

#####

Ninja Wars

#####

By Joseph Heller

Members:

ageorgie (Александра Георгиева)
achoreav (Ангел Раев)
mita4itu (Димитър Митев)
mihaikalichkov (Михаил Каличков)
tedidim12 (Теодора Димитрова)

#####

Game Objectives:

Ninja Wars is an RPG game. Main objective is to survive being shot by enemy space ships. There are more dangers like flying meteorites which also can take you damage and run you down. Fortunately, you can gain lives and health while taking the special items.

#####

General Requirements:

- 5 interfaces
- 15 classes
- 3 abstract
- 1 exception class
- 3 levels of depth in inheritance
- 1 polymorphism usage
- 1 structure
- 1 event - 4 events raised
- 1 design pattern - used Singleton

#####

Optional Requirements:

- Static members
- Constants
- LINQ
- Namespaces

#####

URLs of the GitHub repository:

<https://github.com/dnmitev/Team-Joseph-Heller>

#####

Implementation:

The game is implemented with the best practices for OO design. There are five interfaces that are responsible for the collision detection, object producing, rendering data on the console and using user interface. The basic class `GameObject` implements main fields, properties and methods for all of the game objects that are used. It is inherited by the `MovingObject` class which gives main logic for all moving objects in the game. It is done like this in order to implement later a `StaticObject` class which will hold data for non-moving objects. The `MovingClass` object is inherited by all objects currently implemented in the game. `Ship` is a base class to define the player and its direct enemies.

The coordinates to define position and speed are held into the structure `MatrixCoord`. There are some static classes responsible for the borders and the random values in the game.

Keyboard control is defined into `KeyboardInterface` class and there are several events raised while controls are pressed.

Main logic of the game execution is presented into the `Engine` class.

