

You're cool, the engine's hot, the girl's gorgeous,  
a tank full of gas and an open road.... the rest is up to you..!

Screenshot from Amstrad version.



Screenshot from CBM 64/128 version.



# OutRun™



Screenshot from Spectrum version.

# START



Screenshot from Atari ST version.



The ultimate experience in motor sports simulation, the absolute challenge to nerve and reflexes. Feel the wind in your hair and the pull of the G-force as you power your way along the highways and byways, a girl by your side, and open country before you. Experience the exhilaration and excitement of driving a high performance sports car in this teasing time trial where your co-ordination and nerve will be tested to unbelievable limits. Out Run the coin op starred as an arcade sensation. Out Run the computer simulation mirrors this exciting all action spectacle.

CBM 64/128  
CASSETTE £9.99 DISK £11.99  
SPECTRUM  
CASSETTE £8.99

AMSTRAD  
CASSETTE £9.99 DISK £14.99  
ATARI ST  
DISK £19.99

## VIDEOJUEGOS



OutRun Clon  
computer reality on December 10th.

Curso 2019-2020

Andrés Gavín Murillo - 716358  
Rubén Rodríguez Esteban - 737215

SEGA

U.S. Gold Ltd., Units 2/3 Holland Way, Holland,



Birmingham B6 7AX. Tel: 021 356 3388.

# Tabla de contenidos

<b>Introducción</b>	<b>3</b>
<b>Concepto del juego</b>	<b>3</b>
<b>Características principales</b>	<b>3</b>
<b>Historia</b>	<b>4</b>
<b>Género</b>	<b>4</b>
<b>Público dirigido</b>	<b>4</b>
<b>Plataforma</b>	<b>5</b>
<b>Jugabilidad</b>	<b>5</b>
<b>Personajes</b>	<b>7</b>
Personajes jugables	7
Ferrari Testarossa	7
Personajes no jugables	8
Vehículos de tráfico	8
Abanderado	9
Público	9
Periodistas y reporteros	9
<b>Escenarios</b>	<b>10</b>
Animaciones	11
Logo de Sega	11
Comienzo de carrera	12
Llegada a la meta	14
Generador automático de mapas	16
Objetos	16
Fondo	16
Trazado del mapa	16
<b>Sistema de puntuación</b>	<b>25</b>
<b>Interfaz</b>	<b>26</b>
Logotipo inicial	26
Menú inicial	27
Menú de selección de música	28
Pantalla de juego	30
Pantalla de Game over	31
Menú de pausa	32
Menú de opciones	32
Menú de ajustes de sonido	33

Menú de ajustes de gráficos	34
Menú de configuración del vehículo	36
Menú de ranking	37
<b>Gráficos</b>	<b>38</b>
<b>Música y SFX</b>	<b>40</b>
Pistas musicales	40
SFX	40
<b>Fuentes de texto</b>	<b>41</b>
<b>Detalles de implementación</b>	<b>41</b>
Back doors	41
Cámara y renderizado	41
Generación de enemigos	42
Inteligencia Artificial	43
Físicas del vehículo	44
Geometría del mapa	45
Curvas	45
Elevaciones	45
Bifurcaciones	46
Geometría 3D	48
Salida y meta	49
Mapeador de teclas	49
Aleatoriedad	49
Audio	50
<b>Gestión del proyecto software</b>	<b>50</b>
<b>Software utilizado</b>	<b>51</b>
<b>Problemas encontrados</b>	<b>51</b>
<b>Bibliografía</b>	<b>53</b>

# Introducción

Este proyecto ha consistido en diseñar y crear desde cero una nueva versión de un popular videojuego de la década de los 80. De entre una gran variedad de candidatos, ha sido el exitoso juego, denominado *Out Run*, el escogido finalmente para poder desarrollar un clon aproximado de él, pero con algunas mejoras de cara a la jugabilidad de los usuarios.

A lo largo del siguiente documento, se ha procedido a explicar cuál es la idea de dicho videojuego, cuáles son las características y mejoras de esta versión con respecto de la original, así como los principales problemas encontrados durante el diseño.

## Concepto del juego

*Out Run* es un videojuego de carreras y conducción creado en 1986 por Yū Suzuki y Sega-AM2, publicado inicialmente para máquinas recreativas, aunque con el transcurso de los años se han lanzado versiones para otras plataformas como Sega o Play Station.

El esquema de juego es muy sencillo, consiste en recorrer un total de cinco etapas o escenarios antes de que se acabe el tiempo disponible de juego. Para poder lograr tal objetivo el jugador se debe poner al volante de un Ferrari Testarossa descapotable rojo, llevando a una mujer rubia como acompañante.

Dicha idea en cuestión ha sido tomada como el cimiento principal, sobre la que posteriormente se ha desarrollado el resto del videojuego. No obstante, se han incorporado ciertas mejoras adicionales con el propósito de ganar en jugabilidad. Dichas mejoras, explicadas en apartados posteriores, se han tomado como referencia de la versión de *Out Run* lanzada para la Sega Mega Drive en el año 1991.

## Características principales

*Out Run* ha sido diseñado teniendo en cuenta en todo momento los criterios que a continuación se exponen:

- **Interfaz minimalista y funcional:** la GUI (Guide User Interface) ha sido pensada para ser preatentiva visualmente, de tal manera que los jugadores solo con verla sepan al instante como poder navegar por ella fácilmente. El uso de ciertos elementos como botones, textos concisos, colores vistosos y efectos de sonido han permitido que el modelo mental de la interfaz sea sumamente sencillo.
- **Realismo y velocidad:** dado que es un juego de conducción en el que los jugadores cruzan diversos escenarios para ir desde la salida a la meta, los gráficos han sido creados con el fin de simular mundos realistas, estéticos y, sobre todo, muy visuales con el fin de maravillar e impresionar al jugador. Además, ciertos aspectos como el

manejo del coche, las colisiones, o el tráfico han sido diseñados para sumergir a los jugadores en un clima de conducción lo más realista posible.

- **Desafío constante:** Out Run es un juego, por partida doble, altamente desafiante. Por un lado, cuenta con una amplia gama de niveles de dificultad con el objetivo de ser un reto para cualquier tipo de jugador, desde simples amateurs hasta jugadores expertos. Los trazados de los escenarios, el tiempo de juego disponible para poder completarlos y el tráfico controlado por la IA, logran que las partidas sean emocionantes, diferentes y difíciles, independientemente del tiempo que se lleve ya jugado.
- **Rivalidad:** para ser el mejor no sólo basta con llegar a la meta, además hay que hacerlo en el menor tiempo posible y con la mejor puntuación. Pues son estos dos últimos factores los que determinan la posición de los jugadores en la tabla de clasificación o ranking, donde sólo aparecen los siete mejores jugadores.

## Historia

La historia del videojuego está ambientada en el Japón de la década de los 80, concretamente en el año 1986. El multimillonario piloto de carreras Maxwell Dillon ha adquirido como su última pieza de colección el vehículo más rápido y lujoso del momento, un Ferrari Testarossa rojo descapotable. Maxwell Dillon se va a embarcar en un reto junto a su novia, la rubia modelo Ilythia Baker, en la que van a recorrer los lugares más icónicos del país con el objetivo de batir sus propias marcas de tiempo.

## Género

Out Run es un videojuego arcade de carreras 2.5D, pues el modelado tanto de los paisajes con todos sus elementos como de los vehículos emplean geometría en 3D para su ubicación, pero no son objetos diseñados en 3D sino imágenes en 2D.

## Público dirigido

El videojuego propuesto ha sido pensado para ser dirigido a todos los públicos. Es un juego de habilidad que mezcla rivalidad, coordinación, velocidad y diversión. Tales elementos hacen que cualquier persona, independientemente de la edad y del sexo, pueda verse con el deseo de querer jugar y pasarlo bien. Sin embargo, se recomienda que no jueguen personas menores de 7 años, tal y como indica el PEGI (Pan European Game Information) del juego. La razón estriba en que es un juego que puede resultar complejo de entender y controlar para usuarios tan pequeños.

# Plataforma

El juego se ha diseñado para poder ser jugado exclusivamente en computadores de sobremesa y/o en PC's. El juego es compatible principalmente con Windows, siempre y cuando la arquitectura del sistema operativo sea de 64 bits. También se ha probado su funcionamiento en Linux y Mac mediante WineHQ 5.0. Además, cabe destacar que el juego ha sido preparado para que los usuarios no tengan que instalar ningún tipo de aplicación auxiliar en Windows para poderlo arrancar dado que los módulos auxiliares que necesita para funcionar vienen integrados en el mismo, otorgándole así, portabilidad.

## Jugabilidad

El objetivo del videojuego *Out Run* es atravesar cinco escenarios, yendo desde la salida hasta la meta antes de que se acabe el tiempo. Para ello, el jugador deberá efectuar el trayecto evitando salirse de la carretera, chocar con los obstáculos y elementos del paisaje, así como evitar colisionar con el tráfico.

El jugador controla el Ferrari Testarossa rojo de los protagonistas, pudiendo moverse a la izquierda, a la derecha, acelerar y frenar haciendo uso del teclado.

Al comienzo del juego, el jugador puede ir al menú de opciones y configuración para modificar aquellos parámetros que deseé, como el nivel de dificultad, el control de la IA, el control de volumen de efectos y de las pistas de música, el ajuste de gráficos y resolución de la pantalla o el control del vehículo. Paralelamente, el jugador puede aceptar la configuración inicial por defecto y comenzar directamente a jugar.

Previamente a correr, el jugador deberá seleccionar una de las tres pistas musicales habilitadas en el juego. Dicha pista sonará de manera ininterrumpida, a no ser que el jugador decida pausar el juego, mientras el jugador corra. Tras seleccionar una banda sonora comienza la partida.

