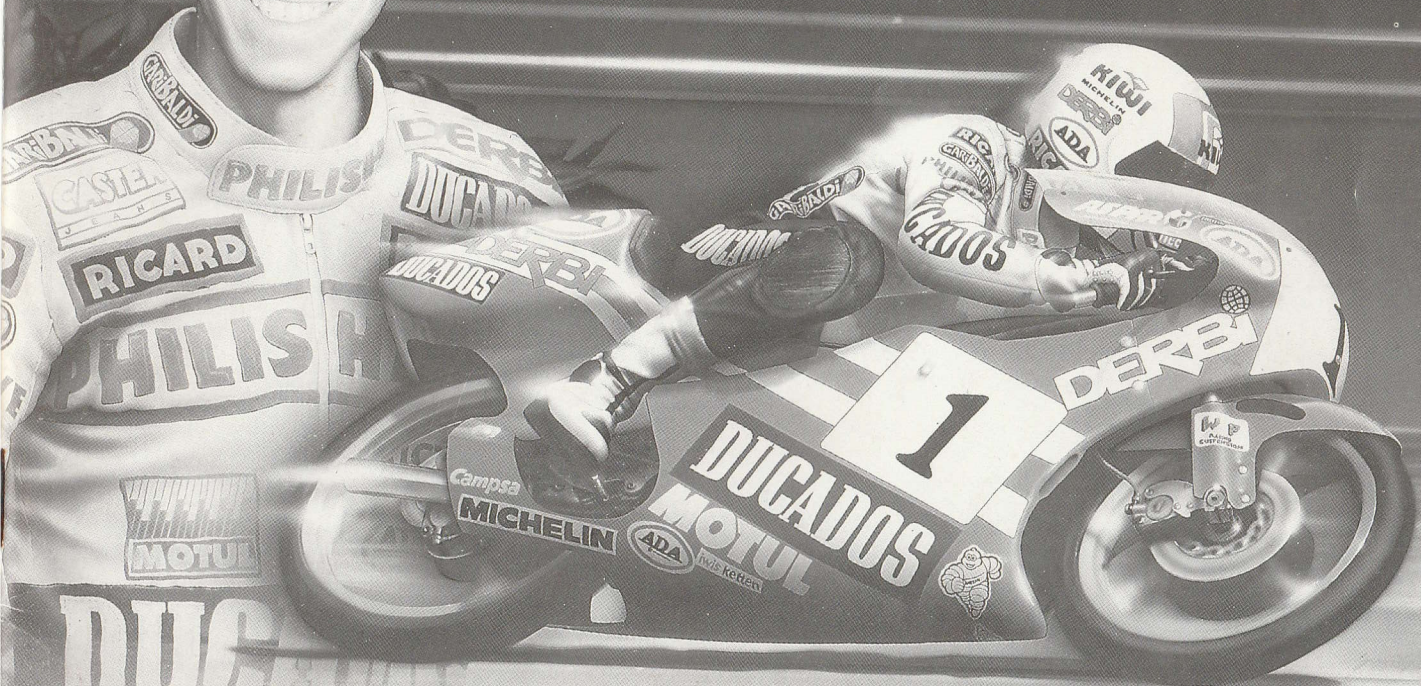


ASPAR



COMO SE HIZO ASPAR G.P. MASTER

"Sin duda uno de los proyectos más ambiciosos de Dinamic; por fin ha llegado a su meta. Acompañado de muchos problemas que han tenido que ser solucionados a lo largo de la competición, en tiempo récord.

El equipo Dinamic, una vez más, ha pisado a fondo el acelerador para conseguir esta *pole position*."

JAVIER CUBEDO
Productor Artístico

LOS SEIS G.P. (GRANDES PROBLEMAS)

Muchos han sido los problemas que hemos tenido que afrontar desde que nos pusimos a trabajar en el mes de Mayo del 88, sólo algo teníamos muy claro, hasta el más mínimo detalle del juego, debería aportar innovación en lo que se refiere a este tipo de simulaciones deportivas. Para empezar sólo contábamos con un nombre: "Aspar", y una idea "Campeonato del Mundo de Velocidad"; ahí empezó todo.

