

Taller No.9

Presentado a:

Ing Juan Camilo Gutierrez

Presentado por:

Mario Roberto Samudio Martinez

Artículo IEEE Simulación función seno en blender

Universidad de Cundinamarca

Sede Chía

Programa de ingeniería de sistemas

lunes 25 de mayo de 2020



UDEEC
UNIVERSIDAD DE
CUNDINAMARCA

Modelación



Facultad de
Ingeniería

SIMULACIÓN FUNCIÓN SENO EN BLENDER

Mario Roberto Samudio Martinez, Facultad de ingeniería, Universidad de Cundinamarca
Chía(Cundinamarca), Colombia
mrsamudio@ucundinamarca.edu.co

Abstract

This article presents the explanation of the code to generate a sinusoidal trajectory of a 3d object in blender in the python programming language, which allows the use of various libraries in order to draw the wavelengths by means of vectors for the recorder object. and can verso in the blender program.

Index Terms – Blender, Numbers, Mathematics, Python, Simulation

Resumen

Este artículo presenta la explicación del código para generar una trayectoria sinusoidal de un objeto 3d en blender en el lenguaje de programación python, que permite el uso de diversas librerías con el fin de dibujar las longitudes de onda por medio de vectores para que el objeto la recorra y pueda verse en el programa blender.

Índice de Términos – Blender, números, matemáticas, python, simulación

1. Introducción

En este artículo se presenta la explicación del código fuente[1] para simular la trayectoria de un objeto siguiendo la función seno en Blender, “un programa libre y de código abierto que soporta simulaciones, animaciones, modelado 3d edición de video, entre otros” [2],

Blender soporta la ejecución de scripts en python y posee una interfaz muy completa en la que se pueden ejecutar dichos scripts; La librería bpy es la interfaz de programación de aplicaciones de blender[3]. Se utiliza la función trigonométrica seno se encuentra definida en la librería math del lenguaje de programación python[4]. También se utiliza

la librería numpy, “orientada a el uso de arreglos multi-dimensionales. Funciones matemáticas, generación de números aleatorios, rutinas de álgebra lineal, transformadas de Fourier”[5], entre otras funcionalidades.

Luego de graficar la función se crea una restricción que enlaza a la cámara con el objeto, y finalmente se reproduce el recorrido del objeto a través de la función seno en la linea de tiempo de la interfaz gráfica de usuario de Blender.

2. Llamado a librerías y configuración del script en la interfaz blender

Al entrar en la interfaz scripting de blender[6] se encuentra el editor de texto, una consola interactiva, la vista 3d y las propiedades. En el editor de texto se ingresaron las librerías de las que depende el programa realizado para trazar la trayectoria de la función seno de la siguiente forma:

```
# Función seno
import bpy
import numpy as np
from math import sin
```

La instrucción import para estas tres primera líneas de código permitieron utilizar el API de blender para el acceso a los objetos y datos del programa. Por otro lado, a través de la librería numpy se logró el uso de vectores que fueron necesarios para listar los datos de las coordenadas de la función en un solo vector, mientras que la única función que se utilizó de librería math fue sin().

3. Variables, escena, objetos y funciones de utilidad

Lo habitual en la creación de scripts es declarar la variables globales del mismo al inicio, esto con el propósito de que las funciones e instrucciones posteriores a estas pudieran ubicar la escena, los grados de la circunferencia y los objetos con sus correspondientes opciones en la vista:

```
# Configuración de las variables
globales
```

```
scene = bpy.data.scenes['Scene']
g = 360 # grados de la circunferencia
obj = bpy.data.objects
objops = bpy.ops.object
objactive = bpy.context.scene.objects
```

En los siguientes bloques de funciones de código se muestran las funciones que fueron necesarias para seleccionar, borrar y crear objetos. Las dos primeras funciones fueron tomadas del trabajo de Conlan, que hacen parte de “un conjunto de herramientas mínimas para la creación, selección y transformación de objetos”[7]:

```
# Seleccionar el objeto por nombre
def seleccionar(objName):
    bpy.ops.object.select_all(action='DESELECT')
    bpy.data.objects[objName].select = True
```

```
# Borrar objeto por nombre
def borrar(objName):
    seleccionar(objName)
    bpy.ops.object.delete(use_global=False)
```

```
# Crear un objeto
def crearObjeto():
    bpy.ops.mesh.primitive_uv_sphere_add(size=0.5, location=(0, -8, 0))
    bpy.context.object.name = 'Esfera'
    pass
```

