

Tarea Programada - Rust

Problema

En esta tarea usted implementará un juego de solitario en el lenguaje de programación Rust. La baraja de naipes consiste de 52 cartas, el valor/número de la carta será representado de la forma natural, excepto por el diez (10) que representaremos con la letra Z. Los colores de las cartas serán Corazones (C), Espadas (E), Florecillas o Tréboles (T) y Diamantes (D). A continuación una secuencia de un juego visto gráficamente y en la forma de texto que su juego lo desplegaría. El As de corazones rojo, por ejemplo, debe desplegarlo como ACr.

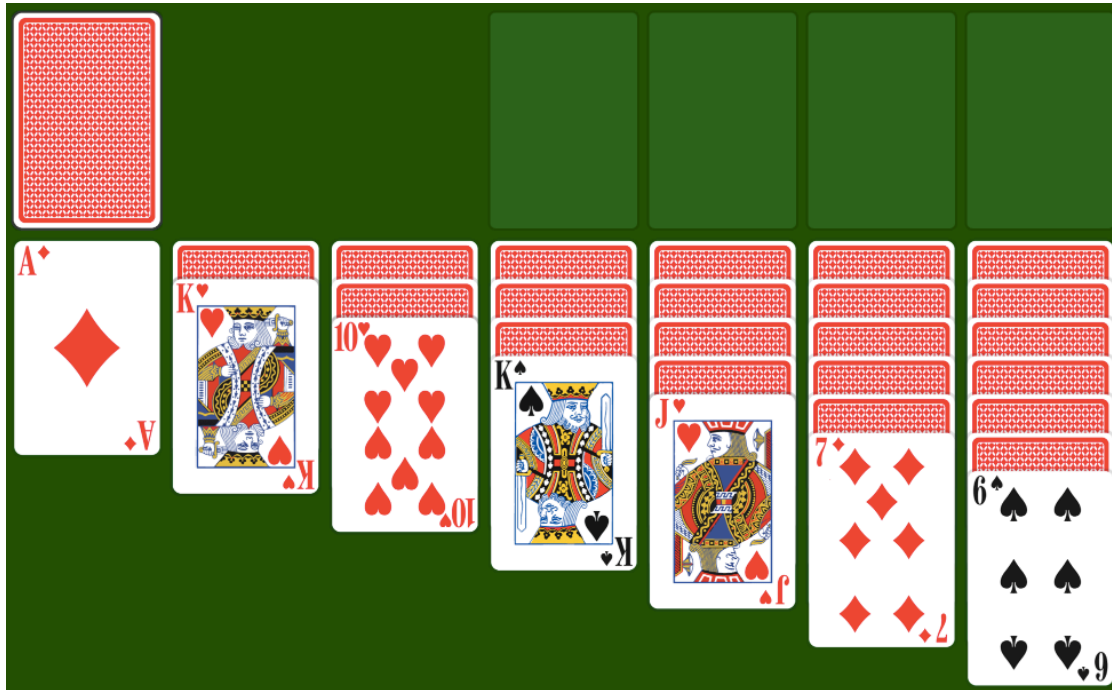
Puede implementar un solitario como Nestor, Punce, Pyramid, o Russian Bank.

NO Puede implementar el clásico juego de solitario, explicado a continuación para fines ilustrativos. La especificación de los solitarios los puede encontrar en la Wikipedia buscando, por ejemplo [nestor solitaire wikipedia] en Google.

La especificación dada a continuación se con fines ilustrativos para que vea las dos formas en que puede implementar su juego, en texto o con interfaz gráfico.

Cartas: 2 3 4 5 6 7 8 9 Z J Q K A

Colores C (Rojo), T (Negro), D (Rojo), E (Negro)

