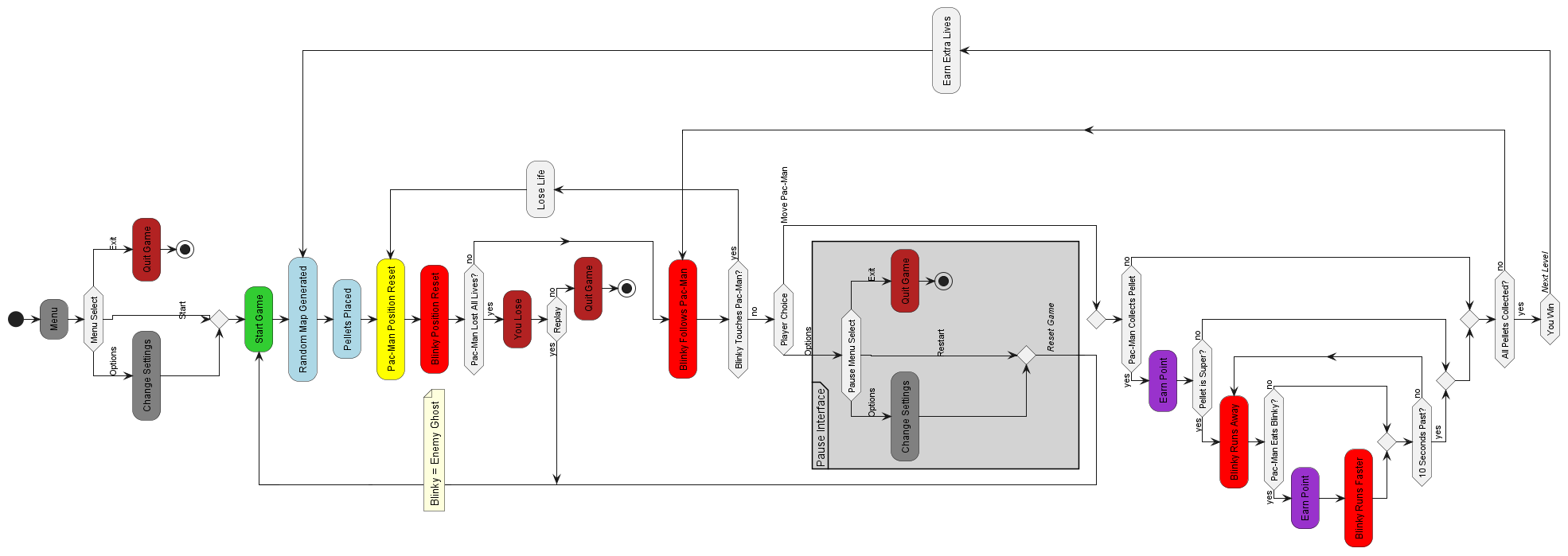


Figura 3 - Design dell'architettura del sistema

## Design delle interfacce

### Interfaccia pagina home

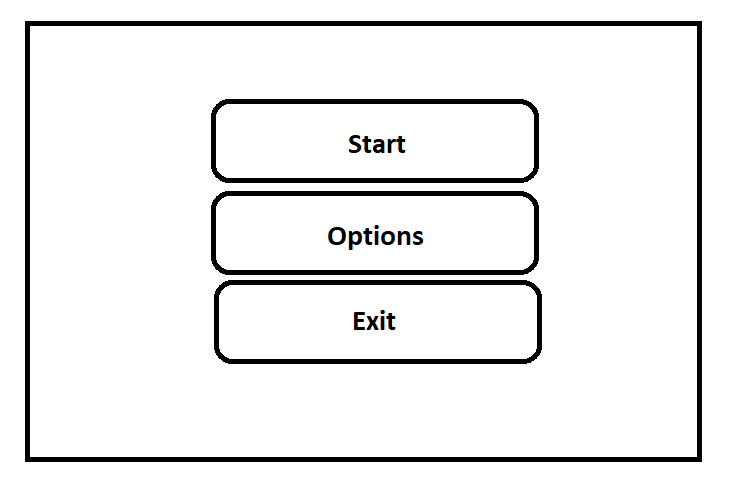


Figura 4 – Pagina di home

## Bottoni per iniziare la partita, andare alla pagina delle opzioni od uscire dal gioco.

### Interfaccia pagina Opzioni

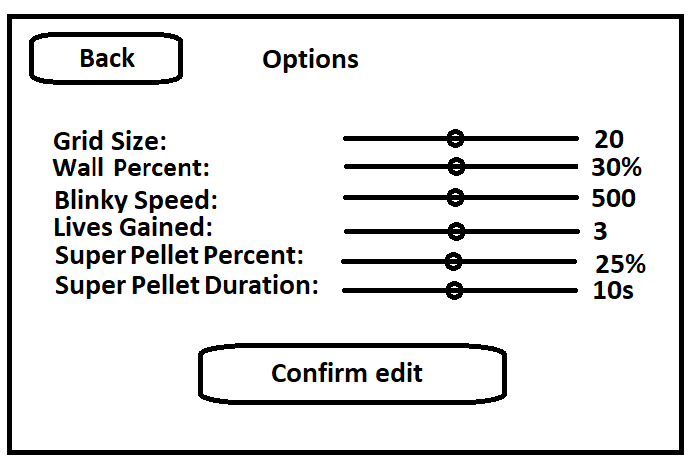


Figura 5 - Pagina settaggi

Pagina opzioni in cui si possono modificare le variabili di gioco.

