|  |
| --- |
|  |
| GAME “GRIFFIN” |
| C# - PART 2, Team Work Project Final Report |
|  |
| **Telerik Software Academy** |
| **26-Feb-15** |

|  |
| --- |
|  |

Играта „Падащи букви“ е разработена като отборна работа в рамките на обучението по учебен курс C# - Part 2” в Софтуерна Академия Телерик – София, България.

# Състав на отбора „GRIFFIN”:

|  |  |  |  |
| --- | --- | --- | --- |
| N | Име, Презиме | Потребителско име | Студентски Номер |
| 1. | Камен Тодоров | kamen\_t | ??????????????? |
| 2. | Васил Вълков | BaSk3T | ??????????????????? |
| 3. | Нико Стоянов | Zenix | ????????????????? |
| 4. | Николай Николов | nikolay.nikolov | ???????????????? |
| 5. | Ангела Тенева | ellapt | 1000563 |
| URL на проекта: https://github.com/Zeneex/TSA15-CS2-TW-Griffin | | | |
|  | | | |

(картинката на грифона) ?

# 1. Описание на проекта

Целта на проекта е разработване на игра, която има за цел да затвърди знанията на участниците, придобити в курса.

Създадената от отбор Griffin игра представлява конзолно приложение, при което играчът (три последователни букви G - от Griffin, названието на отбора) се движи хоризонтално наляво-надясно в долната част на конзолата. Вертикално надолу падат букви от азбуката, на групи от 1 до 4 броя, като целта е играчът да "хване" последователно буквите, съставящи определена (по негов избор) дума на английски език. Буквите, дължината на "групите", както и разстоянието между тях се генерират в процеса на играта с помощта на генератор на псевдослучайни числа. Скоростта на падане с развитието на играта се повишава.

Отчетена е по-ниската вероятност на "случване" на гласните букви, които в английската азбука са само 5 на брой. За тази цел в изходния масив от букви, които се подават на вход на генератора за случайни числа, гласните се повтарят по 3 пъти, разпределени равномерно между съгласните букви.

Движението на играча се извършва посредством стрелките за движение наляво и надясно, а моментът, в който е завършено конструирането на определена дума, се фиксира с натискане на клавиша Enter.

По-нататък "думата" се подлага на проверка за това, дали тя съществува във външен файл-речник и ако съществува, играчът получава точки; в случай, че думата не съществува в речника, играчът се наказва. Съществува и възможност за "хващане" на бонуси, за което се получават допълнителни точки. ?????????

Конзолата е разделена на две части, по-голямата от които е игралното поле, а вдясно от него се изобразява менюто с присъщите за многопотребителска игра бутони, както и две таблици с постиженията на най-добрите и най-лошите играчи. Резултатите се записват във файлове , така че играта може да бъде прекъсната от даден играч и на по-късен момент да бъде възстановена.

# 2. Приложението съответства на основните изисквания както следва:

* Използван е двумерен масив ????
* Два масива от тип List<GameObject>, където GameObject е структура, съдържаща координатите, символа и цвета на движещите се обекти - съответно за генерираните и текущите букви;
* Масив от тип char[], съдържащ изходните букви от латинската азбука, с добавени редундантни (допълнителни) гласни букви;
* Масив от тип ConsoleColor [], съдържащ използваните в играта цветове на букви и символи;
* Два масива от тип List<string> с данни съответно за най-добрите и най-слабите играчи;
* Следните 14 метода, които допринасят за доброто структуриране и читаемост на програмата: ReadInfoFromFile, PlayerInfo, PrintMenu, PrintOldStats, CheckIsInClasation, WriteToFiles (обслужващи менюто); CheckForCollision, MoveLetters, MoveGriffin, InitializeGriffin, DrawObjectOnPosition, DrawStringOnPosition (управляващи движението на буквите, на играча и обработката на колизиите); GenerateLetterChain ("залепва""хванатите" букви една след друга, за да генерира думата за проверка) и FindWord (проверява наличността на думата във файла-речник wordlist.txt, като връща резултат true/false);
* Използвани са широко класовете на .NET Framework Class Library, например: System.Console, System.Random, System.Threading, System.Text (за StringBuilder), System.IO, System.Exception, System.Collections.Generic, System.DateTime;
* Обработка на 4 вида системни изключения (exceptions) - при обмена с файла-речник, като са подбрани най-вероятните, според вероятността на случване (използва се нотацията try...catch); при невалиден аргумент (прекалено дълга дума)програмата хвърля (throw) дефинирано от програмиста изключение ArgumentException("The word is too long!");
* Използвани са 3 външни текстови файла: 1) wordlist.txt - съдържа речника с еталонните думи; 2) BestPlayers.txt и WorstPlayers.txt - за съхраняване и отчитане на съревнованието между отделните играчи.

# Допълнителни изисквания

1. Проектът Griffin е разработен и се съхранява изцяло в хранилище на базата на Git, с активен обмен и сътрудничество от страна на участниците през целия период на разработка. Включително в хранилището се съхраняваха идеите за това, какво да представлява отборната игра с цел вземане на оптимално и консенсусно решение.

Както е отбелязано по-горе, URL на проекта е следният:

https://github.com/Zeneex/TSA15-CS2-TW-Griffin

2. Настоящият документ представлява кратко описание на груповия проект.

# Опционални изисквания

1. Използвани са следните средства на ООП:

- структура GameObject, която съхранява координатите по X и Y, ASCII кода и цвета (ConsoleColor) на движещите се обекти;

- класът Player, с помощта на който се създават инстанции на отделните участници в играта, съхраняващи техните имена, събрани точки, дата на последна игра и последна съставена от тях дума.

2. Използвани са елементарни звуци на конзолата при преминаване през различните стадии на менюто. ???????

# Завършеност на проекта:

Играта е завършена и работеща. Изпълнени са всички изисквания по проекта, включително някои от опционалните.