La función anterior “**crearObjeto()**” contiene dos instrucciones que sirvieron para crear una esfera con las coordenadas(x,y,z) en la posición ‘(0,-8,0)’ y nombrar el objeto ‘**Esfera**’ respectivamente. En el siguiente bloque de instrucciones se declararon las funciones necesarias para limpiar el objeto creado automáticamente al iniciar blender, elegir la escena y crear el objeto que siguió la trayectoria de la función seno:

```
borrar('Cube')
escena()
crearObjeto()
```

4. Trayectoria en la función seno

La función seno y la trayectoria del objeto creado(Esfera) se unieron en este bloque de instrucciones:

```
for i, j in zip(range(g), np.arange(0, 1, 0.00025)):
    bpy.context.scene.frame_float = i
    bpy.ops.transform.translate(value=(0, j, sin(i)))
    scene.timeline_markers.new(f'{i}', frame=i)
```

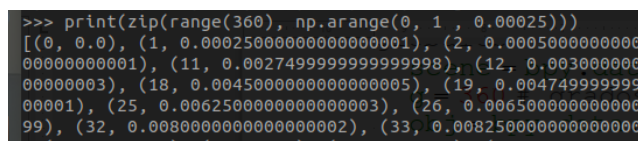
El ciclo se declaró con dos variables de control. La variable **i** que se asoció directamente con la función ‘**range(g)**’, que creó una lista de 360 elementos(parámetro definido previamente en la configuración de variables). La

variable **j** que se asoció con la función de la librería numpy ‘**arange(0,1, 0.00025)**’ que permitió crear un vector de 4000 elementos, de los cuales se utilizaron los primeros 360 ; esto fué debido a que dicha función se insertó como parámetro de la función ‘**zip(a,b)**’.

Las instrucciones que se relacionaron con la escena, sirvieron para fijar una posición por cada fotograma, que al final permitió la reproducción de la trayectoria.