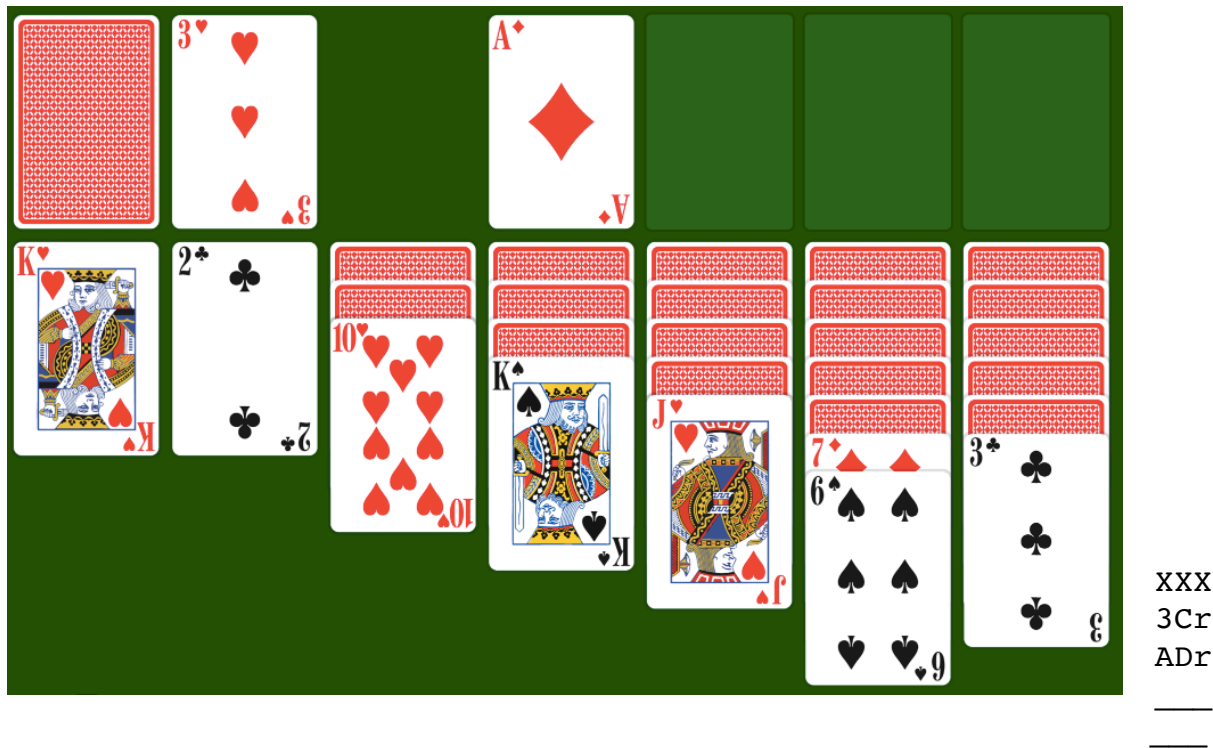


```

XXX  _____
ADr  XXX  XXX  XXX  XXX  XXX  XXX
      KCr  XXX  XXX  XXX  XXX  XXX
            ZCr  XXX  XXX  XXX  XXX
                  KEn  XXX  XXX  XXX
                        JCr  XXX  XXX
                              7Dr  XXX
                                    6En

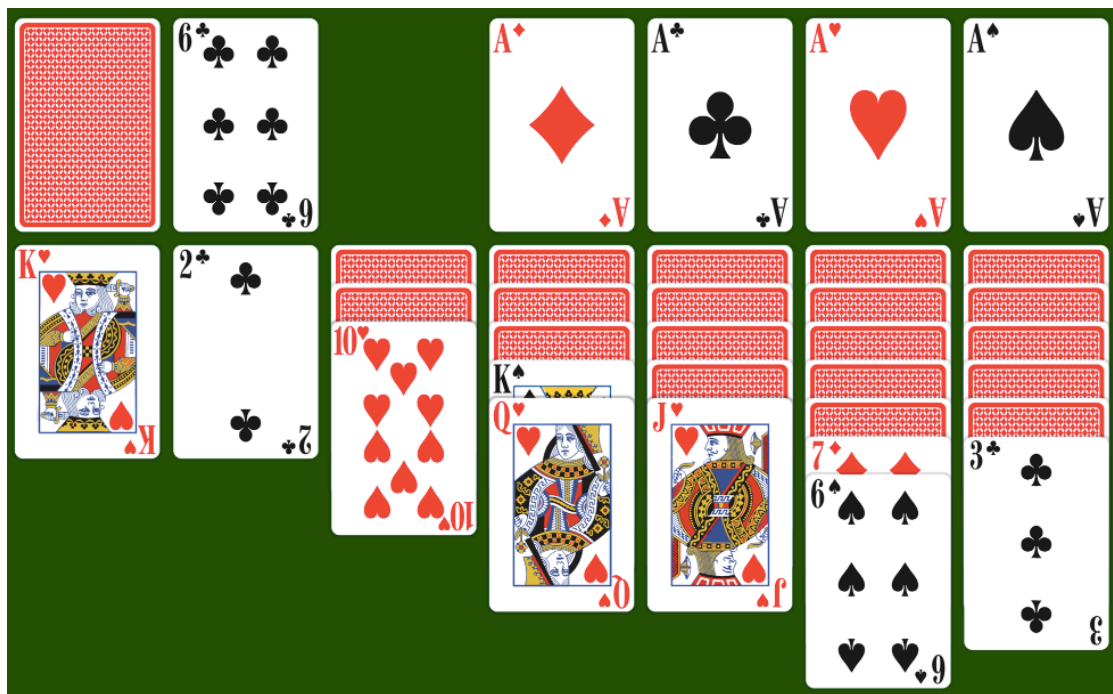
```

Más adelante en el juego ...



KCr 2Tn XXX XXX XXX XXX XXX
 XXX XXX XXX XXX XXX
 ZCr XXX XXX XXX XXX
 KEn XXX XXX XXX
 JCr XXX XXX
 7Dr 3Tn
 6En

Todavía más adelante ...

