

Modelado y animación por computador-Practica 4

Por Octavio Gregorio Guerrero

ÍNDICE

1. Idea y desarrollo.....	1
2. Apartados del add-on y explicación del código.....	3
2.1. Creación de escenario.....	3
2.2. Inicio de la lluvia.....	5
3. Utilización del add-on paso a paso.....	7

1. Idea y desarrollo

La **idea inicial** del proyecto era realizar un **Add-on** que simulará un terremoto. El pensamiento era que el usuario pudiera determinar la intensidad del sismo así como los edificios de la escena a simular.

Sin embargo, cuando me puse a ello, me percaté que la ambición del proyecto era demasiado alta, debido a que nunca antes había utilizado **Python** y se me hacía cuesta arriba el tener que aprender un lenguaje de 0.

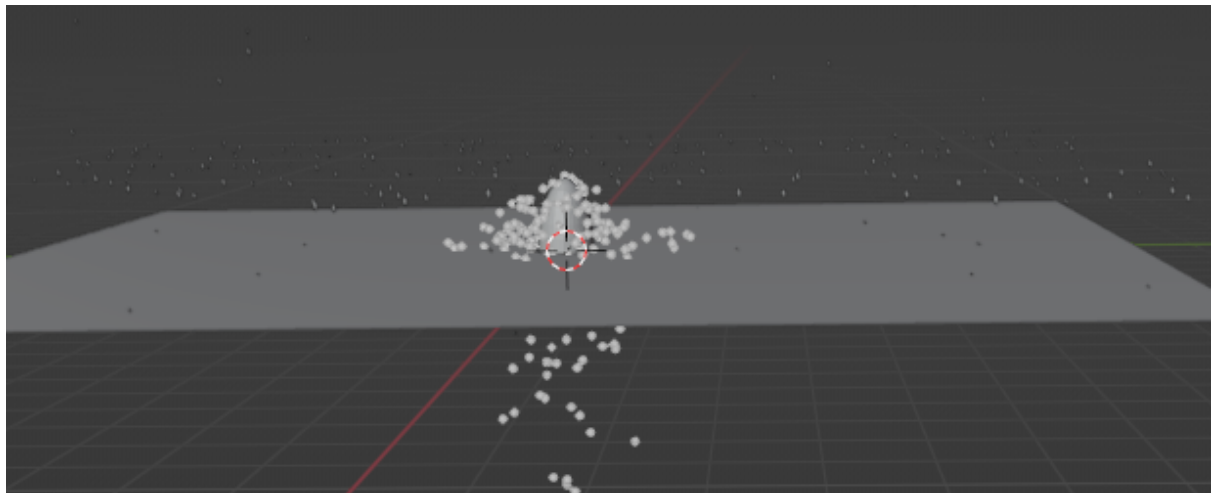
Poco a poco fui comprendiendo el lenguaje, realicé las pruebas de los apuntes de la asignatura, y me puse a trabajar con el **simulador**.

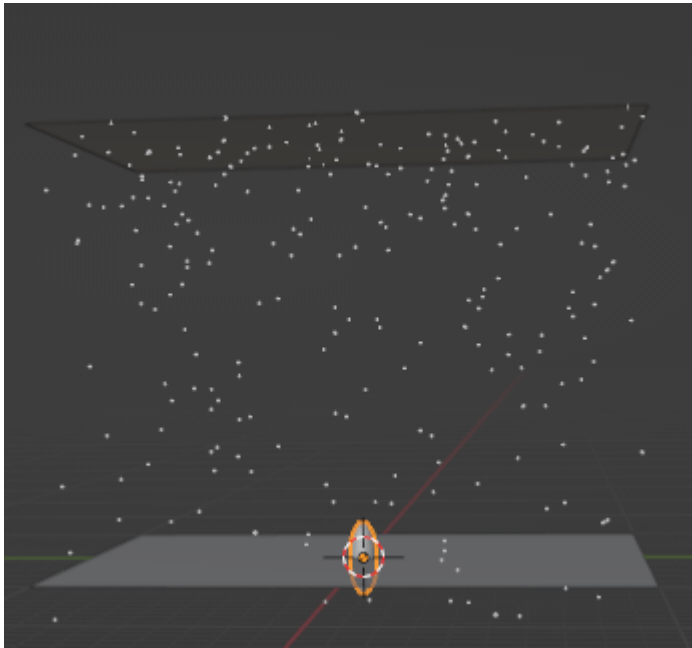
No obstante, mis ideas iban variando mucho mientras el tiempo se acababa. Pensé en realizar desde la caída de un meteorito, hasta un tornado, o incluso un patito de goma en una bañera que actuara en base a las ondas del agua.

