Documentación de la clase Land

Con esta clase definimos nuestra "tierra", esta consta de **sociedades**, **características**, **dependencias** e **influencias**.

In [ ]:
```
import os
from sys import path
path.append(os.path.abspath(os.path.join('',os.pardir)))
from Simulation.land import Land
from Simulation.species import Species

terreno = Land([0,0])

(terreno.entities, terreno.characteristic, terreno.characteristic_dependences)
```

Out[ ]:
```
({'': <Simulation.land.Land at 0x22ddca82e00>}, {}, [])
```

1. Sociedades

En un terreno puede habitar varias sociedades de disímiles especies, estas son guardadas en un diccionario **entities** con llaves string del nombre de la sociedad(ejemplo "cubanos") y el valor la instancia de la sociedad.

Estas sociedades se pueden gestionar con los métodos:

- **Add_Society**
- **Delete_Society**

In [ ]:
```
#Add Society tiene por entrada el nombre de la sociedad y la especie
humanidad = Species("Humano")
terreno.Add_Society("cubano", humanidad)
terreno.Add_Society("ruso", humanidad)
terreno.Add_Society("español", humanidad)
#Delete society recibe solo el nombre de la especie
terreno.Delete_Society("ruso")

terreno.entities
```

Out[ ]:
```
{'': <Simulation.land.Land at 0x22ddca82e00>,
 'cubano': <Simulation.society.Society at 0x22ddca81960>,
 'español': <Simulation.society.Society at 0x22ddc9dce50>}
```

diccionario entities

*Además es válido aclarar que el diccionario **entities** guarda como primer elemento al mismo terreno, por una cuestión de comodidad a la hora de trabajar con las dependencias que podía afectar a la misma land o a las sociedades.*

Adicionalmente se trabaja con este diccionario entitites con las funciones

- **Get_Entities_Characteristic_value**
- **Change_Entities_Characteristic**
- **Update_Entities_Characteristic**
- **Delete_Entities_Characteristic**
- **Set_Default_Entities_Characteristic**

Que cada uno de estos métidos llama a ejecutar una función en la entidad que le pasen por argumento, por supuesto como land y sociedades tienen un método del mismo nombre entonces solo basta llamar a este método sin necesiadad de validar con que clase estamos trabajando.

2. Características

Un terreno posee una lista de **características** propias como pueden ser recursos actuales, capacidad, fertilidad, temperatura, altura, etc.

La misma es una instancia de la clase characteristic y se puede interactuar con ella usando los métodos *Change_Characteristic* el que recibe como entrada el nombre de la característica y su valor, si existe la modifica sino se agrega, y el método *Delete_Characteristic* que recibe de entrada el nombre de la característica que quieres borrar y la elimina, y podemos modificar solo el valor de la característica con el método *Change_Characteristic_Value* así como acceder al mismo con el método *Get_Characteristic_Value*. Las características pueden tomar valor numérico o rangos.

In [ ]:
```
terreno.Change_Characteristic('actual_resources', 500)
terreno.Change_Characteristic('fertility', 3)
terreno.Delete_Characteristic('fertility')
terreno.Change_Characteristic('fertility', [2,8])
terreno.Change_Characteristic('fertility', [25,60])
terreno.Change_Characteristic('temperature', 25)
terreno.Delete_Characteristic('fertility')
```

```
terreno.characteristic
```

Out[ ]:
```
{'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
 'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>}
```

Además se trabaja en el proyecto con la lista de entidades al usar dependencias que al realizarse se manda a ejecutar las funciones anteriormente mencionadas

In [ ]:
```
terreno.Change_Entities_Characteristic("", "fertility", [3,9])

(terreno.characteristic, terreno.Get_Entities_Characteristic_value('', "fertility"))
```

Out[ ]:
```
({'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 [3, 9])
```

3. Dependences e influences

Un terreno puede además poseer **dependencias** o **influencias** entre las características para expresar por ejemplo como dependencias de la natalidad en la población al pasar un día, o como influye el avance médico en la esperanza de vida Estas dependencias e influencias son representadas con una instancia de dependence.

