|  |
| --- |
|  |
| 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 отбор „GRIFFIN”:

|  |  |  |
| --- | --- | --- |
| N | Име, Презиме | Потребителско име |
| 1. | Камен Тодоров | [kamen\_t](https://telerikacademy.com/Users/kamen_t) |
| 2. | Ангела Тенева | [ellapt](https://telerikacademy.com/Users/ellapt) |
| 3. | Васил Вълков | [BaSk3T](https://telerikacademy.com/Users/BaSk3T) |
| 3. | Нико Стоянов | [Zenix](https://telerikacademy.com/Users/Zenix) |
| 4. | Николай Николов | [nikolay.nikolov](https://telerikacademy.com/Users/nikolay.nikolov) |

# URL на хранилището на проекта:

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



# 1. Описание на проекта

Целта на проекта е разработване на конзолна игра, с което да се затвърдят 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 английската азбука, на групи от 1 до 3 броя (вкл.), като целта е играчът да събере последователно буквите, съставящи смислена дума (по негов избор) 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 се повишава с развитие на играта.

Отчетена е по-ниската вероятност за случаен избор на гласните букви (от страна на играта), които в английската азбука са само 6 на брой (A, E, I, O, U, Y). 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 с отнемане на точки. Съществува и възможност за "хващане" на бонуси, за което се получават допълнителни точки. [*или опции (неразработено)*]

Конзолата е разделена на две части, по-голямата от които е игралното поле, а вдясно от него се изобразяват статистиките на играта – име и текущ резултат на играча, както и два списъка, представящи постиженията на най-добрите и най-лошите играчи. Резултатите се записват във файлове. [*Играта може да бъде прекъсната от даден играч и на по-късен момент може да бъде възстановена (неразработено)*]

# 2. Приложението съответства на основните изисквания както следва:

* Използван е двумерен масив (низова матрица) за съхранение и печат на загл. екран;
* Два масива от тип **List<GameObject>**, където **GameObject** е структура, съдържаща координатите, символа и цвета на движещите се обекти - съответно за новогенерираните и за 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 четимост на програмният код:
  + обслужващи менюто: **ReadInfoFromFile**, **PlayerInfo**, **PrintMenu**, **PrintOldStats**, **CheckIsInClasation**, **WriteToFiles**;
  + управляващи движението на буквите, на играча и обработката на колизиите: **CheckForCollision**, **MoveLetters**, **MoveGriffin**, **InitializeGriffin**, **DrawObjectOnPosition**, **DrawStringOnPosition**;
  + за събиране на думата в низ: **GenerateLetterChain**;
  + за реяникова валидация на подадената дума: **FindWord**;
* Използвани са широко класовете на .NET Framework Class Library, например:
  + **System.Console, System.Random, System.IO.File, System.Exception, System.DateTime, System.Text.StringBuilder, System.Collections.Generic.List<T>;**
* Обработка на 4 вида системни изключения (exceptions):
  + при обмена с файла-речник, като са подбрани най-вероятните (използвайки конструкция **try...catch**);
  + при невалиден аргумент (прекалено дълга дума) програмата хвърля дефинирано от програмиста изключение **ArgumentException("The word is too long!")**;
* Използвани са 3 външни текстови файла:
  + **wordlist.txt** - съдържа речника с еталонните думи;
  + **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.
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.

1. Използвани са елементарни звуци на конзолата за някои събития в играта.

# Завършеност на проекта:

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