
# Reporte Tarea 2b

Pablo Muñoz Soto

10/06/2020

CC3501

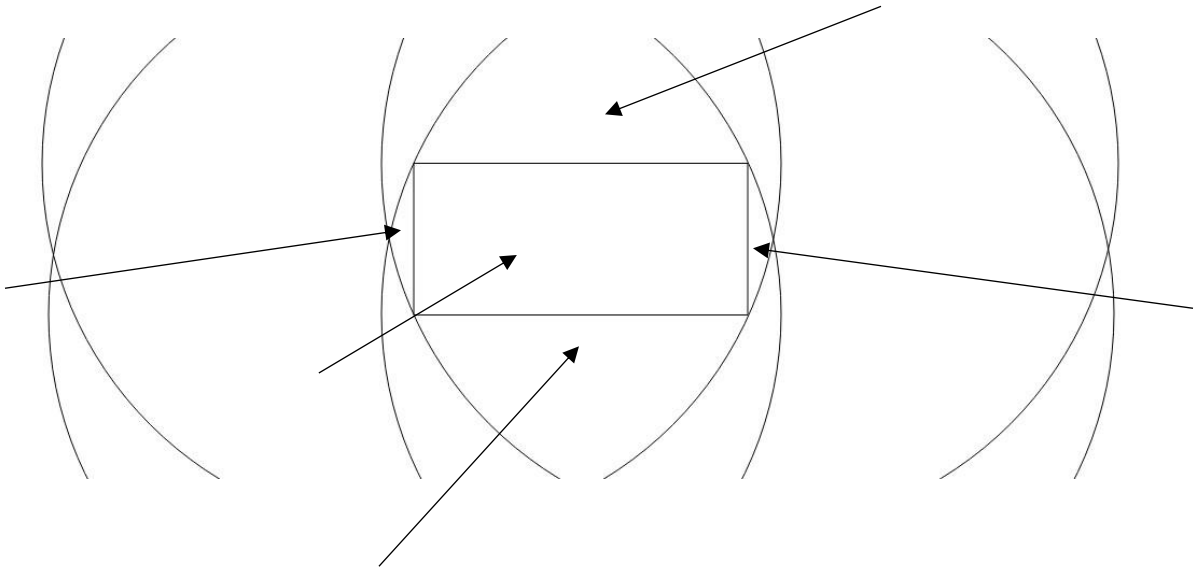
## Solución propuesta:

Los elementos gráficos del programa son el auto, el cielo de fondo, la pista, el sol, el contador de vueltas y el avión, además de utilizar el método de Gouraud para simular la luz, para crear la curva se usó una rounded non-uniform spline, para esto se calcula la velocidad en cada punto como la recta perpendicular a la bisectriz del ángulo formado por el punto anterior y posterior, dado que considerar la variable z en este cálculo lo vuelve muy complicado se optó por aproximar la coordenada z de la velocidad como el promedio del punto anterior y posterior, luego este vector es normalizado y tenemos la velocidad en el punto.

Como se quiere delimitar la pista por 2 curvas, se tiene que generar otra, esto se hace a partir de la primera, calculando el vector perpendicular a la dirección en cada punto, como hay 2 posibles vectores perpendiculares nos quedamos con el que al realizar el producto cruz entre la dirección y este, se obtenga un vector con la coordenada $z>0$, para simplificar los cálculos se asumió que las secciones transversales de la pista son paralelas al piso . Al obtener el vector perpendicular este se suma a la posición del punto para obtener el punto de la 2da curva, notar que la lista que contiene los puntos de la primera curva y la lista que contiene los puntos de la segunda curva son de igual longitud, debido a esto es posible generar la malla usando los mismos puntos de la curva 1 y 2, para esto generamos rectángulos tal que los puntos izquierdos son de la curva 1 y los derechos de la curva 2, como ambas curvas tienen la misma longitud con esto se logra cubrir toda la pista

Para lograr el movimiento vertical del auto fue necesario implementar un sistema para saber en que parte de la pista esta el auto, para esto se utilizo el criterio de que el auto está en el rectángulo n si el auto está en la intersección de las circunferencias de cada punto de el rectángulo n, tal que el radio de estas circunferencias es el largo de la diagonal, este método además usa la posición anterior del auto para no tener que verificar en todos los puntos de la curva, si no solo en un rango de 100 puntos desde su posición anterior. Con la posición del auto solo basta trasladarlo en el eje z tanto como el punto de la curva que esta en esa misma posición. Notar que este método de posicionamiento solo funciona si el auto no sale de la pista en ningún momento, en el siguiente diagrama se muestra un esquema de el sistema descrito, las flechas indican las zonas donde el auto será detectado como si estuviera en la posición n. A pesar de que hay un exceso en la

zona de detección, la posición solo tiene que ser cercana, no exacta a la correcta, por lo que este sistema sirve para lo requerido.



## Las texturas fueron sacadas de:

auto= https://opengameart.org/content/simple-car-textures

pista=https://pixabay.com/images/search/asphalt%20texture/

cielo=https://es.brusheezy.com/texturas/43311-textura-del-cielo-azul

sol=http://misdibujos.de/dibujos-del-sol/

avion=https://es.wikipedia.org/wiki/Avi%C3%B3n

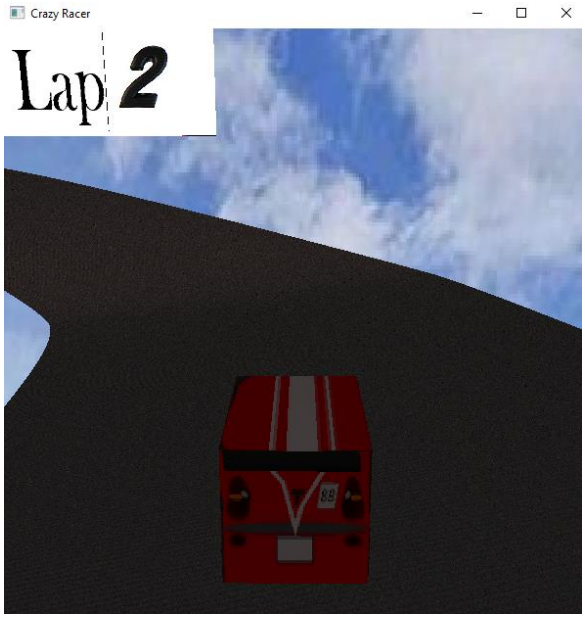
numeros=https://www.cgstudio.com/texture/black-semi-glossy-numbers-1-10-778499

Instrucciones de ejecución:

Para utilizarlo se necesitan los archivos basic_shapes, easy_shaders, ex_curves, lighting_shaders, local_shapes, scene_graph y transformations, ademas de los modulos numpy, glfw, pillow y pyOpenGl

Para ejecutarlo se usa el cmd en la carpeta con los archivos necesarios llamando a Python crazy-racer.py, luego de esto se usan las flechas direccionales para girar, avanzar y retroceder

Screenshots:


Pista vista desde arriba