

# HÁZI FELADAT

## Programozás alapjai 3.

### Dokumentáció

Pribék Jonatán

P1PGS0

2022. november 28.

---

### TARTALOM

1. Specifikáció eredeti szövege .....	2
2. A kész program módosított szövege .....	3
3. Osztálydiagram.....	4
4. Metódusok leírása.....	5
5. Felhasználói kézikönyv .....	8

# 1. Specifikáció eredeti szövege

## Pac-Man

A program a Pac-Man néven ismert videójátékot valósítja meg. Ebben a játékban a játékos egy sárga fejet irányít egy labirintusban, miközben őt üldöző szellemek elől menekül. Pac-Man a mezőkön elszórt fehér golyókat gyűjti össze, amikkel pontokat szerez a játék során. A pályán véletlenszerűen megjelenhetnek nagyobb, színes golyók, amik erőt adnak Pac-Mannek egy rövid ideig, így ha szellemmel találkozik, akkor a szellem jár pórul, nem a játékos. Ha Pac-Man minden golyót összeszed egy adott pályán, akkor azt a szintet teljesítette, és továbbjut a következő szintre, ahol már gyorsabb ellenségek várják a játékost. A játéknak nincs utolsó szintje, így csak a ranglistán lévő első helyért lehet küzdeni. A játékos 3 élettel kezd, amiket úgy tud elveszíteni, ha a szellemek utoléri Pac-Mant. Ha mindhárom élet elhasználódott, akkor a játéknak vége, és az elért pontszám bekerül a ranglistába.

A program indításakor az ablak tetején egy menürendszer jelenik meg. Itt lehet megtekinteni a ranglistát, megnézni a beállításokat, illetve kilépni a programból. A beállítások menüpontot kiválasztva több opció is megjelenik, amivel jobban személyre szabhatóvá válik a játék. Ilyen például a nehézség kiválasztása, labirintus színe, illetve ranglista betöltése és elmentése.

### Megoldási ötlet vázlatos ismeretete:

A program a Java Swing GUI funkcióit használja fel. A kezdőképernyőt egy BorderLayout-ban lévő JLabel és JTextField készíti el. A menürendszert egy JMenuBar valósítja meg, a bennük lévő JMenu-k és JMenuItem-ek segítségével. Bizonyos elemekhez Listenerek tartoznak, amikkel eseményeket lehet kezelni, illetve meghívják az akkor szükséges metódusokat.

A beállítások közül a ranglista megjelenítését JTable-vel készíttem el, ranglista betöltését és elmentését a JFileChooser segítségével, illetve standard I/O funkciókkal valósítom meg. A nehézségek változtatják meg a játékban lévő szellemek számát és azok sebességét. A játék végén ha ranglistás helyezést ért el a játékos, akkor beírhatja a nevét a megszerzett pontszáma mellé, ezt a nevet egy JOptionPane-el elkészített párbeszédablak elfogadása után lehet megerősíteni.

A játékban Pac-Mant a billentyűzet nyíl gombjaival lehet irányítani, illetve a 'P' (Pause) gomb megnyomásával lehet ideiglenesen leállítani a játékot, amit majd a gomb ismételt megnyomásával lehet a leállított pillanattól folytatni.

## 2. A kész program módosított szövege

### Pac-Man

A program a Pac-Man néven ismert videójátékot valósítja meg. Ebben a játékban a játékos egy sárga fejet irányít egy labirintusban, miközben őt üldöző szellemek elől menekül. Pac-Man a mezőkön elszórt fehér golyókat gyűjti össze, amikkel pontokat szerez a játék során. A pályán véletlenszerűen megjelenhetnek nagyobb, színes golyók, amik erőt adnak Pac-Mannek egy rövid ideig, így ha szellemmel találkozik, akkor a szellem jár pórul, nem a játékos. Ha Pac-Man minden golyót összeszed egy adott pályán, akkor azt a szintet teljesítette, és továbbjut a következő szintre, ahol **több ellenség várja** a játékost. A játéknak nincs utolsó szintje, így csak a ranglistán lévő első helyért lehet küzdeni. A játékos **a nehézségtől függően 2, 3, vagy 4 élettel** kezd, amiket úgy tud elveszíteni, ha a szellemek utoléri Pac-Mant. Ha **az összes élet** elhasználódott, akkor a játéknak vége, és az elért pontszám bekerül a ranglistába, **ha az elérte legalább az ötödik helyezett pontját**.

A program indításakor az ablak tetején egy menürendszer jelenik meg. Itt lehet megtekinteni, **betölteni, vagy elmenteni** a ranglistát, **egy adott pályát betölteni**, és megnézni a beállításokat. A beállítások menüpontot kiválasztva több opció is megjelenik, amivel jobban személyre szabhatóvá válik a játék. Ilyen például **a nehézség vagy a labirintus színe kiválasztása**.

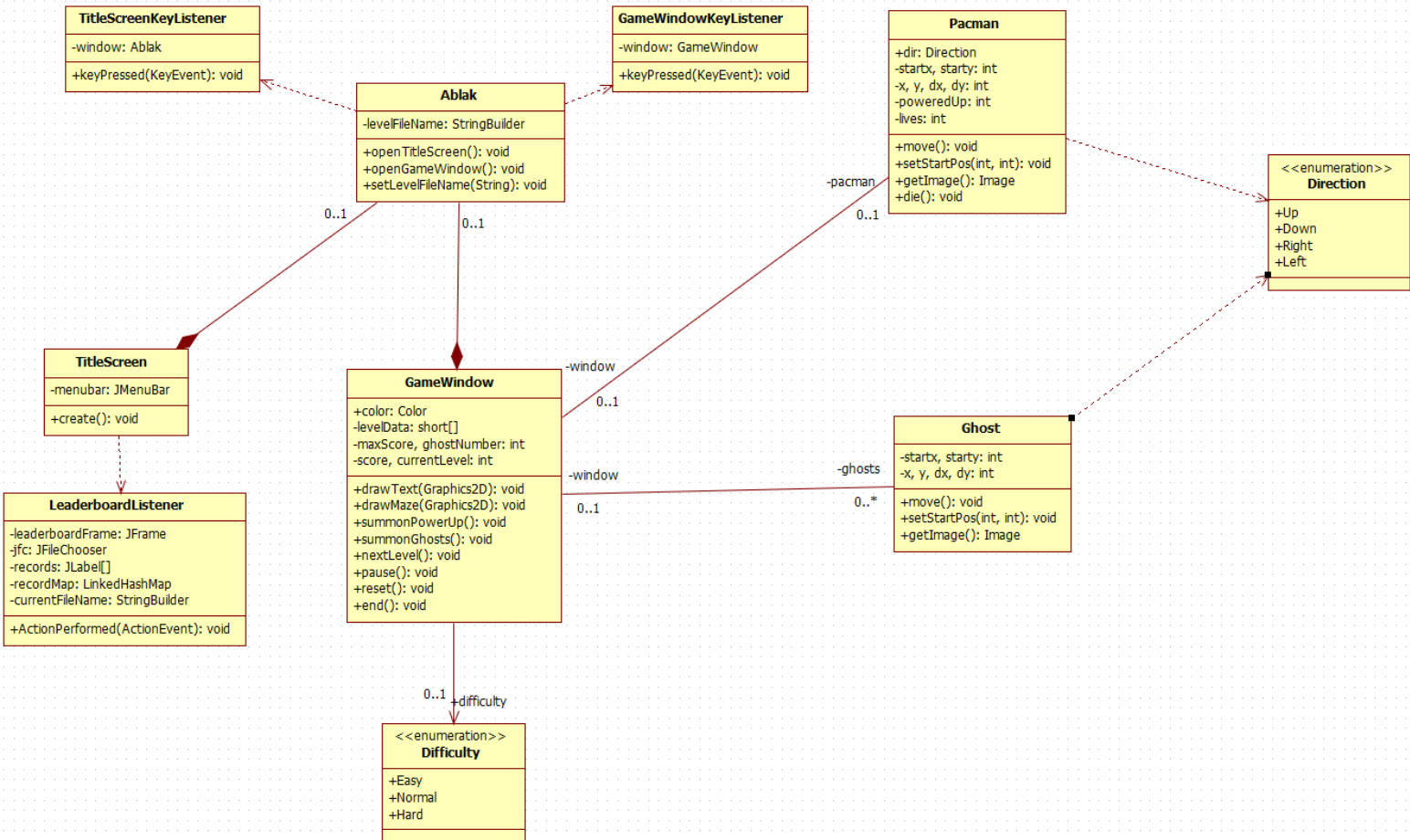
Megoldási ötlet vázlatos ismeretete:

A program a Java Swing GUI funkcióit használja fel. A kezdőképernyőt egy BorderLayout-ban lévő **kép és két szöveges mezőt megvalósító JTextField** készíti el. A fenti menürendszert egy JMenuBar valósítja meg, a bennük lévő JMenu-k és JMenuItem-ek segítségével. Bizonyos elemekhez Listenerek tartoznak, amikkel eseményeket lehet kezelni, illetve meghívják az akkor szükséges metódusokat.