La función ‘**zip(a,b)**’ concatenó la lista y el vector en un solo vector de tipo *llave:valor* el cual contenía 360 elementos que fueron utilizados para posicionar el objeto ‘**Esfera**’ con los valores ‘**x,y,z**’ que aumentaron en el ciclo ejecutado con la siguiente instrucción:

```
bpy.ops.transform.translate(value=(0, j, sin(i)))
```



```
>>> print(zip(range(360), np.arange(0, 1, 0.00025)))
[(0, 0.0), (1, 0.00025000000000000001), (2, 0.00050000000000000002), (3, 0.00075000000000000003), (4, 0.00100000000000000004), (5, 0.00125000000000000005), (6, 0.00150000000000000006), (7, 0.00175000000000000007), (8, 0.00200000000000000008), (9, 0.00225000000000000009), (10, 0.0025000000000000001), (11, 0.00274999999999999998), (12, 0.00300000000000000003), (13, 0.00325000000000000007), (14, 0.0035000000000000001), (15, 0.00375000000000000015), (16, 0.0040000000000000002), (17, 0.00425000000000000025), (18, 0.0045000000000000003), (19, 0.00474999999999999995), (20, 0.00500000000000000005), (21, 0.0052500000000000001), (22, 0.00550000000000000015), (23, 0.0057500000000000002), (24, 0.00600000000000000025), (25, 0.0062500000000000003), (26, 0.00650000000000000035), (27, 0.0067500000000000004), (28, 0.00700000000000000045), (29, 0.0072500000000000005), (30, 0.00750000000000000055), (31, 0.00774999999999999995), (32, 0.00800000000000000005), (33, 0.0082500000000000001), (34, 0.00850000000000000015), (35, 0.0087500000000000002), (36, 0.00900000000000000025), (37, 0.0092500000000000003), (38, 0.00950000000000000035), (39, 0.0097500000000000004), (40, 0.01000000000000000045), (41, 0.0102500000000000005), (42, 0.01050000000000000055), (43, 0.0107500000000000006), (44, 0.01100000000000000065), (45, 0.0112500000000000007), (46, 0.01150000000000000075), (47, 0.0117500000000000008), (48, 0.01200000000000000085), (49, 0.0122500000000000009), (50, 0.01250000000000000095), (51, 0.01274999999999999995), (52, 0.01300000000000000005), (53, 0.0132500000000000001), (54, 0.01350000000000000015), (55, 0.0137500000000000002), (56, 0.01400000000000000025), (57, 0.0142500000000000003), (58, 0.01450000000000000035), (59, 0.0147500000000000004), (60, 0.01500000000000000045), (61, 0.0152500000000000005), (62, 0.01550000000000000055), (63, 0.0157500000000000006), (64, 0.01600000000000000065), (65, 0.0162500000000000007), (66, 0.01650000000000000075), (67, 0.0167500000000000008), (68, 0.01700000000000000085), (69, 0.0172500000000000009), (70, 0.01750000000000000095), (71, 0.01774999999999999995), (72, 0.01800000000000000005), (73, 0.0182500000000000001), (74, 0.01850000000000000015), (75, 0.0187500000000000002), (76, 0.01900000000000000025), (77, 0.0192500000000000003), (78, 0.01950000000000000035), (79, 0.0197500000000000004), (80, 0.02000000000000000045), (81, 0.0202500000000000005), (82, 0.02050000000000000055), (83, 0.0207500000000000006), (84, 0.02100000000000000065), (85, 0.0212500000000000007), (86, 0.02150000000000000075), (87, 0.0217500000000000008), (88, 0.02200000000000000085), (89, 0.0222500000000000009), (90, 0.02250000000000000095), (91, 0.02274999999999999995), (92, 0.02300000000000000005), (93, 0.0232500000000000001), (94, 0.02350000000000000015), (95, 0.0237500000000000002), (96, 0.02400000000000000025), (97, 0.0242500000000000003), (98, 0.02450000000000000035), (99, 0.0247500000000000004), (100, 0.02500000000000000045), (101, 0.0252500000000000005), (102, 0.02550000000000000055), (103, 0.0257500000000000006), (104, 0.02600000000000000065), (105, 0.0262500000000000007), (106, 0.02650000000000000075), (107, 0.0267500000000000008), (108, 0.02700000000000000085), (109, 0.0272500000000000009), (110, 0.02750000000000000095), (111, 0.02774999999999999995), (112, 0.02800000000000000005), (113, 0.0282500000000000001), (114, 0.02850000000000000015), (115, 0.0287500000000000002), (116, 0.02900000000000000025), (117, 0.0292500000000000003), (118, 0.02950000000000000035), (119, 0.0297500000000000004), (120, 0.03000000000000000045), (121, 0.0302500000000000005), (122, 0.03050000000000000055), (123, 0.0307500000000000006), (124, 0.03100000000000000065), (125, 0.0312500000000000007), (126, 0.03150000000000000075), (127, 0.0317500000000000008), (128, 0.03200000000000000085), (129, 0.0322500000000000009), (130, 0.03250000000000000095), (131, 0.03274999999999999995), (132, 0.03300000000000000005), (133, 0.0332500000000000001), (134, 0.03350000000000000015), (135, 0.0337500000000000002), (136, 0.03400000000000000025), (137, 0.0342500000000000003), (138, 0.03450000000000000035), (139, 0.0347500000000000004), (140, 0.03500000000000000045), (141, 0.0352500000000000005), (142, 0.03550000000000000055), (143, 0.0357500000000000006), (144, 0.03600000000000000065), (145, 0.0362500000000000007), (146, 0.03650000000000000075), (147, 0.0367500000000000008), (148, 0.03700000000000000085), (149, 0.0372500000000000009), (150, 0.03750000000000000095), (151, 0.03774999999999999995), (152, 0.03800000000000000005), (153, 0.0382500000000000001), (154, 0.03850000000000000015), (155, 0.0387500000000000002), (156, 0.03900000000000000025), (157, 0.0392500000000000003), (158, 0.03950000000000000035), (159, 0.0397500000000000004), (160, 0.04000000000000000045), (161, 0.0402500000000000005), (162, 0.04050000000000000055), (163, 0.0407500000000000006), (164, 0.04100000000000000065), (165, 0.0412500000000000007), (166, 0.04150000000000000075), (167, 0.0417500000000000008), (168, 0.04200000000000000085), (169, 0.0422500000000000009), (170, 0.04250000000000000095), (171, 0.04274999999999999995), (172, 0.04300000000000000005), (173, 0.0432500000000000001), (174, 0.04350000000000000015), (175, 0.0437500000000000002), (176, 0.04400000000000000025), (177, 0.0442500000000000003), (178, 0.04450000000000000035), (179, 0.0447500000000000004), (180, 0.04500000000000000045), (181, 0.0452500000000000005), (182, 0.04550000000000000055), (183, 0.0457500000000000006), (184, 0.04600000000000000065), (185, 0.0462500000000000007), (186, 0.04650000000000000075), (187, 0.0467500000000000008), (188, 0.04700000000000000085), (189, 