Marcos Reyes, 07/2021

Para el reto SOFKA

Planeación del código:
Entender requerimientos usuario, objetos iniciales, diagrama de flujo.
Definir estructura del juego e interacción con el usuario.
Programar funciones por orden de prioridad, primero las fundamentales y luego los detalles.
Documentar el proceso.
Llevar control de versiones con github.

Planeación del proyecto

Entender requerimientos de usuario:

Se debe "modelar un concurso de carros". Lo que se busca en este juego es crear unos carros y posicionarlo en una pista (cada carro tiene un conductor), puede existir tantos carros como carriles, cada pista deberá tener el mismo límite de distancia (kilómetros) para el recorrido del carro, los carros avanzan de forma aleatoria aumentado su distancia por medio de metros (los kilómetros de debe convertir a metros para que el avance sea en metros)
Cada avance debe existir un dado (de 1 a 6) que permite mover el carro y se debe multiplicar por 100, donde si se tira el dado y sacas 5 entonces debería ser 5*100 = 500 metros de recorrido.
Al final debe existir un podio donde se clasifique primer, segundo y tercer ganador.

● Configurar Juego: Crear juego con jugadores, el juego debe tener los límites de kilómetros por cada pista (un jugador puede ser un conductor y un conductor debe tener un carro asociado y un carro debe estar asociado a un carril que a su vez debe estar en una pista)
● Iniciar el juego: iniciar con un identificado del juego, se debe tener la lista de carros en donde se pueda iterar y avanzar según la posición de la pista o carril, esto debe ser de forma aleatoria (por medio del dado).
● Asignar podio (fin del juego): Se debe seleccionar primer, segundo y tercer lugar en la medida que los carros llegan a la meta (asignar al podio).
● Guardar datos: Se debe persistir los resultados con los nombres de los conductores en la posición del podio y agregar un contador de las veces que ha ganado (txt o cvc) (manejo de errores en estos archivos)

Se debe usar:
●Manejo de clases u objetos (Creación de objetos de entidades; pista, juego, carril, carro, conductor, jugador, podio)
● Persistencia de datos
● Manejos de listas o colecciones
● Conocimiento de cualquier lenguaje de programación.

● **Manejo de Git (versión de control)**
●Métodos con menos de 6 líneas de código segregados en métodos más pequeños.

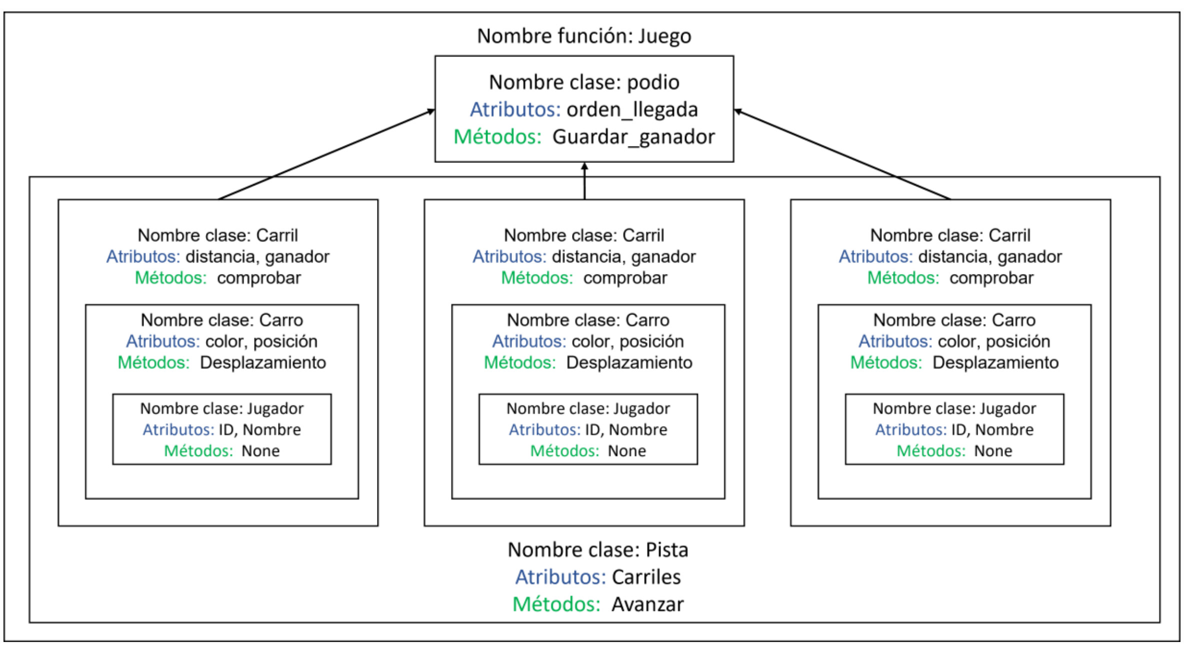
Resume

Crear juego de carreras. Con los siguientes objetos: Jugador, conductor, carro, carril, pista, podio, juego.
Todos los carriles tienen su distancia individual y es igual para todos. La regla para avanzar es la siguiente "un numero aleatorio de 1 a 6 multiplicado por 100". Clasificar el orden de llegada y guardarlo en un archivo (realizar análisis de los resultados (estadísticas juego): quien ganó las últimas 3 carreras, quien ha ganado más carreras, porcentaje de las victorias, quien ha estado mayor veces en el podio). Organizar los carros en una lista donde se iteran para avanzar.

Estructura del programa y variables

Crear 5 jugadores, (5 conductores que son los mismos jugadores), 5 carros, 5 carriles, 1 pista, 1 podio, 1 juego.

Objetos: crear clases hereditarias con una super clase inicial (**jugador**), esta clase contiene el ID del jugador y su nombre. Le sigue el **carro**, que hereda a jugador y tiene las propiedades de **color_carro** y **posición** (con el método de desplazamiento que indica cuanto debe avanzar). Le sigue el objeto de **carriles**, que contiene la distancia del carril en kilómetros, también, si el carro en ese carril ya ganó (método como comprobador de que ya se ganó). **Pista** ya es una clase independiente y tendrá entre sus atributos, una lista que contiene los carriles en los que van los carros y (es el iterable para designar cuando debe avanzar un carro). El **objeto** podio tendrá el lugar de los tres primeros por independiente (con un método que guarde el orden de llegada y el orden de todos los demás). Y **juego** será la función principal que ejecuta todo el código.
El esquema quedaría de la siguiente manera:



Interfaz de usuario por consola:

En la primera versión del programa, este mostrará las opciones del usuario en la consola. Lo primero será una bienvenida a juego y recomendación a leer las instrucciones (deben ser breves). Más que un juego donde se debe escoger un corredor y jugar con él, será un juego de apuestas, donde el jugador tendrá a su disposición un dinero inicial y su propósito es ganar más y no quedar en quiebra (game over). Puede elegir a quien apostarle en base a las estadísticas de todos los juegos anteriores. El porcentaje de lo que puede ganar, dependerá directamente del porcentaje de victorias (será inversamente proporcional), entre menor sea el porcentaje, más puede ganar. Puede elegir entre apostar quien va a ganar, o quien va a quedar montado en el podio. Guardar el nombre del concursante y colocar un ranking con los que más plata hayan ganado. Se inicia con 10 mil pesos colombianos. El jugador puede elegir cuanta plata apostar. Finaliza si se acaba el dinero o el jugador no quiere seguir apostando. (si apuesta 666 pesos, saldrá, haz hecho un pacto con el diablo y perderás todo tu dinero o ganar 666000 pesos)

Diagrama de flujo información usuario

Inicia el juego

Se lee el archivo, se guardan e inicializan las variables, se crean los objetos.

Se le dice al jugador que tiene 10k para apostar, puede apostar quien va a ganar (gana el doble de lo apostado) o quien puede quedar en el podio (tres mejores), (gana 1.3 veces lo apostado). Se le cobran 500 pesos por comisión en cada ronda. Jugador selecciona cual camino.

Se verifica si tiene la plata para seguir jugando, se asigna 10k a la variable de dinero

Se le dice que puede seleccionar ya el corredor o primero desea ver las estadísticas de los x partidos anteriores, pero esto tiene un costo de 500 pesos.

Se toma la decisión que toma el usuario

(se verifica si tiene dinero para jugar) Si selecciona estadísticas, se le cobra y se le muestran las estadísticas. (realizar análisis de los resultados (estadísticas juego): quien ganó las últimas 3 carreras, quien ha ganado más carreras, porcentaje de las victorias, quien ha estado mayor veces en el podio). Se limita por tiempo y se borra de la consola, Después:

Se verifica si tiene dinero para esto, Se realizan los cálculos estadísticos con los datos de los archivos y se muestran en pantalla. Puede saltar a la siguiente pestaña o hasta que se acabe el tiempo

Si dijo que no o ya vio las estadísticas, debe seleccionar un corredor

Guardar el corredor que se haya seleccionado

Después define el monto que desea apostar y comienza la carrera. (se puede graficar de cierta manera limpiando la pantalla y utilizando caracteres) (si apuesta 666 pesos, saldrá, haz hecho un pacto con el diablo y perderás todo tu dinero o ganar 666000 pesos). Apuesta mínima de 1000

Mostrarle cuanto dinero tiene, limitar la apuesta a lo que tiene. Ver si activa el secreto

Ejecutar la función de juego que va haciendo avanzar a los carros y guarda los ganadores en data. También ir guardando el número de juegos que se han realizado. Y la fecha y hora del último juego

Se muestran los ganadores, puede ser ahí mismo en la grafica de la carrera, que vayan saliendo "primero, segundo, tercero"

Se muestra en pantalla quien ganó

Se le informa al jugador si gano dinero o lo perdió. Se le informa su estado de cuenta y se verifica si aun tiene dinero para apostar y pagar la comisión por el juego.

Se calcula cuanto ganó el jugador, su estado de cuenta final, se verifica si puede seguir jugando y si desea seguir jugando o se retira

Si tiene dinero, se lleva al menú inicial y comienza de nuevo.

Se inicializan otra vez las variables necesarias y se guarda el dinero. Dependiendo del porcentaje que haya apostado se le puede dar un mensaje en la siguiente apuesta "con que eres de tomar grandes riesgos, ¿otra apuesta grande?" "un apostador cauteloso, apostando ni poco ni mucho" "¿miedo a perder dinero?, el que no arriesga no gana"

Si no, se le agradece por alimentar nuestros bolsillos y se cierra.

Se le da un balance de cuanto ganó y cuanto perdió y cuantas rondas duro

El diagrama queda de la siguiente forma.

Estructura de los archivos:

Data contiene en sus tres primeras filas, la información sobre los 5 corredores (id, nombre, figura de su carro para graficar), el resto de sus filas es un arreglo que guarda las posiciones de llegada de los corredores.

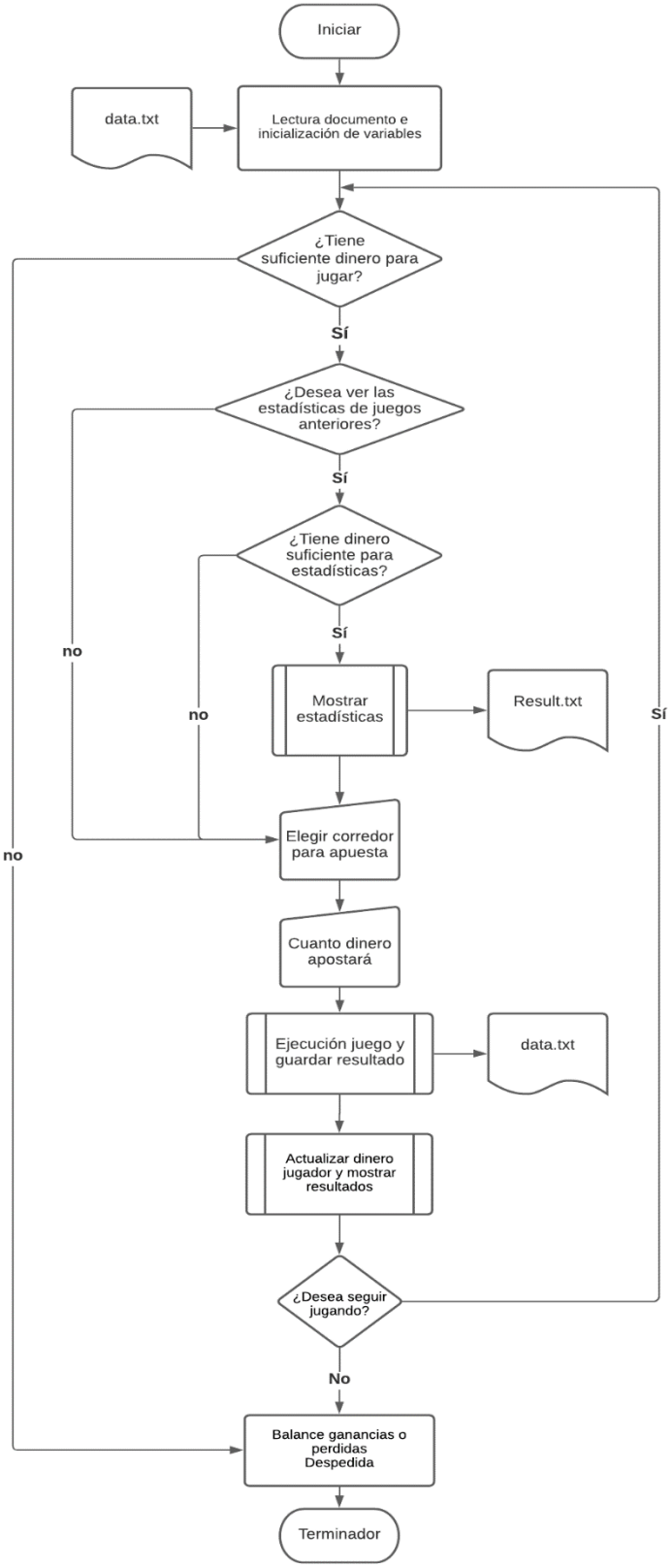
```
0, 1, 2, 3, 4
Marcos,Daniel,Andres,Camilo,Wilson
 ]>, }>, )>, =>, #>
3,4,5,1,2
1,2,4,3,5
4,3,1,2,5
2,1,4,5,3
2,1,3,4,5
1,5,2,3,4
```

