



Universidad de Chile

Facultad de Ciencias Físicas y Matemáticas

Departamento de Ciencias de la Computación

CC3501-1 Modelación y Computación Gráfica para Ingenieros

Tarea 2c

Migración de las aves

Alumna: Julia Paredes

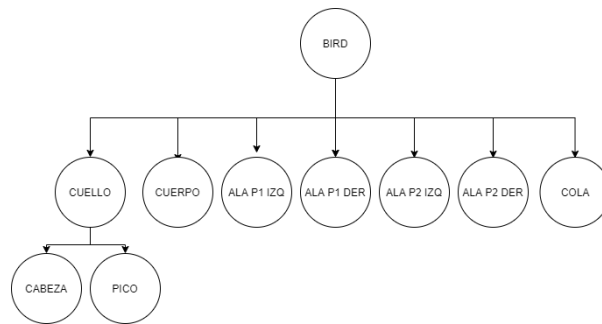
RUT: 20.240.283-6

Profesor: Daniel Calderón

Auxiliares: Alonso Utreras - Nelson Marambio

Fecha de entrega: 13 DE MAYO, 2020

Lo primero a programar fue la el pajarito...



Explicando de manera más detallada:

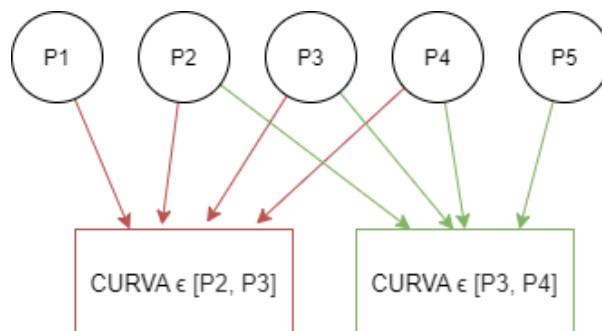
El pajarito está hecho de cubos, los cuales se hicieron con la función `createColorNormalsCube(r,g,b)`, pues se necesita de cubos con normales para implementar iluminación. Hay un cubo para cada parte del pajarito y cada uno fue escalado y trasladado de la manera correspondiente. Estos cubos se unieron mediante grafos de escena. La función encargada de realizar esto se llama `createbird()`.

Para el aleteo del pajarito se creó la función `aleteo(x, birdnode)`, la cual recibe un `x` cambiante, que en el primer programa (`bird`) viene siendo la coordenada `y` del mouse (dentro de la ventana), y también recibe el pajarito creado por la función `createbird()`.

Esta función `aleteo`, mediante ecuaciones rota algunas partes del pájaro, simulando un movimiento de cuello, de "hombros", de codos y de cola. En general es un movimiento análogo cuando el mouse sube y baja, excepto para las alas; las cuales cuando el mouse baja estas bajan de manera recta, solo con una rotación del "hombro", no del codo.^a diferencia de cuando suben que hacen una rotación de ambos.

Como shader program se utilizan dos, `es.SimpleModelViewProjectionShaderProgram()` para dibujar los ejes axiales y `ls.SimplePhongShaderProgram()` para dibujar el pájaro.

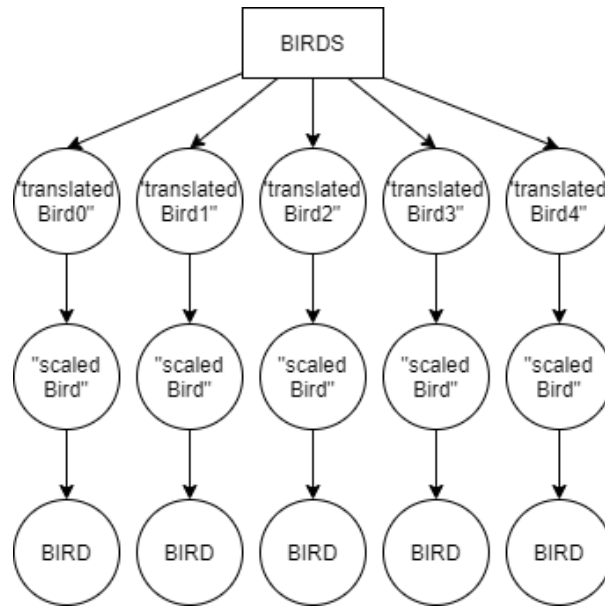
Por otra parte; en el archivo `bird - herd.py`, el cual se ejecuta con el comando: `"bird-herd.py nombre-del-archivo-de-puntos.csv"`, lo primero que hice fue leer los 5 puntos del archivo que contiene los puntos (entrego 3 archivos con distintos puntos: `"path1.csv"`, `"path2.csv"` y `"path3.csv"`) y con ellos hacer una curva Catmull-Rom, la cual en su forma base recibe solo 4 puntos, por lo cual para que se formara una curva con los 5 puntos dados:



Se realizó una curva con la función base de CR con los primeros 4 puntos, lo cual resultó en una curva entre los puntos 2 y 3, y se realizó una segunda curva con los últimos 4 puntos, por ende se formó una curva entre los puntos 3 y 4. Juntando ambas curvas se puede obtener la curva resultante de los 5 puntos iniciales. Cabe destacar que con CR los puntos primero y último no se utilizan más que para calcular la

tangente de sus puntos vecinos. Todo esto se ejecuta en la función *CR* que recibe como argumento los 5 puntos y está ubicada en el archivo *ex_curves.py*.

