

# Dokumentáció

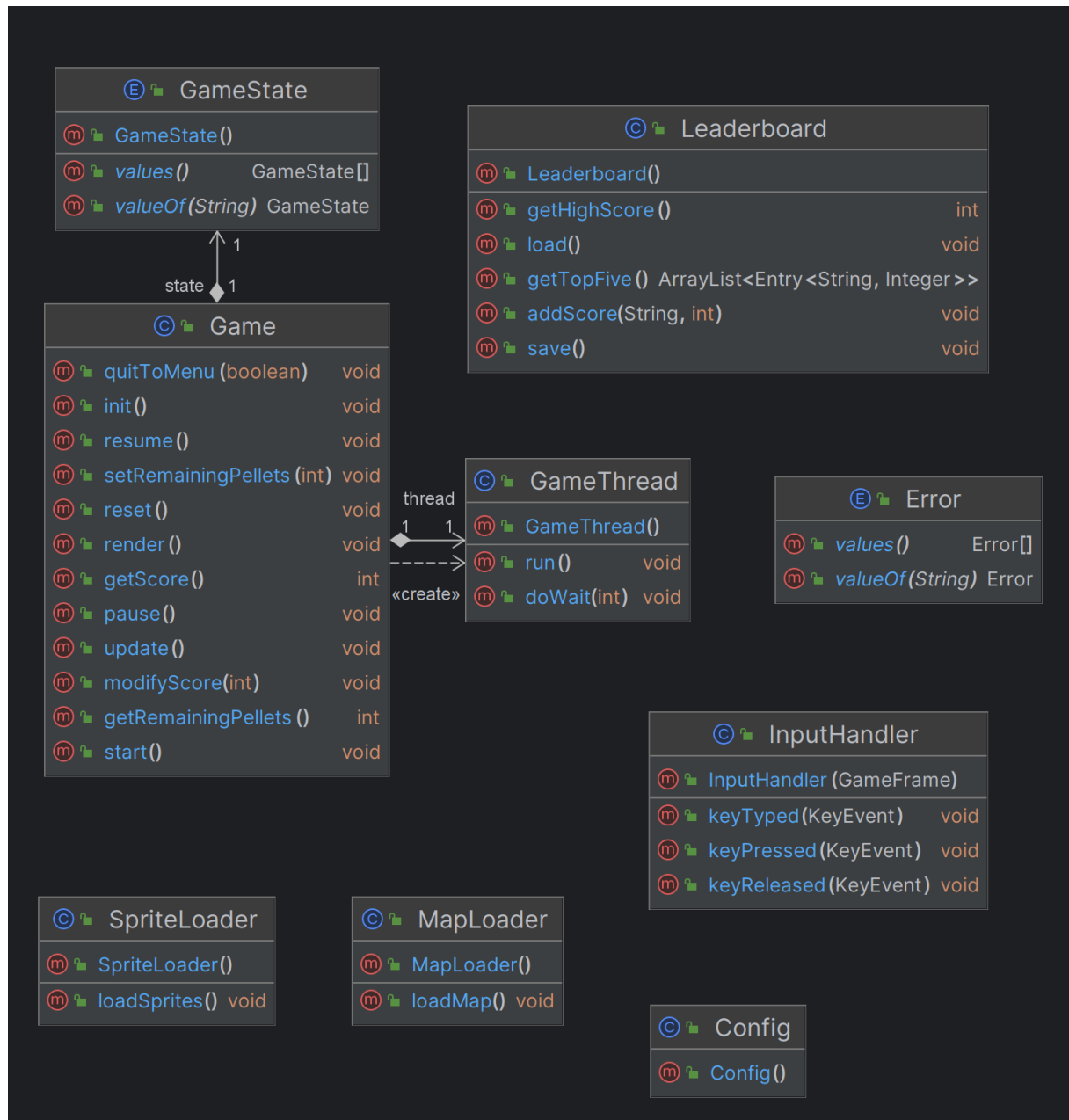
Készítette: Lénárt Levente

Ez a Programozás alapjai 3 tárgyhoz készített házi feladatom dokumentációja.

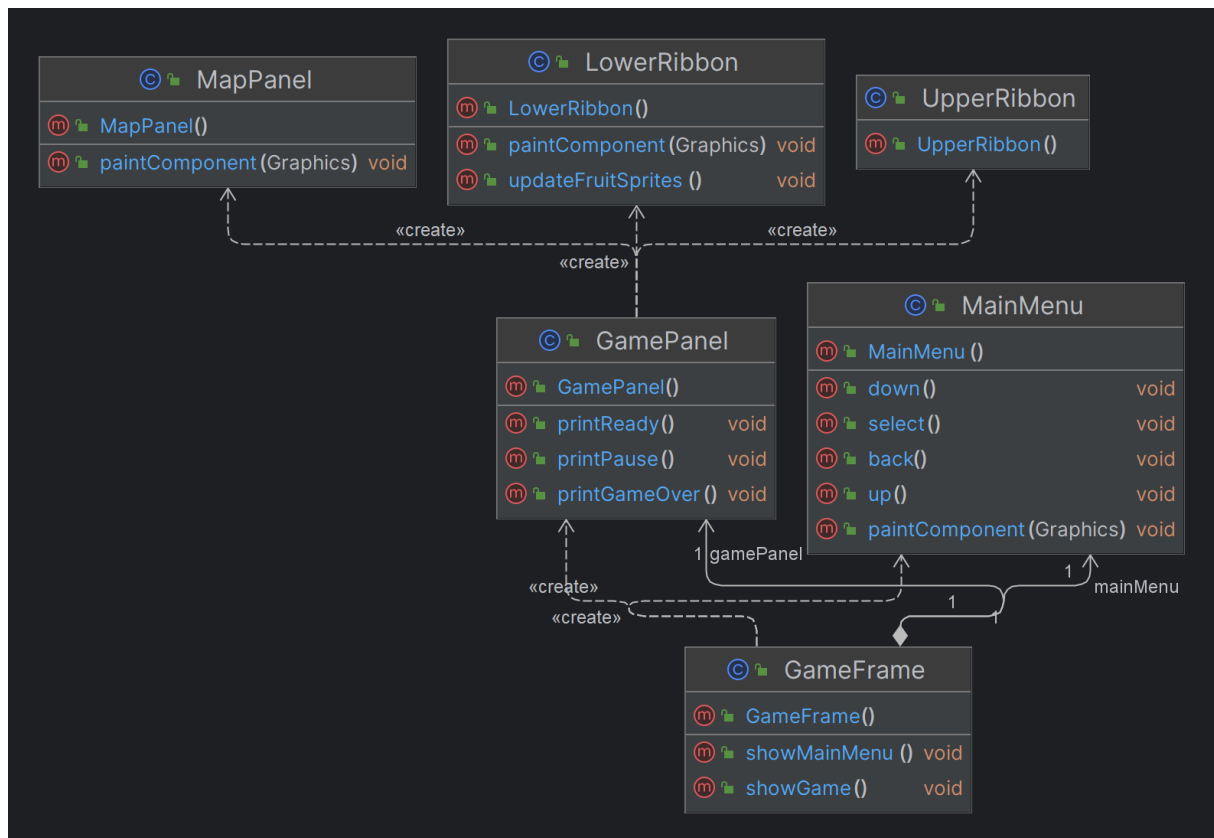
## A program osztályainak UML-diagramja

Ebben a részben csomagonként láthatjuk a program osztályainak diagramját.

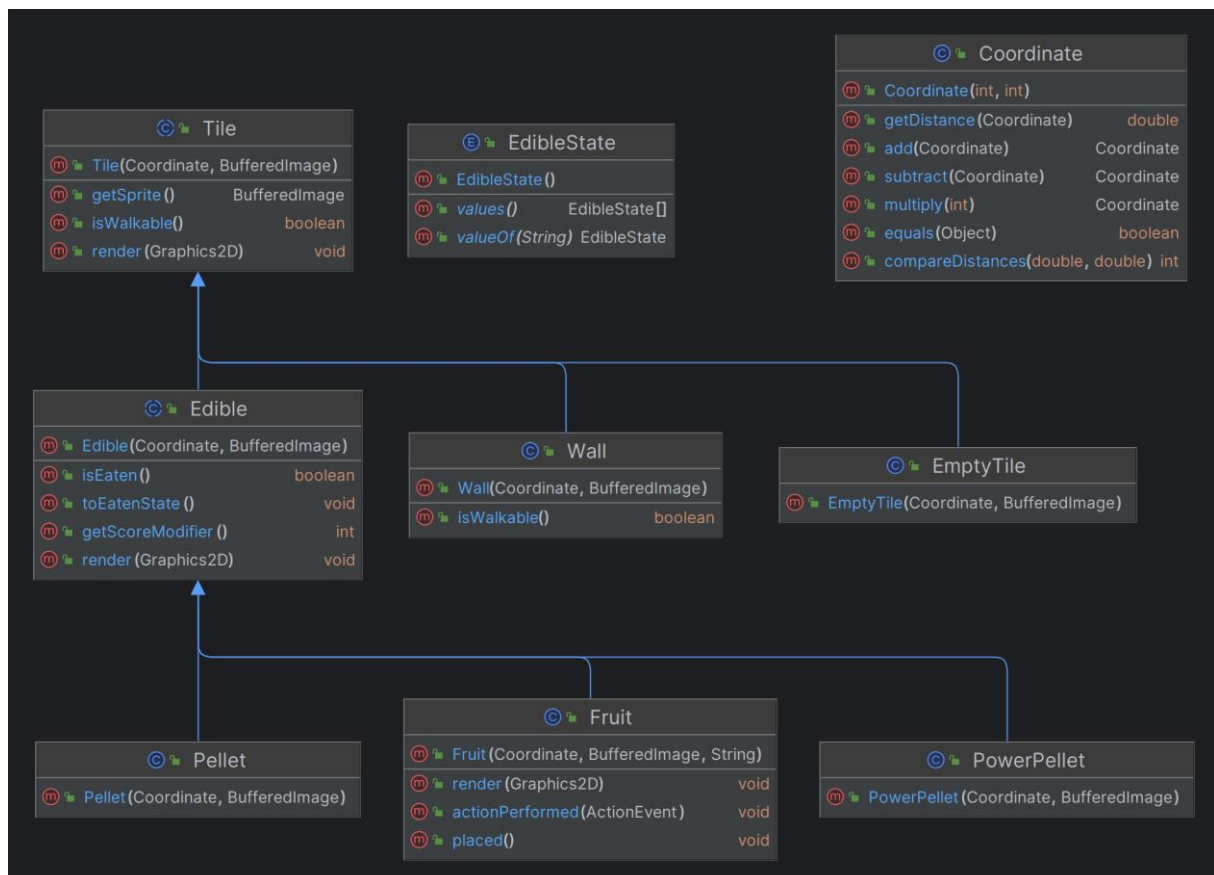
*Game csomag:*



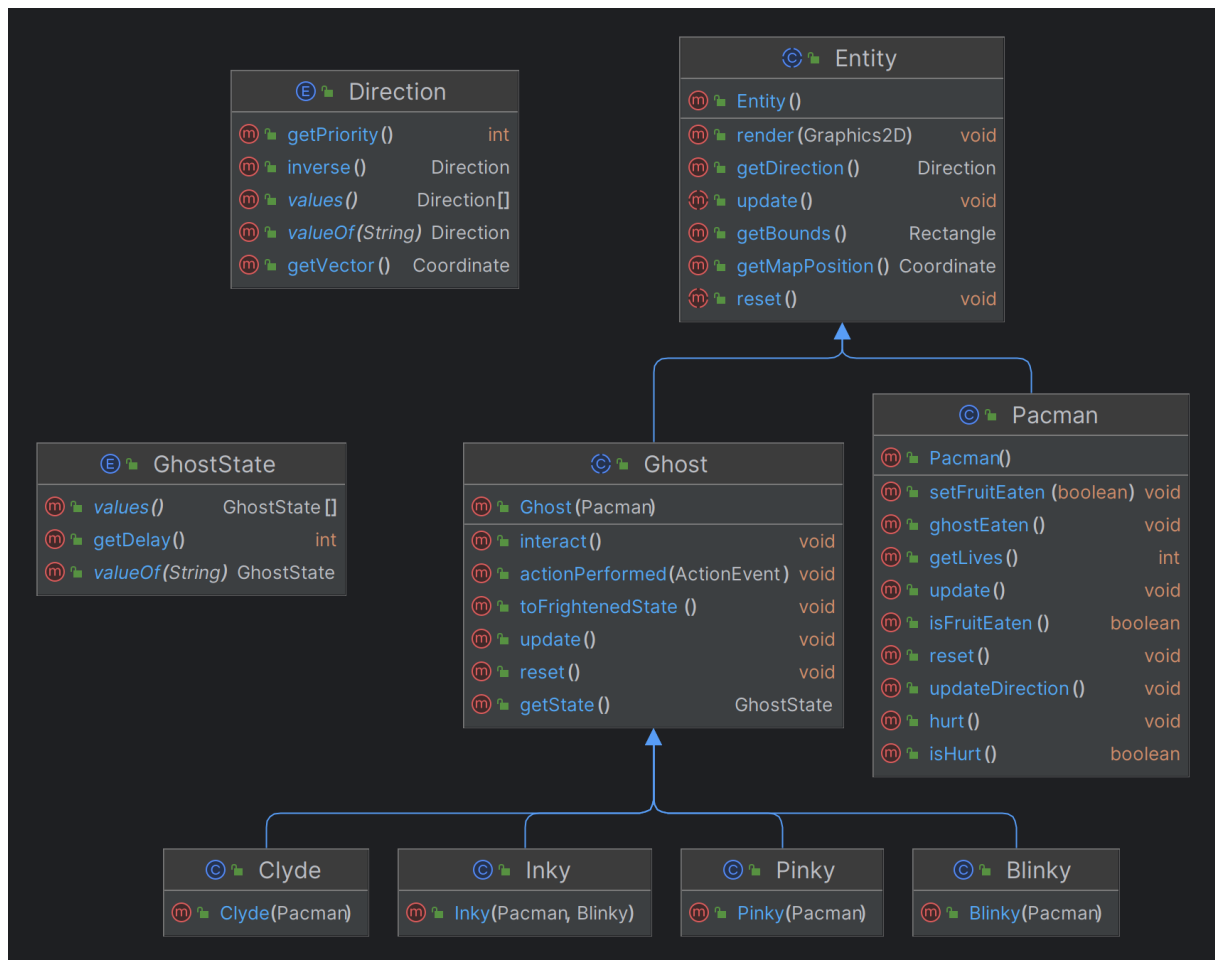
Display csomag:



Tile csomag:



Entity csomag:



## A program teljes dokumentációja

A következőkben a Doxygen által generált dokumentációt láthatjuk.

Pacman

Generated by Doxygen 1.9.8



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ActionListener	
pacman.game.entity.ghost.Ghost	??
pacman.game.entity.ghost.Blinky	??
pacman.game.entity.ghost.Clyde	??
pacman.game.entity.ghost.Inky	??
pacman.game.entity.ghost.Pinky	??
pacman.game.tile.edible.Fruit	??
pacman.game.util.Config	??
pacman.game.tile.Coordinate	??
pacman.game.entity.Direction	??
pacman.game.tile.edible.EdibleState	??
pacman.game.entity.Entity	??
pacman.game.entity.ghost.Ghost	??
pacman.game.entity.pacman.Pacman	??
pacman.game.util.Error	??
pacman.game.Game	??
pacman.game.GameState	??
pacman.game.entity.ghost.GhostState	??
JFrame	
pacman.game.display.GameFrame	??
JPanel	
pacman.game.display.GamePanel	??
pacman.game.display.LowerRibbon	??
pacman.game.display.MainMenu	??
pacman.game.display.MapPanel	??
pacman.game.display.UpperRibbon	??
KeyListener	
pacman.game.input.InputHandler	??
pacman.game.Leaderboard	??
pacman.game.util.MapLoader	??
pacman.game.util.SpriteLoader	??
Thread	
pacman.game.GameThread	??
pacman.game.tile.Tile	??
pacman.game.tile.EmptyTile	??

pacman.game.tile.edible.Edible . . . . .	??
pacman.game.tile.edible.Fruit . . . . .	??
pacman.game.tile.edible.Pellet . . . . .	??
pacman.game.tile.edible.PowerPellet . . . . .	??
pacman.game.tile.wall.Wall . . . . .	??

## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">pacman.game.entity.ghost.Blinky</a>	Blinky (piros) szellemet reprezentáló osztály . . . . .	??
<a href="#">pacman.game.entity.ghost.Clyde</a>	Clyde (narancssárga) szellemet reprezentáló osztály . . . . .	??
<a href="#">pacman.game.util.Config</a>	Konfigurációért felelős osztály . . . . .	??
<a href="#">pacman.game.tile.Coordinate</a>	Koordinátákat és vektorokat kezelő osztály . . . . .	??
<a href="#">pacman.game.entity.Direction</a>	Írányt reprezentáló enum Az irányt a képernyőhöz viszonyítva kell érteni . . . . .	??
<a href="#">pacman.game.tile.edible.Edible</a>	Ehető elemeket reprezentáló osztály . . . . .	??
<a href="#">pacman.game.tile.edible.EdibleState</a>	Edible-k állapotait kezelő enum . . . . .	??
<a href="#">pacman.game.tile.EmptyTile</a>	Üres Tile-t reprezentáló osztály . . . . .	??
<a href="#">pacman.game.entity.Entity</a>	Mozgó entitást reprezentáló osztály . . . . .	??
<a href="#">pacman.game.util.Error</a>	Lehetséges hibajelzéseket, és azok üzeneteit tároló enum . . . . .	??
<a href="#">pacman.game.tile.edible.Fruit</a>	Gyümölcsöt reprezentáló osztály A ggyümölcsök mind ugyan ott jelennek meg a pályán bizonyos időközönként . . . . .	??
<a href="#">pacman.game.Game</a>	A játékot kezelő osztály . . . . .	??
<a href="#">pacman.game.display.GameFrame</a>	Ebben az ablakban jelenik meg a játék Ez az osztály felelős még a hiba dialog box-ok megjelenítéséért is . . . . .	??
<a href="#">pacman.game.display.GamePanel</a>	Ezen a panelen jelenik meg a pálya és az alsó és felső szalagok . . . . .	??
<a href="#">pacman.game.GameState</a>	A játék állapotát reprezentáló enum . . . . .	??
<a href="#">pacman.game.GameThread</a>	A game loop-ért felelős szál . . . . .	??
<a href="#">pacman.game.entity.ghost.Ghost</a>	Szellemet reprezentáló osztály . . . . .	??



<a href="#">pacman.game.entity.ghost.GhostState</a>	
A szellemek állapotait tároló enum . . . . .	??
<a href="#">pacman.game.entity.ghost.Inky</a>	
Inky (kék) szellemet reprezentáló osztály . . . . .	??
<a href="#">pacman.game.input.InputHandler</a>	
Bemenetet kezelő osztály . . . . .	??
<a href="#">pacman.game.Leaderboard</a>	
Pontszámok tárolásáért felelős osztály . . . . .	??
<a href="#">pacman.game.display.LowerRibbon</a>	
A képernyőn megjelenő alsó szalagot kezelő osztály . . . . .	??
<a href="#">pacman.game.display.MainMenu</a>	
A főmenüt megjelenítő osztály . . . . .	??
<a href="#">pacman.game.util.MapLoader</a>	
A pálya betöltéséért felelős osztály . . . . .	??
<a href="#">pacman.game.display.MapPanel</a>	
Ezen a panelen razolódik ki a pálya . . . . .	??
<a href="#">pacman.game.entity.pacman.Pacman</a>	
Pacmant reprezentáló osztály . . . . .	??
<a href="#">pacman.game.tile.edible.Pellet</a>	
Ehető pontot reprezentáló osztály . . . . .	??
<a href="#">pacman.game.entity.ghost.Pinky</a>	
Pinky (rózsaszín) szellemet reprezentáló osztály . . . . .	??
<a href="#">pacman.game.tile.edible.PowerPellet</a>	
Ehető pontot reprezentáló osztály Ezt elfogyasztva a szellemek Frightened állapotba kerülnek .	??
<a href="#">pacman.game.util.SpriteLoader</a>	
Grafikák betöltéséért felelős osztály . . . . .	??
<a href="#">pacman.game.tile.Tile</a>	
Egy, a pályán lévő cellát reprezentáló osztály . . . . .	??
<a href="#">pacman.game.display.UpperRibbon</a>	
A felső szalagot megjelenítő osztály . . . . .	??
<a href="#">pacman.game.tile.wall.Wall</a>	
Falat reprezentáló osztály . . . . .	??

## Chapter 3

# Class Documentation

### 3.1 pacman.game.entity.ghost.Blinky Class Reference

Blinky (piros) szellemet reprezentáló osztály.

#### Public Member Functions

- [Blinky](#) (Pacman [pacman](#))  
*Blinky default konstruktor.*

#### Public Member Functions inherited from [pacman.game.entity.ghost.Ghost](#)

- [Ghost](#) (Pacman [pacman](#))  
*Ghost default konstruktor.*
- void [reset](#) ()  
*Kezdő állapotba helyezi a szellemeket.*
- void [update](#) ()  
*Minden belső logikát kezelő függvény.*
- void **actionPerformed** (ActionEvent e)  
*Frissíti a szellem állapotát A játék elején a szellemek a CHASE és SCATTER állapotok között váltanak, majd CHASE állapotban maradnak.*
- void [toFrightenedState](#) ()  
*FRIGHTENED állapotba váltja a szellemet.*
- void **interact** ()  
*Interakció Pacman és a szellem között Pacman hívja meg, amikor egy Tile-re kerülnek a szellemmel.*
- [GhostState](#) [getState](#) ()  
*State getter.*

## Public Member Functions inherited from [pacman.game.entity.Entity](#)

- **Entity ()**  
*Entity default konstruktor.*
- **Direction getDirection ()**  
*Direction getter.*
- void **render** (Graphics2D graphics)  
*Az Entity megjelenítése a képernyőn.*
- Coordinate **getMapPosition ()**  
*Átkonvertálja a képernyő koordinátákat pálya koordinátákra.*
- Rectangle **getBounds ()**  
*Collision detection-höz használt függvény.*

## Protected Member Functions

- void **toStartingPos ()**  
*Kezdőállapotba állítja a szellemet.*
- void **initSprites ()**  
*Sprite-okat inicializáló metódus.*
- Coordinate **getChaseTarget ()**
- Coordinate **getScatterTarget ()**

## Protected Member Functions inherited from [pacman.game.entity.ghost.Ghost](#)

- void **updateSprite ()**  
*Frissíti a szellem aktuális sprite-ját Több féle sprite-ja van, más logika szerint működik, mint az Entity metódusa.*
- void **init ()**  
*Inicializáló metódus.*
- void **updateDirection ()**  
*Frissíti a szellem irányát.*
- void **updateTargetTile ()**  
*Frissíti a szellem célját.*
- boolean **isInsideHouse ()**
- boolean **isInFrontOfHouse ()**

## Protected Member Functions inherited from [pacman.game.entity.Entity](#)

- Coordinate **getDrawPosition ()**  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell A Tile és az Entity sprite-ok méreteinek különbsége miatt külön ki kell számolni, hova kell rajzolni az Entity-ket.*
- void **checkWallCollisions ()**  
*Fallal való ütközést vizsgálja Ha ütközést talál, visszaállítja az Entityt az előző helyére.*
- void **checkOutOfFrame ()**  
*Azt vizsgálja, mikor megy ki az Entity a képernyőről Ha kimegy, a pálya másik oldalára rakja azt.*

## Additional Inherited Members

### Protected Attributes inherited from [pacman.game.entity.ghost.Ghost](#)

- Pacman **pacman**  
*Pacman referenciája.*
- List< BufferedImage > **frightenedSprites**  
*A FRIGHTENED állapotú szellemek sprite-jai.*
- Map< [Direction](#), BufferedImage > **eatenSprites**  
*Az EATEN állapotú szellemek sprite-jai.*

### Protected Attributes inherited from [pacman.game.entity.Entity](#)

- Coordinate **position**  
*Az Entity-t reprezentáló sprite középpontjának helye a képernyőn Valójában nem teljesen a középpontját tároljuk, a tárolt koordinátát úgy kapjuk meg, hogy a sprite bal felső sarkának koordinátáiból kivonunk a (ENTITY\_SIZE - TILE\_SIZE) \* SCALE egész számot.*
- [Direction](#) **direction**  
*Az Entity aktuális haladási iránya.*
- int **speed**  
*Az entitás aktuális sebessége (pixel / frame)*
- Map< [Direction](#), ArrayList< BufferedImage > > **defaultSprites**  
*Az Entity sprite-jai irányonként rendezve A megfelelő irányt megadva a map-ből megkapjuk azon sprite-ok sorozatát, melyeken index szerint végighaladva egy animációt kapunk.*
- ArrayList< BufferedImage > **spriteList**  
*A jelenlegi iránynak megfelelő sprite-ok listája Ezen spite-ok egymásutánja adja ki a karakter animációját.*
- int **spriteIndex** = 0  
*A jelenleg használt sprite indexe az iránynak megfelelő sprite-ok listájában.*
- BufferedImage **sprite**  
*Az aktuális sprite.*
- int **animationDrawCounter** = 0  
*A legutóbbi sprite váltás óta eltelt framek száma.*

## 3.1.1 Detailed Description

Blinky (piros) szellemet reprezentáló osztály.

## 3.1.2 Constructor & Destructor Documentation

### 3.1.2.1 Blinky()

```
pacman.game.entity.ghost.Blinky.Blinky (
    Pacman pacman )
```

Blinky default konstruktor.

## Parameters

<i>pacman</i>	Pacman referenciája
---------------	---------------------

### 3.1.3 Member Function Documentation

#### 3.1.3.1 getChaseTarget()

`Coordinate pacman.game.entity.ghost.Blinky.getChaseTarget ( ) [protected]`

## Returns

A szellem célja CHASE állapotban

Reimplemented from [pacman.game.entity.ghost.Ghost](#).

#### 3.1.3.2 getScatterTarget()

`Coordinate pacman.game.entity.ghost.Blinky.getScatterTarget ( ) [protected]`

## Returns

A szellem célja SCATTER állapotban

Reimplemented from [pacman.game.entity.ghost.Ghost](#).

#### 3.1.3.3 initSprites()

`void pacman.game.entity.ghost.Blinky.initSprites ( ) [protected]`

Sprite-okat inicializáló metódus.

Reimplemented from [pacman.game.entity.Entity](#).

#### 3.1.3.4 toStartingPos()

`void pacman.game.entity.ghost.Blinky.toStartingPos ( ) [protected]`

Kezdőállapotba állítja a szellemet.

Reimplemented from [pacman.game.entity.Entity](#).

## 3.2 pacman.game.entity.ghost.Clyde Class Reference

Clyde (narancssárga) szellemet reprezentáló osztály.

