



Tarea 2: OpenGL 3D

Reporte de Documentación tarea 2c

Curso: CC3501-1

Alumno: Sebastián Salinas

Profesor: Daniel Calderón

Auxiliares: Alonso Utreras

Nelson Marambio Q.

Ayudantes: Beatriz Grabolosa M.

Heinich Porro Sufan

Nadia Decar

Tomás Calderón R.

Fecha: 15 de Junio del 2020

1. Descripción del Problema y Arquitectura del Programa

La tarea2c consistía en simular la migración de las aves utilizando un modelo 3D articulado de estas. Este modelo sería animado posteriormente, logrando así generar la sensación de vuelo de cada una de las aves.

Para completar este objetivo se pedía hacer 2 programas, *bird.py* y *bird-herd.py*. Donde en el primero se tenía que crear el modelo 3D de una ave, agregar iluminación y generar un movimiento de aleteo en las alas si es que se movía el mouse. Y en el segundo, crear un escenario adecuado y múltiples aves que siguieran una trayectoria hecha por una spline de Catmul-Rom, la cual sería creada a través de la interpolación de unos puntos especificados en un archivo con extensión *csv*, solucionando de esta forma el problema de modelamiento de la migración de las aves.

Para implementar estos programas se ocupó como base cinco de los módulos presentados por el profesor, estos son *transformations.py*, *basic_shapes.py*, *scene_graph.py*, *easy_shaders.py* y *lighting_shaders.py*. Además se crearon tres módulos extra, los cuales manejan ciertos aspectos y lógicas que se explicarán a continuación.

El módulo *createBird.py*, el cual contiene una clase *bird()* con atributos como la posición del ave en los planos XYZ y la localización dentro de la bandada de aves. Además de por supuesto encargarse de crear el modelo 3D del ave, utilizando grafos de escena.

El módulo *reader.py*, en el cual se creó la función *csvToArray()* que recibe como parámetro el nombre de un archivo con extensión *csv* que contenga una secuencia de puntos pertenecientes a \mathbb{R}^3 y los guarda en un arreglo, para posteriormente retornarlo y que pueda ser utilizado para generar la spline de Catmull-Rom.

Por último el módulo *catmullRomSplines.py*, en el cual se creó la función *merchSplinesPoints()*, que recibe un arreglo que contenga puntos en \mathbb{R}^3 (idealmente el arreglo retornado por la función *csvToArray()*) y un número N , que será la cantidad de pasos en que se dividirá la spline. Esta función se encarga de generar splines de Catmull-Rom (con ayuda de algunas funciones presentadas por el profesor) y guarda los puntos que siguen estas curvas en una lista. Para finalmente retornar en un arreglo todos los puntos que siguen las splines. Los puntos contenidos en este arreglo serán utilizados como parámetros en las transformaciones de traslación de las aves, generando así una traslación continua por la spline de Catmull-Rom que fue hecha gracias a los puntos contenidos en el archivo *csv*.

Es así que con todos estos módulos se procedió a implementar los programas *bird.py* y *bird-herd.py*, siguiendo el patrón de diseño Modelo-Vista-Controlador. En el caso de *bird.py*, el controlador maneja la mecánica de movimiento de las alas a través de la posición del mouse. En el modelo se definen algunas constantes, se crea la ave y se definen características como su posición en la escena. Y en la vista se procede a dibujar a la ave y las animaciones para el aleteo que dependen de si el usuario efectivamente movió el mouse por la ventana durante tiempo transcurrido al ejecutar el programa.

De manera similar en el programa *bird-herd.py* el controlador se encarga de la mecánica del movimiento de la cámara a través de la posición del mouse. En el modelo se crean las *gpuShapes* del escenario y de las aves, además de definir sus posiciones dentro de la bandada y en el mundo 3D. También se analiza el archivo *csv* entregado y se crea la spline de Catmul-Rom correspondiente, para que las aves sigan esta trayectoria. Y por último en la vista se dibuja toda la escena (la bandada de pajaros y las texturas de fondo), además de las animaciones para el aleteo de toda la bandada.

