

1. Feladat rövid ismertetése

A sejtautomaták úgy működnek, hogy egy 2D rács minden mezője egy adott állapotban van. Az **életjáték** a legismertebb sejtautomata. Minden cella két állapotban lehet (élő/üres), és a következő iterációban azok a cellák maradnak életben, amelyeknek 2 vagy 3 élő szomszédja van, és azok az üres cellákban kelnek életre, amelyeknek pontosan 3 szomszédja van (Moore-környezetben, tehát a környező 8 cellából négyzetrács esetében). A programot úgy kell megírni, hogy a szabályokat a kezelőfelületen meg lehessen adni: ennek a szokásos jelölése B.../S... (born / survive), pl. a Conway-féle klasszikus Game of Life kódja B3/S23, de pl. a David Bell-féle Day and Night kódja B3678/S34678. Vannak még egyéb érdekes szabályok, amik a programban szerepelni fognak. Az automata három különböző rácson futtatható majd hatszög, négyszög és háromszög alapún. Kiválasztható funkció lesz még hogy a sejt az élő állapotból üresbe (világít/nem világít) nem rögtön váltással, hanem elhalványulással menjen.

2. Felhasználói útmutató

A programot elindítva a felhasználót egy egyszerű menü fogadja, itt eldöntheti majd, hogy új játékot szeretne-e létrehozni, egy már megléptöt betölteni, vagy kilépni. Ha az új játékot választja, akkor kiválaszthatja, hogy milyen rácson szeretné futtatni az automatát, a fentebb említett 3 közül. Miután kiválasztott egy rács típust, a felhasználó megadhatja, hogy milyen szabályokkal szeretné elindítani a szimulációt. Minden rácsnál ajánl a program pár neves szabály típust, amik érdekes mintákat eredményezhetnek. Ezek után a kiválasztott rács megjelenik, a szimuláció állni fog és a felhasználó kedve szerint berajzolhat egy kezdőpozíciót, a megfelelő cellára kattintással változtatva annak állapotát. Ha elégedett, akkor egy gombnyomással elindíthatja a szimulációt, megállíthatja, illetve lesz lehetősége a sebesség állítására is. Itt még van lehetőség a játék állásának mentésére és a főmenübe való visszalépésre. A főmenüben meglévő játék betöltésekor a program betölti a megfelelő pályát, szabályokat és kezdőállapotot, lényegében onnan folytatva a szimulációt, ahogy azt a felhasználó lementette. Kilépés választásakor a program kilép.

3. Megoldási ötletek.

Swing UI-t fogok használni a megjelenítésre.

A pályák lényegében cellák 2 dimenziós rácsba rendszerezve, a hatszög és háromszög esetében is működőképes ez, csak a megjelenítésnél bizonyos sorokat el kell tolni, és ugye oda kell figyelni, hogy egy négyszögnek 8, hatszögnek 6, háromszögnek 3 szomszédja van összesen. Ezért a Rács lehet

absztrakt osztály és a különböző alapú rácsok ennek a „véglegesített” osztályai.
A Rács fix méretű lesz. A szimuláció állásának mentésére a Serializable interfészt fogom megvalósítani.