Decidimos en primer lugar darle una perspectiva distinta a la de todos los juegos existentes de características similares. Finalmente elegimos la panorámica superior de plano picado (visto desde arriba). Este fue nuestro primer gran problema, idear un sistema de construcción de los circuitos que nos permitiese reproducir cualquiera de ellos de una manera más o menos cómoda. Además, que pudiesen estar en memoria los siete circuitos que componen el Campeonato del Mundo de Velocidad de 80 cc. 1988, y evitar de esta manera la carga múltiple.

1. CREACION DE LOS CIRCUITOS

El sistema de creación de los circuitos ha sido complejo en su creación, pero ha facilitado, con la ayuda de un mapeador, la elaboración de los siete circuitos, aunque previamente se han tenido que diseñar cada uno de ellos en papel.

Con un total de 12K. de gráficos partidos en bloques de 2×2 caracteres se forman piezas más grandes representando diversas porciones de trazado: curvas, rectas, etc., de tamaños variables, con un sistema de encaje que permite enlazar unas piezas con otras como si de un "Scalextric" se tratase.

Se trabaja en ocho direcciones y un octavo de círculo es la porción más pequeña de curva que podemos representar, que junto con otro octavo forman un cuadrante de dicho círculo. Se compone pues de 8 porciones de curva y 6 de recta (conteniendo ambos bordillos de la pista) que con otras 4 más pequeñas y otras invertidas, componen un total de 23 piezas de trazado. Siendo posible unir un octavo de curva con la otra del mismo cuadrante, formando una curva de giro de 90 grados; cualquier octavo de curva con una recta siendo el giro de 135 grados, pudiendo empalmar curvas simétricas de distintos cuadrantes e incluso unir un octavo de curva con su simétrico invertido.

A esto se le añaden un total de 5 piezas para conformar la recta de tribuna, parrilla de salida y línea de llegada. Otro grupo de bloques están destinados a la decoración (por decirlo de alguna manera) de los circuitos ya sean vallas publicitarias, público, tribunas, vegetación, etc., que han sido colocados posteriormente al mapeado de los trazados de cada circuito.

JAVIER CUBEDO

2. LA RUTINA MAPEADORA

Si hace tres meses me llega a decir un adivino que un día me pelearía a muerte con mis amigos por una discusión sobre el montaje de un "Scalextric", me habría estado riendo una semana. Y, así fue. Estando el programa muy avanzado, a alguien se le ocurrió la brillante idea de decirme que montar un circuitillo de la susodicha marca era un juego de niños. Mi amigo y todos los que estaban de acuerdo con él, terminaron el fin de semana en la UVI.

Este comentario pudiera parecer inadecuado y fuera de lugar, ya que muchos se estarán preguntando que tiene que ver montar un "Scalextric" con el mapeado de los circuitos. Aunque sólo sea una mera justificación por haber mandado a mis colegas al hospital voy a tratar de explicarlo.

Cada uno de los siete circuitos que componen el juego, debía de construirse a partir de 23 piezas de diversos tamaños (curvas, rectas, empalmes especiales, etc.). Estas piezas estaban compuestas a su vez por bloques gráficos de 4×16 bytes en la versión AMSTRAD. Una rutina mapeadora normal tomaría los códigos de las piezas de una tabla y a partir de ahí la expandiría vertiendo los resultados (códigos gráficos) en el mapa de bloques. La forma más normal consistiría en asignar tres bytes para cada pieza dentro de la tabla de la pista, ya que se necesitan dos bytes para cada pieza dentro de la tabla de la pista, ya que se necesitan dos bytes para definir la posición dentro del mapa de bloques, sin contar con que haría falta otro más para las piezas con repetición. Sin embargo, en un juego como ASPAR que pretende mantener simultáneamente en memoria los datos de siete circuitos diferentes no es la solución más idónea. El reto consistía en asignar sólo un byte por pieza, indicando únicamente las coordenadas de la pieza de salida. A continuación vendría el código de la pieza que enganchará con la anterior y así sucesivamente hasta completar el circuito.

