**FREE SPACE INVADERS**

1. [Introduction](#intro)
2. [Team](#contribution)
   1. [Members contribution](#contribution)
3. [Game Description](#gameDescr)
   1. [Gameplay](#gameplay)
   2. [Programming details](#programmingDetails)
      1. [Used Data structures](#dataStruct)
      2. [Object-Oriented Programming](#oop)
      3. [Used .NET classes](#classes)
      4. [Using an external file](#txtFile)
      5. [Exception handling](#excHandl)
4. [Team Foundation Server](#tfs)
5. **Introduction**

The console game FREE SPACE INVADERS was developed as a team project for the C# Part 2 Course in [Telerik Academy](http://academy.telerik.com/).

The following description contains information about the team members, the gameplay, brief programming details, and the URL of the Team Foundation Server Repository.

|  |  |
| --- | --- |
| 1. **Team**   Team "Goofy" with members:   * Cvetelina Valeva ([Cvetelinavaleva](http://telerikacademy.com/Users/cvetelinavaleva)) * **Deyan Yordanov** ([Drail](http://telerikacademy.com/Users/Drail)) * **Martin Nikolov** ([martin.nikolov](http://telerikacademy.com/Users/martin.nikolov)) * Milen Matev ([mmatev](http://telerikacademy.com/Users/mmatev)) * **Yoana Nikolova** ([yoana.v.nikolova](http://telerikacademy.com/Users/yoana.v.nikolova)) | **C:\Users\Yoana\Desktop\goofy_telerik2.jpg** |

* 1. **Members contribution**

|  |  |
| --- | --- |
| **Martin Nikolov** | * project architecture * objects spawing and movement * objects shooting and collision * objects drawing * ranking |
| **Yoana Nikolova** | * objects models * menu field integration * bug hunting * sounds |
| **Deyan Yordanov** | * game suggestion * initial game architecture |
| **Cvetelina Valeva** | Did not make any contribution to the project |
| **Milen Matev** | Did not make any contribution to the project |
| **Shared** between active members (1-2) | * exception handling * player lives * player score * bug fixing * colors |

1. **Game Description**
   1. **Gameplay:**

The player controls a horizontally moving laser at the bottom of the console, while three rows of 26 aliens (from bottom to top: first row – 8 yellow aliens, second row – 8 pink aliens, third row – 10 blue aliens) are moving above. There are three stationary defenses, where the player ‘hides’ the laser from the aliens’ shots. The aliens are moving horizontally back and forth and vertically toward the bottom of the console. A spaceship is occasionaly moving horizontally on top of the console in the opposite of the aliens’ direction.

The aim is to defeat all the aliens by shooting them with the laser. At the beginning the player has three lives and earns point by shooting the aliens or the spaceship. A yellow alien brings 50 points, a pink alien brings 25 points, a blue alien brings 10 points, and hitting the spaceship adds minimum 150 points to the result. Its point increase as aliens come closer over time. The aliens throw projectiles and destroy the defenses or decrease the player`s lives when they hit the laser.

If the aliens reach the bottom, the game is over. If the player has defended all the aliens before they ‘invade’, comes ‘the boss’ (big flying saucer). The smaller spaceship starts moving from left to right and vice versa. The boss is flying in different directions and hitting it brings 50 points. The player has to kill the boss with 30 shots. In this case, the player receives 1000 points, the game ends and the player wins.

* 1. **Programming details** *(оptional requirements are marked in blue)***:**
     1. **Used data structures:**
* Strings;
* Lists (such as List <Coordinates[]>, where Coordinates is a structure with two fields);
* One-dimensional arrays;
* Multi-dimensional arrays;
  + 1. **Object-Oriented Programming** -> the logic of the program is separated into:
* Structure (2);
* Individual classes (9);
* Inheritance;
* Methods (over 35);
* Threads (asynchronous programming);
  + 1. **Used .NET classes** (about 8):
* Collections.Generic Class (lists);
* Text Class (IO Encoding of characters);
* IO Class (read / write file);
* Threading Class (threads - asynchronous processes);
* Media.SoundPlayer Class (Sound effects);
* Random Class (random number generator);
* Drawing Class;
* Linq Class;
  + 1. **Using an external file:**
* Text file for reading and writing the ranking;
  + 1. **Exception handling:**
* Read / write a text file (ranking);
* Read / open audio effects;

1. **Team Foundation Server:**

<https://goofy.visualstudio.com>

You can use prepared test account:

* Username: [telerik-tfs@hotmail.com](mailto:telerik-tfs@hotmail.com)
* Password: Teleriktest1