

HÁZI FELADAT

Programozás alapjai 3.

Feladatválasztás/feladatspecifikáció

GraphMaster

Kurcsi Norbert Y3ZTEI

-2022. október 30

Tartalomjegyzék

1. Feladatspecifikáció.....	2
1.1. A feladat témája és fő funkciói	2
1.2. A kezdőlap.....	2
1.3. Új gráf létrehozása	2
1.4. Már létrehozott gráf betöltése	2
1.5. Gráf különböző ábrázolási formái.....	2
1.6. Gráf bejárásai.....	2
1.7. Legrövidebb út	3
1.8. Gráf elmentése.....	3
2. Megoldási ötlet	3
2.1. Grafikus felület megvalósítása	3
2.2. Fájlformátum, szerializálás.....	3

1.Feladatspecifikáció

1.1. A feladat témája és fő funkciói

A program egy felhasználói felületet biztosít arra, hogy könnyedén és hatékonyan kirajzoljunk irányítatlan vagy akár irányított gráfokat. A gráf adott éleinek megszabhatunk súlyokat (két adott pont közötti távolság) vagy akár minden él hosszát egységesnek vehetünk. Miután gráfunk elkészült, azt el tudjuk menteni fájlba, hogy később esetleg újra betöltsük azt, vagy különböző algoritmusokat (pl.: szélességi-, mélységibejárás, Dijkstra algoritmus) futtathatunk rajta, közben a program ezt grafikusan szemléltetve.

1.2. A kezdőlap

A program indításakor a kezdőlap fog megjelenni. Itt egy rövid gráf ismertetőt tudunk olvasni arról, hogy a gráfok mennyire is fontosok napjainkban és mire is lehet őket használni. Emellett az ismertető mellett, egy menü is látható. A menüpontokat a kurzor segítségével tudjuk kiválasztani. A főoldalon a menü három részből áll, az első "Új gráf létrehozása" gombbal egy új gráfot tud kirajzolni a felhasználó. A második, "Gráf betöltése", menüponttal egy azelőtt már elmentett gráfot tud a felhasználó betölteni és azon változtatni vagy algoritmusokat futtatni. A harmadik menüpont a program leállítására szolgál.

1.3. Új gráf létrehozása

Ha a felhasználó ezt a menüpontot választotta akkor a program megkérdezi tőle, hogy irányított vagy irányítatlan gráfot szeretne létrehozni. Ezt kiválasztva egy olyan szerkesztői felület fogja várni a felhasználót, ahol egy bizonyos "+" gombra kattintva egy új csomópontot lehet hozzáadni a gráfhoz. Ezek a pontok automatikusan lesznek sor számozva a program által. A pontokat mozgatni is lehet csakugyan a kurzor segítségével. A csomópontok közé úgy lehet behelyezni egy élet, hogy a kezdőpont valamiféleképpen kijelölt tartományára rákattintva, odahúzzuk a kurzort a végpontra, majd ott elengedjük. Ha egy csomópontot úgy mozgat a felhasználó, hogy már tartoznak hozzá élek, akkor az adott élek is követik a pont helyzetét. Az éleket meg a pontokat törölni is lehet, ekkor az adott ponthoz tartozó élek is törlődni fognak. A felhasználó felületen van egy szerkesztés kikapcsolása gomb, ezt megnyomva lehetőség nyílik a gráfhoz tartozó ábrázolások, algoritmusok megtekintésére, futtatására.

1.4. Már létrehozott gráf betöltése

A program lehetőséget nyújt arra is, hogy egy, már régebben létrehozott és elmentett gráfot újra betöltsön. Ha ez az opció kerül választásra, akkor kilistázódnak az ezelőtt elmentett gráfok, és bármelyikre rákattintva az betöltésre kerül, és újra szerkeszthetővé válik, akárcsak mint ha egy újonnan létrehozott gráfot szerkesztenénk.

1.5. Gráf különböző ábrázolási formái

A gráfokat több módon is tudjuk ábrázolni. Ahhoz, egy gráfot anélkül, hogy azelőtt láttuk volna, le tudjuk rajzolni, kell tudnunk, hogy milyen csomópontok között, milyen hosszúságú élek helyezkednek el. A programban négy féle ábrázolási módot (Boole mátrix, Pont-él mátrix, Szomszédsági lista, Éllista) lehet megjeleníteni az éppen kirajzolt gráfra. Ennek az elemnek az a célja, hogy a felhasználó betekintést nyerjen, hogy milyen alternatívák vannak a gráfok tárolására.

