



**dcc**

CIENCIAS DE LA COMPUTACIÓN  
UNIVERSIDAD DE CHILE

CC3501-1 Modelación y Computación Gráfica para Ingenieros

---

# Tarea 1

*ROLLER COASTER*

**Alumno:** Cristóbal Torres Gutiérrez.

**Profesor:** Daniel Calderón.

**Auxiliares:** Nelson Marambio, Alonso Utreras.

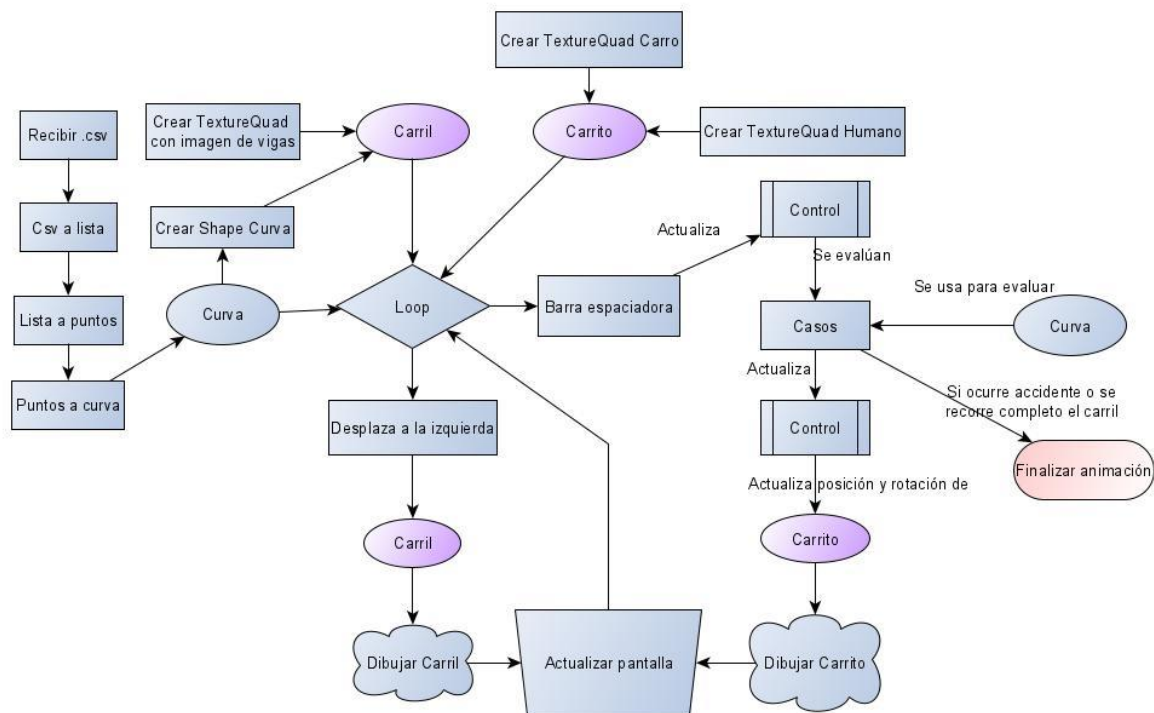
**Ayudantes:** Tomás Calderón, Nadia Decar, Beatriz Grabolosa, Heinich Porro.

**Curso:** CC3501-1 Modelación y Computación Gráfica para Ingenieros.

**Fecha:** 01/05/2020.

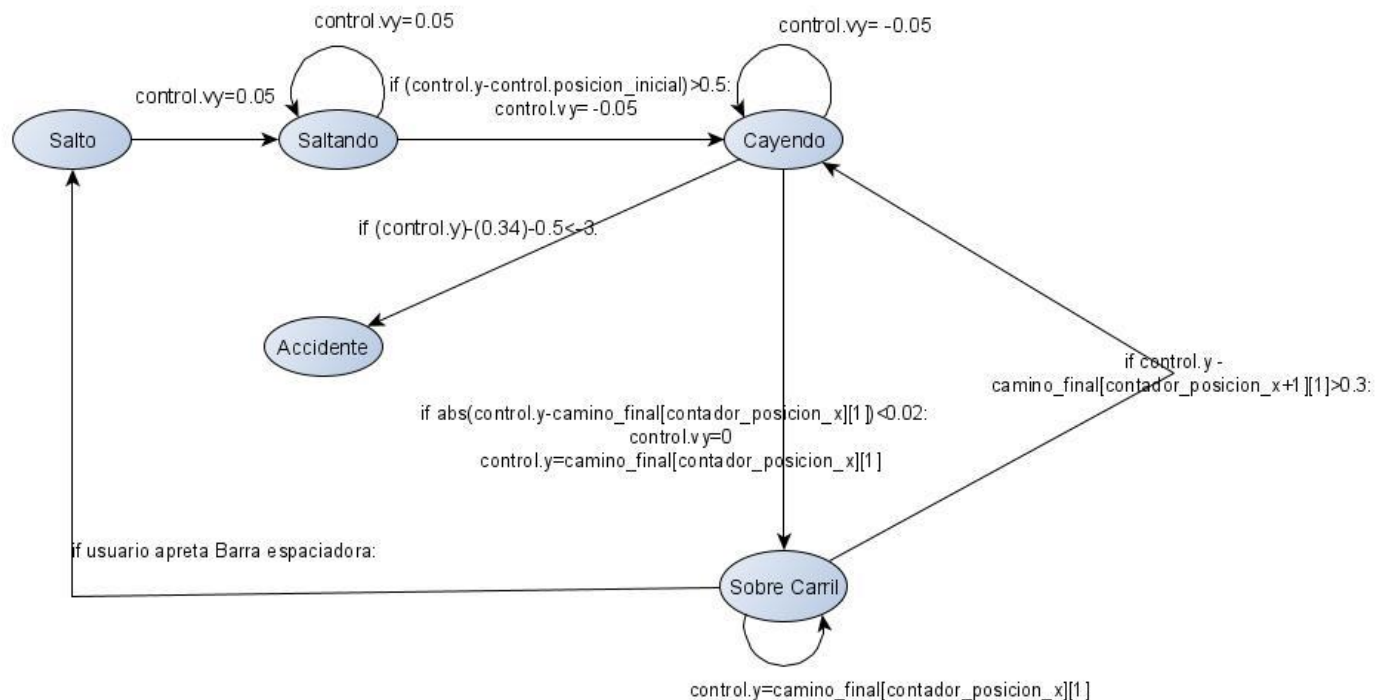
### Solución Propuesta:

**Diagrama General:**



### Casos y Control:

Se crea una clase Controller, la cual almacena diversas variables (posición X, posición Y, velocidad Y, derivada) y estados (salto, saltando, cayendo, accidente). Las variables se ven alteradas según el estado en el que se está y los puntos de la montaña rusa, los cambios de estado se llevan a cabo en el sector “Casos” del diagrama general. La descripción más profunda de estos casos se rige por el siguiente diagrama de estados:



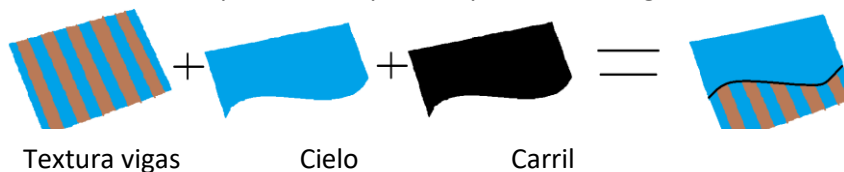
## Armando curvas

La solución parte por definir una forma de recibir los datos de entrada. Para esto se genera un módulo con múltiples funciones, que en primera instancia transforman el archivo .csv a una lista de coordenadas, se calcula la coordenada Y máxima de la lista y se guarda como una variable (máximo), este valor dividirá después los puntos para que el tamaño de la pista sea acorde a la ventana. A la lista se le añade un punto al inicio y uno al final, luego se separan los puntos en los que serán curvas y los que serán agujeros (los que tienen una “x” en el archivo).

A estos puntos separados se les agrega una coordenada z (de valor 0) y se convierten en arreglos de numpy. Donde deberá haber curvas se crean de tipo Spline de Catmul-Rom y donde hay agujeros se crean rectas a una altura muy inferior (coordenada Y con valores de -10). Por último, todos estos puntos concatenan y se dividen por el máximo calculado anteriormente.

## Dibujado

Para crear el dibujado del camino, arriba de cada punto obtenido anteriormente se crean puntos nuevos, a una altura superior (valores de 2), se une cada uno de estos con los creados en la etapa anterior y se les aplica un color representando el cielo. Un poco debajo de esta se hace lo mismo, pero con color negro. Detrás de ambas se agregan 5 cuadrados con una textura de vigas. Todo esto se moverá haciendo parecer que el carrito se desplaza en su eje X, sin embargo, los 5 cuadrados con textura de vigas reiniciarán su posición constantemente, pareciendo que siempre se crean vigas nuevas.



Para el carrito se dibujan las texturas, luego se arman los nodos incluyendo estas texturas. El carrito avanza por cada punto del arreglo de puntos anterior, sin embargo, al dibujarse, la posición “y” es la del control, pero la posición “x” siempre es 0, pareciendo que avanza, aunque solo se mueve en un eje.

## Inclinación/Rotación

Se usa el concepto de derivada, calculando la variación entre el punto actual y el siguiente de la montaña rusa, luego se calcula su arco tangente y este resultado representa el ángulo que debe tener el carrito, entonces si el ángulo es mayor al actual, se le suma un valor muy pequeño (del orden de  $10^{-9}$ ), en cada fotograma, sin que supere el ángulo obtenido antes, en caso de ser menor se hace análogo pero restando ese número.

## Instrucciones de ejecución:

Ejecución del programa: Escribir en la terminal: `python roller-coaster.py track.csv`  
Asegurándose estar en el directorio correcto y haber creado un archivo track.csv.

Consideraciones y Supuestos: Usar números enteros positivos para coordenadas, el juego ajusta su uso en la pantalla acorde a la mayor coordenada “Y” que recibe, prefiera números menores a 40.

Uso: Si desea saltar debe apretar la barra espaciadora de su teclado.



## ScreenShots:

