

TAREAS EXTRA-NAVIDAD

Para aquellos que habéis acabado las tareas y todo lo propuesto para navidad 2020/2021, os propongo algunos ejercicios para no estar parados todas las fiestas.

Tenéis libre opción de hacerlo como queráis, pero que más o menos haga lo básico de forma similar a los requisitos dados. (ANALISIS)

Hacer los programas básicos las personas que vais adelantados o a un ritmo superior, luego ya puedes hacer un segundo archivo con mejoras (aunque sepas más, haz versión básica por favor). Esto lo digo por poder enseñar vuestro código a alumnos que van más flojos y puedan entenderlo.

Además, ya empezamos a enlazar conceptos con Entornos de Desarrollo. Y será así para el resto del curso, van cogidas de la mano casi. (DISEÑO E IMPLEMENTACIÓN BÁSICA)

Luego me mandáis las versiones que tengáis (Version 1, Version 2, etc..), la básica y la que lleva vuestras mejoras, métodos más complejos o no vistos en clase y la refactorización de código que pudierais hacer de la versión 0.0 del programa o juego. (CONTROL DE VERSIONES)

Documentar con comentarios los pasos seguidos. (DOCUMENTACIÓN)

Y, además, las pruebas correspondientes de que vuestro programa hace lo correcto y funciona bien. (PRUEBAS)

Mejoras de una nueva versión, revisión de fallos, refactorización de código candidato, arreglo de bugs, etc. (MANTENIMIENTO)

Extra-1: Dotar de IA al tic, tac, toe

Dotar de inteligencia a la máquina del programa tic, tac, toe y darle las opciones de:

- modo fácil (usa la elección random de movimiento (TONTACA LA MÁQUINA))
- modo medio (la máquina juega a no perder (SE DEFIENDE))
- modo difícil (la máquina juega a ganar (ESTA ENCABRONADA))
- modo extremo (la máquina juega a ganar y no perder (TE QUIERE MATAR))

Extra-2: Realizar una fuerza bruta para romper cifrado Cesar

Se trata de hacer un programa que dado un mensaje codificado por el método Cesar, sea capaz de descifrarlo por fuerza bruta (probar todas las combinaciones posibles).

Como consejos se dan:

- Entender bien el cifrado Cesar (desplazamiento, aritmética modular, congruencia de datos o resta de longitudes para conseguir listas circulares)

- Trabajar con el alfabeto de 27 letras, ñ incluida, para dar con la clave y afinar el algoritmo. Luego cualquier alfabeto dado es válido, pero pongamos caso base el alfabeto que conocemos con ñ.
- ¿Si desplazo hacia delante cada letra de un mensaje con el cifrado Cesar, tal vez tendré que hacer pruebas a la inversa para calcular el desplazamiento que se aplicó al mensaje original y probar todas las combinaciones posibles...??

Ya te he ayudado mucho y espero que seas feliz haciendo tu primer hack de una clave Cesar



Extra-3: Realizar criptografía, pero con el código Vigenere

Se trata de realizar los mismos ejercicios que en la Tarea 3 – Criptografía, pero usando el código Vigenère.

Escribe la función `codVig()` que, dado un mensaje, un alfabeto y una clave (ahora una cadena de caracteres del alfabeto), devuelve el String resultante de codificar el mensaje según el alfabeto dado utilizando el método Vigenère con la clave proporcionada. El perfil de la función debe ser:

def codVig (msg, alf, clave)

La función deberá codificar uno a uno los caracteres del mensaje `m`, para lo cual se deberá invocar al método `desplazar`, habiendo calculado previamente el valor de desplazamiento `k` que le corresponde a cada carácter de `m`.

Para probar su correcto funcionamiento podemos usar este ejemplo:

String codVig(String m, String alf, String clave)

Cifrado.codVig(" Lo que estamos haciendo con el planeta es insostenible" ,
 "abcdefghijklmnopqrstuvwxyz" ,
 "bluej")

Cancelar Aceptar

(a)

BlueJ: Method Result

```
// Codifica con el metodo Vigenere el mensaje m usando la clave proporcionada,
// segun el alfabeto alf.
// @param m, mensaje a codificar
// @param alf, alfabeto a utilizar
// @param clave, clave del metodo Vigenere
// @return el mensaje codificado segun el metodo Vigenere
String codVig(String m, String alf, String clave)
```

Cifrado.codVig() returned:

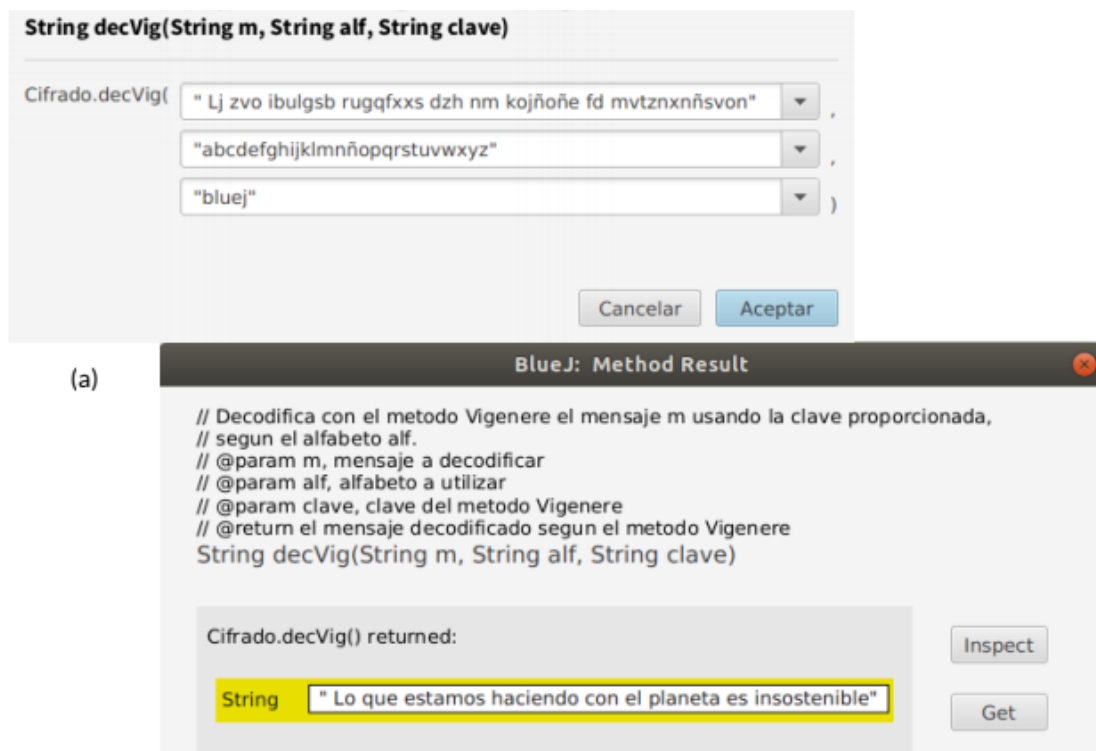
String " Lj zvo ibulgsb rugqfxxs dzh nm kojñoñe fd mvtznxnñsvon"

Inspect Get

A continuación, escribe la función decVig() que decodifica un mensaje utilizando el método Vigenère. El perfil del método debe ser:

def decVig (msg, alf, clave)

Ten en cuenta que en esta ocasión no es posible delegar en el método codVig para realizar la tarea del descifrado. En su lugar, hay que decodificar uno a uno los caracteres del mensaje m, calculando su desplazamiento personalizado según la posición del carácter correspondiente de la clave dentro del alfabeto original, y a continuación llamar al método desplazar, pero proporcionándole el alfabeto inverso.



Información de Vigenère:

https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re

Extra-4: Hacer el juego del Camello

La idea original del Camello proviene del Heath Users Group y fue publicada en el More BASIC Computer Games en 1979.

La idea es cabalgar con tu camello por el desierto mientras eres perseguido. Tienes que gestionar tu sed, el cansancio del camello y a que distancia estás de los nativos que te persiguen.

Aquí podéis inventaros la historia que queráis, es decir, el argumento del juego.

¿Tal vez un Star Wars?? ¿¿O una carrera de coches?? ¿¿Mario Kart con plátanos?? ¿¿Un matrix peleando o fluyendo por los mundos virtuales??

Lo que se os ocurra pero que tenga como básico lo que se pide en el enunciado (tienes que gestionar tu sed (cansancio, vida, coraza, etc.), el cansancio del camello (fuel, armadura nave, defensas del jugador, etc.) y a que distancia estás de los nativos que te persiguen (naves enemigas, enemigos, coches, etc.).

Como he dicho, libre elección, pero ser consecuentes, hacerlo básico y luego ya añadir mejoras y florituras.

Una de las cosas que se suelen poner son ventiscas del desierto random para afectar al avance del jugador, perseguidores o tal vez para ambos. También se suele poner un oasis random que aparece de vez en cuando para el jugador, donde reinicia todo lo que afecta a la vida, sed, cansancio del camello y demás valores a 0.

Ejemplo del juego ejecutado:

Bienvenido al Camello!

Has robado un camello para atravesar el gran desierto del Mobi.

¡Los nativos quieren que les devuelvas su camello y salen en persecución tuya! Tendrás que sobrevivir al desierto y correr más que los nativos.

- A. Beber de la cantimplora.
- B. Velocidad moderada hacia adelante.
- C. A toda velocidad hacia adelante.
- D. Para y descansa.
- E. Comprueba tu estado.
- Q. Salir.

¿Qué eliges? **C**

Has recorrido 12 millas.

- A. Beber de la cantimplora.
- B. Velocidad moderada hacia adelante.
- C. A toda velocidad hacia adelante.
- D. Para y descansa.
- E. Comprueba tu estado.
- Q. Salir.

¿Qué eliges? **C**

Has recorrido 17 millas.

- A. Beber de la cantimplora.
- B. Velocidad moderada hacia adelante.
- C. A toda velocidad hacia adelante.
- D. Para y descansa.
- E. Comprueba tu estado.
- Q. Salir.

¿Qué eliges? **E**

Millas recorridas: 29

Veces que has bebido de la cantimplora: 3

Los nativos están a 31 millas detrás de ti.

- A. Beber de la cantimplora.
- B. Velocidad moderada hacia adelante.
- C. A toda velocidad hacia adelante.
- D. Para y descansa.
- E. Comprueba tu estado.
- Q. Salir.

¿Qué eliges? **B**

Has recorrido 6 millas.

...y así hasta que... (**NOTA: Esto no se printea en el programa solo es para seguir la secuencia de ejemplo hasta que se dé una situación contemplada**)

- A. Beber de la cantimplora.
- B. Velocidad moderada hacia adelante.
- C. A toda velocidad hacia adelante.
- D. Para y descansa.
- E. Comprueba tu estado.

Q. Salir.

¿Qué eliges? **C**

Has recorrido 12 millas.

¡Los nativos se están acercando! ← Situación nueva ya que se da una condición

A. Beber de la cantimplora.

B. Velocidad moderada hacia adelante.

C. A toda velocidad hacia adelante.

D. Para y descansa.

E. Comprueba tu estado.

Q. Salir.

¿Qué eliges? **C**

Has recorrido 11 millas.

¡Los nativos se están acercando!

¡Has conseguido atravesar el desierto! ¡Has ganado! → millas >200 (Luego lo vemos)

En este ejemplo hemos usado mucho la opción C, luego como veremos, esto conlleva un gasto de sed y energía del camello.

Guía de programación:

Estos son los pasos para completar este taller. Tienes total libertad de modificar y ampliar el taller. Intenta jugarlo con tus amigos y familia.

1 - Crea un programa nuevo e imprime las instrucciones sobre la pantalla. Haz esto con múltiples declaraciones print. No vayas a usar una sola declaración print y múltiples caracteres \n para separar cada cosa en una línea.

Bienvenido al Camello!

Has robado un camello para atravesar el gran desierto del Mobi.

¡Los nativos quieren que les devuelvas su camello y salen en persecución tuya! Tendrás que sobrevivir al desierto y correr más que los nativos.

2 - Crea una variable booleana llamada hecho y establécela a False.

3 - Crea un bucle while que iterará mientras hecho sea False.

4 - Imprime lo siguiente dentro del bucle:

A. Beber de la cantimplora.

- B. Velocidad moderada hacia adelante.
- C. A toda velocidad hacia adelante.
- D. Para y descansa.
- E. Comprueba tu estado.
- Q. Salir.

5 - Pregúntale al jugador por la opción que elige.

6 - Si el jugador elige Q, entonces estableces hecho como True. Usando algo como usuario elige.upper(), en lugar de solo usuario elige en tu declaración if, puedes hacer que no hayan problemas con las minúsculas/mayúsculas.

7 - Pruébalo y asegúrate de que puedes salir del juego.

8 - Antes del bucle principal del programa, créate variables para millas recorridas, sed, cansancio del camello. Inicialízalas con cero.

9 - Crea una variable para la distancia recorrida por los nativos y establécela a -20 (Veinte millas por detrás.)

10 - Crea y establece un número inicial de sorbos en la cantimplora.

11 - Añade un elif en el bucle principal de tu programa que mire si el usuario está preguntado por su estado. Si es así, imprime algo como:

Millas recorridas: 0

Veces que has bebido de la cantimplora: 3

Los nativos están a 10 millas detrás de ti.

12 - Añade un elif en el bucle principal de tu programa que gestione si el jugador quiere detenerse por la noche. Si es así, reinicia el cansancio del camello a 0. Imprime que el camello se congratula de ello, y desplaza a los nativos en una cantidad aleatoria de entre 7 y 14 millas más o menos.