0.0472500000000000009), (190, 0.04750000000000000095), (191, 0.04774999999999999995), (192, 0.04800000000000000005), (193, 0.0482500000000000001), (194, 0.04850000000000000015), (195, 0.0487500000000000002), (196, 0.04900000000000000025), (197, 0.0492500000000000003), (198, 0.04950000000000000035), (199, 0.0497500000000000004), (200, 0.05000000000000000045), (201, 0.0502500000000000005), (202, 0.05050000000000000055), (203, 0.0507500000000000006), (204, 0.05100000000000000065), (205, 0.0512500000000000007), (206, 0.05150000000000000075), (207, 0.0517500000000000008), (208, 0.05200000000000000085), (209, 0.0522500000000000009), (210, 0.05250000000000000095), (211, 0.05274999999999999995), (212, 0.05300000000000000005), (213, 0.0532500000000000001), (214, 0.05350000000000000015), (215, 0.0537500000000000002), (216, 0.05400000000000000025), (217, 0.0542500000000000003), (218, 0.05450000000000000035), (219, 0.0547500000000000004), (220, 0.05500000000000000045), (221, 0.0552500000000000005), (222, 0.05550000000000000055), (223, 0.0557500000000000006), (224, 0.05600000000000000065), (225, 0.0562500000000000007), (226, 0.05650000000000000075), (227, 0.0567500000000000008), (228, 0.05700000000000000085), (229, 0.0572500000000000009), (230, 0.05750000000000000095), (231, 0.05774999999999999995), (232, 0.05800000000000000005), (233, 0.0582500000000000001), (234, 0.05850000000000000015), (235, 0.0587500000000000002), (236, 0.05900000000000000025), (237, 0.0592500000000000003), (238, 0.05950000000000000035), (239, 0.0597500000000000004), (240, 0.06000000000000000045), (241, 0.0602500000000000005), (242, 0.06050000000000000055), (243, 0.0607500000000000006), (244, 0.06100000000000000065), (245, 0.0612500000000000007), (246, 0.06150000000000000075), (247, 0.0617500000000000008), (248, 0.06200000000000000085), (249, 0.0622500000000000009), (250, 0.06250000000000000095), (251, 0.06274999999999999995), (252, 0.06300000000000000005), (253, 0.0632500000000000001), (254, 0.06350000000000000015), (255, 0.0637500000000000002), (256, 0.06400000000000000025), (257, 0.0642500000000000003), (258, 0.06450000000000000035), (259, 0.0647500000000000004), (260, 0.06500000000000000045), (261, 0.0652500000000000005), (262, 0.06550000000000000055), (263, 0.0657500000000000006), (264, 0.06600000000000000065), (265, 0.0662500000000000007), (266, 0.06650000000000000075), (267, 0.0667500000000000008), (268, 0.06700000000000000085), (269, 0.0672500000000000009), (270, 0.06750000000000000095), (271, 0.06774999999999999995), (272, 0.06800000000000000005), (273, 0.0682500000000000001), (274, 0.06850000000000000015), (275, 0.0687500000000000002), (276, 0.06900000000000000025), (277, 0.0692500000000000003), (278, 0.06950000000000000035), (279, 0.0697500000000000004), (280, 0.07000000000000000045), (281, 0.0702500000000000005), (282, 0.07050000000000000055), (283, 0.0707500000000000006), (284, 0.07100000000000000065), (285, 0.0712500000000000007), (286, 0.07150000000000000075), (287, 0.0717500000000000008), (288, 0.07200000000000000085), (289, 0.0722500000000000009), (290, 0.07250000000000000095), (291, 0.07274999999999999995), (292, 0.07300000000000000005), (293, 0.0732500000000000001), (294, 0.07350000000000000015), (295, 0.0737500000000000002), (296, 0.07400000000000000025), (297, 0.0742500000000000003), (298, 0.07450000000000000035), (299, 0.0747500000000000004), (300, 0.07500000000000000045), (301, 0.0752500000000000005), (302, 0.07550000000000000055), (303, 0.0757500000000000006), (304, 0.07600000000000000065), (305, 0.0762500000000000007), (306, 0.07650000000000000075), (307, 0.0767500000000000008), (308, 0.07700000000000000085), (309, 0.0772500000000000009), (310, 0.07750000000000000095), (311, 0.07774999999999999995), (312, 0.07800000000000000005), (313, 0.0782500000000000001), (314, 0.07850000000000000015), (315, 0.0787500000000000002), (316, 0.07900000000000000025), (317, 0.0792500000000000003), (318, 0.07950000000000000035), (319, 0.0797500000000000004), (320, 0.08000000000000000045), (321, 0.0802500000000000005), (322, 0.08050000000000000055), (323, 0.0807500000000000006), (324, 0.08100000000000000065), (325, 0.0812500000000000007), (326, 0.08150000000000000075), (327, 0.0817500000000000008), (328, 0.08200000000000000085), (329, 0.0822500000000000009), (330, 0.08250000000000000095), (331, 0.08274999999999999995), (332, 0.08300000000000000005), (333, 0.0832500000000000001), (334, 0.08350000000000000015), (335, 0.0837500000000000002), (336, 0.08400000000000000025), (337, 0.0842500000000000003), (338, 0.08450000000000000035), (339, 0.0847500000000000004), (340, 0.08500000000000000045), (341, 0.0852500000000000005), (342, 0.08550000000000000055), (343, 0.0857500000000000006), (344, 0.08600000000000000065), (345, 0.0862500000000000007), (346, 0.08650000000000000075), (347, 0.0867500000000000008), (348, 0.08700000000000000085), (349, 0.0872500000000000009), (350, 0.08750000000000000095), (351, 0.08774999999999999995), (352, 0.08800000000000000005), (353, 0.0882500000000000001), (354, 0.08850000000000000015), (355, 0.0887500000000000002), (356, 0.08900000000000000025), (357, 0.0892500000000000003), (358, 0.08950000000000000035), (359, 0.0897500000000000004), (360, 0.09000000000000000045), (361, 0.0902500000000000005), (362, 0.09050000000000000055), (363, 0.0907500000000000006), (364, 0.09100000000000000065), (365, 0.0912500000000000007), (366, 0.09150000000000000075), (367, 0.0917500000000000008), (368, 0.09200000000000000085), (369, 0.0922500000000000009), (370, 0.09250000000000000095), (371, 0.09274999999999999995), (372, 0.09300000000000000005), (373, 0.0932500000000000001), (374, 0.09350000000000000015), (375, 0.0937500000000000002), (376, 0.09400000000000000025), (377, 0.0942500000000000003), (378, 0.09450000000000000035), (379, 0.0947500000000000004), (380, 0.09500000000000000045), (381, 0.0952500000000000005), (382, 0.09550000000000000055), (383, 0.0957500000000000006), (384, 0.09600000000000000065), (385, 0.0962500000000000007), (386, 0.09650000000
```