Para crear las 5 aves, está la función *create5aves()* la cual también esta creada con grafos de escena:



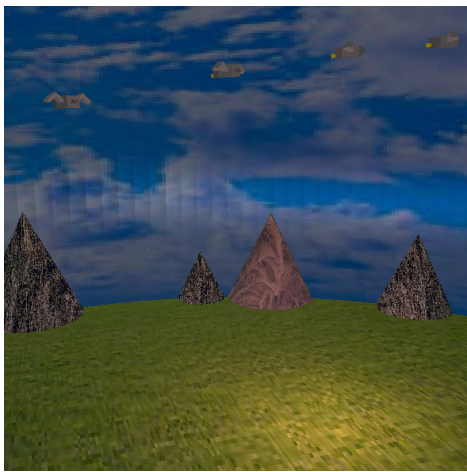
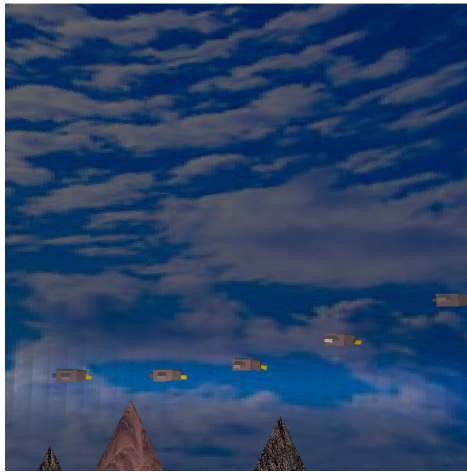
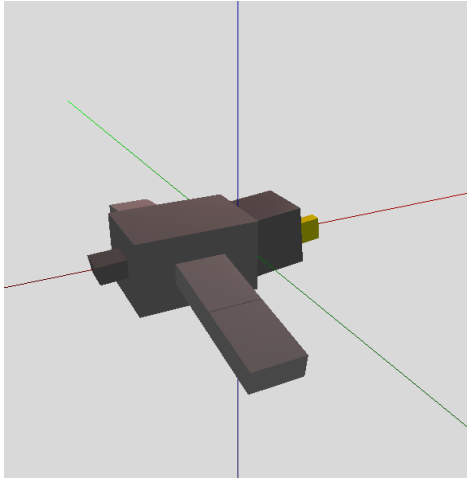
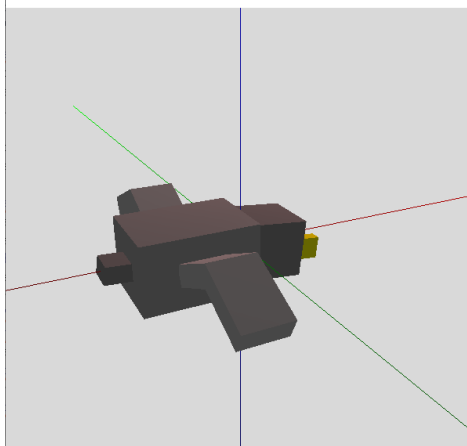
Se le aplico un escalamiento a las aves, y luego se puso cada una de las aves en un nodo llamado *translatedBirdi*, donde *i* es el índice de cada ave, lo cual servirá para indicarle la curva a cada ave por separado, para lograr un distanciamiento entre las aves.

Como paisaje, se crearon dos cerros, con la figura de un cono, con la función *generateTextureCerro* la cual está ubicada en el archivo *formas.py*, como argumentos tiene la discretización del cono, la textura, radio, z base, z top. Esta función está hecha de manera análoga al cilindro de un auxiliar. También se creó un cilindro con la textura del cielo con la función *generateTextureCylinder* ubicada en el archivo *cylinders.py*, la cual recibe, el numero de discretizaciones, la textura, radios, z base, z top. y un cubo con la textura del pasto. Por último se creó un cubo con la textura de pasto con la función *createTextureNormalsCube*, ubicada en *basic_shapes.py*, la cual no detallaré porque es una función dada más que conocida a estas alturas.

En relación a la cámara, se ubicó en una posición fija, más o menos a la mitad de altura del cilindro, el vector *at* se rota con respecto a la ubicación (para esto se ocupó coord. esféricas) y el vector *up* es $[0, 0, 1]$

Por último, el movimiento de las aves a través de la curva, se trabaja en cada nodo *translatedBirdi* comenzando en movimiento de cada ave en distintos puntos para conseguir una distancia entre ellas y se consiguió una rotación siguiendo la curva con el ángulo producido por los dos puntos vecinos al actual en las coordenadas *x* e *y*. Con esos datos se le aplicó una transformación a cada nodo *translated*. El aleteo se aplicó con la función ya creada en el archivo *bird.py*, pero ahora de argumento '*x*' tiene la coord. *y* de un movimiento sinusoidal con respecto al tiempo del programa, y de segundo argumento tiene al ave ubicada en el nodo '*scaledBird*', presente en cada nodo *translatedBirdi*. Se utilizaron dos shader program *SimplePhongShaderProgram()* para el ave, sin texturas, y *SimplePhongShaderProgram()* para el paisaje.

Los screenshots:



Estas últimas dos fotos son con *path2.csv* y *path1.csv* respectivamente.