Result guarda los datos estadísticos de todas las carreras realizadas hasta el momento

```
====================Reporte estadisticas====================

Numero de carreras totales realizadas: 120

Corredor     juegos ganados      porcentaje victorias
Marcos              33                  28%
Daniel              21                  18%
Andres              24                  20%
Camilo              24                  20%
Wilson              18                  15%

Numero de veces en el podio
Marcos              75
Daniel              63
Andres              85
Camilo              75
Wilson              62

Ganadores ultimas 3 carreras: Andres   Daniel   Andres
```

Diagrama reto Sofka

Marcos Daniel Reyes Garces | August 6, 2021



Funciones necesarias

- Función para lectura y escritura de archivos
- Función para ingreso de información del usuario (con programación defensiva)
- Función para el control del dinero u objeto, objeto concursante
- Función para las estadísticas
- Función del juego carro y su gráfica en consola
- Función para verificar si tiene dinero.
- Función para graficar los carros

Programación defensiva

Verificara si es el tipo de dato correcto

Verifica si está dentro del rango limite

Comprobar errores en la lectura del archivo (archivo no existe entre otros)

Muchos intentos erróneos del usuario

Verificar si los archivos tienen la información correcta o falta, pedir que se borre y se remplacen los archivos del repositorio.

Prioridad para programar

Programar la estructura central. El juego.

Programar la lectura y escritura de archivos

Programar las estadísticas

Programar concursante

Programar input de usuario

Desarrollo “Programar la estructura central. El juego”

La función inicializa las variables y crea los objetos con la información contenida en “data.txt”

Se utiliza el objeto pista para guardar los carriles y tiene el método para recorrer todos objetos y hacer que avancen.

Los carriles tienen la función de avanzar.

Se van graficando los carros en la consola a medida que avanzan.

Una clase independiente está vigilando constantemente quien va llegando a la meta y va guardando los resultados.

Una vez todos finalicen, se guarda el resultado en “data.txt”

Desarrollo “Programar la lectura y escritura de archivos”

La función recibe el archivo y el modo de lectura

Comprueba si no hay error y entrega el archivo

Entregará el archivo procesado (lectura) o el archivo listo para escribir.

Desarrollo “Programar las estadísticas”

Se recibe la tabla con toda la información de los corredores

Se realizan cálculos como: quien ganó las últimas 3 carreras, quien ha ganado más carreras (total carreras en primer puesto), porcentaje de las victorias, quien ha estado mayor veces en el podio

Guardar los resultados en variables

Mostrar resultados en consola

Darle un tiempo máximo para ver los resultados

Guardar los resultados en “result.txt”

Desarrollo “concursante”

Es un objeto que va almacenando el avance del jugador a lo largo del juego

Tiene atributos de dinero, intentos, riesgo asumido.

Desarrollo “input jugador”

Esta función tomara el input del usuario

Verificara si es numero lo que piden y el da

Verifica si está dentro del rango limite

Si falla alguna, se le pide que ingrese valor valido hasta que lo haga

Si repite 6 veces, cerrar el juego

Entrega el input final del usuario

Secretos

Si apuesta 666 pesos, saldrá, haz hecho un pacto con el diablo y perderás todo tu dinero o ganar 6660 pesos.

Sí logra duplicar su dinero inicial, se le hace entrega del premio de sololearn para que haga un curso en un lenguaje de programación.

No se pudieron incluir por limite de tiempo