

# SISTEMA SOLAR

---

PROYECTO MAXSCRIPT – MODELADO Y ANIMACIÓN POR  
COMPUTADOR

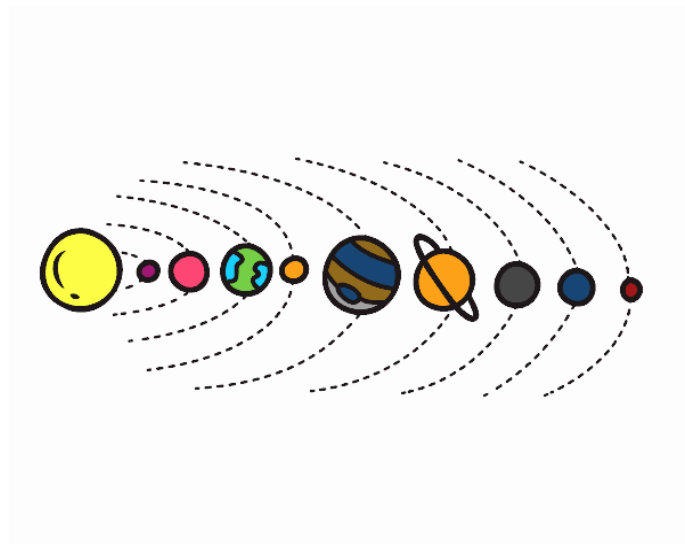
PABLO GUILLÉN, JOSÉ LUIS SEGURA, JOSÉ LISÓN  
INGENIERIA MULTIMEDIA | UNIVERSIDAD ALICANTE

## 1. Introducción

Nuestro grupo se ha decantado por la “Opción A = PLUGIN MAXSCRIPT de entidad + casos de uso”.

Nuestro plugin consistirá en la creación de un sistema planetario según una serie de parametrizaciones personalizados por el usuario. Este sistema planetario estará formado principalmente por una estrella principal en el centro del mismo (sol) y una serie de cuerpos celestes que giran alrededor. Nuestra idea es imitar nuestro sistema solar.

También añadiremos sistemas de partículas para simular explosiones o materiales especiales simulando a los planetas.



### 1.1. Referencias

Después de mandar la tutoría para validar la idea del proyecto, el profesor nos comentó que la idea en sí ya existía y estaba documentada en un libro de 3ds Max. Por lo que con la aprobación del profesor decidimos apoyarnos en este libro, mejorar sus parametrizaciones y añadir muchas nuevas.

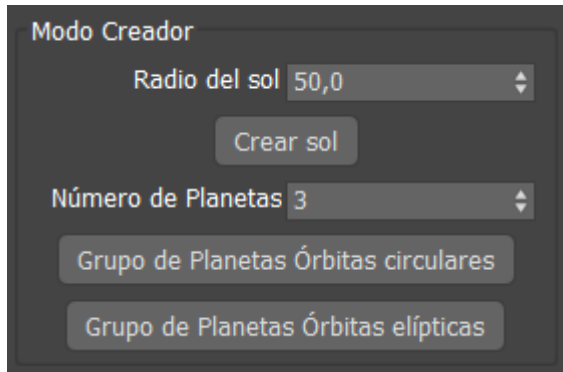
Este libro se llama “3ds Max MAXScript Essentials” de Autodesk y estaba en la bibliografía de la asignatura.

## 2. RollOut

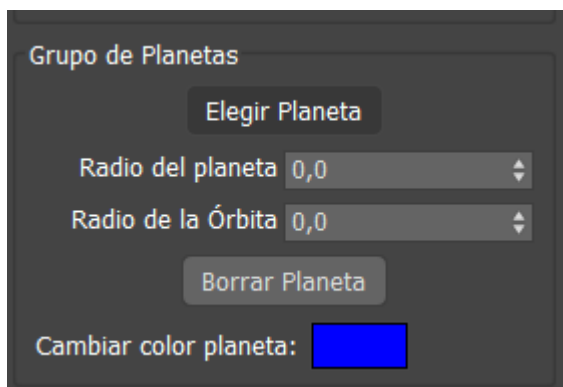
En cuanto al RollOut, hemos organizado las diferentes acciones en grupos según la acción o parametrización que realizan.

Primero tenemos una sección llamada ‘Modo Creador’. En ella tenemos diferentes opciones como crear el Sol eligiendo previamente su radio, y crear un grupo de planetas. En este segundo caso,

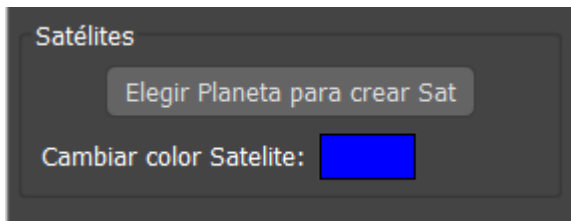
seleccionaremos previamente el número de planetas que queremos y ofrecemos dos opciones, crear los planetas con órbitas circulares o con órbitas elípticas, para lo que tenemos dos botones distintos uno encima de otro.



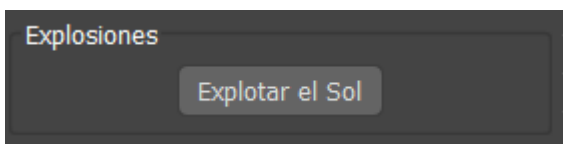
El siguiente grupo tiene el nombre 'Grupo de Planetas'. Aquí las opciones que ofrecemos son para modificar planetas y para eliminarlos. Para el primer caso debemos utilizar el botón 'Elegir planetas' y seleccionar uno de ellos, entonces podremos modificar su radio y el radio de su órbita. Además, podremos cambiar el color de dicho Planeta clickando en un recuadro donde se selecciona este en una paleta de colores. Para el segundo debemos clickar en el botón 'Borrar Planeta' y seleccionar el planeta que deseamos eliminar.



En tercer lugar, tenemos el grupo 'Satélites' en el que utilizamos el botón 'Elegir planeta para crear Sat' para posteriormente clickar en un planeta en el que automáticamente se generará un satélite con su propia órbita alrededor del planeta. Además tenemos un recuadro para cambiar el color de dicho satélite.



Por último, tenemos el grupo 'Explosiones', con un único botón que genera una animación en la que se crean unas partículas a modo de 'Lluvia de fuego' desde el interior del sol.



### 3.Código

```

rollout ssRoll "Solar System"
(
  group "Modo Creador"
  (
    spinner spn_sunRadius "Radio del sol" range:[1,1000,50]
    button but_createSun "Crear sol"
    spinner spn_numPlanets "Número de Planetas" type:#integer range:[1,10,3]
    button but_createPlanets "Grupo de Planetas Órbitas circulares"
    button but_createPlanetsElip "Grupo de Planetas Órbitas elípticas"
  )
  group "Grupo de Planetas"
  (
    pickbutton pbt_pickPlanet "Elegir Planeta"
    spinner spn_planetRadius "Radio del planeta"
    spinner spn_orbitRadius "Radio de la Órbita" range:[0,1000,0]
    pickbutton pbt_destruir"Borrar Planeta"
    colorpicker theColor "Cambiar color planeta:" color:[0,0,255] modal:false
  )
  group "Satélites"
  (
    pickbutton pbt_pickPlanetSat "Elegir Planeta para crear Sat"
    colorpicker colorsat "Cambiar color Satellite:" color:[0,0,255] modal:false
  )
  group "Explosiones"
  (
    button disparo "Explotar el Sol"
  )
)

```

Al principio del código declaramos el 'rollout'. Cada uno de estos grupos corresponde al código de creación de la interfaz previamente explicada en el punto 3. Con 'group' agrupamos diferentes tipos de botones. Dentro de ellos podemos encontrar 'button' que son botones normales los cuales al pulsarlos producen alguna acción, 'pickbutton', con los cuales deberemos seleccionar algún objeto después de presionarlos para que se produzca algún efecto sobre el, 'spinner', que son ruletas para seleccionar números en este caso, y 'colorpicker', que corresponden a los recuadros para seleccionar un color.

