



# WIZARD MAZE

Programozás alapjai 3 Házi Feladat - Dokumentáció

Vass Csongor – AHQQ3A

## Tartalom

1.	A program rövid leírása.....	2
2.	A program használata (user manual).....	2
3.	Felépítés.....	4
3.1.	Package-k .....	4
3.2.	Osztályok, metódusok .....	4
3.2.1.	Osztályok leírása .....	4
3.2.2.	Osztálydiagram.....	24
4.	Fájlkezelés .....	27
5.	Fejlesztési lehetőségek, gondolatok.....	27
6.	Források.....	27

## 1. A program rövid leírása

A Wizard Maze egy többszemélyes akciójáték, melyben egy labirintusban kell az ellenfel(ek)et legyőzni. A kezdeti menüben ki lehet választani, hogy 2 vagy 3 játékos szeretne játszani. Van egy basic támadó varázslat, ami, ha eltalál egy játékost, az azonnal meghal. Ez tud a falakon pattogni egy ideig, és azt is megölheti, aki eredetileg lőtte. Ezen felül a pályán random időközönként megjelennek különleges varázslatok, amiknek más-más hatása lehet, ha felveszik őket. Az utolsó életben maradó pontja növekszik eggyel. A játék bármeddig folytatódhat, az exit gombbal lehet befejezni, a játékállást bármikor menteni lehet a játék folyamán.

## 2. A program használata (user manual)

A Wizard Maze megnyitható fejlesztői környezetben, valamint egyszerűen a WizardMaze.jar file segítségével.

A főmenüben három gomb jelenik meg, ezek a „new game”, „load” és „exit”. Az első gomb a „new game” menüre visz, a második a „load” menüre, a harmadik pedig értelemszerűen bezárja a játékot.

Új játék kezdésénél ki kell választanunk, hogy kettő („2 players”) vagy három („3 palyers”) játékos szeretne játszani. Bármelyiket megnyomva a megfelelő számú játékosal elindul a játék, megjelenik a pálya. A „Back” gomb a főmenübe visz vissza.

Játék betöltésénél két listából választhatunk annak megfelelően, hogy hány fős korábbi mentést szeretnénk betölteni. Bal oldalon a két személyes mentések, jobb oldalon a három személyes mentések. Ha valamelyiket kiválasztjuk, elérhetővé válik a „Load game” gomb, ami elindítja megfelelő számú játékosal a játékot, megjeleníti a pályát, és a mentésnek megfelelően beállítja a pontokat. Mindig csak egy pálya lehet kiválasztva. A „Back” gomb a főmenübe visz vissza.

Ha betöltött a játék, akkor **bele kell kattintani a képernyőbe**, hogy lehessen a playereket irányítani. Minden playerhez 4 gomb tartozik. Ezek az előre, hátra, balra forgás, jobbra forgás és varázslás előre. Player1: W, S, A, D, Q. Player2: fel nyíl, le nyíl, balra nyíl, jobbra nyíl, M. Player3: Num 8, Num 5, Num 4, Num 6, Num 0.

A playerek a falakon nem tudnak keresztül menni, azok között kell utat találniuk, és az ellenfeleket megölniük. Ehhez varázslatokat használnak. Alap varázslatként egy normál méretű fekete gömbbűbáj áll rendelkezésükre, ha nem vesznek fel semmit, akkor ez meghatározott idővel újratöltődik. Közvetlenül fal mellett nem tudnak lőni. A pályán random időközönként megjelennek különböző varázslat dobozok, amelyeket a játékosok felvehetnek, így egy darab különleges varázslatot használhatnak. Mindig a legutóbb felvett doboz számít. A varázslatok különböző ideig lehetnek a pályán és különböző lehet a sebességük, méretük.

Felvehető varázslatok fajtái:

- Villám varázslat: Egész pályáján sebző, falakról visszapattanó varázslat.
- Nagy, hosszú idejű varázslat: A basic varázslat nagyobb méretű, több ideig a pályán lévő változata.
- Kontroll bűbáj: A játékos tudja irányítani, miközben ő maga nem mozdul el.
- Szapora rontás: Kis méretű, nagyon gyors varázslat.

- **Ultimum magicae:** Falakon áthatoló, hatalmas méretű varázslat. Ennek a felvétele 10% eséllyel megöli a playert.

Ha már csak egy játékos maradt életben, a pontszáma megnövekszik eggyel, és új pálya tölt be. Ilyenkor mindenki újra a basic spellel rendelkezik. A pontszámok a képernyő alján jelennek meg,

A játék mentésének neve (új játéknál alapértelmezetten „Unnamed Game”, betöltötnél a betöltött neve) a képernyő alján lévő textfieldben látható. Ez bármikor szerkeszthető, és a „Save” gomb megnyomásával olyan néven menti el/írja felül a játékot (a játékosok számának megfelelő mappába), amelyik a megnyomás pillanatában ott szerepel. Amennyiben érvénytelen név szerepel a megnyomáskor, a textfield átvált „Invalid name!” üzenetre, és a mentés sikertelen.

Az „Exit” gomb megnyomásával visszakerülünk a főmenübe.

## 3. Felépítés

### 3.1. Package-k

- Wizard\_Maze
  - display
    - screens
  - gfx
  - mazes
  - states
  - things
    - input
    - movings
      - spells
    - positioning
    - statics
      - magics
  - tiles
  - utils

### 3.2. Osztályok, metódusok

#### 3.2.1. Osztályok leírása

##### 3.2.1.1. *Launcher*

#### Felelősségek

Az indító main-t tartalmazza, példányosít egy Game-t, és elindítja azt.

**Package:** Wizard\_Maze

-

#### Metódusok

+ <u>main()</u>	Elindítja a játékot.
-----------------	----------------------

##### 3.2.1.2. *Game*

#### Felelősségek

A játékot vezérlő, külön szálon futó osztály, game loop.

**Package:** Wizard\_Maze

#### Attribútumok

-display:Display	Kijelző
-title:String	Kijelző címe
-width:int	Kijelző szélessége
-height:int	Kijelző magassága
-running:boolean	Fut-e a játék, default false
-thread:Thread	Játék vezérlő szál
+menuState:State	Menü állapot
+gameState:State	Játék állapot
+loadState:State	Betöltés állapot
+newGameState:State	Új játék állapot

-timer:Timer	Beépített időzítő
--------------	-------------------

### Metódusok

+Game(String title, int width, int height)	Konstruktor, beállítja az értelemszerűen kapott attribútumokat
+tick()	Aktuális állapotot lépteti
+render()	Aktuális állapotot kirajzolja
+startGameState(int numberOfPlayers, String saveName)	Beállítja a játék állapot attribútumot egy új példányra a megadott játékoszámmal és mentésnévvel
+init()	Inicializálja a változókat, elindítja a grafikus elemeke betöltését, beállítja a menü állapotot
+run()	Thread run metódusa, meghívja az init() függvényt, elindítja a gameloop-ot, ~60 alkalommal másodpercenként meghívja a tick() és render() metódusokat, kiírja a konzolra az FPS számot, a loop végén leállítja a threadet, a timert, és kilép
+synchronized start()	Létrehozza és elindítja a threadet, amennyiben az még nincs elindítva
+synchronized stop()	Leállítja a threadet, ha még fut
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

#### 3.2.1.3. Display

### Felelősségek

Kijelző, a frame-t és paneleket létrehozza és beállítja.

**Package:** Wizard\_Maze.display

### Attribútumok

-frame:JFrame	A játék ablaka, kerete
-title:String	Az ablak címe
-width:int	Az ablak szélessége
-height:int	Az ablak magassága
-screens:JPanel	Az eltérő állapotok paneleit tartalmazó, card layouttal rendelkező panel
-menu_screen:MenuScreen	A menü képernyő
-game_screen:GameScreen	A játék képernyő
-load_screen:LoadScreen	A betöltés képernyő
-new_game_screen:NewGameScreen	Az új játék képernyő

### Metódusok

+Display(String title, int width, int height)	Konstruktor, beállítja a kapott adatokat, elindítja az „összerakást”
+createDisplay()	Létrehozza és beállítja és összerakja a megfelelő elemeket.
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

#### 3.2.1.4. Screen

##### Felelősségek

JPanelből leszármaztatott absztrakt osztály a különböző képernyők általános kezelésére.

**Package:** Wizard\_Maze.display.screens

##### Attribútumok

#width:int	A képernyő szélessége
#height:int	A képernyő magassága
#display:Display	A kijelző, amely létrehozta

##### Metódusok

+Screen(String name, Display display)	Konstruktor, beállítja az értelemszerű attribútumokat
+abstract createScreen()	Absztrakt metódus a képernyő generálására
+abstract refresh()	Absztrakt metódus a képernyő frissítésére

#### 3.2.1.5. GameScreen

##### Felelősségek

A játék képernyője, Screen osztály leszármazottja

**Package:** Wizard\_Maze.display.screens

##### Attribútumok

-canvas:Canvas	A képernyő vászna
-buttonPanel:JPanel	A gombok panelje
-saveButton:JButton	Mentés gomb
-exitButton:JButton	Kilépés gomb
-saveIcon: MyScaledImageIcon	Mentés gomb ikonja
-saveOverIcon: MyScaledImageIcon	Mentés gomb ikonja, mikor rajta van az egér
-exitIcon: MyScaledImageIcon	Kilépés gomb ikonja
-exitOverIcon:MyScaledImageIcon	Kilépés gomb ikonja, mikor rajta van az egér

##### Belül implementált osztályok

SaveButtonActionListener	ActionListenert implementáló osztály a save gombhoz. Eseménye a játék mentése
BackButtonActionListener	ActionListenert implementáló osztály a back gombhoz. Eseménye visszalépés a főmenübe

##### Metódusok

+GameScreen(String name, Display display)	Konstruktor, beállítja az attribútumokat, hozzáadja az action listenereket a gombokhoz
+createScreen()	Legenerálja a képernyőt
+refresh()	Absztrakt ősosztály metódusa, nincs működés társítva
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

### 3.2.1.6. LoadScreen

#### Felelősségek

A játék betöltés képernyője, Screen osztály leszármazottja

**Package:** Wizard\_Maze.display.screens

#### Attribútumok

-buttonPanel:JPanel	Panel a gomboknak
-scrollPanePanel:JPanel	Panel a scrollPane-eknek
-textPanel:JPanel	Panel a textfieldnek
-backButton:JButton	Vissza gomb
-startButton:JButton	Start gomb
-twoPlayerScroll:JScrollPane	Két játékos mentések scrollPane-je
-threePlayerScroll:JScrollPane	Három játékos mentések scrollPane-je
-twoPlayerListModel:DefaultListModel<String>	Listamodell a két játékos mentéseknek
-threePlayerListModel:DefaultListModel<String>	Listamodell a három játékos mentéseknek
-twoPlayerList:JList<String>	Lista a két játékos mentéseknek
-threePlayerList:JList<String>	Lista a három játékos mentéseknek
-twoPlayer:JLabel	Két játékos felirat
-threePlayer:JLabel	Három játékos felirat
-loadName:String	Mentés neve, default"Unnamed"
-numberOfPlayers:int	Betölteni kívánt játék hány személyes
-twoPlayerSaveFiles:String[]	Két játékos mentések string tömbje
-threePlayerSaveFiles:String[]	Három játékos mentések string tömbje
-bw:int	Buttonok szélessége
-bh:int	Buttonok magassága
-loadGameImageIcon: MyScaledImageIcon	Load game gomb ikonja
-loadGameOverIcon: MyScaledImageIcon	Load game gomb ikonja, ha az egér fölötte van
-backIcon: MyScaledImageIcon	Back gomb ikonja
-backOverIcon: MyScaledImageIcon	Back gomb ikonja, ha rajta van az egér

#### Belül implementált osztályok

StartButtonActionListener	ActionListenert implementáló osztály a start gombhoz. Eseménye a kiválasztott játék betöltése, és játékállapotra váltás
BackButtonActionListener	ActionListenert implementáló osztály a back gombhoz. Eseménye visszalépés a főmenübe
TwoPlayerListActionlistener	ActionListenert implementáló osztály a twoPlayerListhez. Eseménye a másik lista kijelölésének törlése, a start button engedélyezése és a megfelelő attribútumok beállítása
ThreePlayerListActionlistener	ActionListenert implementáló osztály a threePlayerListhez. Eseménye a másik lista kijelölésének törlése, a start button engedélyezése és a megfelelő attribútumok beállítása

#### Metódusok

+LoadScreen(String name, Display display)	Konstruktor, beállítja a megfelelő értékeket, meghívja a paneleket összepakoló függvényt, és társítja az action listenereket
---	--



+createScreen()	Legenerálja a képernyőt
+refresh()	Absztrakt ősosztály metódusa, frissíti a listát, deaktiválja a start gombot
+setList()	Betölti a listákat

### 3.2.1.7. *MenuScreen*

#### Felelősségek

A menü képernyője, Screen osztály leszármazottja

**Package:** Wizard\_Maze.display.screens

#### Attribútumok

-logo:JLabel	Logó JLabel-je
-newGameButton:JButton	Új játék gomb
-loadButton:JButton	Betöltés gomb
-exitButton:JButton	Kilépés gomb
-dy:int	Függőleges távolságegység
-bw:int	Gomb szélesség
-bh:int	Gomb magasság
-logoIcon: MyScaledImageIcon	Logó ikon
-startIcon: MyScaledImageIcon	Új játék gomb ikonja
-startOverIcon: MyScaledImageIcon	Új játék gomb ikonja, ha egér fölötte
-loadIcon: MyScaledImageIcon	Betöltés gomb ikonja
-loadOverIcon: MyScaledImageIcon	Betöltés gomb ikonja, ha egér fölötte
-exitIcon: MyScaledImageIcon	Kilépés gomb ikonja
-exitOverIcon: MyScaledImageIcon	Kilépés gomb ikonja, ha egér fölötte

#### Belül implementált osztályok

NewGameButtonActionListener	ActionListenert implementáló osztály a new game gombhoz. Eseménye a new game állapotra váltás
LoadButtonActionListener	ActionListenert implementáló osztály a new game gombhoz. Eseménye a load állapotra váltás
ExitButtonActionListener	ActionListenert implementáló osztály az exit gombhoz. Eseménye a játék leállítása

#### Metódusok

+MenuScreen(String name, Display display)	Konstruktor, beállítja az értékeket, meghívja a generáló függvényt, hozzákapcsolja a listenereket a gombokhoz
+createScreen()	Legenerálja a képernyőt
+refresh()	Absztrakt ősosztály absztrakt metódusa implementálása, nincs társítva működés

### 3.2.1.8. *NewGameScreen*

#### Felelősségek

Az új játék képernyője, Screen osztály leszármazottja

**Package:** Wizard\_Maze.display.screens

### Attribútumok

-logo: JLabel	Logó JLabelje
-twoPlayerButton: JButton	Két játékos gomb
-threePlayerButton: JButton	Három játékos gomb
-backButton: JButton	Vissza gomb
-dy: int	Függőleges távolságegység
-bw: int	Gomb szélesség
-bh: int	Gomb magasság
-logoIcon: MyScaledImageIcon	Logó ikon
-twoPlayerIcon: MyScaledImageIcon	Két játékos gomb ikon
-twoPlayerOverIcon: MyScaledImageIcon	Két játékos gomb ikon, ha egér fölötté
-threePlayerIcon: MyScaledImageIcon	Három játékos gomb ikon
-threePlayerOverIcon: MyScaledImageIcon	Három játékos gomb ikon, ha egér fölötté
-backIcon: MyScaledImageIcon	Vissza gomb ikon
-backOverIcon: MyScaledImageIcon	Vissza gomb ikon, ha egér fölötté

### Belül implementált osztályok

TwoPlayerButtonActionListener	ActionListenert implementáló osztály a two player gombhoz. Eseménye két játékosal játék elindítása
ThreePlayerButtonActionListener	ActionListenert implementáló osztály a three player gombhoz. Eseménye három játékosal játék elindítása
BackButtonActionListener	ActionListenert implementáló osztály az back gombhoz. Eseménye visszalépés a menübe

### Metódusok

+NewGameScreen(String name, Display display)	Konstruktor, beállítja az értékeket, meghívja a generáló függvényt, hozzákapcsolja a listenereket a gombokhoz
+createScreen()	Legenerálja a képernyőt
+refresh()	Absztrakt ősosztály absztrakt metódusa implementálása, nincs társítva működés

#### 3.2.1.9. Assets

### Felelősségek

Textúrákat tárolja, statikusan elérhető osztály.

**Package:** Wizard\_Maze.gfx

### Attribútumok

-final width:int	Textúra alap szélessége
-final height:int	Textúra alap magassága
<name>:BufferedImage	A textúrák
<name>:BufferedImage[]	A több fajtájú textúrák tömbjei

### Metódusok

+init()	A rácslapról betölti a textúrákat a változókbá
+loadImage():BufferedImage	A rácslap betöltése

### 3.2.1.10. *SpriteSheet*

#### Felelősségek

Rácslap egy képből

**Package:** Wizard\_Maze.gfx

#### Attribútumok

-sheet:BufferedImage	A rácslap kép
----------------------	---------------

#### Metódusok

+SpriteSheet(BufferedImage sheet)	Konstruktor
+crop(intx, int y, int width, int height):BufferedImage	Visszatér a rácslap megfelelő koordinátájú és méretű részével

### 3.2.1.11. *MyScaledIcon*

#### Felelősségek

Átméretezett ikonok swing objektumokhoz, az ImageIcon osztály leszármazottja.

**Package:** Wizard\_Maze.gfx

#### Attribútumok

-

#### Metódusok

MyScaledImageIcon(BufferedImage image, int width, int height)	Konstruktor, megfelelő méretűre méretezett Image objektum a megadott BufferedImage-ből
---	--

### 3.2.1.12. *Maze*

#### Felelősségek

A pálya, kezeli a rajta lévő thingeket.

**Package:** Wizard\_Maze.mazes

#### Attribútumok

-width: int	A labirintus szélessége
-height: int	A labirintus magassága
-tiles: int[][]	A labirintus „lapjainak” mátrixa
-spawnX: int[]	A játékosok spawn pontja x koordinátáinak tömbje
-spawnY: int[]	A játékosok spawn pontja y koordinátáinak tömbje
-players: Player[]	A labirintusban levő játékosok tömbje
-gameState: GameState	A játékállapot, amely létrehozta a labirintust
-path: String	A pályák mappájának elérési útja
-thingManager: ThingManager	A labirintus ThingManagere a rajta lévő entitások számontartására és kezelésére

-magicManager: MagicManager	labirintus ThingManagere a rajta lévő varázslat dobozok számontartására és kezelésére
-----------------------------	---

### Metódusok

+Maze(GameState gameState, String path, int numberOfPlayers)	Konstruktor, inicializálja a változókat, meghívja a betöltést, elindítja a magic managert
+changeMaze()	Betölt egy új pályát, visszaállítja a kezdeti értékeket
+tick()	Ha már csak egy player van életben, pályát vált, a thingManagert tickeli
+render(Graphics g)	Rendereli a pálya (szilárd) elemeit és a thingManager elemeit
+loadMaze(String path)	Betölti egy file-ból a pályát, beállítja a spawn pontokat
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

#### 3.2.1.13. State

### Felelősségek

Abstract osztály a játékállapotoknak.

**Package:** Wizard\_Maze.states

### Attribútumok

<u>-currentState:State</u>	Statikus state, az épp aktuális állapotot jelzi. Default null
#game:Game	A Game példány, amelyik létrehozta
#screen:Screen	Az állapothoz tartozó képernyő

### Metódusok

<u>+setState(Game game, State state)</u>	Átállítja az aktuális állapotot, váltja a képernyőt és frissíti azt, amire váltottunk
<u>+getState():State</u>	Visszaadja az aktuális állapotot
+State(Game game, Screen screen)	Konstruktor
+abstract tick()	Absztrakt tick metódus
+abstract render()	Absztrakt render metódus
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

#### 3.2.1.14. GameState

### Felelősségek

A játék állapot, az állapot osztály leszármazottja.

**Package:** Wizard\_Maze.states

### Attribútumok

-maze:Maze	A játékállapot labirintusa
-saveName:String	A játék mentésének neve
-numberOfPlayers:int	Játékosok száma
-gameScreen:GameScreen	Az állapothoz tartozó képernyő

-bs:BufferStrategy	BufferStrategy a rajzoláshoz
-g:Graphics	Graphics osztály a rajzoláshoz
-keyManager:KeyManager	Irányításhoz KeyManager

### Metódusok

+GameState(Game game, Screen screen, int numberOfPlayers, String saveName)	Konstruktor, értékeket ad a változóknak
+tick()	A labirintusát tickeli
+render()	A labirintus elemeit rendereli, valamit kiírja a pontszámot
+saveGame()	Lementi a játékot a beállított néven (ha az valid)
+loadGame(String path)	Betölti a játékot a megfelelő útvonalról
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

#### 3.2.1.15. LoadState

##### Felelősségek

A load állapot, az állapot osztály leszármazottja.

**Package:** Wizard\_Maze.states

##### Attribútumok

-

##### Metódusok

+LoadState(Game game, Screen screen)	Konstruktor, meghívja az ős konstruktorát
+tick()	Az ős osztály absztrakt tick() metódusa, nincs működés társítva hozzá
+render()	Az ős osztály absztrakt render() metódusa, nincs működés társítva hozzá

#### 3.2.1.16. MenuState

##### Felelősségek

A menü állapot, az állapot osztály leszármazottja.

**Package:** Wizard\_Maze.states

##### Attribútumok

-

##### Metódusok

+MenuState(Game game, Screen screen)	Konstruktor, meghívja az ős konstruktorát
+tick()	Az ős osztály absztrakt tick() metódusa, nincs működés társítva hozzá
+render()	Az ős osztály absztrakt render() metódusa, nincs működés társítva hozzá

#### 3.2.1.17. NewGameState

##### Felelősségek

Az új játék állapot, az állapot osztály leszármazottja.

**Package:** Wizard\_Maze.states

#### Attribútumok

-

#### Metódusok

+NewGameState(Game game, Screen screen)	Konstruktor, meghívja az ősz konstruktorát
+tick()	Az ősz osztály absztrakt tick() metódusa, nincs működés társítva hozzá
+render()	Az ősz osztály absztrakt render() metódusa, nincs működés társítva hozzá

#### 3.2.1.18. Thing

#### Felelősségek

„Dolog” osztály, a pályán megtalálható bármilyen entitás absztrakt őssztálya.

**Package:** Wizard\_Maze.things

#### Attribútumok

#x: float	x koordináta
#y: float	y koordináta
#angle:	Elfordulási szög
#width: int	Szélesség
#height: int	Magasság
#texture: BufferedImage	Textúra
#gameState: GameState	A GameState, amelyik (közvetve) létrehozta
#hitBox: Shape	Az ütközést detektáló alakzat
#hitBoxWidth: int	A hitbox szélessége
#hitBoxHeight: int	A hitbox magassága

#### Metódusok

+ Thing(GameState gameState, int x, int y, double angle, int width, int height, int hitBoxWidth, int hitBoxHeight, BufferedImage texture)	Konstruktor, beállítja az értékeket
+tick()	Alap esetben nincs működés a tick metódushoz, leszármazottak felülírhatják
+render(Graphics g)	Alap thing kirajzolás
+ checkThingCollisionsWithThings(float xOffset, float yOffset)	Megvizsgálja, hogy ütközik-e másik thinggel, ha igen, meghívja annak a collideWith metódusát átadva önmagát
+ doCollide(Thing t): boolean	Visszaadja, hogy a két thing hitboxa metszi-e egymást
+ collideWith(Thing t)	Jelzi a másik thingnek, hogy ütköztek
+ hitBy(Player p)	Metódus, ha playerrel ütközik (default nem definiált)
+ hitBy(Spell b)	Metódus, ha varázslattal ütközik (default nem definiált)

Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat
--	---

### 3.2.1.19. *ThingManager*

#### Felelősségek

Thingeket tároló és kezelő osztály

**Package:** Wizard\_Maze.things

#### Attribútumok

-things:ArrayList<Thing>	Thingeket tároló ArrayList
--------------------------	----------------------------

#### Metódusok

+ThingManager()	Konstruktor, létrehozza az ArrayList-et
+tick()	Meghívja a listában lévő elemekre azok tick metódusát
+render(Graphics g)	Meghívja a listában szereplő elemekre azok render metódusát
+addThing(Thing t)	Thing hozzáadása a listához
+removeThing(Thing t)	Thing kitörlése a listából
+removeAll()	Teljes lista ürítése
+getThings():ArrayList<Thing>	Az ArrayList átadása

### 3.2.1.20. *KeyManager*

#### Felelősségek

Gombnyomás értékek figyelése, tárolása és visszaadása. A KeyListenert implementálja

**Package:** Wizard\_Maze.things.input

#### Attribútumok

-keys:boolean[]	A billentyűzet gombjainak számával indexelt boolean tömb
-----------------	--

#### Metódusok

+KeyManager()	Konstruktor, létrehozza a tömböt
+keyPressed(KeyEvent e)	Ha meg van nyomva egy gomb, akkor az indexedig elemét a tömbnek igazra állítja
+keyReleased(KeyEvent e)	Ha el van engedve egy gomb, akkor az indexedig elemét a tömbnek hamisra állítja
+keyTyped(KeyEvent e)	Absztrakt ősosztály kötelezően implementált, de nem használt metódusa
+isKeyPressed(int i):boolean	Visszaadja, hogy az adott indexű gomb meg van-e nyomva

### 3.2.1.21. *MovingThing*

#### Felelősségek

Absztrakt osztály a mozgó thingek kezelésére. A Thing leszármazottja.

**Package:** Wizard\_Maze.movings

### Attribútumok

+final DEFAULT_SPEED:float	Alapértelmezett sebesség
#speed:float	Sebesség
#xMove:int	x irányú mozgás
#yMove:int	y irányú mozgás
#rotateAngle:double	Szögelfordulás

### Metódusok

+MovingThing(GameState gameState, int x, int y, double angle, int width, int height, int hitBoxWidth, int hitBoxHeight, BufferedImage texture)	Konstruktor, beállítja értékeket
+move()	Meghívja a két koordinátára a moveCoordinate-t, és a szöggel a rotate-t
+moveCoordinate(int xMove, int yMove)	Ha elmozgatás után nem metszene falat, akkor mozgatja a hitboxot
+rotate(double angle)	Ha forgatás után nem metszene falat, akkor elforgatja a hitboxot
+hitWall(int tileX, int tileY)	Mit tegyen, ha falnak ütközött. Default nincs működés társítva, leszármazottak felülírhatják
+abstract kill()	Absztrakt metódus egy mozgó dolog „halálára”
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

#### 3.2.1.22. Player

### Felelősségek

Játékos osztálya, MovingThing leszármazottja.

**Package:** Wizard\_Maze.movings

### Attribútumok

+final DEFAULT_PLAYER_WIDTH: int	Játékos alapértelmezett szélessége
+final DEFAULT_PLAYER_HEIGHT: int	Játékos alapértelmezett magassága
-PLAYERS_ALIVE:int	Élő játékosok, default 0
-points:int	Játékos pontjai
-control_keys:int[]	Írányító gombok kódjai tömbje
-casted:boolean	Lőtt-e, default false
-alive:boolean	Él-e, default true
-spellManager:SpellManager	A SpellManager példánya
-frozen:boolean	Le van-e fagyasztva, default false

### Metódusok

+Player(GameState gameState, int x, int y, double angle, BufferedImage texture)	Konstruktor, beállítja az értékeket, növeli az élő játékosok számát eggyel
+tick()	Olvas a billentyűzetről, és ha nem fagyott, mozog (MovingThing tick() felülírása)
+kill()	Alive falsr, player eltávolítása a labirintusból, élő játékosok számának csökkentése



+collideWith(Thing t)	Ha ütközik egy thinggel, meghívja annak hitBy(Player p) metódusát önmagát átadva
+hitBy(Player/Magic/Spell p)	Absztrakt őssztály implementált metódusai, nincs működés társítva
+cast()	Varázslás, a spellManagerének cast() függvényét hívja meg
+getInput()	xMove, yMove és rotateAngle beállítása a lenyomott billentyűk és egyéb feltételek alapján
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

### 3.2.1.23. Spell

#### Felelősségek

Varázslatok absztrakt osztálya, MovingThing leszármazottja.

**Package:** Wizard\_Maze.spells

#### Attribútumok

#DEFAULT TRIPTIME:int	A pályán maradás idejének alapértelmezett, statikus értéke
#tripTime:int	A pályán maradás ideje
#timer:Timer	Beépített időzítő
#player:Player	A játékos, aki megidézte

#### Metódusok

+Spell(GameState gameState, int x, int y, double angle, int width, int height, BufferedImage texture, int tripTime)	Konstruktor, beállítja az értékeket
+tick()	Meghívja a getDirection() és move() függvényeket (őssztály tick függvényének felülírása)
+getDirection()	Beállítja xMove és yMove értékét
+shoot(Player player)	Ha megidézi a játékos a varázslatot, akkor a tripTime leteltével az tűnjön el, és a player SpellManagere resetelődjön
+hitWall(int xMove, int yMove)	Felüldefiniált hitWall, a falról való lepattanást biztosítja
+hitByPlayer(Player p)	Alapesetben, ha egy varázslat eltalálja a játékost, az meghal és a varázslat eltűnik
+kill()	Önmaga kitörlése a labirintus ThingManageréből
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

### 3.2.1.24. BasicSpell

#### Felelősségek

Az alapértelmezett varázslat, a Spell leszármazottja.

**Package:** Wizard\_Maze.spells

#### Attribútumok

-

### Metódusok

+BasicSpell(GameState gameState)	Konstruktor, őssosztály konstruktorát hívja, beállítja a sebességet
----------------------------------	---

#### 3.2.1.25. *ControllSpell*

### Felelősségek

Player által irányított varázslat, a Spell leszármazottja.

**Package:** Wizard\_Maze.spells

### Attribútumok

-

### Metódusok

+ControllSpell(GameState gameState)	Konstruktor, őssosztály konstruktorát hívja, beállítja a sebességet
+move()	Felüldefiniálja az alapértelmezett move() metódust, lefagyasztja a playert, és átveszi annak forgásait

#### 3.2.1.26. *LightningSpell*

### Felelősségek

Teljes pályáján sebző, falról lepattanó varázslat, a Spell leszármazottja.

**Package:** Wizard\_Maze.spells

### Attribútumok

-path:Path2D.Double	A mozgás pályája
---------------------	------------------

### Metódusok

+LightningSpell(GameState gameState)	Konstruktor, őssosztály konstruktorát hívja, beállítja a sebességet, inicializálja a path-ot
+tick()	Felüldefiniálja az alapértelmezett tick metódust, még hozzávesz egy append műveletet
+render(Graphics g)	Felüldefiniálja az alapértelmezett render metódust, kirajzolja a pályát
+doCollide(Thing t):boolean	Felüldefiniálja az alapértelmezett doCollide metódust, a teljes pályára nézi

#### 3.2.1.27. *LongBigSpell*

### Felelősségek

Az alapértelmezett varázslat nagyobb, és hosszabb ideig pályán lévő változata, a Spell leszármazottja.