La partida comienza siempre en el mismo escenario, *Coconut Beach*, con el Ferrari Testarossa posicionado en la salida listo para arrancar en el momento en que tanto el semáforo como el abanderado lo indiquen.

Existen un total de quince escenarios disponibles, junto con los submapas adicionales creados para representar la salida, las bifurcaciones y la meta. Todos ellos son muy distintos visualmente. Durante el desarrollo de una partida, el jugador solamente podrá circular por única y exclusivamente cinco de ellos. Para cada uno de estos niveles existe un tiempo específico para poder completarlo que depende tanto de la complejidad del propio escenario como del nivel de dificultad seleccionado por el jugador.

Los escenarios muestran determinados tipos de paisajes como desiertos, playas, colinas, campos agrarios, etcétera, con el fin de sorprender al jugador de manera visual cada vez

que cambia el nivel. En cada escenario, el jugador deberá sortear al tráfico, evitar colisionar con los obstáculos del mapa y tratar de no salirse de la carretera.

Al finalizar cada escenario aparece una bifurcación en la que el jugador podrá decidir libremente si ir hacia la izquierda o hacia la derecha. Tras seleccionar una bifurcación, se alcanza un punto de control donde se le informa al usuario del tiempo empleado por vuelta, y se actualiza el tiempo disponible de juego sumando, al tiempo que ha sobrado de completar el escenario anterior, el nuevo tiempo necesario para completar el actual. Dicho tiempo depende del nivel de dificultad seleccionado.



Figura 1  
Jugador alcanzado un punto de control

Durante el recorrido, existe un contador de puntos que se incrementa siempre y cuando el jugador se encuentre en marcha, es decir, si el jugador está parado el contador no suma puntos. El incremento consiste en la velocidad actual multiplicada por un valor dependiente de la dificultad. Este contador de puntos es el factor que determina la posición del jugador en el ranking de los siete mejores jugadores.

En cada escenario pueden ocurrir dos situaciones. La primera es que se le agote el tiempo al jugador antes de alcanzar el punto de control, y en ese caso, tenga que volver a comenzar desde el principio. La segunda es que el jugador alcance la bifurcación y se incremente el tiempo de juego, y así hasta que llegue a la meta.

Tras llegar a la meta, el jugador podrá ver el ranking de puntuación en el que se encuentran los siete mejores jugadores. Si el nuevo jugador bate un nuevo record, es decir, se encuentra entre los siete mejores, será introducido en la clasificación. En caso contrario, no será incluido y su puntuación obtenida junto con su tiempo serán desechados. Los jugadores se ordenan por orden de puntuación decreciente, mostrando para cada uno su

nombre, sus puntos y el tiempo de duración de partida. Tanto el nombre como el tiempo son valores meramente informativos.

## Personajes

Seguidamente se van a mostrar todos y cada uno de los personajes que aparecen en el videojuego diseñado. Se ha optado por presentar a los personajes clasificándolos en dos grupos, personajes jugables y personajes no jugables.

### Personajes jugables

En esta categoría se encuentran todos aquellos personajes del videojuego que pueden ser controlados por el jugador.

#### Ferrari Testarossa

Es el vehículo en el que se desplazan los protagonistas. Es controlado por el jugador por medio del teclado. El vehículo puede acelerar y frenar, variando su velocidad de 0 a 300, expresada en km/h. El coche puede chocar tanto con objetos del mapa como con coches del tráfico. Los choques contra los objetos del mapa se han denominado como “choques duros” ya que cuando ocurren el coche hace trompos hasta llegar al centro de la carretera perdiendo toda la velocidad que llevaba. Las colisiones contra los vehículos son “choques blandos” ya que cuando se producen el coche realiza trompos despedido hacia el otro lado de la carretera. En este tipo de choques la velocidad del coche no se reduce a cero como en el otro tipo, sólo se reduce una parte. Adicionalmente, a la hora de frenar, al mismo tiempo que se reduce la velocidad, se encienden las luces traseras (efecto que en las versiones iniciales no está presente) con la finalidad de poder representar el comportamiento del bólido con un mayor grado de realismo.

A continuación se muestran algunos de los sprites, sin escalar, empleados para representar el vehículo:



Figura 2  
Ferrari Testarossa rojo en dirección  
recta sin frenar



Figura 3  
Ferrari Testarossa rojo en dirección  
girando a la izquierda



Figura 4  
Ferrari Testarossa rojo en descenso



Figura 5  
Ferrari Testarossa rojo en dirección recta frenando

## Personajes no jugables

En esta categoría se encuentran todos aquellos personajes del videojuego que no pueden ser controlados por el jugador. En esta categoría se engloban tanto los personajes de decoración de los mapas como los distintos automóviles que componen el tráfico.

### Vehículos de tráfico

A continuación se muestran los distintos tipos de vehículos empleados para poder modelar el tráfico del videojuego. Todos estos vehículos son controlados por la IA. La pauta de comportamiento viene determinada única y exclusivamente por la dificultad si está activada la IA, y sino se mantienen constantes. Los vehículos que lo componen son los siguientes:



Figura 6  
1985 Porsche 911



Figura 7  
Camioneta

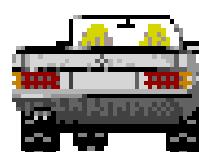


Figura 8  
1985 BMW 325i Cabriolet E30

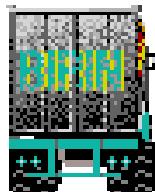


Figura 9  
Camion



Figura 10  
1971 Chevrolet Corvette



Figura 11  
1972 Volkswagen Beetle

## Abanderado

Este personaje sale en el juego al principio y al final de la carrera. Las únicas funciones que tiene es informar al jugador de cuando puede arrancar el vehículo, en caso de estar en la salida, y al mismo tiempo de cuando ha llegado a la meta.

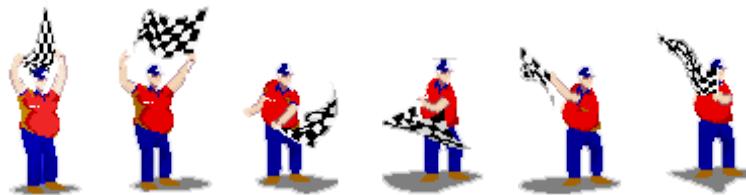


Figura 12  
Abanderado que anuncia la salida y llegada

## Público

Estos personajes, al igual que en el caso del abanderado, solamente aparecen en el primer escenario, concretamente en el lado izquierdo de la carretera. La tarea de este tipo de personajes es lograr que el paisaje quede más visual y estético.



Figura 13  
Público del primer escenario  
Coconut Beach

## Periodistas y reporteros

Estos personajes están presentes tanto en el primer escenario como en cualquiera de los últimos. Al igual que los personajes del público, su función es totalmente decorativa. Se encuentran tanto a la izquierda como a la derecha del Ferrari.



Figura 14  
Periodistas localizados a la izquierda  
del Ferrari



Figura 15  
Periodistas localizados a la derecha  
del Ferrari

## Escenarios

Como ya se ha explicado anteriormente, el juego cuenta con un total de 15 escenarios o mapas jugables, y adicionalmente, submapas para la animación de salida, las bifurcaciones, y la llegada del coche a la meta. La construcción de los mapas ha sido un proceso complicado y de larga duración en el que se ha desarrollado una técnica de generación automática de escenarios a través de ficheros de texto. Dicha técnica permite especificar los colores del mapa, el trazado y colores de la carretera, las curvas junto con su radio de curvatura y orientación (curva a la izquierda o curva a la derecha), los cambios de rasante (subidas y bajadas de montañas) y todos los obstáculos y elementos del paisaje.

Esta técnica se caracteriza por ser extremadamente sencilla, flexible y fácil de comprender para cualquier usuario. De este modo no solo se ha logrado facilitar el trabajo a los propios desarrolladores del videojuego, sino que además, se posibilita que cualquier usuario pueda modificar o incluso tenga la opción de añadir nuevos mapas al videojuego cambiando ficheros de texto sin necesidad de tener que tocar el código fuente que se encarga de procesarlos y renderizarlos.

Los submapas empleados para la creación de la salida, la meta y las bifurcaciones han sido modelados de forma diferente. Dichos mapas se añaden a los mapas anteriores de manera automática (como se detalla en los apartados de implementación). La razón principal reside en que se pensó desde un principio que dichas animaciones no pudiesen ser personalizadas por los usuarios, sino que fuesen lo más fieles posibles a las de las entregas originales, tanto la recreativa de 1986 como la de la Sega de 1991. Automatizando las animaciones se ha conseguido que siempre se visualicen de forma bastante fiel a como son en los juegos originales.

También han influido otras razones, aunque en menor medida, que reivindicaban la hipótesis de que era mejor hacerlo en el propio código que en fichero de texto. Por ejemplo, era bastante complejo poder encontrar una forma de representar en fichero toda la información que se debe mostrar en la salida y en la llegada. El movimiento del abanderado

para efectuar la salida, el efecto de derrape inicial del coche al arrancar o el cambio de las luces del semáforo son algunos ejemplos que reflejan no sólo la gran cantidad de sprites que intervienen y que deben mostrarse para conseguir animaciones correctas visualmente, sino también lo difícil que sería montar una técnica o patrón que pudiese procesar toda esa cantidad de datos.

Es importante tener en cuenta que han salido muchas variantes de *Out Run* desde 1986 hasta la actualidad. Una de estas variantes es por ejemplo la variación del orden de los escenarios a recorrer, es decir, los escenarios son los mismos pero el orden en el que se recorren no. Para la representación de las etapas hay dos posibles versiones, la original y la japonesa.

De estas dos se ha tomado la versión original, que es la que aparece en la recreativa de 1986. Los escenarios aparecen organizados de la siguiente manera:

Etapa					Meta
1	2	3	4	5	
				Vineyard	A
			Wilderness		
		Desert		Death Valley	B
	Gateway		Old Capital		
Coconut Beach		Alps		Desolation Hill	C
	Devil's Canyon		Wheat Field		
		Cloudy Mountain		Autobahn	D
			Seaside Town		
				Lakeside	E

Figura 16  
Esquema de las etapas de *Out Run* en la versión original

## Animaciones

Para el desarrollo del juego se han llevado a cabo un conjunto de animaciones que se han empleado a modo de puerta de enlace entre los diferentes componentes que lo forman. Cabe enfatizar que dichas animaciones han sido desarrolladas tomando como referencia tanto las animaciones del *Out Run* de la versión recreativa como las de la Sega, escogiendo de cada una, las partes que más han gustado a los diseñadores. Seguidamente se presentan cuáles son las animaciones que se han desarrollado.

## Logo de Sega

Esta animación tiene lugar al comienzo. Tiene como objetivo presentar el videojuego al jugador y no puede ser saltada de ningún modo por medio de ningún atajo de teclado. El

motivo de esta decisión es muy simple, rendir tributo al videojuego de la Sega Mega Drive de 1991. Pues ha sido esta entrega, junto con la de la versión recreativa las que han servido de inspiración para el desarrollo de esta nueva versión.

## Comienzo de carrera

Dicha animación ocurre al comienzo de partida, en ella se pueden observar elementos como el indicador de salida, el público, los periodistas, el abanderado, el semáforo cambiando de luz e indicando la salida al jugador, y lo más importante, el Ferrari Testarossa rojo derrapando y colocándose en posición de salida y listo para correr.



Figura 17  
Ferrari Testarossa derrapando para ponerse en  
posición de salida



Figura 18  
Ferrari Testarossa en posición de salida



Figura 19  
Ferrari Testarossa en posición de salida  
con semáforo anunciando salida



Figura 20  
Ferrari Testarossa arrancando

## Llegada a la meta

Esta animación ocurre en el momento en el que el jugador ha logrado completar un itinerario de 5 etapas. Aquí se puede observar la bonificación de puntos que el jugador recibe en función del tiempo que le ha sobrado para completar el nivel. Además, en esta animación el coche circula solo, sin que el jugador lo controle y derrapa llegando a la segunda meta donde se encuentran el abanderado de llegada y los periodistas. Dicha animación, igual que la de salida ha sido inspirada, que no clonada, de las versiones de *Out Run* para la recreativa y para la Sega.



Figura 21  
Jugador recibiendo bonificación por tiempo sobrante  
tras llegar a la meta



Figura 22  
Ferrari Testarossa derrapando al llegar a la meta

# Generador automático de mapas

El generador automático de mapas genera los 15 mapas del juego a partir de las carpetas map1, map2, ..., map15 ubicadas en la carpeta *resources*. Cada carpeta contiene toda la información relativa a cada mapa, estructurada de la siguiente manera:

## Objetos

Los objetos están definidos por las imágenes de los sprites ordenadas desde 1 al número de objetos de ese mapa de manera 1.png, 2.png, ..., y por la información opcional de cada objeto de manera 1.png.info, 2.png.info, ...

Los ficheros .info relativos a cada objeto son simples ficheros de texto opcionales que pueden contener los siguientes atributos también opcionales SCALE, HIT, HIT\_RIGHT, HIT\_LEFT, HIT\_SIDES.

Cada uno de estos atributos se define mediante su nombre, dos puntos, espacio y el valor decimal establecido. En el caso de SCALE toma valores positivos desde 0.0 hasta 3.40282e+038 y multiplica el tamaño del objeto por ese valor. HIT toma valores positivos desde 0.0 hasta 1.0, que indican el porcentaje de la imagen que es sólida, donde 0.0 establece que el objeto es completamente atravesable y 1.0 indica que toda la imagen es sólida y el coche se puede chocar. HIT define la parte sólida del objeto desde el centro del mismo (ej. para definir que solo se permiten los choques con los troncos de los árboles). HIT\_RIGHT y HIT\_LEFT definen la parte sólida del objeto desde la derecha y la izquierda del mismo respectivamente (ej. para definir que solo se permiten choques con el poste de las farolas en L) y HIT\_SIDES es la unión de HIT\_RIGHT y HIT\_LEFT de manera simultánea (ej. para definir que solo se permiten choques con los extremos de los arcos). A continuación, se muestra la información del objeto 3.png del mapa 1:

```
SCALE: 1.25
HIT_RIGHT: 0.388
```

## Fondo

El fondo del mapa está definido en el fichero bg.png y consiste en una imagen sólida de alta calidad y resolución.

## Trazado del mapa

El trazado del mapa está definido en el fichero map.info y contiene toda la información relativa al mapa, desde sus colores hasta su modelado, formado por rectángulos horizontales. Cada rectángulo contiene el trazado de la carretera, curvado o recto, llano o con elevación, y un o ningún objeto a cada lado de la carretera.

- **Colores.** Al comienzo del fichero se indican los colores RGB del mapa en el siguiente orden: Color de la carretera para los rectángulos pares, color de la

carretera para los rectángulos impares, color del suelo para los rectángulos pares, color del suelo para los rectángulos impares, color de las rayas de los bordes y color de las rayas discontinuas.

- **Elevaciones.** Dentro del fichero pueden establecerse zonas de subida, zonas de bajada y zonas llanas (por defecto). Para establecer un cambio de elevación se debe indicar DROP o CLIMB y un valor de elevación (no son metros, es un valor inventado) para las bajadas y las subidas (ej. DROP 1000), y FLAT para las zonas llanas. Esta elevación perdura hasta que se indica una nueva o termina el mapa.
- **Curvas.** Dentro del fichero pueden establecerse curvas hacia la derecha, curvas hacia la izquierda y rectas (por defecto). Para establecer una curva se debe indicar CURVE y un índice de curvatura positivo para las curvas hacia la derecha y negativo para las curvas hacia la izquierda (ej. CURVE 0.25), y STRAIGHT para las rectas. Esta curvatura perdura hasta que se indica una nueva o termina el mapa.
- **Carretera.** Cada rectángulo de la carretera está definido por ROAD y sus atributos, que pueden contener un objeto a la izquierda de la carretera con sus modificadores, deben contener el símbolo -, y pueden contener otro objeto a la derecha de la carretera con sus modificadores. A continuación, se detallan los atributos de ROAD con mayor claridad:
  - 1º Opcional: + ;Indica que es un objeto repetido infinitamente hacia la izquierda. Por defecto no se repite.
  - 2º Opcional: # ;Indica el índice del objeto colocado a la izquierda de la carretera (#.png). Por defecto no hay objeto.
  - 3º Opcional: #.# ;Indica el offset desde el borde de la carretera hasta el objeto. Por defecto es 0.0.
  - 4º Obligado: - ;Representa la carretera.
  - 5º Opcional: + ;Indica que es un objeto repetido infinitamente hacia la derecha. Por defecto no se repite.
  - 6º Opcional: # ;Indica el índice del objeto colocado a la derecha de la carretera (#.png). Por defecto no hay objeto.
  - 7º Opcional: #.# ;Indica el offset desde el borde de la carretera hasta el objeto. Por defecto es 0.0.
- **Aleatoriedad.** Se puede añadir un fragmento de carretera aleatorio mediante RANDOM. Para ello se indica el número de rectángulos que se deben generar de manera aleatoria con los posibles objetos que pueden aparecer (ej. RANDOM 100 14 15 17 18).

```

37 ROAD      -
38 ROAD      -
39 ROAD      8 0.2 - 8 0.2
40 ROAD      -
41 ROAD      -
42 ROAD      -
43
44 RANDOM 100 16 17 18
45
46 CURVE -1.2
47
48 ROAD + 14 0.2 - + 14 0.2
49 ROAD      -
50 ROAD      -
51 ROAD + 14 0.2 - + 14 0.2
52 ROAD      -
53 ROAD      -
54 ROAD + 14 0.2 - + 14 0.2
55 ROAD      -
56 ROAD      -
57 ROAD + 14 0.2 - + 14 0.2
58 ROAD      -
59 ROAD      -
60 ROAD + 14 0.2 - + 14 0.2
61 ROAD      -
62 ROAD      -
63 ROAD + 14 0.2 - + 14 0.2
64 ROAD      -
65 ROAD      -
66 ROAD + 14 0.2 - + 14 0.2
67 ROAD      -

```

Figura 23  
Ejemplo de representación de fichero map.info

En este punto es importante reseñar que para la realización de los escenarios, solamente se han podido encontrar los sprites originales del primer escenario y algunos aislados como los del abanderado, los periodistas, y en definitiva todos los que aparecen en el primer escenario, Coconut Beach. Todos los mapas, a excepción del primero han tenido que ser desarrollados por los propios diseñadores desde cero empleando la técnica de generación automática de mapas explicada anteriormente. El primero también ha sido desarrollado desde cero por los diseñadores pero tomando como inspiración el de la máquina recreativa de 1986. Seguidamente se muestran imágenes de los escenarios diseñados por los propios creadores del juego:



Figura 24  
Escenario de Coconut Beach



Figura 25  
Escenario de Gateway



Figura 26  
Escenario de Devil's Canyon



Figura 27  
Escenario de Desert



Figura 28  
Escenario de Alps



Figura 29  
Escenario de Cloudy Mountain



Figura 30  
Escenario de Wilderness



Figura 31  
Escenario de Old Capital



Figura 32  
Escenario de Wheat Field



Figura 33  
Escenario de Seaside Town



Figura 34  
Escenario de Vineyard



Figura 35  
Escenario de Death Valley



Figura 36  
Escenario de Desolation Hill



Figura 37  
Escenario de AutoBahn



Figura 38  
Escenario de Lakeside

## Sistema de puntuación

El sistema de puntuación o ranking empleado para la clasificación de los jugadores es muy sencillo y ha sido realizado igual que en la versión de *Out Run* para la recreativa de 1986. Básicamente, el ranking almacena los siete mejores jugadores ordenados por orden decreciente de puntuación. Al igual que en la versión recreativa, para cada jugador se almacena su nombre, su puntuación y el tiempo que le ha llevado completar la partida, expresado en minutos, segundos y centésimas de segundo.

El criterio es el orden de puntuación. De este modo jugadores que tienen puntuaciones mayores irán por delante de los que tienen puntuación más bajas. El jugador con mayor ratio de puntos va en primera posición, y así sucesivamente.

Cuando un jugador completa una partida y aparece la pantalla de ranking, si dicho jugador ha batido un récord de puntuación y se ha colado entre los siete mejores, se mostrarán los jugadores del ranking actuales junto con dicho jugador en la posición que le corresponde dentro de la clasificación, desechariendo así, al jugador que tenía la menor puntuación de entre los anteriores.

Los jugadores en cuestión se hallan almacenados en un fichero de texto denominado *score.info*, localizado en el directorio principal *resources*. La estructura de este fichero cuenta con 7 líneas de texto, cada una para uno de los jugadores con su respectiva información.

En cada línea se almacena el nombre del jugador, su puntuación, y su tiempo en minutos, segundos y centésimas de segundo. Todos los campos de información en cuestión se hallan separados por un único espacio.

scores.info						
1	28369879	ASD	4	57	12	
2	26417947	ZGZ	5	18	26	
3	25922526	ASS	5	22	56	
4	25722530	JHN	5	37	39	
5	25422530	YYY	5	52	42	
6	24822530	ADS	6	01	13	
7	23722530	EDW	6	05	47	

Figura 39  
Visualización del fichero de ranking de jugadores

## Interfaz

En esta sección se va a describir la GUI del videojuego. Para cada una de las pantallas se ha explicado cómo se puede acceder a ellas, qué acciones puede hacer el jugador en cada una, y a qué nuevas pantallas se puede acceder desde su pantalla actual.

### Logotipo inicial

Esta sección se corresponde con el arranque del juego. En este estado se puede observar la animación del logo de SEGA. La razón por la que se ha optado por exponer el logotipo de esta empresa de videojuegos es que *Out Run* ha sido diseñado tomando como base la versión arcade de la máquina recreativa, pero también se han añadido ciertos aportes adicionales procedentes de la versión de este juego para la plataforma de Sega Mega Drive.

La presencia de este logo, es por tanto, un indicador indirecto de la fuente de inspiración de donde se han extraído las mejoras que presenta la versión de *Out Run* aquí presentada. A continuación se muestra una captura del logo introducido en cuestión.



Figura 40  
Logotipo de Sega que aparece en la animación inicial

## Menú inicial

Es la segunda pantalla del juego, visible después de haber concluido la animación del logo de Sega. En dicha pantalla aparece la portada y el rótulo originales de *Out Run* para la versión recreativa. Adicionalmente, existe un rótulo de texto de color verde parpadeante en el que pone "Press Start". en la parte inferior izquierda información sobre los controles para navegar en la interfaz. Haciendo uso de las flechas del cursor, el jugador puede moverse a través de las distintas opciones de los menús. Para poder seleccionar una opción, el usuario debe presionar el botón *Start*, la tecla ENTER, y para salir debe presionar la tecla ESC.



Figura 41  
Pantalla del menú principal

## Menú de selección de música

El menú de selección de música está compuesto por tres pantallas auxiliares que representan las tres bandas sonoras disponibles para escuchar durante la partida. De igual modo que ocurre tanto en la versión recreativa como en la versión de Sega, estas pantallas sólo son accesibles después de presionar el botón Start en el menú principal. En este menú el usuario, haciendo uso de las flechas del cursor izquierda y derecha, puede determinar qué melodía emplear. Un hecho a destacar de este menú que no está presente en ninguna de las versiones empleadas como referencia es que se reproduce la pista musical que el jugador tiene seleccionada. De este modo, hay total seguridad de que el jugador comienza la partida con la melodía que más le gusta.



Figura 42  
Pantalla con la primera pista musical disponible  
*Magical Sound Shower*



Figura 43  
Pantalla con la segunda pista musical disponible  
*Passing Breeze*



Figura 44

Pantalla con la tercera pista musical disponible *Splash Wave*

## Pantalla de juego

Esta pantalla está visible en todo momento mientras el jugador se encuentra en la partida. Se ha intentado que la HUD (Head-Up Display) sea lo más fiel posible a la versión recreativa. Seguidamente, se muestra una comparativa entre esta interfaz y la de la recreativa.



Figura 45

HUD de la versión recreativa de *Out Run* de 1986



Figura 46  
HUD de la versión diseñada de *Out Run*

## Pantalla de Game over

Esta pantalla de la interfaz aparece, de manera automática, durante la partida cuando al jugador se le acaba el tiempo dado para completar el nivel (indicador de la esquina superior izquierda). Para salir de ahí basta con pulsar *Start*.



Figura 47  
Pantalla de Game Over

## Menú de pausa

Este menú es una de las mejoras que se ha integrado a la versión original. El menú de pausa solamente puede ser accesible desde la pantalla de juego presionando el botón ESC. En dicho menú aparecen cuatro posibles botones. El primer botón representa la opción “Resume”, de tal forma que si el jugador presiona *Start* teniendo dicho botón seleccionado se reanuda la partida. El segundo botón representa la opción de “Options”. Este botón fue creado con el objetivo de permitir al usuario que cambiase tanto las opciones de juego al principio como durante la partida. El tercer botón representa la opción “Home”, que permite volver a la pantalla de inicio del juego. Por último, el botón con la opción “Exit”, si esta opción es seleccionada el juego termina su ejecución de manera directa.

Además, para una mayor flexibilidad la opción que tiene marcada el jugador está pintada de un color distinto a las que no. De este modo, el jugador a nivel visual sabe qué opción tiene marcada actualmente.



Figura 48  
Menú de pausa

## Menú de opciones

Este menú puede ser accesible, como ya se ha explicado anteriormente, tanto desde el menú principal teniendo marcada la opción de “Options” y pulsando *Start* o desde el menú de pausa, seleccionando el botón de “Options” y presionando *Start*. En dicho menú de configuración es donde se lleva a cabo la gestión de todos los parámetros de configuración del juego. Las flechas del cursor arriba y abajo permiten al jugador moverse entre las

distintas opciones de configuración. Nótese que al igual que ocurría en el menú de pausa las opción marcada está de un color distinto a las que no. Seguidamente se explica cómo el usuario puede configurar las diferentes opciones de juego.

Al entrar en el menú de opciones, por defecto, siempre se encuentra marcada la opción de dificultad. Usando las flechas del cursor izquierda y derecha el usuario puede seleccionar el modo de dificultad que más le guste. La siguiente opción determina si los enemigos están controlados por la IA. Este parámetro está deshabilitado por defecto ya que el juego original no presenta ninguna IA (los coches siguen movimientos estáticos sin depender del jugador), pero se puede habilitar y la IA toma el control de los enemigos.

Las tres opciones siguientes representan menús para cambiar el ajuste de las bandas musicales y efectos de sonido, de los gráficos (resolución de pantalla y pixel art) y el manejo del vehículo del jugador.

A diferencia de las opciones anteriores, para poder acceder a los submenús que permiten modificar estos parámetros se debe pulsar C y no *Start*.



Figura 49  
Menú de opciones de *Out Run*

## Menú de ajustes de sonido

Este menú de la interfaz solamente puede ser accesible desde el menú de opciones si y sólo si la opción de *Sound* está marcada. Al presionar C, aparece un submenú en el que el jugador puede modificar dos posibles parámetros, el volumen de efectos y el volumen de bandas musicales. Por defecto, el volumen de las bandas musicales está al 80% y el de los

efectos al 100%. Usando las teclas del cursor izquierda y derecha el jugador puede determinar el volumen de efectos y de música con los que jugar. Una funcionalidad interesante que se ha incluído en este punto ha sido la opción de que los volúmenes se incrementen o decrementen de uno en uno con el fin de que el jugador tenga un mayor control de los rangos de valores permitidos. Además, el volumen es ajustado de manera automática en directo, es decir, mientras el jugador modifica ambos parámetros, se van reproduciendo los sonidos con la nueva magnitud de volumen.

Una vez que el usuario está conforme con los valores seleccionados, pulsando *Start*, vuelve al menú de opciones con la nueva configuración de sonido guardada.



Figura 50  
Menú de ajustes de sonidos

## Menú de ajustes de gráficos

El menú de ajuste de gráficos sólo es accesible desde el menú de opciones teniendo marcada la opción de “Graphics”. Al presionar C, aparece un submenú en el que el jugador puede modificar dos posibles parámetros, la resolución de la pantalla y la utilización del pixel art. Por defecto, la resolución está en 921 x 691 píxeles y el pixel art está habilitado, esto es debido a que la carretera y los sprites concuerdan mejor con la versión recreativa. Usando las teclas del cursor izquierda y derecha el jugador puede modificar ambos parámetros. Cabe destacar que existen una gran cantidad de resoluciones, y ante ello se han escogido de entre las más comunes, las que a continuación se exponen:

- Pantalla completa.
- 921 x 691.
- 1280 x 720.
- 1366 x 768.
- 1920 x 1080.
- 2560 x 1440.
- 3640 x 2160.

Todas las medidas anteriormente citadas están en píxeles. Las resoluciones 2560 x 1440 y 3640 x 2160 han sido pensadas para situaciones en las que se desea probar el juego en dispositivos con un mayor grado de resolución que las pantallas de los computadores normales. Por ejemplo, son muy prácticas en casos en los que se desea conectar el computador a una tv.

Además, la resolución definida como “pantalla completa” es una resolución responsive ya que se adapta a las dimensiones de la pantalla del computador en el que el juego está siendo ejecutado. Una vez que el usuario está conforme con los valores seleccionados, pulsando *Start*, vuelve al menú de opciones con la nueva configuración de gráficos.



Figura 51  
Menú de ajustes de gráficos

## Menú de configuración del vehículo

Esta pantalla es accedida al igual que las anteriores desde el menú de opciones. Para ello, el jugador debe tener seleccionada la opción “Controllers” y presionar la tecla C. Tras presionar dicha tecla se abre un nuevo submenú en el que el jugador puede determinar los controles para acelerar, frenar, y girar el vehículo tanto a la izquierda como a la derecha. Para poder optar a cambiar un parámetro de control se debe presionar Espacio. Seguidamente, se debe presionar la tecla que se desea asignar para poder controlar el vehículo.

En este apartado es importante resaltar que no se han hecho jugables todas las teclas del teclado, solamente aquellas que son reconocidas de acuerdo al framework SFML. De este modo, cuando un usuario presiona una tecla pueden ocurrir uno de los siguientes tres casos:

- Si la tecla presionada es válida, es decir, es reconocida por SFML y es jugable, se emite un sonido de cambio para así poder alertar al usuario de que la nueva tecla ha sido aceptada.
- Si ha metido una tecla que no es conocida por la propia librería de SFML, como por ejemplo son las teclas F1.. FN, la tecla de impresión de pantalla, la tecla de mayúsculas, etcétera, el videojuego no emite ningún tipo de sonido. Si ocurre esta situación es porque las teclas que está metiendo el usuario están siendo ignoradas.
- Si ha metido una tecla que es inválida, osea que es reconocida por SFML pero no es jugable se emite un sonido de corrección para alertar al usuario de que debe escoger otra.

Para poder visualizar en el menú la nueva tecla metida por el usuario, se debe soltar la tecla Espacio. Al soltar dicha tecla, se confirma la nueva configuración y se actualizan los nuevos cambios. Si la tecla no se suelta, no se hacen visibles los nuevos cambios puesto que el control todavía está siendo editado por el jugador.

Para salir del menú de configuración del vehículo tras estar conforme con los nuevos cambios, se debe presionar *Start* para regresar al menú de opciones.



Figura 52  
Menú de configuración del vehículo del jugador

## Menú de ranking

Esta es la última pantalla del juego, y aparece cuando el jugador ha completado los cinco escenarios obligatorios. Este menú, al igual que las animaciones de salida y llegada, se ha intentado que desde un principio fuese lo más fiel posible a la máquina de la recreativa pero con algún cambio adicional. El ranking que se muestra en dicha pantalla variará en base a si el jugador en cuestión ha batido un récord o no, es decir, si se ha puesto por delante de alguno de los siete mejores jugadores.

Si el jugador ha batido un récord, en el menú se retroalimenta al usuario que meta unas iniciales para guardar su nombre. Dichas iniciales sólo pueden representarse mediante letras de la A a la Z, por lo que cualquier otra tecla es interpretada como error. En estos casos se emite un sonido de error para informar al jugador de manera directa que se ha equivocado. Tras introducir el nombre, el usuario deberá presionar *Start* para poder volver al menú principal.

Si no se bate récord se muestra el ranking de los siete mejores jugadores. Para salir se debe presionar *Start*. La presentación de los jugadores en el ranking ha sido realizada de la misma manera que en el juego original de la recreativa. Primero el nombre del jugador, seguido de los puntos y del tiempo invertido para completar la partida. Los jugadores están ordenados por orden decreciente de puntuación.



Figura 53  
Pantalla del ranking cuando el jugador no bate récord



Figura 54  
Pantalla del ranking cuando el jugador se ha clasificado dentro de los siete mejores

## Gráficos

En cuanto al tema de los gráficos se refiere, se ha optado desde un primer momento por que la apariencia del juego, en la medida de lo posible, fuese fiel a la versión de la máquina recreativa. De manera acorde a la temática de los videojuegos arcade, el videojuego se ha diseñado usando gráficos 2D a través de sprites. Todos y cada uno de los sprites

empleados están almacenados en un fichero con extensión *.png*. Dichos ficheros se encuentran estructurados en diversos subdirectorios dentro de la carpeta principal *resources*.

A continuación, se enumeran todos los subdirectorios de la carpeta *resources* donde se hallan almacenados dichos sprites junto con una breve explicación del contenido de cada una. La razón por la que se ha optado por enumerar las subcarpetas en vez de mostrar los sprites directamente en el documento estriba en el elevado número que se ha empleado. Los subdirectorios son los que seguidamente se presentan.

- *carX*: son los directorios empleados para almacenar los sprites de los distintos tipos de vehículos que son visibles en el tráfico. Cada coche ha sido almacenado en su propio directorio. Por ello, dado que hay seis tipos de vehículos de tráfico, existen seis carpetas distintas. La X toma valores en el rango del 1 al 6. Cada vehículo en cuestión tiene un total de 16 sprites.
- *Ferrari*: es el directorio usado para guardar todos los sprites que representan el vehículo Ferrari Testarossa rojo descapotable controlado por el jugador. Hay exactamente un total de 136 sprites.
- *GamePanel*: es la carpeta donde se encuentran todos los sprites que permiten construir la HUD que se visualiza durante la pantalla de juego.
- *MainMenu*: directorio donde se recogen los sprites que aparecen en la pantalla inicial del juego.
- *MapX*: conjunto de carpetas que almacenan los sprites de los diferentes escenarios. Exactamente hay un total de 15 directorios, a razón de uno por mapa, y cada una de esas carpetas almacenan los sprites de ese escenario en concreto. Nótese que algunos sprites están repetidos en varias de estas carpetas. Ésto es debido a que las operaciones tanto de relleno de texturas como de escalado de los sprites son independientes en cada mapa. La X es un número con rango de valores del 1 al 15.
- *MapCommon*: es la carpeta donde se encuentran todos los sprites usados para la construcción de las animaciones de salida y llegada del Ferrari del jugador.
- *OptionsMenu*: directorio donde se encuentran los sprites usados en la creación de los menús de configuración de opciones del videojuego.
- *MusicMenu*: carpeta donde se localizan los sprites usados para la representación de los menús de la interfaz en los que el jugador escoge la pista musical a escuchar durante la partida.
- *RankingMenu*: directorio donde se almacenan los sprites usados para la construcción de la pantalla final de ranking del videojuego.
- *SegaAnimation*: carpeta donde residen los sprites empleados para la animación del logo inicial de Sega.

# Música y SFX

El audio que ha sido empleado para la realización de esta versión de *Out Run* engloba sonidos y efectos de la entrega original de *Out Run* para la recreativa, del *Super Hang On* para la Sega Mega Drive, del *Mario Kart Super Circuit* para la Game Boy Advance, del *Cruis'n USA* de la Nintendo 64, y del *Smash Court Tennis 3* para la Playstation Portable. Para representar tanto las bandas musicales como el SFX se ha empleado el formato *ogg*. Existen numerosas razones de peso por las que se ha optado por escoger este formato de audio frente a otros como por ejemplo *flac*. En mayor detalle, las principales razones han sido que *ogg* es un formato muy recomendable por su alto grado de eficiencia en el *streaming* y en la compresión de archivos, factor que es muy importante dado que aunque las bandas sonoras se reproducen de manera estática, el SFX se comporta de manera tanto dinámica como interactiva ya que depende tanto de los eventos que ocurren en el escenario (sonidos del semáforo, colisiones de los coches, derrapes) como de acciones controladas por el propio jugador (arrancar y manejar el coche). Adicionalmente, *ogg* es un formato de contenedor multimedia de uso libre y soportado por la API de SFML.

## Pistas musicales

Los archivos de audio que se han empleado como bandas musicales del videojuego son réplicas de las pistas originales de la versión para la máquina recreativa. El juego tiene, concretamente tres pistas musicales para poder reproducir durante el transcurso de la partida, y que son elegidas como se ha explicado en apartados anteriores del documento en el menú de selección de música.

Dichas bandas sonoras son *Magical Sound Shower*, *Passing Breeze* y *Splash Wave*. La pista seleccionada de entre estas tres anteriores se reproduce de igual modo que en la versión recreativa y en la versión de Sega, en bucle y sin posibilidad de volver a cambiarse.

Adicionalmente, existen tres pistas musicales más. La primera pista se conoce como *Last Wave*, que se reproduce única y exclusivamente durante la pantalla de ranking. La segunda pista se reproduce tanto en el menú como en los submenús de opciones. Esta pista es concretamente la extraída del *Smash Court Tennis 3*, lanzado en 2007. Pese a ser de un videojuego mucho más reciente que el *Out Run*, se ha optado por escoger este audio porque es una melodía que puede pasar sin ninguna dificultad como un sonido de 16-bits. Por último existe una melodía más, que se reproduce durante la pantalla de Game Over, esta banda sonora es conocida como “*Finished*”, extraída del *Super Hang On*.

## SFX

La versión de *Out Run* diseñada cuenta con un total de 31 efectos de sonido. Algunos de los efectos más importantes son por ejemplo los sonido de acelerar y derrapar del ferrari, los sonidos que se reproducen cuando el jugador se mueve por la interfaz de los botones, el sonido del tráfico, etcétera.

# Fuentes de texto

La fuentes de texto que aparecen en esta versión de *Out Run* han sido buscadas con el fin de otorgar una apariencia textual muy similar a la de las entregas originales. Un ejemplo de esta decisión puede observarse en las fuentes empleadas para la elaboración de la HUD, visible durante la partida. Además, todas estas fuentes se caracterizan por ser de uso libre y totalmente gratuitas. Dichas fuentes de texto pueden encontrarse en el subdirectorio *Fonts* de la carpeta principal *resources*. Cada fuente de texto está almacenada en un fichero *.ttf*.

## Detalles de implementación

A continuación, se detallan los principales aspectos de implementación.

### Back doors

Se han diseñado puertas traseras a modo de depuración para poder establecer el mapa inicial y el tiempo deseados. Para su activación, se debe crear un fichero con nombre “backdoor.info” en la carpeta “resources”. Este fichero puede contener los siguientes atributos (separados con espacios o finales de línea):

- **Mapa inicial.** Para establecer el mapa de inicio se debe indicar el argumento MAP, seguido de dos puntos, seguido del número de mapa (1 a 15). Ej: MAP: 15
- **Tiempo extra.** Para establecer el tiempo extra de partida se debe indicar el argumento TIME, seguido de dos puntos, seguido del tiempo extra en segundos. Ej: TIME: 500

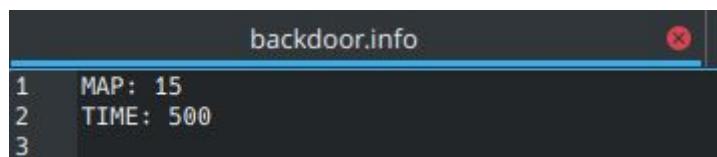


Figura 55  
Ejemplo de fichero “backdoor.info”.

### Cámara y renderizado

El mapa está dividido en rectángulos (explicados en el apartado del generador automático de mapas), y a su vez cada rectángulo está dividido en otros más pequeños (denominados “Lines” o líneas). Ésto se ha diseñado así para que los objetos no queden demasiado juntos y el coche no avance a saltos de rectángulo en rectángulo, sino que avance poco a poco dentro de cada rectángulo.

Se ha posicionado una cámara detrás y encima del coche del jugador (que es la que realmente avanza por la carretera, ya que el coche se mantiene en la misma posición de la pantalla). Esta cámara tiene una distancia de renderizado establecida en 300 líneas, es decir, se muestran todos los vehículos, objetos, carretera y suelo que estén a 300 líneas o menos de la posición del jugador. El renderizado comienza por la línea más alejada y termina en la línea del jugador (de esta manera se ven primero los objetos y vehículos más cercanos).

Según la distancia a la que se encuentra la línea a renderizar, se escala todo su contenido a un valor u otro. De esta manera se consigue que las líneas más alejadas se muestren con una escala más pequeña y las líneas más cercanas con una escala más grande, es decir, se consigue una sensación de profundidad.

Debido a que la carretera y algunas imágenes tienen una gran resolución, se ha diseñado un efecto de pixel art que consiste en renderizar la imagen en una resolución menor y escalarla a la resolución de pantalla. De esta manera se consigue que los píxeles sean más visibles y dé la impresión de un efecto retro. En concreto, para la resolución 921 x 691 se renderiza la imagen en 460 x 345, y para las demás resoluciones se renderiza la imagen en 640 x 360. Los menús siguen siendo generados para la resolución nativa de la pantalla, ya que si se realiza el mismo proceso de escalado la letra resulta ilegible.

Además, se ha forzado la renderización de 60 imágenes por segundo (60 fps). De esta manera, los tiempos y lógica utilizada siempre serán los mismos (si no se llegaran a estos fps la jugabilidad se vería reducida).

## Generación de enemigos

El número de vehículos enemigos es estático en todo momento y depende de la dificultad. En modo pacífico (PEACEFUL) no hay ningún enemigo, en fácil (EASY) hay 5 enemigos, en normal (NORMAL) hay 10 enemigos y en difícil (HARD) hay 15 enemigos.

Los enemigos se eliminan cuando se entra en una bifurcación (por simplicidad técnica) y cuando el jugador los ha dejado atrás a una distancia de, al menos, 50 líneas.

Los vehículos enemigos se generan al inicio de cada mapa o cuando se han eliminado por quedarse atrás. Cada enemigo es diferente de los demás, y a continuación se detalla el proceso de generación de cada uno:

- La velocidad inicial se establece de manera aleatoria con un valor comprendido entre el 25% y el 75% de la velocidad máxima del jugador (300 km/h).
- La posición inicial se establece de manera aleatoria con un valor comprendido entre la línea del último enemigo generado + 25 (o la línea renderizada más alejada si el último enemigo generado está dentro del campo de visión) y 100 líneas más a dicho valor (ej. entre 500 y 600). De esta manera se consigue que no aparezcan enemigos

de la nada dentro del campo de visión del jugador o que aparezcan múltiples enemigos en la misma línea.

- La agresividad de la IA se establece de manera aleatoria con un valor comprendido entre la mitad del valor de la agresividad global y el valor de la agresividad global. El tipo de IA se establece de manera aleatoria entre los tres posibles tipos disponibles. En el siguiente apartado se detalla en qué consisten estos valores.

## Inteligencia Artificial

Existen 4 comportamientos diferentes para cada enemigo: el original sin IA (por defecto) y tres tipos de IA diferentes (OBSTACLE, EVASIVE y INCONSTANT).

En cada movimiento actúa la IA o no de manera aleatoria en función de la agresividad del vehículo. La agresividad de un vehículo está comprendida por un valor entre 0.0 y agresividad global (en el caso de que la IA sea de tipo INCONSTANT su agresividad se multiplica x2). La agresividad global es 0.0 si la IA está desactivada o la dificultad es pacífica (los coches se comportan como en el juego original), 0.25 si la IA está activada y la dificultad es fácil, 0.5 si la IA está activada y la dificultad es normal, y 0.75 si la IA está activada y la dificultad es difícil.

Si en el movimiento actual la distancia entre el jugador y el vehículo enemigo es mayor a la distancia de renderizado (el jugador no ve al enemigo), o se elige un número aleatorio entre 0.0 y 1.0 y es mayor a la agresividad de la IA del vehículo, no actúa la IA y el comportamiento del vehículo es como en el original. Este comportamiento consiste en mantener la velocidad y trayectoria constantes. La trayectoria varía de manera aleatoria cada cierto tiempo, pero la mayor parte de las trayectorias serán rectas.

En caso contrario (no se cumple ninguna de las condiciones anteriores) actúa la IA, que dependiendo de su tipo tiene un comportamiento u otro:

- **OBSTACLE.** Este modo intenta chocar con el jugador poniéndose en su trayectoria e intentando alcanzarlo de la siguiente manera: Si el vehículo no se encuentra en la trayectoria del jugador, realiza un giro para acercarse a la trayectoria del jugador. Si el vehículo se encuentra en la trayectoria del jugador y está por detrás suyo, acelera para intentar chocar con el jugador. Si el vehículo se encuentra en la trayectoria del jugador y está por delante suyo, frena para intentar chocar con el jugador.
- **EVASIVE.** Este modo hace que el vehículo huya al carril más alejado del jugador para intentar evitarlo.
- **INCONSTANT.** Este modo hace que cambie de carriles muy a menudo sin tener en cuenta la posición del jugador. Cuando llega a un cambio de carril decide de manera aleatoria si realizar el cambio de carril o seguir en el carril que está.

## Físicas del vehículo

El vehículo del jugador posee una variable que regula la aceleración y otra que regula la velocidad (no son km/h, es una medida inventada), cuya relación es la siguiente:

$$speed = \sqrt{acceleration}$$

De esta manera se consigue que el momento donde más rápido gane velocidad el coche sea durante el arranque, cuya gráfica se detalla a continuación:

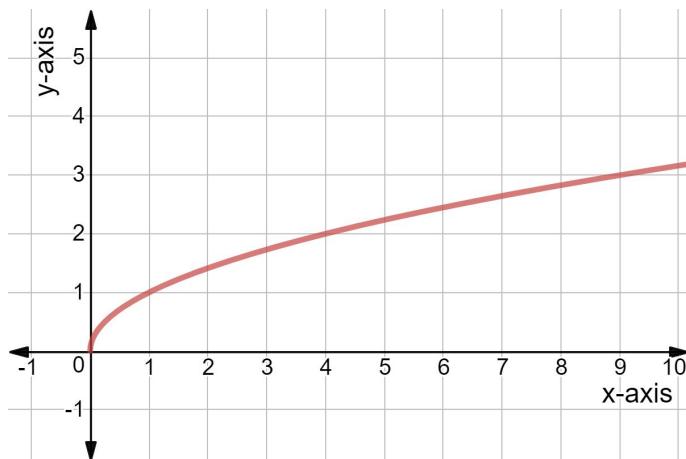


Figura 56  
Gráfica que relaciona la aceleración (eje X)  
con la velocidad del Ferrari (eje Y)

Para que el jugador vea la velocidad en km/h, se multiplica la velocidad por un multiplicador que relaciona, de manera ficticia, la magnitud de velocidad del vehículo con km/h.

En cada movimiento se incrementa o decrementa la aceleración, y la velocidad es sólo el resultado final. Para que la velocidad y la aceleración no crezcan indefinidamente, se ha limitado la aceleración máxima a un valor predefinido (calculado mediante las relaciones anteriores).

Si el jugador acelera, la aceleración se incrementa en 0.01. Si el jugador frena, la aceleración se decrementa en 0.02. Si el jugador no acelera ni frena, la aceleración se reduce en 0.01. Además, si el vehículo se ha salido de la carretera, la aceleración máxima se verá reducida a una cuarta parte y media de esta.

Los giros tanto a derecha como a izquierda dependen de la velocidad del vehículo, cuanto más rápido va el vehículo más gira. Una curva empuja al vehículo en dirección contraria a la curva, efecto causado en la física por la fuerza centrífuga, siendo este empuje proporcional al índice de curvatura de la línea en la que se encuentra.

Si el vehículo colisiona, se bloquea el control del vehículo y se reproduce una animación que reduce la aceleración y velocidad del vehículo dependiendo del tipo de choque, como se ha explicado en el apartado “Ferrari Testarossa” de “Personajes”.

## Geometría del mapa

Además de la división en rectángulos del mapa detallada en apartados anteriores, caben destacar los modelados de las curvas, las elevaciones y las bifurcaciones.

### Curvas

Una curva está representada por un sumatorio de los índices de curvatura de las líneas adyacentes, es decir, si la cámara se encuentra en una línea con índice de curvatura mayor que 0.0 y todas sus líneas posteriores (dentro de la distancia de renderizado) tienen un índice de curvatura mayor que 0.0, significa que el jugador está dentro de una curva hacia la derecha que perdura durante todo el campo de visión del jugador.

En definitiva, si la cámara se encuentra en una línea con índice de curvatura distinto de 0.0 significa que el jugador está en una curva, y si es 0.0 es que está en una recta. Además, si alguna de las líneas (varias para que se aprecie) dentro del campo de visión del jugador tiene un índice de curvatura distinto al de la línea en la que se encuentra la cámara, se apreciará cómo varía la trayectoria de la carretera.

### Elevaciones

Cada mapa tiene una elevación inicial que viene determinada por la elevación final del mapa anterior (0.0 en la mapa inicial). De esta manera se cambia de mapa sin producirse ningún salto.

La elevación del propio mapa está modelada mediante la función coseno, ya que así se produce de manera progresiva simulando una colina. Para ello, cada rectángulo tiene una elevación determinada por la siguiente ecuación (las líneas que forman el rectángulo realizan un incremento de la elevación de manera equitativa [ej. un rectángulo incrementa su elevación en 100 respecto a su rectángulo predecesor, y este rectángulo está formado por 5 líneas, por lo que cada línea incrementa su elevación en 20 respecto a su predecesora]):

$$elevation = initialElevation + elevationIndex + elevationIndex \times \cos(pi + \frac{pi \times lineIndex}{numLines})$$

Figura 57  
Ecuación de cálculo de cambio de rasante

La elevación inicial corresponde a la elevación que había antes de comenzar esta zona de elevación (inicialmente es la elevación final del mapa anterior).

El índice de elevación es el correspondiente al establecido en el generador de mapas (positivo en las subidas y negativo en las bajadas).

El número de líneas es el correspondiente al número de líneas que tiene la zona de elevación, y el índice de línea es un valor correspondiente entre 0 y número de líneas - 1.

A continuación se muestra un ejemplo con elevación inicial = 0, índice de elevación = 1.5 y número de líneas = 10. Como se puede observar, en la línea 10 ha alcanzado su punto máximo.

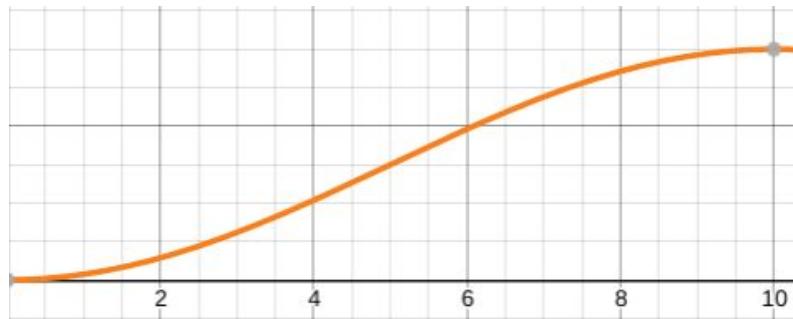


Figura 58  
Gráfica que muestra el modelado de cambios de rasante

## Bifurcaciones

Al final de cada mapa de las etapas 1 a 4 se genera una bifurcación que conecta el mapa actual con sus dos vecinos de la siguiente etapa. Esta bifurcación está modelada mediante la unión de los fragmentos de 2 circunferencias con centros y radios precalculados. De esta manera y sin girar la cámara, se divide la carretera en 2 idénticas (con los colores del mapa con el que se conecta). Cuando el jugador está en un lado u otro de la bifurcación se carga un mapa u otro.

Sólo se calcula la bifurcación izquierda, ya que la bifurcación derecha es su reflejo (simetría). En la bifurcación a calcular, existen 2 puntos de corte, el primero conecta la primera circunferencia con la segunda, y el segundo conecta la segunda circunferencia con el resto de la carretera recta hasta comenzar el siguiente mapa.

La primera circunferencia está modelada por la siguiente ecuación, para que se curve hacia la izquierda desde el origen (0, 0):

$$x = -\sqrt{R^2 - index^2} + R$$

Donde x es el centro de la carretera, R el radio de la circunferencia (con valor 10) e index es el índice de la línea.

La segunda circunferencia está modelada por la siguiente ecuación:

$$x = \sqrt{R^2 - (index - A)^2} + B$$

Donde A y B son valores constantes que se han precalculado para que el primer punto de corte entre las dos circunferencias coincida con los primeros 45° recorridos de la primera circunferencia desde el origen, y el segundo punto de corte coincida con los primeros 45° recorridos de la segunda circunferencia desde el primer punto de corte. De esta manera se consigue que no haya una curva imposible y se restablece la trayectoria de la carretera hacia la misma dirección desde la que se partía.

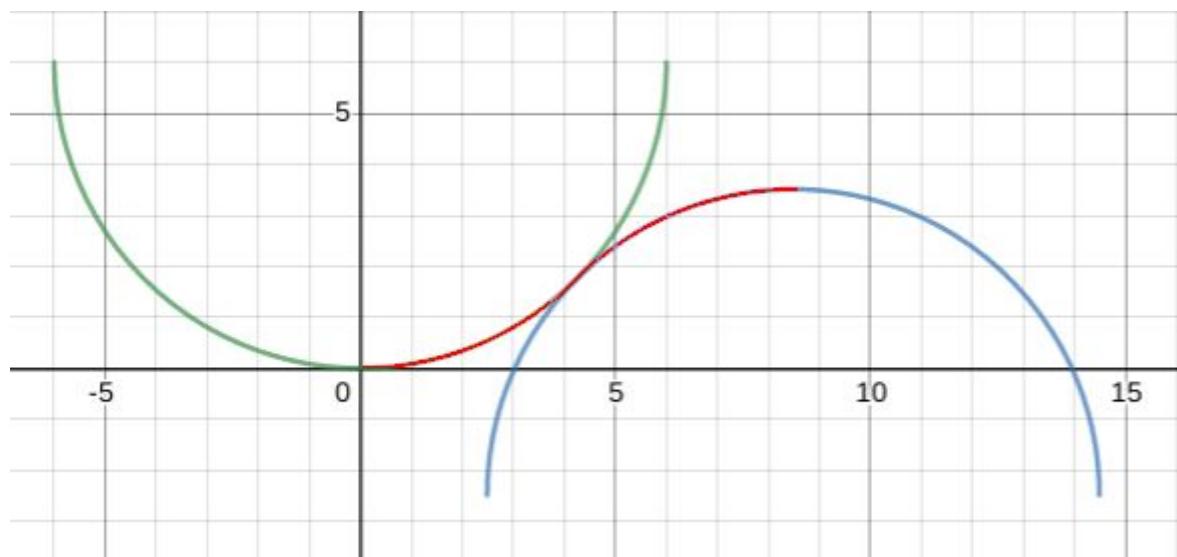


Figura 59  
Gráfica que muestra la apariencia de las bifurcaciones

La semicircunferencia verde representa la primera circunferencia, la semicircunferencia azul representa la segunda circunferencia, y el trazado rojo representa el trazado de la carretera en la bifurcación izquierda. Considerando sólo el trazado rojo y creando un reflejo simétrico se obtiene el trazado real de la bifurcación:

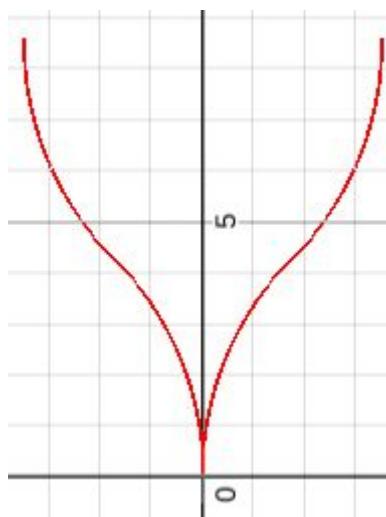


Figura 60  
Gráfica que muestra el renderizado de las bifurcaciones



Figura 61  
Representación visual de las bifurcaciones

## Geometría 3D

Cada objeto y vehículo (y la cámara) está representado mediante un sistema de coordenadas en 3D del mapa, este sistema de coordenadas está formado por los ejes X, Y y Z que se describen a continuación:

- El eje Z representa la línea en la que se encuentra dicho objeto o vehículo (desde la línea de salida, hasta la línea de conexión con el siguiente mapa).
- El eje Y representa la elevación de dicha línea (tanto positiva como negativa).
- El eje X representa la ubicación horizontal en la línea en la que se encuentra, siendo 0.0 el centro de la carretera, mayor que 0.0 la parte derecha de la carretera y menor que 0.0 la parte izquierda de la carretera. La carretera ocupa desde -1.0 hasta 1.0.

Gracias a este sistema de coordenadas en 3D se puede representar la profundidad del mapa, y escalar las líneas más alejadas como se ha mencionado en el apartado “Cámara y renderizado”.

## Salida y meta

Debido a las limitaciones del generador de mapas, la salida y la meta son añadidas de manera automática al mapa inicial y al mapa final respectivamente.

Al mapa inicial se le añaden, al principio, un conjunto de rectángulos que contienen todos los sprites de la salida (incluidos el abanderado y el semáforo). Durante la animación, el vehículo del jugador está bloqueado (sin poder acelerar) y se cambian dichos sprites acordes a los tiempos que han transcurrido y así generar un efecto visual de movimiento. De la misma manera, el vehículo del jugador realiza un derrape al comienzo de la animación.

