

# Reporte de documentación: Crazy Racer!

CC3501-1 Modelación y Computación Gráfica para Ingenieros

Sebastián Olmos Hernández, 20 de junio de 2020

La ambientación / temática de la escena es sobre una pista de carreras que se encuentra volando en el cielo, por lo que se agregaron las decoración pertinentes a esta elección. Para ordenar de buena manera como se dibujan los objetos, el programa cuenta con 3 grafos de escenas como se ve en la figura 1 (y uno extra para el auto - bot) para dibujar los distintos tipos de objetos (normal, solo con texturas, con texturas e iluminación) con sus pipelines respectivos, para el grafos con elementos de iluminación se crearon nuevas clases y métodos para que cada objeto tenga parámetros de su material ( $ka$ ,  $kd$ ,  $ks$ ) y se puedan dibujar teniendo en cuenta esto.

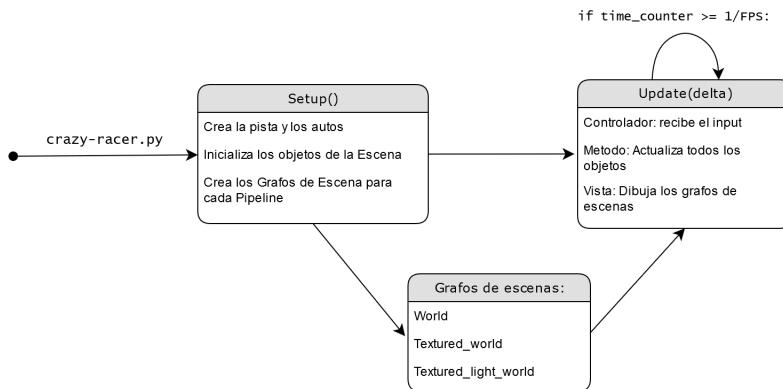


Figura 1: Diagrama de flujo simple.

La pista se creó utilizando una curva con el método otorgado para fabricar una Rounded Non Uniform Spline, implementando las funciones *GetPosition(time)* y *GetData(time)* con un conjunto de nodos, a los cuales también se implementaron clases y funciones para otorgarles los parámetros requeridos para la RNU como velocidad/tangente y relaciones de vecindad. Además se agrega un método para que cada nodo tenga un vector binormal. Implementando una interpolación lineal, se logró que la función *GetData(time)* entregue además de la posición, los vectores binormal, tangente (extraído del documento sobre las RNU) y normal (producto cruz de los anteriores) y así logra formar un triedro de frenet. Con esto último se construye la pista dividiendo la curva en puntos y por cada punto agregar dos vértices en direcciones opuestas gracias al vector normal. También se agregaron caras inferiores y a los lados, además de unas turbinas acopladas a la pista.

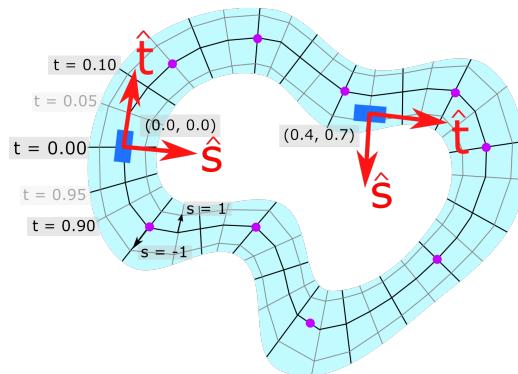


Figura 2: Ejemplo de Parametrización de una curva definida por los nodos morados.

El modelo de los autos se hizo con el programa Blender, del cual solo se obtuvieron los vértices y las normales de las caras, luego el programa en python crea las caras con grafos de escena y les asigna la textura correspondientes, también se agregaron ruedas que giran según la velocidad del auto.

