

Trabajo práctico integrador

CIEN O ESCALERA



Se deberá realizar un juego de dados denominado CIEN o Escalera, que es un juego que puede desarrollarse para un jugador o dos jugadores.

El programa debe hacerse en un proyecto de CodeBlock en C/C++ y contar con un menú principal que permita iniciar un juego nuevo para un jugador, un juego nuevo para dos jugadores, mostrar la puntuación más alta y la opción de Salir del juego.

Juego nuevo para un jugador

Esta opción permite a un solo jugador realizar una partida completa.

Antes de comenzar a jugar deberá solicitar el nombre del jugador.

Una vez finalizada la partida, deberá mostrar el nombre del jugador y el puntaje obtenido.

Juego nuevo para dos jugadores

Esta opción permite a dos jugadores realizar una partida completa.

Antes de comenzar a jugar deberá solicitar el nombre de cada jugador. Una vez finalizada la partida, deberá mostrar el nombre del jugador que haya ganado y el puntaje obtenido.

Mostrar puntuación más alta

Esta opción debe mostrar el nombre del jugador que haya finalizado el juego con el mayor puntaje.

NOTA: como el sistema no tendrá la capacidad de guardar la puntuación en un archivo, los datos de puntuación serán los que surjan de las partidas realizadas en una misma sesión (desde que se ejecuta la primera partida hasta que se sale del programa).

Reglamento del juego

El objetivo del juego es obtener al menos 100 puntos en la menor cantidad posible de rondas, o sacar una escalera en un lanzamiento.

Una ronda está compuesta por 3 lanzamientos. Un lanzamiento consiste en tirar seis dados y evaluar sus valores para determinar el puntaje.

El puntaje de un lanzamiento está determinado por una serie de reglas que figuran en la sección **Combinaciones con puntos**.

El puntaje final de la ronda será el valor máximo de los puntos obtenidos en los 3 lanzamientos (no la suma de los tres, sino el más alto), con las siguientes excepciones:

- Si en un lanzamiento se obtiene escalera el jugador GANA EL PARTIDO en ese momento
- Si en un lanzamiento se obtiene una combinación de 6 dados con valor 6, el puntaje total de la persona se resetea a 0.

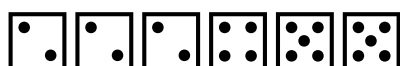
Para el caso de la opción para 2 jugadores, cada jugador debe completar una ronda (3 lanzamientos) alternativamente.

Combinaciones con puntos

Combinación	Nombre jugada	Puntaje otorgado
Menos de 6 dados con valores iguales Ejemplo: 6,5,5,2,2,1	Suma de dados	Suma de los valores de todos los dados Para el ejemplo sería $6+5+5+2+2+1=21$
6 dados iguales (menos para el número 6)	Sexteto X (X es el número del dado)	$X*10$ puntos
Escalera (1, 2, 3, 4, 5, 6) - <i>En cualquier orden</i>	Escalera	Gana la partida en esa ronda
Seis 6	Sexteto6- -	Resetea el puntaje total a 0

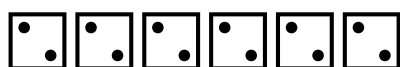
Por cada lanzamiento el programa tiene que elegir la combinación que corresponda.

Por ejemplo:



Suma de dados: $\rightarrow 20$ puntos

El puntaje a otorgar deberá ser de 20 puntos.



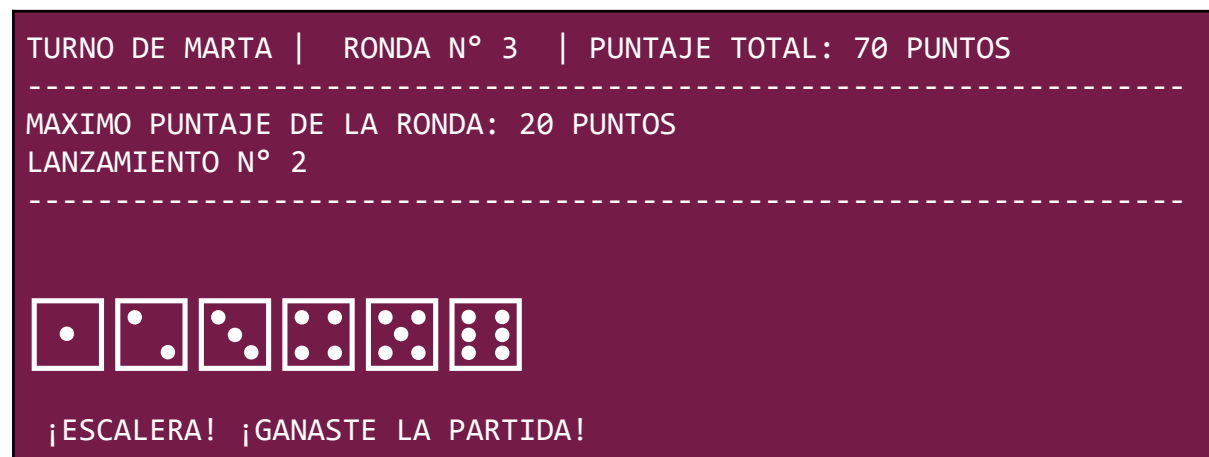
SextetoX de 2: $2*10 \rightarrow 20$ puntos.

