

# Nagy Házi Specifikáció

Egyszerű Gráf szerkesztő program

## **Alap koncepció (terv):**

Egy szerkesztő programot tervezek csinálni, melyben egyszerű gráfokat lehetne létrehozni grafikusan és ezeken alap algoritmusokat futtatni/megjeleníteni.

## **Részletes leírás:**

A Program 3 fő részből állna:

- Menü
- Szerkesztő
- Szimulátor

### **Menü:**

Három menüpont (gombok):

- Szerkesztő: Ha ezt választjuk, egy mellék menübe lépünk, ahol:
  - Új gráf szerkesztése: Ezt választva egy teljesen új vászon (canvas) nyílik meg, melyen elkezdhetünk szerkeszteni egy gráfot teljesen a 0-ról.
  - Meglévő gráf szerkesztése: Ezt választva felsorolja a már korábban megkezdett, illetve elmentett, gráfjainkat, melyekből választhatunk egyet. Választás után megnyitja a kívánt gráfot, mely szerkesztését ott folytathatjuk, ahol legutóbb elmentettük.
  - Vissza a főmenübe: Visszatérünk a főmenübe
- Szimulátor: Ha ezt választjuk a korábban említett meglévő gráf megnyitásához hasonlóan, itt is megjelennek a mentett gráfjaink és ezek közül kiválaszthatjuk, hogy melyiket nyissuk meg a szimulátorban.
- Kilépés: Kilép a programból.

**Szerkesztő:**

Itt valósul meg a program egyik lényeges működése, maga a szerkesztés. Két fő részből áll: eszköztár és vászon.

Az **eszköztár**, az ablak bal oldalán fog elhelyezkedni. Innen lehet kiválasztani, hogy a felhasználó mit is szeretne a vásznon éppen csinálni:

- Választható eszközök: Egyszerre, csak az egyik lehet aktív.
  - Mozgató eszköz: A csúcspontokat lehet vele mozgatni, illetve ekkor jelenik meg a vásznon 4 szélén 1-1 mozgató nyíl, mellyel az egész vásznat lehet majd mozgatni.
  - Új csúcspont eszköz: Ha ez aktív a vásznon üres helyre kattintva létrehoz egy új csúcsot az egérmutató pontjában. A nevét automatikus állítja be, ezt később nem lehet módosítani, globálisan lesz követve a csúcsok id-ja(neve).
  - Új él/él törlés eszköz: Ha aktív egy meglévő csúcsra kattintva ki lehet jelölni az él első csúcsát. Ha ezután üres helyre történik a második kattintás, akkor visszavonódik a művelet. Ha a második kijelölés még nem történt meg, de az első igen, ezt vizuálisan lehet látni, hisz az első kiválasztott csúcstól egy vonal fog vezetni a kurzorig. A második csúcs sikeres kiválasztása után két lehetőség van:
    - Ha a két kiválasztott csúcs között nincs kapcsolat, a kapcsolat létrejön és innentől össze lesznek, vizuálisan és logikailag, kötve.
    - Ha a két kiválasztott csúcs között már van meglévő kapcsolat, a már meglévő kapcsolat törlődik és innentől köztük nem lesz se logikai se vizuális kapcsolat.
- Név(id)/fokszám választó eszköz: Egy ComboBox melynek két értéke lehet: id és fokszám. Amelyik aktív az jelenik meg a csúcsokban.
- Rendezés eszköz: Egy gomb, amit, ha a felhasználó megnyom, akkor a meglévő csúcsokat rendezi kör alakban. Pontosabban a körbe írható n db (n = csúcsok száma) csúcsú szabályos sokszög csúcsaira.
- Ugrás szimulátor módba: Gomb, amellyel az aktuális gráfunkkal átléphetünk a szimulátorba.

A **vászon**, a maradék helyet tölti ki. Itt történik maga a szerkesztés. A fentiek leírtakat kiegészítve a vásznon, akármelyik választható eszköz aktív, ha a kurzor egy csúcson van, akkor ez grafikusan is látható, tehát más lesz a háttere.