**Package:** Wizard\_Maze.spells

## Attribútumok

### -Metódusok

+LongBigSpell(GameState gameState)	Konstruktor, őssztály konstruktorát hívja, beállítja a sebességet
------------------------------------	---

### 3.2.1.28. *SpeedSpell*

#### Felelősségek

A gyors varázslat, a Spell leszármazottja.

**Package:** Wizard\_Maze.spells

## Attribútumok

-

### Metódusok

+SpeedSpell(GameState gameState)	Konstruktor, őssztály konstruktorát hívja, beállítja a sebességet
----------------------------------	---

### 3.2.1.29. *UltimateSpell*

#### Felelősségek

Falakon áthatoló, teljes pályáján sebző, nagy varázslat, a Spell leszármazottja.

**Package:** Wizard\_Maze.spells

## Attribútumok

-path:Path2D.Double	A mozgás pályája
---------------------	------------------

### Metódusok

+LightningSpell(GameState gameState)	Konstruktor, őssztály konstruktorát hívja, beállítja a sebességet, inicializálja a path-ot
+tick()	Felüldefiniálja az alapértelmezett tick metódust, még hozzávesz egy append műveletet
+render(Graphics g)	Felüldefiniálja az alapértelmezett render metódust, kirajzolja a pályát
+doCollide(Thing t):boolean	Felüldefiniálja az alapértelmezett doCollide metódust, a teljes pályára nézi
+hitByPlayer(Player p)	Felüldefiniálja a hitBy(Player p) metódust, a varázslat nem tűnik el attól, hogy ütköznek, csak a player hal meg
+moveCoordinate(int xMove, int yMove)	Felüldefiniálja az alapértelmezett moveCoordinate függvényt, átmegy a falakon

### 3.2.1.30. *SpellManager*

#### Felelősségek

Spell objektumokat tároló és kezelő osztály.

**Package:** Wizard\_Maze.spells

#### Attribútumok

-player:Player	A player, akié a spellmanager
-ammunition:Spell	A „tárban” lévő varázslat
-gameState:GameState	A játékállapot, ami (közvetve) létrehozta

#### Metódusok

+ SpellManager(GameState gameState, Player player)	Konstruktor, beállítja a változókat, reseteli az ammunitiont
+cast()	Az ammunitionban lévő varázslat megidézése a labirintusba a playertől a player irányába nézve. Csak akkor lehet, ha nem közvetlen falba kerülne.
+reset()	Kiolvasztja a playert, ha fagyva van, és egy basicSpell-lel meghívja az add függvényt
+add(Spell spell)	Az ammunitiont a spell-re állítja és hozzárendeli a playert
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

#### 3.2.1.31. Transformer

##### Felelősségek

Shapeket manipuláló segédosztály.

**Package:** Wizard\_Maze.positioning

#### Attribútumok

-shape:Shape	A módosítani kívánt alakzat
--------------	-----------------------------

#### Metódusok

+Transformer(Shape shape)	Konstruktor, beállítja a shapet
+rotete(double angle):Shape	Visszaadja az alakzat közepe körül angle szögben elforgatottját
+rotate(double angle, int x, int y):Shape	Visszaadja az alakzat (x,y) pont körüli anfle szögben elforgatottját
+move(int tx, int ty):Shape	Visszaadja az alakzat x irányban tx és y irányban ty-nal eltoltját

#### 3.2.1.32. Static Thing

##### Felelősségek

Absztrakt osztály a nem mozgó thinegnek. A Thing osztály leszármazottja.

**Package:** Wizard\_Maze.statics

#### Attribútumok

-

#### Metódusok

StaticThing(GameState gameState, int x, int y, double angle, int width, int height, BufferedImage texture)	Konstruktor, meghívja az őssztály konstruktorát
--	---

### 3.2.1.33. *Magic*

#### Felelősségek

A felvehető varázsdobozok osztálya. A StaticThing leszármazottja.

**Package:** Wizard\_Maze.magics

#### Attribútumok

+final DEFAULT_WEAPON_WIDTH: int	Alapértelmezett doboz szélesség
+final DEFAULT_WEAPON_HEIGHT: int	Alapértelmezett doboz magasság
#containedSpell:Spell	A tárolt bűbáj

#### Metódusok

+Magic(GameState gameState, int x, int y, BufferedImage texture, Spell containedSpell)	Konstruktor, meghívja az őssztály konstruktorát, beállítja a containedSpellt
+hitBy(Player player)	Ha playerrel ütközik, akkor kapja meg a player a benne található varázslatot, majd tűnjön el
+kill()	Kiveszi magát a labirintus ThingManageréből

### 3.2.1.34. *ControllMagic*

#### Felelősségek

ControllSpellt tárol, Magic osztály leszármazottja.

**Package:** Wizard\_Maze.magics

#### Attribútumok

-

#### Metódusok

+ public ControllMagic(GameState gameState, int x, int y)	Konstruktor, meghívja az őssztály konstruktorát
---	---

### 3.2.1.35. *LightningMagic*

#### Felelősségek

LightningSpellt tárol, Magic osztály leszármazottja.

**Package:** Wizard\_Maze.magics

#### Attribútumok

-

#### Metódusok

+ public LightningMagic(GameState gameState, int x, int y)	Konstruktor, meghívja az őssztály konstruktorát
--	---

### 3.2.1.36. LongBigMagic

#### Felelősségek

LongBigSpellt tárol, Magic osztály leszármazottja.

**Package:** Wizard\_Maze.magics

#### Attribútumok

-

#### Metódusok

+ public LongBigMagic(GameState gameState, int x, int y)	Konstruktor, meghívja az őssztály konstruktorát
--	---

### 3.2.1.37. SpeedMagic

#### Felelősségek

SpeedSpellt tárol, Magic osztály leszármazottja.

**Package:** Wizard\_Maze.magics

#### Attribútumok

-

#### Metódusok

+ public SpeedMagic(GameState gameState, int x, int y)	Konstruktor, meghívja az őssztály konstruktorát
--	---

### 3.2.1.38. UltimateMagic

#### Felelősségek

UltimateSpellt tárol, Magic osztály leszármazottja. 10%, hogy megöli a playert, ha az felveszi.

**Package:** Wizard\_Maze.magics

#### Attribútumok

-

#### Metódusok

+ public UltimateMagic(GameState gameState, int x, int y)	Konstruktor, meghívja az őssztály konstruktorát
+hitBy(Player player)	Felülírja az alapértelmezett hitByPlayer metódust, 10% eséllyel megöli a playert, aki felveszi

### 3.2.1.39. MagicManager

#### Felelősségek

A labirintusban lévő varázslat dobozokat kezelő osztály.

