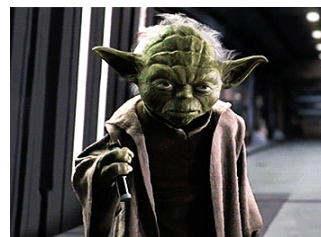
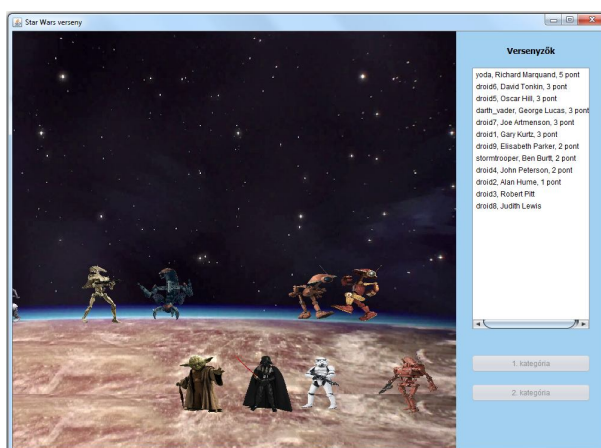
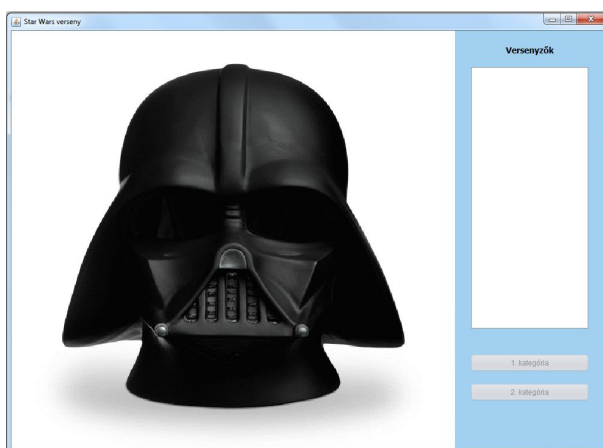


Star Wars

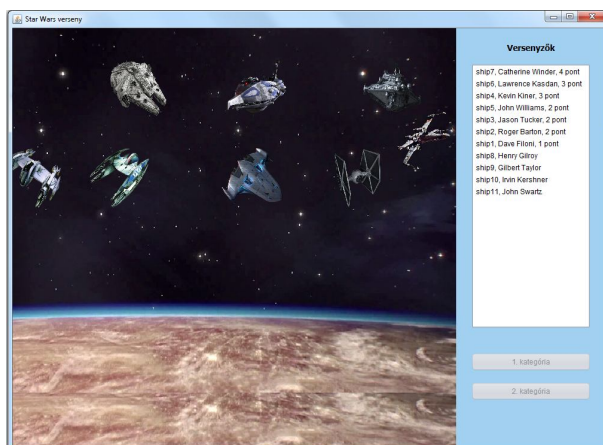
A felfokozott érdeklődésre való tekintettel hirdessünk meg egy saját InfOscar díjat, amelyért a Star Wars figurák tervezői versenyezhetnek. Két kategóriában lehet benevezni, az egyikben a földi figurák, a másikban a levegőben mozgó figurák indulhatnak.



Induláskor mindkét gomb inaktív. Az induló képre kattintva megváltozik a háttér, és aktívvá válik az 1. kategória feliratú gomb. A gombot megnyomva indulnak az első kategóriába tartozó versenyzők. Egyforma időközönként indul el egy-egy figura a képernyő bal széléről, és egyforma tempóban mennek (a fájlban lévő eredeti sorrend szerint követik egymást). Egy „kört” tesznek (jobbra a panel széléig, fel valamekkora távolságra, majd vissza balra), majd kimennek a színről. Eközben lehet szavazni rájuk. Amelyikre rákattintunk, annak eggyel növekszik a pontszáma.



Amikor elindulnak, a kategória gomb inaktívvá válik. Ha az utolsó is elhagyja a szint, akkor a 2. kategória feliratú gomb aktívvá válik, és megkezdődhet a második verseny. Ez ugyanolyan, mint az első kategóriás, csak a figurák a levegőben tesznek meg egy kört.



Egy-egy kategória-verseny indulásakor a jobboldali listában az adott kategóriában indulók adatai láthatóak a nevezési (azaz a fájlban adott eredeti) sorrendben. Soronként a figura, illetve a tervezője nevét olvashatjuk. Ha valakire szavazat érkezik (rákattintottak), akkor ezek mellé kiíródik az eddigi pontszáma is. Ekkor már pontszám szerint csökkenően vannak rendezve.

Amikor a második kategória utolsó versenyzője is elhagyja a terepet, megjelenik a premierre invitáló reklám.

A javasolt felületméret: 1000 x 700 pixel, amelyből a jobboldali rész 250 pixel széles.