Cada dependencia tiene por atributo los elementos elementos: posición del terreno, entidad y característica de *a*, posición del terreno, entidad y característica de *b*, índice de dependencia *c*, funciones de calcular dependencia *plus* y *mult*. Al pasar un dia se dice que ocurre **a -> b * c**, que se traduce a: *b* += *a* c*, en este caso las posiciones del terreno son las del propio terreno en que se encuentra las dependencias, y la función de suma y multiplicaciones las cambiammos con las que tenga la dependencia
Si entre las características hay una que tiene de valor un rango entonces se haya random un valor dentro del rango con una función de distribución que guarda la misma característica

Para el manejo de las dependencias se implemento tres métodos: *Add_Dependences* que recibe de entrada (a,b,c) y añade la dependencia a la lista, *Change_Dependences_Value* que recibe de entrada (a,b,c) y cambia el valor de c de una dependencia definida con anterioridad, *Delete_Dependences* que recibe de entrada (a,b) y elimina la dependencia de la lista. Adicionalmente uno como auxiliar **Delete_All_Specific_Dependence** para cuando eliminamos una caractgerística pues borramos todas las dependencias que contengan dicha caracterísitica. Métodos similares tiene las influencias que trabajan similares a las dependencias con la diferencias de que las mismas toman en cuenta no el valor de a sino el valor del cambio que tuvo en un día.

In [ ]:
```
terreno.Add_Dependence("",'temperature',"", 'fertility', 3)
terreno.Add_Dependence("",'actual_resources', "",'actual_resources', 0.05)
terreno.Add_Dependence("",'fertility', "",'actual_resources', 5.5)
terreno.Add_Dependence("",'actual_resources', "",'fertility', 0.001)
terreno.Change_Dependences_Value("",'temperature', "",'fertility', -0.1)
terreno.Delete_Dependence("",'actual_resources',"", 'fertility')
terreno.Add_Influences("",'temperature',"", 'fertility', 3)
terreno.Add_Influences("",'actual_resources', "",'actual_resources', 0.05)
terreno.Add_Influences("",'fertility', "",'actual_resources', 5.5)
terreno.Add_Influences("",'actual_resources', "",'fertility', 0.001)
terreno.Change_Influences_Value("",'temperature', "",'fertility', -0.1)
terreno.Delete_Influences("",'actual_resources',"", 'fertility')

(terreno.characteristic_dependences, terreno.characteristic_influences)
```

Out[ ]:
```
([<Simulation.dependence.Dependence at 0x22ddc9de3e0>,
  <Simulation.dependence.Dependence at 0x22ddc9de8f0>,
  <Simulation.dependence.Dependence at 0x22ddc9de3b0>],
 [<Simulation.dependence.Dependence at 0x22ddc9de8c0>,
  <Simulation.dependence.Dependence at 0x22ddc9de380>,
  <Simulation.dependence.Dependence at 0x22ddc9dd510>])
```

Simulación de un día

Para esto existe un método llamado *Move_One_Day* que no recibe argumentos, simula un día de acuerdo a las incidencias de las **dependencias** e **influencias** en las **características**. Primero ejecuta todas las dependencias guardando en una lista los cambios, luego ejecuta las influencias apoyándose en dicha lista y por último actualiza los valores de las carácteristicas.

In [ ]:
```
time_line = [terreno.characteristic.copy()]
for i in range(10):
    # Avanza un día la simulación
    terreno.Move_One_Day()
    time_line.append(terreno.characteristic.copy())

time_line
```

Out[ ]:

```
[{'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 {'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 {'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 {'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 {'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 {'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 {'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 {'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 {'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 {'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>},
 {'actual_resources': <Simulation.characteristic.Characteristic at 0x22ddca827a0>,
  'temperature': <Simulation.characteristic.Characteristic at 0x22ddca82890>,
  'fertility': <Simulation.characteristic.Characteristic at 0x22ddca81e70>}]
```