Posteriormente, el código consiste en darle una funcionalidad a cada uno de estos botones u opciones. Comencemos explicando paso a paso cada uno de ellos.

```
on but_createSun pressed do
(
    sun = sphere radius:spn_sunRadius.value
    sun.wirecolor = yellow
)
```

Este botón corresponde a 'Crear sol' y lo que hace es crear una esfera a modo de sol con el radio que le hayamos dado antes con el spinner 'spn\_sunRadius'. Se le asigna el color amarillo.

```
on but_createPlanets pressed do
(
    for i = 1 to spn_numPlanets.value do
    (
        -- Crear planeta
        planet = sphere()
        -- Establecer radio
        planet.radius = random 10.0 30.0
        planet.wirecolor = color (random 0 255) (random 0 255) (random 0 255)
        -- Crear órbita circular
        orbit = circle radius:(i*100)
        orbit.wirecolor = white
        -- Poner el planeta en la órbita
        planet.pos.controller = Path_Constraint()
        planet.pos.controller.path = orbit
        -- Rotar
        orbit.rotation.z_rotation = random 0.0 360.0
    )
)
```

Este botón corresponde a 'Grupo de planetas con órbitas circulares'. En él, valiéndonos de un bucle for para crear el número de planetas que hayamos seleccionado en el spinner spn\_numPlanets, creamos una esfera con radio y color aleatorios, una órbita en relación al número del contador del bucle para crearlos cada vez más lejos y le asignamos una órbita para poder rotar sobre ella.

```

on but_createPlanetsElip pressed do
(
  for i = 1 to spn_numPlanets.value do
  (
    --GRUPO DE PLANETAS CERCANOS AL SOL PEQUEÑOS

    planet = sphere() -- Create the planet
    planet.radius = random 10.0 18.0
    -- Color planetas aleatorios
    planet.wirecolor = color (random 0 255) (random 0 255) (random 0 255)
    orbit = ellipse()
    orbit.length = i * random 300 310
    orbit.width = i * random 200 210
    --Pintar color de la órbita para diferenciarla facilmente
    orbit.wirecolor = green
    planet.pos.controller = Path_Constraint()
    planet.pos.controller.path = orbit
    orbit.rotation.z_rotation = random 0.0 360.0

    --GRUPO DE PLANETAS LEJANOS AL SOL CON ANILLOS

    esfera = sphere()
    esfera.wirecolor = color (random 0 255) (random 0 255) (random 0 255)
    nose = cylinder()
    nose.wirecolor = yellow
    esfera.radius = random 25.0 35.0
    nose.radius = random 40.0 45.0
    nose.height = 0.2
    orbita = ellipse()
    orbita.length = i * random 1500 1515
    orbita.width = i * random 1100 1115
    orbita.wirecolor = white
    esfera.pos.controller = Path_Constraint()
    nose.pos.controller = Path_Constraint()
    esfera.pos.controller.path = orbita
    nose.pos.controller.path = orbita
    orbita.rotation.z_rotation = random 0.0 360.0
  )
)

```

Este botón tiene la misma base que el anterior, creación de planetas con sus órbitas, solo que en este caso, al declarar la órbita no creamos un círculo, creamos una elipse. Este elipse se la asignamos como órbita al planeta correspondiente que estemos creando en ese momento.