Al mapa final se le añaden, al final, un conjunto de rectángulos que contienen todos los sprites de la meta (incluido el abanderado). Durante la animación, el vehículo del jugador mantiene una velocidad constante mientras aparecen mensajes por pantalla (sin poder acelerar ni frenar) y va cambiando dicho sprite para generar un efecto de movimiento similar al de la animación anterior (no se ha realizado una meta más exuberante debido a limitaciones técnicas y temporales). De la misma manera, el vehículo del jugador realiza un derrape al final de la animación.

## Mapeador de teclas

El mapeador de teclas es un módulo auxiliar empleado única y exclusivamente para la configuración del control del Ferrari. La razón por la que se ha visto la necesidad de implementar este componente reside en que la API de SFML es capaz de llevar a cabo la captura de teclas en base a un código numérico, asociado por defecto. Dichos códigos de tecla están configurados en la propia API y no pueden ser modificados.

El mapeador cuenta con un registro interno que relaciona los códigos de las teclas con representaciones literales de éstos. Por ejemplo, al código de la tecla A, que es el 0, el mapeador le asigna la etiqueta 'A', y así de manera sucesiva con el resto de teclas.

Cuando el jugador modifica un parámetro de control del vehículo, se obtiene el código de la tecla que ha pulsado en cuestión. Dicho código es leído por el mapeador de teclas, y acto seguido obtiene la etiqueta asignada a dicho código para mostrarla. De este modo, el jugador puede verificar visualmente que el cambio de control de tecla ha resultado válido y efectivo.

## Aleatoriedad

Se ha hecho un gran uso de la aleatoriedad en este juego, con motivo de incrementar el dinamismo y que todas las partidas sean diferentes. Para ello se han utilizado valores aleatorios controlados, es decir, no se permite cualquier situación, sino que en cada acción o generación aleatoria se elige entre un conjunto de posibilidades. No todas las posibilidades tienen la misma probabilidad, de este modo se evita que siempre esté

actuando la IA (sería imposible jugar, ya que el ordenador siempre es más rápido), o que todos los enemigos tengan siempre la misma velocidad y posición, etc.

Para que la función de aleatoriedad sea suficientemente uniforme, se ha elegido el generador de números pseudoaleatorios de C++ mt19937.

## Audio

La API de SFML da soporte a la manipulación de archivos de audio por medio de dos tipos de datos predefinidos, *Sound* y *Music*. Ambos hacen las mismas tareas pero de forma diferente, pues *Sound* para poder reproducir los sonidos debe cargarlos en memoria desde el archivo en cuestión mientras que *Music* abre el fichero y reproduce el sonido en streaming, es decir, es mucho más rápida y eficiente esta gestión, sobre todo, si se va a reproducir sonidos largos o repetidos de forma periódica. Por consiguiente, se ha usado *Music* para poder gestionar internamente tanto el audio como el SFX del videojuego. Además, dicha gestión ha sido dinámica, con punteros inteligentes, que permiten la gestión automática de la memoria abstrayendo completamente al programador de ciertas tareas como el borrado de memoria.

Dada la gran cantidad de audio que se emplean en el diseño de esta versión de *Out Run*, se ha optado por almacenar por un lado las pistas musicales del videojuego y por otro el SFX. Así, las bandas musicales se hallan almacenadas en el subdirectorio *Soundtrack* y los efectos de sonido en el subdirectorio *SoundEffects*, ambos localizados en la carpeta *resources*.

## Gestión del proyecto software

El desarrollo del videojuego del *Out Run*, ha sido sin duda uno de los proyectos que más ha costado hacer, principalmente, porque ha sido el primer videojuego que hemos realizado. Para simplificar los posibles problemas que pudiesen ocurrir desde un principio se ha llevado una clara organización de las diferentes tareas que había que hacer, así como el tiempo para poder hacerlas.

El código del proyecto ha estado sometido en todo momento a un sistema de control de versiones. Como plataforma se ha escogido *Github*. De este modo, el código ha permanecido tanto en los propios computadores locales de los miembros de equipo como en el correspondiente repositorio en *Github*.

Los cambios en el código se han efectuado de la siguiente manera. En el repositorio existen tres ramas. La rama principal es la rama *master*, donde únicamente se encuentra código funcionalmente correcto. Paralelamente, existen otras dos ramas auxiliares más, a razón de una por programador. Estas ramas han sido creadas para que cada programador realice sus tareas personales ahí.

Cuando una tarea era completada, se hacía el correspondiente merge entre la rama del programador en cuestión y la rama *master*. De esta manera, el código siempre estaba actualizado y correcto.

Adicionalmente, se proporciona un diagrama de Gantt, adjunto en un fichero denominado *Gantt.pdf*, en el que se refleja la planificación seguida para la realización del videojuego, así como la distribución de tareas y el tiempo dedicado por cada miembro del equipo. Para verlo de manera nítida se recomienda agrandarlo ya que en dicho diagrama se recogen todos los días que han compuesto el periodo de desarrollo del videojuego.

## Software utilizado

Para la implementación de esta versión de *Out Run* se ha usado el lenguaje de programación en alto nivel C++ dado que es un lenguaje compilado muy rápido, tiene un alto rendimiento, es multiplataforma, y además está en continua actualización.

Para llevar a cabo el diseño de gráficos y sonidos se ha usado la API de SFML, aunque también se había optado por emplear otras alternativas como Allegro o SDL. Sin embargo, tras la lectura de la documentación por parte de ambos miembros del equipo, se optó por escoger SFML ya que es gratuita, resulta fácil de configurar, ofrece soporte para Windows y tiene una amplia comunidad de usuarios.

Para llevar a cabo la compilación se ha empleado el compilador de C++ para Windows, llamado MinGW. Además, cabe destacar que el código se ha compilado con el máximo nivel de rendimiento (flag -O3 en las opciones de optimización).

Para poder implementar el videojuego se han empleado dos IDE'S, CLion (versión 2020.1) y CodeBlocks (versión 17.02). Con CLion se ha trabajado desde Linux (Ubuntu 19.10) y con CodeBlocks se ha trabajado desde Windows (Windows 10), pero el desarrollo ha sido para Windows.

## Problemas encontrados

A lo largo de la elaboración del proyecto se han tenido que abordar diversos problemas que han dificultado el progreso a la hora de diseñar el videojuego. Algunos de los más destacados son los que a continuación se citan.

- Ninguno de los dos miembros del equipo conocía SFML para la creación de videojuegos. Por lo que fue necesario dedicar tiempo para poder leer y comprender toda la documentación de la API, siendo en algunos casos difícil de encontrar ejemplos.
- La familiarización con los videojuegos de carreras en 2.5D, también denominados pseudo 3D racing, supuso un obstáculo difícil de superar en un primer momento dado que a pesar de haber leído ambos miembros del equipo la documentación, no

se conseguían generar escenarios ricos a nivel visual ni tampoco muy realistas. También fue difícil de controlar la cámara, especialmente en elevaciones, curvas y bifurcaciones.

- La generación de los mapas de los escenarios fue otro problema importante al que hacer frente ya que solamente se han podido obtener los sprites auténticos del primer nivel de la versión recreativa. El resto de escenarios tuvieron que ser inventados de cero por los desarrolladores. Además, hubo que pensar una técnica para poder generar los mapas de una forma automática desde fichero en vez de estar “hardcodeados”. Hubo que pensar una metodología sencilla y rápida tanto de programar como de “debuggear” para depurar los posibles errores. Estas controversias en cuestión desencadenaron ciertas discusiones entre los miembros del equipo.
- La generación de las bifurcaciones también supuso un reto dado que costó encontrar una técnica que permitiese la generación de éstas de una forma realista y adecuada desde el punto de vista estético. Además, hubo que hacer frente a diversos cálculos trigonométricos complejos y a largos períodos de pruebas hasta verificar que el videojuego se comportaba de manera aceptable.
- La incorporación de las distintas resoluciones jugables fue otro problema que costó solventar por motivos de diversa índole. En primer lugar no había un consenso claro entre los miembros del grupo sobre qué resoluciones permitir y finalmente se optó por permitirlas todas. En segundo lugar, el redimensionado y posicionamiento de todos los sprites, los elementos de la HUD y los de la interfaz fueron un quebradero de cabeza a la hora de intentar encontrar un patrón de escalado que se adaptara bien a todas las resoluciones propuestas.
- Otro problema fue la necesidad de intentar compatibilizar la resolución y calidad entre los elementos del paisaje y el propio mapa ya que algunos tenían una calidad de imagen muy buena y otros no tanta. Esta incompatibilidad de aspectos no gustó, entre otros, a los propios miembros del equipo. Por ello, hubo que habilitar un modo de pixel art para conseguir el efecto retro global deseado.
- Proporcionar al jugador control del vehículo a través de un gamepad o mando también fue un duro reto que no se pudo superar debido a que SFML no ofrece compatibilidad con los drivers de ciertos controladores como es el caso de los mandos de PS3, que no eran detectados. De igual modo, se intentó probar con mandos de PS2 pero la librería por razones desconocidas no reconocía todos los botones del mando, sólo algunos.
- A última hora, para preparar el entregable de la versión Beta del juego tuvimos problemas para compilar en Windows por parte de ambos miembros del equipo. Al darnos cuenta en el último momento quedó sin resolver algún fallo con el ratón que hacía que el juego dejase de responder, pero pudimos solucionarlo rápidamente.

# Bibliografía

[1] Información sobre videojuego de Out Run

"Out Run - Wikipedia, la enciclopedia libre." [https://es.wikipedia.org/wiki/Out\\_Run](https://es.wikipedia.org/wiki/Out_Run).

[2] Documentación de SFML

"Main Page (SFML / Learn / 2.5.1 ...." <https://www.sfml-dev.org/documentation/2.5.1/>.

[3] Obtención de fondos y sprites

"Shutterstock." <https://www.shutterstock.com/es/>

[4] Sprites y sonidos del Out Run original de la recreativa

"Dixan7/Outrun: A tribute to the arcade game Outrun - GitHub." <https://github.com/Dixan7/Outrun>.