JEGYZŐKÖNYV

Komputergrafika és képfeldolgozás

JAVA kétdimenziós grafikai alkalmazás

Készítette: **Mátyás Tibor**

Neptunkód: K4WZH6

Dátum: 2024. május 16

Sárospatak, 2024.

Tartalomjegyzék

1. Bevezetés

Komputergrafika és képfeldolgozás tárgy kapcsán első sorban a JAVA nyelv grafikai lehetőségeit vizsgáltuk a félév során. Az első gyakorlati feladat egy olyan ablak létrehozása volt, amely grafikai elemeket, grafikai primitíveket tartalmaz, képet, címet (grafikai módon előállított szöveget).

A feladat kiírásának megfelelően olyan alkalmazást készítettem, amely a jawax.swing osztály objektumait és metódusait használva lehetővé teszti ablak példányosotokat.

A java.awt osztály segítségével valósítottam meg a grafikai elemek létrehozását és paraméterezését, például a szöveg és kép elhelyezését és formázását az ablakban.

Az awt.geom osztály importálása lehetőséget ad a geometriai alakzatok létrehozására. Segítségével rajzolhatunk kört / ellipszist, négyszöget és egyéb geometriai alakzatokat. Illetve különböző mintákat, vágómaszkot hozhatunk létre, kitölthetjük az alakzatokat színekkel stb.

A képfeldolgozás területén vektorgrafikus és pixelgrafikus szerkesztőprogramokat tekintettünk át, így áttekintettünk olyan programokat mint a CorelDRAW vagy az Adobe Photoshop. Nekem jelenleg az Adobe Photoshop CS4 volt elérhető, azért ezt a programot választottam, hogy egy plakátot elkészítsek, amelyen keresztül be tudom mutatni egy pixelgrafikus képszerkesztő program lehetőségeit. A képpel szembeni elvárásokat a szociál média szempontjai szerint alakítottam ki (képméret, tömörítés mértéke, képarány, tartalom).

2. Tervezés

2.1 Projekt létrehozása

A tervezés során először kiválasztottam a céloknak megfelelő fejlesztőkörnyezetet. Az IntelliJ IDEA Community Edition 2022.2.4 fejlesztőkörnyezetet használtam az első feladat elkészítéséhez, amelyet a félév során még az Eclipse és a NetBeans mellett használtunk JAVA projektek fejlesztéséhez.
Mivel a projekt meghatározásban szerepelt a képi elemek megjelenítése eredetileg három képet választottam a példa kedvéért, amit később a projekt gyökér könyvtárában helyeztem el, ami lehetővé tette az egyszerű linkelést a képekre.

1. **ábra** A grafikai alkalmazáshoz választott képek

2.2 Elrendezés kialakítása

Az alkalmazás meglehetősen egyszerű, ezért minimális tervezést igényelt. Első lépésként a megjelenítendő információk összegyűjtése történt meg, így néhány szó megadása és az ablak fejlécszövege, valamint a méretek kerültek, meghatározásra. Az alkalmazásnak nincsenek valódi felhasználói funkciói, ezért egyéb tervezést nem végeztem.

3. Implementálás

3.1 Ablak létrehozása

Az ablak létrehozásához importálni szükséges a swing osztályt:

```
import javax.swing.*;
```

Ezt követően létrehoztam a feladat kiírásának megfelelően a K4WZH6_bead01 osztályt amelyet a JFrame osztályból származtatunk:

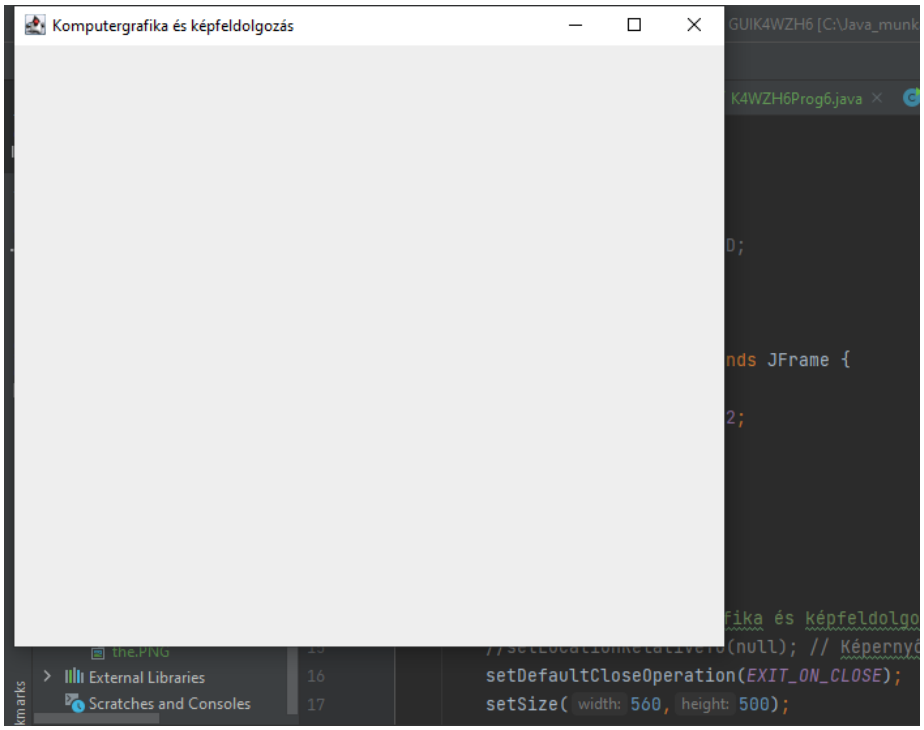
```
public class K4WZH6_bead01 extends JFrame { }
```

Ebben az osztályban hoztam létre a K4WZH6_bead01() metódust, amelyben létre tudtam hozni egy ablakot a JFrame osztály lehetőségeit felhasználva.

Majd megadtam az ablak fejlécszövegét:
```
super("Komputergrafika és képfeldolgozás");
```

A setSize() függvény két paramétere az ablak szélessége és magassága. Segítségével az ablak méreteit tudjuk megadni. A setDefaultCloseOperation() függvény segítségével az ablak bezárása utáni alkalmazandó operációt tudunk megadni. Az itt megadott EXIT_ON_CLOSE paraméter hatására a program futása véget fog érni 0 exit kóddal.

Ezek után már egy működő üres ablakot kapunk, amit adott méretű és bezárható.



2. ábra: Az alkalmazás ablak

További függvények is rendelkezésre állnak, amelyekkel az ablak megjelenése formázható. Például a setBackground() függvény segítségével az ablak háttérszínét tudjuk megváltoztatni. Használhatjuk az előre definiált színeket is:

```
setBackground(Color.yellow);
```

A fenti példa a sárga háttérszín beállítását mutatja.

3.2 Feliratok

A feliratok létrehozását a java.awt importálása után a font osztály egy példányának létrehozásával érhetjük el. Amelynek konstuktorai a betűtípus, a stílus és a méret. Ezek segítségével létrejön egy objektumpéldányunk, amelyet további tulajdonságokkal tudunk felruházni. setColor()metódussal a betűszínt tudjuk beállítani. A drawString() metódus a megjelenítendő szöveget tartalmazza. A felirat ezek után már meg fog jelenni az ablakunkban a megjeleníteni kívánt cím:

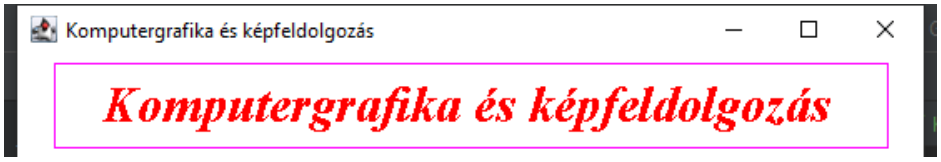
```
Font f = new Font("Times New Roman", Font.BOLD+Font.ITALIC,30);
g.setFont(f);
g.setColor(Color.RED);
g.drawString("Komputergrafika és képfeldolgozás", 60, 70);
```

3.3 Alakzatok rajzolása és kitöltése

Négyszögek rajzolására a java.awt importálása után nyílik lehetőségünk. Elérhetővé válik a Rectangel típus a változó számára és a drawRect() metódus, amelynek 4 fontos paramétere a balfelső csúcs kezdőkoordinátái és a négyszög dimenziói:

```
g.drawRect(30,35,500,50);
```

Természetesen a szín módosítására itt is lehetőség van a setColor segítségével.



3. ábra Téglalap keret rajzolása

Amennyiben kitöltött négyszög alakzatot szeretnénk rajzolni, akkor szükségünk lesz az alábbi importra is:

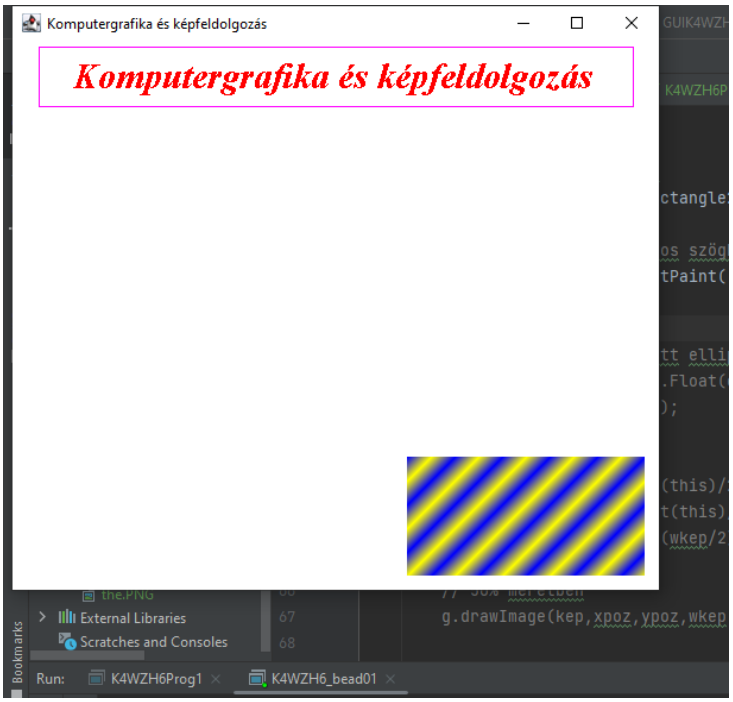
```
import java.awt.geom.Rectangle2D;
```

A `Rectangle2D` osztály esetén egy alakzatot kell példányosítani, amely hasonlóan történt mint a keret megrajzolásánál:

```
Shape teglalap = new Rectangle2D.Float(r.width-220,r.height-120,
200,100 );
```

Az elvárt paraméterek itt is a kiinduló koordináták és a dimenziók. A példában az „r” az JFrame által létrehozott ablak, amelynek a szélességét és magasságát megfelezve keressük meg az ablak középpontját. Ebből kivonva a téglalap szélességének a felét a téglalap abszolút középen lesz horizontálisan, ugyanezt a magasság adatokkal elvégezve vertikálisan is abszolút középre tudunk pozícionálni az ablakon belül.

A setPaint() metódus segítségével tudjuk a kitöltés színét, mintázatát beállítani, amelyet a fill() metódus segítségével el is végezhetünk a paraméterben megadott alakzaton.



4. ábra: Téglalap vonalas mintával, 45 fokos szögben kék és sárga színek közötti átmenettel

A kör alakzat létrehozása is hasonlóan egyszerű módon történik, csak itt a következő importálásra van szükségünk:

```
import java.awt.geom.Ellipse2D;
```

Az Ellipse2D osztály egy új példányát hozzuk létre, amelynek paraméterei szintén a kezdő koordináta valamint a horizontális és a vertikális átmérők hossza. A koordináta nem a középpont lesz, hanem a kör köré rajzolható négyszög bal felső csúcskoordinátája.

5. ábra: Kitöltött ellipszis

3.4 Képek kezelése

A java.awt osztály lehetőséget kínál képek megjelenítésére is. Ezt az imageIcon() alosztály példányosításával tehetjük meg, aminek a konstruktora egy képfájl lesz:

```
// 1. kép
ikonKep1 = new ImageIcon("mg1.jpg");
kep = ikonKep1.getImage();
```

A fenti példa az mg1.jpg fájl elérését mutatja a getImage() metódus felhasználásával. A létrehozott objektumpéldányt szintén a public void paint(Graphics g) osztályon keresztül fogjuk felhasználni és tulajdonságokkal felruházni a drawImage() metóduson keresztül. Ennek szükséges paraméterei a kép objektum neve, balfelső csúcskoordinátái és a kép megjelenítési dimenziói. Az alábbi példa az ablakhoz képest horizontálisan középre igazított 50% méretben megjelenő kép paraméterezését mutatja be:

```
int wkep = kep.getWidth(this)/2;
int hkep = kep.getHeight(this)/2;
int xpoz = (r.width/2)-(wkep/2), ypoz = 95;
g.drawImage(kep,xpoz,ypoz,wkep,hkep,this);
```

A getWith() és getHeight() metódusok a kép szélességi és magassági adatait adják vissza.

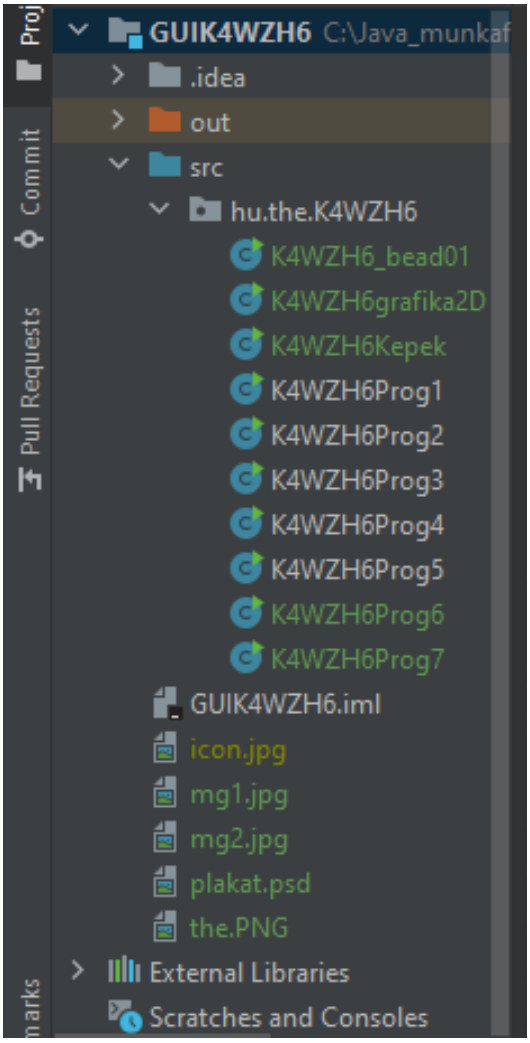
6. ábra: Kép megjelenítése középre igazítva 50% méretben

4. Fájl és mappaszerkezet

Az alkalmazást a neptunkódom megadásával a félév során használt JAVA munkafájlok közé helyeztem el. A csomag megadása a következő:

```
package hu.the.K4WZH6;
```

A mappaszerkezet a fejlesztőkörnyezet által létrehozott struktúra. Munkámat a többi osztály fájl közé helyeztem el és itt is történt meg a tesztelés illetve a hibák javítása.

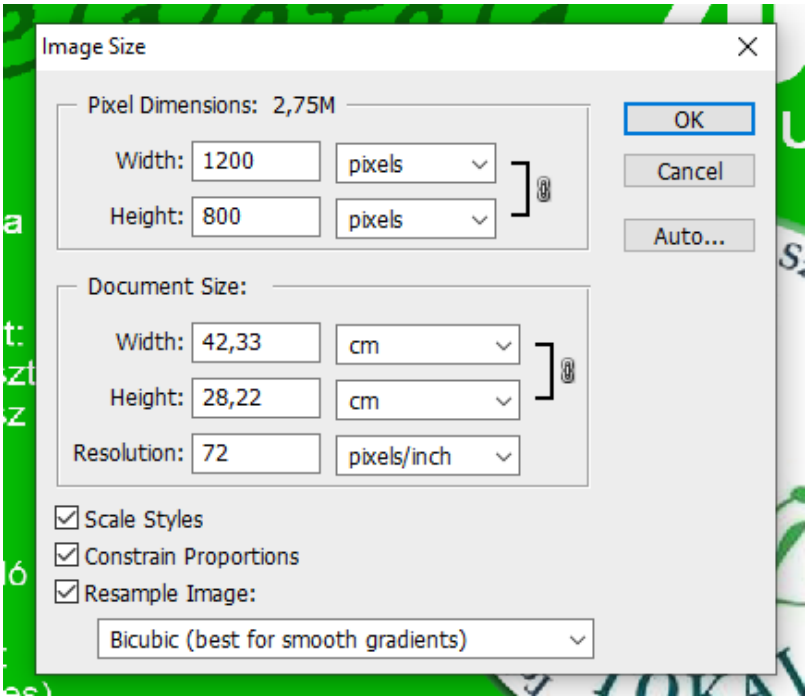


7. ábra: A projekt mappaszerkezete

5. Képszerkesztés Photoshop CS4 segítségével

5.1. Dokumentum létrehozása, háttér réteg, vászon paraméterei

A pixelgrafikus képfájl létrehozásának a célja az volt, hogy a közösségi médiában alkalmazott méretű plakátot készítsek. Az egyes szociális médiák elvárt képméreteit közé szokták tenni és a célunknak megfelelő méretben tudjuk elkészíteni a legoptimálisabban megjelenő képet. Én ehhez a https://features.hu/facebook-kepmeretek/#megosztott-kepek oldalt használtam. Mivel hírfolyamban megjelenő képet hoztam létre ezért 1200x800 pixel méretű képfájlt hoztam létre, amely 72 pixel / inch felbontású.



8. ábra: Fájl méretei

Az új PhotoShop fájl létrejötte után egy üres írásvédett háttér látható, amely az általunk megadott dimenziókkal rendelkezik. A háttér írásvédettsége feloldható, amennyiben konvertáljuk szerkeszthető réteggé vagy duplikáljuk és a másolat már gond nélkül szerkeszthető. Én ezt alkalmaztam.

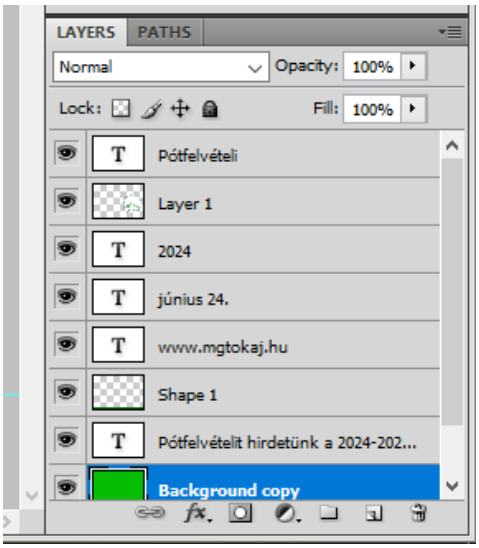
A hátteret ezt követően az elvárásoknak megfelelően zöldre (#3d9502) színeztem a kiöntő eszközzel.