Al final, me terminé decantando por un **simulador de lluvia**, y como tenía poco tiempo, me puse manos a la obra

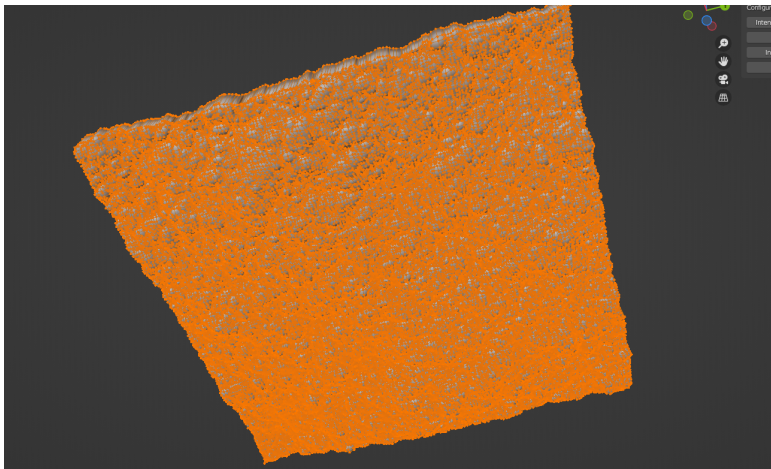
En el proceso fui entendiendo un poco más python, aunque fue complicado ya que solo tenía de experiencia lo antes mencionado.

Empecé a toquetear con **partículas**, **creación de objetos**, **movimiento**, **escalas** ect...





A la vez, me puse con las **nubes**, que fue sin duda lo que más tiempo y trabajo me llevó, pero que lo veía necesario para que el usuario pudiera adaptar la escena a su gusto con diferentes iluminaciones.



Cabe destacar, que durante el proceso de la nube, el proyecto se me **eliminó** 3 veces, ya que al parecer al realizar CTR+Z, el programa se cerraba sin rastro alguno del archivo.(A día de hoy sigo sin entender porqué pasa)

Esto causó que recortara aún mas mi tiempo para fusionar la lluvia y la nube lo antes posible.

Así que añadí la programación de la **iluminación** y la personalicé para su uso directo y aproveché para añadir unos **keyframes** para así simular el **viento** en las nubes

2. Apartados del add-on y explicación del código

2.1. Creación de escenario

En esta primera parte, eliminamos los objetos de la escena para **evitar duplicados y problemas**.

```
class CreaEscena(bpy.types.Operator):
    bl_idname = "object.crea_escena"
    bl_label = "Crea Escena"

    def execute(self, context):
        # Eliminar objetos existentes en la escena
        bpy.ops.object.select_all(action='DESELECT')
        bpy.ops.object.select_by_type(type='MESH')

        for obj in bpy.context.selected_objects:
            # Verificar si el objeto tiene materiales
            if obj.type == 'MESH' and obj.data.materials:
                # Eliminar cada material del objeto
                for material_slot in obj.material_slots:
                    material = material_slot.material
                    if material:
                        bpy.data.materials.remove(material)
                        obj.data.materials.clear()

        bpy.ops.object.delete()

        bpy.ops.object.select_by_type(type='LIGHT')
        bpy.ops.object.delete()
```

Eliminamos tanto objetos, como materiales o luces, por lo que hay que preparar la escena bien antes de comenzar a trabajar en cualquier proyecto.