Intuitivamente la idea se asemeja mucho al montaje práctico de un Scalextric, donde colocaríamos secuencialmente una detrás de otra todas las piezas que lo componen y cuya situación vendría siempre determinada por la posición de la primera pieza.

Traducir esta sencillísima idea a lenguaje ensamblador me costó exactamente 14 tazas de café, 2 paquetes de cigarrillos y un Domingo sin salir. Como se podrá comprender, no pretendo explicar aquí como lo hice, y sólo diré que la clave está en asignar a cada pieza dos puntos de enganche diferentes, que determinarán coordenadas relativas para la pieza siguiente, dependiendo de la dirección actual del recorrido.

Esto fue lo más fácil, pero durante el desarrollo del juego hubo que hacer innumerables añadidos, entre los cuales estaba el algoritmo de direccionalidad para los códigos gráficos del interior de la carretera, que permitiría a las motos seguir con fluidez su curso sin salirse de ella.

Una vez que la rutina mapeadora ya funcionaba perfectamente, el siguiente paso consistía en codificar los 7 circuitos que Javier diseñó basándose en los reales. Por lo tanto, paralelamente se tuvo que crear un potente editor de circuitos que permitiese construir fácilmente cualquier tipo de pistas en un santiamén, con sólo pulsar unas cuantas teclas, siguiendo las instrucciones de los menús.

ORLANDO ARAUJO

3. SEGUIR LA PISTA

Ya tenemos los diseños de los circuitos y las motos dispuestas a comenzar la carrera. Se realiza la primera prueba y todas las motos salen disparadas en línea recta... se salen en la primera curva y van a chocarse contra el borde de la pista.

Mucho más difícil que programar el movimiento de Aspar, nuestro protagonista, es gestionar el movimiento de las otras motos, o cómo detectar la proximidad de una curva y cómo realizarla.

La solución ya no podíamos encontrarla los Domingos por la mañana viendo las carreras. Pensamos varias soluciones:

Una serie de puntos claves hacia dónde se encaminarían todas las motos y, una vez allí se dirigían al siguiente. Hubo que rechazarlo porque, por una parte eran demasiados puntos clave, y por otra, todas las motos pasarían por el mismo sitio.

Una segunda posible solución era seguir el mapa de bloques tipo Scalextric, pero éste estaba codificado y, para poder seguirlo habría que expandirlo, para lo que no había memoria suficiente.

Lo que sí estaba expandido era el mapa de bloques pequeños (caracteres de dos por dos) y la solución definitiva vino por ahí. Los bloques tipo Scalextric se componen a su vez de otros bloques más pequeños, de los cuales los lisos marcan el centro de la pista. Las motos, por lo tanto tendrían que tender a ir hacia los bloques lisos. Si delante de la moto no hubiese un bloque de pista es que habría de torcer hacia derecha o izquierda en la dirección en que sí los hubiese. Ahora aparecía el problema de que la derecha unas veces es arriba, otras abajo e incluso a la izquierda, por lo que más de una vez las motos podrían dar la vuelta tranquilamente y seguir en dirección contraria. Había que darle a los bloques una direccionalidad. Hubo que mapear de nuevo los circuitos desdoblado los caracteres del centro de la pista

en ocho caracteres iguales pero con códigos diferentes de forma que la moto no sólo supiese que va por el centro de la pista, sino cual es la dirección que debe seguir.

Los problemas no acabaron aquí. Las motos se salían en las curvas muy cerradas por exceso de velocidad lo que hizo necesario una sofisticada rutina para reducir y acelerar, y por otra parte un algoritmo para retomar la carretera en caso de que a pesar de todo se saliesen.

PEDRO SUDON

4. LOS ADELANTAMIENTOS

El problema de los adelantamientos había sido dejado para ser resuelto después del de la conducción. No me había parado a pensar en ello, pero mi suposición es que las motos tomarían posiciones en la primera curva y se mantendrían en ese orden hasta el final de la carrera.

Fue enorme mi sorpresa cuando al poner de nuevo el programa en modo "demo" las motos se adelantaban unas a otras como si de una carrera de verdad se tratase. Todo el equipo vino corriendo a ver una espectacular carrera en la que Aspar salía en cuarto lugar, fue adelantado en la segunda curva llegando a quedar el quinto. A partir de ese momento fue tomando posiciones de forma que al final de la segunda vuelta estaba el segundo y en la tercera tomó la primera posición.

Fue algo alucinante todo el mundo contempla en el ordenador una carrera más emocionante que las de la televisión y me preguntaban que cómo lo había hecho, que cuál era el algoritmo de adelantamientos y cómo había programado especialmente a Aspar para que ganase la carrera.

Yo, con los ojos como platos no podía dar crédito a lo que veía. Muy amenudo he dicho que programar vídeo-juegos es un poco ser como Dios: Tú los creas y ellos se mueven solos y te sorprenden, pero nunca me había sorprendido como ahora.

No había programado los adelantamientos y Aspar era sencillamente una de las motos que se gestionaba por las mismas rutinas que todos los demás.

Tuve que ver dos o tres veces más una "demo" para convencerme de que ciertamente las motos adelantaban y que Aspar, aunque no ganase siempre terminaba la carrera de los primeros.

¿Cómo explicar esto?: El algoritmo que hacía frenar en las curvas o retomaba la carretera si se salían las motos de ella, hacía que al salir de la curva cada moto tuviese una velocidad diferente que les permitía adelantar a las que mejor habían tomado la curva anterior.

De todas formas las motos se salían aún a menudo y, lo que es más grave, al adelantar, unas motos se superponían a otras como si pasasen por encima. Una vez terminada la rutina de choques volverían a aparecer los problemas de salidas de la carretera y de adelantamientos.

Al programa definitivo todavía le faltaba mucho a pesar de los milagros.

PEDRO SUDON

5. LOS CHOQUES

A estas alturas ya podía verse una carrera en modo "demo" en que las motos se adelantaban pasando por encima de las otras y en caso de salirse de la carretera atravesaban anuncios, vallas publicitarias y público en general.

Fue fácil hacer una rutina de detección de choques con la que el problema fue resuelto drásticamente: al final de la segunda vuelta habían perecido los ocho motoristas, cuatro de ellos en lo alto de un árbol, treinta o cuarenta espectadores y algunos jueces.

El problema difícil no era pues elaborar una rutina de choques, sino una rutina de no-choques.

Un primer intento consistía en que cada moto estudiaba como si del ajedrez se tratase, cada uno de los posibles movimientos y sus consecuencias. El resultado era magnífico: no se chocaba ni una moto y los adelantamientos se hacían a la perfección, pero tenía un pequeño fallo, como en el ajedrez, cada moto tardaba mucho tiempo en decidir hacia dónde moverse y el resultado era una carrera en cámara lenta.

Para conseguir un efecto de tiempo real hubo que sacrificar la perfección en los adelantamientos y evitación de choques, simplificando al máximo las rutinas mediante algoritmos, que por supuesto no publicaremos.

PEDRO SUDON

6. EL REGATEO

Sin duda el problema que más tiempo y disgustos nos llevó fue el decidir que es lo que quitábamos a medida que adaptamos cosas nuevas.

La memoria enseguida se nos quedó pequeña y empezamos a arremeter contra un anuncio de Marlboro que era el más bonito, pero que ocupaba un precioso medio K. Tras diversas luchas con Javier, que como todo dibujante es el que siempre sale perdiendo en todo esto, reducimos un grupo de espectadores, varios anuncios, le quitamos unos preciosos brillitos al marcador y no sé cuantas cosas más. Por nuestra parte los programadores dejamos de lado varias maravillosas rutinas que esperamos poder utilizar en el próximo proyecto.

PEDRO SUDON