### Interfaccia pagina di gioco

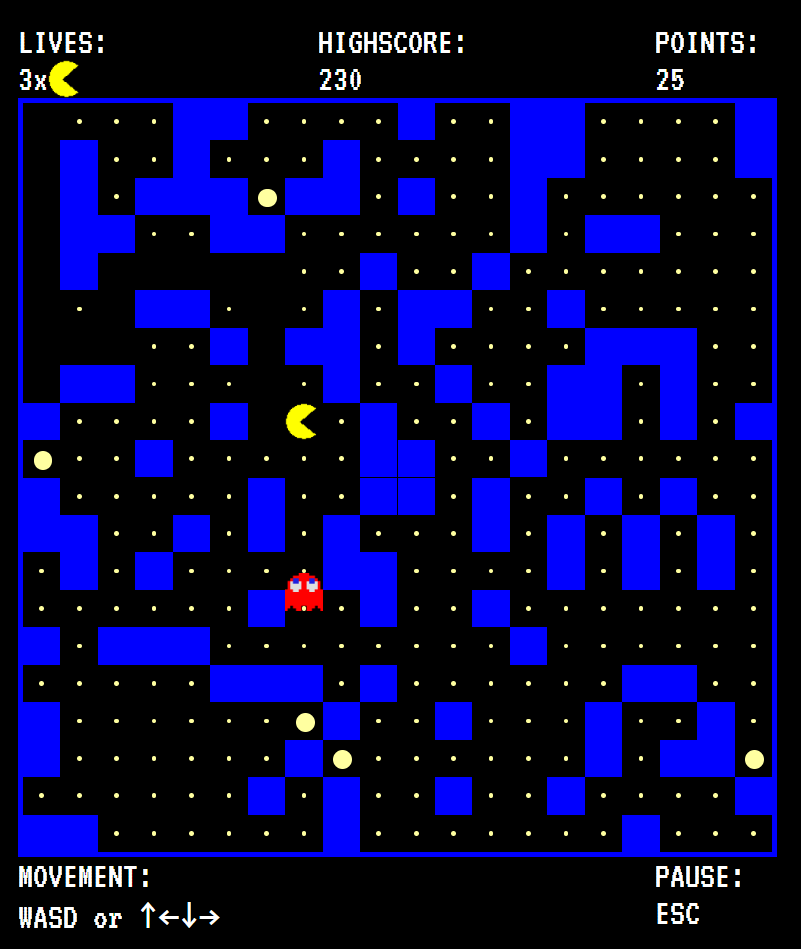


Figura 6 - Interfaccia pagina di gioco

Pagina di gioco con il labirinto ed il personaggio da muovere in centro ed i comandi sotto.  
In cima alla pagina a sinistra ci sono il numero di vite rimaste, in centro c’è il miglior punteggio ottenuto, ed ha destra il punteggio corrente della partita.

## Design procedurale

### Iniziale

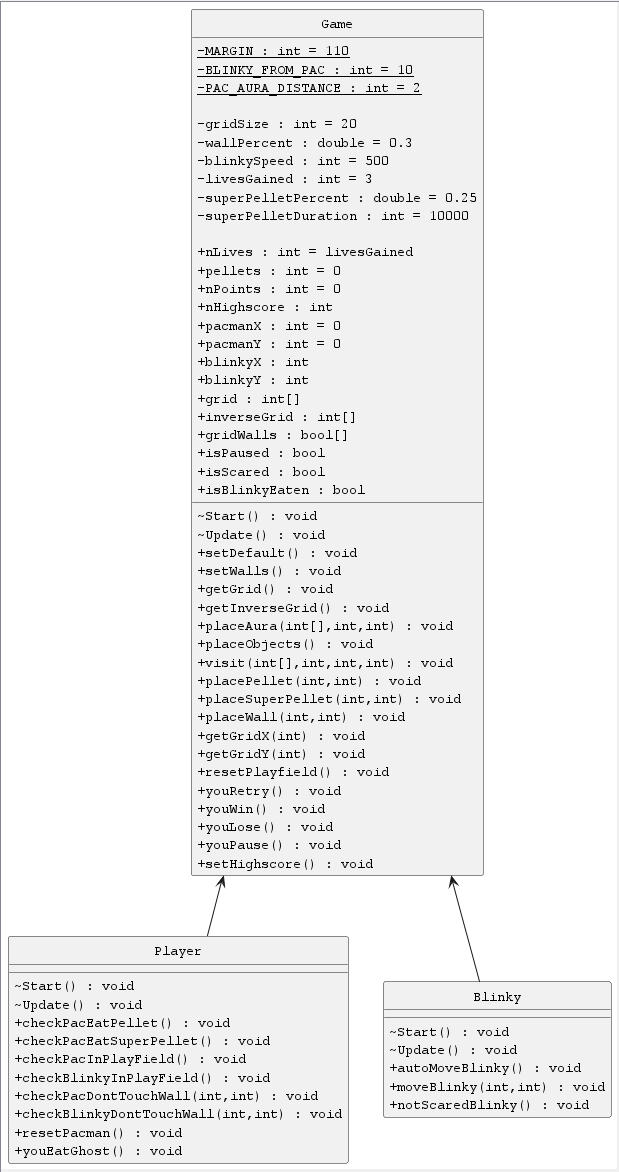


Figura 7 - Diagramma delle classi

Ci sono 3 classi:

1. **Game:** Contiene tutti i parametri modificabili del gioco e decrementa le vite quando il giocatore viene preso.  
   Genera il labirinto e ci inserisce dentro le pillole e super-pillole ed incrementa il punteggio.  
   Resetta i personaggi quando il giocatore od il fantasma viene preso.  
   Gestisce il menu di pausa permettendo al giocatore di uscire o resettare la partita.  
   Controlla che il giocatore vinca, e quindi genera una nuova mappa aumentando vite e punteggio, o perde la partita e quindi manda il giocatore alla pagina di game over.
2. **Player:** Ha dentro i metodi per far mangiare le pillole e le super pillole dal giocatore ed i metodi per controllare che i personaggi rimangano dentro il labirinto quando si muovono.
3. **Blinky:** Ha dentro i metodi per muovere il fantasma AI per il labirinto verso il giocatore o farlo scappare dal giocatore a dipendenza del valore di un boolean.

### Finale

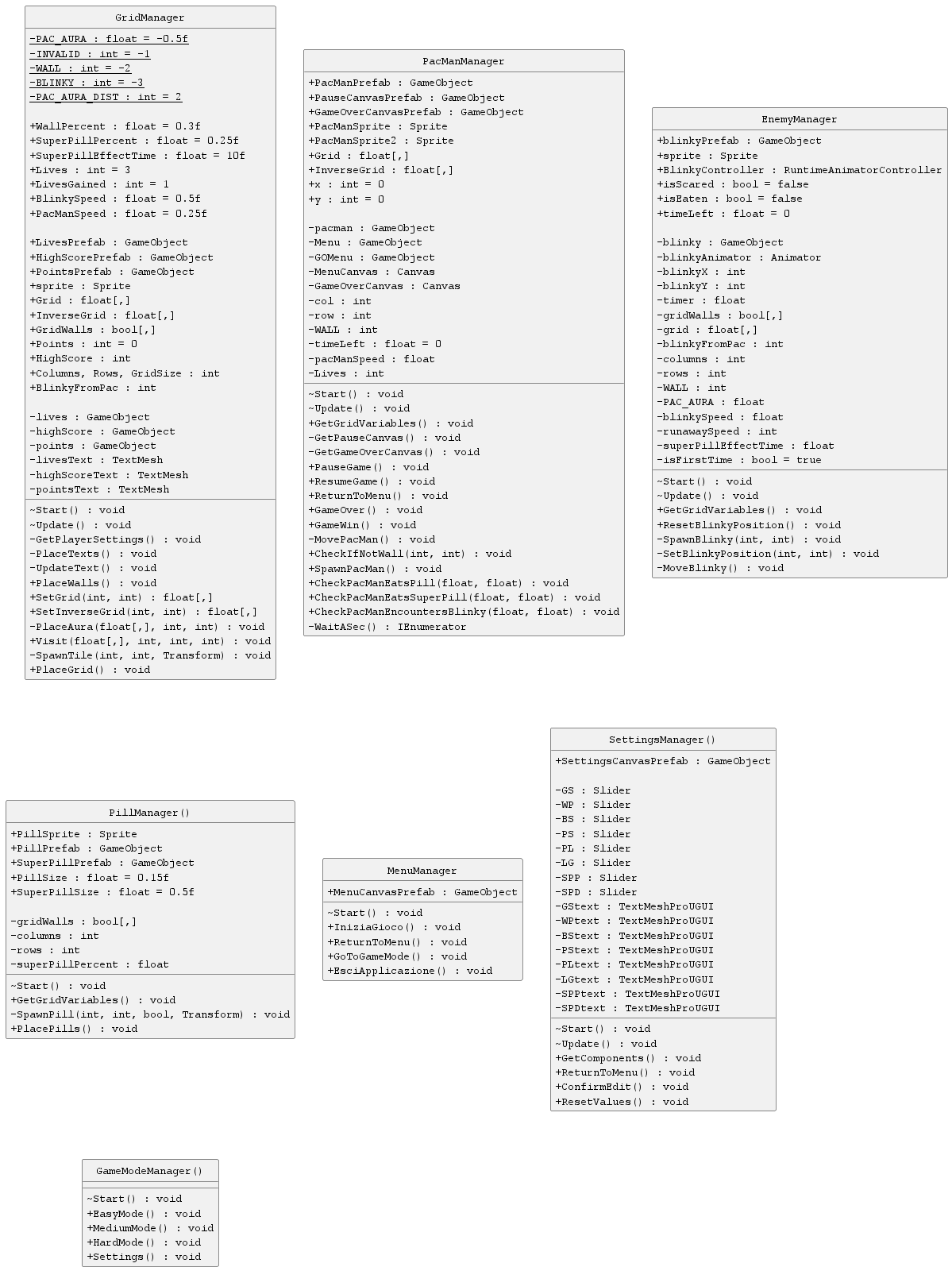


Figura 8 - Diagramma delle classi finale

Nel diagrmma finale ci sono molte più classi:

1. **Game:** Contiene tutti i parametri modificabili del gioco e decrementa le vite quando il giocatore viene preso.  
   Genera il labirinto e ci inserisce dentro le pillole e super-pillole ed incrementa il punteggio.  
   Resetta i personaggi quando il giocatore od il fantasma viene preso.  
   Gestisce il menu di pausa permettendo al giocatore di uscire o resettare la partita.  
   Controlla che il giocatore vinca, e quindi genera una nuova mappa aumentando vite e punteggio, o perde la partita e quindi manda il giocatore alla pagina di game over.
2. **Player:** Ha dentro i metodi per far mangiare le pillole e le super pillole dal giocatore ed i metodi per controllare che i personaggi rimangano dentro il labirinto quando si muovono.
3. **Blinky:** Ha dentro i metodi per muovere il fantasma AI per il labirinto verso il giocatore o farlo scappare dal giocatore a dipendenza del valore di un boolean.

# Implementazione

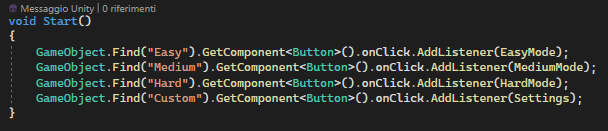
## Grid manager

## Pac-Man manager

## Enemy manager

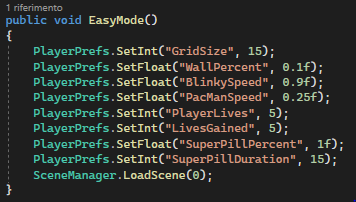
## Game Mode manager

### Start



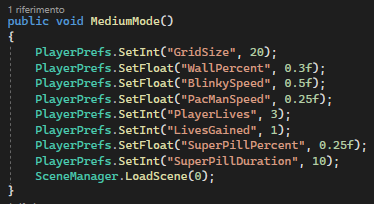
Nel metodo start viene assegnato un Listener a tutti i bottoni per fare in modo che quando un bottone viene premuto viene eseguito il metodo richiesto.

### EasyMode



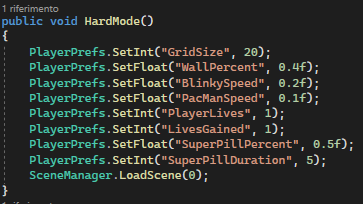
Nel metodo EasyMode vengono impostate le playerPrefs ad i valori per rendere il gioco più facile e poi ti ritorna al menu.

### MediumMode



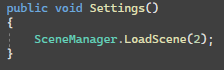
Nel metodo MediumMode vengono impostate le playerPrefs ad i valori per rendere il gioco di difficoltà normale e poi ti ritorna al menu.

### HardMode



Nel metodo HardMode vengono impostate le playerPrefs ad i valori per rendere il gioco più difficile e poi ti ritorna al menu.

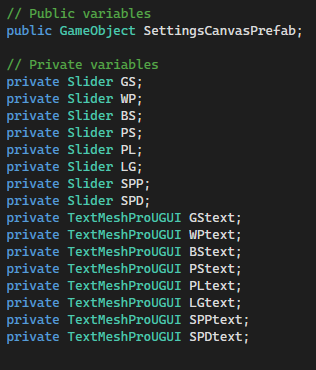
### Settings



Carica la scena per poter modificare le PlayerPrefs manualmente con degli slider.

## Settings manager

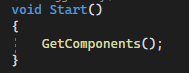
### Variabili



L’unica variabile pubblica è di tipo GameObject per farsi passare la prefab dell’interfaccia.

