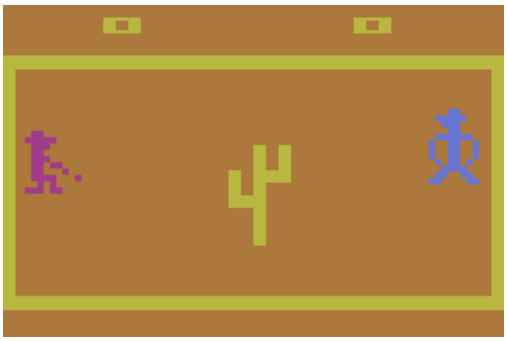

Outlaw

Tartalomjegyzék

Felhasználói dokumentáció

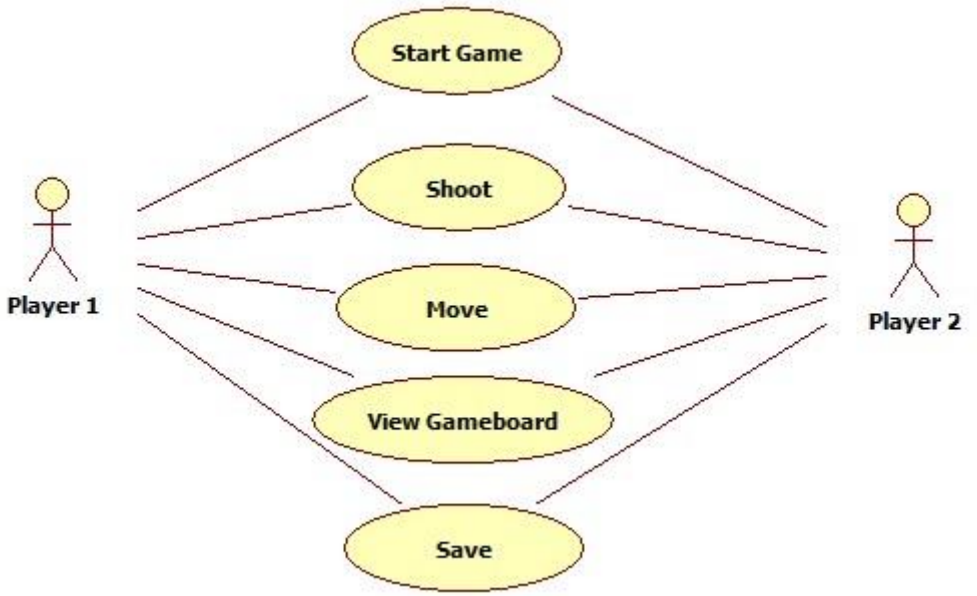
A JÁTÉK ISMERTETÉSE



Minta az eredeti játékból

Az Outlaw egy kétjátékos shooter játék. A két felhasználó rendelkezik egy-egy karakterrel (cowboy-al), amelyek egy kettéosztott pálya egy-egy részén foglalnak helyet. A karakterek tudnak minden irányba mozogni a saját térfelükön, és tudnak lőni is. A pálya közepén egy objektum van elhelyezve, amely fedezékül szolgálhat a játékosoknak. A cél az, hogy a játékos eltalálják az ellenfél karakterét. Minden találat a játékos egy életébe kerül. A játékosoknak a kör kezdetén 10 életük van. A felhasználók egyenesen és ± 45 fokos szögben tudnak lőni. A lövedékek a pálya alsó és felső szegélyén megpattanhat. A középen lévő objektumról, illetve a bal és jobb oldali falakról a lövedék nem pattan vissza. Az a játékos veszít, aki elveszti az összes életét.

USE-CASE DIAGRAM:



USE-CASE LEÍRÁSOK:

Cím	**Start Game**
Leírás	A játék elindítása.
Aktorok	Player 1, Player 2
Fő forgatókönyv	1. Az egyik játékos kiválasztja a „Play” menüpontot. 2. A pálya, a karakterek és egyéb objektumok létrejönnek. 3. A játék elindul.

Cím	**Move**
Leírás	A játékos a karaktert mozgatja.
Aktorok	Player 1, Player2
Fő forgatókönyv	1. A játékos a karaktert balra, jobbra, fel és le mozgathatja.
Alternatív forgatókönyv	1.A.1 A játékos csak a saját térfelén belül képes mozogni.
Alternatív forgatókönyv	1.B.1 A játékos meghal, ha egy golyó eltalálja.
Alternatív forgatókönyv	1.B.2 Ha a játékos már 10-szer eltalálták, akkor vesztett és a játéknak vége.

Cím	**Shoot**
Leírás	A játékos lő a karakterrel.
Aktorok	Player 1, Player 2
Fő forgatókönyv	1. A játékos egyenesen és ± 45 fokban tud lőni.
Alternatív forgatókönyv	1.A.1 Ha a játékos eltalálja a pálya alsó vagy felső részét a golyó visszapattan.
Alternatív forgatókönyv	1.B.1 Ha a játékos eltalálja a középen lévő objektumot, vagy a bal és jobb oldali falat, akkor a golyó nem pattan vissza.
Alternatív forgatókönyv	1.C.1 Ha a játékos eltalálja az ellenfelét, akkor az ellenfél veszít egy életet.
Alternatív forgatókönyv	1.C.2 Ha az ellenfél elvesztette az összes életét a játéknak vége.

Cím	**View Gameboard**
Leírás	A játékos megtekinti a pályát.
Aktorok	Player 1, Player 2
Fő forgatókönyv	1. A rendszer kirajzolja a pálya aktuális állapotát. 2. A játékos megtekinti a pálya aktuális állapotát.

Cím	**Save**
Leírás	A játék aktuális állapotának elmentése.
Aktorok	Player 1, Player 2
Fő forgatókönyv	1. A játékos rákattint a „Save” gombra. 2. A játék aktuális állapota elmentére kerül.

Cím	**Load**
Leírás	Elmentett játék betöltése.
Aktorok	Player 1, Player 2
Fő forgatókönyv	1. A játékos rákattint a „Load” menüpontra. 2. Az előzőleg elmentett játék betöltése.
Alternatív forgatókönyv	1.A.1 Ha nem volt előzőleg elmentett játék akkor egy hibaüzenet jelzi ezt a felhasználónak.

A JÁTÉKOSOK IRÁNYÍTÁSA

Player 1:

- Mozgás: fel – E, le – D, jobbra – F, balra – S
- Lövés egyenesen: Q
- Lövés felfele: Q + E
- Lövés lefele: Q + D

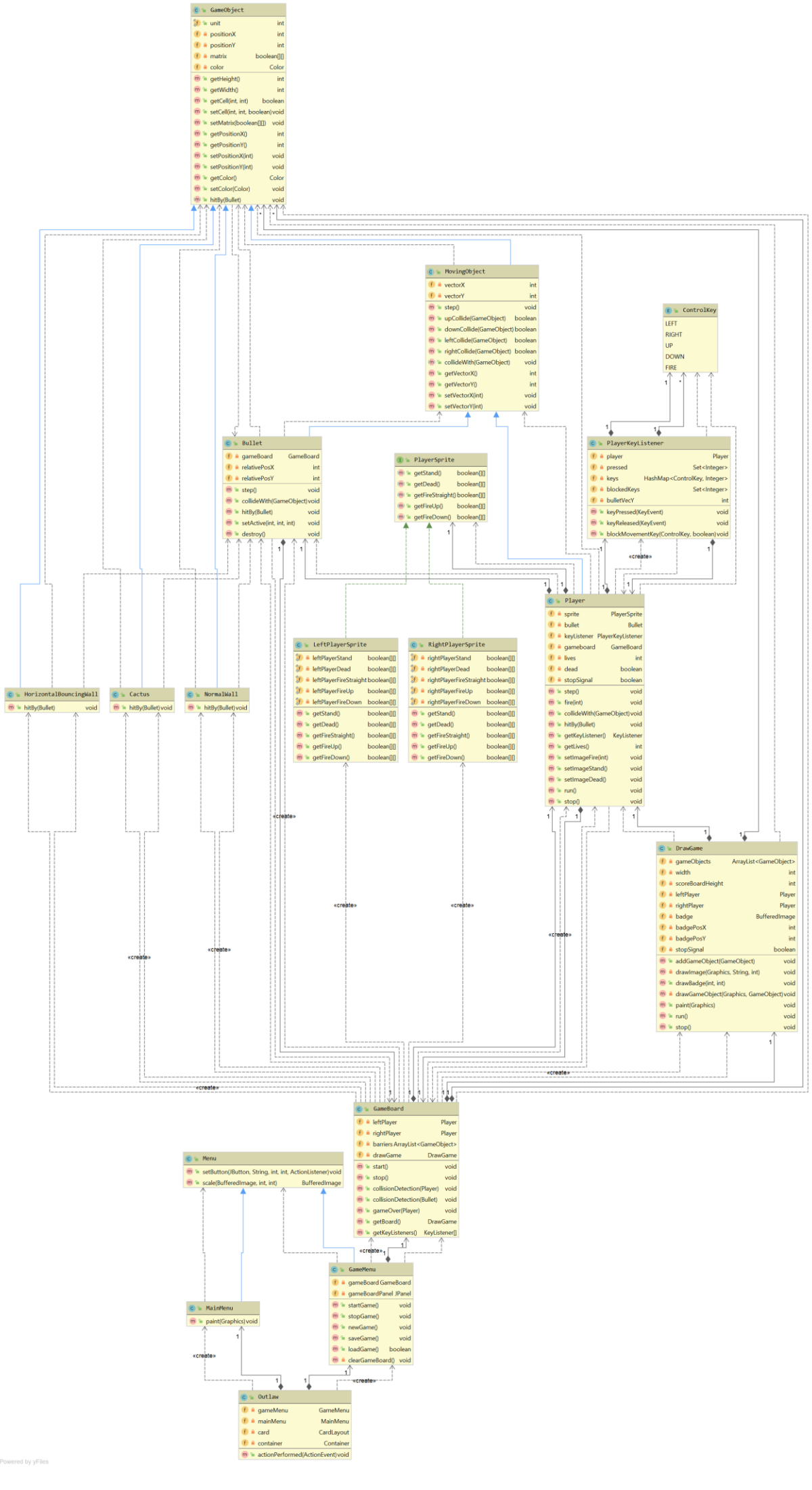
Player 2:

- Mozgás: fel - ▲, le -▼, jobbra - ►, balra - ◄
- Lövés: M
- Lövés felfele: M + ▲
- Lövés lefele: M + ▼

Fejlesztői dokumentáció

Osztálydiagram

A következő oldalon.


GameObject
unit int
positionX int
positionY int
matrix boolean[][]
color Color
getHeight() int
getWidth() int
getCell(int, int) boolean
setCell(int, int, boolean) void
setMatrix(boolean[][]) void
getPositionX() int
getPositionY() int
setPositionX(int) void
setPositionY(int) void
getColor() Color
setColor(Color) void
hitBy(Bullet) void
MovingObject
vectorX int
vectorY int
step() void
upCollide(GameObject) boolean
downCollide(GameObject) boolean
leftCollide(GameObject) boolean
rightCollide(GameObject) boolean
collideWith(GameObject) void
getVectorX() int
getVectorY() int
setVectorX(int) void
setVectorY(int) void
ControlKey
LEFT
RIGHT
UP
DOWN
FIRE
Bullet
gameBoard GameBoard
relativePosX int
relativePosY int
step() void
collideWith(GameObject) void
hitBy(Bullet) void
setActive(int, int, int) void
destroy() void
PlayerSprite
getStand() boolean[][]
getDead() boolean[][]
getFireStraight() boolean[][]
getFireUp() boolean[][]
getFireDown() boolean[][]
PlayerKeyListener
player Player
pressed Set<Integer>
keys HashMap<ControlKey, Integer>
blockedKeys Set<Integer>
bulletVecY int
keyPressed(KeyEvent) void
keyReleased(KeyEvent) void
blockMovementKey(ControlKey, boolean) void
«create»
Player
sprite PlayerSprite
bullet Bullet
keyListener PlayerKeyListener
gameboard GameBoard
lives int
dead boolean
stopSignal boolean
step() void
fire(int) void
collideWith(GameObject) void
hitBy(Bullet) void
getKeyListener() KeyListener
getLives() int
setImageFire(int) void
setImageStand() void
setImageDead() void
run() void
stop() void
LeftPlayerSprite
leftPlayerStand boolean[][]
leftPlayerDead boolean[][]
leftPlayerFireStraight boolean[][]
leftPlayerFireUp boolean[][]
leftPlayerFireDown boolean[][]
getStand() boolean[][]
getDead() boolean[][]
getFireStraight() boolean[][]
getFireUp() boolean[][]
getFireDown() boolean[][]
RightPlayerSprite
rightPlayerStand boolean[][]
rightPlayerDead boolean[][]
rightPlayerFireStraight boolean[][]
rightPlayerFireUp boolean[][]
rightPlayerFireDown boolean[][]
getStand() boolean[][]
getDead() boolean[][]
getFireStraight() boolean[][]
getFireUp() boolean[][]
getFireDown() boolean[][]
HorizontalBouncingWall
hitBy(Bullet) void
Cactus
hitBy(Bullet) void
NormalWall
hitBy(Bullet) void
DrawGame
gameObjects ArrayList<GameObject>
width int
scoreBoardHeight int
leftPlayer Player
rightPlayer Player
badge BufferedImage
badgePosX int
badgePosY int
stopSignal boolean
addGameObject(GameObject) void
drawImage(Graphics, String, int) void
drawBadge(int, int) void
drawGameObject(Graphics, GameObject) void
paint(Graphics) void
run() void
stop() void
GameBoard
leftPlayer Player
rightPlayer Player
barriers ArrayList<GameObject>
drawGame DrawGame
start() void
stop() void
collisionDetection(Player) void
collisionDetection(Bullet) void
gameOver(Player) void
getBoard() DrawGame
getKeyListeners() KeyListener[]
Menu
setButton(JButton, String, int, int, ActionListener) void
scale(BufferedImage, int, int) BufferedImage
GameMenu
gameBoard GameBoard
gameBoardPanel JPanel
startGame() void
stopGame() void
newGame() void
saveGame() void
loadGame() boolean
clearGameBoard() void
MainMenu
paint(Graphics) void
Outlaw
gameMenu GameMenu
mainMenu MainMenu
card CardLayout
container Container
actionPerformed(ActionEvent) void
Powered by yFiles

Osztályok dokumentációja

ControlKey enumeráció

Publikus attribútumok

- **LEFT**
- **RIGHT**
- **UP**
- **DOWN**
- **FIRE**

Részletes leírás

ControlKey enum: A játékoshoz tartozó tevékenységek.

Adattagok dokumentációja

DOWN

A játékos lefele megy.

FIRE

A játékos lő.

LEFT

A játékos balra megy.

RIGHT

A játékos jobbra megy.

UP

A játékos felfele megy.

A dokumentáció ehhez az enum-hoz a következő fájl alapján készült:

- outlaw/enumeration/**ControlKey.java**

GameObject

Ősök: Serializable.

Leszármazottak: **Cactus**, **HorizontalBouncingWall**, **MovingObject** és **NormalWall**.

Publikus tagfüggvények

- **GameObject** (int posX, int posY, int width, int heigth)
- **GameObject** (int posX, int posY, boolean[][] **matrix**)
- int **getHeight** ()
- int **getWidth** ()
- boolean **getCell** (int y, int x)
- void **setCell** (int y, int x, boolean value)
- void **setMatrix** (boolean[][] **matrix**)
- int **getPositionX** ()
- int **getPositionY** ()
- void **setPositionX** (int x)
- void **setPositionY** (int y)
- Color **getColor** ()
- void **setColor** (Color **color**)
- abstract void **hitBy** (**Bullet** bullet)

Statikus publikus attribútumok

- static final int **unit** = 10

Privát attribútumok

- int **positionX**
- int **positionY**
- boolean[][] **matrix**
- Color **color** = Color.black

Részletes leírás

Gamobject abstract osztály: A játéktáblán elhelyezhető objetumok ősosztálya. Az objektumok fájlba írása miatt megvalósítja a Serializable interfészt.

Konstruktorok és destruktorok dokumentációja

GameObject (int *posX*, int *posY*, int *width*, int *heigth*)

1. Konstruktor: A megadott méretű és pozíciójú objektum létrehozása.

Paraméterek

posX	- int, az objektum x pozíciója pixelben értendő
posY	- int, az objektum y pozíciója pixelben értendő
width	- int, az objektum szélessége unit-ban értendő
heigth	- int, az objektum magassága unit-ban értendő

GameObject (int *posX*, int *posY*, boolean *matrix*[][])

2. Konstruktor: A megadott mátrixnak megfelelő kinézetű és pozíciójú objektum létrehozása.

Paraméterek

posX	- int
posY	- int
matrix	- boolean[][]

Tagfüggvények dokumentációja

boolean getCell (int *y*, int *x*)

Cella értékének lekérdezése: Megadja a mátrix y. sorának az x. oszlopában lévő cella értékét.

Paraméterek

y	- int
x	- int

Visszatérési érték

boolean

Color getColor ()

Az objektum színének lekérdezése.

Visszatérési érték

Color

int getHeight ()

A mátrix magasságának lekérdezése.

Visszatérési érték

int

int getPositionX ()

Az objektum X pozíciójának lekérdezése.

Visszatérési érték

int

int getPositionY ()

Az objektum Y pozíciójának lekérdezése.

Visszatérési érték

int

int getWidth ()

A mátrix szélességének lekérdezése.

Visszatérési érték

int

abstract void hitBy (Bullet *bullet*)[abstract]

A lövedékkel való ütközés implemetálása a nem abstract leszármazotakban.

Paraméterek

bullet	**- Bullet**

Újraimplementáló leszármazottak: **Player** ,**Bullet**, **Cactus**, **NormalWall** és **HorizontalBouncingWall**.

void setCell (int *y*, int *x*, boolean *value*)

Cella módosítása: A mátrix y. sorának az x. oszlopában lévő cella értékét beállítja a value értékére.

Paraméterek

y	- int
x	- int
value	- boolean

void setColor (Color *color*)

Az objektum színének módosítása.

Paraméterek

color	- Color

void setMatrix (boolean *matrix*[][])

Mátrix módosítása: A mátrix értékének felülírása egy másik mátrix-al.

Paraméterek

matrix	- boolean[][]

void setPositionX (int *x*)

Az objektum X pozíciójának módosítása.

Paraméterek

x	- int

void setPositionY (int *y*)

Az objektum Y pozíciójának módosítása.

Paraméterek

y	- int

Adattagok dokumentációja

Color color = Color.black[private]

Az objektum színe (alapból fekete).

boolean [][] matrix[private]

Az objektum 2d-s pixeles kinézete.

int positionX[private]

Az objektum X pozíciója.

int positionY[private]

Az objektum Y pozíciója.

final int unit = 10[static]

Az objektum egy egységének mérete pixelben.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/gameobject/**GameObject.java**

MovingObject

ősök: **GameObject**.

Leszármazottak: **Bullet** és **Player**.

Publikus tagfüggvények

- **MovingObject** (int posX, int posY, int width, int heigth, int vecX, int vecY)
- **MovingObject** (int posX, int posY, boolean[][] **matrix**, int vecX, int vecY)
- void **step** ()
- boolean **upCollide** (**GameObject** gameObject)
- boolean **downCollide** (**GameObject** gameObject)
- boolean **leftCollide** (**GameObject** gameObject)
- boolean **rightCollide** (**GameObject** gameObject)
- abstract void **collideWith** (**GameObject** gameObject)
- int **getVectorX** ()
- int **getVectorY** ()
- void **setVectorX** (int **vectorX**)
- void **setVectorY** (int **vectorY**)

Privát attribútumok

- int **vectorX**
- int **vectorY**

Részletes leírás

MovingObject abstract osztály: A **GameObject** osztályból leszármazó, mozgó objektumokat reprezentáló osztály.

Konstruktorok és destruktorok dokumentációja

MovingObject (int *posX*, int *posY*, int *width*, int *heigth*, int *vecX*, int *vecY*)

1. Konstruktor: A megadott méretű, pozíciójú, és irányú objektum létrehozása.

Paraméterek

posX	- int
posY	- int
width	- int
heigth	- int
vecX	- int
vecY	- int

MovingObject (int *posX*, int *posY*, boolean *matrix*[][], int *vecX*, int *vecY*)

2. Konstruktor: A megadott mátrixnak megfelelő kinézetű, pozíciójú és irányú objektum létrehozása.

Paraméterek

posX	- int
posY	- int
matrix	- boolean[][]
vecX	- int
vecY	- int

Tagfüggvények dokumentációja

abstract void collideWith (GameObject *gameObject*)[abstract]

A paraméterként megadott objektummal való ütközés végrehajtandó művelet.

Paraméterek

gameObject	- **GameObject**

Újraimplementáló leszármazottak: **Player** és **Bullet**.

boolean downCollide (GameObject *gameObject*)

Ütközés alulról: Ellenőrzi, hogy az objektum alulról ütközött-e egy másik objektummal. Ha igen, akkor a visszatérési érték true, egyébként false.

Paraméterek

gameObject	- **GameObject**

Visszatérési érték

boolean

int getVectorX ()

Az X irányú vektor lekérdezése.

Visszatérési érték

int

int getVectorY ()

Az Y irányú vektor lekérdezése.

Visszatérési érték

int

boolean leftCollide (GameObject *gameObject*)

Ütközés balodalról: Ellenőrzi, hogy az objektum balodalról ütközött-e egy másik objektummal. Ha igen, akkor a visszatérési érték true, egyébként false.

Paraméterek

gameObject	- Gameobject

Visszatérési érték

boolean

boolean rightCollide (GameObject *gameObject*)

Ütközés jobbodalról: Ellenőrzi, hogy az objektum jobbodalról ütközött-e egy másik objektummal. Ha igen, akkor a visszatérési érték true, egyébként false.

Paraméterek

gameObject	- Gameobject

Visszatérési érték

boolean

void setVectorX (int *vectorX*)

Az X irányú vektor módosítása.

void setVectorY (int *vectorY*)

Az Y irányú vektor módosítása.

void step ()

A objketum a mozgatása a vektoroknak megfelelően.

Újraimplementáló leszármazottak: **Player** és **Bullet**.

boolean upCollide (GameObject *gameObject*)

Ütközés felülről: Ellenőrzi, hogy az objektum felülről ütközött-e egy másik objektummal. Ha igen, akkor a visszatérési érték true, egyébként false.

Paraméterek

gameObject	- Gameobject

Visszatérési érték

boolean

Adattagok dokumentációja

int vectorX[private]

Az objektum X irányú vektora.

int vectorY[private]

Az objektum Y irényú vektora.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/gameobject/**MovingObject.java**

NormalWall

Ősök: **GameObject**.

Publikus tagfüggvények

- **NormalWall** (int posX, int posY, int width, int heigth)
- void **hitBy** (**Bullet** bullet)

Részletes leírás

NormalWall osztály: Egy olyan **GameObject**, ami egy sima falat reprezentál a játéktéren.

Konstruktorok és destruktorok dokumentációja

NormalWall (int *posX*, int *posY*, int *width*, int *heigth*)

Konstruktor: A megadott szélességű, vastagságú és pozíziójú fal létrehozása. A falhoz tartozó mártix minden cellájának true értékre való állítása.

Paraméterek

posX	- int
posY	- int
width	- int
heigth	- int

Tagfüggvények dokumentációja

void hitBy (Bullet *bullet*)

Lövedékkel való ütközéskor a lövedék megsemmisítése.

Paraméterek

bullet	- **Bullet**

Újraimplementált ősök: **GameObject**.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/gameobject/**NormalWall.java**

HorizontalBouncingWall

Ősök: **GameObject**.

Publikus tagfüggvények

- **HorizontalBouncingWall** (int posX, int posY, int width)
- void **hitBy** (**Bullet** bullet)

Részletes leírás

HorizontalBouncingWall osztály: Egy olyan **GameObject**, ami egy horizontális falat reprezentál, amelyről a lövedékek visszapattannak.

Konstruktorok és destruktorok dokumentációja

HorizontalBouncingWall (int *posX*, int *posY*, int *width*)

Konstruktor: A megadott pozíciójú és szélességű fal létrehozása.

Paraméterek

posX	- int
posY	- int
width	- int

Tagfüggvények dokumentációja

void hitBy (Bullet *bullet*)

A lövedékkel való ütközéskor a lövedék Y irányú vektorának negálása.

Paraméterek

bullet	- **Bullet**

Újraimplementált ősök: **GameObject**.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/gameobject/HorizontalBouncingWall.java

Cactus

Ősök: **GameObject**.

Publikus tagfüggvények

- **Cactus** (int posX, int posY, int heigth)
- void **hitBy** (**Bullet** bullet)

Részletes leírás

Cactus osztály: A játékban a kaktusz reprezentáló objektum.

Konstruktorok és destruktorok dokumentációja

Cactus (int *posX*, int *posY*, int *heigth*)

Konstruktor: A megadott magasságú és pozíziójú kaktusz létrehozása.

Paraméterek

posX	- int
posY	- int
heigth	- int

Tagfüggvények dokumentációja

void hitBy (Bullet *bullet*)

Lövedékkel való ütközéskor a lövedék megsemmisítése.

Paraméterek

bullet	- **Bullet**

Újraimplementált ősök: **GameObject** (*o.10*).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/gameobject/**Cactus.java**

Player

Ősök: **MovingObject** és Runnable.

