Készítette: Fecsó András Balázs

Email: [XCF3R2@inf.elte.hu](mailto:XCF3R2@inf.elte.hu)

Feladat:

8. Aszteroidák

Készítsünk programot, amellyel az aszteroidák játékot játszhatjuk. A feladatunk az, hogy egy űrhajó segítségével átnavigáljuk egy aszteroidamezőn. Az űrhajóval a képernyő alsó sorában tudunk balra, illetve jobbra navigálni. A képernyő felső sorában meghatározott időközönként véletlenszerű pozícióban jelennek meg az aszteroidák, amelyek folyamatosan közelednek állandó sebességgel a képernyő alja felé. Az idő múlásával egyre több aszteroida jelenik meg egyszerre, így idővel elkerülhetetlenné válik az ütközés. A játék célja az, hogy az űrhajó minél tovább elkerülje az ütközést. A program biztosítson lehetőséget új játék kezdésére, valamint játék szüneteltetésére (ekkor nem telik az idő, és nem mozog semmi a játékban). Ismerje fel, ha vége a játéknak, és jelenítse meg, mennyi volt a játékidő. Ezen felül szüneteltetés alatt legyen lehetőség a játék elmentésére, valamint betöltésére.

## Elemzés

* A játék követelményei
  + Lehetőség az űrhajó irányítására jobbra és balra a képernyő alján
  + Aszteroidák generálása véletlenszerű pozícióban, az idővel telésével egyre gyakrabban
  + A játéknak detektálnia kell, ha az űrhajó ütközik egy aszteroidával
  + A játék állapotainak változtatása (játék, szüneteltetés, mentés, betöltés
  + Pontozási rendszer (a különböző méretű és sebességű aszteroidák hány pontot adnak)
  + Nehézségi szintek a játék közben
* Megvalósítás
  + Egyablakos asztali alkalmazás Windows Forms alkalmazásával
  + Az ablakban elhelyezésre kerül egy menüsor a következő menüpontokkal: Game(New Game(Új játék), Pause/Play(a játék szüneteltetése és folytatása), Save Game(A játék mentése), Load Game(játékállás betöltése), Reset High Score(A jelenlegi legmagasabb pontszám értékét visszaállítja 0-ra), Exit(A játékból való kilépés)) valamint Help(Controls(Az irányítással és a pontozással kapcsolatos adatok), About(A játékkal kapcsolatos adatok))
  + A játékteret a főablak üres területe reprezentálja, ahol kirajzolásra kerülnek az aszteroidák és az űrhajó is, amelyet a jobbra és balra nyilakkal lehet irányítani
  + A játéknak automatikusan felismeri, ha vége a játéknak (az aszteroida és az űrhajó ütközik) és egy üzenet jelenik meg. Szintén külön ablakokkal működik a mentés és a betöltés is, ahol a fájlneveket a felhasználó adja meg

A képen szöveg, képernyőkép, diagram, sor látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.

## Tervezés

* Programszerkezet:
  + A programot 3 rétegű architektúra határozza meg. A megjelenítés az AsteroidGame (későbbiekben View), a modell az AsteroidGameMechanic.Model (későbbiekben Model) míg a perzisztencia az AsteroidGameMehanic.Persistence (későbbiekben Persistence) névtérben helyezkedik el
* Persistence
  + Feladata a játékállapot tárolása valamint a high score tárolása
  + A GamePersistence osztály csinálja a játékállapot fájlba történő mentését valamint betöltését
  + A HighScore osztály végzi a high score tárolását és betöltését szöveges fájlba/fájlból
  + A játékállapot egy .*save* kiterjesztésű fájlban kerül lementésre amely tartalma:
    - Pontszám
    - Játékidő
    - Képernyő méretei
    - Űrhajó X pozíciója
    - Aszteroidák adatai(X, Y, BaseSize, Speed

A képen szöveg, képernyőkép, Betűtípus, sor látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.

