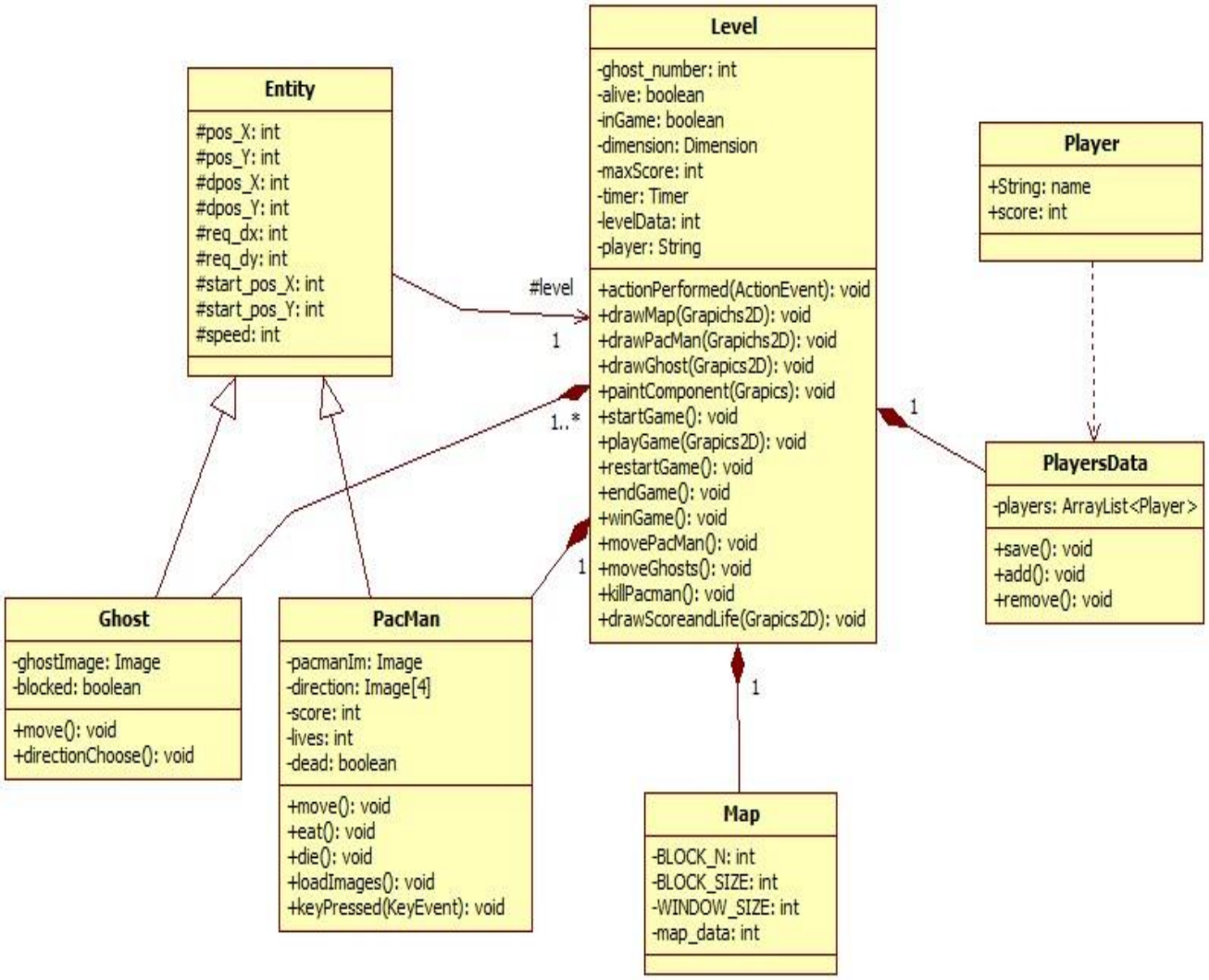
Pacman játék

Dokumentáció

by Petruska Bence (JP5JDU)

Osztálydiagramok:



Osztályok, metódusok:

Entity

A PacMan játékban lévő „egyedeket” valósítja meg, melyeket látunk a játék során.

Minden egyednek van:

- x, y koordinátája: pos_X, pos_Y
- x, y irányba való mozgása: dpos_X, dpos_Y
- x, y irányba kért mozgás: req_dx, req_dy
- x, y start pozíció: start_pos_X, start_pos_Y
- gyorsasága, osztójának kell lennie egy blokk méretének: speed
- Egy szint amihez tartozik: level

Metódusai közt található konstruktor, getterek és setterek.

PacMan

Maga a PacMan a játékban. Entityből származik le.
Attribútumai:

- Az éppen megjelenő képe: pacmanIm
- Minden irányba való forduláskor új kép töltődik be. 4db irány, 4 elemű Image tömb: direction[]
- Az összegyűjtött pontok száma: score
- A hátramaradt életek száma (kezdetben 3): lives
- Boolean, ami megmondja, hogy halott-e a PacMan vagy sem: dead

Metódusai:

- PacMan(int startPoint_X , int startPoint_Y, int sp)
  Konstruktor. Kezdőpozíció, ahol startol, illetve a speed inicializálása. Betölt egy kezdő képet és feltölti az életet 3-ra.
- void move()
  A PacMan mozgatásáért felel. Mivel a játék konstruálása, úgy néz ki, hogy minden objektum kirajzolása a bal felső sarkától indul, ezért mindennek a pozíciója igazából a bal felső sarkának a pozíciója. Emiatt kell külön esetekre bontani, hogy egy objektum milyen irányba halad, mert más-más irányban, máshogy ütközik bele a falakba. Ha beleütközik egy falba, megáll.
- void eat()
  A bogyók evéséért felel. Az előbb említett probléma során itt is külön kell kezelni az irányokat. Növeli a pontot és átállítja a level-nek a levelData-t 2-re, így onnan eltűnik a kirajzolt bogyó.
- void die()
  Meghal a PacMan. Csökkenti az életet, ha elfogyna, akkor átállítja a dead booleant true-ra
- void loadImages()
  Betölti a képeket a direction-be.
- void keyPressed(KeyEvent e)
  A nyilakkal való mozgatásért felel. A req_d-ket és a pacmanIm-t állítja. Amelyik irányba menne, abba az irányba fordul.

Ghost

A játékban lévő szellemek megvalósítása. Az Entity-ből származik le

Attribútumai:

- A betöltött szellem képe: ghostImage
- Boolean ami jelzi, ha a szellem egyhelyben állna: blocked

Metódusai:

- Ghost(int startPoint_X , int startPoint_Y, int sp)
  Konstruktor. Kezdőpozíciók, kép betöltése, illetve kér egy induló irányt jobbra, hogy elinduljon
- void move()
  Hasonló problémák, mint a PacMan mozgásánál. Különbség az irány választása.
- void directionChoose()
  A pacman-t mi mozgatjuk, a szellemek random mozognak. Ha folyosón mozognak akkor folytatják az eddigi irányt, de ha elérnek egy elágazáshoz, random irányba fordul tovább: jobbra, balra, egyenesen. Ha megállna, akkor átállíja a blocked-t és akkor megfordul.

Level

A játék futtatásáért felelő osztály. Kirajzolás, mozgatások, ütközés és események kezelése. JPanel leszármazott, így beletehető egy JFrame-be és akkor külön ablakban fut a játék.

Attribútumai:

- A pályája: map
- A pontok listáját tartalmazo PlayersData: scores
- A szellemek száma, 4 beállítva alapból: ghost_number
- A futást ellenörző boolean, kezdetben false: inGame
- Az JPanel méretezéséhez szükséges Dimension: dimension
- A maximálisan elérhető pont, annak ellenörzéséhez, ha megnyernénk a játékot: maxScore
- A szellemek tömbje: ghosts[]
- A pacman: pacman
- A Timer ami a képfrissítéshez szükséges: timer
- Ide másoljuk át a map_data-t, hogy maga a map ne módosuljon:levelData[][]
- Az éppen játszó játékos neve: String

Metódusai:

- public Level(Map m)
  Beállítja a timert és elindítja, inicializáálja a dimension-t és a scorest, meghívja a konstruktorát a pacmannek és szellemeknek. KeyListenerhez adja a pacmant. A pacman és szellemek elhelyezése fix. A szellemek egy sorba kerülnek egymás mellé, onnan indulnak.
- void actionPerformed(ActionEvent e)
  Újrarajzolás
- void drawMap(Graphics2D g)
  A pálya kirajzolása, a falak kék négyzetek, a bogyók egy négyzet közepén elhelyezkedő körök.

- void drawPacMan(Graphics2D g)
  PacMan grafikus kirajzolása.
- void drawGhost(Graphics2D g)
  Szellemek grafikus kirajzolása.
- void drawScoreandLife(Graphics2D g)
  Az aktuális pont és életek kirajzolása a bal alsó sarokba pozícionálva.
- void paintComponent(Graphics g)
  Összesítő kirajzoló, hogy egybe jelenjenek meg az objektumok. A hátteret feltölti fekete színnek
- void startGame()
  Start pozicióba állítja a pacmant és szellemeket.
- void playGame(Graphics2D g)
  Ha inGame vagyunk, akkor meghívja a mozgásért felelős metódusokat.
- void restartGame()
  Újraindítja a játékot, azaz újraépíti a levelt, feltölti az életét a pacmannek, dead booleant falsera állítja, nullázza a pontokat, beállítja, hogy játékban vagyunk és elindítja a játékot (startGame())
- void endGame()
  Ha elfogyna az életünk, meghívódik ez a függvény. Megjelenít egy Joptionpane-t, hogy újra szeretnénk-e kezdeni vagy sem.
- void winGame()
  Ha az összes bogyót megesszük, meghívódik ez a függvény. Ugyanolyan Joptionpane jelenik meg, mint az endGame-nél.
- void movePacMan()
  A pacman „cselekvései” meghívja, a move()-t, eat()-t, és a killPacMan()-t, és ha a megettük az összes bogyót, akkor a winGame-t is.
- void moveGhosts()
  A szellemek mozgatását összesíti és meghívja őket.
- void killPacman()
  Ha a pacman egy mezőn áll egy szellemmel, akkor megöli a szellemet és kezeli azt az esetet, ha elfogyna a PacMan élete.
- void initLeveldata()
  Inicializálja a level_data-t.

Player

A Scoreboard felépítéséhez szükséges szerializálható osztály. Egy játékost reprezentál egy ponttal.

Attribútumai:

- A játékos neve: name
- A játékos pontjai: score

Metódusai:

- Player(String n, int s)
  Konstruktor, inicializálja a namet meg scoret

PlayersData

A listája a játékosoknak. Beolvas és ment a players.dat fájlba. AbstractTableModel leszármazottja, hogy táblázat szerűen meg lehessen jeleníteni

Attribútumai:

- A játékosok listája, statikus adattag, ArrayList<Player>: players

Metódusai:

- PlayersData()
  Konstruktor. Beolvassa a fájlból a listát.
- void save()
  Elmenti a fájlba a listát.
- void add(String name, int score)
  A listához ad egy Playert a megadott paraméterekkel, majd rendezi a listát.
- void remove(int index)
  Töröl egy adott indexű elemet a listából.
- static Comparator<Player> playerComparator
  Csökkenő sorrendet elősegítő komparátor a rendezéshez
- Az AbstractTableModel -hez szükséges metódusok a maradék...

PacMan_Game

A main-t tartalmazó osztály, Jframeből származik le. A program indításáért felel és a megjelenő ablakokért.