### Public Member Functions

- [Clyde](#) (Pacman [pacman](#))  
*Clyde default konstruktor.*

### Public Member Functions inherited from [pacman.game.entity.ghost.Ghost](#)

- [Ghost](#) (Pacman [pacman](#))  
*Ghost default konstruktor.*
- void [reset](#) ()  
*Kezdő állapotba helyezi a szellemeket.*
- void [update](#) ()  
*Minden belső logikát kezelő függvény.*
- void [actionPerformed](#) (ActionEvent e)  
*Frissíti a szellem állapotát A játék elején a szellemek a CHASE és SCATTER állapotok között váltanak, majd CHASE állapotban maradnak.*
- void [toFrightenedState](#) ()  
*FRIGHTENED állapotba váltja a szellemet.*
- void [interact](#) ()  
*Interakció Pacman és a szellem között Pacman hívja meg, amikor egy Tile-re kerülnek a szellemmel.*
- [GhostState](#) [getState](#) ()  
*State getter.*

### Public Member Functions inherited from [pacman.game.entity.Entity](#)

- [Entity](#) ()  
*Entity default konstruktor.*
- [Direction](#) [getDirection](#) ()  
*Direction getter.*
- void [render](#) (Graphics2D graphics)  
*Az Entity megjelenítése a képernyőn.*
- Coordinate [getMapPosition](#) ()  
*Átkonvertálja a képernyő koordinátákat pálya koordinátákra.*
- Rectangle [getBounds](#) ()  
*Collision detection-höz használt függvény.*

### Protected Member Functions

- void [toStartingPos](#) ()  
*Kezdőállapotba állítja a szellemet.*
- void [initSprites](#) ()  
*Sprite-okat inicializáló metódus.*
- Coordinate [getChaseTarget](#) ()
- Coordinate [getScatterTarget](#) ()

## Protected Member Functions inherited from [pacman.game.entity.ghost.Ghost](#)

- void [updateSprite](#) ()  
*Frissíti a szellem aktuális sprite-ját Több féle sprite-ja van, más logika szerint működik, mint az Entity metódusa.*
- void [init](#) ()  
*Inicializáló metódus.*
- void [updateDirection](#) ()  
*Frissíti a szellem irányát.*
- void [updateTargetTile](#) ()  
*Frissíti a szellem célját.*
- boolean [isInsideHouse](#) ()
- boolean [isInFrontOfHouse](#) ()

## Protected Member Functions inherited from [pacman.game.entity.Entity](#)

- Coordinate [getDrawPosition](#) ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell A Tile és az Entity sprite-ok méreteinek különbsége miatt külön ki kell számolni, hova kell rajzolni az Entity-ket.*
- void [checkWallCollisions](#) ()  
*Fallal való ütközést vizsgálja Ha ütközést talál, visszaállítja az Entityt az előző helyére.*
- void [checkOutOfFrame](#) ()  
*Azt vizsgálja, mikor megy ki az Entity a képernyőről Ha kimegy, a pálya másik oldalára rakja azt.*

## Additional Inherited Members

## Protected Attributes inherited from [pacman.game.entity.ghost.Ghost](#)

- Pacman [pacman](#)  
*Pacman referenciája.*
- List< BufferedImage > [frightenedSprites](#)  
*A FRIGHTENED állapotú szellemek sprite-jai.*
- Map< [Direction](#), BufferedImage > [eatenSprites](#)  
*Az EATEN állapotú szellemek sprite-jai.*

## Protected Attributes inherited from [pacman.game.entity.Entity](#)

- Coordinate [position](#)  
*Az Entity-t reprezentáló sprite középpontjának helye a képernyőn Valójában nem teljesen a középpontját tároljuk, a tárolt koordinátát úgy kapjuk meg, hogy a sprite bal felső sarkának koordinátaiból kivonunk a (ENTITY\_SIZE - TILE\_SIZE) \* SCALE egész számot.*
- [Direction](#) [direction](#)  
*Az Entity aktuális haladási iránya.*
- int [speed](#)  
*Az entitás aktuális sebessége (pixel / frame)*
- Map< [Direction](#), ArrayList< BufferedImage > > [defaultSprites](#)  
*Az Entity sprite-jai irányonként rendezve A megfelelő irányt megadva a map-ből megkapjuk azon sprite-ok sorozatát, melyeken index szerint végighaladva egy animációt kapunk.*
- ArrayList< BufferedImage > [spriteList](#)  
*A jelenlegi iránynak megfelelő sprite-ok listája Ezen sprite-ok egymásutánja adja ki a karakter animációját.*
- int [spriteIndex](#) = 0  
*A jelenleg használt sprite indexe az iránynak megfelelő sprite-ok listájában.*
- BufferedImage [sprite](#)  
*Az aktuális sprite.*
- int [animationDrawCounter](#) = 0  
*A legutóbbi sprite váltás óta eltelt framek száma.*

### 3.2.1 Detailed Description

Clyde (narancssárga) szellemet reprezentáló osztály.

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 Clyde()

```
pacman.game.entity.ghost.Clyde.Clyde (
    Pacman pacman )
```

Clyde default konstruktor.

##### Parameters

<i>pacman</i>	Pacman referenciája
---------------	---------------------

### 3.2.3 Member Function Documentation

#### 3.2.3.1 getChaseTarget()

```
Coordinate pacman.game.entity.ghost.Clyde.getChaseTarget ( ) [protected]
```

##### Returns

A szellem célja CHASE állapotban

Reimplemented from [pacman.game.entity.ghost.Ghost](#).

#### 3.2.3.2 getScatterTarget()

```
Coordinate pacman.game.entity.ghost.Clyde.getScatterTarget ( ) [protected]
```

##### Returns

A szellem célja SCATTER állapotban

Reimplemented from [pacman.game.entity.ghost.Ghost](#).

#### 3.2.3.3 initSprites()

```
void pacman.game.entity.ghost.Clyde.initSprites ( ) [protected]
```

Sprite-okat inicializáló metódus.

Reimplemented from [pacman.game.entity.Entity](#).



### 3.2.3.4 toStartingPos()

```
void pacman.game.entity.ghost.Clyde.toStartingPos ( ) [protected]
```

Kezdőállapotba állítja a szellemet.

Reimplemented from [pacman.game.entity.Entity](#).

## 3.3 pacman.game.util.Config Class Reference

Konfigurációért felelős osztály.

### Static Public Attributes

- static final int **SCREEN\_COLUMNS** = 28
- static final int **SCREEN\_ROWS** = 36
- static final int **MAP\_ROWS** = 31
- static final int **UPPER\_RIBBON\_ROWS** = 3
- static final int **LOWER\_RIBBON\_ROWS** = 2
- static final int **SCALE** = 3
- static final int **TILE\_SPRITE\_SIZE** = 8
- static final int **ON\_SCREEN\_TILE\_SIZE** = TILE\_SPRITE\_SIZE \* SCALE
- static final int **ENTITY\_SPRITE\_SIZE** = 16
- static final int **ON\_SCREEN\_ENTITY\_SIZE** = ENTITY\_SPRITE\_SIZE \* SCALE
- static final int **SCREEN\_WIDTH** = SCREEN\_COLUMNS \* ON\_SCREEN\_TILE\_SIZE
- static final int **SCREEN\_HEIGHT** = SCREEN\_ROWS \* ON\_SCREEN\_TILE\_SIZE
- static final int **MAP\_HEIGHT** = MAP\_ROWS \* ON\_SCREEN\_TILE\_SIZE
- static final int **UPPER\_RIBBON\_HEIGHT** = UPPER\_RIBBON\_ROWS \* ON\_SCREEN\_TILE\_SIZE
- static final int **LOWER\_RIBBON\_HEIGHT** = LOWER\_RIBBON\_ROWS \* ON\_SCREEN\_TILE\_SIZE
- static final int **DISPLAY\_TARGET\_FPS** = 45
- static final int **ENTITY\_ANIMATION\_FPS** = 5
- static final int **PACMAN\_LIVES** = 3
- static final Coordinate **PACMAN\_STARTING\_POS**
- static final int **PACMAN\_SPEED** = SCALE
- static final int **GHOST\_DEFAULT\_SPEED** = PACMAN\_SPEED
- static final int **GHOST\_FRIGHTENED\_SPEED** = GHOST\_DEFAULT\_SPEED / 2
- static final int **GHOST\_EATEN\_SPEED** = GHOST\_DEFAULT\_SPEED \* 2
- static final Coordinate **IN\_FRONT\_OF\_GHOST\_HOUSE** = new Coordinate(14, 11)
- static final Coordinate **INSIDE\_GHOST\_HOUSE** = new Coordinate(14, 14)
- static final Coordinate **BLIKNY\_SCATTER\_TARGET** = new Coordinate(25, -3)
- static final Coordinate **PINKY\_SCATTER\_TARGET** = new Coordinate(2, -3)
- static final Coordinate **INKY\_SCATTER\_TARGET** = new Coordinate(27, 33)
- static final Coordinate **CLYDE\_SCATTER\_TARGET** = new Coordinate(0, 33)
- static final Coordinate **BLINKY\_STARTING\_POS**
- static final Coordinate **PINKY\_STARTING\_POS**
- static final Coordinate **INKY\_STARTING\_POS**
- static final Coordinate **CLYDE\_STARTING\_POS**
- static final Direction **BLINKY\_STARTING\_DIR** = Direction.LEFT
- static final Direction **PINKY\_STARTING\_DIR** = Direction.LEFT
- static final Direction **INKY\_STARTING\_DIR** = Direction.RIGHT
- static final Direction **CLYDE\_STARTING\_DIR** = Direction.RIGHT

- static final int **GHOST\_CHASE\_STATE\_DELAY** = 20 \* 1000
- static final int **GHOST\_SCATTER\_STATE\_DELAY** = 7 \* 1000
- static final int **GHOST\_FRIGHTENED\_STATE\_DELAY** = 8 \* 1000
- static final int **GHOST\_STATE\_INFINITE\_DELAY** = -1
- static final int **PELLET\_SCORE\_MODIFIER** = 10
- static final int **POWER\_PELLET\_SCORE\_MODIFIER** = 50
- static final int **CHERRY\_SCORE\_MODIFIER** = 100
- static final int **STRAWBERRY\_SCORE\_MODIFIER** = 300
- static final int **ORANGE\_SCORE\_MODIFIER** = 500
- static final int **APPLE\_SCORE\_MODIFIER** = 700
- static final ArrayList< String > **FRUIT\_TYPES** = new ArrayList<>()

### Static Package Functions

- [static initializer]

## 3.3.1 Detailed Description

Konfigurációért felelős osztály.

## 3.3.2 Member Data Documentation

### 3.3.2.1 BLINKY\_STARTING\_POS

```
final Coordinate pacman.game.util.Config.BLINKY_STARTING_POS [static]
```

#### Initial value:

```
= new Coordinate(  
    (13 * TILE_SPRITE_SIZE + 3 + 4) * SCALE,  
    (11 * TILE_SPRITE_SIZE + 3) * SCALE)
```

### 3.3.2.2 CLYDE\_STARTING\_POS

```
final Coordinate pacman.game.util.Config.CLYDE_STARTING_POS [static]
```

#### Initial value:

```
= new Coordinate(  
    (15 * TILE_SPRITE_SIZE + 3 + 4) * SCALE,  
    (14 * TILE_SPRITE_SIZE + 3) * SCALE)
```

### 3.3.2.3 INKY\_STARTING\_POS

```
final Coordinate pacman.game.util.Config.INKY_STARTING_POS [static]
```

#### Initial value:

```
= new Coordinate(  
    (11 * TILE_SPRITE_SIZE + 3 + 4) * SCALE,  
    (14 * TILE_SPRITE_SIZE + 3) * SCALE)
```

### 3.3.2.4 PACMAN\_STARTING\_POS

```
final Coordinate pacman.game.util.Config.PACMAN_STARTING_POS [static]
```

#### Initial value:

```
= new Coordinate(  
    (13 * Config.TILE_SPRITE_SIZE + 3 + 4) * Config.SCALE,  
    (23 * Config.TILE_SPRITE_SIZE + 3) * Config.SCALE  
)
```

### 3.3.2.5 PINKY\_STARTING\_POS

```
final Coordinate pacman.game.util.Config.PINKY_STARTING_POS [static]
```

#### Initial value:

```
= new Coordinate(  
    (13 * TILE_SPRITE_SIZE + 3 + 4) * SCALE,  
    (14 * TILE_SPRITE_SIZE + 3) * SCALE)
```

## 3.4 pacman.game.tile.Coordinate Class Reference

Koordinátákat és vektorokat kezelő osztály.

### Public Member Functions

- [Coordinate](#) (int x, int y)  
*Coordinate default konstruktor.*
- [Coordinate add](#) ([Coordinate](#) other)  
*Összead két koordinátát A két koordinátát vektorként kezelhetjük, ez alapján adjuk össze.*
- [Coordinate subtract](#) ([Coordinate](#) other)  
*Kivon két koordinátát egymásból.*
- [Coordinate multiply](#) (int n)  
*Megszoroz egy koordinátát egy egész számmal.*
- boolean **equals** (Object obj)  
*equals metódus override*
- double **getDistance** ([Coordinate](#) other)  
*Visszaadja két koordináta távolságát.*

### Static Public Member Functions

- static int **compareDistances** (double dist1, double dist2)  
*Két lebegőpontos távolság összehasonlítása.*

### Public Attributes

- int **x**
- int **y**

## Static Public Attributes

- static `Coordinate NULLVECTOR` = new `Coordinate(0, 0)`  
*Nullvektor.*

### 3.4.1 Detailed Description

Koordinátákat és vektorokat kezelő osztály.

A koordinátákat a bal felső sarokból kezdve számoljuk, jobbra nő az x, lefelé nő az y.

### 3.4.2 Constructor & Destructor Documentation

#### 3.4.2.1 Coordinate()

```
pacman.game.tile.Coordinate.Coordinate (
    int x,
    int y )
```

Coordinate default konstruktor.

#### Parameters

<i>x</i>	koordináta az x tengelyen
<i>y</i>	koordináta az y tengelyen

### 3.4.3 Member Function Documentation

#### 3.4.3.1 add()

```
Coordinate pacman.game.tile.Coordinate.add (
    Coordinate other )
```

Összead két koordinátát A két koordinátát vektorként kezelhetjük, ez alapján adjuk össze.

#### Parameters

<i>other</i>	Az a koordináta, amit ehhez adunk hozzá
--------------	---

#### Returns

Összeg koordináta

#### 3.4.3.2 compareDistances()

```
static int pacman.game.tile.Coordinate.compareDistances (
    double dist1,
    double dist2 ) [static]
```

Két lebegőpontos távolság összehasonlítása.

#### Returns

0, ha egyenlőek, 1, ha dist1 nagyobb, -1, ha dist1 kisebb

### 3.4.3.3 getDistance()

```
double pacman.game.tile.Coordinate.getDistance (
    Coordinate other )
```

Visszaadja két koordináta távolságát.

#### Parameters

<i>other</i>	másik koordináta
--------------	------------------

#### Returns

a két koordináta távolsága

## 3.5 pacman.game.entity.Direction Enum Reference

Írányt reprezentáló enum Az irányt a képernyőhöz viszonyítva kell érteni.

#### Public Member Functions

- [Direction](#) (int [xOffset](#), int [yOffset](#), int [priority](#))  
*Direction konstruktor.*
- Coordinate [getVector](#) ()  
*Adott irányba való elmozdulás egységvektorát adja vissza.*
- int [getPriority](#) ()
- [Direction inverse](#) ()

#### Public Attributes

- **UP** =(0, -1, 1)
- **LEFT** =(-1, 0, 2)
- **DOWN** =(0, 1, 3)
- **RIGHT** =(1, 0, 4)
- **NONE** =(0, 0, -1)

#### Private Attributes

- final int **xOffset**  
*x tengely irányába való egységnyi eltérés*
- final int **yOffset**  
*y tengely irányába való egységnyi eltérés*
- final int **priority**  
*választási sorrendhez használt prioritás*

### 3.5.1 Detailed Description

Írányt reprezentáló enum Az irányt a képernyőhöz viszónítva kell érteni.

