



UNIVERSIDAD NACIONAL DE
SAN AGUSTÍN



FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS

CIENCIA DE LA COMPUTACIÓN

Laboratorio 1

ALUMNOS:

Pfuturi Huisa, Oscar David
Quispe Menor, Hermogenes
Quiñonez Lopez, Efrain German
Fernandez Mamani, Brayan Gino
Santos Apaza, Yordy Williams

DOCENTE:

MSc. Vicente Machaca Arceda

CURSO:

Computación Gráfica

20 de abril de 2021

Índice

1. Github	3
2. Funciones	3
3. Resultados	4

1. Github

- <https://github.com/oscar-pfuturi-h/Comp-Grafica>

2. Funciones

En el archivo helpers.py se agruparon las principales funciones, tenemos las funciones básicas cube, cone, sphere las cuales configuramos los tamaños de los objetos tanto en los 3 ejes(X, Y, Z)

```

1 def cube(size_x, size_y, size_z):
2     cube_shape = vtk.vtkCubeSource()
3     cube_shape.SetXLength(size_x)
4     cube_shape.SetYLength(size_y)
5     cube_shape.SetZLength(size_z)
6     cube_shape.Update()
7     mapp_object = vtk.vtkPolyDataMapper()
8     mapp_object.SetInputData(cube_shape.GetOutput())
9     return mapp_object
10
11 def cone(radius, height, resolution):
12     cone_shape = vtk.vtkConeSource()
13     ...
14     return mapp_object
15
16 def sphere(radius):
17     sphere_shape = vtk.vtkSphereSource()
18     ...
19     return mapp_object

```

La función actor es la que se encarga de recibir los mappers de los objetos básicos y luego dependiendo de los diferentes parámetros se empieza a personalizar, como ejemplo posición en el plano, ángulos de rotación, color o texturas, al final nos devolverá objetos tipo actor para poder ser agregados al render.

```

1 def actor(mapper, xyz, angle, rgb, texture_name):
2     actor = vtk.vtkActor()
3     #Cargar Texturas
4     if texture_name is not None:
5         reader = vtk.vtkJPEGReader()
6         reader.SetFileName(texture_name)
7         texture = vtk.vtkTexture()
8         if vtk.VTK_MAJOR_VERSION <= 5:
9             texture.SetInput(reader.GetOutput())
10        else:
11            texture.SetInputConnection(reader.GetOutputPort())
12        actor.SetTexture(texture)
13    if rgb is not None:
14        actor.GetProperty().SetColor(transformRGBRange(rgb))
15    if angle[0] is not None:
16        actor.RotateX(angle[0])

```

```

17     if angle[1] is not None:
18         actor.RotateY(angle[1])
19     if angle[2] is not None:
20         actor.RotateZ(angle[2])
21     actor.SetMapper(mapper)
22     actor.SetPosition(xyz)
23     return actor

```

En el caso de que requiera que se creen varios actores, se crearon funciones auxiliares como por ejemplo la creación de vallas.

```

1 def crearVallas(cubos,color,angle,xyz,cantidad,tamx,tamz):
2     vallas = []
3     tam1 = tamx
4     tam2 = tamz
5     for i in range(cantidad):
6         cActor1 = createActor(cubos[0],color,angle,[xyz[0]-tamx,xyz
7 [1],xyz[2]-tamz])
8         cActor2 = createActor(cubos[1],color,angle,[xyz[0]-tamx,xyz
9 [1],xyz[2]-tamz])
10        cActor3 = createActor(cubos[2],color,angle,[xyz[0]-tamx,xyz
11 [1],xyz[2]-tamz])
12        cActor4 = createActor(cubos[3],color,angle,[xyz[0]-tamx,xyz
13 [1],xyz[2]-tamz])
14        tamx = tamx+tam1
15        tamz = tamz+tam2
16        vallas.append(cActor1)
17        vallas.append(cActor2)
18        vallas.append(cActor3)
19        vallas.append(cActor4)
20    return vallas

```

Se creo un función auxiliar en el caso de que se quieran agregar muchos Actores al renderer.

```

1 def addActores(renderer,Actores):
2     for i in Actores:
3         renderer.AddActor(i)

