Universidad Rafael Landívar
Tradición Jesuita en Guatemala

PROYECTO NO° 2 DE LABORATORIO DE PROGRAMACIÓN

Morales Herrera, José Guillermo
Carné: 1115823

GUATEMALA, 12 DE NOV. DE 23

Introducción

Este proyecto tiene como objetivo ofrecer una experiencia simulada de un juego de dados clásico. En este juego, los participantes lanzan dos dados, cada uno con seis caras numeradas del 1 al 6. La dinámica del juego se rige por reglas específicas que añaden emoción y estrategia a cada tirada.

Las reglas del juego son las siguientes: si la suma de los dados es 12 o 6 en el primer tiro, el jugador gana automáticamente 12 puntos. En caso de obtener una suma de 4, 6 o 10 en la primera tirada, el jugador pierde y la "Casa" gana 12 puntos. Si la suma es 2, 3, 5, 7, 8 o 9, esta cifra se convierte en el puntaje del jugador o de la "Casa". Además, un jugador puede perder si la suma es 11 antes de ganar algún punto, en cuyo caso la "Casa" obtiene 6 puntos.

El objetivo del juego es que el jugador decide la cantidad de partidas o tiradas de dados. Al alcanzar esta cantidad, el jugador gana si acumula más puntos que la "Casa"; de lo contrario, "La casa gana".

Existen dos modalidades de simulación. Una de ellas implica el uso de un archivo de texto de entrada que contiene todas las jugadas predefinidas. La otra modalidad genera aleatoriamente los valores de los tiros durante la ejecución del programa.

El código implementa diversos elementos para llevar a cabo la simulación. Utiliza variables enteras para almacenar cantidades como la cantidad de partidas, tiros por partida, puntajes y resultados individuales de los dados. Se emplea una lista de listas de enteros para almacenar los resultados de cada partida, con cada elemento de la lista principal representando un juego y conteniendo una lista interna con los resultados de cada tiro.

Se utilizan bucles "for" para controlar la iteración a través de las partidas y los tiros en cada una de ellas. Estructuras condicionales "if-else" evalúan diferentes condiciones según las reglas del juego y determinan el resultado de cada tiro y partida. La función "LanzarDado" genera un número aleatorio entre 1 y 6, simulando el lanzamiento de un dado.

Se realizan cálculos estadísticos para determinar el ganador, la probabilidad de ganar y se cuentan tiros con números pares, impares e iguales. La salida de información mediante "Console.WriteLine" se utiliza para imprimir resultados detallados de cada partida y al finalizar el juego. Este conjunto de elementos proporciona al usuario una visión completa y detallada de su rendimiento en cada partida, así como estadísticas finales del juego.

Análisis

Entradas del programa

Cantidad de partidas:
Número entero que representa cuántas veces se jugará el juego.
Cantidad de tiros por partida:
Número entero que indica cuántos tiros se realizarán en cada partida.

Salidas del programa

Ganador de la partida:
Mensaje indicando si ganó el jugador, la casa o si fue un empate.
Resultados de los tiros:
Detalles de cada tiro, mostrando los valores de ambos dados.
Tiros ganados por el jugador:
Número de tiros en los cuales el jugador ganó puntos.
Probabilidad de ganar:
Porcentaje que indica la probabilidad de que el jugador gane.
Tiros con números pares e impares:
Cantidad de tiros en los que la suma de los dados fue par o impar.
Tiros con números iguales:
Cantidad de tiros en los que ambos dados tuvieron el mismo valor.
Punteo final:
Puntaje total del jugador y la casa al final de todas las partidas.

Proceso del programa

Inicialización:
Solicitar al usuario la cantidad de partidas y tiros por partida.
Simulación de partidas:
Para cada partida, lanzar los dados M veces y aplicar las reglas del juego.
Almacenamiento de resultados:
Mantener un seguimiento de los resultados de cada tiro y el puntaje acumulado.
Presentación de resultados por partida:
Mostrar los resultados de cada partida al finalizarla.
Cálculo de estadísticas finales:
Calcular la probabilidad de ganar, contar tiros pares/impares, iguales, y sumar el puntaje total.
Presentación de información final:
Mostrar las estadísticas finales al usuario.