### 3.5.2 Constructor & Destructor Documentation

#### 3.5.2.1 Direction()

```
pacman.game.entity.Direction.Direction (
    int xOffset,
    int yOffset,
    int priority )
```

Direction konstruktor.

##### Parameters

<i>xOffset</i>	x tengely irányába való egységnyi eltérés
<i>yOffset</i>	y tengely irányába való egységnyi eltérés
<i>priority</i>	választási sorrendhez használt prioritás

### 3.5.3 Member Function Documentation

#### 3.5.3.1 getPriority()

```
int pacman.game.entity.Direction.getPriority ( )
```

##### Returns

irány prioritása

#### 3.5.3.2 getVector()

```
Coordinate pacman.game.entity.Direction.getVector ( )
```

Adott irányba való elmozdulás egységvektorát adja vissza.

##### Returns

Elmozdulás egységvektora

#### 3.5.3.3 inverse()

```
Direction pacman.game.entity.Direction.inverse ( )
```

##### Returns

Az irány inverze

## 3.6 pacman.game.tile.edible.Edible Class Reference

Ehető elemeket reprezentáló osztály.

### Public Member Functions

- [Edible](#) ([Coordinate](#) `mapPosition`, [BufferedImage](#) `sprite`)  
*Edible default konstruktor.*
- `int` **getScoreModifier** ()  
*Visszaadja azt, hogy az edible elfogyasztása hány pontot ér.*
- `boolean` **isEaten** ()
- `void` **toEatenState** ()  
*EATEN állapotba állítja az Edible-t.*
- `void` **render** ([Graphics2D](#) `graphics`)  
*A képernyőre rajzolja az Edible-t.*

### Public Member Functions inherited from [pacman.game.tile.Tile](#)

- [Tile](#) ([Coordinate](#) `mapPosition`, [BufferedImage](#) `sprite`)  
*Tile default konstruktor.*
- `boolean` **isWalkable** ()  
*Igaz, ha entitások átsétálhatnak ezen a tile-n.*
- [BufferedImage](#) **getSprite** ()  
*Visszaadja a Tile sprite-ját.*

### Protected Attributes

- [EdibleState](#) **state**  
*Az Edible aktuális állapota.*
- `int` **scoreModifier**  
*Az Edible elfogyasztása ennyivel módosítja a pontszámot.*

### Protected Attributes inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) **mapPosition**  
*A tile helye a pályán Nem képernyő koordinátákat tárol.*
- [BufferedImage](#) **sprite**  
*A sprite ami a képernyőn ezt a Tile-t reprezentálja.*

### Additional Inherited Members

### Protected Member Functions inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) **getDrawPosition** ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell.*

### 3.6.1 Detailed Description

Ehető elemeket reprezentáló osztály.

### 3.6.2 Constructor & Destructor Documentation

#### 3.6.2.1 Edible()

```
pacman.game.tile.edible.Edible.Edible (
    Coordinate mapPosition,
    BufferedImage sprite )
```

Edible default konstruktor.

##### Parameters

<i>mapPosition</i>	az Edible helye a pályám
<i>sprite</i>	az Edible-t reprezentáló sprite a képernyőn

### 3.6.3 Member Function Documentation

#### 3.6.3.1 isEaten()

```
boolean pacman.game.tile.edible.Edible.isEaten ( )
```

##### Returns

True, ha az Edible el van fogyasztva

#### 3.6.3.2 render()

```
void pacman.game.tile.edible.Edible.render (
    Graphics2D graphics )
```

A képernyőre rajzolja az Edible-t.

##### Parameters

<i>graphics</i>	Erre történik a festés
-----------------	------------------------

Reimplemented from [pacman.game.tile.Tile](#).

Reimplemented in [pacman.game.tile.edible.Fruit](#).



## 3.7 pacman.game.tile.edible.EdibleState Enum Reference

Edible-k állapotait kezelő enum.

### Public Attributes

- **EATEN**
- **ACTIVE**

### 3.7.1 Detailed Description

Edible-k állapotait kezelő enum.

## 3.8 pacman.game.tile.EmptyTile Class Reference

Üres Tile-t reprezentáló osztály.

### Public Member Functions

- **EmptyTile** ([Coordinate mapPosition](#), [BufferedImage sprite](#))  
*EmptyTile default konstruktor Megegyezik a Tile konstruktorával.*

### Public Member Functions inherited from [pacman.game.tile.Tile](#)

- **Tile** ([Coordinate mapPosition](#), [BufferedImage sprite](#))  
*Tile default konstruktor.*
- boolean **isWalkable** ()  
*Igaz, ha entitások átsétálhatnak ezen a tile-n.*
- void **render** ([Graphics2D graphics](#))  
*A Tile megjelenítése a képernyőn.*
- [BufferedImage](#) **getSprite** ()  
*Visszaadja a Tile sprite-ját.*

### Additional Inherited Members

### Protected Member Functions inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) **getDrawPosition** ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell.*

### Protected Attributes inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) **mapPosition**  
*A tile helye a pályán Nem képernyő koordinátákat tárol.*
- [BufferedImage](#) **sprite**  
*A sprite ami a képernyőn ezt a Tile-t reprezentálja.*

### 3.8.1 Detailed Description

Üres Tile-t reprezentáló osztály.

## 3.9 pacman.game.entity.Entity Class Reference

Mozgó entitást reprezentáló osztály.

### Public Member Functions

- **Entity** ()  
*Entity default konstruktor.*
- abstract void **update** ()  
*Az Entity-hez tartozó logika végrehajtása (pl.*
- **Direction** **getDirection** ()  
*Direction getter.*
- abstract void **reset** ()  
*Kezdőállapotba helyezi az Entity-t.*
- void **render** (Graphics2D graphics)  
*Az Entity megjelenítése a képernyőn.*
- Coordinate **getMapPosition** ()  
*Átkonvertálja a képernyő koordinátákat pálya koordinátákra.*
- Rectangle **getBounds** ()  
*Collision detection-höz használt függvény.*

### Protected Member Functions

- abstract void **init** ()  
*Inicializáló metódus.*
- abstract void **initSprites** ()  
*Sprite-okat inicializáló metódus.*
- abstract void **toStartingPos** ()  
*Kezdőállapotba helyezi az Entity-t.*
- void **updateSprite** ()  
*A sprite frissítése.*
- Coordinate **getDrawPosition** ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell A Tile és az Entity sprite-ok méreteinek különbsége miatt külön ki kell számolni, hova kell rajzolni az Entity-ket.*
- void **checkWallCollisions** ()  
*Fallal való ütközést vizsgálja Ha ütközést talál, visszaállítja az Entityt az előző helyére.*
- void **checkOutOfFrame** ()  
*Azt vizsgálja, mikor megy ki az Entity a képernyőről Ha kimegy, a pálya másik oldalára rakja azt.*

## Protected Attributes

- Coordinate **position**  
*Az Entity-t reprezentáló sprite középpontjának helye a képernyőn Valójában nem teljesen a középpontját tároljuk, a tárolt koordinátát úgy kapjuk meg, hogy a sprite bal felső sarkának koordinátáiból kivonunk a (ENTITY\_SIZE - TILE\_SIZE) \* SCALE egész számot.*
- **Direction** **direction**  
*Az Entity aktuális haladási iránya.*
- int **speed**  
*Az entitás aktuális sebessége (pixel / frame)*
- Map< **Direction**, ArrayList< BufferedImage > > **defaultSprites**  
*Az Entity sprite-jai irányonként rendezve A megfelelő irányt megadva a map-ből megkapjuk azon sprite-ok sorozatát, melyeken index szerint végighaladva egy animációt kapunk.*
- ArrayList< BufferedImage > **spriteList**  
*A jelenlegi iránynak megfelelő sprite-ok listája Ezen spite-ok egymásutánja adja ki a karakter animációját.*
- int **spriteIndex** = 0  
*A jelenleg használt sprite indexe az iránynak megfelelő sprite-ok listájában.*
- BufferedImage **sprite**  
*Az aktuális sprite.*
- int **animationDrawCounter** = 0  
*A legutóbbi sprite váltás óta eltelt framek száma.*

### 3.9.1 Detailed Description

Mozgó entitást reprezentáló osztály.

### 3.9.2 Member Function Documentation

#### 3.9.2.1 getBounds()

```
Rectangle pacman.game.entity.Entity.getBounds ( )
```

Collision detection-höz használt függvény.

#### Returns

Az Entity "szilárd" része

#### 3.9.2.2 getMapPosition()

```
Coordinate pacman.game.entity.Entity.getMapPosition ( )
```

Átkonvertálja a képernyő koordinátákat pálya koordinátákra.

#### Returns

A tile helye a pályán

### 3.9.2.3 init()

```
abstract void pacman.game.entity.Entity.init ( ) [abstract], [protected]
```

Inicializáló metódus.

Reimplemented in [pacman.game.entity.ghost.Ghost](#), and [pacman.game.entity.pacman.Pacman](#).

### 3.9.2.4 initSprites()

```
abstract void pacman.game.entity.Entity.initSprites ( ) [abstract], [protected]
```

Sprite-okat inicializáló metódus.

Reimplemented in [pacman.game.entity.ghost.Blinky](#), [pacman.game.entity.ghost.Clyde](#), [pacman.game.entity.ghost.Inky](#), [pacman.game.entity.ghost.Pinky](#), and [pacman.game.entity.pacman.Pacman](#).

### 3.9.2.5 render()

```
void pacman.game.entity.Entity.render (
    Graphics2D graphics )
```

Az Entity megjelenítése a képernyőn.

#### Parameters

<i>graphics</i>	Erre történik a festés
-----------------	------------------------

### 3.9.2.6 reset()

```
abstract void pacman.game.entity.Entity.reset ( ) [abstract]
```

Kezdőállapotba helyezi az Entity-t.

Reimplemented in [pacman.game.entity.ghost.Ghost](#), and [pacman.game.entity.pacman.Pacman](#).

### 3.9.2.7 toStartingPos()

```
abstract void pacman.game.entity.Entity.toStartingPos ( ) [abstract], [protected]
```

Kezdőállapotba helyezi az Entity-t.

Reimplemented in [pacman.game.entity.ghost.Blinky](#), [pacman.game.entity.ghost.Clyde](#), [pacman.game.entity.ghost.Inky](#), [pacman.game.entity.ghost.Pinky](#), and [pacman.game.entity.pacman.Pacman](#).

### 3.9.2.8 update()

```
abstract void pacman.game.entity.Entity.update ( ) [abstract]
```

Az Entity-hez tartozó logika végrehajtása (pl.

állapot frissítése, collision detection stb.)

Reimplemented in [pacman.game.entity.ghost.Ghost](#), and [pacman.game.entity.pacman.Pacman](#).

### 3.9.2.9 updateSprite()

```
void pacman.game.entity.Entity.updateSprite ( ) [protected]
```

A sprite frissítése.

Reimplemented in [pacman.game.entity.ghost.Ghost](#), and [pacman.game.entity.pacman.Pacman](#).

## 3.10 pacman.game.util.Error Enum Reference

Lehetséges hibajelzéseket, és azok üzeneteit tároló enum.

### Public Member Functions

- **Error** (String message)

### Public Attributes

- **LOADING\_SPRITES** =("Failed to load sprites")
- **LOADING\_MAP** =("Failed to load map")
- **LOADING\_LEADERBOARD** =("Failed to load leaderboard")
- **SAVING\_LEADERBOARD** =("Failed to save leaderboard")
- final String **message**

### 3.10.1 Detailed Description

Lehetséges hibajelzéseket, és azok üzeneteit tároló enum.

## 3.11 pacman.game.tile.edible.Fruit Class Reference

Gyümölcsöt reprezentáló osztály A ggygyümölcsök mind ugyan ott jelennek meg a pályán bizonyos időközönként.

### Public Member Functions

- **Fruit** ([Coordinate mapPosition](#), [BufferedImage sprite](#), [String type](#))  
*Fruit default konstruktor.*
- void **render** ([Graphics2D graphics](#))  
*A képernyőre rajzolja a gyümölcsöt.*
- void **actionPerformed** ([ActionEvent e](#))  
*A gyümölcs eltüntetéséhez használt metódus.*
- void **placed** ()  
*A gyümölcs elhelyezésénél meghívott metódus Elindítja a gyümölcs időzítőjét, ami majd eltünteti azt.*

### Public Member Functions inherited from [pacman.game.tile.edible.Edible](#)

- **Edible** ([Coordinate mapPosition](#), [BufferedImage sprite](#))  
*Edible default konstruktor.*
- int **getScoreModifier** ()  
*Visszaadja azt, hogy az edible elfogyasztása hány pontot ér.*
- boolean **isEaten** ()
- void **toEatenState** ()  
*EATEN állapotba állítja az Edible-t.*

### Public Member Functions inherited from [pacman.game.tile.Tile](#)

- **Tile** ([Coordinate mapPosition](#), [BufferedImage sprite](#))  
*Tile default konstruktor.*
- boolean **isWalkable** ()  
*Igaz, ha entitások átsétálhatnak ezen a tile-n.*
- [BufferedImage](#) **getSprite** ()  
*Visszaadja a Tile sprite-ját.*

### Static Public Attributes

- static [Coordinate](#) **location**  
*A gyümölcsök helye a pályán.*

### Protected Member Functions

- [Coordinate](#) **getDrawPosition** ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a gyümölcs rajzolását kezdeni kell.*

### Static Private Attributes

- static final [Random](#) **random** = new [Random](#)()  
*Random generátor A gyümölcsök eltűnési idejéhez kell.*

## Additional Inherited Members

### Protected Attributes inherited from [pacman.game.tile.edible.Edible](#)

- [EdibleState](#) **state**  
*Az Edible aktuális állapota.*
- `int` **scoreModifier**  
*Az Edible elfogyasztása ennyivel módosítja a pontszámot.*

### Protected Attributes inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) **mapPosition**  
*A tile helye a pályán Nem képernyő koordinátákat tárol.*
- `BufferedImage` **sprite**  
*A sprite ami a képernyőn ezt a Tile-t reprezentálja.*

## 3.11.1 Detailed Description

Gyümölcsöt reprezentáló osztály A gggyümölcsök mind ugyan ott jelennek meg a pályán bizonyos időközönként.

## 3.11.2 Constructor & Destructor Documentation

### 3.11.2.1 Fruit()

```
pacman.game.tile.edible.Fruit.Fruit (
    Coordinate mapPosition,
    BufferedImage sprite,
    String type )
```

Fruit default konstruktor.

#### Parameters

<i>type</i>	A gyümölcs neve
-------------	-----------------

## 3.11.3 Member Function Documentation

### 3.11.3.1 getDrawPosition()

```
Coordinate pacman.game.tile.edible.Fruit.getDrawPosition ( ) [protected]
```

Visszaadja azt a képernyő koordinátát, ahonnan a gyümölcs rajzolását kezdeni kell.

Reimplemented from [pacman.game.tile.Tile](#).

### 3.11.3.2 render()

```
void pacman.game.tile.edible.Fruit.render (
    Graphics2D graphics )
```

A képernyőre rajzolja a gyümölcsöt.

#### Parameters

<i>graphics</i>	Erre történik a festés
-----------------	------------------------

Reimplemented from [pacman.game.tile.edible.Edible](#).

## 3.12 pacman.game.Game Class Reference

A játékot kezelő osztály.

### Static Public Member Functions

- static void **init** ()  
*Inicializáló metódus.*
- static void **start** ()  
*Játék indító metódus.*
- static void **update** ()  
*Frissíti a pályán lévő összes Entity állapotát.*
- static void **render** ()  
*Újrarajzolja a pályát.*
- static void **reset** ()  
*Kezdőállapotba állítja az Entity-ket Ha a játék véget ért, kilép.*
- static void **pause** ()  
*Megállítja a játékot.*
- static void **resume** ()  
*Újraindítja a játékot.*
- static void **quitToMenu** (boolean saveScore)  
*Kilép a menübe.*
- static void **modifyScore** (int modifier)  
*Pontszámot módosító metódus.*
- static int **getScore** ()  
*Score getter.*
- static int **getRemainingPellets** ()
- static void **setRemainingPellets** (int pellets)  
*Beállítja a megmaradt pelletek számát.*



### Static Public Attributes

- static GameFrame **frame**  
*Ebben az ablakban jelenik meg a játék.*
- static [GameThread](#) **thread**  
*Ezen a szálon fut a game loop.*
- static volatile [GameState](#) **state**  
*A játék aktuális állapota.*
- static ArrayList< ArrayList< Tile > > **map** = new ArrayList<>()  
*Pálya.*
- static ArrayList< Ghost > **ghosts** = new ArrayList<>()  
*Szellemek listája.*
- static ArrayList< Fruit > **fruit**  
*Gyümölcsök listája.*
- static int **remainingPellets**  
*Maradék Pellet-ek száma.*
- static int **maxPellets** = 0  
*Pelletek maximális száma.*
- static Pacman **pacman**  
*Pacman referenciája.*

### Private Member Functions

- **Game** ()  
*Impicit konstruktor elrejtése miatt privát.*

### Static Private Member Functions

- static void **initGame** ()  
*A játékot inicializáló módszer.*
- static void **initEntities** ()  
*Az entitásokat inicializáló módszer.*
- static void **placeFruit** ()  
*Gyümölcs elhelyező logika végrehajtása.*

### Static Private Attributes

- static int **score**  
*Játékos aktuális.*
- static boolean **pacmanWasHurt**  
*Igaz, ha Pacman-t éppen megsebeztek.*

## 3.12.1 Detailed Description

A játékot kezelő osztály.

### 3.12.2 Member Function Documentation

#### 3.12.2.1 getRemainingPellets()

```
static int pacman.game.Game.getRemainingPellets ( ) [static]
```

##### Returns

megmaradt Pellet-ek száma

#### 3.12.2.2 modifyScore()

```
static void pacman.game.Game.modifyScore (
    int modifier ) [static]
```

Pontszámot módosító metódus.

##### Parameters

<i>modifier</i>	ezt a számot adja hozzá a pontszámhoz
-----------------	---------------------------------------

#### 3.12.2.3 quitToMenu()

```
static void pacman.game.Game.quitToMenu (
    boolean saveScore ) [static]
```

Kilép a menübe.

##### Parameters

<i>saveScore</i>	ha true, a játékos pontszáma el lesz mentve
------------------	---

## 3.13 pacman.game.display.GameFrame Class Reference

Ebben az ablakban jelenik meg a játék Ez az osztály felelős még a hiba dialog box-ok megjelenítéséért is.

### Public Member Functions

- **GameFrame ()**  
*GameFrame konstruktor.*
- void **showMainMenu ()**  
*A főmenüt mutató metódus.*
- void **showGame ()**  
*A játékot mutató metódus.*

### Public Attributes

- [MainMenu](#) **mainMenu**  
*Főmenü referenciája.*
- [GamePanel](#) **gamePanel**  
*GamePanel referenciája.*

### Static Package Attributes

- static Font **font**  
*A játékban használt betűtípus.*

### Private Member Functions

- void **loadFont** ()  
*A játékban használt betűtípus betöltése.*

## 3.13.1 Detailed Description

Ebben az ablakban jelenik meg a játék Ez az osztály felelős még a hiba dialog box-ok megjelenítéséért is.

## 3.14 pacman.game.display.GamePanel Class Reference

Ezen a panelen jelenik meg a pálya és az alsó és felső szalagok.

### Public Member Functions

- **GamePanel** ()  
*GamePanel default konstruktor.*
- void **printReady** ()  
*Kiírja a "Ready" szöveget a következő rendernél.*
- void **printGameOver** ()  
*Kiírja a "Game Over" szöveget a következő rendernél.*
- void **printPause** ()  
*Kiírja a "Paused" szöveget a következő rendernél.*

### Protected Member Functions

- void **paintComponent** (Graphics g)  
*A GamePanel-t renderelő metódus.*

### Private Member Functions

- void **paintReady** (Graphics2D graphics)  
*Kiírja a "Ready" szöveget.*
- void **paintGameOver** (Graphics2D graphics)  
*Kiírja a "Game over" szöveget.*
- void **paintPause** (Graphics2D graphics)  
*Kiírja a "Paused" szöveget.*

### Private Attributes

- boolean **printReady** = false  
*Igaz, ha ki kell írni a "Ready" szöveget.*
- boolean **printGameOver** = false  
*Igaz, ha ki kell írni a "Game Over" szöveget.*
- boolean **printPause** = false  
*Igaz, ha ki kell írni a "Paused" szöveget.*

## 3.14.1 Detailed Description

Ezen a panelen jelenik meg a pálya és az alsó és felső szalagok.

## 3.14.2 Member Function Documentation

### 3.14.2.1 paintGameOver()

```
void pacman.game.display.GamePanel.paintGameOver (  
    Graphics2D graphics ) [private]
```

Kiírja a "Game over" szöveget.

#### Parameters

<i>graphics</i>	
-----------------	--

## 3.15 pacman.game.GameState Enum Reference

A játék állapotát reprezentáló enum.

### Public Attributes

- STOPPED
- RUNNING
- PAUSED

### 3.15.1 Detailed Description

A játék állapotát reprezentáló enum.

## 3.16 pacman.game.GameThread Class Reference

A game loop-ért felelős szál.

### Public Member Functions

- void **run** ()  
*A game loop indítása.*
- void **doWait** (int millis)  
*Ezt a függvényt meghívva lehet a game loop-ot várakoztatni.*

### Private Attributes

- final Timer **timer** = new Timer()  
*Időzítő, a game loop várakoztatásához kell.*
- volatile boolean **waiting** = false  
*True, ha a game loop éppen várakozik.*

### Static Private Attributes

- static double **deltaTime**  
*A legutóbbi render-update blokk óta eltelt idő (ns)*
- static final double **timeSlice** = 500000000 / (double) Config.DISPLAY\_TARGET\_FPS  
*Két render-update blokk között eltelt idő optimális esetben (ns)*

### 3.16.1 Detailed Description

A game loop-ért felelős szál.

### 3.16.2 Member Function Documentation

#### 3.16.2.1 doWait()

```
void pacman.game.GameThread.doWait (  
    int millis )
```

Ezt a függvényt meghívva lehet a game loop-ot várakoztatni.

## Parameters

<i>millis</i>	Várakozás ideje (ms)
---------------	----------------------

## 3.17 pacman.game.entity.ghost.Ghost Class Reference

Szellemet reprezentáló osztály.

### Public Member Functions

- [Ghost](#) (Pacman [pacman](#))  
*Ghost default konstruktor.*
- void [reset](#) ()  
*Kezdő állapotba helyezi a szellemeket.*
- void [update](#) ()  
*Minden belső logikát kezelő függvény.*
- void **actionPerformed** (ActionEvent e)  
*Frissíti a szellem állapotát A játék elején a szellemek a CHASE és SCATTER állapotok között váltanak, majd CHASE állapotban maradnak.*
- void [toFrightenedState](#) ()  
*FRIGHTENED állapotba váltja a szellemet.*
- void **interact** ()  
*Interakció Pacman és a szellem között Pacman hívja meg, amikor egy Tile-re kerülnek a szellemmel.*
- [GhostState](#) [getState](#) ()  
*State getter.*

### Public Member Functions inherited from [pacman.game.entity.Entity](#)

- **Entity** ()  
*Entity default konstruktor.*
- [Direction](#) [getDirection](#) ()  
*Direction getter.*
- void [render](#) (Graphics2D graphics)  
*Az Entity megjelenítése a képernyőn.*
- Coordinate [getMapPosition](#) ()  
*Átkonvertálja a képernyő koordinátákat pálya koordinátákra.*
- Rectangle [getBounds](#) ()  
*Collision detection-höz használt függvény.*

## Protected Member Functions

- void **updateSprite** ()  
*Frissíti a szellem aktuális sprite-ját Több féle sprite-ja van, más logika szerint működik, mint az Entity metódusa.*
- void **init** ()  
*Inicializáló metódus.*
- abstract Coordinate **getChaseTarget** ()  
*Visszaadja a szellem target celláját CHASE állapotban.*
- abstract Coordinate **getScatterTarget** ()  
*Visszaadja a szellem target celláját SCATTER állapotban.*
- void **updateDirection** ()  
*Frissíti a szellem irányát.*
- void **updateTargetTile** ()  
*Frissíti a szellem célját.*
- boolean **isInsideHouse** ()
- boolean **isInFrontOfHouse** ()

## Protected Member Functions inherited from [pacman.game.entity.Entity](#)

- abstract void **initSprites** ()  
*Sprite-okat inicializáló metódus.*
- abstract void **toStartingPos** ()  
*Kezdőállapotba helyezi az Entity-t.*
- Coordinate **getDrawPosition** ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell A Tile és az Entity sprite-ok méreteinek különbsége miatt külön ki kell számolni, hova kell rajzolni az Entity-ket.*
- void **checkWallCollisions** ()  
*Fallal való ütközést vizsgálja Ha ütközést talál, visszaállítja az Entityt az előző helyére.*
- void **checkOutOfFrame** ()  
*Azt vizsgálja, mikor megy ki az Entity a képernyőről Ha kimegy, a pálya másik oldalára rakja azt.*

## Protected Attributes

- Pacman **pacman**  
*Pacman referenciája.*
- List< BufferedImage > **frightenedSprites**  
*A FRIGHTENED állapotú szellemek sprite-jai.*
- Map< [Direction](#), BufferedImage > **eatenSprites**  
*Az EATEN állapotú szellemek sprite-jai.*

## Protected Attributes inherited from [pacman.game.entity.Entity](#)

- Coordinate **position**  
*Az Entity-t reprezentáló sprite középpontjának helye a képernyőn Valójában nem teljesen a középpontját tároljuk, a tárolt koordinátát úgy kapjuk meg, hogy a sprite bal felső sarkának koordinátáiból kivonunk a (ENTITY\_SIZE - TILE\_SIZE) \* SCALE egész számot.*
- [Direction](#) **direction**  
*Az Entity aktuális haladási iránya.*
- int **speed**  
*Az entitás aktuális sebessége (pixel / frame)*

- Map< [Direction](#), ArrayList< BufferedImage > > **defaultSprites**  
*Az Entity sprite-jai irányonként rendezve A megfelelő irányt megadva a map-ből megkapjuk azon sprite-ok sorozatát, melyeken index szerint végighaladva egy animációt kapunk.*
- ArrayList< BufferedImage > **spriteList**  
*A jelenlegi iránynak megfelelő sprite-ok listája Ezen spite-ok egymásutánja adja ki a karakter animációját.*
- int **spriteIndex** = 0  
*A jelenleg használt sprite indexe az iránynak megfelelő sprite-ok listájában.*
- BufferedImage **sprite**  
*Az aktuális sprite.*
- int **animationDrawCounter** = 0  
*A legutóbbi sprite váltás óta eltelt framek száma.*

### Private Member Functions

- void **updateEatenSprite** ()  
*Sprite frissítése EATEN állapotban.*
- void **updateFrightenedSprite** ()  
*Sprite frissítése FRIGHTENED állapotban.*
- void **toEatenState** ()  
*EATEN állapotba váltja a szellemet.*
- void **endEatenState** ()
- void **updateSpeed** ()  
*Frissíti a szellem sebességét az állapotától függően.*
- List< [Direction](#) > **getValidMoves** ()  
*Visszaadja az érvényes lépéseket a jelenlegi Tile-ből kiindulva.*
- [Direction](#) **bestMove** (List< [Direction](#) > moves)  
*Visszaadja a legjobb lépést A lépéseket a target-től való távolságuk, azon belül a prioritásuk szerint választja ki.*
- boolean **isInNoTurnZone** ()
- boolean **isInSlowZone** ()

### Private Attributes

- [GhostState](#) **state**  
*A szellem aktuális állapota.*
- Timer **stateTimer**  
*Állapotváltást kezelő timer.*
- long **enterFrightened**  
*Ebben az időpontban váltott FRIGHTENED állapotba a szellem.*
- final ArrayList< [GhostState](#) > **nextStates** = new ArrayList<>()  
*A szellem következő állapotai.*
- Coordinate **target**  
*Cella, ami felé megy a szellem.*

### 3.17.1 Detailed Description

Szellemet reprezentáló osztály.



### 3.17.2 Constructor & Destructor Documentation

#### 3.17.2.1 Ghost()

```
pacman.game.entity.ghost.Ghost.Ghost (
    Pacman pacman )
```

Ghost default konstruktor.

## Parameters

<i>pacman</i>
---------------

Pacman referenciája
---------------------

### 3.17.3 Member Function Documentation

#### 3.17.3.1 getChaseTarget()

```
abstract Coordinate pacman.game.entity.ghost.Ghost.getChaseTarget ( ) [abstract], [protected]
```

Visszaadja a szellem target celláját CHASE állapotban.

