

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE

DIPARTIMENTO DI SCIENZE E TECNOLOGIE

CORSO DI PROGRAMMAZIONE 3 E LABORATORIO DI PROGRAMMAZIONE 3



Proponenti:

Salvatore De Rosa, Matricola:0124001981

Mario de Leo, Matricola:0124001945

Progetto Proposto: Pacman revised

A.A. 2021-22

DESCRIZIONE DELLE CARTELLE DEL PROGETTO

Il progetto dispone delle seguenti cartelle

- Src

Vi sono le cartelle com dove vi sono i file Java, img dove vi sono le immagini, sound dove si sono i suoni e javadoc dove vi è la documentazione generata, e infine uml dove vi contengono i diagrammi.

- Out

Vi sono i file .class che contengono bytecode Java.

- Lib

Dove all'interno vi è il file .jar per mysql.

N.B: **per il corretto utilizzo del DB cambiare il path con il vostro.**

DESCRIZIONE DEL PROGETTO

Si è voluti progettare il gioco di pacman, ma con una piccola variazione della modalità di gioco.

Il gioco prevede come obiettivo che la nostra pallina gialla mangi tutte le palline e raggiungere il punteggio più alto possibile, inoltre, il pacman, dovrà evitare di essere toccato dai fantasmi, e soprattutto, di essere il più veloce possibile prima che l'apina possa finire tutti i pallini.

Il campo di gioco è costituito da:

- **PACMAN:**

Come già citata prima la nostra pallina gialla affamata di palline, con l'unico scopo di sopravvivere e avere la pancia piena.

- **BLINKY:**

Il fantasma di colore rosso, situato a sinistra, non mangerà le palline ma vorrà mangiare il pacman.

- **CLYDE:**

Il fantasma di colore arancione, situato in alto a destra, avrà lo stesso obiettivo di Blinky.

- **INKY:**

Il fantasma di colore ciano, situato in alto al centro, avrà lo stesso obiettivo di Blinky.

- **PINKY:**

Il fantasma di colore rosa, situato in basso a destra, avrà lo stesso obiettivo di Blinky.

- **HONEY:**

L'apina paffuta, posizionata nella posizione opposta di pacman, con l'obiettivo di rubare tutti pallini e di abbassarci lo score.

Pacman potrà mangiare i vari tipi di pallini e anche i fantasmi che garantiranno un punteggio che saranno:

- **PACGUM: 10**
- **SUPER PACGUM: 25**
- **FANTASMI SPAVENTATI: 50**

Se invece sarà Honey a mangiare le palline allora avremo un decremento dello score:

- **PACGUM: -5**
- **SUPER PACGUM: -10**

La partita potrebbe terminare in uno dei seguenti casi:

1. Pacman mangia tutti i pallini sopravvivendo dall'attacco dei fantasmi.
2. Pacman verrà mangiato da uno dei 4 fantasmi non spaventati.
3. Honey avrà mangiato tutte le palline.

DESCRIZIONE DELLE CLASSI

N.B. Si è preferito descrivere le altre classi successivamente per avere una maggiore interazione con i pattern utilizzati.

- **MAIN:** La classe da cui avviare il progetto.
- **GAME:** Dove verranno istanziate molte classi sottostanti, setterà il menù nel suo stato iniziale e soprattutto dove disegna i vari aspetti del gioco.
- **MENU:** Dove grazie all'estensione "MouseAdapter" con vari click del mouse potremo esplorare le diverse zone, se il mouse clickerà su "Gioca" avvierà il nostro gioco, altrimenti su "Score" per guardare i nostri score, e infine "Esci" per terminare il programma.
- **PACMAN:** Per potersi muovere, richiamerà la classe Keyboard_input, dopodiché, grazie a movepacman() simuleremo i nostri movimenti e collisioni, si è deciso di utilizzare il "Rectangle" per avere una maggiore precisione con le collisioni con i fantasmi, una volta finita la partita si avvierà il metodo Reset() per resettare le posizioni e il relativo score, checkmaze() serve per controllare se il pacman ha mangiato tutti i pallini.
- **GHOST:** Si è deciso d'implementare l'ActionListener per abbassare l'eccessiva velocità dei fantasmi, i fantasmi possiedono l'enum state per poter cambiare da "spaventato" a "nospaventato" o viceversa, per far muovere il fantasma si è deciso di utilizzare un random nel metodo moveghost(), la classe sarà estesa a Blinky,Clyde,Inky,Pinky attraverso "super".
- **BEE:** Si è deciso d'implementare l'ActionListener per abbassare l'eccessiva velocità dell'ape, anche lei con movebee() farà movimenti randomici e quando si troverà nella posizione corrispondente, mangerà i pacgum o il superpacgum, quando mangerà il superpacgum si muoverà più veloce.
- **DB:** Si è deciso di usare il singleton per avere una sola istanza del DB per tutta la durata dell'esecuzione del programma, attraverso l'URL e il DRIVER effettueremo la connessione per effettuare le operazioni di "insert" e "select".
- **MAZE:** Il labirinto avrà diversi numeri per identificare i vari oggetti nella mappa, 1 pacgum,2 superpacgum,3 linea di destra, 4 linea di sinistra, 5 linea di sopra,6 muri interni, 7 linea di sotto, 0 non apparirà nulla e non avrà interazione di alcun genere, si è deciso di usare numeri diversi a causa di un non corretto funzionamento del "drawline".
- **LOADIMG:** Classe che caricherà tutte le immagini del progetto.
- **WINDOW:** Classe dove metteremo il size della nostra finestra grazie al package di swing.

- **KEYBOARD_INPUT**: Classe che implementa “KeyListener” ossia impostiamo i tasti per i movimenti del pacman.
- **SOUND**: Attraverso l’import della libreria javax.sound.sampled implementiamo i suoni.
- **UTENTE**: Impostiamo un ID,username e punteggio per il DB.

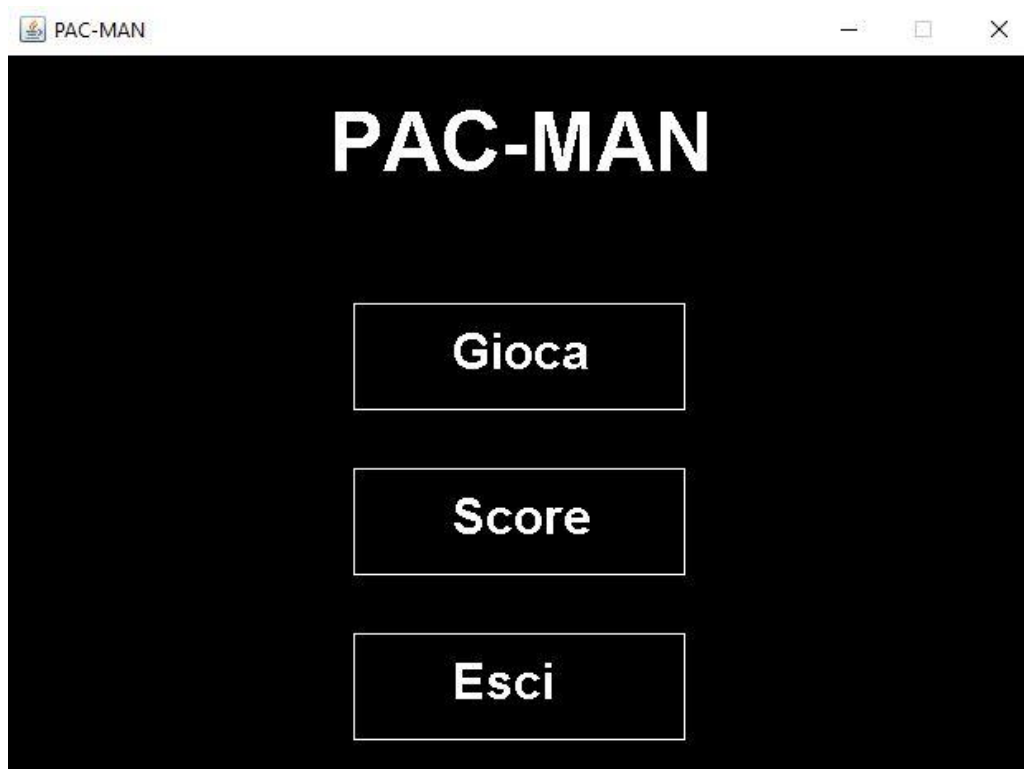
DESCRIZIONE DEI PATTERN UTILIZZATI

- **STRATEGY**: Questo pattern permette di definire il comportamento dei fantasmi utilizzato attraverso l’interfaccia IAGhost e che successivamente sarà implementati ai 4 fantasmi con 4 diversi tipi di strategia, se il fantasma si troverà ad una certa distanza da pacman lo inseguirà, altrimenti se spaventato scapperà disorientando il pacman con dei movimenti al contrario.
- **STATE**: Questo pattern è stato innestato alla classe astratta GhostState, dove si compone in: HouseMode la posizione di partenza dei fantasmi e che una volta mangiati ritorneranno a casa, ChaseState dove appena pacman si avvicinerà i fantasmi passeranno all’attacco e FrightenedState dove dopo che pacman ha mangiato il superpacgum i fantasmi passeranno allo “stato” spaventato e con l’utilizzo di un timer avremo il tempo in cui rimarranno spaventati, finito il timer ritorneranno nel loro stato “nospaventato”, si è deciso di impostare in ChaseState nell’eventualità in cui il pacman si trovasse nelle vicinanze così che i fantasmi possano inseguirlo.
- **PROXY**: Il nostro placeholder, utilizzato per le operazioni di “vedi” e “aggiungi” per evitare che altre classi potessero accedere ai dati sensibili del DB.
N.B. si è deciso di usare solamente 5 persone nella sezione High-score solamente per un fattore estetico.
- **FACTORY PATTERN**: Pattern utilizzato per disegnare i pacgum e i superpacgum presenti nella classe Maze e che richiameranno le classi PacGum e SuperPacGum e l’interfaccia DrawPoint per poter disegnare questi oggetti.

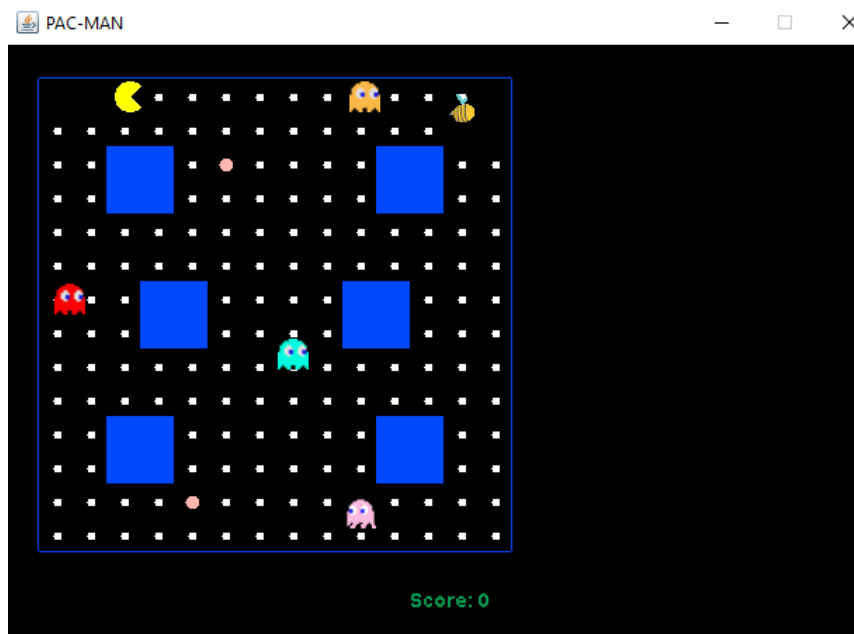
COMPILAZIONE

Per poter compilare e eseguire il gioco basta semplicemente, dalla finestra di intellij avviare con il bottone “Run” nel main il progetto.

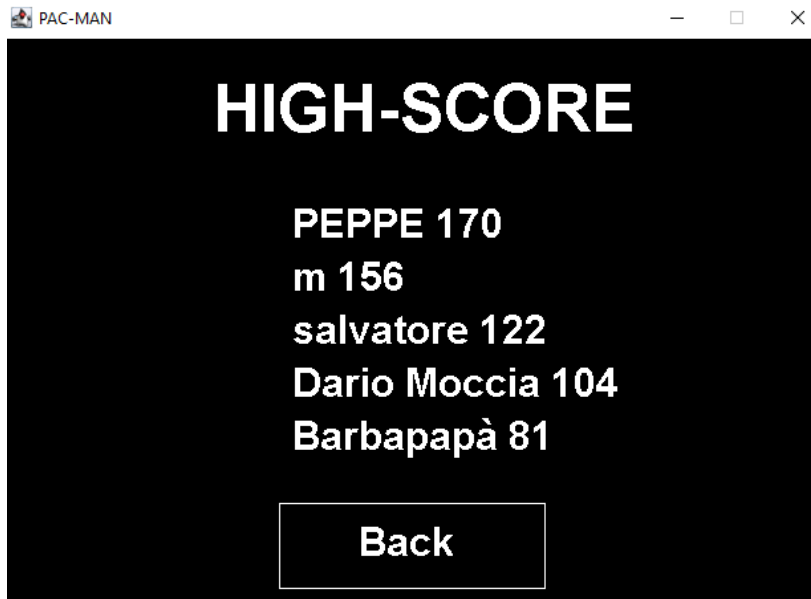
ESEMPIO DI ESECUZIONE



Come già accennato prima, abbiamo la schermata iniziale del menù, con le 3 voci Gioca, Score e Esci.



Schermata del gioco in esecuzione, come si nota lo score è settato a 0 perché Honey ha mangiato lo stesso numero di palline del pacman.



Qui possiamo vedere la schermata dello score dove ci sono le prime 5 persone che hanno ottenuto il punteggio più alto.

DIAGRAMMA DELLE CLASSI