A többi javasolt méretet az adatok között szereplő *Global.java* fájl tartalmazza.

Adatfájl szerkezete soronként: figura neve;tervező neve;kategória. (Emlékeztető: a megadott sql akkor használható, ha "kézzel" hozzuk létre az adatbázist. Ha kódból, akkor a sorok végéről törölni kell a pontosvesszőket.)

Megjegyzések:

- A szálak működését a szálütemező vezérli, ezért hiába állítjuk be az egyenletes sebességet, nem biztos, hogy mindig az lesz. Executorral kellene megoldani, ha akarja, próbálkozhat vele.
- Az elején, végén látható animációk egyszerű gif képek, ha a kép kirajzolásakor a rajzoló metódus utolsó paramétere nem null, hanem a rajzolást végző panel, akkor mozogni fog.
- Ha esetleg több interfészt is implementálni akar, akkor az interfészeket vesszővel választjuk el egymástól. Ha örököltetni és implementálni is akar, akkor előbb az öröklés, majd vessző nélkül folytatható az implementálással.

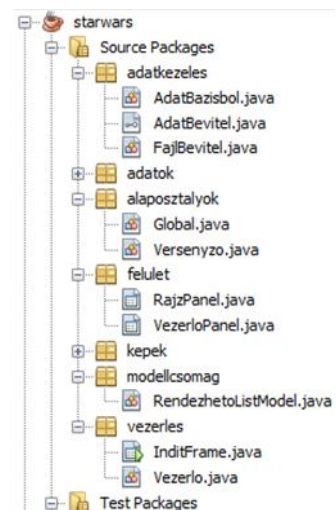
Néhány megoldási ötlet:

Fejtörést talán csak a kétféle kategória okozhat. Első ötletünk akár az öröklődés is lehet, de ha végiggondoljuk, rájövünk, hogy a kétféle versenyző kizárólag csak abban különbözik egymástól, hogy milyen magasságban járja be a felületet, vagyis elég, ha példányosításkor paraméterként megadjuk a kategóriát, és ettől függően más-más magasságot állítunk be.

A versenyzőket szálként kezeljük, és mivel rendezni is szeretnénk őket, ezért nem csak a `Thread` osztályt terjesztik ki, hanem implementálják a `Comparable` interfészt is. (Célszerű rendezhető listamodellt használni.)

A mozgást érdemes három ciklussal megoldani: az elsőben vízszintesen balról a felület jobbszéleig (figyeljen rá, hogy a kép a felületen belül maradjon); a másodikban egy függőleges, a harmadikban egy balra vivő vízszintes mozgás.

Figyelni kell, hogy eltalálták-e (a felületre kattintáskor a vezérlő ellenőrzi az aktuális versenyzőlista összes elemét). Ha eltalálták, akkor eggyel növekszik a pontjai száma.



Kérdés még a versenyzők indítása. Ha csak egyetlen kategória lenne, akkor ez lényegében ugyanaz, mint a lepkevadász feladatban, vagyis a vezérlőt lehetne szálként kezelni, ami bizonyos időközönként elindítja a következő példányt.

De gyakorlatilag ugyanezt tehetjük most is, csak érdemes két külön szálát létrehozni, az első az első kategóriába tartozókat indítja, a második a másodikakat. A kategóriát jelző gomb megnyomásakor meghívjuk a vezérlő `public void categoria(int i)` metódusát. Ez a beolvasott adatok alapján létrehozza az aktuális versenyzők listáját (mivel szálpéldányok, ezért `CopyOnWriteArrayList`), kiírja őket a vezérlőpanellel, és létrehoz egy szálát, amely bizonyos időközönként elindítja a versenyzők listájának következő tagját (egyszerű `for` ciklus). Azt azonban nem szabad elfelejteni, hogy egy szálát csak egyszer lehet elindítani, vagyis két külön szálpéldányra van szükségünk. Ezt legegyszerűbben úgy oldhatjuk meg, hogy a vezérlő osztály implementálja a `Runnable` interfészt, és ebből hozzuk létre a szálpéldányokat. (Meg lehet oldani a feladatot egyetlen szállal is, ekkor a két kategória versenyzése között várakoztatni kell a szálát (`wait - notify`).)

Rajzoláskor a rajzpanel vagy csak kizárólag azt rajzolja, amit a vezérlő mond neki, ekkor a háttérképek rajzoltatása is a vezérlő dolga, vagy a háttérképeket rábízhatjuk a rajzpanelre. Bármelyik megoldást is választja, figyelni kell a háttérképek változtatására. Ahogy a feladat szövegében is szerepelt, ha animált gif-et szeretnénk rajzoltatni, akkor szükségünk van a képrajzoló metódus `ImageObserver` paraméterére is (az utolsó paraméter). Ez jelenleg a rajzpanel. (Ha a panelen rajzoljuk ki a hátteret, akkor `this`, ha a vezérlőben, akkor az ott deklarált `RajzPanel` példány.)