Publikus tagfüggvények

- **Player** (int startPosX, int startPosY, Integer moveUpKey, Integer moveDownKey, Integer moveLeftKey, Integer moveRightKey, Integer fireKey, **PlayerSprite sprite**, **Bullet bullet**, **GameBoard gameboard**)
- void **step** ()
- void **fire** (int vecY)
- void **collideWith** (**GameObject** gameObject)
- void **hitBy** (**Bullet bullet**)
- KeyListener **getKeyListener** ()
- int **getLives** ()
- boolean **isAlive**()
- void **setImageFire** (int direction)
- void **setImageStand** ()
- void **setImageDead** ()
- void **run** ()
- void **stop** ()

Privát attribútumok

- **PlayerSprite sprite**
- **Bullet bullet**
- **PlayerKeyListener keyListener**
- **GameBoard gameboard**
- int **lives** = 10
- boolean **dead** = false
- volatile boolean **stopSignal** = false

Részletes leírás

Player osztály: A játéktáblán a játékost reprezentáló mozgó objektum. A Runnable interfész implemetálásával lehetővé teszi a játékos külön szálon való futtatását.

Konstruktorok és destruktorok dokumentációja

Player (int *startPosX*, int *startPosY*, Integer *moveUpKey*, Integer *moveDownKey*, Integer *moveLeftKey*, Integer *moveRightKey*, Integer *fireKey*, PlayerSprite *sprite*, Bullet *bullet*, GameBoard *gameboard*)

A játékos inicializálása: a kezdő pozíció beállítása, a karakter irányításához szükséges billentyűk megadása, a játékos kinézeteinek megadása, a lövedék beállítása, és a megadott játéktáblához való hozzárendelés.

Paraméterek

startPosX	- int
startPosY	- int
moveUpKey	- Integer
moveDownKey	- Integer
moveLeftKey	- Integer
moveRightKey	- Integer
fireKey	- Integer
sprite	- PlayerSprite
bullet	- **Bullet**
gameboard	- GameBoard

Tagfüggvények dokumentációja

void collideWith (GameObject *gameObject*)

A paraméterként megadott objektummal való ütközés ellenőrzése. Az ütközés irányának megfelelően a karakter mozgásának blokkolása.

Paraméterek

gameObject	- **GameObject**

Újraimplementált ősök: **MovingObject**.

void fire (int *vecY*)

A játékos a paraméterként megadott Y irányba kilövi a hozzá tartozó lövedéket.

Paraméterek

vecY	- int

KeyListener getKeyListener ()

A játékoshoz tartozó keylistener lekérdezése.

Visszatérési érték

Keylistener

int getLives ()

A játékos megmaradt életeinek a számát adja vissza.

Visszatérési érték

int

boolean isAlive()

Megmondja, hogy a játékos életben van-e.

Visszatérési érték

boolean

void hitBy (Bullet *bullet*)

Az ellenfél lövedékével való ütközéskor az ellenfél tölténye megsemmisül, és az eltalált játékos egy életet veszít. Ha a játékos elvesztette az összes életét, akkor meghívja a gameboard gameover metódusát. Emellett beállítja a játékosnak a lelőtt kinézetet 1 mp-re. Erre az időre a játékot is megállítja.

Paraméterek

bullet	- **Bullet**

Újraimplementált ősök: **GameObject**.

void run ()

A Runnable interfészhez tartozó metódus. A játékos külön szálon való futtatását szolgálja.

void setImageDead ()

A lelőtt játékos kinézet beállítása.

void setImageFire (int *direction*)

A lövő játékos kinézet beállítása. A paramétertől függően ez lehet egyenesen, le vagy felfele lövő játékos kinézet.

Paraméterek

direction	- int

void setImageStand ()

Az álló játékos kinézet beállítása.

void step ()

A léptetés előtt az ütközés ellenőrzése, majd a játékoshoz tartozó lövedék, és ha a játékos nem halott, akkor a játékos léptetése.

Újraimplementált ősök: **MovingObject**.

void stop ()

A játékoshoz tartozó futó szál befejezése.

Adattagok dokumentációja

Bullet bullet[private]

A játékoshoz tartozó lövedék.

boolean dead = false[private]

Azt tárolja, hogy a játékos éppen lelőtt állapotba van-e.

GameBoard gameboard[private]

A karakter irányításához szükséges keylistener. A játéktábla, amihez a játékos tartozik.

PlayerKeyListener keyListener[private]

int lives = 10[private]

A játékos életeinek a száma.

PlayerSprite sprite[private]

A játékos kinézeteit tartalmazza.

volatile boolean stopSignal = false[private]

A játékoshoz tartozó futó szál befejezésének jelzése.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/gameobject/**Player.java**

Bullet

Ősök: **MovingObject**.

Publikus tagfüggvények

- **Bullet** (int **relativePosX**, int **relativePosY**, int vecX, **GameBoard gameBoard**)
- void **step** ()
- void **collideWith** (**GameObject** gameObject)
- void **hitBy** (**Bullet** bullet)
- void **setActive** (int posX, int posY, int vecY)
- void **destroy** ()

Privát attribútumok

- **GameBoard gameBoard**
- int **relativePosX**
- int **relativePosY**

Részletes leírás

Bullet osztály: A lövedéket reprezentáló mozgó objektum.

Konstruktorok és destruktorok dokumentációja

Bullet (int *relativePosX*, int *relativePosY*, int *vecX*, GameBoard *gameBoard*)

Kontruktor: A lövedék relative pozíciójának, X irányú vektorának, és a hozzá tartozó játéktaáblájának beállítása. Az X pozíció innentől fix, csak egy irányba lehet lőni. A lövedék egy 1x1-es mátrix, vagyis egy "pixel".

Paraméterek

relativePosX	- int
relativePosY	- int
vecX	- int
gameBoard	- GameBoard

Tagfüggvények dokumentációja

void collideWith (GameObject *gameObject*)

A paraméterként megadott objektummal való ütközés ellenőrzése. Ha történt ütközés, akkor a megadott objektum hitBy metódusának meghívása.

Paraméterek

gameObject	- **GameObject**

Újraimplementált ősök: **MovingObject**.

void destroy ()

A lövedék megsemmisítése, vagyis az objektum egyetlen cellájának false értékre állítása.

void hitBy (Bullet *bullet*)

Ha lövedék ütközik lövedékkel, akkor nem történik semmi.

Paraméterek

bullet	- **Bullet**

Újraimplementált ősök: **GameObject**.

void setActive (int *posX*, int *posY*, int *vecY*)

A lövedék aktiválása, és kilövése a megadott pozícióról + relative pozíció, a megadott Y irányba.

Paraméterek

posX	- int
posY	- int
vecY	- int

void step ()

A lövedék léptetése és ütközések ellenőrzése, ha a lövedék aktív.