Para el movimiento del auto, la pista se parametrizo, como se observa en la Figura 2, con las coordenadas  $t \in [0, 1]$  y  $s \in [-\frac{3}{2}, \frac{3}{2}]$  que representan el tiempo en la curva y la distancia al centro de la curva en la dirección del vector normal, respectivamente. Con esta parametrización y de la misma manera vista en la construcción de la pista se puede formar un triángulo de frenet anclado al vehículo del usuario, que gira con el como se ve en la Figura 3, gracias a esto se puede crear un vector que apunta en la dirección donde se mueve el vehículo, dando un candidato a nueva posición para que el vehículo se mueva. Para que se mantenga anclado a la pista, se proyecta este vector en la pista (función `nearest_track_coord()` en `track.py`) retornando coordenadas de la pista parametrizada, de esta manera se mueve el vehículo. Luego para rotar el auto se rota el vector unitario en el eje x, el cual se le aplican las rotaciones de inclinación del auto, otorgando el vector que apunta en la dirección del movimiento del vehículo mencionada anteriormente.

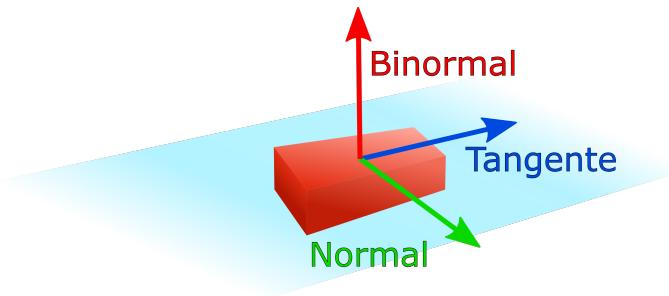


Figura 3: Ejemplo del Triángulo de frenet anclado a la representación de un auto en la pista.

Para dar más dinamismo al programa se añadió otro auto para competir con el, que simplemente avanza con velocidad constante. También se agregaron varios elementos que se mueven por todo la escena. Por último se agrego como HUD un minimapa de la pista indicando en todo momento la ubicación del auto del usuario y del bot, se agrego un indicador de la vuelta actual (máximo 9) y un indicador de la posición con respecto a los otros autos (primer o segundo lugar).

## Instrucciones de Ejecución

Ejecutar el archivo `crazy-racer.py`, el programa demora un poco en abrir al tener que cargar las texturas, luego las teclas que reciben son : *W* mueve el auto hacia adelante, *S* mueve el auto hacia atrás, *A* rota el auto hacia la izquierda, *D* rota el auto hacia la derecha, *V* cambia el tipo de cámara (trasera, frontal, lateral, global), las siguientes teclas son para la vista global: *UP\_ARROW* rota la cámara hacia arriba, *DOWN\_ARROW* rota la cámara hacia abajo, *LEFT\_ARROW* rota la cámara hacia la izquierda, *RIGHT\_ARROW* rota la cámara hacia la derecha, *Z* aumenta el zoom de la cámara, *X* disminuye el zoom de la cámara, *SPACE* cambia el modo de relleno y *ESC* finaliza el programa. El auto no puede salirse ya que al llegar al borde se detiene y no avanza.

