

# Dokumentáció: Game of Life

Készítette: Csupor Ákos EYJ8PS

**Feladat ismertetése:** Az életjátékot (angolul: The game of life) John Horton Conway, a Cambridge-i Egyetem matematikusa találta ki. Játékként való megnevezése megtévesztő lehet, mivel „nullszemélyes” játék; és a „játékos” szerepe mindössze annyi, hogy megad egy kezdőalakzatot, és azután csak figyeli az eredményt. Matematikai szempontból az ún. sejtautomaták közé tartozik. A négyzetrács mezőit celláknak, a korongokat sejteknek nevezzük. Egy cella környezete a hozzá legközelebb eső 8 mező (tehát a cellához képest „átlósan” elhelyezkedő cellákat is figyelembe vesszük, feltesszük hogy a négyzetrácsnak nincs széle). Egy sejt/cella szomszédjai a környezetében lévő sejtek. A játék körökre osztott, a kezdő állapotban tetszőleges számú (egy vagy több) cellába sejteket helyezünk. Ezt követően a játékosnak nincs beleszólása a játékmenetbe. Egy sejttel (cellával) egy körben a következő három dolog történhet: A sejt túléli a kört, ha két vagy három szomszédja van. A sejt elpusztul, ha kettőnél kevesebb (elszigetelődés), vagy háromnál több (túlnépesedés) szomszédja van. Új sejt születik minden olyan cellában, melynek környezetében pontosan három sejt található. Fontos, hogy a változások csak a kör végén következnek be, tehát az „elhalálozók” nem akadályozzák a születést és a túlélést (legalábbis az adott körben), és a születések nem mentik meg az „elhalálozókat”.

## Megvalósítás:

Awt és Swing osztályok segítségével lett létrehozva. Dimension és Point osztályt is felhasználva, hogy ne kelljen külön alkotni hasonló osztályokat. Lett 4 osztály, egyik felelős az ablak kinézetéért és a menüért a másik a táblán megjelenő dolgokért és az adatok tárolásáért, harmadik az egér általi interakciókért, negyedik pedig a fájlba mentésben segít.

### Osztályok:

*public class ConwayGameOfLife extends JFrame implements ActionListener: A menüt és ablakot valósítja meg.*

public ConwayGameOfLife(): A konstruktorban felépül a menü, bekötődnek a Listenerek és megjelenik az ablak.

public void setGameBeingPlayed(boolean isBeingPlayed): A kapott érték értelmében, vagy létrehoz és elindít egy szálat ezzel elindítva a játékot, vagy interrupttal megállítja a szálat és megáll az automata.

public void actionPerformed(ActionEvent ae): A menü műveleteit indítja el.

*public class GameBoard extends JPanel implements Runnable: A lényeges automatában zajló művelet benne valósul meg.*

public void load() throws IOException, ClassNotFoundException: elinítja az adatok beolvasását fájlból majd frissíti a táblát és megjeleníti azt. public void save() throws IOException: Elindítja az adatok szerializálását ezzel fájlba kiírását.

public void set\_d\_gameBoardSize(Dimension dim): Beállítja a kapott dimenzió alapján a tábla dimenzió paraméterét majd elindítja a tábla frissítését.

private void updateArraySize() Megvalósítja a tábla frissítését, ennek során a paramétereken kívül eső cellákat törli majd újra festi a táblát. public void addPoint(int x, int y): Új cellát tárol el a kapott koordináták alapján.

public void addPoint(MouseEvent me): Új cellát ad a táblához egér mozgása alapján. Ha már léteznek törli őket. public void removePoint(int x, int y): Koordináták alapján eltávolítja a cellát.

public void resetBoard(): Törli az összes cellát a tábláról.

public void paintComponent(Graphics g): Kirajzolja a négyzetrácsot és a cellákat, mindent ami megjelenik a táblán.

public void run(): Megvalósítja a szál futásához szükséges függvényt ebben ellenőrzi a szabályok alapján melyik mező lesz életben a következő ciklusban. majd kirajzolja és kezdi előlről.

*public class Backup: A fájlba mentést valósítja meg.* public Backup(Object

mentenivalo): Konstruktor tárolja a menthető Objectumot. public void save() throws IOException: Szerializálással menti a dokumentumot.

public Object load() throws IOException, ClassNotFoundException: Beolvassa fileból az objectumot majd visszatérési értékben meg is adja.

*public class Listeners extends GameBoard implements ComponentListener, MouseListener, MouseMotionListener: Megvalósítja az ablak méretváltozását és az egérrel kapcsolatos műveleteket.* public Listeners(): Konstruktor minden implementálandó Listenert hozzáad.

public void componentResized(ComponentEvent e): Ablak méretének változtatása esetén beállítja az új dimenziókat.

public void mouseReleased(MouseEvent e): egér elengedése esetén meghívja az addPoint függvényt.

public void mouseDragged(MouseEvent e): Egér húzása esetén meghívja az addPoint függvényt.

## Osztály diagram:

