uniPAC-MAN

documentatione

**realizzato da**

**bellosi jacopo ( m..1081058 )**

**longhi lara ( m. 1079261 )**

**poloni Lluca ( m. 1078817 )**

*Indice*

[1. Requirements specification (Standard IEEE 830) 3](#_Toc154826105)

[1.1 Introduzione 3](#_Toc154826106)

[1.1.1 Obiettivo 3](#_Toc154826107)

[1.1.2 Scopo 3](#_Toc154826108)

[1.1.3 Definizioni, acronimi e abbreviazioni 3](#_Toc154826109)

[1.2 Descrizione generale 3](#_Toc154826110)

[1.2.1 Prospettiva del prodotto 3](#_Toc154826111)

[1.2.2 Funzioni del prodotto 3](#_Toc154826112)

[1.2.3 Caratteristiche dell’utente 3](#_Toc154826113)

[1.2.4 Vincoli 3](#_Toc154826114)

[1.2.5 Presupposti e dipendenze 3](#_Toc154826115)

[1.3 Requisiti specifici 3](#_Toc154826116)

[1.3.1 Requisiti dell’interfaccia esterna 3](#_Toc154826117)

[1.3.2 Richieste funzionali 4](#_Toc154826118)

[1.3.3 Requisiti di prestazione 4](#_Toc154826119)

[1.3.4 Vincoli di progettazione 4](#_Toc154826120)

[2. Software lifecycle 4](#_Toc154826121)

[3. Configuration management 4](#_Toc154826122)

[4. People management 4](#_Toc154826123)

[5. Software quality 4](#_Toc154826124)

[6. Requirement engineering 4](#_Toc154826125)

[7. Modeling 4](#_Toc154826126)

[7.1 Il diagramma delle classi 4](#_Toc154826127)

[7.2 Il diagramma della macchina a stati 4](#_Toc154826128)

[7.3 Il diagramma di sequenza 4](#_Toc154826129)

[7.4 Il diagramma di comunicazione 4](#_Toc154826130)

[7.5 Il diagramma dei componenti 4](#_Toc154826131)

[8. Software architecture 4](#_Toc154826132)

[9. Software design 4](#_Toc154826133)

[10. Software testing 4](#_Toc154826134)

[11. Software maintenance 4](#_Toc154826135)

1. Requirements specification (Standard IEEE 830)

Prima di procedere con lo sviluppo del prodotto software, è necessario esplicitare i requisiti alla base del progetto. La specifica è strutturata secondo lo standard IEEE830.

* 1. *Introduzione*

Il progetto consiste nello sviluppare un’applicazione Java per Windows che consenta all’utente di giocare a una versione rivisitata del gioco Pac-Man, la rivisitazione consiste in variazioni grafiche che rappresentano il tema universitario.

* + 1. Obiettivo

L’ obiettivo principale del gioco Pac-Man è mangiare tutti i CFU sparsi nel labirinto. Nel far ciò c’è la possibilità di incombere in quattro fantasmi che cercano di catturare Pac-Man. Quando Pac-Man mangia una power-up, i fantasmi diventano vulnerabili e possono essere mangiati da Pac-Man per un breve periodo di tempo. Il gioco continua finché Pac-Man riesce a mangiare tutti i punti nel labirinto o fino a quando perde tutte le vite disponibili.

Il gioco è strutturato su tre livelli che rappresentano gli anni necessari per il raggiungimento di una laurea triennale.

### *Scopo*

Lo scopo del gioco è ottenere il punteggio più alto possibile mangiando CFU, power-up e fantasmi. Ogni CFU mangiato contribuisce al punteggio del giocatore. Oltre ai punti normali, ci sono anche punti bonus ottenuti mangiando i power-up e a seguito i fantasmi.

* + 1. Definizioni, acronimi e abbreviazioni
* CFU: Crediti formativi universitari, rappresentati dal simbolico pallino verde presente all’interno del libretto elettronico una volta superato l’esame;
* Power-up: oggetto che conferisce una particolare abilità temporanea, nel nostro specifico caso è rappresentato dallo youtuber Elia Bombardelli, conosciuto per il suo canale YT riguardante spiegazioni di argomenti di matematica.