A szerkesztőben egy menüsáv is lesz az ablak tetején:

- File:
  - new: Új gráfot lehet elkezdni, ha nincs mentve a mostani, lesz lehetőség (felugró ablak).
  - open: Be lehet tölteni egy meglévő gráfot. (ha nincs mentve like new)
  - save: Elmenti a gráfot, első mentésnél el is lehet nevezni.
  - rename: Ha megtörtént az első mentés, át lehet nevezni.
  - export: Svg kiterjesztésbe lehet exportálni.
  - quit: A főmenübe lép.

- Edit:
  - undo: Ha történt változás vissza lehet vonni egyet.
  - redo: Ha volt visszavonás, ezt semmissé lehet tenni.

### **Szimulátor „algorithmus futtató”:**

A harmadik pillére a programnak a szimulátor. Ide a felhasználó be tud tölteni, már létrehozott gráfokat. Ezeken különböző algoritmusokat tud leszimulálni.

Két algoritmus lesz implementálva:

- Szélességi keresés egy megadott csúcsból.
- Mélységi keresés egy megadott csúcsból.

Maga a szimulátor hasonlóan néz majd ki, mint a szerkesztő, tehát van eszköztár és vászon. Viszont az eszköztáron más **eszközök** lesznek:

- Algoritmus választó: A két implementált algoritmus közül lehet választani egyet.
- Idő beállító: Egy csúszka hol az animáció sebességét, lehet állítani. Maga az animációról később.
- Start/Reset: Egy gomb, amivel el lehet indítani az animációt, ha minden feltételt teljesítettünk. (kiválasztottuk a kezdő csúcsot, illetve az algoritmust). Ha épp folyik az animáció, ezzel a gombbal lehet leállítani és resetelni az állapotot.
- Ugrás szerkesztő módba: A szerkesztőben lévő „Ugrás szimulátor módba” gomb inverze, tehát a szerkesztőbe visz.

Illetve a vásznon az egérmozgás és annak csúcsokkal való interakciója megtalálható itt is, tehát látszik grafikusán is, ha az egér egy adott csúcs felett van (más háttér). Illetve, mivel mind a két algoritmus igényli, ki lehet választani egy kezdőpontot, ezt egy csúcsra való kattintással lehet megtenni, ekkor a csúcs háttére jelzi, hogy ki lett választva.

Az **animációról**, annyit mondanék, hogy lépésenként zajlik, a lépések közt eltelt időt határozza meg az animáció sebessége. Minden lépésben felvillannak az algoritmus hatására éppen bejárt csúcsok, a bejárást követően a csúcsok felveszik a megfelelő háttér színt (jelezve, hogy már járt itt az algoritmus).

### **Megvalósítás:**

#### **Grafikai megvalósítás:**

A grafikus felületet JavaFX-el tervezem megoldani és ebben a könyvtárban lévő komponensek használatával.

**Adat tárolás és fájl kezelés:**

Eddig, ahogy tervezem a lényeges adatok tárolásához a következő osztályokra lesz szükségem:

- Csúcs: Egy csúcs adatait tárolja: pozíció, szomszédok, háttérszín stb.
- Él: Egy él adatait tárolja: két csúcs, amik között fut stb. (A logikai kapcsolathoz nem szükséges, de egyszerűbb kirajzolás érdekében szerintem tárolom az éleket, így a kirajzolásnál a duplikáció megoldódik).
- Gráf: Csúcsokat és éleket tartalmazó osztály, benne minden szükséges metódus.

Fájlkezeléshez a szerializálást fogom alkalmazni, a fent említett osztályok meg fogják valósítani a Serializable interfészt. Az egyetlen statikus tagot visszaolvasnál ki lehet számolni, tehát nem lesz gond azzal se, hogy ő nem lesz sorosítva.

A gráfok fájlljai a „save” mappába lesznek majd „.dat” kiterjesztéssel. Az exportált gráfokat „.svg” kiterjesztéssel, lehet majd megtalálni, a „final” mappában.