A continuación **creamos una luz** para la nube que inicializamos más adelante y un punto de luz solar para dar ambiente a la escena.

```
bpy.ops.object.select_by_type(type='LIGHT')
bpy.ops.object.delete()

bpy.ops.object.light_add(type='SUN', radius=1, align='WORLD', location=(0, 0, 15), scale=(1, 1, 1))
bpy.ops.transform.rotate(value=1.5708, orient_axis='Z', orient_type='VIEW', orient_matrix=((0.046959, -0.998897, 2.04192e-07), (-0.0759307, -0.00356941, 0.997107), (-0.996007, -0.0468232, -0.0760146)),
orient_matrix_type='VIEW', mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1, use_proportional_connected=False, use_proportional_projected=False, snap=False,
snap_elements={'INCREMENT'}), use_snap_project=False, snap_target='CLOSEST', use_snap_self=True, use_snap_edit=True, use_snap_nonedit=True, use_snap_selectable=False)
bpy.context.object.data.energy = 4
bpy.context.object.data.color = (0.358841, 0.0924809, 0.0133396)

bpy.ops.object.light_add(type='AREA', radius=40, align='WORLD', location=(0, 0, 16), scale=(1, 1, 1))
bpy.ops.transform.rotate(value=3.14159, orient_axis='X', orient_type='GLOBAL', orient_matrix=((1, 0, 0), (0, 1, 0), (0, 0, 1)), orient_matrix_type='GLOBAL', constraint_axis=(True, False, False),
mirror=False, use_proportional_edit=False, proportional_edit_falloff='SMOOTH', proportional_size=1, use_proportional_connected=False, use_proportional_projected=False, snap=False, snap_elements={'INCREMENT'}),
use_snap_project=False, snap_target='CLOSEST', use_snap_self=True, use_snap_edit=True, use_snap_nonedit=True, use_snap_selectable=False)

#bpy.context.object.data.energy = 40000
bpy.context.object.data.energy = 20000
bpy.context.object.data.color = (0.241607, 0.356979, 0.367238)
```

Aprovechamos también para darles **color** y **potencia** a ambas iluminaciones

Ahora creamos los **dos planos** y cambiamos el plano de arriba para que tenga la forma de nube deseada.

```
# Crear el primer plano
bpy.ops.mesh.primitive_plane_add(size=100)
plane1 = bpy.context.object
plane1.location.z = 0

# Crear el segundo plano
bpy.ops.mesh.primitive_plane_add(size=20)
plane2 = bpy.context.object
plane2.location.z = 15

#NUBE
bpy.ops.object.modifier_add(type='OCEAN')
bpy.context.object.modifiers["Ocean"].spectrum = 'PIERSON_MOSKOWITZ'
bpy.context.object.modifiers["Ocean"].viewport_resolution = 10
bpy.context.object.modifiers["Ocean"].choppiness = 1.15
bpy.context.object.modifiers["Ocean"].wave_scale = 1.5
bpy.context.object.modifiers["Ocean"].viewport_resolution = 15
bpy.ops.object.modifier_apply(modifier="Ocean")

bpy.ops.object.editmode_toggle()
bpy.ops.mesh.select_non_manifold()
bpy.ops.mesh.extrude_region_move(MESH_OT_extrude_region={"use_normal_flip":
bpy.ops.transform.resize(value=(1, 1, 0), orient_type='GLOBAL', orient_matr
bpy.ops.mesh.edge_face_add()
bpy.ops.transform.translate(value=(-0, -0, -1.95055), orient_axis_ortho='X
bpy.ops.mesh.select_all(action='SELECT')
bpy.ops.mesh.normals_make_consistent(inside=False)
bpy.ops.object.editmode_toggle()

# Obtén una referencia al objeto al que deseas asignar el material
nombre_objeto = "Plane.001"
objeto = bpy.data.objects[nombre_objeto]

# Crea un nuevo material
nombre_material = "Prueba"
material = bpy.data.materials.new(name=nombre_material)
material.use_nodes = True
```