Restricciones del programa

Entradas válidas:
Verificar que las entradas del usuario sean números enteros positivos.
Cantidad de tiros válida:
Asegurarse de que la cantidad de tiros sea mayor que cero.
Manejo de resultados:
Garantizar que los resultados de los tiros se almacenen y presenten correctamente.
Reglas del juego:
Aplicar correctamente las reglas del juego en cada tiro y partida.
Presentación ordenada:
Mostrar los resultados de manera clara y ordenada al finalizar el programa.

Diseño

Diagrama de flujo

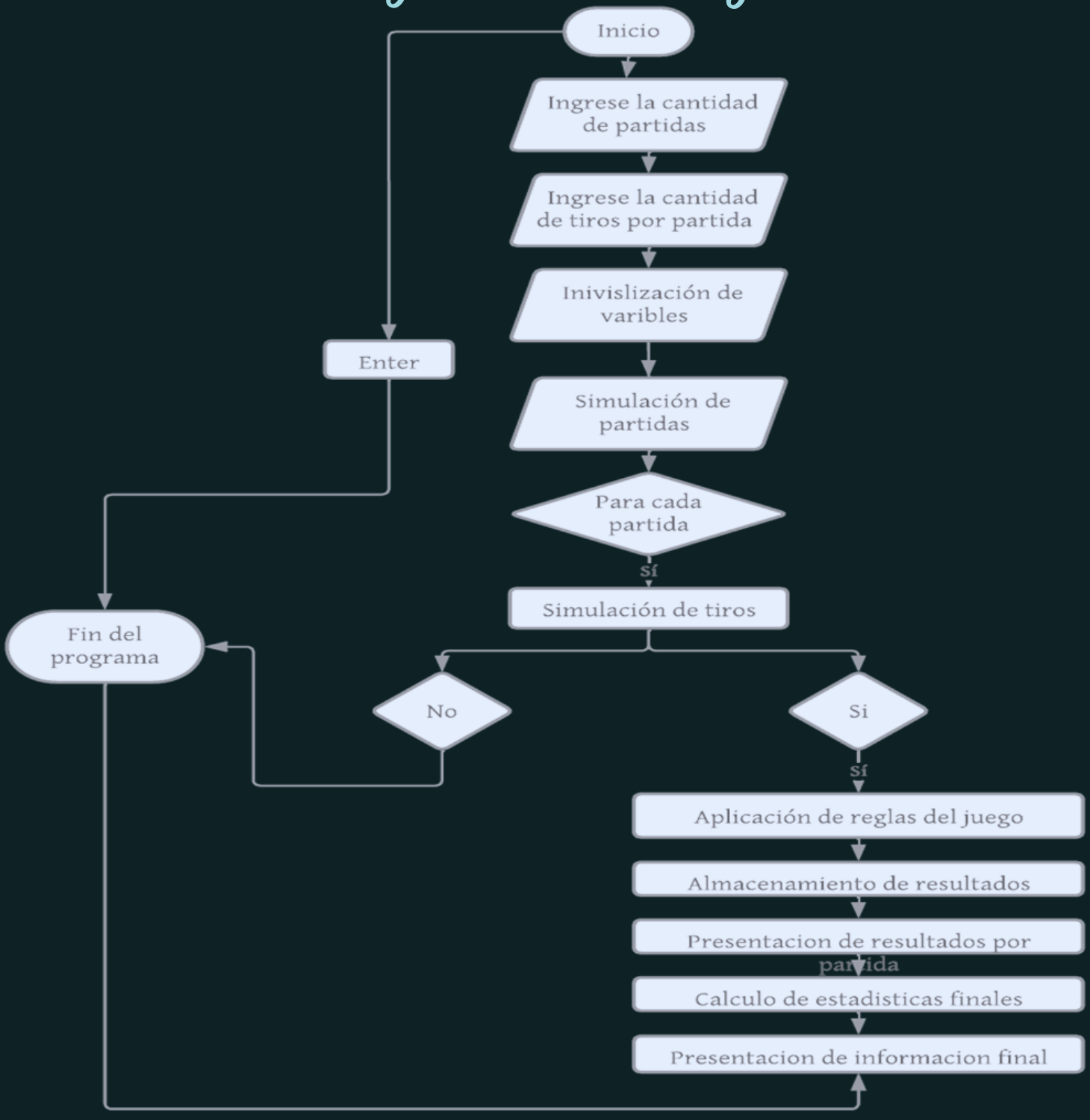

Diagrama de clases

Juego de dados
rand: Random
LanzarDado(): int Main(args: string[]): void

Resultados
Dado1: int Dado2: int
Resultado(dado1: int, dado2: int)

Partida
ResultadosTiro: List<Resultado> PuntajeJugador: int PuntajeCasa: int TirosGanados: int TirosPares: int TirosImpares: int TirosIguales: int
SimularTiro(): void CalcularPuntaje(): void MostrarResultados(): void

Conclusiones

Este proyecto ofrece una simulación detallada y estructurada de un juego de dados clásico, proporcionando a los jugadores una experiencia interactiva y emocionante. Las reglas del juego, que abarcan diversas situaciones según la suma de los dados, añaden complejidad y estrategia a cada tirada, generando un ambiente dinámico y desafiante.

El código implementa un conjunto de elementos clave, como variables enteras para el almacenamiento de datos, listas para mantener resultados detallados de cada partida, bucles y estructuras condicionales para controlar el flujo del juego, y funciones que simulan el lanzamiento de los dados y calculan los puntajes. La posibilidad de elegir entre dos modalidades de simulación, utilizando un archivo de texto o generando valores aleatorios, brinda flexibilidad al usuario.

En cuanto al análisis, se identifican claramente las entradas y salidas del programa, proporcionando al usuario una comprensión clara de cómo interactuar con el sistema y qué resultados esperar. El diseño del programa se presenta de manera organizada, con un diagrama de clases que muestra la estructura fundamental del juego y sus componentes.

El análisis detallado del proceso del programa revela un enfoque claro desde la inicialización hasta la presentación de resultados finales, asegurando un flujo coherente y la correcta aplicación de las reglas del juego. Se han considerado restricciones para garantizar la validez de las entradas y la presentación ordenada de los resultados

Recomendaciones

Se considera organizar el código en módulos más pequeños y funciones reutilizables para mejorar la legibilidad y facilitar futuras expansiones o modificaciones. Asegúrate de que la interacción del usuario sea intuitiva y clara, pudiendo agregar mensajes o indicaciones adicionales para guiar al usuario a través de las opciones disponibles.

Utilizar formatos más visuales o gráficos puede mostrar tendencias y estadísticas de manera más intuitiva. Evalúa oportunidades para optimizar el rendimiento del código, especialmente si trabajas con grandes conjuntos de datos. Esto puede incluir secciones que podrían beneficiarse de una mayor eficiencia.

Refina el manejo de excepciones para anticipar y gestionar posibles errores de manera más específica, mejorando la robustez del programa. Si el proyecto tiene la posibilidad de ser utilizado por usuarios de diferentes regiones, considera la internacionalización del software para adaptarlo a diferentes idiomas y culturas. Establece un mecanismo para recopilar comentarios de los usuarios después de que utilicen el programa, lo cual puede proporcionar información valiosa para futuras mejoras y ajustes.

Además, si aún no lo estás haciendo, considera la implementación de un sistema de control de versiones, como Git. Esto facilitará el seguimiento de cambios y colaboración en el código. Aumenta la cobertura de pruebas para garantizar un comportamiento correcto en una variedad de situaciones y escenarios. Realiza una evaluación de accesibilidad si es relevante para tu audiencia, garantizando que el software sea utilizable por personas con discapacidades.

Evalúa y refuerza la seguridad del programa, especialmente si hay intercambio de datos confidenciales. Verifica la compatibilidad del programa con diferentes sistemas operativos y entornos de ejecución, especialmente si planeas distribuirlo ampliamente. Estas recomendaciones abordan aspectos de diseño, rendimiento, experiencia del usuario y mantenimiento a largo plazo, y al implementarlas, puedes mejorar aún más la calidad y la usabilidad de tu proyecto.

Referencias

Fundamentos de programación (S.F)
https://www.uacj.mx/CGTI/CDTE/JPM/Documents/IIT/fund_programacion/U4-5.html#:~:text=La%20estructura%20repetitiva%20do%2Dwhile,por%20lo%20menos%20una%20vez.

Sentencia If – else en C
https://www.freecodecamp.org/espanol/news/sentencia-if-else-en-c-explicada/#:~:text=La%20sentencia%20if...else,si%20el%20argumento%20es%20cero.

Vectores en programación
https://www.programarya.com/Cursos/C++/Estructuras-de-Datos/Arreglos-o-Vectores

Función lanzar dados en c++
https://es.stackoverflow.com/questions/156709/lanzamiento-de-dados-en-c

For en c++
https://www.programarya.com/Cursos/C++/Ciclos/Ciclo-for

Anexos

Manual de usuario

El siguiente manual de usuario tiene como objetivo explicar la funcionalidad del programa "Juegos de Dados" y proporcionar al usuario la información necesaria para utilizarlo adecuadamente. Para poder dar a entender mejor pasar a la siguiente página.

JUEGO DE DADOS

En qué consiste el juego:

El juego de dados implica lanzar dos dados, cada uno con seis caras numeradas del 1 al 6. Durante una serie de partidas, el jugador realiza una cantidad determinada de tiros, y la suma de los resultados de los dados determina diferentes situaciones y puntajes en el juego.

Objetivo del juego:

El objetivo es acumular más puntos que la "Casa" al finalizar la cantidad predeterminada de partidas o tiros. El jugador gana puntos según las reglas de cada tirada, y la "Casa" también acumula puntos en función de las reglas del juego. Al finalizar la cantidad establecida de partidas, el que tenga más puntos es el ganador.

Cómo se juega:

1. Se decide la cantidad de partidas o tiros de dados que se jugarán.
2. En cada tirada, se lanzan dos dados y se suma el resultado.
3. Según la suma de los dados, se aplican reglas específicas para determinar el puntaje de la partida.
4. El jugador y la "Casa" acumulan puntos en función de las reglas.
5. Si el jugador alcanza la cantidad predeterminada de partidas o tiros, se compara el puntaje final. El jugador gana si tiene más puntos; de lo contrario, la "Casa" gana.

Cómo se juega:

- Si la suma de los dados es 12 o 6 en el primer tiro, el jugador gana automáticamente 12 puntos.
- Si la suma es 4, 6 o 10 en la primera tirada, el jugador pierde, y la "Casa" gana 12 puntos.
- Si la suma es 2, 3, 5, 7, 8 o 9, esta cifra se convierte en el puntaje del jugador o de la "Casa".
- Si la suma es 11 antes de ganar algún punto, el jugador pierde, y la "Casa" obtiene 6 puntos.

Al iniciar se pedirá ingresar la cantidad de partidas que se desean realizar. Cantidad que debe de ser un número entero.

```
Ingrese la cantidad de partidas: 3
```

Posteriormente se le solicitará al usuario ingresar la cantidad de tiros por partida que se desean realizar. Igual a la cantidad de partidas debe de ser un numero entero.

```
Ingrese la cantidad de tiros por partida: 4
```

Posteriormente, el programa simulará partidas de juegos de dados, generando resultados aleatorios. Para cada partida, se registrarán los resultados de cada tiro, contabilizando puntajes y estadísticas.

```
Partida 1: El jugador ganó
Resultados de los tiros:
Tiro 1: Dado 1: 5, Dado 2: 4
Tiro 2: Dado 1: 2, Dado 2: 2
Tiro 3: Dado 1: 3, Dado 2: 3
Tiro 4: Dado 1: 6, Dado 2: 5
Puntaje de la partida - Jugador: 18, Casa: 16
Tiros ganados por el jugador: 1

Partida 2: La casa ganó
Resultados de los tiros:
Tiro 1: Dado 1: 4, Dado 2: 2
Tiro 2: Dado 1: 3, Dado 2: 1
Tiro 3: Dado 1: 4, Dado 2: 2
Tiro 4: Dado 1: 2, Dado 2: 2
Puntaje de la partida - Jugador: 12, Casa: 20
Tiros ganados por el jugador: 0

Partida 3: La casa ganó
Resultados de los tiros:
Tiro 1: Dado 1: 1, Dado 2: 5
Tiro 2: Dado 1: 2, Dado 2: 4
Tiro 3: Dado 1: 4, Dado 2: 3
Tiro 4: Dado 1: 3, Dado 2: 3
Puntaje de la partida - Jugador: 12, Casa: 25
Tiros ganados por el jugador: 0
```

Al finalizar todas las partidas, se mostrará una información detallada, incluyendo el ganador, los tiros realizados, la probabilidad de ganar y otros datos relevantes.