Attribútumai:

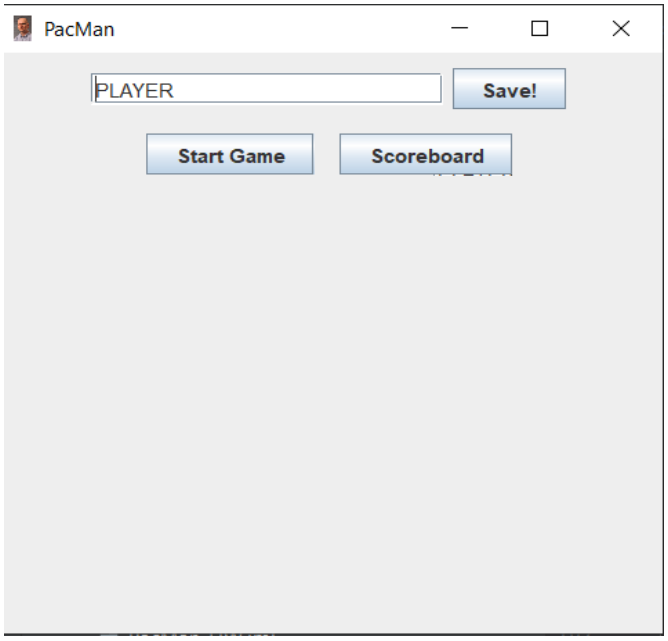
- Egy Jframe amibe a level-kerül, vagyis a futó játék: Game
- Jframe amibe a PlayersData kerül, vagyis a toplista: Scoreboard
- JPanel, ami a menüje a programnak, ez jelenik meg először, amikor megnyitjuk a programot.
- Az aktuális Level: level
- Image, ami az ikonja a programnak: icon
  (Lol egy kis Goldschmidt easter egg)

Metódusai:

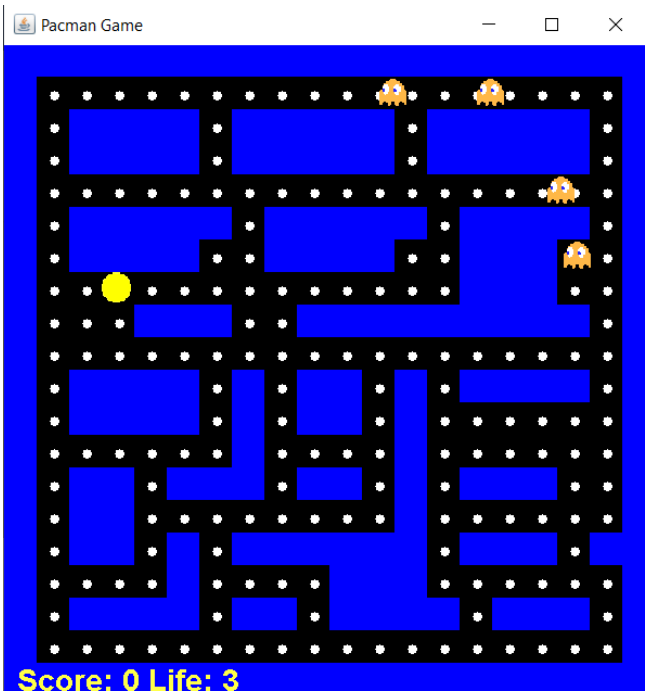
- Pacman_Game(Level l)
  Konstruktor, meghív mindent, felépíti a Jframeket és Jpaneleket.
- static void main(String[] args)
  létrehoz egy Pacman_Game-t

Felhasználói kézikönyv:

1. A program indulásakor egy menü jelenik meg.



2. Ha azt szeretné, hogy majd az elért p pontja elmentődjön a Scoreboardba, akkor írja be a nevét a mezőbe, ahol a PLAYER felirat látható, majd nyomjon a Save! gombra. Ezután nyomjon a Start Game-re
3. Ha a toplistát szeretné megjeleníteni, akkor nyomjon a Scoreboardra.
4. A futó játékban a PACMAN-t a nyilakkal tudjuk irányítani.



5. Ha nyertünk vagy veszítünk van lehetőségünk választani, hogy szeretnénk újra játszani, ellenkező esetben bezárjuk a programot.