Para finalizar la parte de creación de la escena añadimos el **material** y creamos una serie de **modificadores** necesarios para la posterior lluvia.

```
objeto.data.materials.append(material)

objeto.active_material_index = 0

material = bpy.data.materials["Prueba"]

nodo_materiales = material.node_tree.nodes

for enlace in material.node_tree.links:
    material.node_tree.links.remove(enlace)

nodo_principled_volume = nodo_materiales.new(type='ShaderNodeVolumePrincipled')

nodo_output = nodo_materiales.get('Material Output')
enlace_volume = material.node_tree.links.new(nodo_principled_volume.outputs['Volume'], nodo_output.inputs['Volume'])

bpy.ops.object.modifier_add(type='DISPLACE')
bpy.context.object.modifiers["Displace"].strength = -0.25

plane2
```

También dejamos el plano de arriba seleccionada para que no ocurran problemas al hacer el siguiente paso(También se puede seleccionar el plano a mano)

2.2. Inicio de la lluvia

En esta parte del programa creamos las **gotas** de agua para la lluvia las cuales serán añadidas más adelante a las nubes

```
bpy.ops.object.particle_system_add()

# Creamos el splash
bpy.ops.mesh.primitive_ico_sphere_add(subdivisions=1, radius=1, location=(100, 0, 0))
bpy.ops.object.shade_smooth()
splash = bpy.context.object

# Creamos la gota
bpy.ops.mesh.primitive_uv_sphere_add(radius=1, location=(100, 0, 0))
gota = bpy.context.object
gota.scale = (0.5, 0.5, 1.3)
```

Aprovechamos también para crear el objeto de **splash** el cual genera un efecto de salpicar cuando la gota toca el suelo

A continuación realizamos unos ajustes a las **partículas** de las gotas para que la lluvia sea lo **más realista** posible.

```
particle_settings = plano.particle_systems[0].settings

particle_settings.count = int(scene.intensidad * 2000)
particle_settings.frame_start = -100
particle_settings.frame_end = 200
particle_settings.size_random = 0.5

particle_settings.render_type = 'OBJECT'

gota = bpy.data.objects['Sphere']

# Establecer la gota como objeto de renderizado
particle_settings.instance_object = gota
```

También aprovechamos y añadimos el **factor intensidad** mediante el cual el usuario podrá **controlar** posteriormente la **cantidad de lluvia** que cae en la escena

En esta parte del código hacemos algo similar a lo anterior pero esta vez para el **plano de abajo**, el cual tendrá un **efecto de Splash** cada vez que las gotas caen contra el mismo.

```
plano2 = bpy.data.objects['Plane']
modificador_colision = plano2.modifiers.new(name="Colisión", type='COLLISION')

modificador_part = plano2.modifiers.new(name="Particulas", type='PARTICLE_SYSTEM')
bpy.data.particles["ParticleSettings"].count = 2000
bpy.data.particles["ParticleSettings"].frame_start = 0
bpy.data.particles["ParticleSettings"].lifetime = 10

bpy.context.scene.frame_end = 200

particle_settings = plano2.particle_systems[0].settings

particle_settings.count = int(scene.intensidad * 1000)
particle_settings.frame_start = 0
particle_settings.lifetime = 10
particle_settings.lifetime = 10
particle_settings.particle_size = 0.025
particle_settings.size_random = 0.5

particle_settings.render_type = 'OBJECT'

splash = bpy.data.objects['Icosphere']

particle_settings.instance_object = splash
```

Además se le añade **colisión** para que las gotas no atravesen el plano y creen un **mejor efecto** a la vista.

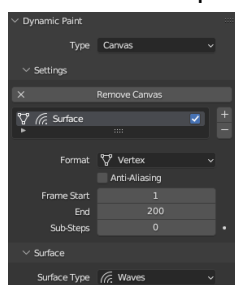
Estos ajustes de aquí son unos **extras**, ya que quería realizar el efecto de una gota cuando caía en agua, sin embargo, para realizar esto no podía acceder a los parámetros que yo quería mediante python así es que tendría que ser a mano.

```
#hacemos que el suelo responda a las gotas
modificador_subdivision = plano2.modifiers.new(name="Subdi", type='SUBSURF')
modificador_subdivision.subdivision_type = 'SIMPLE'
modificador_subdivision.levels = 7
bpy.ops.object.shade_smooth()

modificador_dynamic = plano2.modifiers.new(name="Dynamic", type='DYNAMIC_PAINT')
bpy.ops.dpaint.type_toggle(type='CANVAS')

#bpy.context.object.modifiers["Dynamic Paint"].canvas_settings.canvas_surfaces["Surface"].surface_type = 'WAVE'
#bpy.context.object.modifiers["Dynamic Paint"].canvas_settings.canvas_surfaces["Surface"].brush_influence_scale = 0.5
#bpy.context.object.modifiers["Dynamic"].ui_type = 'BRUSH'
#bpy.context.object.modifiers["Dynamic"].brush_settings.paint_source = 'PARTICLE_SYSTEM'
#bpy.context.object.modifiers["Dynamic"].brush_settings.particle_system = bpy.data.objects["Plane.001"].particle_systems["ParticleSystem"]
```

Sin embargo dejo añadido estas líneas de código por si acaso el usuario quiere realizar este efecto en cualquier momento a mano.



Por último pero no menos importante creamos los **keyframes** para el movimiento de una nube que parezca realista.

```
bpy.ops.object.select_all(action='DESELECT')

obj1 = bpy.data.objects["Plane.001"]
obj2 = bpy.data.objects["Area"]

obj1.select_set(True)
obj2.select_set(True)

obj1.location.x = +15
obj2.location.x = +15

obj1.keyframe_insert(data_path="location", frame=500)
obj2.keyframe_insert(data_path="location", frame=500)

obj1.location.x = -5
obj2.location.x = -5

obj1.keyframe_insert(data_path="location", frame=1)
obj2.keyframe_insert(data_path="location", frame=1)
```

Las gotas de agua **seguirán** todo el **trayecto** de la nube, en base a la velocidad de la misma

3.Utilización del add-on paso a paso

- **Instalar** el add-on accediendo a Preferencias > Add-ons > Install, y selecciona el .py
- Una vez teniendo instalado el add-on, podrás **encontrarlo** dentro de View3D > Tools
- Aquí podrás observar las pestañas disponibles, las cuales, debes pulsar en un orden específico para que funcione:

1 - Selecciona la **intensidad de la lluvia deseada**. A mayor número, más agua

2 - A continuación, selecciona opción de **crear escenario**, la cual, creará una zona en la que más adelante poner tus propios objetos y proyectos

3 - Por último, presiona la opción de **iniciar lluvia**, la cual creará las gotas de agua randomizadas y en base a la intensidad de lluvia puesta anteriormente.(En caso de que no funcione asegúrate de tener seleccionado las nubes antes de presionar el botón, ya que debes de seleccionar de donde aparece la lluvia)

4 - El botón de Iniciar/Detener lluvia te permitirá **detener y reanudar** la animación en cualquier momento

Ahora ya puedes jugar con los ajustes de la luz para que la escena quede a tu gusto.

En este caso, se le añade una **luz interior** a las nubes un poco azulado para hacer un tono más **cartoon** y se le suma una luz del **sol** al atardecer que golpea algunas nubes, el suelo, y las gotas de lluvia, contrastando con un toque de **realismo** que le da vida a cualquier proyecto que quieras poner debajo.