

COLORJETZ



Eva de la Muela Marín, Marco Abellán Tamara, Ignacio Cano Palacios

GRUMPYWAVE @GrumpyWave

ÍNDICE.

DESARROLLO

PROBLEMAS

ANÉCDOTAS

LECCIONES APRENDIDAS

DESARROLLO

Primero empezamos por las físicas ya que lo primero que queríamos era tener un giro amplio para las curvas. Al mismo tiempo el jet empezaba a tener forma ya que se iban añadiendo datos que necesitaban.

Lo siguiente que se implementó fue la orientación del jett, para eso, ya se crearon todos los sprites necesarios, ya que decidimos que habrían 12 direcciones. Todo esto vino junto al tileset para el mapa.

A continuación, hicimos un mapa para empezar las pruebas. Justo después empezaron las pruebas de las colisiones y poco después la IA. Del mismo modo durante este tiempo se perfeccionó el sistema de físicas y se añadió la aceleración y el frenado, para que la salida sea progresiva hasta alcanzar la velocidad máxima o la mínima.

Luego nos dedicamos a perfeccionar el juego, pulir IA, añadir un cronómetro, poner un hud con información sobre las vueltas o las pistas ganadas, un pitido de salida, etc.

Lo último que hicimos fue resolver problemas de última hora.

En el proyecto usamos una implementación de Entidad-Componente-Sistema ya que partimos de la base que hicimos en el Starfield por lo que hay 3 bloques diferenciados:

Manager

Jet.s

El Manager de entidades dónde se encuentra el jett que tienen todas las propiedades del mismo y maneja algunas ya que, por ejemplo, la velocidad y derivados es una propiedad que deberá usar el sistema de físicas, al igual que la posición, que irá cambiando con la velocidad que lleve.

Race.s

Otra cosa que se encuentra es el race, fue un manager que tuvimos que implementar para cosas de la carrera que no encajaban en otro sitio como puede ser el mapa en el que te encuentras, cuántas vueltas lleva, el tiempo transcurrido, etc. Una de las cosas a destacar es que el manejador de interrupciones se encuentra en este otro manager ya que sólo íbamos a hacer un cronómetro y requerían de la ayuda de las interrupciones. Este manager de race también es el encargado de vigilar que no se hagan trampas ya que usamos la línea de la meta para sumar 1 a las vueltas que da.

IA.s

Uno de los archivos más importantes es el contiene toda la información de la IA. La IA al tener que simular el comportamiento humano, usa los mismos métodos, que utiliza el jet pero con ligeras variaciones, esto nos supuso un gran problema de repetición de código. Para la implementación de la IA usamos mucho los métodos del sistema de colisiones.

Del mismo modo que el comportamiento humano a veces es imprevisible, la IA también tiene un componente randomizado.

Sistema

Collision.s

Es el sistema encargado de encontrar la posición del mapa con el tileset y comprobar si choca con los laterales. Del mismo modo, es el encargado de detectar si se ha pasado por la meta y hacer las llamadas necesarias a los distintos sistemas o managers correspondientes para iniciar la respuesta que sea conveniente.

Physics.s

Utiliza muchos datos tanto del jet como de la Inteligencia Artificial y es responsable de todo el movimiento de los dos.

Estos son algunos de los archivos más importantes del juego, pero se incluyen muchos más como la música, el input o el gestor de mapas.

Imágenes

Aquí es dónde se encuentran todas las imágenes que hemos usado durante todo el proceso, se han añadido imágenes en el último minuto, al igual que se han desechado muchas otras.

MANUAL DEL USUARIO

Los controles del juego son:

Q: Acelerar

O: Girar a la izquierda

P: Girar a la derecha

Para poder ganar el jugador deberá hacer 5 vueltas antes que el enemigo para poder ganar

El juego se compone de 3 mapas en total y para ganar la partida se deberá ganar en los 3 mapas.

PROBLEMAS

No ha habido día que no hayamos tenido problemas, desde cómo colocar bien un sprite en pantalla hasta cómo hacer que la música no se escuche tan grave, pasando por las físicas, colisiones, pero quien se lleva la corona es la Inteligencia Artificial.

Con las colisiones nos dimos cuenta que debíamos cambiar la manera en la que comprobamos las mismas, ya que debíamos comprobar 2 esquinas a la vez. La respuesta del sistema de físicas también fue un problema porque no sabíamos ni cómo plantearlo, al final incluso tiene un componente aleatorio para darle dinamismo.

Con la Inteligencia Artificial tuvimos que cambiar los mapas para que los carriles fueran de la misma anchura. Nos rendimos varias veces ya que no éramos capaces de hacer que girara más de una vez.

De los problemas que nos encontramos casi al final del todo fue la gran cantidad de memoria que usábamos ya que eran unos 9000 bytes. Por suerte lo logramos resolver de manera bastante sencilla; comprimiendo las imágenes que usábamos y desechando código repetido e imágenes que no se usan.

Para nuestra IA creímos que la mejor manera que teníamos para implementarla era que "viese por delante" por lo que le hicimos un área invisible que ocupaba todo el ancho del carril. Que fuese recto, era la parte fácil, por lo que la condición que le pusimos al principio era que cuando no detectase colisión con el área grande y comprobando la orientación en la que se encontrase giraría hacia la derecha o la izquierda. El problema vino cuando nos dimos cuenta que el giro no era perfecto, por lo que para la IA la condición terminó siendo al revés, cuando detectase colisión, giraría.

También hay que añadir que muchos de los problemas que surgieron fueron la consecuencia de arreglar problemas anteriores.

ANÉCDOTAS

Una de las ideas que se nos ocurrió a última hora fue hacer un juego de un solo jugador, sin Inteligencia Artificial ya que cuando se nos ocurrió, estábamos en un momento crítico porque no éramos capaces de hacer la IA bien.

El modo de juego sería en 30 segundos o 1 minuto contar cuántas vueltas es capaz de dar al circuito.

La primera idea que tuvimos fue hacer que los jets dejaran un rastro del color de la nave y pudieses acelerar si pasabas por encima del rastro del enemigo y si pasabas por el rastro que tú mismo dejabas, te ralentizaba. De ahí viene el nombre del juego ColorJetz.

En el último día el videojuego funcionaba de manera completamente distinta en los 3 ordenadores.

En una de las veces que el juego dejó de funcionar, hicimos debug y el código se cambiaba dependiendo del momento de la ejecución.

LECCIONES APRENDIDAS

- Hasta que no aprendimos a usar los registros bien pasó bastante tiempo, ya que tardamos un poco en comprender que los registros son un paso intermedio para guardar datos.
- Otra de las cosas que tuvimos que tener en cuenta, en relación a lo anterior, es que había que tener cuidado de no destrozarse un registro que luego sería necesario.
- Algo que debíamos tener en cuenta desde el principio fue la compresión de imágenes, al igual que el uso de la instrucción de halt ya que durante casi todo el proceso nos apoyamos en esa instrucción.
- La optimización del código es algo que debíamos haber tenido presente siempre y no algo que se haga al final.