Para la siguiente sección, declaramos unas variables locales.

```

local pPlanet
local pOrbit
local pSat
local pPos

```

Veamos:

```
on pbt_pickPlanet picked aPlanet do
(
  pPlanet = aPlanet
  -- Elegir planeta
  spn_planetRadius.value = aPlanet.radius
  pOrbit = aPlanet.pos.controller.path
  spn_orbitRadius.value = pOrbit.radius
)
```

Ha este botón le corresponde la función de elegir un planeta. En el, el planeta seleccionado se guarda en una variable local junto con su órbita y radio para ser utilizados posteriormente.

```
on spn_planetRadius changed value do
(
  pPlanet.radius = spn_planetRadius.value
)
```

Aquí nos encargamos de cambiar el valor del radio del planeta seleccionado por el del spinner en cuestión.

```
on spn_orbitRadius changed value do
(
  pOrbit.radius = spn_orbitRadius.value
)
```

En este caso cambiamos el valor de la órbita de dicho planeta por el seleccionado en el spinner en cuestión.

```
on pbt_destruir picked aPlanet do
(
  delete aPlanet
)
```

Aquí, utilizamos un botón de selección para elegir un planeta y automáticamente se pasa como para parámetro para ser eliminado.

```

--SATELITES
on theColor changed new_col do
  selection.wirecolor = new_col

on pbt_pickPlanetSat picked aPlanet do
  (
    --Crear satellite que orbite al planeta seleccionado
    pPlanet = aPlanet
    pOrbit = aPlanet.pos.controller.path

    pPos = aPlanet.position
    crearSatelite(pOrbit)
  )

```

Entramos en el grupo de satélites. En el botón 'pbt\_pickPlanet Sat' (Elegir planeta para crear Sat) guardamos el planeta seleccionado junto con su radio, su órbita y posición. Entonces, al ser una operación de entidad, llamamos a la función crearSatelite creada con anterioridad y le pasamos por parámetro la posición del planeta. Es la siguiente:

```

-- FUNCION SATELITES
fn crearSatelite orb =
  (
    planet = sphere() -- Create the planet
    planet.radius = random 2.0 8.0
    orbit = circle radius:40
    orbit.pos = orb.pos

    --Definir la orbita del satellite en torno al planeta
    --Juntar la orbita del satellite a la orbita del planeta
    planet.pos.controller = Path_Constraint()
    planet.pos.controller.path = orbit
    orbit.pos.controller = Path_Constraint ()
    orbit.pos.controller.path = orb
  )

```

En esta función el plugin genera un satélite que orbitará alrededor del planetas que seleccionemos posteriormente de haber pulsado el botón



```

on disparo pressed do
(
    mediatMaterials[1].Diffuse = yellow
    mediatMaterials[1].ambient = yellow

    if isvalidnode $fuente do(
        particulas = #()
        segundos=60
        for j=1 to segundos
        do(
            numero=35
            for i = 1 to numero
            do(
                radio=6 * random(1-0.45) (1+0.45)

                posx=$fuente.pos.x
                posy=$fuente.pos.y
                posz=$fuente.pos.z

                bala = Sphere radius:radio segs:6 pos:[posx,posy,posz]
                bala.dir = $fuente.dir
                bala.wirecolor = yellow

                rotate bala (angleaxis 453 [random -1 1,random -1 1,random -1 1])
                toolMode.coordsys #local

                append particulas bala

                alcance = 3000 * random 0.7 1.3
                angulo = random 40 60

                dos_angulo = angulo * 2 as float
                alcax = cos(random 0 360)
                alcay = cos(random 0 360)

                v = ((alcance * 1 + posz*1))^0.5 as float
                vx = v* alcax as float
                vy = v* alcay as float
                vz = v * sin(angulo) as float

                frames = segundos

                fotograma_inicial = (sliderTime as integer)/ticksperframe + j-1
                fotograma_final = ceil(fotograma_inicial + frames as float)

                pos_inicial = bala.pos
                x_inicial =bala.pos.x
                y_inicial =bala.pos.y
                z_inicial =bala.pos.z

                t0 = 0

                animate on for fotograma = fotograma_inicial to fotograma_final - 1 by 1 do
                    at time fotograma
                    (
                        x= x_inicial + vx * t0 as float
                        y= y_inicial + vy * t0 as float
                        z= z_inicial + (vz * t0) - (0.5 * 1 * (t0 ^ 2)) as float

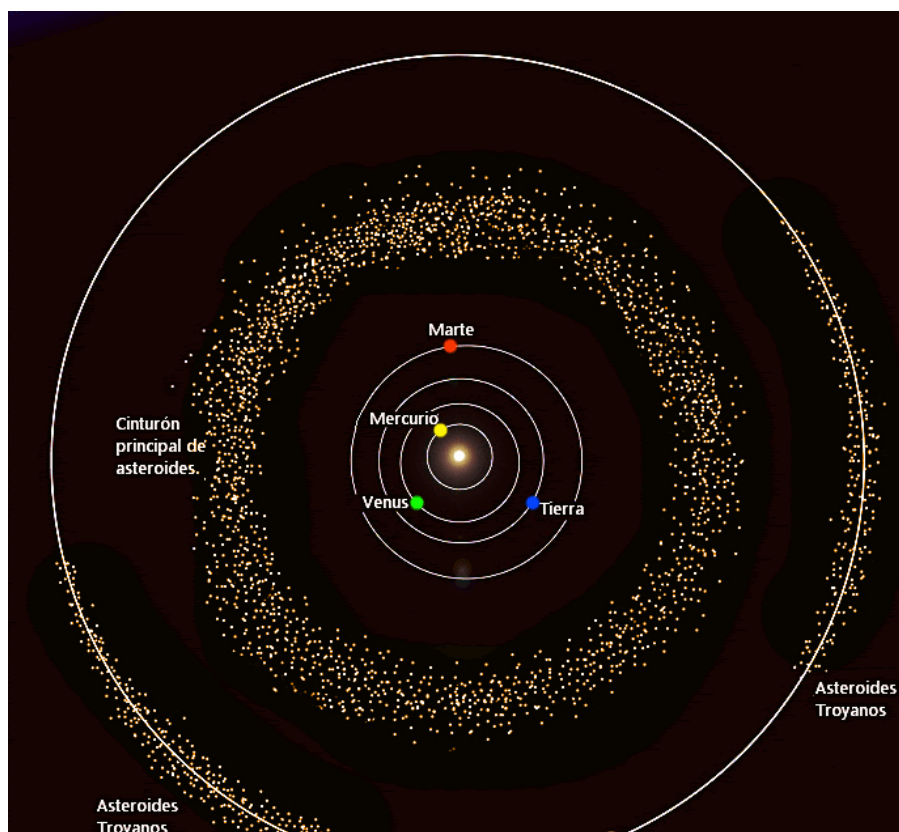
                        bala.pos = [x, y, z]
                        t0 = t0 +1
                    )
                )
            )
        )
    )
    --append todasparticulas particulas
)
)
)