A beállítások közül a ranglista megjelenítését **JLabel-ekkel** készítem el, ranglista betöltését és elmentését a JFileChooser segítségével, illetve standard I/O funkciókkal valósítom meg. A nehézségek változtatják meg a játékban lévő szellemek számát, **illetve a játékos életeinek a számát. A játékot az Enter gomb megnyomásával lehet elkezdeni**. A játék végén ha ranglistás helyezést ért el a játékos, akkor beírhatja a nevét a megszerzett pontszáma mellé, ezt egy JOptionPane-el elkészített párbeszédablak elfogadása után lehet megerősíteni.

A játékban Pac-Mant a billentyűzet nyíl gombjaival lehet irányítani, illetve a 'P' (Pause) gomb megnyomásával lehet ideiglenesen leállítani a játékot, amit majd a gomb ismételt megnyomásával lehet a leállított pillanattól folytatni. **A játékból az Esc megnyomásával hamarabb ki lehet lépni. A kezdőképernyőről pedig az S gomb megnyomásával lehet kilépni.**

### 3. Osztálydiagram



## 4. Metódusok leírása

(A legtöbb get/set függvény nem került leírásra, mivel azok működése triviális.)

### 4.1. Ablak

**4.1.1. OpenTitleScreen(): void** - Létrehozza a kezdőképernyőt, a hozzá tartozó Listener-rel és menüvel együtt.

**4.1.2. OpenGameWindow(): void** - Létrehozza a játékalapot, a hozzá tartozó Listener-rel.

**4.1.3. SetLevelFileName(): void** - Beállítja az adott pálya fájljának a nevét.

### 4.2. TitleScreen

**4.2.1. create(): void** – A menüpontokat és a bennük lévő opciókat készíti el, hozzáadja a szükséges Listener-eket, illetve a képet és a két szövegdobozt rajzolja ki.

### 4.3. TitleScreenKeyListener

**4.3.1. keyPressed(KeyEvent): void** – Az Enter gomb megnyomására a Pacman játék jellegzetes kezdő dallamát indítja el, és meghívja az Ablak OpenGameWindow() metódusát. Az S gomb megnyomására kilép a programból.

### 4.4. GameWindow

- 4.4.1. paintComponent(Graphics): void** – Ez a metódus hívódik meg a Timer minden lépésén, további függvényeket hív meg.
- 4.4.2. draw(Graphics): void** – A Graphics-ként kapott attribútumot Graphics2D-re kasztolja, és azzal hívja meg a többi kirajzoláshoz szükséges függvényeket.
- 4.4.3. movePacman(Graphics2D): void** – Meghívja a pályán lévő Pacman move() függvényét, illetve az irányától függően kirajzolja őt a pályára. Ha golyót vesz fel, növeli a pontot, ha egy színes golyót vesz fel, akkor egy ideig halhatatlanná válik.
- 4.4.4. moveGhosts(Graphics2D): void** – Az összes szellem move() függvényét hívja, illetve kirajzolja a képüket a pályára.
- 4.4.5. drawText(Graphics2D): void** – A jobb alsó sarokban lévő szövegeket írja ki.
- 4.4.6. drawMaze(Graphics2D): void** – A pályát alakító blokkokat és golyókat rajzolja ki a kezdőképernyőben beállított színnel.
- 4.4.7. summonPowerUp(): void** – Random időközönként hívott függvény, ami a pályán lévő golyók közül random helyre egy színes golyót hív le.
- 4.4.8. summonGhosts(int, int): void** – A pályán egy megjelölt helyre, megadott számú szellemet rak le a függvény.
- 4.4.9. getDataAt(int, int): int** – Lekérdezi az adott helyen lévő mező értékét.
- 4.4.10. killGhost(Ghost): void** – Az adott szellemet eltünteti a pályáról.
- 4.4.11. nextLevel(): void** – Ha minden golyó fel lett szedve a pályáról, akkor a Pacman és a szellemek helyét a kezdőpozíciókra teszi, a golyókat újrarajzolja, és indul a következő szint.
- 4.4.12. pause(): void** – A P gomb megnyomására hívódik meg a függvény, mely leállítja a játékmenetet, majd a gomb ismételt megnyomására újra elindítja a játékot ott, ahol az abba lett hagyva.
- 4.4.13. reset(): void** – Ha a Pacman meghal, akkor egy élet levonódik és ő, illetve a szellemek a kezdőpozícióba kerülnek, de az adott szinten lévő golyók megmaradnak.
- 4.4.14. end(): void** – Ha a Pacman meghal, és nincs több élete, akkor a játék véget ér, és a megszerzett pont bekerülhet a ranglistába.

