

HÁZI FELADAT SPECIFIKÁCIÓ (EOPKKH)

Ismertető:

A programom a "turmesz" nevű, 2D Turing-gépen alapuló, sejtautomatát valósítja meg felhasználói felülettel kiegészítve. A felhasználó kiválaszthatja majd, hogy milyen jellegű animációt futtasson a program pl. szimmetrikus/véletlenszerű. Ezentúl beállíthatja az animáció sebességet, leállíthatja/újraindíthatja azt, és el is mentheti egy adott animáció állapotát, majd később újra futathatja azt. Az animációt a program egy 2D-os rácson fogja megjeleníteni, melynek minden cellája két állapotot vehet fel. Az állapottól függ a cellák színe, pl. élő-fehér (1-es), halott-fekete (0-ás), és ahogy fut a program, változik az animáció (változnak a cellák színei) az alábbi folyamat alapján:

Van egy aktuális pozíció és egy irány, ami reprezentálható egy természettel.

Minden iterációban a következőt csinálja a termes:

1. 90 fok többszörössel fordul egyet, annak függvényében, hogy milyen jellegű animációt állított be a felhasználó. Pl.: Egy szimmetrikus animáció esetén mindig csak 90 fokkal fordul.
2. Az aktuális cella értékét (állapotát) változtatja (0 vagy 1). Pl.: Véletlenszerű animáció esetén vegyesen tartalmaznak a cellák 0-kat és 1-eseket, míg szimmetrikus animáció esetén a termes által érintett cellák csak fehér színűek (1-es állapot).
3. Előre lép egyet.
4. Átmegy egy másik állapotba a program elején egy fájlból beolvasott szabálynak megfelelően.

Use-case-ek:

- A program gombokon keresztül kezelhető (elindítás/leállítás/újraindítás).
- A program sebességét és az animáció jellegét egy legördülő listából lehet majd kiválasztani a start nézetben.
- A animáció elindítása majd megállítása után megjelenik egy mentés gomb is, ahol egy adott névvel elmenthető az aktuális animáció.
- A start nézetben a mentett animációk is kiválaszthatóak nevük alapján.

Megvalósítás:

Az alkalmazás megvalósításához a SWING GUI készletet fogom alkalmazni.

Kettő nézettel fog rendelkezni a program: lesz egy menü nézet (JFrame), ahol a felhasználó kiválaszthatja egy-egy legördülő listából az animáció sebességét és jellegét (JComboBox). Ezentúl egy harmadik listából, adott esetben, kiválaszthatja a mentett animációit, és azokat is futathatja. A másik nézetben pedig maga az animáció fog futni, melyet a Graphics osztállyal valósítok meg. Az alkalmazás 4 féle gombot fog tartalmazni: elindítás/megállít/újraindítás, és egy mentés gomb is megjelenik abban az esetben, ha a felhasználó megállítja az éppen futó animációt. A mentés esetén egy JTextField-ben adható meg az aktuális animáció neve. A mentést szerializálással fogom megvalósítani, olyan módon, hogy az animációnak a nevét a name.txt, és az azt megvalósító objektum tartalmát a save.txt írom ki.

A szabályokat, amik leírják, hogy a természet egy adott iterációban a különböző állapotok függvényében mit csináljon (Pl. O-ás állapotban O-ás mezőre lép akkor mit csináljon) a rule.xml fájlból olvasom be, mikor létrehoz a felhasználó egy új animációt.

A program osztályainak metódusait a JUnit-al fogom tesztelni.

A felhasználói kézikönyv:

Az alkalmazás elindulás után a start menü fogadja a felhasználót. Ezután kettőféle lehetőség van:

1. Egy új animáció futtatása. Ehhez a Saved Animations résznél a New Animation-t kell kiválasztani, majd a Speed résznél az animáció sebessége, míg az Animation Type résznél a jellege állítható be. Miután elindult az animáció megjelenik egy új ablak, ahol fut az animáció. Először 2 gomb található az ablak alján, amivel leállítható/újraindítható az animáció. Ha megállítjuk az animációt akkor megjelenik egy continue, és egy save gomb plusz egy szövegdoboz, amelyek segítségével folytatható/elmenthető az animáció. Amennyiben nem adunk meg nevet a mentés során, default névvel mentődik el az animáció. Fontos, hogy minden animációnak egyedi nevet kell adni. A mentett animáció futtatásához újra kell indítani a programot.
2. Egy mentett animáció futtatása. Ekkor már csak a sebességét változtathatjuk az aktuális animációnak, a típusát nem. A mentés ebben az esetben úgy történik, hogy ha a felhasználó megnyomja a stop gombot az animáció futása közben, akkor egyben ment is a rendszer, nem csak leállítja az animációt.

UML diagramm:

