

TUDNIVALÓK:

Most is és a következő gyakorlatokon is – akkor is, ha külön nem emeljük ki – az órán meg nem oldott feladatok **HÁZI FELADAT**-ként megoldandóak!!!

Ez fontos a tárgy sikeres teljesítéséhez!



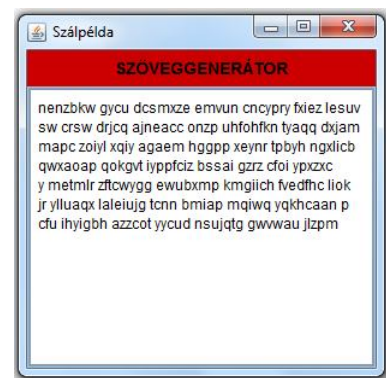
1. feladat

Készítsen grafikus „szöveg”-generátort!

a) Egy szál segítségével negyed másodpercenként (250 ezredmásodperc) írjon ki 'a' – 'z' közötti véletlenszerű betűket.

b) Egy másik szál segítségével 4-8 betűnként törje meg szóköz-zel a folyamatos szöveget (a szavak hossza véletlenszerű).

c) A „majdnem” sorkizárt írásmód kedvéért egy harmadik szál 40 betűnként emeljen sort.



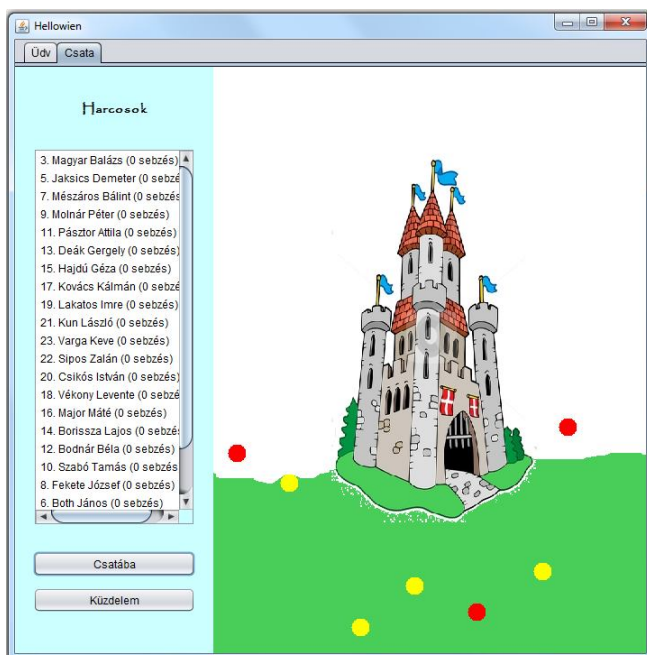
A szöveg a program bezárásáig íródjon ki a képernyőre.

1/b. (HF) feladat

Az előadáson tárgyalt „értelmes”-szöveg generátor kért javítása.

(A névelőnél próbáljon a nyelvhelyességre is figyelni, vagyis magánhangzó előtt „Az”, mássalhangzó előtt „A” legyen.) Ezt is grafikus felületre írja!

2. feladat



Harcoljanak Mátyás katonái!

Vagyis a csatába küldés után változzon aktívra a „Küzdelem” feliratú gomb (előtte nyilván inaktív volt), hatására pedig bizonyos időközönként menjen arrébb egy-egy, a harcolók közül véletlenül kiválasztott katona. Az „arrébb menni” most azt jelenti, hogy egy másik véletlen helyre ugrik.

A küzdelmet csak egyszer lehet elindítani, ezért megnyomása után a gomb felirata változzon meg „Leállít”-ra, és ha ismét megnyomjuk, álljon le a csata. (Segítség: Célszerű a vezérlő osztályt átalakítani szál osztállyá.)

FONTOS: Ha a vezérlő osztály működését korábban a `start()` metódus indította, akkor most kezdje avval, hogy ezt átnevezi `modjuk indit()`-ra (Refaktorizációval.)

3. feladat

Készítsen egy 500x500-as grafikus alkalmazást egy analóg óra szimulálására!

a) Először készítse el a másodpercmutatót. – ezt most egy vonal helyettesítse, amely 60 helyett 12 lépésben járja be a kört. (Másodpercenként 5 másodpercnyit ugrik.) A mutató a program indulásakor automatikusan elindul, vagyis még nem a gombnyomás hatására.

b) Az előző feladatot alakítsa át úgy, hogy egérekattintás hatására változzon meg az órajárás iránya.

c) (HF) A feladatot most úgy alakítsa át, hogy egy gombnyomás hatására változzon meg az órajárás iránya. Figyeljen rá, hogy a gomb bármilyen átméretezés esetén is az ablak közepén maradjon (vízszintes értelemben).

d) Most úgy alakítsa át, hogy a mutatót gombnyomás hatására lehessen elindítani (vagyis amíg nem nyomtuk meg a gombot, addig áll az óra). Újabb gombnyomás hatására álljon meg, majd újabb hatására induljon el, stb. Figyeljen a gomb feliratára!

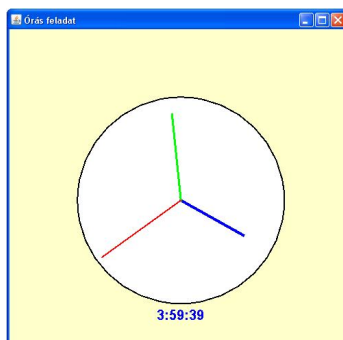
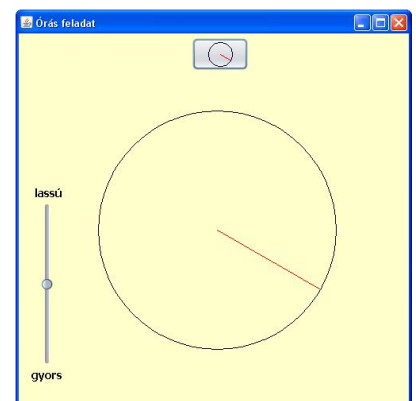
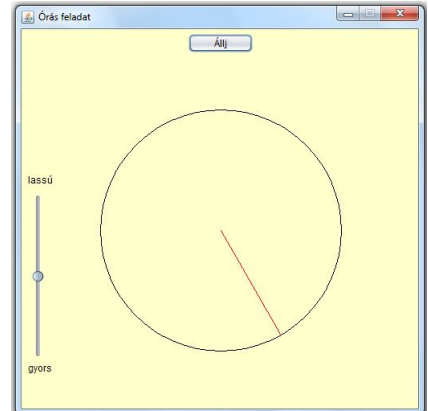
e) (HF) Oldja meg ugyanezt két gombbal.

f) Oldja meg, hogy a képen látható módon, egy csúszka segítségével lehessen változtatni a mutató sebességét.

g) (HF) Alakítsa át a feladatot úgy, hogy az óra 12 számjegye is ki legyen írva
- eleve már a grafika indulásakor
- vagy úgy, hogy az első körbemozgó vonal hatására jönnék létre, és kint is maradnak.

h) Most a gombon ne felirat jelenjen meg, hanem az órával szinkron módon működő másik kis óra. A gombra kattintva vagy mindkét óra induljon el, vagy mindkettő álljon meg. (Segítség: írjon saját JButton osztályt.)

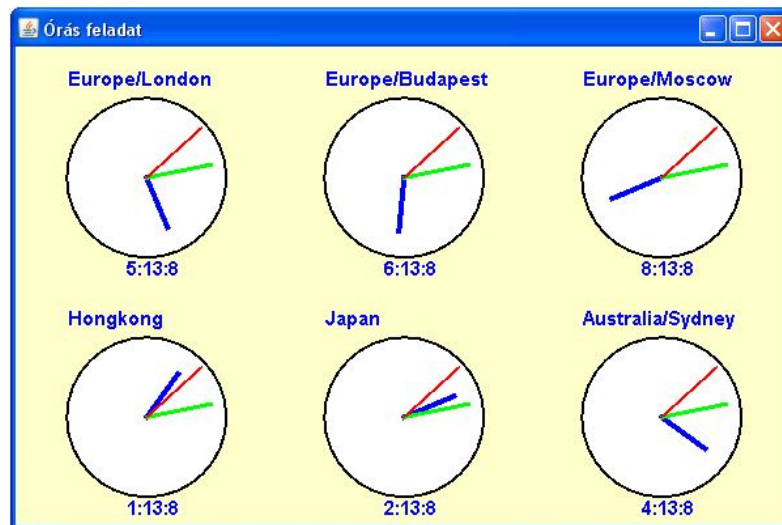
i) (HF) Alakítsa át a programot „valódi” órává. A másodpercmutató mellett legyen perc- és óramutató is. A mutatók elrendezése egy valódi órát szimuláljon. Mindhárom mutató különböző színű és hosszúságú legyen.



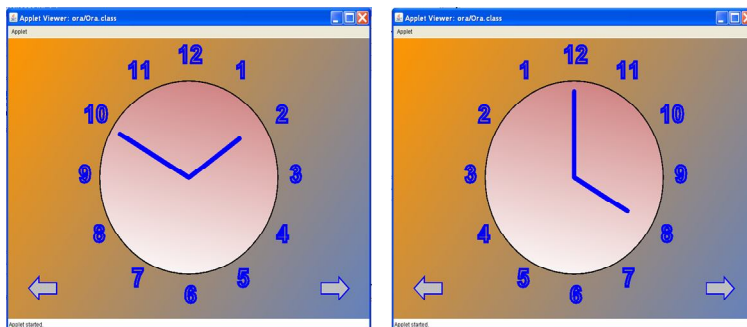
j) (HF) Az analóg óra alatt jelenítse meg az órán mutatott időpont digitális változatát. Mindkettő (az analóg is és a digitális is) az operációs rendszer rendszeridejét mutassa. (Segítség: Date és Calendar osztály, ill. ld. előadás – szerintem könnyebb ez alapján megrajzolni.)

k) (HF) Az analóg óra mutatóinak vastagságát is állítsa be az ábrán látható módon. (Segítség: használja a Graphics2D-t.)

l) Készítsen olyan alkalmazást, amelyen különböző időzónák aktuális idejét láthatjuk:



(Segítség: Használja fel az előzőekben megírt Ora és RajzPanel osztályokat, és a Frame-re Grid Layout elrendezéssel rakja fel az n db panelt. A zónaidő meghatározásához használja a TimeZone osztályt, vagy a java.time csomagot.)



m) Készítsen appletet vagy grafikus alkalmazást, amelyben egy óra mutatói mozognak egy óra mutatóihoz hasonlóan (természetesen az igazi órához hasonlóan, vagyis mialatt a nagymutató körbeér, a kicsi az alatt tesz meg egy egységnyit). A nyilakra kattintva megadható, hogy milyen irányban forogjon a mutató – ekkor a számok kiírási

sorrendje is változzon.

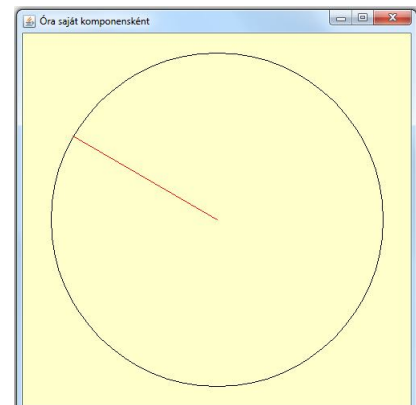
Természetesen ennél sokkal szebb megoldásokat is létrehozhat (pl. szebb színek, vagy az óralapon belül vannak a számok ☺, vagy a mutatók „vonszolják” magukkal a számokat akár úgy, hogy mindig csak az aktuális szám látszódik vagy úgy, hogy egy körbejárásakor a nagymutató kirajzolja, majd ha körbeért, letörli, és kezdődik előlről, stb.) A megoldáshoz használja a Java2D lehetőségeit!

n) Most oldja meg úgy a feladatot, hogy az Ora nem csak egy rajzot készítő osztály, hanem a JComponent osztály leszármazottja.

Segítség:

Ekkor a rajzol() metódust nem a panelről, hanem az Ora osztály saját paintComponent() metódusából kell meghívni.

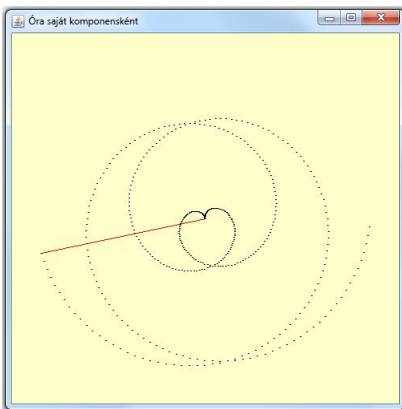
A kör középpontja a komponens középpontja, sugara lehet mondjuk a szélesség és magasság közül a kisebbiknek a 90%-a.



A konstruktorban most nem kell megadni az óra középpontját, sugarát, de kellene az Ora osztályban egy beallit() metódus, amely a megadott paraméterek alapján beállítja a komponens méretét (setSize()) és bal felső sarkának koordinátáit (setLocation()).

A létrehozás után this.add(ora); módon lehet hozzáadni a panelhez. Ha azonban üres konstruktorral definiálja, és az összes többi beállítandó adatot setterrel adja meg, akkor ez az Ora komponens ugyanúgy ráhúzható a panelre, mint bármilyen más komponens. Csak persze, majd be kell állítani a méreteit.

Mivel az Ora osztály már leszármazott, ezért most a Runnable interfész implementálásával kell definiálnunk. A szálindítást ennek megfelelően kell majd átalakítani. (Ld. előadás).



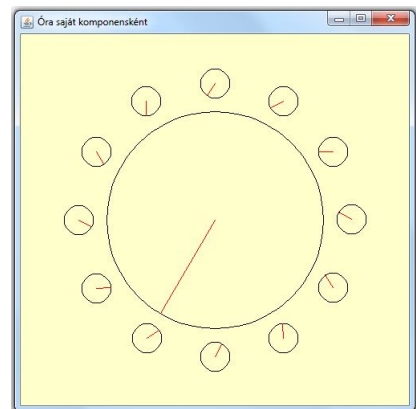
o) Kicsit eljátszhat az előző megoldással.

Vegye kicsit sűrűbbre a lépéseket (kisebb szöggel lépjen), és minden lépéskor csökkentse a kör sugarát 0-ig, majd vissza az eredeti méretig. Az óra keretét szimbolizáló kör helyett most a mutató aktuális helyének végpontja köré rajzoljon egy-egy kis pöttyöt úgy, hogy ezek a pöttyöcskék meg is maradjanak. Az ábrán láthatóhoz hasonló nyomvonal alakul majd ki a mozgás közben. Ha megtetszett, kitalálhat más formációkat is.

p) Ha már saját komponensként kezeli az órát, akkor akárhányat fel lehet rakni a panelre, akár így is.

Természetesen mindegyik óra jár.

Próbálja megoldani!



q) Készítsen grafikus Java programot, amely egy radarral ellátott csatahajó működését szimulálja.

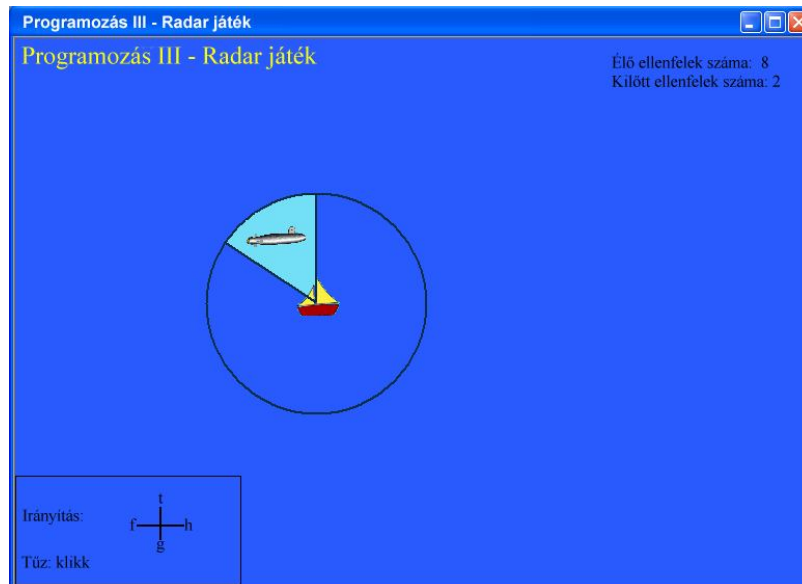
A program specifikációja a következő:

- a radar egy hajón üzemel, amely ellenséges tengeralattjárók felderítését és likvidálását végzi
- a radar a hajó körül meghatározott sugarú, kör alakú területet képes pásztázni
- a hajó körül körbe forgó körcikk jelzi, hogy hol tart radarunk a felület letapogatásával. A körcikken belül látszik, hogy mi zajlik az adott területen a víz alatt. A körcikk által le nem fedett területek üres képet mutatnak (csak a víz látszik).

- A tengeralattjárók véletlenszerűen mozognak a háttérben (nem csak felbukkannak, hanem véletlenszerű irányba véletlenszerű sebességgel mozognak, természetesen ésszerű határok között)
- A harcteret a teljes grafikus felület jelenti, ezen belül tudnak mozogni a tengeralattjárók. A radar természetesen ennek csak egy részét, a hajó körüli területet képes pásztázni
- Adjon lehetőséget a hajóval történő mozgásra (pl. a billentyűzet megfelelő billentyűinek lenyomásával vagy valamilyen módon az egérrel)
- Adjon lehetőséget az ellenséges tengeralattjárók likvidálására, azaz amikor a radar képernyőjén látszik az ellenfél, akkor le lehet lőni (pl. rákattintunk az egérrel)
- A játék addig tart, amíg vannak ellenséges tengeralattjárók (ezek száma egyébként a játék kezdetekor véletlenszerűen kerül meghatározásra)

A program elkészítésekor használja a fantáziáját, a fentiek csak irányelvek. Törekedjen a lehető „legszebb” grafikus felület kialakítására.

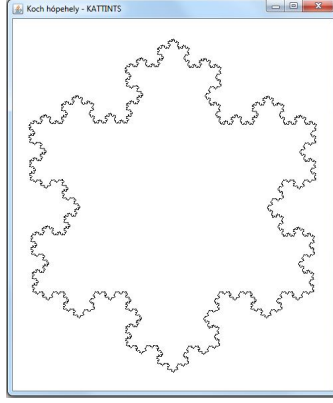
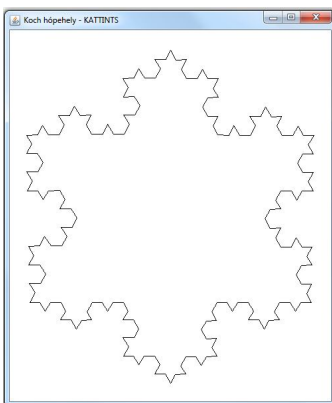
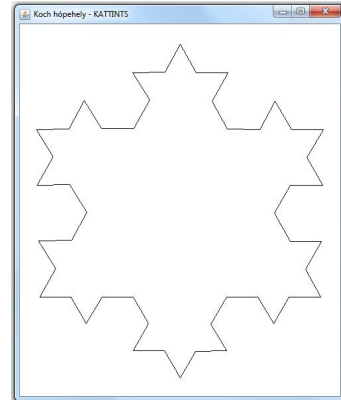
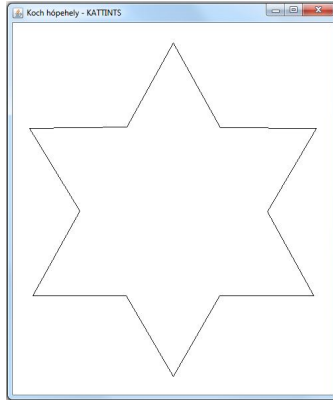
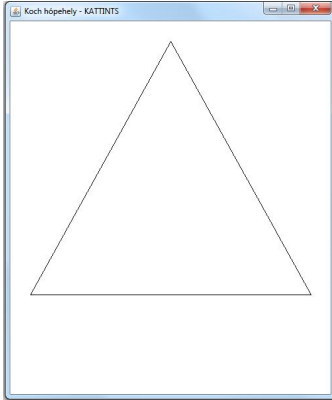
Egy lehetséges grafikus felület (természetesen ettől merőben eltérőt is készíthet):



TOVÁBBI FELADATOK

(vegyesen vannak köztük egyszerűbbek és összetettebbek)

1. feladat



Az itt látható méretarány:

Frame: 500*600

Felső, bal-, jobboldaltól
való távolság: 30.

Alsó távolság: 150

Készítsen alkalmazást a Koch-hópehely (<http://hu.wikipedia.org/wiki/Koch-görbe>) alakulásának bemutatására.

Igen, tudom, hogy már volt, ez volt az előző gyakorlat 3. feladata, de most úgy oldja meg, hogy ne kattintásra mutassa az újabb szintet, hanem animált módon, mégpedig így: a háromszögből indulva jusson el mondjuk a 6. szintig (tovább nem nagyon érdemes, mert túlzottan forrásigényes a rekurzió), majd csökkenjen vissza a háromszögig, aztán megint jusson el a 6. szintig, megint csökkenjen vissza, stb.

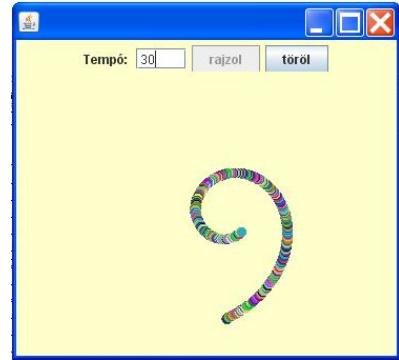
2. feladat

Készítsen olyan grafikus alkalmazást, amely N-től számol visszafelé 0-ig. Minden egyes szám megjelenése után legyen egy kis várakozás.

Alakítsa át a programot úgy, hogy a visszaszámlálás végén legyen valamilyen meglepetés ☺ („robbanás” ... stb.)

3. feladat

a) Készítsen grafikus alkalmazást, amely egy gombnyomás hatására lassan, pontonként változó színekkel elkezd rajzolni egy csigavonalat, egy másik hatására pedig lassan visszatörli. A rajzolás sebességét egy szövegmezőbe gépelve adja meg. Oldja meg, hogy induláskor csak a rajzol gomb legyen aktív, és a törlés gomb akkor váljon aktívvá, mielőtt elkezdődik a rajzolás. Ekkor viszont a rajzol gomb váljon inaktívvá, és csak akkor váljon ismét aktívvá, ha kitörölte a rajzot. Ekkor ismét a rajzol gomb legyen aktív. Figyeljen rá, hogy az ablak tetején lévő komponensek átméretezéskor is középre kerüljenek!



b) Próbálkozzon egyéb matematikai függvényekkel, így változtatgassa a csigavonalat alkotó pöttyök útvonalát, jópofa alakzatokat lehet így előállítani.

