**Juan Esteban Caicedo Arroyave**

**Nombre del juego: Contra el trafico**

**Concepto:**

El juego tiene como concepto un carro el cual va en contravía, la vía es de tres carriles y por esta vía van a estar viniendo carros repetitivamente, el usuario va a manejar el carro que viene en contravía y va a tener que esquivar los carros que vienen por la vía, en el momento en que el usuario colisione con los carros que vienen por la vía pierde el juego.

A medida que el usuario vaya avanzando y vaya esquivando carros los carros irán viniendo aun con más rapidez.

El juego al final le dará la puntuación al usuario, la puntuación será cuantos carros logró esquivar el usuario.

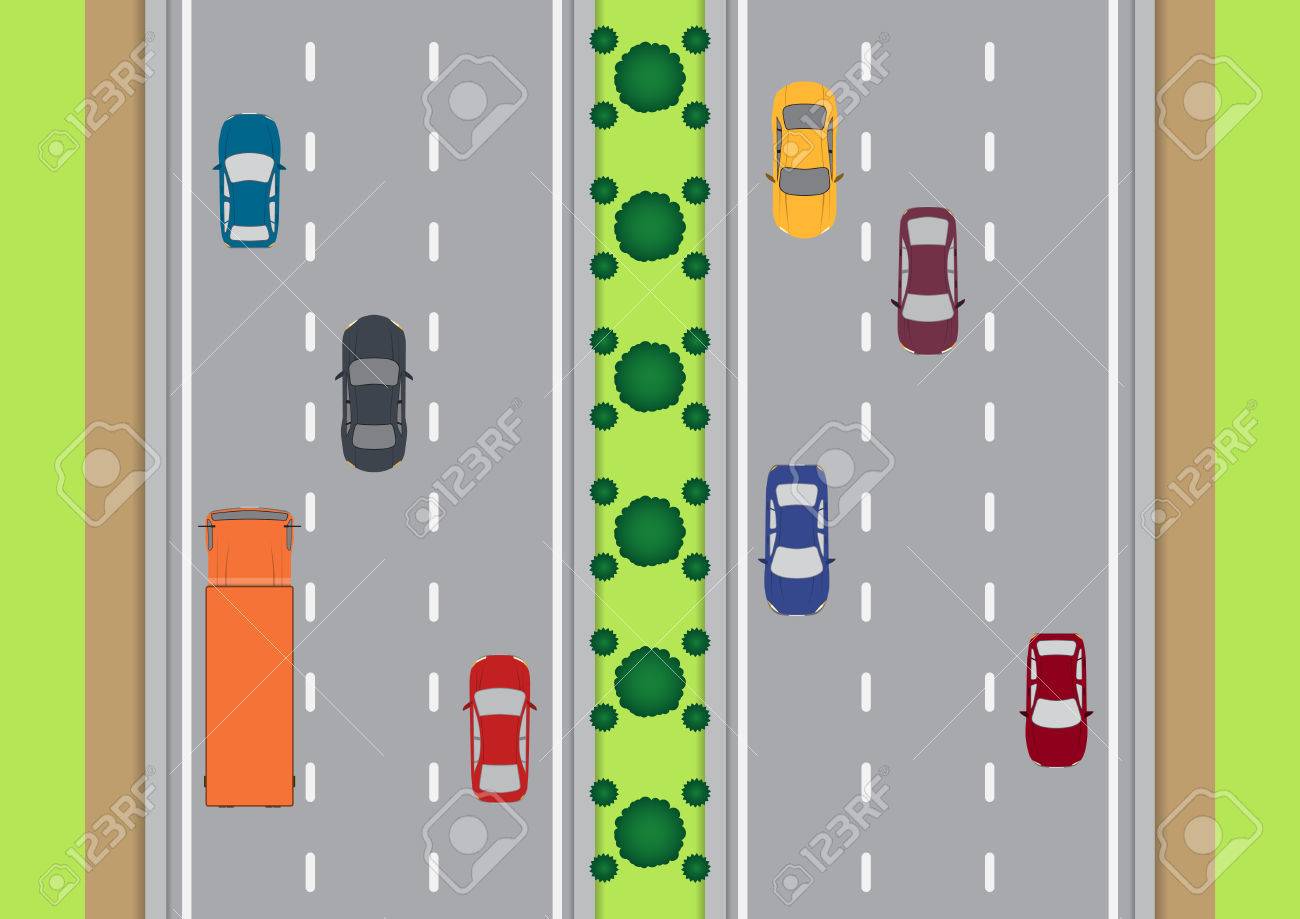
Los carros tendrán casi la misma dimensión de un carril de la vía, pero el coche será un poco más pequeño de ancho, los carros que tiene que esquivar el usuario andarán siempre por el mismo carril desde el inicio hasta que salgan de la escena. Es decir, no tendrán movimiento hacia los lados y siempre andarán por el centro del carril.