Újraimplementált ősök: **MovingObject**.

Adattagok dokumentációja

GameBoard gameBoard[private]

Az a játéktábla, amelyikhez a lövedék tartozik.

int relativePosX[private]

A lövedék relative X pozíciója a játékoshoz képest.

int relativePosY[private]

A lövedék relative Y pozíciója a játékoshoz képest.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/gameobject/**Bullet.java**

GameBoard

Ősök: Serializable.

Publikus tagfüggvények

- **GameBoard** (int width, int height)
- void **start** ()
- void **stop** ()
- void **collisionDetection** (**Player** player)
- void **collisionDetection** (**Bullet** bullet)
- void **gameOver** (**Player** player)
- **DrawGame getBoard** ()
- KeyListener[] **getKeyListeners** ()

Privát attribútumok

- **Player leftPlayer**
- **Player rightPlayer**
- ArrayList< **GameObject** > **barriers** = new ArrayList<>()
- **DrawGame drawGame**

Részletes leírás

GameBoard osztály: A játéktábla, amely inicializálja, beállítja és tárolja a rajta elhelyezkedő objektumokat, és kezeli a közöttük lévő kapcsolatokat. Az objektumok fájlba írás miatt megvalósítja a Serializable interfészt.

Konstruktorok és destruktorok dokumentációja

GameBoard (int *width*, int *height*)

A megadott méretű pálya létrehozása. A játékosok, az azokhoz tartozó töltények, a játéktáblán lévő falak létrehozása. A játéktábla kirajzolásáért felelős drawGame beállítása.

Paraméterek

width	- int
height	- int

Tagfüggvények dokumentációja

void collisionDetection (Bullet *bullet*)

Ellenőrzi, hogy a megadott lövedék a játéktáblán ütközik-e valamelyik fallal vagy játékossal.

Paraméterek

bullet	- Bullet

void collisionDetection (Player *player*)

Ellenőrzi, hogy a megadott játékos a játéktáblán ütközik-e valamelyik fallal.

Paraméterek

player	- Player

void gameOver (Player *player*)

A játék vége. A paraméterben megadott játékos veszített. A nyertes játékos számára a drawGame segítségével egy jelvény kirajzolása.

Paraméterek

player	- Player

DrawGame getBoard ()

A kirajzolásért felelős drawGame lekérdezése.

Visszatérési érték

DrawGame

KeyListener [] getKeyListeners ()

A játékosok keylistener-eivel tér vissza.

Visszatérési érték

KeyListener[]

void start ()

A játék elindítása. Mindkét játékos és a kirajzolásért felelős drawGame külön szálon futtatása.

void stop ()

A játék befejezése. A játékosok és a kirajzolásért felelős példány szálának megállítása.

Adattagok dokumentációja

DrawGame drawGame[private]

A játéktábla képernyőre rajzolásáért felelős.

Player leftPlayer[private]

A játéktáblán lévő bal oldali játékos.

Player rightPlayer[private]

A játéktáblán lévő jobb oldali játékos.

ArrayList<GameObject> barriers = new ArrayList<>()[private]

A játéktéren lévő akadályokat tartalmazó lista.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/gameboard/**GameBoard.java**

DrawGame

Ősök: JPanel, Serializable és Runnable.

Publikus tagfüggvények

- **DrawGame** (int **width**, int height, **Player leftPlayer**, **Player rightPlayer**)
- void **addGameObject** (**GameObject** g)
- void **drawBadge** (int **badgePosX**, int **badgePosY**)
- void **paint** (Graphics g)
- void **run** ()
- void **stop** ()

Privát tagfüggvények

- void **drawImage** (Graphics g, String imageSrc, int posX)
- void **drawGameObject** (Graphics g, **GameObject** object)

Privát attribútumok

- ArrayList< **GameObject** > **gameObjects** = new ArrayList<>()
- int **width**
- int **scoreBoardHeight**
- **Player leftPlayer**
- **Player rightPlayer**
- BufferedImage **badge**
- int **badgePosX**
- int **badgePosY**
- volatile boolean **stopSignal** = false

Részletes leírás

DrawGame osztály: A játéktábla kirajzolásáért felelős osztály. Az objektumok fájlba írása miatt megvalósítja a Serializable interfészt. A külön szálon való futtatás miatt megvalósítja a Runnable interfészt.

Konstruktorok és destruktorok dokumentációja

DrawGame (int *width*, int *height*, Player *leftPlayer*, Player *rightPlayer*)

A rajzterület méretének beállítsa és az attribútumainak inicializálása.

Paraméterek

width	- int
height	- int
leftPlayer	- Player
rightPlayer	- Player

Tagfüggvények dokumentációja

void addGameObject (GameObject *g*)

Kirajzolandó objektum hozzáadása.

Paraméterek

g	- GameObject

void drawBadge (int *badgePosX*, int *badgePosY*)

A jelvény pozíciójának megadása, és a jelvény képének beolvasása.

Paraméterek

badgePosX	- int
badgePosY	- int

void drawGameObject (Graphics *g*, GameObject *object*)[private]

A paraméterként átadott objektum kirajzolása.

Paraméterek

g	- Graphics
object	- Gameobject

void drawImage (Graphics *g*, String *imageSrc*, int *posX*)[private]

A megadott helyre a megadott forrású kép kirajzolása.

Paraméterek

g	- Graphics
imageSrc	- String
posX	- int

void paint (Graphics *g*)

A pontszámok, a jelvény és a játéktáblán lévő objektumok kirajzolása.

Paraméterek

g	- Graphics

void run ()

A Runnable interfészhez tartozó metódus. A játéktábla kirajzolásának külön szálon való futtatását szolgálja.

void stop ()

A játéktábla kirajzolásához tartozó futó szál befejezése.

Adattagok dokumentációja

BufferedImage badge[private]

A jelvény képe, amit a győztes játékos kap meg.

int badgePosX[private]

A jelvény X pozíciója.

int badgePosY[private]

A jelvény Y pozíciója.

ArrayList<GameObject> gameObjects = new ArrayList<>()[private]

Player leftPlayer[private]

Az eredményjelző magassága. A bal oldali játékos.

Player rightPlayer[private]

A jobb oldali játékos.

int scoreBoardHeight[private]

A rajzterület szélessége.

volatile boolean stopSignal = false[private]

A játéktábla kirajzolásához tartozó futó szál befejezésének jelzése.

int width[private]

A kirajzolandő objektumok listája.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/gameboard/**DrawGame.java**

PlayerKeyListener

Ősök: KeyAdapter és Serializable.