## 4.5. GameWindowKeyListener

4.5.1. **keyPressed(KeyEvent): void** – Az Esc gomb megnyomásával a játékos visszakerül a kezdőképernyőre, a P gomb megnyomása a GameWindow pause() függvényét hívja meg, az iránygombok pedig Pacman mozgásáért felelősek.

## 4.6. Pacman

4.6.1. **getImage(): Image** – Ha Pacman épp egy színes golyó hatása alatt van, akkor egy piros Pacman képet ad vissza, egyébként egy sárga Pacman képet ad vissza.

4.6.2. **setStartPos(int, int): void** – Beállítja a kezdőpontot, ahová a játék során visszakerül halál vagy szint megnyerése után, ezzel együtt a jelenlegi pozícióját is ide állítja be.

4.6.3. **move(): void** – Ha Pacman épp egy mező közepén áll akkor az utoljára lenyomott gomb irányába mozog. Ha épp halhatatlan, akkor annak idejét csökkenti. Ha Pacman útjában egy blokk van, vagy már a pálya szélén van, akkor oda nem fog átlépni.

4.6.4. **die(): void** – Levon egy életet, és ha még marad, akkor a GameWindow reset()-et hívja, ha már nem marad, akkor pedig az end()-et hívja meg.

## 4.7. Ghost

4.7.1. **move(): void** – Egy random irányba mozdul el, ha ott nincs blokk vagy nem a pálya szélén van. Ha egy Pacman-el ütközik, akkor ha az halhatatlan, a szellemet öli meg, ha nem, akkor pedig Pacman-t.