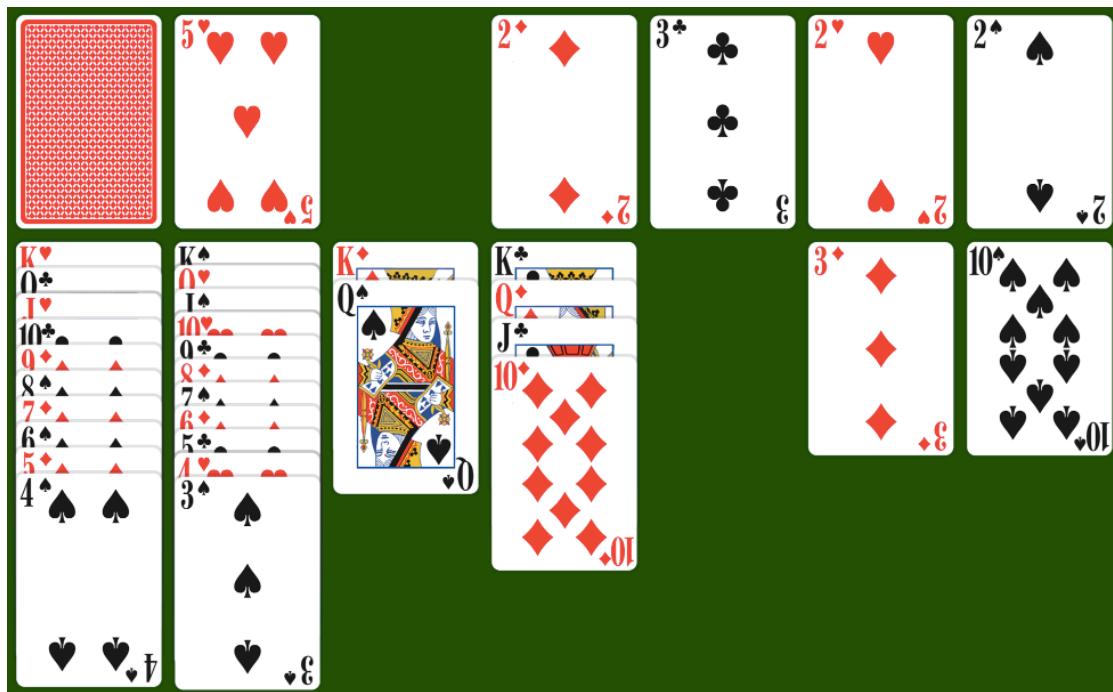


```

XXX 6Tn      ADr ATn ACr AEn
KCr 2Tn XXX XXX XXX XXX XXX
      XXX XXX XXX XXX XXX
      ZCr XXX XXX XXX XXX
            KEn XXX XXX XXX
            QCr JCr XXX XXX
                  7Dr 3Tn
                  6En

```

Casi terminando ...



XXX

5Cr 2Dr 3Tn 2Cr 2En

KCr KEn KDr KTn 3Dr ZEn

QTn QCr QEn QDr

JCr JEn JTn

ZTn ZCr ZDr

9Dr 9Tn

8En 8Dr

7Dr 7En

6En 6Dr

5Dr 5Tn

4En 4Cr

3En

Interacción (ejemplo, depende del juego)

Su programa debe aceptar los siguientes comandos desde el teclado:

1. <ESC> — Salir.
2. n/N — Juego nuevo (aleatorio)
3. <RET> — Carta nueva (de las que están cubiertas en la esquina superior izquierda)
4. 1,2,3,4,5,6,7 — Indicador de cual columna utilizar, lo que se debe hacer se saca por contexto (el programa debe deducirlo). Si hay más de una acción posible el programa debe con un prompt solicitar al usuario cual acción tomar.
5. u/U — Undo.

Su programa debe generar barajas aleatorias por defecto (bien barajadas). Debe aceptar como opción en la línea de comandos número entero, si entra un número por la línea de comandos este se debe utilizar de semilla para el generador de números aleatorios, esto con el fin de poder repetir juegos idénticos de una corrida a otra si es el caso.

Salidas

El programa debe generar el output tanto a pantalla como a un archivo de logging en el cual se guarda toda la interacción con el usuario. La interacción guardada en el archivo de logging debe contener el comando que hace el usuario, y guardar el estado del juego con todas las cartas hacia arriba, esto es, a diferencia de lo que se le despliega al usuario con cartas ocultas marcadas como XXX, debe guardar en el log el valor de estas también.

Puntos extra

Se darán 20 puntos extra en la tarea si le hace un interfaz gráfico.

Restricciones

No debe utilizar bibliotecas externas para sus estructuras de datos, debe utilizar los tipos de datos básicos del Rust: tablas de hash, vectores, matrices, o punteros.

Puede usar bibliotecas para la aleatoriedad, manejo de archivos, configuración (si es del caso), parseo de la línea de comando, para la secuencia de comandos para implementar Undo, y para algún interfaz gráfico.

No debe utilizar código de ningún tercero para resolver el problema de efectuar el solitario, se podrá una nota de 0 si se incurre en plagio.