Publikus tagfüggvények

- **PlayerKeyListener** (**Player player**, Integer moveUpKey, Integer moveDownKey, Integer moveLeftKey, Integer moveRightKey, Integer fireKey)
- void **keyPressed** (KeyEvent e)
- void **keyReleased** (KeyEvent e)
- void **blockMovementKey** (**ControlKey** key, boolean block)

Privát attribútumok

- **Player player**
- Set< Integer > **pressed** = new HashSet<Integer>()
- HashMap< **ControlKey**, Integer > **keys** = new HashMap<**ControlKey**, Integer>()
- Set< Integer > **blockedKeys** = new HashSet<Integer>()
- int **bulletVecY**

Részletes leírás

PlayerKeyListener osztály: A játékosok megadott billentyűkkel való irányítását végző osztály. Az objektumok fájlba írása miatt megvalósítja a Serializable interfészt.

Konstruktorok és destruktorok dokumentációja

PlayerKeyListener (Player *player*, Integer *moveUpKey*, Integer *moveDownKey*, Integer *moveLeftKey*, Integer *moveRightKey*, Integer *fireKey*)

Konstruktor: A megadott játékos irányításához szükséges billentyűk beállítása.

Paraméterek

player	- Player
moveUpKey	- Integer
moveDownKey	- Integer
moveLeftKey	- Integer
moveRightKey	- Integer
fireKey	- Integer

Tagfüggvények dokumentációja

void blockMovementKey (ControlKey *key*, boolean *block*)

A block paraméter értékétől függően a paraméterben megadott billentyű blokkolása, vagy engedélyzése.

Paraméterek

key	- ControlKey
block	- boolean

void keyPressed (KeyEvent *e*)

A lenyomott billentyűk detektálás, és ennek megfelelően a játékos mozgási és lövési irányának beállítása.

Paraméterek

e	- KeyEvent

void keyReleased (KeyEvent *e*)

Az elengedett billentyűk deketálása, és ennek megfelelően a játékos mozgási irányának 0-ba állítása, és a lövedék kilövése.

Paraméterek

e	- KeyEvent

Adattagok dokumentációja

Set<Integer> blockedKeys = new HashSet<Integer>()[private]

A blokkolt billentyűket tároló halmaz.

int bulletVecY[private]

A lövedék Y irányú vektora.

HashMap<ControlKey, Integer> keys = new HashMap<ControlKey, Integer>()[private]

A megfelelő ControlKey értékekhez tartozó billentyűkódokat tárolja.

Player player[private]

Az a játékos, aminek az irányításáért felelős.

Set<Integer> pressed = new HashSet<Integer>()[private]

A lenyomott billentyűket tároló halmaz.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/keylistener/PlayerKeyListener.java

PlayerSprite

Ősök: Serializable.

Leszármazottak: **LeftPlayerSprite** és **RightPlayerSprite**.

Publikus tagfüggvények

- boolean[][] **getStand** ()
- boolean[][] **getDead** ()
- boolean[][] **getFireStraight** ()
- boolean[][] **getFireUp** ()
- boolean[][] **getFireDown** ()

Részletes leírás

PlayerSprite interfész: A játékosok kinézeteit reprezetáló interfész.

Tagfüggvények dokumentációja

boolean [][] getDead ()

Lelőtt játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítják a következők: **LeftPlayerSprite** és **RightPlayerSprite**.

boolean [][] getFireDown ()

A lefele lövő játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítják a következők: **LeftPlayerSprite** és **RightPlayerSprite**.

boolean [][] getFireStraight ()

Az egyenesen lövő játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítják a következők: **LeftPlayerSprite** és **RightPlayerSprite**.

boolean [][] getFireUp ()

A felfele lövő játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítják a következők: **LeftPlayerSprite** és **RightPlayerSprite**.

boolean [][] getStand ()

Az álló játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítják a következők: **LeftPlayerSprite** és **RightPlayerSprite**.

Ez a dokumentáció az interfészről a következő fájl alapján készült:

- outlaw/sprite/PlayerSprite.java

LeftPlayerSprite

Ősök: **PlayerSprite**.

Publikus tagfüggvények

- boolean[][] **getStand** ()
- boolean[][] **getDead** ()
- boolean[][] **getFireStraight** ()
- boolean[][] **getFireUp** ()
- boolean[][] **getFireDown** ()

Statikus privát attribútumok

- static final boolean[][] **leftPlayerStand**
- static final boolean[][] **leftPlayerDead**
- static final boolean[][] **leftPlayerFireStraight**
- static final boolean[][] **leftPlayerFireUp**
- static final boolean[][] **leftPlayerFireDown**

Részletes leírás

LeftPlayer Sprite osztály: A bal oldali játékos kinézeteit tároló osztály.

Tagfüggvények dokumentációja

boolean [][] getDead ()

A bal oldali lelőtt játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítja a következőket: **PlayerSprite.**

boolean [][] getFireDown ()

A bal oldali lefele lövő játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítja a következőket: **PlayerSprite**.

boolean [][] getFireStraight ()

Az bal odali egyenesen lövő játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítja a következőket: **PlayerSprite**.

boolean [][] getFireUp ()

A bal oldali felfele lövő játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítja a következőket: **PlayerSprite**.

boolean [][] getStand ()

A bal oldali álló játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítja a következőket: **PlayerSprite**.

Adattagok dokumentációja

final boolean [][] leftPlayerDead[static], [private]

A bal oldai lelőtt játékos képe.

final boolean [][] leftPlayerFireDown[static], [private]

A bal oldai lefele lövő játékos képe.

final boolean [][] leftPlayerFireStraight[static], [private]

A bal oldai egyenesen lövő játékos képe.

final boolean [][] leftPlayerFireUp[static], [private]

A bal oldai felfele lövő játékos képe.

final boolean [][] leftPlayerStand[static], [private]

A bal oldai álló játékos képe.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/sprite/LeftPlayerSprite.java

RightPlayerSprite

Ősök: **PlayerSprite**.

Publikus tagfüggvények

- boolean[][] **getStand** ()
- boolean[][] **getDead** ()
- boolean[][] **getFireStraight** ()
- boolean[][] **getFireUp** ()
- boolean[][] **getFireDown** ()

Statikus privát attribútumok

- static final boolean[][] **rightPlayerStand**
- static final boolean[][] **rightPlayerDead**
- static final boolean[][] **rightPlayerFireStraight**
- static final boolean[][] **rightPlayerFireUp**
- static final boolean[][] **rightPlayerFireDown**

Részletes leírás

RightPlayerSprite osztály: A jobb oldali játékos kinézeteit tároló osztály.