```

```

1 vallas=crearVallas(cubos,[255, 225, 53],[0,90,None],[36,0,78],9,8,0)
2 addActores(renderer,vallas)

```

3. Resultados

En el trabajo realizado se creo una granja. Para armar objetos como molino o animales se siguió la siguiente estructura, la cual nos permite configurar la mayoría de cosas en una simple linea de código.

```

1 #-ACTORS(mapper/[position]/[angle]/[color]/texture_name)
2 act_techo=actor(techo,[20,14,20],[None,None,90],[219, 172, 121],None)
3 act_body=actor(body,[20,8,20],[None,None,90],[209, 204, 190],None)
4 act_aspa1=actor(asma,[20,16,26],[None,None,None],[163, 130, 93],None)

```

```
5 act_aspa2=actor(aspera, [20,16,26], [None,None,90], [163, 130, 93],None)  
6 act_estaca=actor(estaca,[20,16,23],[None,None,90],[163, 130, 93],None)
```



Figura 1: Patos y caballo

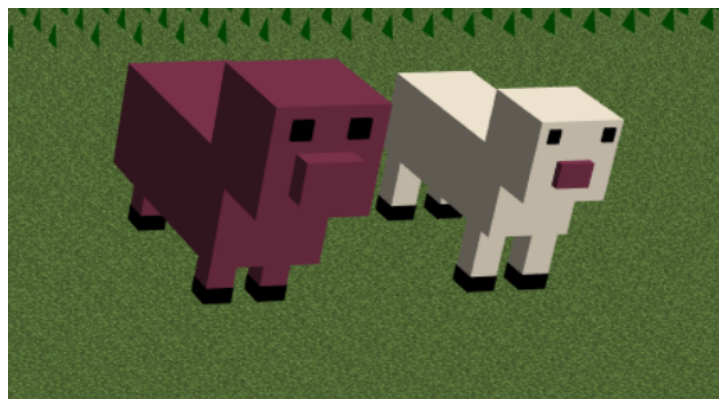


Figura 2: cerdo y oveja



Figura 3: Molino



Figura 4: Vallas

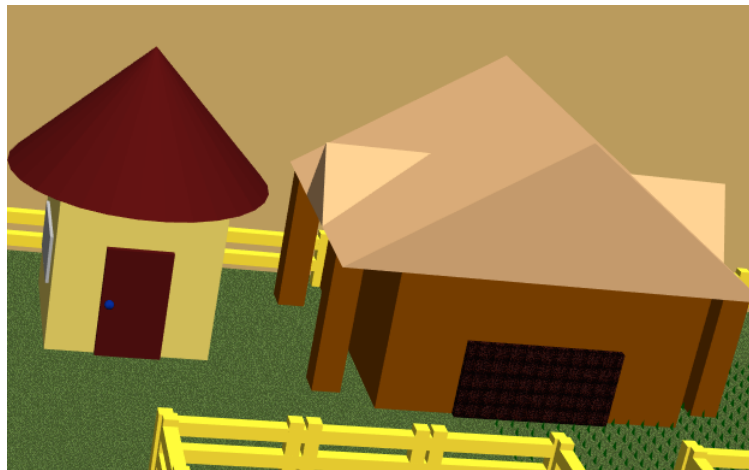


Figura 5: Casa y granero

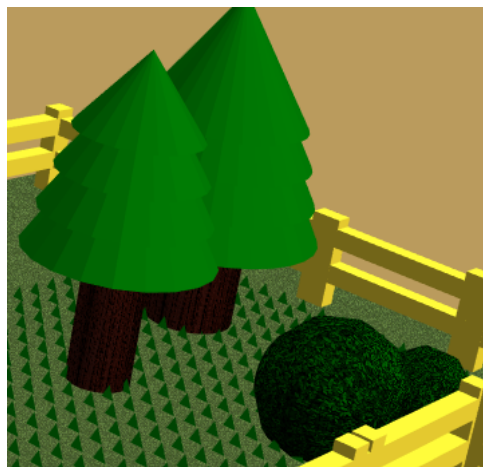


Figura 6: Vegetación