Los coches que esquivara el usuario no serán únicamente carros, por la vía también pueden venir otros vehículos, como buses o grúas, menos motos.

El vehículo del usuario si será un carro, este será rojo.

**Vistas:**

La vista del juego será desde arriba como en el siguiente ejemplo:



<https://es.123rf.com/photo_42929978_vector-vista-tr%C3%A1fico-de-la-carretera-de-arriba.html>

La idea de la vista es igual a la de la foto, solo que los carriles serán un poco más anchos, para que el jugador se pueda mover libremente por toda la vía.

La vista de los carros también es la misma solo que faltaría el carro del usuario que estará en sentido contrario, pero tendrá la misma vista.

**Mecánicas:**

El usuario para mover el carro usara los movimientos típicos de teclado, w para mover el carro hacia arriba, a para mover el carro hacia la izquierda, s para mover el carro hacia abajo y d para mover el carro hacia la derecha.

El carro se podrá mover libremente por toda la vía, no obstante, no podrá salirse de la vía el movimiento solo se hará dentro de esta.

**Requerimientos:**

**-**Un escenario que tendrá como imagen de fondo la vía completa con sus alrededores como en el anterior ejemplo de la vista.

-Un objeto que será el carro que manejará el usuario.

-Objetos que serán diferentes tipos de vehículo como carros, buses o volquetas que serán los obstáculos del usuario en la vía.

-Una función que reciba las claves que ingrese el usuario desde el teclado y modifique la posición del usuario.

-Una función que identifique si el carro del usuario ha colisionado con alguno de los vehículos obstáculos.

-Una función que cree los vehículos obstáculo dependiendo del tiempo y los ponga en alguno de los carriles, estos se van a crear en un ciclo y el ciclo se va a repetir mientras el usuario no colisione.

-Una función que elimine los objetos obstáculo cuando salgan de la vía.

-Una variable contadora tipo int que cuente cuantos objetos ha esquivado el usuario

-Una variable tipo int con la velocidad de cada vehículo, tanto de los vehículos obstáculo como el vehículo del usuario.

-Una función que mueva automáticamente los vehículos obstáculo en la vía.

**Objetos y clases:**

-Objeto base- Vehículos: Hereda Qobject

Private:

Int posicionx

Int posicióny

Public:

Getterandsetter posicionx

Getterandsetter posiciony

Void Posición ( ): Modifica la posición del vehículo en la escena utilizando setpos( ).

-Objetos derivados:

°Vehículos obstáculo: Estos podrán ser de diferentes clases dependiendo del tipo de vehículo, por ejemplo, si es un bus está será su clase, si es un carro esta será su clase, así consecuentemente-

A continuación, un ejemplo.

Bus: Hereda Vehículos y hereda Qpixmapitem.

Private:

Qpixmap imagen

Int velocidad

Public:

Getterandsetter velocidad.

Void movimientoautomatico( ): Este mueve el vehiculo por toda la via hasta que salga de la escena cuando este note que salió de la escena invoca la función eliminarobjeto( ).

Void eliminarobjeto( ): Elimina la instancia de la clase.

Constructor: En el constructor se inicializan las variables velocidad e imagen con sus respectivas cualidades.

°Vehiculo usuario: Este tendrá los mismos atributos de los vehiculos obstáculo exceptuando los algunos públicos, los cuales serán movientoautomatico y eliminarobjeto.

-Mainwindow

Esta clase tendrá en sus atributos privados cada uno de los tipos de carro que se vayan a añadir a la escena, incluyendo el carro del usuario. Además de los siguientes.

Private:

Qgraphicsscene \*scene

bool colisiona

int vehiculosesquivados

Public:

Void Keypressevent( ): Esta función leerá las teclas que ingrese el usuario y dependiendo de la tecla moverá el carro del usuario por toda la vía.

Void Juego( ): Esta función se encargara de crear los objetos obstáculo en intervalos de tiempo en un ciclo que se ira repitiendo, esta estará invocando repetitivamente a la función haycolision para saber si el usuario colisiono con alguno de los obstáculos.

Void haycolisión: Esta función revisara si el usuario ha colisionado con alguno de los vehículos obstáculo y le cambiara el valor al atributo colisiona dependiendo de si colisiona o no.