**Package:** Wizard\_Maze.magics

### Attribútumok

-maze:Maze	A labirintus, ahova spawnolja a varázslatokat
-timer:Timer	Beépített időzítő

### Metódusok

+MagicManager(Maze maze)	Konstruktor, beállítja az értékeket
+start()	Beindítja az időzítőt, és 5 másodpercenként 75 százalék valószínűséggel spawnol egy random varázslat dobozt egy random koordinátára

#### 3.2.1.40. *Tile*

### Felelősségek

A pályát felépítő alapelemek osztálya.

### Attribútumok

+tyle types:Tile[]	Tile típusok tömbje
+floorTile:Tile	Nem tömör padlózat fajta
+WallTile:Tile	Tömör fal fajta
+TILEWIDTH:int	Egy csempe szélessége
+TILEHEIGHT:int	Egy csempe textúrája
#texture:BufferedImage	Az adott csempe textúrája
#final id:int	Az adott csempe azonosítója

### Metódusok

+Tile(BufferedImage texture, int id)	Konstruktor, beállítja az attribútumokat
+render(Graphics g, int x, int y)	Kirajzolja az adott textúrát az adott koordinátákra
+isSolid():boolean	Tömör-e az adott csempe, alapértelmezetten nem
Az osztályhoz tartozó getterek és setterek	Beállítják vagy visszaadják az attribútumokat

#### 3.2.1.41. *FloorTile*

### Felelősségek

Nem tömör, padlócssempe osztály. Tile leszármazottja.

### Attribútumok

-

### Metódusok

+FloorTile(int id)	Konstruktor, meghívja az ősoosztály konstruktorát
--------------------	---

#### 3.2.1.42. *WallTile*

### Felelősségek

Tömör fal csempe. A Tile leszármazottja.

#### Attribútumok

-

#### Metódusok

+WallTile(int id)	Konstruktor, meghívja az őssztály konstruktorát
+isSolid():boolean	Felülírja az alapértelmezett isSolid() függvényt, igazat ad vissza

#### 3.2.1.43. Utils

#### Felelősségek

Fájlkezelő osztály, melynek statikus függvényei mindenhol elérhetők.