## Resultados

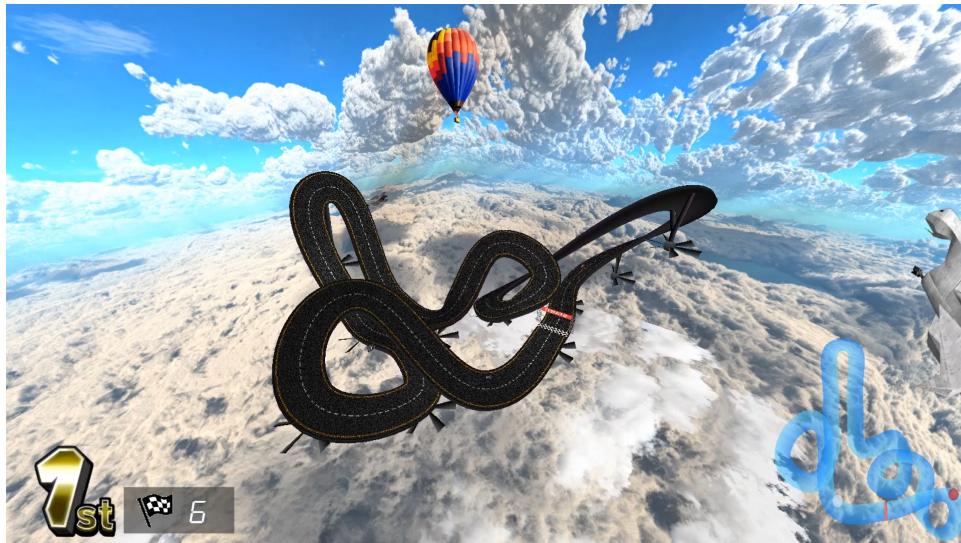


Figura 4: Vista global de la escena / pista.

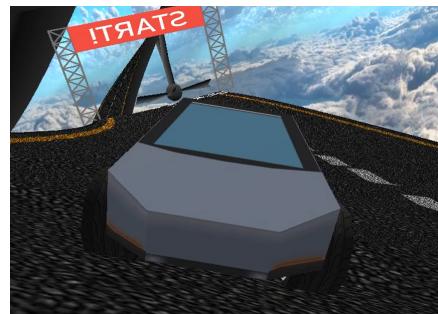


Figura 5: Vista frontal del auto.



Figura 6: Vista lateral del auto.

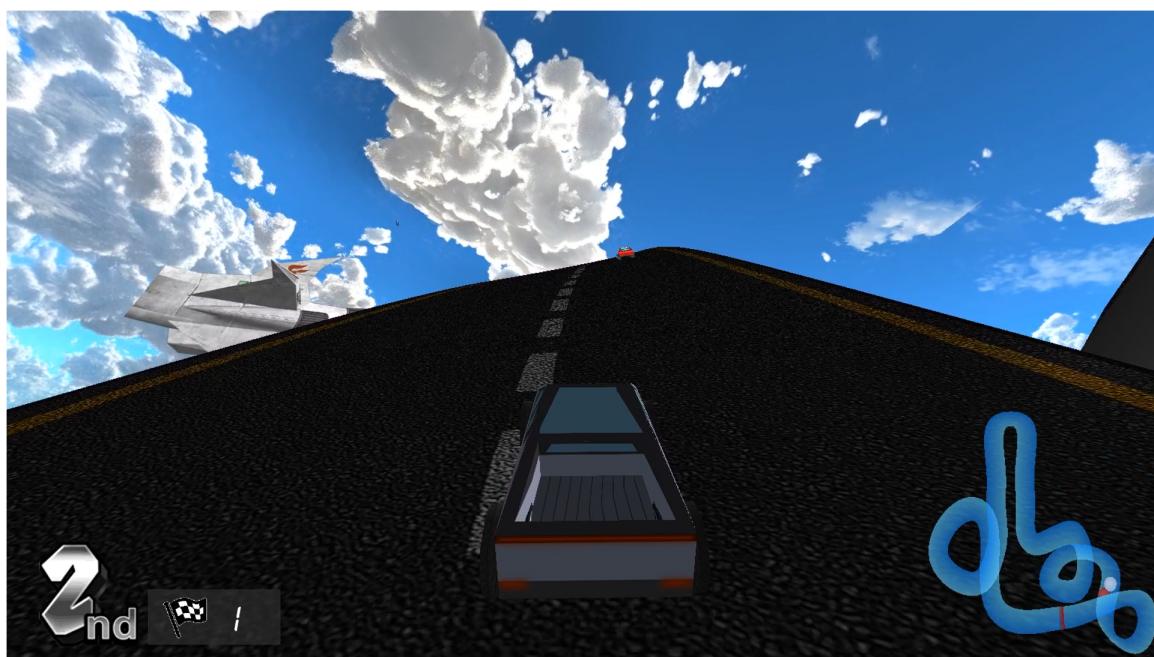


Figura 7: Parte de la pista orientada opuestamente a la fuente de luz.

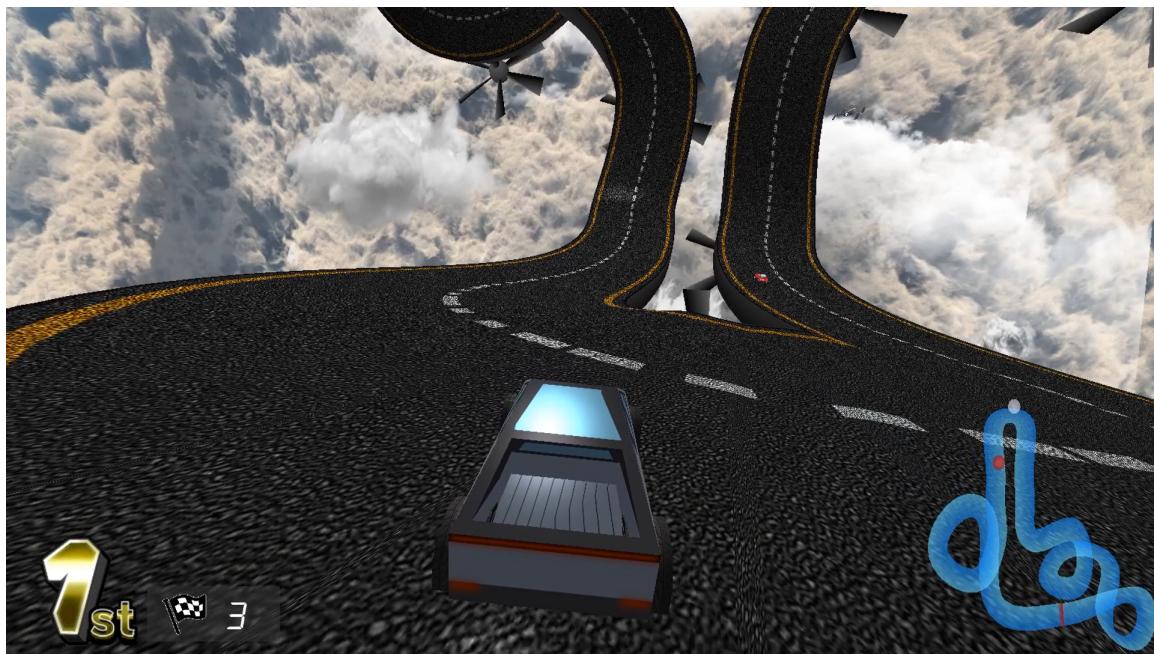


Figura 8: Parte de la pista orientada a la fuente de luz, reflejos en el auto.