Interfaz

Durante el turno de un jugador

Se deberá mostrar el nombre del jugador, el número de ronda, y el puntaje de ese momento. A medida que avanza la ronda se deberá mostrar el puntaje provisorio y la cantidad de lanzamientos.

Más abajo, se deberán listar los dados que conforman el lanzamiento y la combinación obtenida (el nombre de la combinación y el puntaje otorgado).

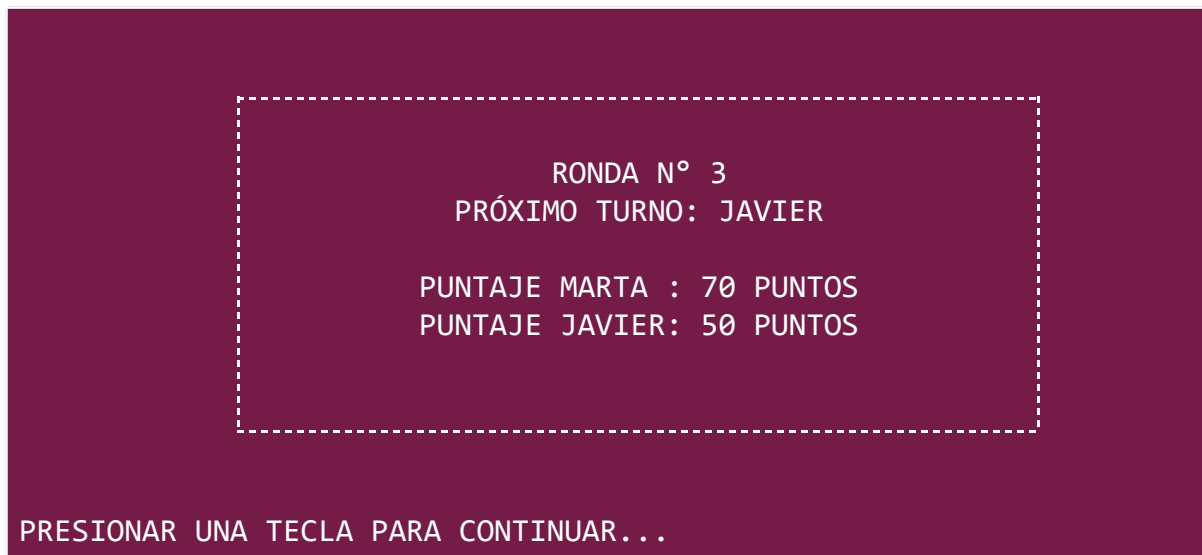


NO ES OBLIGATORIO QUE SE MUESTREN LOS DADOS POR PANTALLA. ES

SUFICIENTE QUE SE MUESTREN LOS NÚMEROS DE LOS DADOS.

Entre turnos

Cuando finaliza un turno y comienza el otro, deberá mostrar una pantalla que indique a quién le corresponde el próximo turno, el número de ronda y los puntajes de ambos jugadores. En el caso de que sea un juego para un jugador, debe indicar el número de ronda, el nombre del jugador y el puntaje.



Ganador

Cuando un jugador gana la partida, se deberá mostrar una pantalla que indique el nombre y apellido de quien ganó y la cantidad de rondas que le llevó ganar la partida.

Gana la partida quien haya obtenido los 100 puntos primero o escalera en un lanzamiento.

Características opcionales

1 - Existe la posibilidad de empate. Esto ocurre cuando ambos jugadores logran los 100 puntos o más en la misma ronda. En dicho caso se puede desempatar contando la cantidad de lanzamientos y favoreciendo a quien obtuvo el puntaje máximo en la menor cantidad de tiradas.

2 - Predeterminar una cantidad de rondas al inicio de la partida. Si ningún jugador obtuvo los 100 puntos al finalizar las rondas, el ganador de la partida será aquel que haya obtenido el puntaje más alto.

Nivel obligatorio para aprobar: Modo 1 jugador

Para acceder a la puntuación máxima:

- Modo de dos jugadores
- Características opcionales
- Solicitar los nombres de los jugadores

NOTA: el programa debe tener una opción (MODO SIMULADO) para cargar los valores de los dados de manera manual (se ingresan por teclado los valores de cada dado), en lugar de que los valores se asignan de manera aleatoria.

Esta opción se puede mantener oculta, y habilitarse al momento de la corrección.

CONSIDERACIONES GENERALES SOBRE EL DESARROLLO DEL TP

- Proyecto Codeblock en grupo.
- Se deben usar vectores y funciones
- Se tiene que usar al menos un .h para organizar las funciones
- En el main() sólo debe dejarse el menú principal
- Cada opción del switch del menú debe llamar a una función (o varias funciones)
- No se pueden usar herramientas no vistas en la materia
- No se pueden usar variables globales.
- Para la defensa de forma virtual todos los integrantes del grupo deben tener activas las cámaras y los micrófonos.