Segítség: A csiga képlete:

$$x = kpx + (\text{int})(\text{sugar}/(1+\text{szog}/2)*\text{Math.sin}(\text{szog}));$$
$$y = kpy + (\text{int})(\text{sugar}/(1+\text{szog}/2)*\text{Math.cos}(\text{szog}));$$

ahol (kpx, kpy) a csigavonal középpontja, (x,y) az aktuális pötty középpontja, sugar a csigavonal sugara, szog pedig a vízszintessel bezárt szög – ennek változik az értéke valamilyen lépésközzel.

Saját eseményt kell definiálni, amely értesítést küld, ha elfogyott a rajz.

4. feladat

Készítsen „Fényújságot”!

Az ablak tetején lévő szövegmezőbe beírt szöveg az indít gomb hatására kezdjen el beúszni a jobboldalról, közben a színe is változzon. Ha a baloldalon eltűnt a szöveg, akkor kezdődjön ismét a jobboldalon.

b) Szál helyett használja a Timer osztályt!



5. feladat

Készítsen egy grafikus alkalmazást, amely kiírja az aktuális dátumot.

a) Alakítsa át az alkalmazást úgy, hogy az aktuális dátumot 100 pt méretű TimesRoman betűtípussal írja ki. (Font osztály)

b) Alakítsa át úgy, hogy csak az időt mutassa ÓÓ:PP:MM formátumban (pl.: 12:53:03) (SimpleDateFormat osztály)

c) Alakítsa át úgy, hogy az óra működjön is! (Szálkezelés)

d) Játsszon a feladattal! (próbálgassa az idő kijelzésének sebességét, színesítse a feladatot, tegyen bele értelmes funkciójú nyomógombokat [pl.: nyomógomb hatására megáll az óra...stb.]

e) Alakítsa át appletté!

6. feladat

Készítsen kis marslakókat, amelyek egy grafikus felületen (vagy appletben) mászkálnak úgy, hogy ha falnak mennek, akkor új irányba indulnak. Egér kattintásra egy új marslakó kerüljön be az animációba (új szál induljon).

7. feladat

a) Készítsen egy grafikus alkalmazást vagy appletet, amelyen véletlenszerű helyeken megjelenít egy teli négyzet de csak egy rövid időre.

b) Legyen lehetőség az egérrel kattintani a felületen. Ha a megjelenő négyzeten belül kattintottunk, akkor egy kijelzőn növeljük a pontszámunkat.

c) Bonyolítsuk a játékot azzal, hogy véletlenszerűen piros és zöld színnel jelenik meg a négyzet. Amennyiben piros négyzetre kattintunk, úgy a rossziút lőttük le, tehát kapunk plusz pontot, ha pedig a zöld színű négyzetre kattintunk, úgy ártatlan embert lőttünk le és ezért pontlevonás jár (-2p).

d) Alakítsa a programot a fantáziája szerint (pl. lehet egy Állj! nyomógomb, melynek hatására a játék befejeződik és az elért pontszámnak megfelelő értékelést láthatunk, stb.)

8. feladat

Készítse el egy DVD lejátszó szimulációját.

Rajzolja meg grafikus felületen a gombokat és a kijelzőt.

Az alábbi nyomógombokat valósítsa meg:

- Film indítása (Strat)
- Film leállítása (Stop)
- Pillanat állj (Pause)
- Lejátszás folytatása (Continue)

A feladat megoldásához használja egy szál életének élekciklusait!