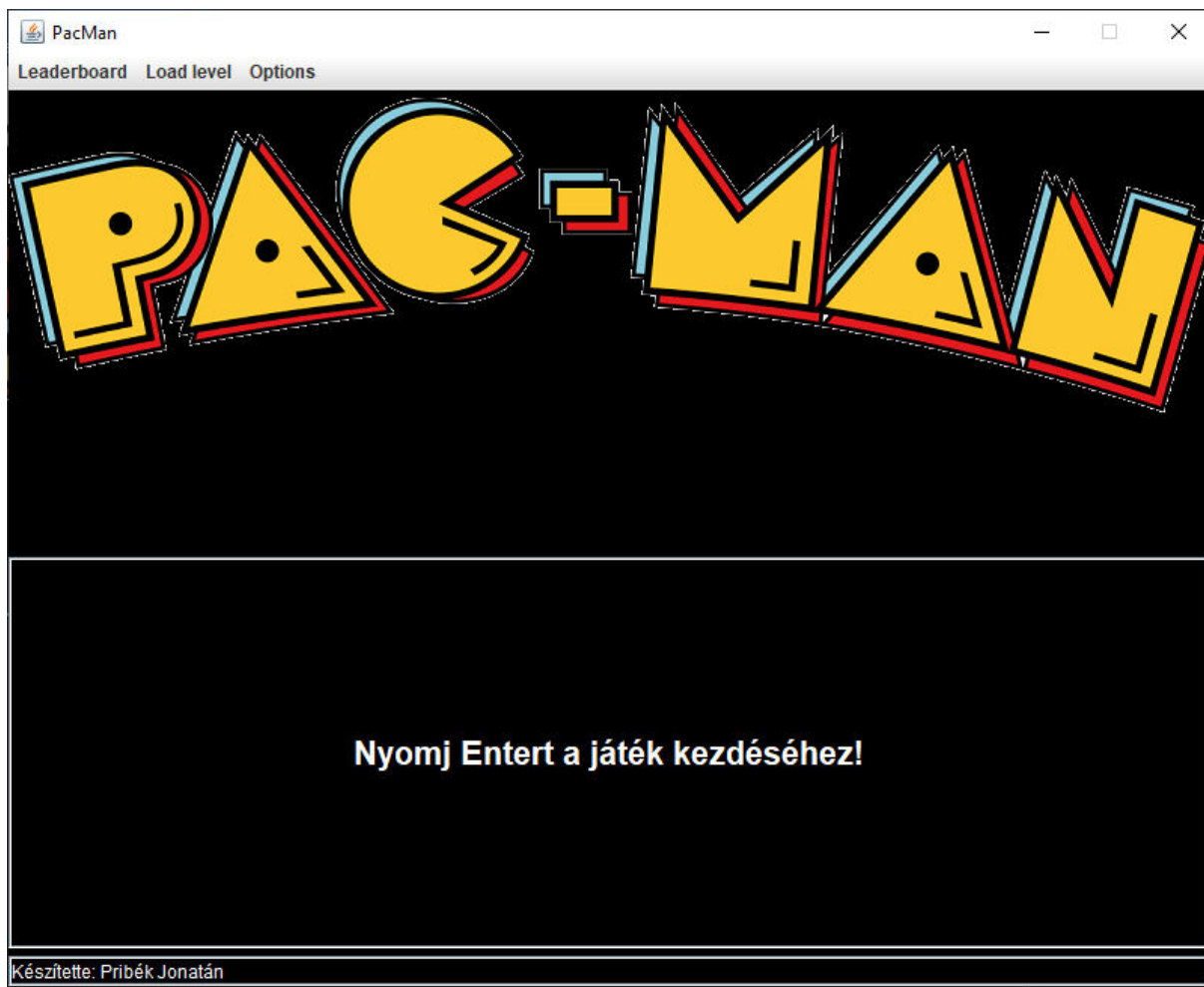
## 4.8. LeaderboardListener

4.8.1. **actionPerformed(ActionEvent): void** – A gomb megnyomásától függően cselekszik. Ha Load, akkor egy fájlkiválasztás után betölti a kiválasztott textfájlból a ranglistát. Ha Save, akkor egy fájlkiválasztás után elmenti egy textfájlba a ranglistát. Ha Open, akkor egy külön ablakban megnyitja az elmentett ranglistát.

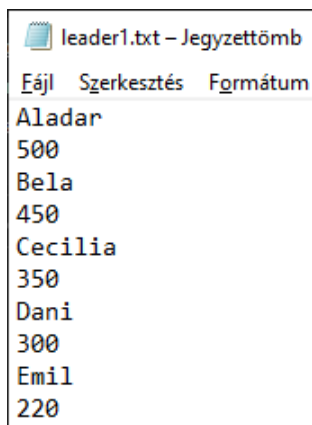
4.9. **Difficulty (Enum)** – Lehet Easy (Könnyű), Normal (Közepes) vagy Hard (Nehéz). Ettől függően változik a szellemek száma, illetve Pacman életeinek a száma.

## 5. Felhasználói kézikönyv

A program megnyitásakor a következő ablak jelenik meg:

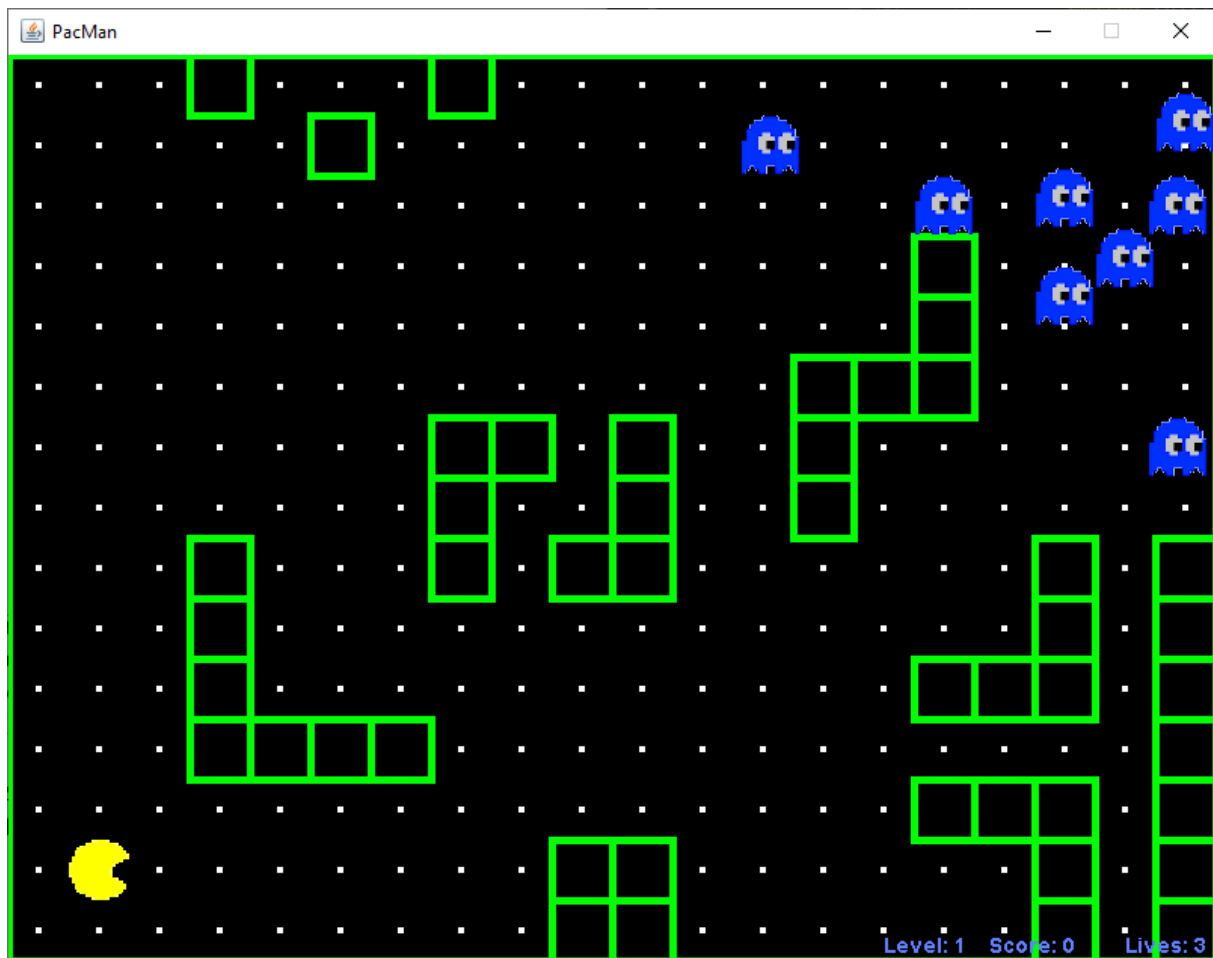


Itt a fenti menüsorban három gomb jelenik meg, ezek a ranglista, a pályabetöltés, illetve a beállítások. A ranglistára való kattintás után három opció jelenik meg: Open (megnyitás), Load (betöltés), Save (elmentés). Megnyitásnál egy új ablakban jelenik meg a program szerint öt legtöbb pontot elérő játékos neve és pontszáma. Új ranglista betöltésénél fontos, hogy a szöveges fájl formátuma a következőképpen nézzen ki: (alapértelmezett: leaderboard.txt)









Itt Pacman a négy iránynyíl segítségével tud mozogni az adott irányokba. A cél, hogy minden fehér golyót összegyűjtsön anélkül, hogy a szellemekkel találkozzon. Ha ezt megteszi, akkor az adott szintet megnyerte és a következő szintre kerül, ahol a golyók újratermelődnek, és egyre több szellem jelenik meg. Időnként egy nagyobb, piros golyó jelenik meg a pályán, ha ezt a játékos felveszi, akkor néhány másodpercig halhatatlan lesz, azaz ha szellemmel találkozik, akkor Pacman helyett a szellem fog meghalni. Egyébként, ha Pacman nem vett fel egy erősítő golyót, akkor találatakor ő hal meg, és egy élet levonódik. Ha minden élet elfogy, akkor a játék véget ér, és a játékos visszakerül a kezdő képernyőre.