Telerik Software Academy

C# Part 2

Teamwork project

Console based game

Team

Royal Griffin Team

Project

Flappy Griffin

Telerik Academy 2015/2016 **Date: 04.03.2015**

Summary:

1. Overview:

This documentation describes the technical details about the Royal Griffin Team project – Console game application "**Flappy Griffin**" based on the notorious "**Flappy Bird**". This project realization takes place during the C# Part 2 course of the Telerik Software Academy season 2015/2016.

2. Team members:

- Alexander Markov @Github - fr0wsTyl
- Kristian Terziev @Github - tercinho
- Ivan Nikolov @Github - Spurch
- Asen Gruncharov @Github – sunidzver
- Niko Roid @Github - NicoRoyd
- Viktor Ivanov @Github - vicho86
- Plamen Paunov @Github – Puncky

3. Technical details and realization:

The console game/project is composed of 4 .cs class files and 2 .txt text files total.

The .cs class files:

- FlappyGriffin.cs
- GamePlay.cs
- Obstacles.cs
- Objects.cs

The .txt files:

- TopScores.txt
- Config.txt

The **Obstacles.cs** and **Objects.cs** represent the classes that we use to initialize and generate the obstacles in the game scenario and the Griffin "**G**" symbol that moves through the screen and the obstacles. Those classes consists of the following methods:

- **Obstacle ChangeObstaclePosition(List<Obstacle> Obstacles, int i, out Obstacle oldObstacle)** – Method that gets a list of the current obstacles on the game field and changes their position.
- **PrintObstacle(int x, int y1, int y2, int y3, int y4, int y5, int y6, string c, ConsoleColor color = ConsoleColor.Green)** – Method that prints the obstacles on the screen during the game.
- **GenerateDifferentObstacles(int chance, int playFieldWidth, List<Obstacle> Obstacles)** – Method that generates different kind of obstacles during the game.
- **Objects CreateGriffin(string griffin, int x, int y, ConsoleColor color = ConsoleColor.White)** – Method that creates an instance of the Griffin "G" symbol on the given coordinates on the screen.
- **Objects CheckGriffin(Objects griffin, out ConsoleKeyInfo keyPressed)** – Method that checks if the user is moving the Griffin symbol and changes the symbol coordinates accordingly.

The **GamePlay.cs** class holds most of the program logic, methods and variables that we use during the game. This class initializes the game field using the **Config.txt** to read the initial settings. It also reads the current top score player with its score from the **TopScore.txt** and initializes the need variables accordingly. The **GamePlay.cs** consists of the following methods:

- **WindowsSize(string ConfigFile)** – Method that reads the Config.txt file and initializes the game field.
- **PrintStringOnPosition(int x, int y, string c, ConsoleColor color = ConsoleColor.Gray)** – Method that prints the string values we need during the game play.
- **ReturnRandomValue(int minRange, int maxRange)** – Method that returns a random value in a given range that we use during the game.
- **PrintInfo(long score, int lives, long topScore, string userNameBestScore, string userName)** – Method that prints the info about the current players score and the top player score.
- **ReadScore()** - Method that reads the top score player and score from the TopScore.txt file.
- **WriteScore(string ValueToWrite)** – Method that writes the top player and it's score in the TopScore.txt
- **PlayMusic(string whichSound)** – Method that plays sounds during the game.

The **FlappyGriffin.cs** holds the base logic that controls the game play and realize heavily on the **GamePlay.cs** methods and variables to do it's job. **FlappyGriffin.cs** controls and checks the movement of the Griffin symbol, obstacles impact conditions, scoring etc. This class also holds the static strings that refer to the .txt files we are using. Basically the whole **FlappyGriffin.cs** logic is based on calling variables and methods of the other three classes.

4. Additional notes:

This project was realized using the Git based **GitHub** version control system @ https://github.com/fr0wsTyl/RoyalGriffin

Team members where using both **TortoiseGit** and the **GitHub** windows GUI client.