



CIPFP Mislata

Centre Integrat Públic
Formació Professional Superior

Pràctica Git Avançat

Autor: Joan Puigcerver Ibáñez

Correu electrònic: j.puigcerveribanez@edu.gva.es

Curs: 2023/2024

Llicència: BY-NC-SA



Objectius

L'objectiu d'aquesta pràctica és aplicar els coneixements de Git avançats a un entorn hipotètic.

Requisits

Per poder realitzar la pràctica, cal tindre un repositori anomenat **CognomNom-ED** en l'organització del mòdul.

- Realitzar el [Preparació entorn](#).

1. Enunciat

Ens han demanat implementar la solució del següent problema (<https://acceptaelreto.com/problem/statement.php?id=634&cat=152>)

Albert Una es aficionado a los juegos de ritmo y orgulloso poseedor de reliquias como Guitar Hero, Rock Band o Guitar Master, con todos los cachivaches de las distintas plataformas en las que se publicaron.

Pero su verdadera afición es la bandurria. Poder tocar en su consola de videojuegos canciones clásicas de la tuna como "Clavelitos", "Cielito lindo" o "De colores" le haría recordar otros paisajes y otros tiempos... y conseguir muchos más puntos que con cualquier otro juego.

De modo que, ni corto ni perezoso, está aprendiendo a programar para poderse hacer su propio Bandurria Hero. De momento, está atascado con el cálculo de la puntuación por culpa de los combos. Quiere que si se consigue tocar correctamente una nota se sumen 10 puntos. Si a continuación se acierta con la siguiente nota, esta se premie con 20 puntos, la siguiente con 30 y así sucesivamente, incrementando la valoración de 10 en 10, siempre que no se falle. En cuanto se pulse incorrectamente una cuerda o se salte una nota, el combo se termina y la siguiente nota correcta volverá a ser valorada con 10 puntos.

1.1. Entrada

El programa deberá leer, de la entrada estándar, un primer número indicando cuántos casos de prueba vendrán a continuación.

Cada caso de prueba consiste en una línea compuesta de una secuencia de "O" (vocal "o" mayúscula) y "." indicando, respectivamente, una nota correcta o un fallo. Todos los casos tienen al menos un carácter y ninguno tiene más de 10.000.

1.2. Eixida

Por cada caso de prueba se indicará la puntuación que se debe dar al jugador teniendo en cuenta la valoración de los combos, donde la primera nota de una serie consecutiva de aciertos se valora con 10 puntos y las siguientes notas de la serie proporcionan 10 puntos más que la anterior.

1.3. Entrada d'exemple

```
4
0.0.
00...
00.00..
.0000...
```



Explicar



1.4. Eixida d'exemple

```
20
30
60
100
```



2. Passos a seguir

Per resoldre el problema anem a seguir els següents passos:

1. En la branca principal `main`, crea el commit **P4: Esquelet** on s'afegeix els següent fitxer al package `ud4.practices`.

```
package ud4.practices;

import java.util.Scanner;

public class BandurriaHero {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
    }
}
```

 Explicar



2. Crea la branca `p4/entrada`.
3. Crea la branca `p4/puntuacio`.
4. Des de la branca `p4/entrada`, fes el *commit* **P4: Definició mètode puntuacio()** amb els següents canvis:

```
diff --git a/BandurriaHero.java b/BandurriaHero.java
index 681ef2d..a0b00a8 100644
--- a/BandurriaHero.java
+++ b/BandurriaHero.java
@@ -3,6 +3,10 @@ package ud4.practices;
import java.util.Scanner;

public class BandurriaHero {
+ public static int puntuacio(String notesCorrectes){
+ return 0;
+ }
+
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
    }
}
```

 Explicar



5. Ens hem equivocat! Aquests canvis no es corresponen a aquesta branca. Des de la branca `p4/entrada`, revertix el *commit* anterior. Deixa el missatge per defecte.
6. Des de la branca `p4/entrada`, fes el *commit* **P4: Processa entrada** amb els següents canvis:

```
diff --git a/BandurriaHero.java b/BandurriaHero.java
index 681ef2d..55f163b 100644
--- a/BandurriaHero.java
+++ b/BandurriaHero.java
@@ -5,5 +5,10 @@ import java.util.Scanner;
import java.util.Scanner;

public class BandurriaHero {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
    }
}
```

 Explicar



```

public class BandurriaHero {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        + int nombreCasos = in.nextInt();
        +
        + for (int i = 0; i < nombreCasos; i++) {
        + String liniaBandurria = in.nextLine();
        + }
    }
}

```

7. Des de la branca `p4/puntuacio`, obté els canvis realitzats en el *commit* **P4: Definició mètode puntuacio()** creat anteriorment.
8. Des de la branca `p4/puntuacio`, completa el mètode en el *commit* **P4: Càlcul puntuació**:



Explicar



```

diff --git a/BandurriaHero.java b/BandurriaHero.java
index 0aeb593..848f8ce 100644
--- a/BandurriaHero.java
+++ b/BandurriaHero.java
@@ -4,7 +4,18 @@ import java.util.Scanner;

public class BandurriaHero {
    public static int puntuacio(String notesCorrectes){
- return 0;
+ int puntuacioAcumulada = 0;
+ int bonus = 10;
+ for (int i = 0; i < notesCorrectes.length(); i++) {
+ char actual = notesCorrectes.charAt(i);
+ if (actual == '0') {
+ puntuacioAcumulada += bonus;
+ bonus += 10;
+ } else {
+ bonus = 10;
+ }
+ }
+ return puntuacioAcumulada;
}

    public static void main(String[] args){

```

9. Incorpora la branca `p4/entrada` a `main` (sense fer `rebase`).
10. Incorpora la branca `p4/puntuacio` a `main` (sense fer `rebase`).
11. Des de la branca `main`, completa el programa en el *commit* **P4: Processa els casos d'entrada**:



Explicar



```

diff --git a/BandurriaHero.java b/BandurriaHero.java

```

```

index 562a437..5cb781e 100644
--- a/BandurriaHero.java
+++ b/BandurriaHero.java
@@ -24,6 +24,7 @@ public class BandurriaHero {

    for (int i = 0; i < nombreCasos; i++) {
        String liniaBandurria = in.nextLine();
        + System.out.println(puntuacio(liniaBandurria));
    }
}
}
}

```

12. Publica les branques `main`, `p4/entrada` i `p4/puntuacio`

13. Crea les següents etiquetes i publica-les:

- `P4` : en l'últim *commit* realitzat.
- `P4-Revert` : en el *commit* que revertim els canvis.
- `P4-CherryPick` : en el *commit* on s'ha realitzat el `cherry-pick`.

2.1. Estat final del fitxer

```

package ud4.practices;

import java.util.Scanner;

public class BandurriaHero {
    public static int puntuacio(String notesCorrectes){
        int puntuacioAcumulada = 0;
        int bonus = 10;
        for (int i = 0; i < notesCorrectes.length(); i++) {
            char actual = notesCorrectes.charAt(i);
            if (actual == '0') {
                puntuacioAcumulada += bonus;
                bonus += 10;
            } else {
                bonus = 10;
            }
        }
        return puntuacioAcumulada;
    }

    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        int nombreCasos = in.nextInt();

        for (int i = 0; i < nombreCasos; i++) {

```



Explicar



```
String liniaBandurria = in.nextLine();
System.out.println(puntuacio(liniaBandurria));
}
}
}
```

2.2. Estat final del repositori

```
jpuigcerver@fp:~/PuigcerverJoan-ED (main) $ gla
* 11ac44a - (10 minuts ago) P4: Processa els casos d'entrada - Joan Puigcerver (HEAD -> main,
origin/main, tag: P4)
* 28665c9 - (10 minutes ago) Merge branch 'p4/puntuacio' into main - Joan Puigcerver
| \
| * c138cd3 - (10 minutes ago) P4: Càlcul puntuació - Joan Puigcerver (p4/puntuacio, origin/p4/puntuacio)
| * 0196a4f - (10 minutes ago) P4: Definició mètode puntuacio() - Joan Puigcerver (tag: P4-CherryPick)
* | abbce40 - (10 minutes ago) P4: Processa entrada - Joan Puigcerver (p4/entrada, origin/p4/entrada)
* | d72608c - (10 minutes ago) Revert "P4: Definició del mètode puntuacio()" - Joan Puigcerver (tag: P4-
Revert)
* | b188750 - (10 minutes ago) P4: Definició del mètode puntuacio() - Joan Puigcerver
| /
* f377404 - (10 minutes ago) P4: Esquelet - Joan Puigcerver
```