

# Tartalomjegyzék

## Fejlesztői dokumentáció

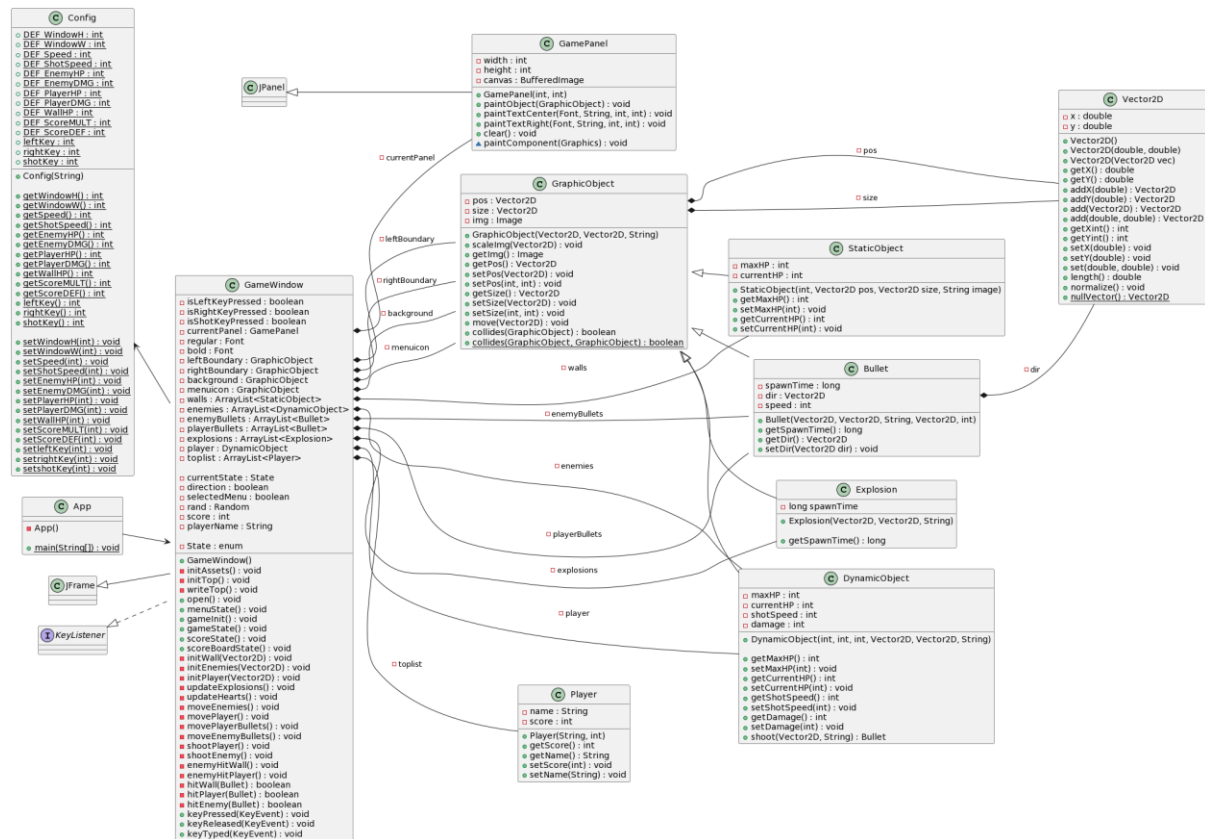
Osztálydiagrammok.....	2
Primitív osztályok .....	2
spaceinvaders.objects.primitives.Config.....	2
spaceinvaders.objects.primitives.Vector2D .....	2
spaceinvaders.objects.primitives.Player .....	3
spaceinvaders.objects.primitives.GraphicObject.....	3
Játékelemek osztályai.....	3
spaceinvaders.objects.Bullet.....	3
spaceinvaders.objects.DynamicObject .....	3
spaceinvaders.objects.StaticObject .....	3
spaceinvaders.objects.Explosion .....	4
spaceinvaders.GamePanel .....	4
spaceinvaders.GameWindow.....	4
Adattagok .....	4
Konstruktor .....	5
open() .....	5
Állapotok függvényei.....	5
Inicializáló függvények .....	5
Játékállapot frissítésére használt függvények .....	6
Ütközés ellenőrzés .....	6
KeyListener felülírt függvények.....	6