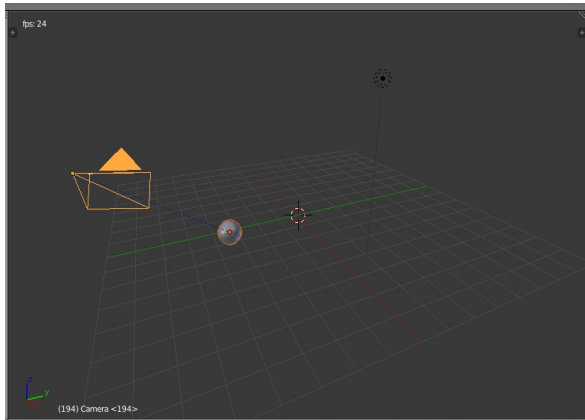


Ilustración 2: Objeto siguiendo trayectoria sinusoidal

6. Conclusiones

Por medio de blender se pudo evidenciar que es posible el uso de scripts que permitan simular cualquier tópico de cualquier área de conocimiento. Prueba de ello fue el recorrido del objeto siguiendo una trayectoria sinusoidal, por medio de el concepto del seno, la trigonometría y la programación.

7. Referencias bibliográficas

[1] M. Samudio, «mrsamudio/simulacion», may 25, 2020. <https://github.com/mrsamudio/simulacion> (accedido may 25, 2020).

[2] B. Foundation, «blender.org - Home of the Blender project - Free and Open 3D Creation Software», *blender.org*. <https://www.blender.org/> (accedido may 25, 2020).

[3] «Reference API Usage — Blender 2.79.0 855d2955c49 - API documentation». https://docs.blender.org/api/2.79/info_api_reference.html (accedido may 25, 2020).

[4] «math — Mathematical functions — Python 3.8.3 documentation». <https://docs.python.org/3/library/math.html> (accedido may 25, 2020).

[5] «NumPy». <https://numpy.org/> (accedido may 25, 2020).

[6] *El interface de Blender.* .

[7] C. Conlan, *The Blender Python API*. Berkeley, CA: Apress, 2017.



Mario Roberto Samudio Martinez nació en la ciudad de Bogotá, en 1986. Recibió el título profesional de pregrado en ingeniería de sistemas de la universidad de Cundinamarca, en el municipio de Chía, en el año 2020.