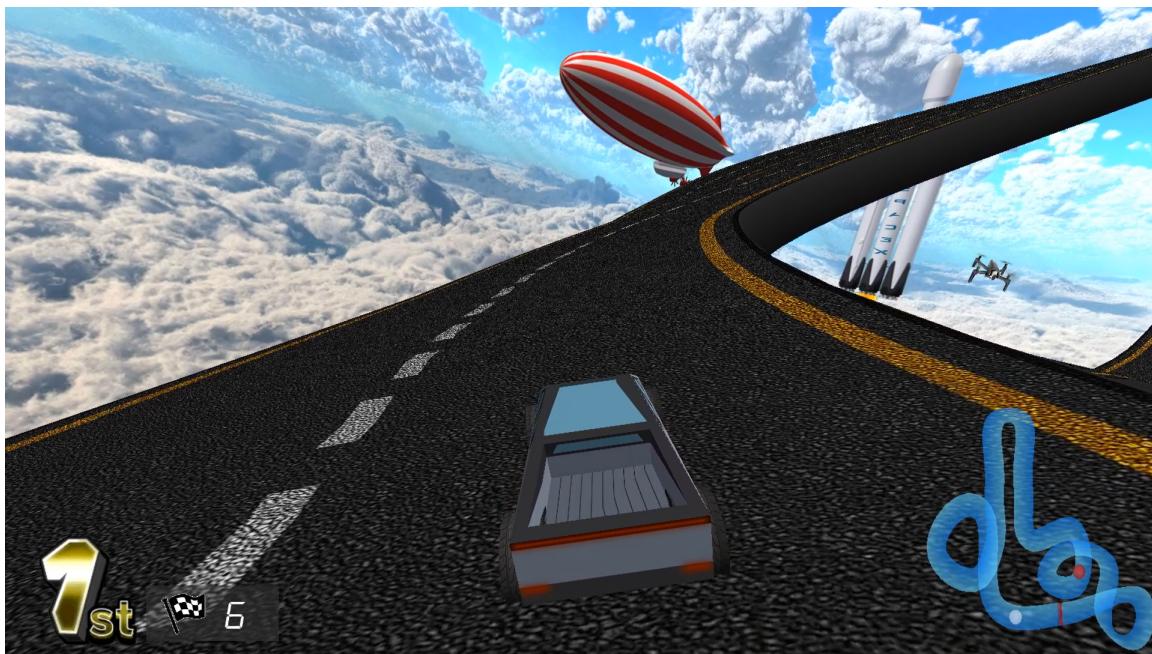


Figura 9: Contador de vueltas actualizado (esquina inferior izquierda) tras varias vueltas.

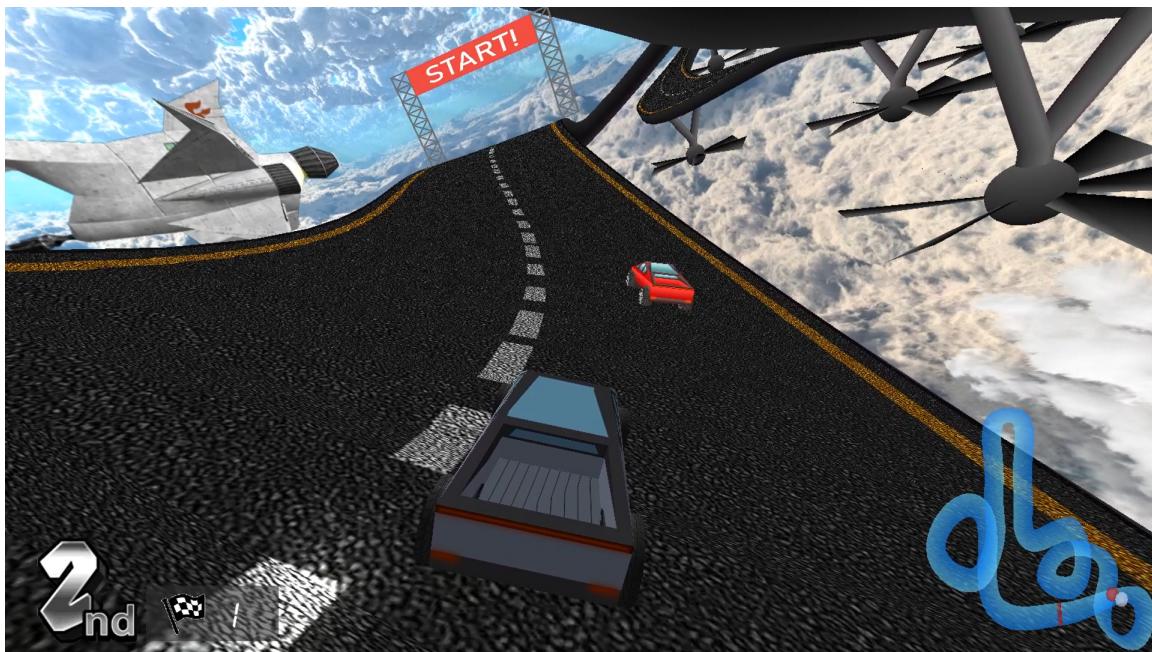


Figura 10: Segunda posición al no superar al bot (auto rojo).