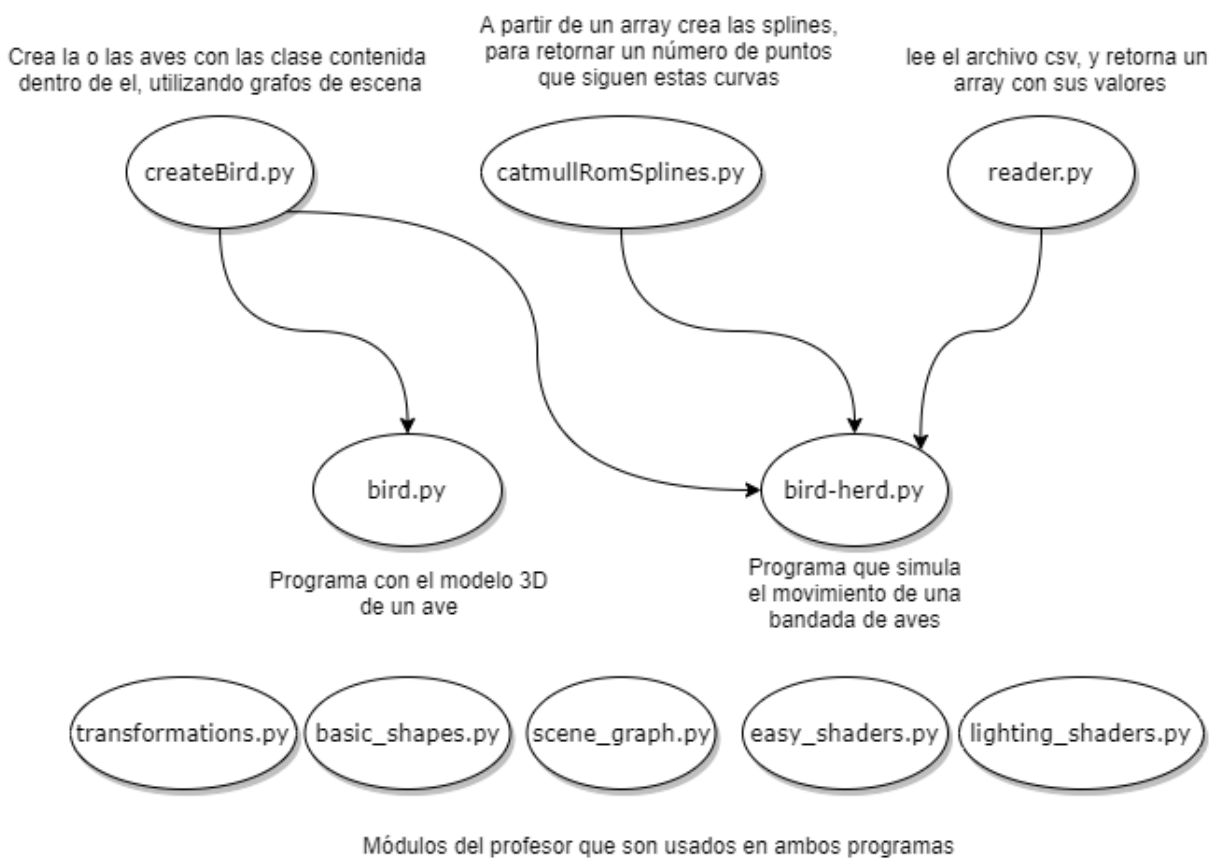
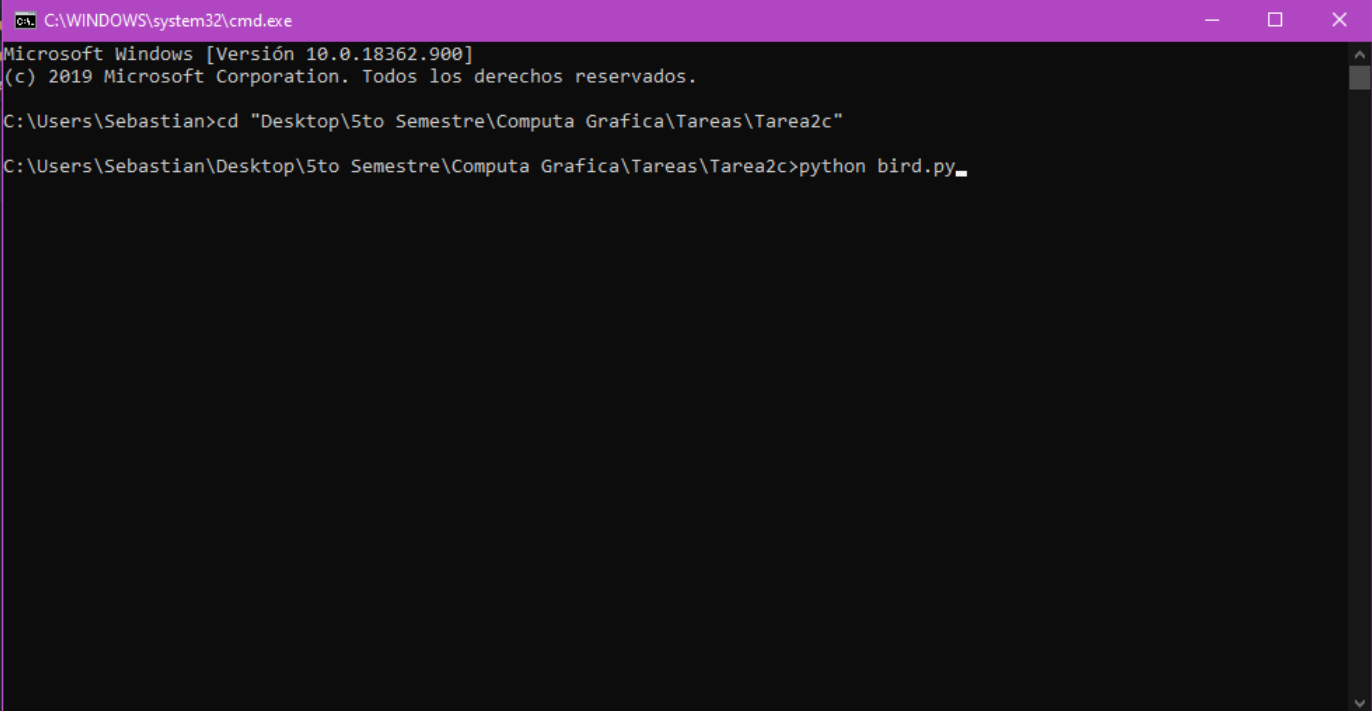


