Telerik Academy 08.04.2015

Teem Honeydew

|  |  |
| --- | --- |
| Zhenia Racheva | zhenia.racheva |
| Svetla Petkova | SvetlaPetkova |
| Nikolai Atanasov | ni4ka7a |
| Galin Ivanov | reddog |
| Kaloian Iliev | iliev.kalo |
| Teodor Mihov | teodor.mihov.9 |
| Daniel Karadaliev | Daniel\_Karadali |
| Tsvetanka Chipilova | zvet80 |
| Hristo Nenov | hristo.nenov |

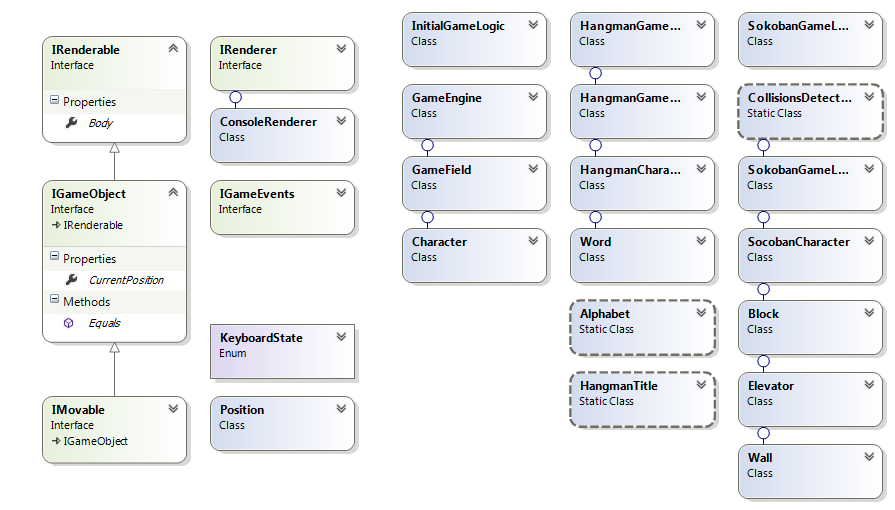
The goal of the current project is to implement simple puzzle games, accessible and played trough console interface.

The games have single game menu (Initial Game) giving the user a choice of several simple logical games to play. This project enables users to measure their abilities in solving different problems using their logic, memory and space orientation or just to pass the time.

**Games** implemented are: **Sokoban and Hangman**

Both games are accessible troughh common game menu from the console.

**Git repository URL: https://github.com/SvetlaPetkova/HoneyDew\_TW**

**Class Diagram**

For each game, a separate class is used to manage the game logic, each of the game objects and in the case of Sokoban – the collisions in game.

Static classes is used to display a font for the English alphabet (Hangman game) and other inscriptions.

**Game objects**

Game objects are objects used to carry out the logic of the game and to be display. All of them can be rendered on the console. Objects that cannot be moved by user implement IGameObject interface, while the movable objects as Character and Block implement the IMovable interface.

**Game engines**

Those classes are used to start the games and call game logic, rendering of the game objects and pass events. The initial game engine and the game engines of Sokoban and Hangman pass control over the execution of the program between them, depending on the game progress and user input.

**Game Logic**

Game logic classes are used to process the input from the user, control the collisions, basically the represent the rules of the game.

**User controls:**

Initial game is controlled using the arrow keys. If the user moves the character in the left side of the field and presses enter, Sokoban sub-game is being started. If the character is in the right side of the field, Hangman game is started.

Sokoban sub-game is controlled using the arrow keys only. Hangman game uses letters as input. Both sub-games return control over the execution if ESC button is pressed.

Game start point is Game.Program.