## *Descrizione generale*

### *Prospettiva del prodotto*

### *Funzioni del prodotto*

### *Caratteristiche dell’utente*

L’applicazione è facile e intuita, quindi non necessita di preparazione pregressa, qualunque tipo di utente può giocarci e comprendere velocemente le funzionalità del gioco.

### *Vincoli*

### *Presupposti e dipendenze*

## *Requisiti specifici*

### *Requisiti dell’interfaccia esterna*

L’interfaccia utente prevede una schermata iniziale dove l’utente potrà decidere se iniziare una nuova partita o uscire dalla schermata. A seguito della pressione del tasto *enter* in relazione a “NUOVA PARTITA” apparirà la schermata di gioco dove l’utente potrà muovere il Pac-Man con AWSD.

Immagine che contiene testo, software, schermata, Software multimediale

Descrizione generata automaticamenteImmagine che contiene testo, computer, schermata, software

Descrizione generata automaticamenteImmagine che contiene testo, computer, software, schermata

Descrizione generata automaticamenteImmagine che contiene testo, computer, software, schermata

Descrizione generata automaticamente

### *Richieste funzionali*

### *Requisiti di prestazione*

Non sono richieste risorse computazionali elevate per questo applicativo, non necessità nemmeno di una connessione internet.

### *Vincoli di progettazione*

1. Software lifecycle

Il modello di processo scelto per lo sviluppo dell’applicativo è un metodo agile nello specifico eXtreme Programming. La scelta di un process model di tipo AGILE ci permette di non preoccuparci troppo dei pieni ma di adattarci ai cambiamenti e agli sviluppi che notiamo necessari in corso d’opera.

Nell’ambito specifico dell’ eXtreme Programming (XP) troviamo l’utilizzo in forte misura di diverse pratiche:

* Whole team, tutto il gruppo è stato coinvolto in tutte le fasi del lavoro;
* Cliente in loco, noi siamo sia utente che parte della squadra;
* Pair programming, il codice spesso verrà scritto su una sola macchina con supporto da remoto di un altro programmatore che lo aiuta e lo corregge;
* Proprietà collettiva, tutti hanno accesso a tutto e possono modificarlo in qualsiasi momento;
* Sviluppo guidato dei test.

1. Configuration management

Come strumento per la gestione della configurazione abbiamo utilizzato GitHub, il quale ci ha permesso di tenere traccia di tutti gli artefatti realizzati, e delle modifiche effettuate dagli altri componenti del gruppo. Ogni commit con annessa descrizione della modifica/progresso rappresenta i cambiamenti eseguiti nel codice.

1. People management

Essendo un’ambiente prettamente dedicato allo sviluppo software, possiamo individuare un’organizzazione a matrice dove, singolarmente o con supporto da parte di un altro gruppo abbiamo elaborato le varie funzionalità.

|  |  |  |  |
| --- | --- | --- | --- |
| Funzioni\Persone | Bellosi | Longhi | Poloni |
| Costruzione Panel | x | x |  |
| Grafica |  | x | x |
| Pac-Man | x |  |  |
| Fantasmi | x |  |  |
| Arena |  |  | x |
| Tunnel |  | x |  |
| Vite | x | x |  |
| Stati di gioco |  | x | x |
| Power-up |  |  | x |
| Score Manager | x |  |  |
| Suoni | x | x | x |
| Livelli |  |  | x |

1. Software quality
2. Requirement engineering
3. Modeling

## *7.1 Il diagramma delle classi*

## *7.2 Il diagramma della macchina a stati*

## *7.3 Il diagramma di sequenza*

## *7.4 Il diagramma di comunicazione*

## *7.5 Il diagramma dei componenti*

1. Software architecture
2. Software design
3. Software testing
4. Software maintenance