Team “Kitana”

Gergana Georgieva (gerchenceto)

Aleksandar Damyanov (Aleksandar.Damyanov)

Andrei Traikov (andrei_pl)

Galimir Gachev (djungalio)

Ivan Dimitrov (Ivan.Dimitrov.bg)

Miroslav Dimitrov (streetboyyy)

Nikola Stefanov (nikola.stefanov)

Radoslav Angelov (half.human)

Simeon Cholakov (sboymc)

Project Description

Our goal was to develop a game that is original, interesting and educative.

The game represents an English language test, where the player has a sentence consisting of words in a mixed order. In order to improve both their knowledge and their score players have to put the words in the correct order using the keyboard arrows. In order to achieve this they have to beware of the stones meant to make the game more attractive and difficult. If a word reaches a stone it stays on top of it resulting in lost point. The player also has a hint that can be used only once for each sentence which will tell them the exact order of the current word. Once the player has placed all words (correctly or not) they will see the correct sentence and word order displayed at the bottom of the screen.

The console is divided into two parts – the main where all the magic happens and a menu at the right side where the player can see the name they have entered before starting the game, the time left to finish it, their current and all game (total) score.

Once the player has completed a sentence they will hear the 'Game over, man ' and can start another one choosing from the 3 levels: Easy (4-word sentences), Medium (5-word sentences), Difficult (6-word sentences).

Good luck!

General Requirements

- 6 **multi-dimensional array:** string[,] *GetRow*; string[,] *output*; string[,] *orderedAndShuffleWord*; int[,] *hintBox*; int[,] *stoneBox*; bool[,] *isOccupied*
- 6 **one-dimensional arrays:** string[] *Shuffle*; string[] *shuffledArray*; string[] *dificultyRow*; string[] *words*; string[] *shuffled*; string[] *whichNumber*
- **Methods**(separating the application's logic): *ChooseDifficulty*(): string; *PrepareGame*(): void; *Shuffle*(string[]): string[]; *GetRow*(string): string[,]; *ResetGame*(string, ref int, string): void; *GameOver*(ref string, ref int, string): void; *MoveWord*(ConsoleKeyInfo, string): bool; *endPoints*(string): void; *countDown*(): void; *countDownSound*(): void; *stoneWordSound*(): void; *gameOverSound*(): void; *wordEndSound*(): void; *PrintAtPosition*(int, int, string, [ConsoleColor],[ConsoleColor]): void; *RandomGenerator*(): int; *Hint*(int,int): void; *Stone*(int,int): void; *WordsOnPlate*(ref bool[,], int,int, ref string, ref int, string, string): bool; *AddWordToWindow*(bool[,], int, int, ref string, ref int, string):void; *Main*(): void
- 5 **existing .NET classes**(like *System.Math* or *System.DateTime*): *System.Console; System.Random; System.ConsoleColor; System.ConsoleKeyInfo; System.DateTime; System.TimeSpan*
- 4 **exception handlings:** FileNotFoundException; IndexOutOfRangeException; ApplicationException; Exception.

- 3 **uses of external text file:** playerNamePath = @"../../filesTXT/PlayerName.dat"; name = "filesTXT/" + dificulty + ".dat"; readPlayerName = @"../../filesTXT/PlayerName.dat"

Additional Requirements

We have used TFS and Codeplex in order to track changes and for team collaboration. Our project can be found at: https://kitana.codeplex.com/

Optional Requirements

- **Object Oriented Programming > classes:** *Timer, Score, Player, SaveAndReadToFile; Validation; Kitana*
- **Serialization:** *playerScore*
- **Sound effects:** *gameOverSound; countDownSound; stoneWordSound; wordEndSound*

Non-Required Work

Completely finished project – It is indeed. However, we are full of ideas and features that could be implemented in Kitana 2.0.