13- Añade un elif en el bucle principal de tu programa que gestione si el jugador quiere avanzar a toda velocidad. Si es así, avanza una cantidad aleatoria entre 10 y 20 incluidos. Imprime cuántas millas ha recorrido. Suma 1 a sed. Suma un número aleatorio entre 1 y 3 al cansancio del camello. Desplaza a los nativos una cantidad entre 7 y 14 millas.

14 - Añade un elif en el bucle principal de tu programa que gestione si el jugador quiere avanzar a velocidad moderada. Si es así, avanza una cantidad aleatoria entre 5 y 12, ambos incluidos. Imprime cuántas millas ha recorrido. Suma 1 a sed. Suma 1 al cansancio del camello. Desplaza los nativos una cantidad entre 7 y 14 millas.

15 - Añade un elif en el bucle principal de tu programa que gestione si el jugador quiere beber de la cantimplora. Si es así, asegúrate que queden tragos en la cantimplora. Si los hay, réstale 1 y devuelve la sed del jugador a cero. En caso contrario, que imprima error.

16 - En el bucle, que imprima "Estás sediento" si la sed del jugador está por encima de 4.

17 - Que imprima "Has muerto de sed" si la sed del jugador está por encima de 6. Establece hecho como true. Asegúrate de que tu código no imprima, a la vez, ¡"Estás sediento" y "Haz muerto de sed!" Utiliza elif cuando sea pertinente.

18 - Que imprima "Tu camello se está cansando." si el cansancio del camello está por encima de 5.

19 - Imprime "Tu camello ha muerto." si el cansancio del camello está por encima de 8. Al igual que con los anteriores pasos, imprime uno o el otro. Sería una buena idea incluir una comprobación en la variable hecho de forma que no fueras a imprimir que tu camello se está cansando, después de que ya hubiera muerto de sed.

20 - Si los nativos te han capturado, imprime que han capturado al jugador y finaliza el juego.

21 - ¡Si los nativos están a menos de 15 millas del jugador, que imprima "Los nativos se están acercando!"

22 - Si el jugador ha recorrido 200 millas por el desierto, que imprima que ha ganado y finaliza el juego. Asegúrate, eso sí, de que no hayan muerto antes de declararlos ganadores.

23 - Añade una oportunidad entre 20, de encontrar un oasis. Imprime que el jugador lo encuentra, que rellena la cantimplora, reinicia la sed del jugador y que el camello descanse.

24 - Prueba el juego, y afina los números para que se convierta en un reto y no en algo imposible.

25 - Arregla cualquier fallo que encuentres.

Pistas:

Recuerda que es una buena idea colocar líneas en blanco entre los bloques lógicos de tu programa. Por ejemplo, pon una línea en blanco después de las instrucciones y entre cada comando de usuario.

Queda mejor escribir (una cuestión de estilo) while not hecho: que while hecho == False:

Para evitar combinaciones confusas de mensajes, tales como "¡Has muerto de sed!" y "¡Has encontrado un oasis!" a la vez, usa el operador and si hace falta. Como en if not hecho and sed > 4 (recuerda cómo funcionan el and y el or, en caso de dudas repasa el módulo 3 correspondiente a dichos operadores, tampoco son obligatorios de usar en este ejercicio).

Extra-5: Hacer el juego hundir la flota

Se trata de hacer el juego hundir la flota con matrices, es decir, listas dentro de listas como en el tic, tac toe, indexación por fila y columna: `tablero[fila][columna]`. Un tablero de 10x10 es decir 100 (0-99) celdas, y tendremos varios barcos:

1 de 1 posición (Lancha)

1 de 2 posiciones (Crucero)

1 de 3 posiciones (Submarino)

1 de 4 posiciones (Buque)

1 de 5 posiciones (Portaaviones)

(ESTE NO HACERLE CASO DE MOMENTO, LO SUBIRE CUANDO TENGA CLARO QUE PEDIR. LO HARÉ EN ESTOS DÍAS Y VERE QUE REQUISITOS MÍNIMOS HA DE CUMPLIR ESTE EXTRA.)

EL RESTO DE LOS EXTRAS PARA LOS QUE VAIS BIEN Y TENEÍS ACABADAS LAS TAREAS DE NAVIDAD PODEIS DARLE CAÑA, ESTE LO PONDRE PARA EL 28 MÁS O MENOS SUPONGO, PERO YA SABÉIS DE QUE VA PARA IR RUMIENDO IN THE MIND 😊

<https://famiyaiycole.files.wordpress.com/2012/09/instrucciones-del-juego-de-los-barcos.pdf>