Reimplemented in [pacman.game.entity.ghost.Blinky](#), [pacman.game.entity.ghost.Clyde](#), [pacman.game.entity.ghost.Inky](#), and [pacman.game.entity.ghost.Pinky](#).

#### 3.17.3.2 getScatterTarget()

```
abstract Coordinate pacman.game.entity.ghost.Ghost.getScatterTarget ( ) [abstract], [protected]
```

Visszaadja a szellem target celláját SCATTER állapotban.

Reimplemented in [pacman.game.entity.ghost.Blinky](#), [pacman.game.entity.ghost.Clyde](#), [pacman.game.entity.ghost.Inky](#), and [pacman.game.entity.ghost.Pinky](#).

#### 3.17.3.3 getState()

```
GhostState pacman.game.entity.ghost.Ghost.getState ( )
```

State getter.

## Returns

a szellem aktuális állapota

#### 3.17.3.4 getValidMoves()

```
List< Direction > pacman.game.entity.ghost.Ghost.getValidMoves ( ) [private]
```

Visszaadja az érvényes lépéseket a jelenlegi Tile-ből kiindulva.

## Returns

Érvényes lépések listája

### 3.17.3.5 init()

```
void pacman.game.entity.ghost.Ghost.init ( ) [protected]
```

Inicializáló metódus.

Reimplemented from [pacman.game.entity.Entity](#).

### 3.17.3.6 isInFrontOfHouse()

```
boolean pacman.game.entity.ghost.Ghost.isInFrontOfHouse ( ) [protected]
```

#### Returns

True, ha a ház előtt van

### 3.17.3.7 isInNoTurnZone()

```
boolean pacman.game.entity.ghost.Ghost.isInNoTurnZone ( ) [private]
```

#### Returns

True, ha egy olyan zónában van, ahol nem fordulhat

### 3.17.3.8 isInsideHouse()

```
boolean pacman.game.entity.ghost.Ghost.isInsideHouse ( ) [protected]
```

#### Returns

True, ha a házban van

### 3.17.3.9 isInSlowZone()

```
boolean pacman.game.entity.ghost.Ghost.isInSlowZone ( ) [private]
```

#### Returns

True, ha egy olyan zónában van, ahol lassan megy

### 3.17.3.10 reset()

```
void pacman.game.entity.ghost.Ghost.reset ( )
```

Kezdő állapotba helyezi a szellemeket.

Reimplemented from [pacman.game.entity.Entity](#).

### 3.17.3.11 toEatenState()

```
void pacman.game.entity.ghost.Ghost.toEatenState ( ) [private]
```

EATEN állapotba váltja a szellemet.

Ez a metódus a szellemet a legközelebbi Tile közepére helyezi, így biztosan el tud majd fordulni.

### 3.17.3.12 toFrightenedState()

```
void pacman.game.entity.ghost.Ghost.toFrightenedState ( )
```

FRIGHTENED állapotba váltja a szellemet.

Pacman hívja meg, amikor elfogyaszt egy PowerPellet-et.

### 3.17.3.13 update()

```
void pacman.game.entity.ghost.Ghost.update ( )
```

Minden belső logikát kezelő függvény.

Reimplemented from [pacman.game.entity.Entity](#).

### 3.17.3.14 updateSprite()

```
void pacman.game.entity.ghost.Ghost.updateSprite ( ) [protected]
```

Frissíti a szellem aktuális sprite-ját Több féle sprite-ja van, más logika szerint működik, mint az Entity metódusa.

Reimplemented from [pacman.game.entity.Entity](#).

## 3.18 pacman.game.entity.ghost.GhostState Enum Reference

A szellemek állapotait tároló enum.

### Public Member Functions

- **GhostState** (int [delay](#))
- int [getDelay](#) ()

### Public Attributes

- **CHASE** =(Config.GHOST\_CHASE\_STATE\_DELAY)
- **SCATTER** =(Config.GHOST\_SCATTER\_STATE\_DELAY)
- **FRIGHTENED** =(Config.GHOST\_FRIGHTENED\_STATE\_DELAY)
- **EATEN** =(Config.GHOST\_STATE\_INFINITE\_DELAY)
- **INFINITE\_CHASE** =(Config.GHOST\_STATE\_INFINITE\_DELAY)

### Private Attributes

- final int **delay**  
*Ennyi ideig tart az adott állapot.*

### 3.18.1 Detailed Description

A szellemek állapotait tároló enum.

### 3.18.2 Member Function Documentation

#### 3.18.2.1 getDelay()

```
int pacman.game.entity.ghost.GhostState.getDelay ( )
```

#### Returns

Az állapot időtartama

## 3.19 pacman.game.entity.ghost.Inky Class Reference

Inky (kék) szellemet reprezentáló osztály.

### Public Member Functions

- [Inky](#) (Pacman [pacman](#), [Blinky](#) [blinky](#))  
*Inky default konstruktor.*

### Public Member Functions inherited from [pacman.game.entity.ghost.Ghost](#)

- [Ghost](#) (Pacman [pacman](#))  
*Ghost default konstruktor.*
- void [reset](#) ()  
*Kezdő állapotba helyezi a szellemeket.*
- void [update](#) ()  
*Minden belső logikát kezelő függvény.*
- void **actionPerformed** (ActionEvent e)  
*Frissíti a szellem állapotát A játék elején a szellemek a CHASE és SCATTER állapotok között váltanak, majd CHASE állapotban maradnak.*
- void [toFrightenedState](#) ()  
*FRIGHTENED állapotba váltja a szellemet.*
- void **interact** ()  
*Interakció Pacman és a szellem között Pacman hívja meg, amikor egy Tile-re kerülnek a szellemmel.*
- [GhostState](#) [getState](#) ()  
*State getter.*

## Public Member Functions inherited from [pacman.game.entity.Entity](#)

- **Entity** ()  
*Entity default konstruktor.*
- **Direction** **getDirection** ()  
*Direction getter.*
- void **render** (Graphics2D graphics)  
*Az Entity megjelenítése a képernyőn.*
- Coordinate **getMapPosition** ()  
*Átkonvertálja a képernyő koordinátákat pálya koordinátákra.*
- Rectangle **getBounds** ()  
*Collision detection-höz használt függvény.*

## Protected Member Functions

- void **toStartingPos** ()  
*Kezdőállapotba állítja a szellemet.*
- void **initSprites** ()  
*Sprite-okat inicializáló metódus.*
- Coordinate **getChaseTarget** ()
- Coordinate **getScatterTarget** ()

## Protected Member Functions inherited from [pacman.game.entity.ghost.Ghost](#)

- void **updateSprite** ()  
*Frissíti a szellem aktuális sprite-ját Több féle sprite-ja van, más logika szerint működik, mint az Entity metódusa.*
- void **init** ()  
*Inicializáló metódus.*
- void **updateDirection** ()  
*Frissíti a szellem irányát.*
- void **updateTargetTile** ()  
*Frissíti a szellem célját.*
- boolean **isInsideHouse** ()
- boolean **isInFrontOfHouse** ()

## Protected Member Functions inherited from [pacman.game.entity.Entity](#)

- Coordinate **getDrawPosition** ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell A Tile és az Entity sprite-ok méreteinek különbsége miatt külön ki kell számolni, hova kell rajzolni az Entity-ket.*
- void **checkWallCollisions** ()  
*Fallal való ütközést vizsgálja Ha ütközést talál, visszaállítja az Entityt az előző helyére.*
- void **checkOutOfFrame** ()  
*Azt vizsgálja, mikor megy ki az Entity a képernyőről Ha kimegy, a pálya másik oldalára rakja azt.*

## Package Attributes

- **Blinky** **blinky**  
*Blinky referenciája.*

## Additional Inherited Members

### Protected Attributes inherited from [pacman.game.entity.ghost.Ghost](#)

- Pacman **pacman**  
*Pacman referenciája.*
- List< BufferedImage > **frightenedSprites**  
*A FRIGHTENED állapotú szellemek sprite-jai.*
- Map< [Direction](#), BufferedImage > **eatenSprites**  
*Az EATEN állapotú szellemek sprite-jai.*

### Protected Attributes inherited from [pacman.game.entity.Entity](#)

- Coordinate **position**  
*Az Entity-t reprezentáló sprite középpontjának helye a képernyőn Valójában nem teljesen a középpontját tároljuk, a tárolt koordinátát úgy kapjuk meg, hogy a sprite bal felső sarkának koordinátáiból kivonunk a (ENTITY\_SIZE - TILE\_SIZE) \* SCALE egész számot.*
- [Direction](#) **direction**  
*Az Entity aktuális haladási iránya.*
- int **speed**  
*Az entitás aktuális sebessége (pixel / frame)*
- Map< [Direction](#), ArrayList< BufferedImage > > **defaultSprites**  
*Az Entity sprite-jai irányonként rendezve A megfelelő irányt megadva a map-ből megkapjuk azon sprite-ok sorozatát, melyeken index szerint végighaladva egy animációt kapunk.*
- ArrayList< BufferedImage > **spriteList**  
*A jelenlegi iránynak megfelelő sprite-ok listája Ezen spite-ok egymásutánja adja ki a karakter animációját.*
- int **spriteIndex** = 0  
*A jelenleg használt sprite indexe az iránynak megfelelő sprite-ok listájában.*
- BufferedImage **sprite**  
*Az aktuális sprite.*
- int **animationDrawCounter** = 0  
*A legutóbbi sprite váltás óta eltelt framek száma.*

## 3.19.1 Detailed Description

Inky (kék) szellemet reprezentáló osztály.

## 3.19.2 Constructor & Destructor Documentation

### 3.19.2.1 Inky()

```
pacman.game.entity.ghost.Inky.Inky (
    Pacman pacman,
    Blinky blinky )
```

Inky default konstruktor.

## Parameters

<i>pacman</i>	Pacman referenciája
<i>blinky</i>	Blinky referenciája

### 3.19.3 Member Function Documentation

#### 3.19.3.1 getChaseTarget()

Coordinate pacman.game.entity.ghost.Inky.getChaseTarget ( ) [protected]

## Returns

A szellem célja CHASE állapotban

Reimplemented from [pacman.game.entity.ghost.Ghost](#).

#### 3.19.3.2 getScatterTarget()

Coordinate pacman.game.entity.ghost.Inky.getScatterTarget ( ) [protected]

## Returns

A szellem célja SCATTER állapotban

Reimplemented from [pacman.game.entity.ghost.Ghost](#).

#### 3.19.3.3 initSprites()

void pacman.game.entity.ghost.Inky.initSprites ( ) [protected]

Sprite-okat inicializáló metódus.

Reimplemented from [pacman.game.entity.Entity](#).

#### 3.19.3.4 toStartingPos()

void pacman.game.entity.ghost.Inky.toStartingPos ( ) [protected]

Kezdőállapotba állítja a szellemet.

Reimplemented from [pacman.game.entity.Entity](#).

## 3.20 pacman.game.input.InputHandler Class Reference

Bemenetet kezelő osztály.



## Public Member Functions

- [InputHandler](#) (GameFrame [frame](#))  
*InputHandler default konstruktor.*
- void [keyTyped](#) (KeyEvent e)  
*Üres metódus.*
- void [keyPressed](#) (KeyEvent e)  
*Billentyű lenyomásánál meghívott metódus.*
- void [keyReleased](#) (KeyEvent e)  
*Billentyű felengedésénél meghívott metódus.*

## Static Public Attributes

- static boolean [upPressed](#)  
*Igaz, ha valamelyik "fel" gomb éppen le van nyomva.*
- static boolean [downPressed](#)  
*Igaz, ha valamelyik "le" gomb éppen le van nyomva.*
- static boolean [rightPressed](#)  
*Igaz, ha valamelyik "jobbra" gomb éppen le van nyomva.*
- static boolean [leftPressed](#)  
*Igaz, ha valamelyik "balra" gomb éppen le van nyomva.*
- static boolean [escPressed](#)  
*Igaz, ha az "Esc" billentyű le van nyomva.*
- static boolean [enterPressed](#)  
*Igaz, ha az "Enter" le van nyomva.*

## Private Attributes

- final GameFrame [frame](#)  
*A játékot megjelenítő ablak.*

### 3.20.1 Detailed Description

Bemenetet kezelő osztály.

### 3.20.2 Constructor & Destructor Documentation

#### 3.20.2.1 InputHandler()

```
pacman.game.input.InputHandler.InputHandler (
    GameFrame frame )
```

InputHandler default konstruktor.

#### Parameters

<i>frame</i>	A játékot megjelenítő ablak
--------------	-----------------------------

### 3.20.3 Member Function Documentation

#### 3.20.3.1 keyTyped()

```
void pacman.game.input.InputHandler.keyTyped (
    KeyEvent e )
```

Üres metódus.

##### Parameters

<i>e</i>	the event to be processed
----------	---------------------------

## 3.21 pacman.game.Leaderboard Class Reference

Pontszámok tárolásáért felelős osztály.

### Static Public Member Functions

- static void **load** ()  
*Betölti a leaderboard-ot.*
- static void **save** ()  
*Elmenti a leaderboard-ot.*
- static int **getHighScore** ()  
*Visszaadja a legmagasabb pontszámot.*
- static ArrayList< Map.Entry< String, Integer > > **getTopFive** ()  
*Visszaadja az öt legmagasabb pontszámot elért játékos nevét pontszámát.*
- static void **addScore** (String name, int score)  
*Hozzáad egy pontszámot a leaderboard-hoz.*

### Static Private Attributes

- static Map< String, Integer > **scores** = new HashMap<>()  
*HashMap, amiben a játékosok neveit és elért pontszámait tároljuk.*

#### 3.21.1 Detailed Description

Pontszámok tárolásáért felelős osztály.

### 3.21.2 Member Function Documentation

#### 3.21.2.1 addScore()

```
static void pacman.game.Leaderboard.addScore (
    String name,
    int score ) [static]
```

Hozzáad egy pontszámot a leaderboard-hoz.

## Parameters

<i>name</i>	Játékos neve
<i>score</i>	Játékos pontszáma

### 3.21.2.2 getHighScore()

```
static int pacman.game.Leaderboard.getHighScore ( ) [static]
```

Visszaadja a legmagasabb pontszámot.

## Returns

Legmagasabb pontszám a leaderboardon, ha üres akkor 0

## 3.22 pacman.game.display.LowerRibbon Class Reference

A képernyőn megjelenő alsó szalagot kezelő osztály.