1.6. Gráf bejárásai

Kétféle gráfbejárást tud a program szemléltetni: szélességi és mélységi bejárás. A bejárás futtatásakor sorra színeződnek más színre azok a pontok, amelyeket az algoritmus már bejárt. A bejárás végén a képernyőn az is megjelenik, hogy milyen sorrendben lettek a csomópontok feldolgozva. A bejárás indításához a felhasználónak ki kell választania egy kezdőpontot, ahonnan az algoritmus induljon.

1.7. Legrövidebb út

Két pont közötti legrövidebb út meghatározásához a Dijkstra algoritmust használja a program. Az algoritmus indításához ki kell választanunk egy kezdőpontot, ezután a program kiszámítja az összes többi ponthoz a legrövidebb utat.

1.8. Gráf elmentése

Miután sikeresen létrehozta a felhasználó az adott gráfot, a program lehetőséget nyújt arra, hogy ezt elmentse. Ez azért jó, mert hogy ha egy másik alkalommal meg szeretné nézni az adott gráf mélységi bejárását például, akkor nem kell újból létrehozni az összes csomópontot és élet, mert csak egyszerűen betölti a régebben elmentett szerkezetet. Mikor a gráfot el akarja menteni a felhasználó, kell adjon neki egy címet, amely egyedi kell, hogy legyen, tehát nem egyezhet meg bármelyik azelőtt elmentett gráf címével. A program a mentés során elmenti a pontos dátumot és időt is, ahhoz, hogy majd könnyebben meg lehessen találni következő alkalommal a gráfot.

2. Megoldási ötlet

2.1. Grafikus felület megvalósítása

A program grafikus felületét a Java Swing GUI segítségével fogom megvalósítani. Ez az API nagyon sok olyan megoldást biztosít, olyan kihívásokra, amelyekkel a programom fejlesztése során fogok találkozni.

Mivel a tervezett applikációban, nagyrészt a kurzor segítségével fog lehetni majd navigálni, ezért sok gombra lesz szükségem, amelyeket a *JButton* segítségével fogok megvalósítani. Ahhoz, hogy a felhasználó kiválaszthassa, hogy irányított vagy irányítatlan gráfot szeretne létrehozni, egy legördülő lista fog állni rendelkezésére, amelyet *JComboBox*-al fogok megoldani. Amikor egy adott gráf tárolási, vagy ábrázolási módját szeretné a felhasználó megtekinteni, ekkor ez táblázat formában fog megjelenni neki, ezt a táblázatot *JTable* osztály segítségével fogom megvalósítani. A táblázat többek között szerkeszthető is lesz, és a táblázatban végzett szerkesztések alapján a gráf kirajzolása is változni fog. Ahhoz, hogy mentéskor a felhasználó megadja a gráf nevét, illetve a gráf rajzolása során, megadja például egy él hosszát *JTextField* osztály segítségével megvalósított szövegdoboz fog a rendelkezésére állni.

Amikor, egy már korábban létrehozott és elmentett gráfot szeretnénk újra betölteni, egy *JComboBox* segítségével megvalósított legördülő listából lehet majd kiválasztani a létező gráfok közül azt, amelyet be szeretne a felhasználó tölteni.

2.2. Fájlformátum, szerializálás

A létrehozott gráfokat, illeszkedési mátrix segítségével fogom tárolni egy erre a célra tervezett osztályban, valamint ez az osztály implementálni fogja a *Serializable* interfészt, így könnyedén menthető, illetve újból betölthető lesz egy .txt formátumú fájlból a gráf. Az illeszkedési mátrixon kívül, ami azt tárolja, hogy mely csomópontok között van él, illetve, hogy mennyi az adott él súlya, az adott osztály a csomópontok koordinátáit is tárolni fogja. Ez ahhoz szükséges, hogy ha egy gráfot újból betöltünk, akkor az pont olyan elhelyezésbe jelenjen meg, ahogy utoljára a felhasználó elmentette.