Poi ci sono 8 variabili private per il valore degli 8 Slider, e ci sono altre 8 variabili di tipo TextMeshProGUI per poter mostrare il valore degli slider su testo di fianco agli slider.

### Start



Il metodo start richiamna il metodo GetComponents.

### GetComponents

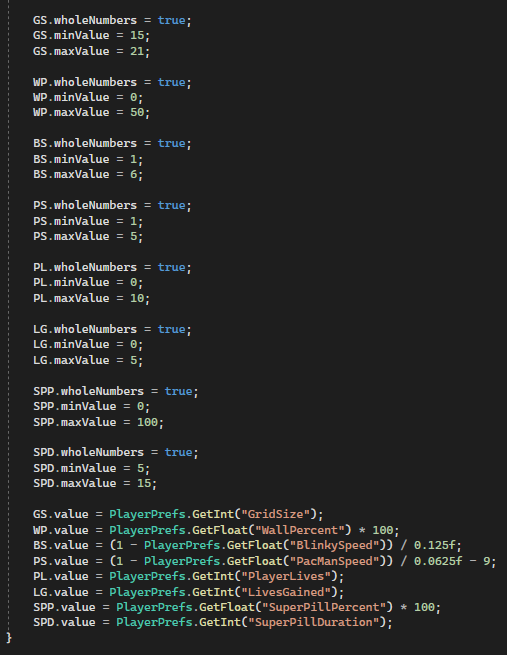


Nel metodo GetComponents viene istanziata la prefab in una variabile con la sua posizione e poi gli viene dato il nome Canvas.

Viene assegnato un Listener a tutti i bottoni per fare in modo che quando un bottone viene premuto viene eseguito il metodo richiesto.

In tutte le variabili di tipo slider viene messo dentro il componente di tipo slider.

In tutte le variabili di testo viene messo dentro il componente di tipo TextMeshProUGUi.

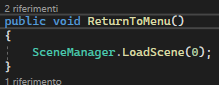


A tutti gli slider viene messo su true il parametro per rendere selezionabili solo numeri interi sullo slider.

Poi viene impostato il valore minimo e massimo per tutti gli slider.

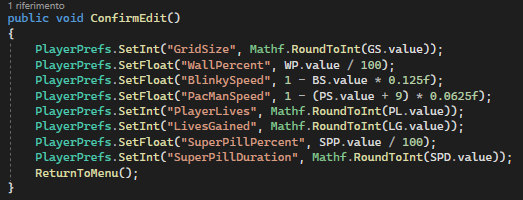
Come valore degli slider al caricamento della pagina vengono presi i valori precedentemente impostati.

### ReturnToMenu



Il metodo ReturnToMenu carica la scena del menu.

### ConfirmEdit



Il metodo ConfirmEdit setta le PlayerPrefs come sono state impostate dall’utente usando gli slider e poi ti ritorna al menu.

### ResetValues



Il metodo Resetvalues setta tutte le PlayerPrefs aî valori di default ed attribuisci lo stesso valore pure agli slider.

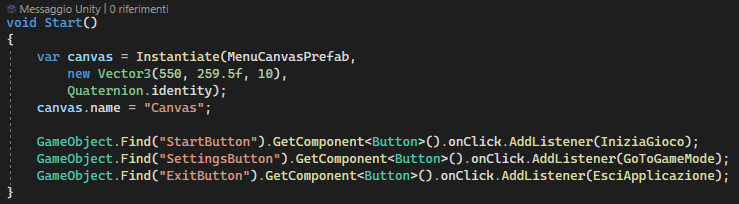
## Menu manager

### Variabili



L’unica variabile che ha è una variabile pubblica di tipo GameObject per farsi passare una Prefab con dentro tutti gli elementi del menu.

### Start

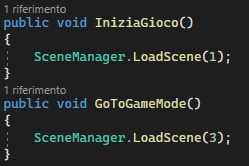


Il metodo start del menu manager inizia istanziando la prefab passata in una variabile di tipo canvas con la sua posizione e poi gli dà il nome Canvas.

Poi assegna un Listener a tutti i bottoni per fare in modo che quando un bottone viene premuto viene eseguito il metodo richiesto.

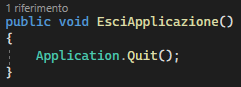
### Update

### Cambio scena



I metodi IniziaGioco e GoToGameMode servono tutti a caricare una scena diversa. IniziaGioco carica la pagina di gioco e GoToGameMode porta alla pagina di scelta della difficoltà del gioco.

### Esci applicazione



Il metodo EsciApplicazione chiude l’applicazione.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-01 | **Nome:** | Mappa Generata Casualmente |
| **Descrizione:** | Verrà generata una mappa senza interruzioni tra il percorso. | | |
| **Procedura:** | 1. Fare partire l'applicativo in Unity. 2. Avviare una partita. 3. Controllare che la mappa sia stata generata correttamente. | | |
| **Risultati attesi:** | Una mappa casuale senza punti irraggiungibile in cui le mura bloccano il passaggio. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-02 | **Nome:** | Movimento Pac-Man |
| **Descrizione:** | Muovere Pac-Man usando i tasti sulla tastiera. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. | | |
| **Risultati attesi:** | Il Pac-Man si muove in 4 direzioni usando i tasti specificati precedentemente. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-02 | **Nome:** | Movimento Pac-Man Attraverso il Labirinto |
| **Descrizione:** | Muovere Pac-Man dentro la mappa. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. 4. Andare contro un muro in tutte le 4 direzioni | | |
| **Risultati attesi:** | Pac-Man si muove in 4 direzioni usando i tasti specificati precedentemente, però non può entrare nei muri della mappa. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-03 | **Nome:** | Piazzamento Pillole nella Mappa |
| **Descrizione:** | Vengono piazzate pillole su ogni cella della mappa. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Controllare la mappa | | |
| **Risultati attesi:** | Le pillole appaiono solo nelle celle della mappa libere e non dentro le mura. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-03 | **Nome:** | Consumazione delle pillole da Pac-Man |
| **Descrizione:** | Pac-Man consuma le pillole quando sale sulla loro cella. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. 4. Mangiare qualche pillola | | |
| **Risultati attesi:** | Quando Pac-Man sale su una cella con una pillola e poi cambia cella la cella di prima sarà vuota. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-04 | **Nome:** | Avvicinamento del Fantasma AI al Pac-Man Statico. |
| **Descrizione:** | Il fantasma si avvicina, blocco per blocco, al giocatore Pac-Man fermo. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Controllare che il fantasma si avvicina di un blocco ogni x secondi (variabile) al Pac-Man. | | |
| **Risultati attesi:** | Il fantasma si avvicina a Pac-Man fermo blocco per blocco ogni x secondi. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-04 | **Nome:** | Avvicinamento del Fantasma AI al Pac-Man Dinamico |
| **Descrizione:** | Il fantasma si avvicina, blocco per blocco, al giocatore Pac-Man che si muove. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. 4. Controllare che il fantasma si avvicina di un blocco ogni x secondi (variabile) al Pac-Man. | | |
| **Risultati attesi:** | Il fantasma si avvicina a Pac-Man che si sta spostando blocco per blocco ogni x secondi. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-04 | **Nome:** | Avvicinamento del Fantasma AI al Pac-Man dentro le mura della mappa. |
| **Descrizione:** | Il fantasma si avvicina, blocco per blocco, al giocatore Pac-Man che si muove attraversando i percorsi del labirinto. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Controllare che il fantasma si avvicina blocco per blocco ogni x secondi (variabile) al Pac-Man senza uscire dalla mappa o scontrandosi contro le mura. | | |
| **Risultati attesi:** | Il fantasma si avvicina a Pac-Man che si sta spostando blocco per blocco ogni x secondi percorrendo il labirinto senza passare sopra le mura. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-009  REQ-05 | **Nome:** | Consumazione della Super-Pillola Invoca l’invincibilità. |
| **Descrizione:** | Le Super-Pillole dopo che sono consumate dal Pac-Man gli danno l'invincibilità' contro il fantasma per x secondi (variabile). | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. 4. Salire su un blocco con una Super-Pillola. | | |
| **Risultati attesi:** | Pac-Man diventa invincibile per x secondi. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-010  REQ-05 | **Nome:** | Consumazione della Super-Pillola vittimizza il fantasma. |
| **Descrizione:** | Le Super-Pillole dopo che sono consumate dal Pac-Man fanno scappare il fantasma per x secondi (variabile). | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. 4. Salire su un blocco con una Super-Pillola. | | |
| **Risultati attesi:** | Il fantasma inizia a scappare da Pac-Man senza uscire dalla mappa o scontrarsi contro le mura. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-011  REQ-06 | **Nome:** | Conteggio punteggio. |
| **Descrizione:** | Ogni volta che una pillola viene consumata il punteggio mostrato accanto incrementa di uno. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. 4. Salire su un blocco con una pillola. | | |
| **Risultati attesi:** | Il punteggio mostrato accanto incrementa di un punto. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-012  REQ-06 | **Nome:** | Salvataggio punteggio. |
| **Descrizione:** | Se il punteggio ottenuto è superiore all’highscore viene salvato come nuovo punteggio massimo. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. 4. Salire su un blocco con una pillola. 5. Farsi prendere dal fantasma fino a quando si finiscono le vite | | |
| **Risultati attesi:** | Viene salvato e mostrato il punteggio migliore del giocatore dopo che perde. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-013  REQ-07 | **Nome:** | Vittoria. |
| **Descrizione:** | Quando tutte le pillole sono consumate, una nuova mappa viene generata dove i personaggi vengono ripristinati e il giocatore ottiene x vite (variabile). | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. 4. Salire su tutti i blocchi della mappa con su una pillola. | | |
| **Risultati attesi:** | Una nuova mappa viene generata dove i personaggi vengono ripristinati e il giocatore ottiene x vite (variabile). | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-014  REQ-07 | **Nome:** | Perdita vita. |
| **Descrizione:** | Quando Pac-Man viene preso dal fantasma perde una vita se ne ha ancora e viene ricaricato nel punto di partenza. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. 4. Farsi prendere dal fantasma. | | |
| **Risultati attesi:** | Il conteggio delle vite mostrato sulla pagina diminuisce di uno e Pac-Man ed il fantasma vengono ricaricati al punto di partenza. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-015  REQ-07 | **Nome:** | Perdita partita. |
| **Descrizione:** | Ogni volta che Pac-Man viene toccato perde una vita, quando perde tutte le sue vite il gioco finisce. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Usare i diversi tasti seguenti per fare muovere Pac-Man, in diverse direzioni, di un blocco sulla mappa:    * Tasto "w" o freccia in su: sposta in su.    * Tasto "a" o freccia a sinistra: sposta a sinistra.    * Tasto "s" o freccia in giù: sposta in giù.    * Tasto "d" o freccia a destra: sposta a destra. 4. Farsi prendere dal fantasma fino a quando non si ha più vite. | | |
| **Risultati attesi:** | Il gioco finisce e appare un bottone per ricominciare la partita oppure uscire dal gioco. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-016  REQ-07 | **Nome:** | Pausa. |
| **Descrizione:** | Cliccando un bottone si mette in pausa il gioco. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Avviare una partita. 3. Cliccare il tasto “esc”. 4. Selezionare “Retry”. 5. Cliccare il tasto “esc”. 6. Selezionare “Exit”. | | |
| **Risultati attesi:** | In partita premendo “esc” si mette in pausa il gioco, poi premendo su “Retry” riinizia la partita e premendo su “Exit” si trona al menu. | | |

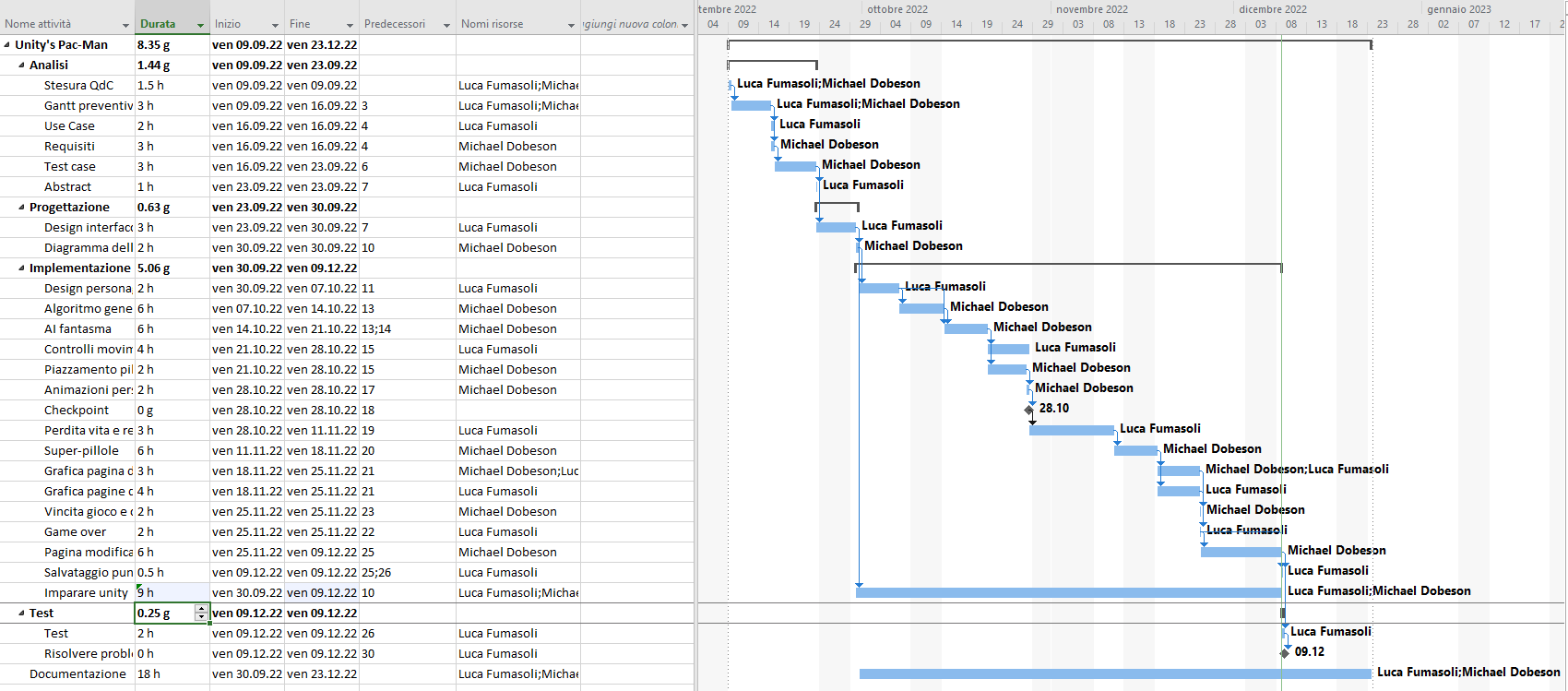
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-017  REQ-10 | **Nome:** | Controller. |
| **Descrizione:** | Si può collegare un gaming controller per giocare. | | |
| **Procedura:** | 1. Fare partire L'applicativo in Unity. 2. Collegare il controller al dispositivo usato per giocare. 3. Navigare per il menu con il joystick/ 4 freccette direzionali. 4. Premere “X” per confermare | | |
| **Risultati attesi:** | Si può navigare il menu, avviare la partita e giocare usando il controller. | | |

## Risultati test

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Risultato | Commento | Data |
| TC-001 | Passato | La mappa viene generata correttamente.    Figura 9 - Mappa generata | 09.12.2022 |
| TC-002 | Passato | Pac-Man si muove in tutte e quattro le direzioni alla premuta dei tasti. | 09.12.2022 |
| TC-003 | Passato | Se Pac-Man va contro un muro in qualsiasi direzione non gli viene permesso di andare avanti e rimane fermo. | 09.12.2022 |
| TC-004 | Passato | Viene piazzata una pillola su ogni cella della mappa che non è un muro od una super-pillola. | 09.12.2022 |
| TC-005 | Passato | Una volta avviata la partita quando Pac-Man sala su una pillola viene mangiata e non compare più sulla mappa ed il punteggio aumenta.    Figura 10 - Manga pillole | 09.12.2022 |
| TC-006 | Passato | Il fantasma si avvicina a Pac-Man ogni tot secondi definiti da una variabile quando il giocatore rimane fermo. | 09.12.2022 |
| TC-007 | Passato | Il fantasma si avvicina a Pac-Man ogni tot secondi definiti da una variabile e quando il giocatore si muove il fantasma lo segue e gli va sempre contro. | 09.12.2022 |
| TC-008 | Passato | Il fantasma si avvicina a Pac-Man ogni tot secondi definiti da una variabile attraverso il labirinto senza andare mai a sbattere od attraversare le mura. | 09.12.2022 |
| TC-009 | Passato | Una volta che viene mangiata una Super-Pillola Pac-Man diventa invincibile. | 09.12.2022 |
| TC-010 | Passato | Quando Pac-Man mangia una Super-Pillola il fantasma cambia animazione ed inizia a scappare dal giocatore.    Figura 11 - Fantasma vittimizzato | 09.12.2022 |
| TC-011 | Passato | Il punteggio incrementa di 1 ogni volta che una pillola viene mangiata.    Figura 12 - Incremento punteggio | 09.12.2022 |
| TC-012 | Passato | Se si ottiene un punteggio più alto dell’highscore questo viene salvato come nuovo highscore e viene mostrato in nuove partite anche se si riavvia l’applicazione. | 09.12.2022 |
| TC-013 | Passato | Una volta che vengono mangiate tutte le pillole viene generata una nuova mappa dove si tiene il punteggio di prime e le vite aumentano della quantità settata nelle variabili.    Figura 13 - Generazione nuova mappa | 09.12.2022 |
| TC-014 | Passato | Quando si viene preso dal fantasma viene rimossa una vita ed il giocatore riparte dalla cella iniziale. | 09.12.2022 |
| TC-015 | Passato | Quando si viene presi dal fantasma e si hanno finito le vite la partita finisce ed appare la schermata di game over con il proprio punteggio e l’highscore.    Figura 14 - Game Over | 09.12.2022 |
| TC-016 | Passato | Quando si preme esc appare la schermata di pausa e premendo su resume la partita continua mentre se si preme su main menu porta al menu principale correttamente.    Figura 15 - Pausa | 09.12.2022 |

## Mancanze/limitazioni conosciute

# Consuntivo



# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc.

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

# Glossario

Inserite una semplice tabella con due colonne che spieghi i termini specifici del progetto (lista dei termini in ordine alfabetico A-Z)

Esempio:

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| AJAX | **Asynchronous JavaScript And XML**: una tecnica che permette di eseguire richieste ed ottenere dati da una pagina web in modo asincrono. |
| CSS | **Cascading Style Sheets**: linguaggio che permette di definire il layout e la grafica di una pagina web. |

# Bibliografia

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.