### Public Member Functions

- **LowerRibbon ()**  
*LowerRibbon default konstruktor.*
- void **paintComponent** (Graphics g)  
*A LowerRibbon-t renderelő metódus.*

### Static Public Member Functions

- static void **updateFruitSprites** ()  
*Frissíti a gyümölcsök sprite-jait.*

### Private Member Functions

- void **updateDisplayedFruit** ()  
*Frissíti a megjelenítendő gyümölcsök számát.*

### Static Private Attributes

- static int **displayedFruit**  
*Megjelenített gyümölcsök száma.*
- static ArrayList< BufferedImage > **fruitSprites**  
*Gyümölcsök sprite-jai.*
- static BufferedImage **pacmanSprite**  
*Pacman egyik sprite-ja.*

### 3.22.1 Detailed Description

A képernyőn megjelenő alsó szalagot kezelő osztály.

## 3.23 pacman.game.display.MainMenu Class Reference

A főmenüt megjelenítő osztály.

### Public Member Functions

- **MainMenu** ()  
*Főmenü konstruktor.*
- void **paintComponent** (Graphics g)  
*A főmenüt renderelő metódus.*
- void **down** ()  
*Egy menüponttal lejjebb lép.*
- void **up** ()  
*Egy menüponttal feljebb lép.*
- void **select** ()  
*Kiválasztja a menüpontot.*
- void **back** ()  
*Visszamegy a főmenübe.*

### Private Member Functions

- void **paintTitleScreen** (Graphics2D graphics)  
*Kirajzolja a főmenüt.*
- void **paintLeaderboard** (Graphics2D graphics)  
*Kirajzolja a leaderboard-ot.*
- void **paintGuide** (Graphics2D graphics)  
*Kirajzolja az útmutatót.*

### Private Attributes

- final ArrayList< String > **items**  
*Menüpontok listája.*
- final String **START** = "START GAME"
- final String **LEADERBOARD** = "LEADERBOARD"
- final String **GUIDE** = "HOW TO PLAY"
- final String **QUIT** = "QUIT"
- boolean **inTitleScreen**  
*Igaz, ha a főmenüben vagyunk.*
- int **selected** = 0  
*A kiválasztott menüpont sorszáma.*
- final Color **itemColor**  
*A kiválasztott menüpontot kiemelő szín.*

### 3.23.1 Detailed Description

A főmenüt megjelenítő osztály.

## 3.24 pacman.game.util.MapLoader Class Reference

A pálya betöltéséért felelős osztály.

### Static Public Member Functions

- static void **loadMap** ()  
*Ez a metódus tölti be a pályát.*

### Static Private Member Functions

- static void **error** ()
- static String **getWallSpriteKey** (char c)  
*Egy fal sprite-jának kulcsát adja vissza.*
- static void **initFruit** (Coordinate coordinate)  
*Inicializálja a gyümölcsöket.*

### 3.24.1 Detailed Description

A pálya betöltéséért felelős osztály.

### 3.24.2 Member Function Documentation

#### 3.24.2.1 getWallSpriteKey()

```
static String pacman.game.util.MapLoader.getWallSpriteKey (  
    char c ) [static], [private]
```

Egy fal sprite-jának kulcsát adja vissza.

#### Parameters

<i>c</i>	A fal reprezentációja a map.txt-ben
----------	-------------------------------------

#### Returns

A fal sprite-jának kulcsa a SpriteLoader.tileSprites map-ben

### 3.24.2.2 initFruit()

```
static void pacman.game.util.MapLoader.initFruit (
    Coordinate coordinate ) [static], [private]
```

Inicializálja a gyümölcsöket.

#### Parameters

<i>coordinate</i>	Ezen a helyen fognak megjelenni a gyümölcsök
-------------------	--

## 3.25 pacman.game.display.MapPanel Class Reference

Ezen a panelen razolódik ki a pálya.

### Public Member Functions

- **MapPanel** ()  
*MapPanel default konstruktor.*
- void **paintComponent** (Graphics g)  
*A pályát renderelő metódus.*

### 3.25.1 Detailed Description

Ezen a panelen razolódik ki a pálya.

## 3.26 pacman.game.entity.pacman.Pacman Class Reference

Pacmant reprezentáló osztály.

### Public Member Functions

- **Pacman** ()  
*Pacman default konstruktor.*
- int **getLives** ()  
*Visszaadja Pacman életeinek számát.*
- void **reset** ()  
*Kezdőállapotba helyezi Pacman-t.*
- void **update** ()  
*Belső logika végrehajtása.*
- void **updateDirection** ()  
*Frissíti Pacman irányát.*
- void **hurt** ()  
*Pacman elveszt egy életet Az a szellem hívja meg, aki megsebezte.*
- boolean **isHurt** ()
- void **ghostEaten** ()  
*Módosítja a pontszámot a megfelelő számmal Az a szellem hívja meg, akit Pacman megevett.*
- boolean **isFruitEaten** ()
- void **setFruitEaten** (boolean **fruitEaten**)  
*FruitEaten setter.*

## Public Member Functions inherited from [pacman.game.entity.Entity](#)

- **Entity** ()  
*Entity default konstruktor.*
- **Direction** **getDirection** ()  
*Direction getter.*
- void **render** (Graphics2D graphics)  
*Az Entity megjelenítése a képernyőn.*
- Coordinate **getMapPosition** ()  
*Átkonvertálja a képernyő koordinátákat pálya koordinátákra.*
- Rectangle **getBounds** ()  
*Collision detection-höz használt függvény.*

## Protected Member Functions

- void **init** ()  
*Inicializáló metódus.*
- void **initSprites** ()  
*Sprite-okat inicializáló metódus.*
- void **updateSprite** ()  
*Sprite frissítése.*
- void **toStartingPos** ()  
*A kezdő pozíciójába helyezi Pacman-t.*

## Protected Member Functions inherited from [pacman.game.entity.Entity](#)

- Coordinate **getDrawPosition** ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell A Tile és az Entity sprite-ok méreteinek különbsége miatt külön ki kell számolni, hova kell rajzolni az Entity-ket.*
- void **checkWallCollisions** ()  
*Fallal való ütközést vizsgálja Ha ütközést talál, visszaállítja az Entityt az előző helyére.*
- void **checkOutOfFrame** ()  
*Azt vizsgálja, mikor megy ki az Entity a képernyőről Ha kimegy, a pálya másik oldalára rakja azt.*

## Private Member Functions

- **Direction** **getChosenDirection** ()  
*Visszaadja a felhasználó által éppen kiválasztott irányt.*
- void **interactWithTile** ()  
*Elfogyasztjuk az Edible-t, amin éppen vagyunk.*
- void **checkGhostCollisions** ()  
*Szellemekkel való ütközést kezelő metódus.*
- void **powerPelletEaten** ()  
*PowerPellet elfogyasztásakor meghívott függvény.*

## Private Attributes

- ArrayList< BufferedImage > **hurtAnimation**  
*Ez az animáció játszódik le, amikor Pacman-t megsebzik.*
- int **lives**  
*Pacman maradék életeinek száma.*
- boolean **isHurt**  
*Igaz, ha megsebeztek Pacman-t.*
- int **ghostsEaten**  
*Megevett szellemek száma egy PowerPellet elfogyasztása után. Azért kell, mert minél több szellemet eszik meg Pacman, annál több pontot kap.*
- int **perfectRun**  
*Azon alkalmak száma, amikor a játékos egy PowerPellet elfogyasztása után mind a 4 szellemet elkapta.*
- boolean **fruitEaten**  
*Igaz, ha Pacman megette az aktuálisan a pályán lévő gyümölcsöt.*
- **Direction** **nextDirection**  
*A következő lehetőségénél Pacman ebbe az irányba fordul. Akkor van értelme, amikor a játékos kiválaszt egy olyan irányt, amerre Pacman éppen nem tud menni (ez kereszteződések előtt fordul elő)*

## Additional Inherited Members

## Protected Attributes inherited from [pacman.game.entity.Entity](#)

- Coordinate **position**  
*Az Entity-t reprezentáló sprite középpontjának helye a képernyőn. Valójában nem teljesen a középpontját tároljuk, a tárolt koordinátát úgy kapjuk meg, hogy a sprite bal felső sarkának koordinátáiból kivonunk a  $(ENTITY\_SIZE - TILE\_SIZE) * SCALE$  egész számot.*
- **Direction** **direction**  
*Az Entity aktuális haladási iránya.*
- int **speed**  
*Az entitás aktuális sebessége (pixel / frame)*
- Map< **Direction**, ArrayList< BufferedImage > > **defaultSprites**  
*Az Entity sprite-jai irányonként rendezve. A megfelelő irányt megadva a map-ből megkapjuk azon sprite-ok sorozatát, melyeken index szerint végighaladva egy animációt kapunk.*
- ArrayList< BufferedImage > **spriteList**  
*A jelenlegi iránynak megfelelő sprite-ok listája. Ezen sprite-ok egymásutánja adja ki a karakter animációját.*
- int **spriteIndex** = 0  
*A jelenleg használt sprite indexe az iránynak megfelelő sprite-ok listájában.*
- BufferedImage **sprite**  
*Az aktuális sprite.*
- int **animationDrawCounter** = 0  
*A legutóbbi sprite váltás óta eltelt framek száma.*

### 3.26.1 Detailed Description

Pacmant reprezentáló osztály.



### 3.26.2 Member Function Documentation

#### 3.26.2.1 init()

```
void pacman.game.entity.pacman.Pacman.init ( ) [protected]
```

Inicializáló metódus.

Reimplemented from [pacman.game.entity.Entity](#).

#### 3.26.2.2 initSprites()

```
void pacman.game.entity.pacman.Pacman.initSprites ( ) [protected]
```

Sprite-okat inicializáló metódus.

Reimplemented from [pacman.game.entity.Entity](#).

#### 3.26.2.3 isFruitEaten()

```
boolean pacman.game.entity.pacman.Pacman.isFruitEaten ( )
```

##### Returns

True, ha Pacman megette az aktuális gyümölcsöt

#### 3.26.2.4 isHurt()

```
boolean pacman.game.entity.pacman.Pacman.isHurt ( )
```

##### Returns

True, ha Pacmant éppen megsebezték

#### 3.26.2.5 reset()

```
void pacman.game.entity.pacman.Pacman.reset ( )
```

Kezdőállapotba helyezi Pacman-t.

Reimplemented from [pacman.game.entity.Entity](#).

#### 3.26.2.6 toStartingPos()

```
void pacman.game.entity.pacman.Pacman.toStartingPos ( ) [protected]
```

A kezdő pozíciójába helyezi Pacman-t.

Reimplemented from [pacman.game.entity.Entity](#).

### 3.26.2.7 update()

```
void pacman.game.entity.pacman.Pacman.update ( )
```

Belső logika végrehajtása.

Reimplemented from [pacman.game.entity.Entity](#).

### 3.26.2.8 updateSprite()

```
void pacman.game.entity.pacman.Pacman.updateSprite ( ) [protected]
```

Sprite frissítése.

Reimplemented from [pacman.game.entity.Entity](#).

## 3.26.3 Member Data Documentation

### 3.26.3.1 perfectRun

```
int pacman.game.entity.pacman.Pacman.perfectRun [private]
```

Azon alkalmak száma, amikor a játékos egy PowerPellet elfogyasztása után mind a 4 szellemet elkapta.

Ha ez mind a 4 PowerPellet után megtörténik, a játékos 12000 extra pontot kap.

## 3.27 pacman.game.tile.edible.Pellet Class Reference

Ehető pontot reprezentáló osztály.

### Public Member Functions

- [Pellet](#) ([Coordinate mapPosition](#), [BufferedImage sprite](#))

*Pellet default konstruktor.*

### Public Member Functions inherited from [pacman.game.tile.edible.Edible](#)

- [Edible](#) ([Coordinate mapPosition](#), [BufferedImage sprite](#))  
*Edible default konstruktor.*
- int [getScoreModifier](#) ()  
*Visszaadja azt, hogy az edible elfogyasztása hány pontot ér.*
- boolean [isEaten](#) ()
- void [toEatenState](#) ()  
*EATEN állapotba állítja az Edible-t.*
- void [render](#) ([Graphics2D graphics](#))  
*A képernyőre rajzolja az Edible-t.*

## Public Member Functions inherited from [pacman.game.tile.Tile](#)

- [Tile](#) ([Coordinate](#) `mapPosition`, [BufferedImage](#) `sprite`)

*Tile default konstruktor.*

- boolean `isWalkable` ()

*Igaz, ha entitások átsétálhatnak ezen a tile-n.*

- [BufferedImage](#) `getSprite` ()

*Visszaadja a Tile sprite-ját.*

## Additional Inherited Members

## Protected Member Functions inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) `getDrawPosition` ()

*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell.*

## Protected Attributes inherited from [pacman.game.tile.edible.Edible](#)

- [EdibleState](#) `state`

*Az Edible aktuális állapota.*

- int `scoreModifier`

*Az Edible elfogyasztása ennyivel módosítja a pontszámot.*

## Protected Attributes inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) `mapPosition`

*A tile helye a pályán Nem képernyő koordinátákat tárol.*

- [BufferedImage](#) `sprite`

*A sprite ami a képernyőn ezt a Tile-t reprezentálja.*

### 3.27.1 Detailed Description

Ehető pontot reprezentáló osztály.

### 3.27.2 Constructor & Destructor Documentation

#### 3.27.2.1 Pellet()

```
pacman.game.tile.edible.Pellet.Pellet (
    Coordinate mapPosition,
    BufferedImage sprite )
```

Pellet default konstruktor.

## Parameters

<i>mapPosition</i>	a Pellet helye a pályán
<i>sprite</i>	a Pellet-et reprezentáló sprite

## 3.28 pacman.game.entity.ghost.Pinky Class Reference

Pinky (rózsaszín) szellemet reprezentáló osztály.