## Felhasználói dokumentáció

Program indítása előtt.....	7
Kezdőképernyő.....	7
Játékmenet.....	8
Játék vége.....	8

# Fejlesztői dokumentáció

## Osztálydiagram



## Primitív osztályok

### spaceinvaders.objects.primitives.Config

A játék beállításainak tárolásához használt osztály globális adattagokkal. Többek között: Képernyő méret, játék sebessége, lövési sebesség, ellenfelek életerejé, sebzése, játékos életerejé, sebzése, védőfalak életerejé, kezdőpont, pont szorzó, irányítás gombjai.

#### Függvények:

- Getterek/Setterek

### spaceinvaders.objects.primitives.Vector2D

2 dimenziós vektorok tárolásához, vektorműveletek végzéséhez használt osztály.

#### Függvények:

- Getterek/Setterek
- add(Vector2D), addX(double), addY(double): Hozzáadja a paraméterként kapott értéket az adott adattag(ok)hoz.
- normalize(): Normalizálja a vektort.
- nullVector(): Visszatér (0,0) értékű vektorral.
- length(): Visszatér a vektor hosszával.

### [spaceinvaders.objects.primitives.Player](#)

A toplista által használt osztály, amely a játékosok neveit és elért pontszámai tárolását segíti.

Függvények:

- Getterek/Setterek

### [spaceinvaders.objects.primitives.GraphicObject](#)

A program grafikai elemeinek megjelenítésére használt alaposztály. Tárol egy képet (Image), a kép pozícióját (Vector2D), illetve méretét (Vector2D).

Függvények:

- Getterek/Setterek
- `scaleImage(Vector2D)`: Méretezi a tárolt képet a paraméterként kapott méretre.
- `move(Vector2D)`: Mozdítja a kép pozícióját a paraméterként kapott vektorral.
- `collides(GraphicObject)`: Igaz/Hamis értékkel tér vissza. Ellenőrzi, hogy a paraméterként kapott GraphicObject átfedésben van az adott objektummal.
- `collides(GraphicObject, GraphicObject)`: Megnézi, hogy a paraméterként kapott két objektum átfedésben van-e.

## Játékelemek osztályai

### [spaceinvaders.objects.Bullet](#)

A GraphicObject osztály leszármazottja. Tárolja az alap adattagokon felül a létrehozásának idejét (long), a lövedék irányát (Vector2D) és sebességét (int). Az objektum inicializálásakor a program az aktuális rendszeridőt állítja be az objektum létrehozási idejének, illetve normalizálja az irány vektorát.

Függvények:

- Getterek/Setterek

### [spaceinvaders.objects.DynamicObject](#)

A játék interaktív elemeinek osztálya, a GraphicObject osztály leszármazottja. Tárolja az adott játékelem maximális, illetve jelenlegi életerejét (int), lövési sebességét (int), sebzését (int). Ez az osztály valósítja meg az ellenfeleket, illetve a játékos által irányított játékelemet.

Függvények:

- Getterek/Setterek
- `shoot(Vector2D, String)`: Az adott objektum közepén (pozíció + méretének fele) az iránytól függően létrehoz egy Bullet objektumot a paraméterként kapott iránnyal és textúrával, majd visszatér vele.

### [spaceinvaders.objects.StaticObject](#)

A játék statikus elemeit megvalósító osztály, a GraphicObject osztály leszármazottja. Ez az osztály valósítja meg a védőfalakat, tárolja azoknak maximális, illetve aktuális életerejét (int).

Függvények:

- Getterek/Setterek

## spaceinvaders.objects.Explosion

A játékelemek találatakor megjelenített effektusok megvalósításához használt osztály, a GraphicObject osztály leszármazottja. Tárolja az aktuális találat időpontját (long), amit az inicializáláskor beállít az aktuális rendszeridőre.