```

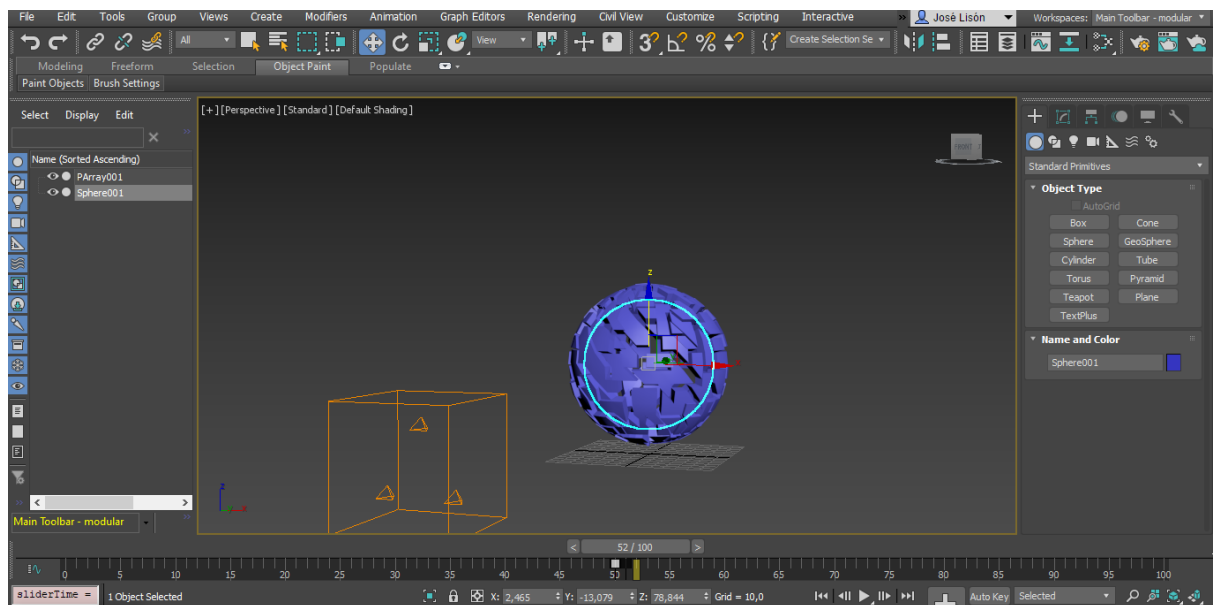
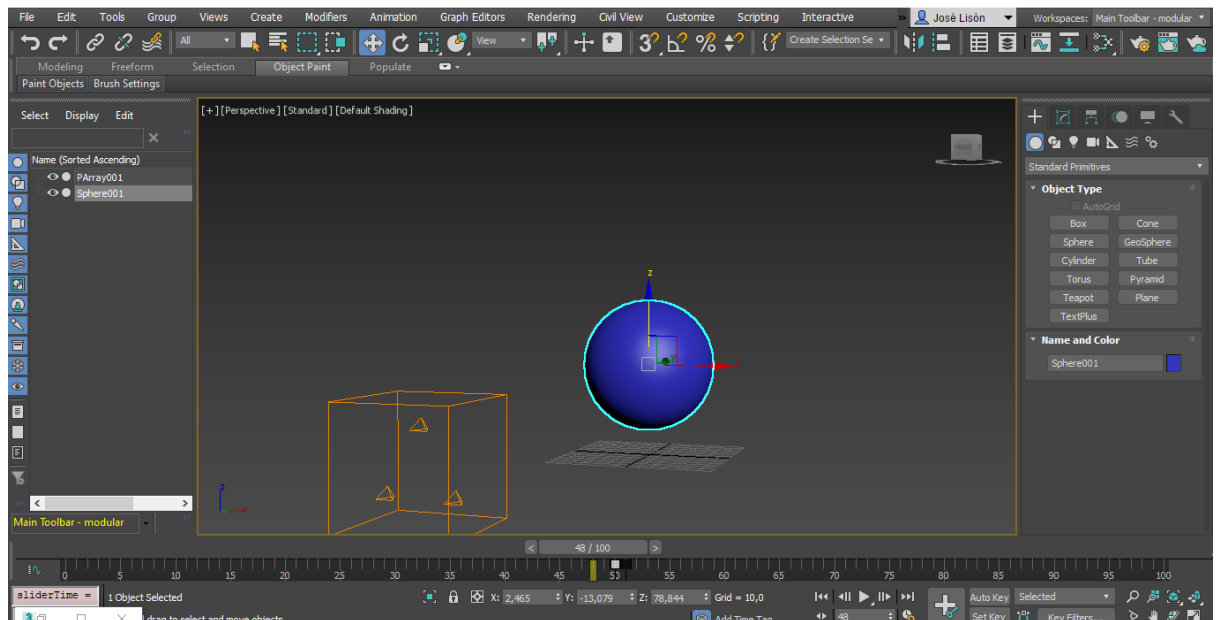
Esta función nos permitirá crear la animación de las partículas y activa la emisión de las partículas exactamente en la posición de creación del sol, ya que este es el punto de referencia en torno al cual gira todo el sistema con la intención de simular la infinita extensión de estrellas del universo. En esta función asignaremos todos los parámetros como la distancia de la explosión, el nº de partículas, los ángulos de los proyectiles y el nº de segundos de la animación.