Figura 1: Diagrama con los programas y modulos utilizados.

2. Instrucciones de ejecución y Resultados

Para iniciar el programa *bird.py* no se pide ni un parametro. En el caso de que se quiera ver la animación de aleteo de la ave simplemente se tiene que desplazar el mouse de arriba a abajo dentro de la ventana de la aplicación. Además es posible rotar la cámara hacia la izquierda o hacia la derecha si es que se apreta en el teclado la flecha izquierda o derecha respectivamente.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.18362.900]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sebastian>cd "Desktop\5to Semestre\Computa Grafica\Tareas\Tarea2c"

C:\Users\Sebastian\Desktop\5to Semestre\Computa Grafica\Tareas\Tarea2c>python bird.py_
```

Figura 2: Ejecución bird.py

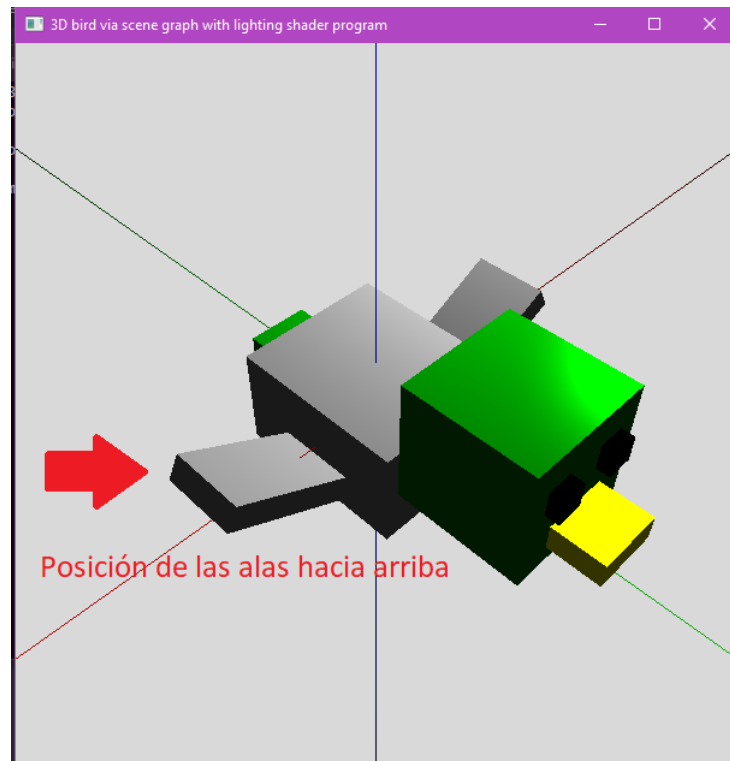


