



2.1_ACT3.DOCX



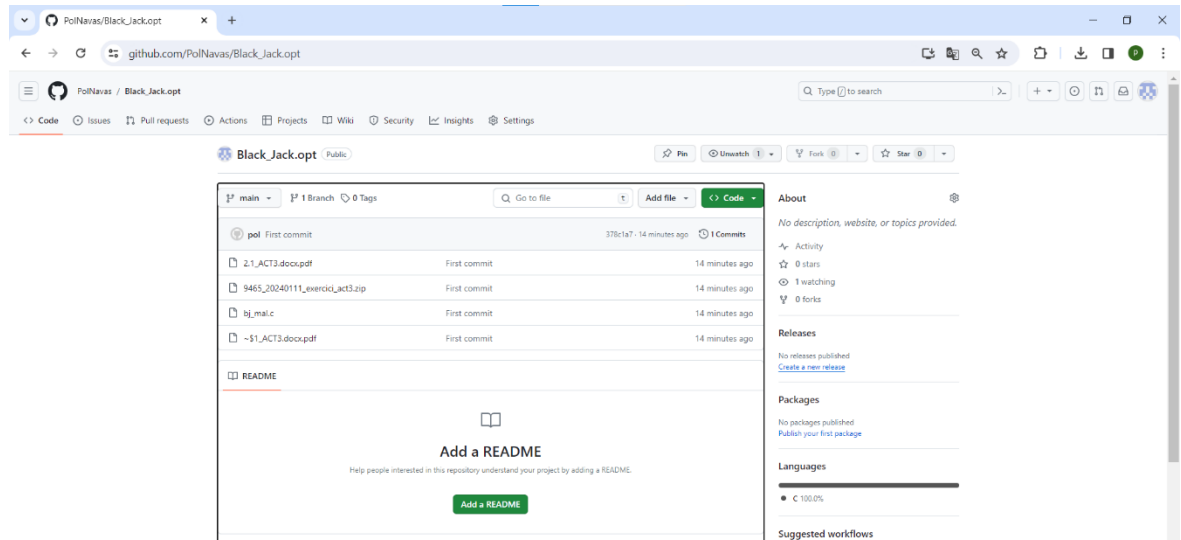
21 DE ENERO DE 2024
POL NAVAS

Mòdul 5: Entorns de desenvolupament

2.1_ACT3 : Optimitzar i refactoritzar



1. Utilitzar GIT



```

polna@DESKTOP-AH9VIKN MINGW64 ~
$ cd ../..

polna@DESKTOP-AH9VIKN MINGW64 /c
$ ls
'Recycle.Bin'/'      MSI/'      Users/
'WINDOWS.~BT/'      OneDriveTemp/  WEB/
'SwinREAgent'/'     Pokemon/'     WEB_amor/
'Windows.~WS'/'     'Program Files'/'  Windows/
'Archivos de programa'@ 'Program Files (x86)'/' XboxGames/
DAW/'               ProgramData/'     hiberfil.sys
'Documents and Settings'@ 'Recovery/'      msdia80.dll*
DumpStack.log        'Riot Games/'    oraclexe/
DumpStack.log.tmp    SteamLibrary/'   pagefile.sys
ESD/'                 'System Volume Information'/' steam/
Intel/'               UDEM/'           swapfile.sys

polna@DESKTOP-AH9VIKN MINGW64 /c
$ cd DAW/Entornos\ de\ desarrollo/UF2/

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2
$ ls
ACT1/  ACT3/      DAW1_M5_UF2_Control_versions.pptx.1.pdf  GIT/
ACT2/  APTS.docx  DAW1_M5_UF2_optimtzacio.pptx.1.pdf

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2
$ cd ACT3/

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$ ls
2.1_ACT3.docx  2.1_ACT3.docx.pdf  9465_20240111_exercici_act3.zip  bj_mal.c  bj_mal.exe*  bj_mal.o  '~$1_ACT3.docx'

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$ git init
Reinitialized existing Git repository in C:/DAW/Entornos de desarrollo/UF2/ACT3/.git/

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$ git tag -a V1.1 -m "Version 1.0"

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 156 bytes | 156.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PolNavas/Black_Jack.opt.git
* [new tag]          v1.1 -> v1.1

```

```

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$ git add .

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   2.1_ACT3.docx
        new file:   bj_mal.exe
        new file:   bj_mal.o
        new file:   ~$1_ACT3.docx
        deleted:    ~$1_ACT3.docx.pdf

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$ git commit -m "Cambio a ingles"
[main a04c795] Cambio a ingles
5 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2.1_ACT3.docx
create mode 100644 bj_mal.exe
create mode 100644 bj_mal.o
create mode 100644 ~$1_ACT3.docx
delete mode 100644 ~$1_ACT3.docx.pdf

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$ git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 172.81 KiB | 24.69 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PolNavas/Black_Jack.opt.git
378c1a7..a04c795 main -> main
branch 'main' set up to track 'origin/main'.

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$ git tag -a V1.2 -m "version 1.2 de cambio de idioma"

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 170 bytes | 170.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PolNavas/Black_Jack.opt.git
* [new tag]          v1.2 -> v1.2

polna@DESKTOP-AH9VIKN MINGW64 /c/DAW/Entornos de desarrollo/UF2/ACT3 (main)
$

```

2. Entendre el Codi:

- Analitza com es representa una carta i una baralla.
- Observa com s'inicialitza i baralla la baralla, i com es manegen les cartes del jugador.

3. Jugar i Observar:

- Compilar i executar el joc. Jugar unes quantes rondes per a entendre com funciona i trobar errors.

4. Identificar Oportunitats d'Optimització:

- Hi ha parts del codi que es poden escriure de manera més eficient?
- Hi ha funcions que poden ser factoritzades per a millorar la claredat o el rendiment?

5. Aplicar Canvis i Comparar:

- Facin canvis en el codi i observin com afecten el joc i el seu rendiment. - Considerin usar un perfilador per a identificar colls d'ampolla.

Lo primero que encontramos es que las variables están escritas en diferentes idiomas así que el primer paso va a ser cambiar el nombre de las variables para dejar todo el código en inglés, menos el texto que aparecerá por pantalla. Después de eso hacemos un commit para dejarlo todo preparado y poder empezar con la refactorización de código.

En este punto, lo más importante es analizar y entender el código, para mí fue bastante complicado, así que necesité ayuda para entender ciertas funciones y ciertas partes del código, pero un rato después entendí que hacía cada función y entendí cuáles eran las cosas que tenía que cambiar. Una de las cosas que me llamó la atención fue encontrar una función que solo hiciera un printf, ya que parar el código únicamente para escribir algo no me parecía lo más óptimo. Aparte una vez jugué un rato, vi que la sumas no se hacían bien así que eso es otro punto que tenía que refactorizar. Otra de las cosas que observé fue que los If no seguían el estándar al que estaba acostumbrado así que pensé el estandarizarlo, ya que no sería un gran cambio nivel optimización. Otra de las cosas que hice únicamente para mejorar como se veía lo que se muestra por pantalla fue añadir guiones a modo de línea para separar cada apartado cada vez que se repartían cartas.

En este punto quería refactorizar la función ShowCards, encargada de mostrarle al usuario, cuáles son las cartas que se le han repartido esta vez, ya que, como he dicho antes, una función que pare el código para únicamente hacer un 'printf', realmente no es lo más óptimo, el problema es que no acabo de entender bien el código y el cambio de variables

entonces a la hora de compilar me da un error, así que dejaré el apunte teórico de que eso habría el cambiarlo, de momento.

Otra de las cosas que quería llevar a cabo, ha sido el hecho de comentar en cada función de qué se trata y cuál es su funcionalidad.

A la hora de hacer las sumas, se equivoca, he estado probando un buen rato, pero no he encontrado un patrón que se repita, lo encuentro aleatorio. Revisando el código no acabo de entender algunos puntos en concreto, he probado a refactorizarlo usando otros métodos, pero conservando la esencia de la función, cambiando parámetros y demás pero tampoco he conseguido arreglarlo

Otro punto que me gustaría desarrollar es que en varios puntos del código podemos encontrar que hay condiciones dentro de bucles, por ejemplo: en la función InitializeDeck podemos encontrar un For con dos if dentro y en la función Main también podemos encontrar un conjunto de condiciones dentro del While. Es un punto que deberíamos cambiar ya que como se explica, hay varias maneras de no llamar a condiciones dentro de bucles para mejorar el código, por ejemplo: llamar a la condición antes de empezar, y si esa condición se cumple hacer un bucle, evitando ciertas iteraciones del bucle.

También me hubiera gustado haber utilizado los Perfiladores para encontrar los diferentes cuellos de botella que puede haber en el código. El único problema es que no he sabido cómo se utilizan y no encontrado la manera, pero tengo entendido que nos ayudaría a detectar ciertas partes donde el código se ralentiza debido a la falta de optimización de ciertas partes.

La conclusión que consigo sacar de todo esto es que entiendo la optimización del código, entiendo el proceso de entender mediante jugar, analizar resultados y los cambios que se deben hacer, el problema no viene de ahí, sino de la programación, hay ciertas partes del programa que no entiendo, cosas que no se como desarrollar y algunos detalles que imagino que no estoy enfocando desde el punto de vista correcto, así que podría asegurar que la parte analítica, teniendo en cuenta que es la primera vez que lo hago, no está mal, aunque debería mejorarse, mi fallo en esta práctica ha sido la programación.

(En mi repositorio de git puedes ver los cambios):

PolNavas / Black_Jack.opt

Q Type to search

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Commits

main

All users

All time

Commits on Jan 19, 2024

Pruebas sin exito

pol committed now

7b681e4

<>

Commits on Jan 18, 2024

Explicacion y analisis en word -1-

pol committed yesterday

8ee3214

<>

Estandarización 1.1 -if-

pol committed yesterday

f466302

<>

Commits on Jan 16, 2024

Cambio a ingles

pol committed 3 days ago

a04c795

<>

Commits on Jan 11, 2024

First commit

pol committed last week

378c1a7

<>