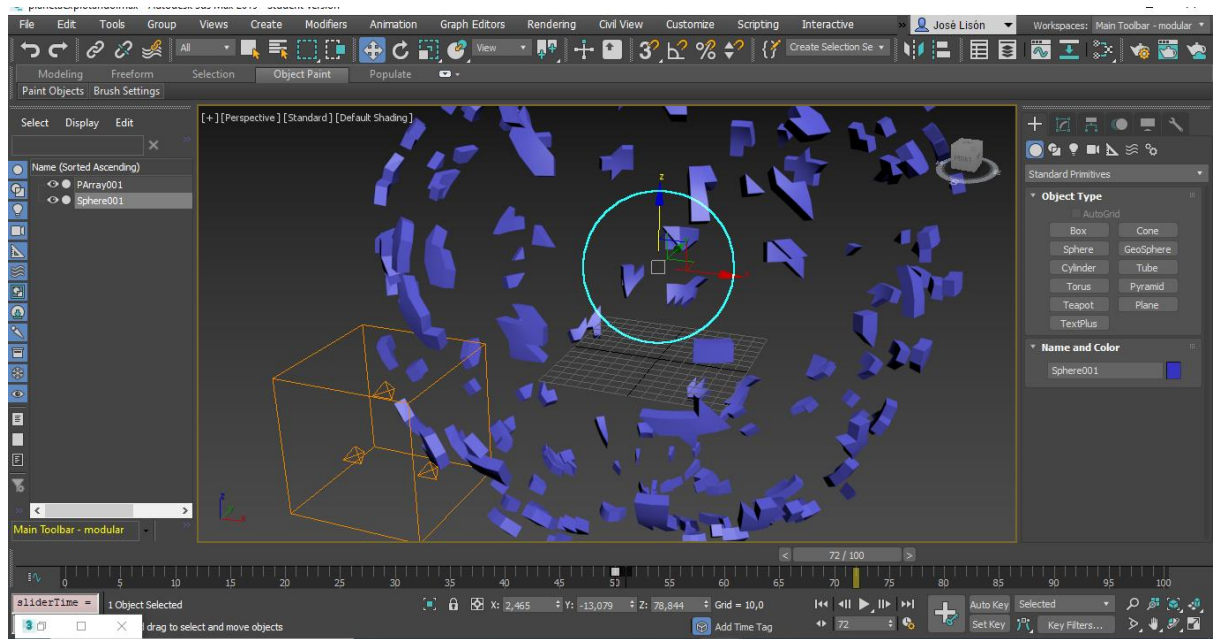
#### 4. Notas, comentarios y funcionalidades que nos hubiera gustado añadir pero que no lo hemos logrado.

##### Cinturón de asteroides



## Explosión del planeta





No conseguimos implementar una función explotar planeta que realizara lo conseguido en 3dsmax , intentado guiándonos con el listener.

### 4.1 Como nos hemos organizado y repartido las tareas.

Hemos trabajado todos juntos vía Discord, implementando cada uno en su parte en diferentes script, y posteriormente, cada final de sesión de trabajo, juntábamos todos los avances.

Antes de pasar al trabajo de programación con el código todos hemos buscado información y compartido con los compañeros para tener más clara una idea que íbamos hacer y que no, para así organizarnos mejor las tareas.

## 5. Video mostrando el funcionamiento del plugin.

<https://drive.google.com/file/d/1l8H5s172CUATwDt2m12NzTBV8hrmJZNB/view?usp=sharing>