Tagfüggvények dokumentációja

boolean [][] getDead ()

A jobb oldali lelőtt játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítja a következőket: **PlayerSprite**.

boolean [][] getFireDown ()

A jobb oldali lefele lövő játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítja a következőket: **PlayerSprite**.

boolean [][] getFireStraight ()

Az jobb odali egyenesen lövő játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítja a következőket: **PlayerSprite**.

boolean [][] getFireUp ()

A jobb oldali felfele lövő játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítja a következőket: **PlayerSprite**.

boolean [][] getStand ()

A jobb oldali álló játékos képének visszaadása.

Visszatérési érték

boolean[][]

Megvalósítja a következőket: **PlayerSprite** (*o.30*).

Adattagok dokumentációja

final boolean [][] rightPlayerDead[static], [private]

A jobb oldali lelőtt játékos képe.

final boolean [][] rightPlayerFireDown[static], [private]

A jobb oldali lefele lövő játékos képe.

final boolean [][] rightPlayerFireStraight[static], [private]

A jobb oldali egyenesen lövő játékos képe.

final boolean [][] rightPlayerFireUp[static], [private]

A jobb oldali felfele lövő játékos képe.

final boolean [][] rightPlayerStand[static], [private]

A jobb oldali álló játékos képe.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/sprite/RightPlayerSprite.java

Menu

Ősök: JPanel.

Leszármazottak: **GameMenu** és **MainMenu**.

Publikus tagfüggvények

- void **setButton** (JButton button, String name, int width, int height, ActionListener actionListener)
- BufferedImage **scale** (BufferedImage imageToScale, int width, int height)

Részletes leírás

Menu osztály: A frame-re helyezhető menük osztálya.

Tagfüggvények dokumentációja

BufferedImage scale (BufferedImage *imageToScale*, int *width*, int *height*)

Egy kép átméretezése a paraméterként megadott méretre.

Paraméterek

imageToScale	- BufferedImage
width	- int
height	- int

Visszatérési érték

BufferedImage

void setButton (JButton *button*, String *name*, int *width*, int *height*, ActionListener *actionListener*)

A panelen lávő gombok értékeinek beállítását végzi.

Paraméterek

button	- JButton
name	- String
width	- int
height	- int
actionListener	- ActionListener

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/menu/**Menu.java**

GameMenu

Ősök: **Menu**.

Publikus tagfüggvények

- **GameMenu** (int width, int height, ActionListener actionListener)
- void **startGame** ()
- void **stopGame** ()
- void **newGame** ()
- void **saveGame** ()
- boolean **loadGame** ()

Privát tagfüggvények

- void **clearGameBoard** ()

Privát attribútumok

- **GameBoard gameBoard**
- JPanel **gameBoardPanel** = new JPanel()

Részletes leírás

GameMenu osztály: A játéktáblát és a navigáló, játék állapotát visszaállító, elmentő gombokat tartalmazó és kezelő menü.

Konstruktorok és destruktorok dokumentációja

GameMenu (int *width*, int *height*, ActionListener *actionListener*)

A megadott méretnek megfelelő panel létrehozása. A panelen lévő gombok beállítása, és a paraméterként kapott actionListener hozzárendelése a gombokhoz.

Paraméterek

width	- int
height	- int
actionListener	- ActionListener

Tagfüggvények dokumentációja

void clearGameBoard ()[private]

A játéktábla eltávolítása a panelről, és a játéktábla törlése.

boolean loadGame ()

A játék betöltése fájlból serializálással. Ha sikeres volt a beolvasás a visszatérési érték true.

Visszatérési érték

boolean

void newGame ()

Új játék létrehozása, és hozzáadása a menühöz.

void saveGame ()

A játék aktuális állapotának elmentése fájlba serializálásal.

void startGame ()

A játék elindítása.

void stopGame ()

A játék befejezése.

Adattagok dokumentációja

GameBoard gameBoard[private]

A játéktáblát tartamazó változó.

JPanel gameBoardPanel = new JPanel()[private]

A menü azon része, ami játékot jeleníti meg.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/menu/GameMenu.java

MainMenu

Ősök: **Menu**.

Publikus tagfüggvények

- **MainMenu** (int width, int height, ActionListener actionListener)
- void **paint** (Graphics g)

Részletes leírás

MainMenu osztály: A program főmenüjét reprezentáló osztály.

Konstruktorok és destruktorok dokumentációja

MainMenu (int *width*, int *height*, ActionListener *actionListener*)

A megadott méretnek megfelelő panel létrehozása. A panelen lévő gombok beállítása, paraméterként kapott actionListener hozzárendelése, majd hozzáadása a panelhez.

Paraméterek

width	- int
height	- int
actionListener	- ActionListener

Tagfüggvények dokumentációja

void paint (Graphics *g*)

A főmenü hátterének kirajzolása.

Paraméterek

g	- Graphics

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/menu/MainMenu.java

Outlaw

Ősök: JFrame és ActionListener.

Publikus tagfüggvények

- **Outlaw** (int width, int height)
- void **actionPerformed** (ActionEvent e)

Privát attribútumok

- **GameMenu gameMenu**
- **MainMenu mainMenu**
- CardLayout **card** = new CardLayout()
- Container **container**

Részletes leírás

Outlaw osztály: A program megjelenítéséért felelős ablak.

Konstruktorok és destruktorok dokumentációja

Outlaw (int *width*, int *height*)

A program ablakának inicializálása a megadott méreteknek megfelelően. Egy mainMenu és egy gameMenu panel létrehozása és hozzáadása a JFrame tárolójához.

Paraméterek

width	- int
height	- int

Tagfüggvények dokumentációja

void actionPerformed (ActionEvent *e*)

A programban lévő gombokjoz tartozó események detektálásáért felelős. A Play esemény hatására létrehoz egy új játékot, átvált a játék menüre és elindítja a játékot. A Load esemény hatására, ha volt elmentett játék betölti. Az Exit esemény hatására kilép a játékból. A Back esemény hatására visszalép a fő menübe. A Reset esemény hatására újraindítja a játékot. A Save esemény hatására elmenti a játékot.

Paraméterek

e	- ActionEvent

Adattagok dokumentációja

CardLayout card = new CardLayout()[private]

CardLayout biztosítja a menük közötti váltást.

Container container[private]

A JFramehez tartozó tároló, ami a komponenseket tartalmazza.

GameMenu gameMenu[private]

A játékot tartalmazó menü.

MainMenu mainMenu[private]

A program főmenüje.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- outlaw/Outlaw.java