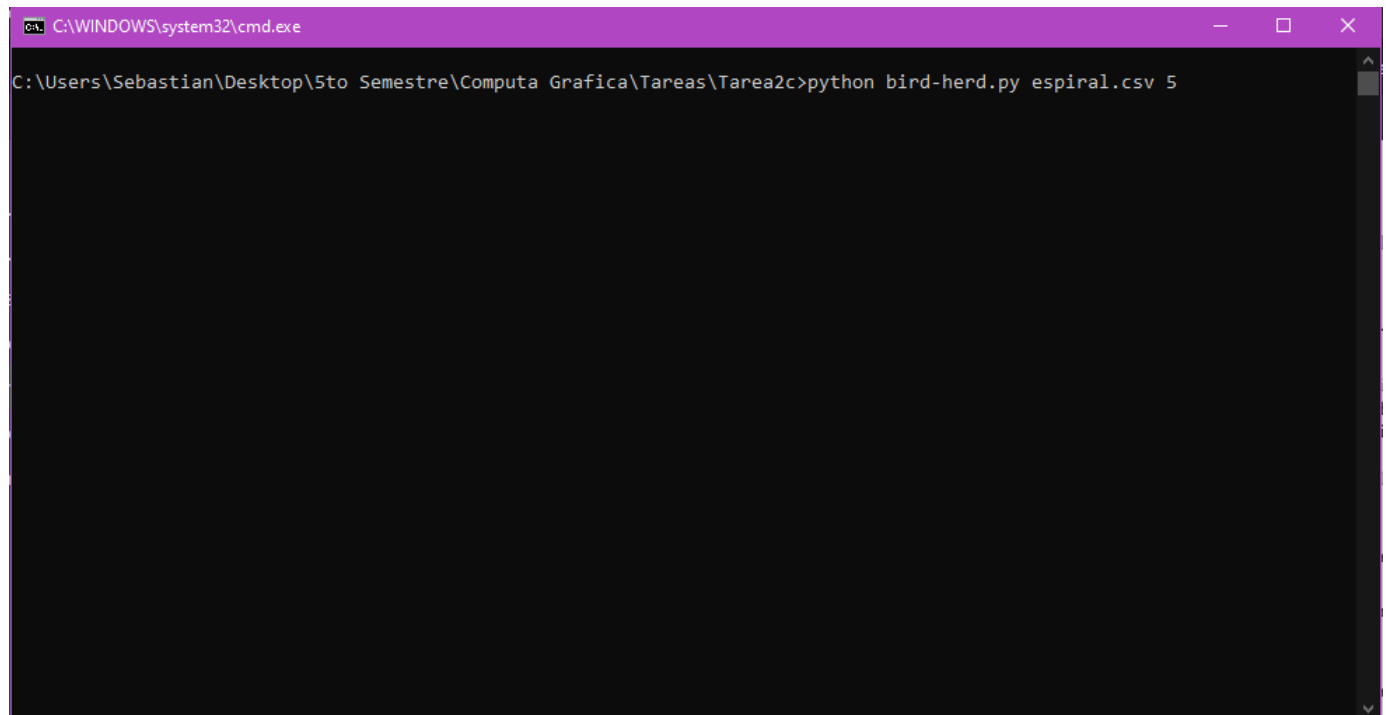
Figura 3: Escena 1 bird.py



Figura 4: Escena 2 bird.py

Por otro lado para iniciar el programa *bird-herd.py* se pide ingresar un archivo *csv* (para generar la trayectoria de movimiento de las aves) y un número (que definirá la cantidad de aves que se dibujarán). El programa comenzará con las aves en la escena y dependiendo del archivo *csv* ingresado se trasladarán por esta, siempre manteniendo su animación de aleteo.

Cabe mencionar que el movimiento de la cámara se controla con la posición del mouse dentro de la ventana de la aplicación.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Sebastian\Desktop\5to Semestre\Computa Grafica\Tareas\Tarea2c>python bird-herd.py espiral.csv 5
```

Figura 5: Ejecución bird-herd.py



Figura 6: Escena 1 bird-herd.py



Figura 7: Escena 2 bird-herd.py