* Model
  + A GameModel osztály valósítja meg a játék lényegi reszeit, szabályozza a játékmenetet valamint a játék paramétereit
  + A SpaceShip osztály az űrhajót reprezentálja, kezeli a mozgatást a képernyő határainak figyelembe vételével. Tulajdonságai: X, Y, Widht, Height, metódusai: MoveLeft(), MoveRight()
  + Az Asteroid osztály tartalmazza az aszteroidát, kezeli a mozgást és a méret- és sebességváltozásokat a különböző instanceok esetében. Tulajdonságai: X, Y, Width, Height, Speed, metódusai: Move()
  + A játék időbeli kezelését egy külső időzítő oldja meg és hívja meg az update metódust. A játékidő az Update hívások között eltelt idő összegéből számolódik
  + Az állapotváltozásról események tájékoztatnak (GameOver, ScoreChanged, GameTimeChanged, HighScoreChanged)
  + A játék a nehézséget dinamikusan számolja a model az idő és a pontszám alapján:

1. 1double timeFactor = Math.Min(\_gameTime.TotalSeconds / 60.0, 2.0);
2. double scoreFactor = Math.Min(\_score / 1000.0, 2.0);
3. double difficulty = 1.0 + (timeFactor + scoreFactor) / 2.0;
   * Az aszteroidák generálása valószínűségi alapon történik, a nehézségtől függően:
     + Kis aszteroida(40% esély): 15-25 pixel, gyors
     + Közepes aszteroida(30% esély): 25-40 pixel, közepes sebesség
     + Nagy aszteroida(20 % esély): 40-60 pixel, lassú
     + Óriási aszteroida(10% esély): 60-80 pixel, nagyon lassú

* View
  + A nézetet a Form1 osztály biztosítja amely tárol egy GameModel (\_gameModel) objektumot és a perzisztencia objektumokat (\_persistence, \_highScoreManager)
  + A játékteret a periódikusan frissülő OnPaint metódus hozza létre. A frissüléseket a \_gameTimer segítségével kb 60 FPS-re próbáltam beállítani
  + A felületen létrehozzuk a megfelelő menüpontokat és dialógus ablakokat a megfelelő eseménykezelőkkel
  + Rajzolási metódusok:
    - DrawSpaceship()
    - DrawAsteroids(): Érdekessége, hogy az aszteroidák méretétől függ a részletességük
    - DrawStarts(): A csillagoka a háttérben
    - DrawUIText(): Játékállapot szöveges megjelenítése
    - DrawHighScore(): A legmagasabb pontszám, az aktuállis pontszám és a nehézségi szint megjelnítése
  + Billentyűzet eseménykezelői:
    - OnKeyDown: nyíl billenyűk és space kezelése
    - OnKeyUp: nyíl billentyűk felengedésének kezelése

A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.A képen szöveg, képernyőkép, Betűtípus, szám látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen. A képen szöveg, sor, Betűtípus, szám látható

Előfordulhat, hogy az AI által létrehozott tartalom helytelen.

* Tesztelés
  + A modell funkcionalitását MSTest egységtesztek végzik az AsteroidGameTest osztályban
  + A teszteléshez Moq keretrendszert használtunk mock objektumok létrehozására
  + Az alábbi tesztek kerültek megvalósításra:
    - Test\_GameInitialization: Játék indítása, kezdeti állapot ellenörzése
    - Test\_SpaceshipMovement\_Left/Right: Űrhajó mozgatásának és határainak tesztelése
    - Test\_AsteroidSpawning\_OverTime: Aszteroidák generálásának időbeli változásának tesztelése
    - Test\_AsteroidSpawning\_Positions: Az aszteroidák generálási helyeinek tesztelése
    - TestCollisionDetection\_GameOver: Ütközés detektálása és játék vége
    - TestCollisionDetection\_NoCollision: Nincsen fp ütközés felvéve
    - Test\_ScoringSystem\_AsteroidPassing: Pontozási rendszer tesztelése
    - Test\_PauseFunctionality\_...: Szüntelés tesztelése
    - Test\_HighScoreTracking\_...: a legmagasabb pontszám követése és frissítésének tesztelése
    - Test\_DifficultyProgression\_OverTime: Nehézségi szint megfelő változásának tesztelése
    - Test\_GameStatePersistence\_SetState: Játékállapot beállítása
    - Test\_AsteroidSizeVeriation\_Types: Aszteroidák méretválltozásának tesztelése