5.2 Feliratok létrehozása, formázása, méretezése

A pixelgrafikus képszerkesztés fontos elemei a feliratok. A program a feliratot egy új rétegre helyezi, amely a PSD fájlban még tetszőlegesen alakítható objektum. Amíg nincs raszterizálva veszteségmentesen nagyíthatjuk, formázhatjuk és effektekkel láthatjuk el.

A könnyebb azonosíthatóság és a tipográfiai elvárások miatt több szöveg réteget is létrehoztam. A különböző paraméterű szövegobjektumoknak mindig új réteget hoztam létre.

Eltérő paraméter lehet a szín, betűtípus, méret, elhelyezkedés vagy az alkalmazott effektek. A rétegek sorrendje is fontos, hiszen a felső réteg fedheti az alatta elhelyezkedőket.



9. ábra: Szövegrétegek elhelyezése

5.3 Alakzatok

Alakzatok alkalmazása egy helyen történt a láblécben. A Rectangle tools eszköz segítségével a háttér réteg fölé egy téglalapot helyeztem el # 056805 hexadecimális színkódú színnel kitöltve. A színek sem véletlenszerűek. Az egyes színeket az intézmény logójából olvastam ki a ColorPicker pipetta eszközével:

10. ábra: A pipetta használata

A pipettával a logó megfelelő felületére kattintva a kiválasztott pixelek színkódja kinyerhető.

5.4 Kép beillesztése

Az Adobe Photoshopban már létező képállományt vágólapról tudunk beilleszteni. Tehát vagy megnyitjuk a képfájlt a programban szerkesztésre és ott kijelölve a képet tartalmazó réteget beillesztjük a munkánkba, vagy más programban a vágólapra helyezzük és beillesztjük az aktuális réteg fölé.

A beillesztett kép egy új rétegre kerül. A rétegek sorrendje itt is fontos a láthatóság szempontjából. Amennyiben transzparens képet illesztünk be (pl. PNG fájl), az alsó réteg vagy rétegek láthatóak maradnak a kép környezetében a vászon területén belül.

Az egyes rétegek láthatósága ki- és bekapcsolható a rétegeket kezelő panel kis szem ikonjára kattintva. Én egy PNG képet illesztettem be, amely az intézmény logóját ábrázolja és transzparens. A háttér réteget kikapcsolva négyzetrács szemlélteti, hogy mely területeken lesz transzparens a réteg.

11. ábra: Transzparens réteg

Az Adobe Photoshop szolgáltatásait egy feladaton keresztül bemutatni lehetséges, hiszen sok és egyre több lehetőség kínálkozik a képek pixelgraikus szerkesztésére.

5.5 Projekt és kép mentése

Amennyiben elkészültünk a kívánt szerkesztési munkákkal a projektünk eredménye, többnyire két fájlt szoktam elmenteni. Az első az Adobe Photoshop fájl, amely .PSD kiterjesztést kap alapértelmezetten. A PSD fájl lehetővé teszi, hogy a szerkesztést ott folytathassuk, ahol abbahagytuk, betöltve a rétegeket és a beállításokat, amelyeket a kép szerkesztése folyamán alkalmaztunk. Amennyiben ettől eltérő állományra van szükségünk akkor másként is elmenthetjük az állományunkat. Számos képformátum közül választhatunk. Én jelen estben PNG fájlt hoztam létre, amely kevésbé veszteséges, mint a JPG tömörítés. JPG formátum esetén a mentésnél a tömörítés mértéke, a tulajdonképpeni minőség állítható egy 12 fokozatú skálán. A PNG formátum ajánlatos, amennyiben a kép szövegeket is tartalmaz, amelyeknek olvashatónak kell maradniuk.

A PNG fájlokat úgy is elmenthetjük, hogy az egyes rétegek áttetszősége megőrzésre kerül. Ez a számítógépes grafika felhasználószintű alkalmazásának egyik populáris eleme. WEB-es felhasználásra is jól alkalmazható a végeredmény.