Exercice 3.6 : Jeu du pendu L'algorithme lit un mot proposé par un premier joueur.

Ce mot a une longueur minimum de 5 caractères (à contrôler). L'algorithme affiche ensuite le mot où toutes les lettres sauf la première et la dernière sont remplacées par un tiret. Un deuxième joueur propose des lettres une à une. Chaque fois que la lettre se trouve dans le mot, l'algorithme remplace les tirets qui remplaçaient cette lettre et réaffiche le mot. Le second joueur a droit à un maximum de 6 essais pour retrouver toutes les lettres.

wordToFind = Console.ReadLine();

} while (wordToFind.Length < minWordLength);

Console.Clear();

foundLetters = FirstAndLastLetterOfWord(wordToFind);

Console.WriteLine($"{new string(foundLetters)}");

do

{

noMistake = false;

Console.WriteLine("Enter a letter, please ");

suggestedLetter = Console.ReadLine()[0];

for (i = 0; i < wordToFind.Length; i++)

{

if (suggestedLetter == wordToFind[i])

{

foundLetters[i] = suggestedLetter;

noMistake = true;

}

}

if (!noMistake)

{

attemptsCounter = attemptsCounter + 1;

}

if (new string(foundLetters).Equals(wordToFind))

{

noVictory = false;

Console.WriteLine("you win. the word was: " + wordToFind);

}

else

{

Console.WriteLine($"{new string(foundLetters)} Attempts : {maxAttempts - attemptsCounter}");

}

} while (attemptsCounter < maxAttempts && noVictory);

if (attemptsCounter >= maxAttempts)

{

Console.WriteLine("failled");

}

}

private static char[] FirstAndLastLetterOfWord(string word)

{

char[] result;

const char dash = '-';

int i;

result = new char[word.Length];

result[0] = word[0];

result[word.Length - 1] = word[word.Length - 1];

for ( i = 1; i < word.Length-1; i++)

{

result[i] = dash;

}

return result;

}

}

}