#### Attribútumok

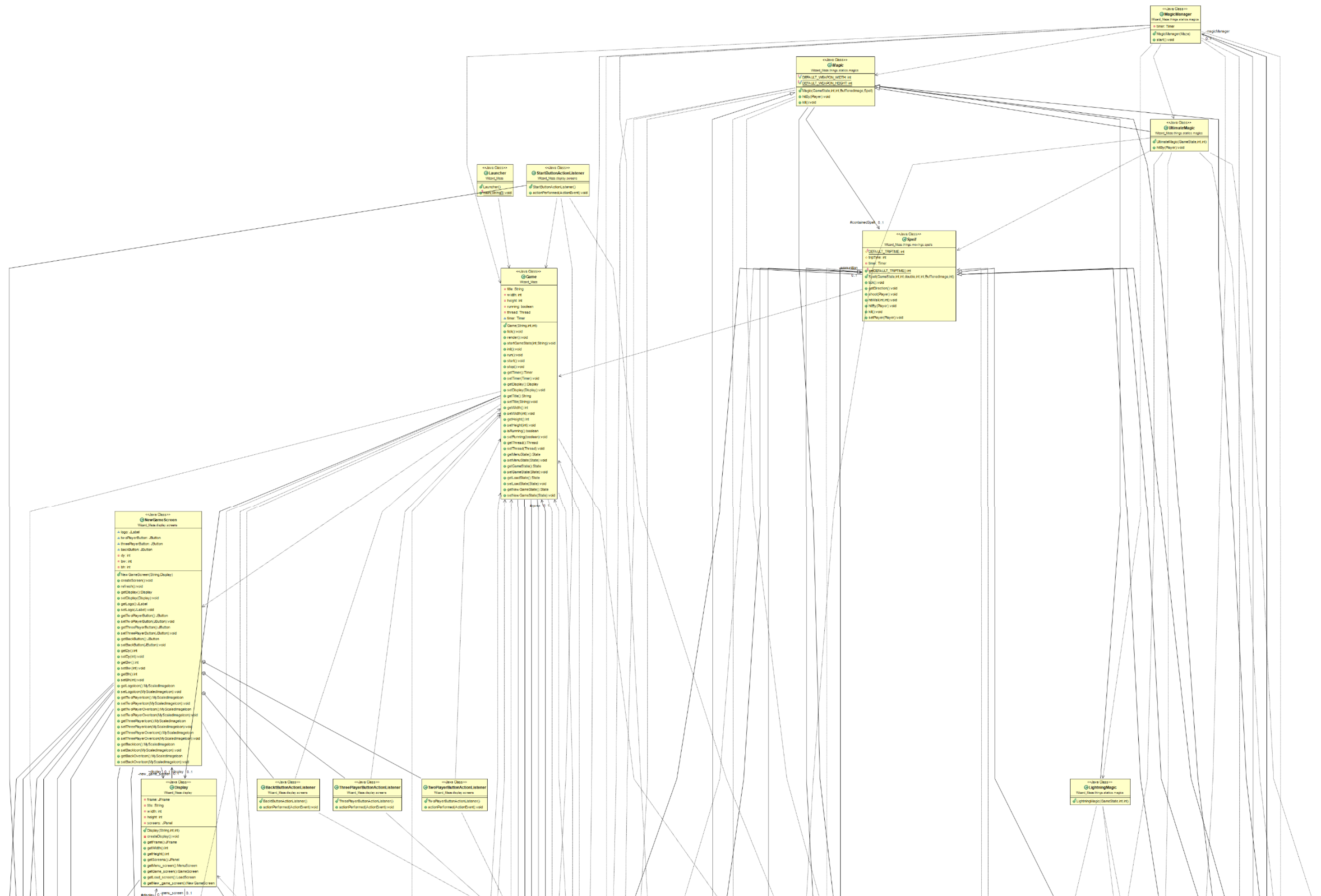
-

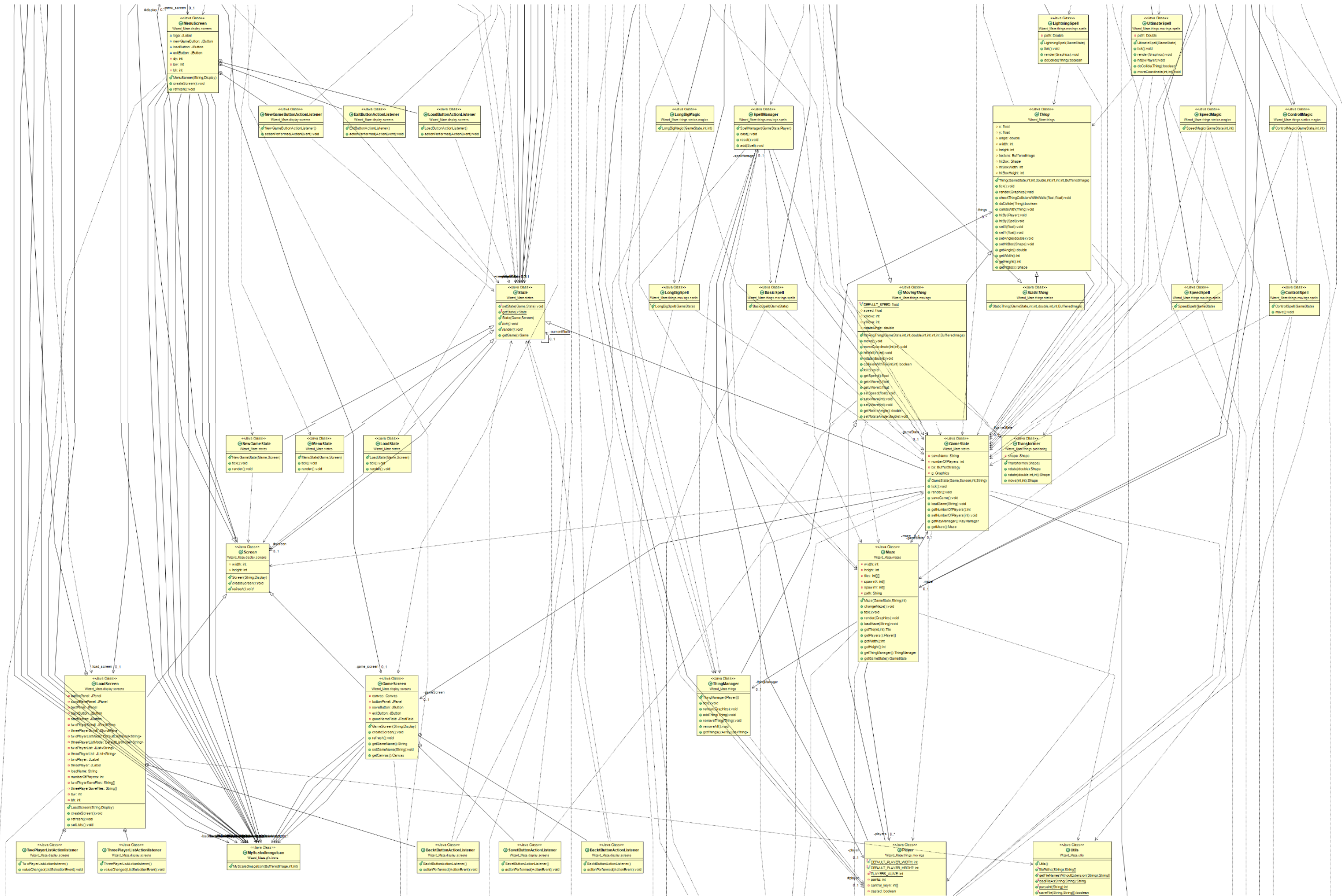
#### Metódusok

<u>+filePaths(String dir_path):String[]</u>	Visszaadja az adott mappában található file-ok neveit
<u>+getFileNamesWithoutExtension(String dir_path):String[]</u>	Visszaadja az adott mappában található file-ok neveit kiterjesztés nélkül
<u>+loadFileAsString(String path):String</u>	Betölti az adott file tartalmát egy stringbe és azt adja vissza
<u>+parseInt(String number):int</u>	Egész számmá konvertál egy stringet, ha tud
<u>+ saveFile(String path, String[] contentInStringArray)</u>	Lement egy String tömbben megkapott adatot a megkapott elérési úton lévő fájlba. Ha nem valid az elérési út, kivételt dob



### 3.2.2.Osztálydiagram









## 4. Fájlkezelés

Két fajta file kezelés van a programban, mindkettő rendkívül egyszerű, így egyszerű txt-be mentést alkalmaztam, ahol whitespacek választják el az adatokat. Az egyik a pályák mentése, ennek formátuma a következő:

<width>            <height>

<Player1X>        <Player1Y>

<Player2X>        <Player2Y>

<Player3X>        <Player3Y>

<width\*height-os mátrix 0-kból és 1-ekből, ahol 0 a floor, 1 a wall>

A másik pedig a pontok mentése, ez egyszerűen:

<Player1 point>            <Player2 point>            <Player3 point>

## 5. Fejlesztési lehetőségek, gondolatok

Bár nagyon sok munkám van ebben a játékban, látom, hogy bőven lehet még hova fejleszteni. Itt főleg a struktúráltságra gondolok, az egyes működési egységek csoportosítását lehet optimalizálni, osztályokat újra csoportosítani, összeolvasztani. Rendkívül sok lett az osztály a végére, és ezek közül talán sok indokolatlan a program ezen fázisában. A továbbfejlesztés esetén derülne ki igaztán, hogy van-e ezekre szükség.

Sok szempontból lehet még működésileg fejleszteni a játékot, például finomabb ütközés detektálás, újabb fegyverek, random player spawnolások, több labirintus, hatékonyabb kirajzolás (pl csak a mozgó dolgok legyenek újrarajzolgatva), csak hogy párat említsek.

## 6. Források

- Játéktér, alap mechanizmusok:  
<https://www.youtube.com/playlist?list=PLah6faXAgguMnTBs3JnEJY0shAc18XYQZ>
- Kép forgatás:  
<https://stackoverflow.com/questions/26828438/how-to-correctly-rotate-my-players-java>
- Terület forgatása ütközés detektáláshoz:  
<https://stackoverflow.com/questions/5920638/java-collision-detection-for-rotated-rectangles>  
<https://www.codota.com/code/java/methods/java.awt.geom.Area/intersect>
- JList update  
[https://www.youtube.com/watch?v=LHTeJu6Bo-s&ab\\_channel=StudyViral](https://www.youtube.com/watch?v=LHTeJu6Bo-s&ab_channel=StudyViral)