Függvények:

- Getterek/Setterek

## spaceinvaders.GamePanel

A JPanel osztály leszármazottja. Ez az osztály felelős a játékelemek képernyőre való kirajzolásáért. Ebben az osztályban vannak megvalósítva az alacsonyszintű grafikai rutinok. Tárolja az adott ablakon kirajzolt felület szélességét és magasságát (int) illetve egy BufferedImage változóban a megjelenítendő képkockát.

Az objektum inicializálásakor beállítja a kép hátterét feketére és a szélességét és magasságát a paraméterek alapján.

Függvények:

- paintObject(GraphicObject): Megjeleníti a képen a paraméterként kapott objektumot.
- paintText(Font, String, int, int): Megjeleníti a képen a paraméterként kapott karakterláncot a kapott betűtípussal az adott pozícióra bal oldalra igazítva.
- paintTextCenter(Font, String, int, int): Megjeleníti a képen a paraméterként kapott karakterláncot a kapott betűtípussal az adott pozícióra középre igazítva.
- paintTextRight(Font, String, int, int): Megjeleníti a képen a paraméterként kapott karakterláncot a kapott betűtípussal az adott pozícióra jobb oldalra igazítva.
- clear(): Kitölti a képet feketére.

## spaceinvaders.GameWindow

A program lényeges része ebben az osztályban van megvalósítva. Ez az osztály örökli a JFrame osztályt és implementálja a KeyListener interfészt a felhasználói irányítás megvalósításához.

### Adattagok

Alapértelmezetten a KeyListener interfész felülírt függvényei folytonosan meghívódnak minden gomb eseményre, így az osztályon belüli egyéb függvények folytonos meghívásának elkerülése érdekében 3 adattagban tárolja a program a játék alatt használt gombok éppen aktuális állapotát. (isLeftKeyPressed, isRightKeyPressed, isShotKeyPressed) Ezek a változók igaz állapotra váltanak amennyiben az aktuális gomb le van nyomva, illetve hamissá, ha a felhasználó elengedi ezeket a gombokat.

Az osztály tárolja az aktuális képkockát a currentPanel (GamePanel) adattagban, illetve a program által használt szövegek egy címsorokhoz (bold) illetve általános szövegekhez (regular) használt betűtípusokat. Emellett szintűgy tárolja a háttér, illetve menü ikon (GraphicObject) objektumokat, hogy ezeknek a fájlból való betöltése ne fogyasszon erőforrást minden képkocka betöltése alatt.

Szintűgy letárolja a képkockán megjelenített játékelemeket. Egy-egy listában tárolja a: walls, enemies, enemyBullets, playerBullets, explosions objektumokat, illetve szeparálva a játékos által irányított elemet. Ezeknek az itt való tárolása egyszerűsíti az objektumok kezeléseit, hiszen az ablak által függvények mind privát láthatóságúak, így ezeknek a paraméterként való átadása csak bonyolítaná a programot.

Egyéb adattagok:

- topList: a toplistában tárolt játékosok nevei és pontjai.
- currentState: a program működéséhez használt állapotgép éppen aktuális állapota. (Enum: { MENU, GAMEINIT, GAME, SCORE, SCOREBOARD })
- direction: Az ellenfelek aktuális irányát tárolja.
- selectedMenu: A menüben éppen aktuálisan kijelölt menüpontot tárolja.
- rand: Véletlenszerű ellenfél lövéshez szükséges.
- score: Játékos aktuális pontszáma.
- playerName: Játékos neve.

### Konstruktor

Az ablak inicializálásakor betölti az egyedi betűtípusokat, illetve a folyton újra használt elemeket (background, menuicon, boundaries). Ezen felül beállítja az ablak nevét, ikonját, méretét.

Mivel a toplistában szereplő játékosok adatai az állapotgép több állapotában is szükséges, hogy meg legyen jelenítve, így ezt szintúgy itt tölti be a program.

### open()

Ez az osztály egy külsőleg használt publikus láthatóságú függvénye. Ebben található meg az állapotgép, ami a konfigurációs fájlban megadott időközönként frissül.

### Állapotok függvényei

A felhasználói dokumentációban látható az állapotok pontos kinézete, ezek a függvények csak az egyéb függvények meghívására, illetve a játékelemek megjelenítésére lettek létrehozva.

- menuState()
- gameInit()
- gameState()
- scoreState()
- scoreBoardState()

### Inicializáló függvények

A gameInit() függvény által meghívott függvények, a program ezekben tölti be a játékelemeket a paraméterként megadott pozíciókra.

- initWalls()
- initEnemies()
- initPlayer()

## Játékállapot frissítésére használt függvények

A `gameState()` függvény által meghívott függvények. Ezek frissítik a játékelemek aktuális állapotait.

- `updateExplosion()`: Megjeleníti az eltalált objektumok által hagyott robbanásokat.
- `updateHearts()`: Megjeleníti a képernyő jobb alsó sarkában a játékos éppen aktuális életerejét.
- `moveEnemies()`: Frissíti az ellenfelek aktuális pozícióját. Mozgatja őket balra/jobbra, illetve lefelé amennyiben elérték a `leftBoundry` vagy `rightBoundry` objektumokat.
- `movePlayer()`: A felhasználó által lenyomott gombok alapján frissíti a játékos pozícióját.
- `movePlayerBullets()`, `moveEnemyBullets()`: Frissíti a lövedékek pozícióját, illetve meghívja az ütközés detekciós függvényeket.
- `shootPlayer()`: Kezeli, hogy a játékos éppen lőhet-e vagy sem, illetve létrehozza a játékos által lőtt új lövedékeket.
- `shootEnemy()`: Az ellenfelek sebessége szerint véletlenszerűen létrehoz lövedékeket a játékos irányába.

## Ütközés ellenőrzés

- `enemyHitWall()`, `enemyHitPlayer()`: Detektálja, hogy az ellenfelek mely más játékelemekkel ütköznek.
- `hitWall(Bullet)`: Csökkenti az életerejét vagy törli az eltalált falakat. Ezen felül létrehoz egy új robbanás effektust.
- `hitEnemy(Bullet)`: Csökkenti az életerejét vagy törli az eltalált ellenfeleket. Ezen felül létrehoz egy új robbanás effektust.
- `hitPlayer(Bullet)`: Csökkenti az életerejét vagy tovább lépteti a játékost. Ezen felül létrehoz egy új robbanás effektust.

## KeyListener felülírt függvények

Ezekben a függvényekben találhatóak az irányítás által használt igaz/hamis változók frissítései. Ezen felül a menükben használt egyéb gombokat ezekben a függvényekben érzékeli a program.

# Felhasználói dokumentáció

## Program indítása előtt

A program mellett található egy config.json fájl, amelyben megtalálhatóak a program beállításai. Amennyiben az alapértelmezettől el szeretnénk térni, ebben tudjuk módosítani az irányítást és az ellenfelek, valamint a játékos egy-egy aspektusát.

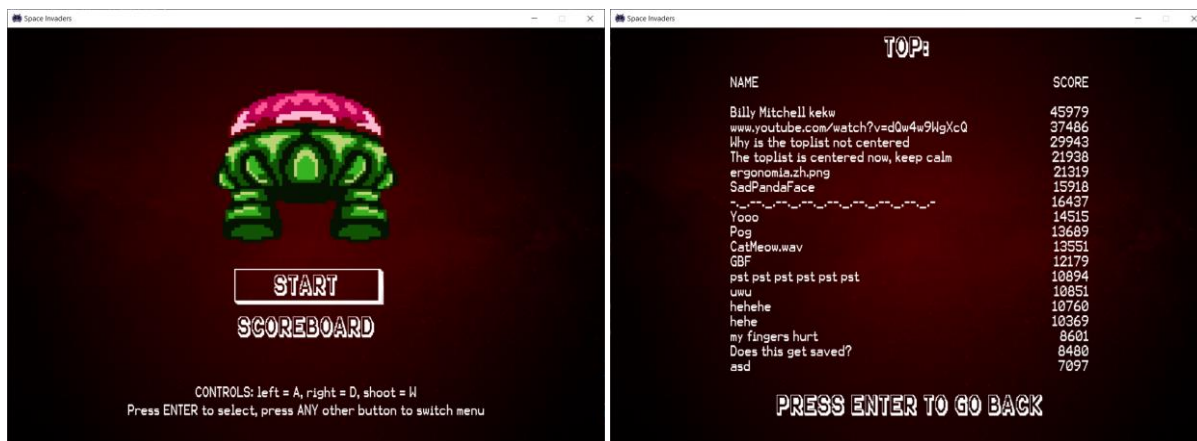
A fájlban pontosan az alábbi beállítások állíthatók be:

- Irányító gombok (bal, jobb, lövés)
- Játék sebessége (másodpercenként hány képkockát jelenít meg a játék)
- Lövedékek sebessége
- Ellenfelek életereje és sebzése
- Játékos életereje és sebzése
- Védőfalak életereje
- Kezdő pont
- Pont szorzó

# Kezdőképernyő

A játék indításakor a bal oldali menüt láthatja a felhasználó. Az irányítás gombjai a képernyő alján vannak kiírva.

A START menüpontot választva a program elindítja a játékot, a SCOREBOARD menüpontot választva a korábbi játék eredményeket listázza ki a program.



## Játékmenet

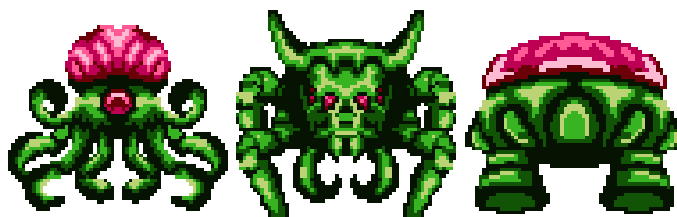
A játék során a játékos célja, hogy a képernyő tetején lévő összes ellenfelet eltalálja emellett kitérjen az ellenfél lövései elől. A játékos pontja (SCORE) a kezdőbeállításokon felül minden egyes ellenfél kiiktatása után növekszik. Emellett ez a pont a játék alatt az idővel folyamatosan csökken, így fontos, hogy minél gyorsabban találja el az ellenfeleket a játékos.

Az ellenfelek a játék során jobbra és ballra mozognak míg el nem ér a legszélső ellenfél az ablak szélére. Amikor ez megtörténik visszafordulnak a másik irányba és lejjebb mennek a játékos felé. A mozgás mellett az ellenfelek szintúgy vissza lőnek a játékosra, amelyek leviszik a játékos életerejét. Amennyiben a játékos életereje elfogy, a játéknak vége. Az ellenfelek lövedékei elől való kitérésben segítségként vannak védőfalak, amelyek megvédik a játékos. Ezeket minden egyes eltalált lövedék sebzi, attól függetlenül, hogy azokat az ellenfelek vagy a játékos lőtte. Amennyiben elegendő lövedék találta el a védőfalakat, azok eltűnnek.

Amennyiben az ellenfelek ütköznek a falakkal, a falak életerejét ignorálják, és azok egyből eltűnnek. Szintúgy, ha az ellenfelek elérik a játékost a játék véget ér, bármennyi életereje is marad a játékosnak.

A különböző ellenfelek különböző tulajdonsággal rendelkeznek:

- **Octo:** Az alapértelmezettnél lényegesen gyorsabban lő. (4x az alap lövési sebességnek)
- **Spidey:** Az alapértelmezett életerőhöz képest kétszeres mennyiségű életereje van.
- **Ufo:** Az alapértelmezetthez képes kétszerese az életereje és a lövési sebessége.



## Játék vége

Amennyiben a játékos kiiktatta az összes ellenfelet, vagy az ellenfelek tették meg ezt a játékkal a játék véget ér. A játék vége után a program kiírja a korábbi legjobb 10 eredményt, valamint bekéri az aktuális játékos nevét. Amennyiben a játékos ezt a mezőt üresen hagyja, a program „Anon” néven fogja elmenteni az adott játék eredményét a toplistába.