### Public Member Functions

- [Pinky](#) (Pacman [pacman](#))  
*Pinky default konstruktor.*

### Public Member Functions inherited from [pacman.game.entity.ghost.Ghost](#)

- [Ghost](#) (Pacman [pacman](#))  
*Ghost default konstruktor.*
- void [reset](#) ()  
*Kezdő állapotba helyezi a szellemeket.*
- void [update](#) ()  
*Minden belső logikát kezelő függvény.*
- void [actionPerformed](#) (ActionEvent e)  
*Frissíti a szellem állapotát A játék elején a szellemek a CHASE és SCATTER állapotok között váltanak, majd CHASE állapotban maradnak.*
- void [toFrightenedState](#) ()  
*FRIGHTENED állapotba váltja a szellemet.*
- void [interact](#) ()  
*Interakció Pacman és a szellem között Pacman hívja meg, amikor egy Tile-re kerülnek a szellemmel.*
- [GhostState](#) [getState](#) ()  
*State getter.*

### Public Member Functions inherited from [pacman.game.entity.Entity](#)

- [Entity](#) ()  
*Entity default konstruktor.*
- [Direction](#) [getDirection](#) ()  
*Direction getter.*
- void [render](#) (Graphics2D graphics)  
*Az Entity megjelenítése a képernyőn.*
- Coordinate [getMapPosition](#) ()  
*Átkonvertálja a képernyő koordinátákat pálya koordinátákra.*
- Rectangle [getBounds](#) ()  
*Collision detection-höz használt függvény.*

### Protected Member Functions

- void [toStartingPos](#) ()  
*Kezdőállapotba állítja a szellemet.*
- void [initSprites](#) ()  
*Sprite-okat inicializáló metódus.*
- Coordinate [getChaseTarget](#) ()
- Coordinate [getScatterTarget](#) ()

### Protected Member Functions inherited from [pacman.game.entity.ghost.Ghost](#)

- void [updateSprite](#) ()  
*Frissíti a szellem aktuális sprite-ját Több féle sprite-ja van, más logika szerint működik, mint az Entity metódusa.*
- void [init](#) ()  
*Inicializáló metódus.*
- void [updateDirection](#) ()  
*Frissíti a szellem irányát.*
- void [updateTargetTile](#) ()  
*Frissíti a szellem célját.*
- boolean [isInsideHouse](#) ()
- boolean [isInFrontOfHouse](#) ()

### Protected Member Functions inherited from [pacman.game.entity.Entity](#)

- Coordinate [getDrawPosition](#) ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell A Tile és az Entity sprite-ok méreteinek különbsége miatt külön ki kell számolni, hova kell rajzolni az Entity-ket.*
- void [checkWallCollisions](#) ()  
*Fallal való ütközést vizsgálja Ha ütközést talál, visszaállítja az Entityt az előző helyére.*
- void [checkOutOfFrame](#) ()  
*Azt vizsgálja, mikor megy ki az Entity a képernyőről Ha kimegy, a pálya másik oldalára rakja azt.*

### Additional Inherited Members

### Protected Attributes inherited from [pacman.game.entity.ghost.Ghost](#)

- Pacman [pacman](#)  
*Pacman referenciája.*
- List< BufferedImage > [frightenedSprites](#)  
*A FRIGHTENED állapotú szellemek sprite-jai.*
- Map< [Direction](#), BufferedImage > [eatenSprites](#)  
*Az EATEN állapotú szellemek sprite-jai.*

## Protected Attributes inherited from [pacman.game.entity.Entity](#)

- Coordinate **position**  
Az Entity-t reprezentáló sprite középpontjának helye a képernyőn Valójában nem teljesen a középpontját tároljuk, a tárolt koordinátát úgy kapjuk meg, hogy a sprite bal felső sarkának koordinátáiból kivonunk a  $(ENTITY\_SIZE - TILE\_SIZE) * SCALE$  egész számot.
- [Direction](#) **direction**  
Az Entity aktuális haladási iránya.
- int **speed**  
Az entitás aktuális sebessége (pixel / frame)
- Map< [Direction](#), ArrayList< BufferedImage > > **defaultSprites**  
Az Entity sprite-jai irányonként rendezve A megfelelő irányt megadva a map-ből megkapjuk azon sprite-ok sorozatát, melyeken index szerint végighaladva egy animációt kapunk.
- ArrayList< BufferedImage > **spriteList**  
A jelenlegi iránynak megfelelő sprite-ok listája Ezen spite-ok egymásutánja adja ki a karakter animációját.
- int **spriteIndex** = 0  
A jelenleg használt sprite indexe az iránynak megfelelő sprite-ok listájában.
- BufferedImage **sprite**  
Az aktuális sprite.
- int **animationDrawCounter** = 0  
A legutóbbi sprite váltás óta eltelt framek száma.

### 3.28.1 Detailed Description

Pinky (rózsaszín) szellemet reprezentáló osztály.

### 3.28.2 Constructor & Destructor Documentation

#### 3.28.2.1 [Pinky\(\)](#)

```
pacman.game.entity.ghost.Pinky.Pinky (
    Pacman pacman )
```

Pinky default konstruktor.

Parameters

<code>pacman</code>	Pacman referenciája
---------------------	---------------------

### 3.28.3 Member Function Documentation

#### 3.28.3.1 [getChaseTarget\(\)](#)

```
Coordinate pacman.game.entity.ghost.Pinky.getChaseTarget ( ) [protected]
```

Returns

A szellem célja CHASE állapotban

Reimplemented from [pacman.game.entity.ghost.Ghost](#).

### 3.28.3.2 getScatterTarget()

```
Coordinate pacman.game.entity.ghost.Pinky.getScatterTarget ( ) [protected]
```

#### Returns

A szellem célja SCATTER állapotban

Reimplemented from [pacman.game.entity.ghost.Ghost](#).

### 3.28.3.3 initSprites()

```
void pacman.game.entity.ghost.Pinky.initSprites ( ) [protected]
```

Sprite-okat inicializáló metódus.

Reimplemented from [pacman.game.entity.Entity](#).

### 3.28.3.4 toStartingPos()

```
void pacman.game.entity.ghost.Pinky.toStartingPos ( ) [protected]
```

Kezdőállapotba állítja a szellemet.

Reimplemented from [pacman.game.entity.Entity](#).

## 3.29 pacman.game.tile.edible.PowerPellet Class Reference

Ehető pontot reprezentáló osztály Ezt elfogyasztva a szellemek Frightened állapotba kerülnek.

#### Public Member Functions

- [PowerPellet](#) ([Coordinate mapPosition](#), [BufferedImage sprite](#))

*PowerPellet default konstruktor.*

#### Public Member Functions inherited from [pacman.game.tile.edible.Edible](#)

- [Edible](#) ([Coordinate mapPosition](#), [BufferedImage sprite](#))  
*Edible default konstruktor.*
- **int** [getScoreModifier](#) ()  
*Visszaadja azt, hogy az edible elfogyasztása hány pontot ér.*
- **boolean** [isEaten](#) ()
- **void** [toEatenState](#) ()  
*EATEN állapotba állítja az Edible-t.*
- **void** [render](#) ([Graphics2D graphics](#))  
*A képernyőre rajzolja az Edible-t.*

## Public Member Functions inherited from [pacman.game.tile.Tile](#)

- [Tile](#) ([Coordinate](#) `mapPosition`, [BufferedImage](#) `sprite`)

*Tile default konstruktor.*

- boolean `isWalkable` ()

*Igaz, ha entitások átsétálhatnak ezen a tile-n.*

- [BufferedImage](#) `getSprite` ()

*Visszaadja a Tile sprite-ját.*

## Additional Inherited Members

## Protected Member Functions inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) `getDrawPosition` ()

*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell.*

## Protected Attributes inherited from [pacman.game.tile.edible.Edible](#)

- [EdibleState](#) `state`

*Az Edible aktuális állapota.*

- int `scoreModifier`

*Az Edible elfogyasztása ennyivel módosítja a pontszámot.*

## Protected Attributes inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) `mapPosition`

*A tile helye a pályán Nem képernyő koordinátákat tárol.*

- [BufferedImage](#) `sprite`

*A sprite ami a képernyőn ezt a Tile-t reprezentálja.*

### 3.29.1 Detailed Description

Ehető pontot reprezentáló osztály Ezt elfogyasztva a szellemek Frightened állapotba kerülnek.

### 3.29.2 Constructor & Destructor Documentation

#### 3.29.2.1 PowerPellet()

```
pacman.game.tile.edible.PowerPellet.PowerPellet (
    Coordinate mapPosition,
    BufferedImage sprite )
```

PowerPellet default konstruktor.



## Parameters

<i>mapPosition</i>	a PowerPellet helye a pályán
<i>sprite</i>	a PowerPellet-et reprezentáló sprite

### 3.30 pacman.game.util.SpriteLoader Class Reference

Grafikák betöltéséért felelős osztály.

#### Static Public Member Functions

- static void **loadSprites** ()

*Grafikák betöltése.*

#### Static Public Attributes

- static ImageIcon **icon**
- static BufferedImage **guide**
- static Map< String, BufferedImage > **tileSprites** = new HashMap<>()
- static Map< Direction, ArrayList< BufferedImage > > **pacmanSprites** = new HashMap<>()
- static ArrayList< BufferedImage > **pacmanHurtAnimation** = new ArrayList<>()
- static BufferedImage **pacmanMainMenu**
- static Map< Direction, ArrayList< BufferedImage > > **blinkySprites** = new HashMap<>()
- static Map< Direction, ArrayList< BufferedImage > > **inkySprites** = new HashMap<>()
- static Map< Direction, ArrayList< BufferedImage > > **pinkySprites** = new HashMap<>()
- static Map< Direction, ArrayList< BufferedImage > > **clydeSprites** = new HashMap<>()
- static ArrayList< BufferedImage > **ghostFrightenedSprites** = new ArrayList<>()
- static Map< Direction, BufferedImage > **ghostEatenSprites** = new HashMap<>()

#### Static Private Member Functions

- static void **error** ()
- static void **loadMenuImages** ()
- static void **loadTileSprites** ()
- static void **loadWallSprites** ()
- static void **loadEdibleSprites** ()
- static void **loadEntitySprites** ()
- static void **loadPacmanSprites** ()
- static void **loadGhostSprites** ()
- static void **loadBlinkySprites** ()
- static void **loadPinkySprites** ()
- static void **loadInkySprites** ()
- static void **loadClydeSprites** ()
- static void **loadFrightenedSprites** ()
- static void **loadEatenSprites** ()

### 3.30.1 Detailed Description

Grafikák betöltéséért felelős osztály.

## 3.31 pacman.game.tile.Tile Class Reference

Egy, a pályán lévő cellát reprezentáló osztály.

### Public Member Functions

- [Tile](#) ([Coordinate](#) [mapPosition](#), [BufferedImage](#) [sprite](#))  
*Tile default konstruktor.*
- boolean [isWalkable](#) ()  
*Igaz, ha entitások átsétálhatnak ezen a tile-n.*
- void [render](#) ([Graphics2D](#) [graphics](#))  
*A Tile megjelenítése a képernyőn.*
- [BufferedImage](#) [getSprite](#) ()  
*Visszaadja a Tile sprite-ját.*

### Protected Member Functions

- [Coordinate](#) [getDrawPosition](#) ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell.*

### Protected Attributes

- [Coordinate](#) [mapPosition](#)  
*A tile helye a pályán Nem képernyő koordinátákat tárol.*
- [BufferedImage](#) [sprite](#)  
*A sprite ami a képernyőn ezt a Tile-t reprezentálja.*

### 3.31.1 Detailed Description

Egy, a pályán lévő cellát reprezentáló osztály.

### 3.31.2 Constructor & Destructor Documentation

#### 3.31.2.1 Tile()

```
pacman.game.tile.Tile.Tile (  
    Coordinate mapPosition,  
    BufferedImage sprite )
```

Tile default konstruktor.

## Parameters

<i>mapPosition</i>	A Tile helye a pályán
<i>sprite</i>	A Tile-t megjelenítő sprite

### 3.31.3 Member Function Documentation

#### 3.31.3.1 getDrawPosition()

`Coordinate` `pacman.game.tile.Tile.getDrawPosition ( )` [protected]

Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell.

Reimplemented in [pacman.game.tile.edible.Fruit](#).

#### 3.31.3.2 isWalkable()

`boolean` `pacman.game.tile.Tile.isWalkable ( )`

Igaz, ha entitások átsétálhatnak ezen a tile-n.

Reimplemented in [pacman.game.tile.wall.Wall](#).

#### 3.31.3.3 render()

`void` `pacman.game.tile.Tile.render (`  
     `Graphics2D graphics )`

A Tile megjelenítése a képernyőn.

## Parameters

<i>graphics</i>	Erre történik a festés
-----------------	------------------------

Reimplemented in [pacman.game.tile.edible.Edible](#), and [pacman.game.tile.edible.Fruit](#).

## 3.32 pacman.game.display.UpperRibbon Class Reference

A felső szalagot megjelenítő osztály.

### Public Member Functions

- **UpperRibbon ( )**

*UpperRibbon default konstruktor.*

### Protected Member Functions

- void **paintComponent** (Graphics g)  
*A felső szalagot kirajzoló metódus.*

### Package Attributes

- int **highScore**  
*Maximális pontszám.*

## 3.32.1 Detailed Description

A felső szalagot megjelenítő osztály.

## 3.33 pacman.game.tile.wall.Wall Class Reference

Falat reprezentáló osztály.

### Public Member Functions

- **Wall** ([Coordinate mapPosition](#), [BufferedImage sprite](#))  
*Wall default konstruktor.*
- boolean [isWalkable](#) ()

### Public Member Functions inherited from [pacman.game.tile.Tile](#)

- **Tile** ([Coordinate mapPosition](#), [BufferedImage sprite](#))  
*Tile default konstruktor.*
- void [render](#) ([Graphics2D graphics](#))  
*A Tile megjelenítése a képernyőn.*
- [BufferedImage](#) **getSprite** ()  
*Visszaadja a Tile sprite-ját.*

### Additional Inherited Members

### Protected Member Functions inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) [getDrawPosition](#) ()  
*Visszaadja azt a képernyő koordinátát, ahonnan a sprite rajzolását kezdeni kell.*

### Protected Attributes inherited from [pacman.game.tile.Tile](#)

- [Coordinate](#) **mapPosition**  
*A tile helye a pályán Nem képernyő koordinátákat tárol.*
- [BufferedImage](#) **sprite**  
*A sprite ami a képernyőn ezt a Tile-t reprezentálja.*

### 3.33.1 Detailed Description

Falat reprezentáló osztály.

### 3.33.2 Member Function Documentation

#### 3.33.2.1 isWalkable()

```
boolean pacman.game.tile.wall.Wall.isWalkable ( )
```

##### Returns

False, mivel a falon nem lehet átmenni

Reimplemented from [pacman.game.tile.Tile](#).