

Documentación de la clase Simulation

Con esta clase definimos nuestra "simulación", esta consta de **species**, **terrenos** o un mapa, **inter-dependencias**. AL inicializarse una simulación se crea un mapa que no es más que una lista de listas de terrenos, a este se le puede aplicar el método **redimention** que recibe de entrada las nuevas dimensiones del mapa

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

simulacion = Simulation(3,3)
simulacion.Re_Dimension_Map(2,2)

(simulacion.actual_species, simulacion.inter_dependencies, simulacion.map)

Out[ ]: ({},
[],
[[<Simulation.land.Land at 0x18cac8cb7f0>,
<Simulation.land.Land at 0x18cac6389d0>],
[<Simulation.land.Land at 0x18cac638d90>,
<Simulation.land.Land at 0x18cac638df0>]])
```

Species

Se tiene guardado un diccionario de especies donde la llave es un string del nombre de la especie y el valor es la instancia.

Para trabajar con las mismas existen los métodos:

- **Add_Species**: Crea y añade una especie a la lista species
- **Delete_Species**: Elimina una especie de la lista species
- **Change_Species_Characteristic**: Añade o cambia el valor de una característica en una especie
- **Delete_Species_Characteristic**: elimina una característica de una especie

```
In [ ]: simulacion.Add_Species('Humano')
simulacion.Add_Species('Marciano')
simulacion.Delete_Species('Marciano')
```

Map

Map, como anteriormente fue mencionado, no es más que una matriz de terrenos, para acceder y trabajar con los terrenos que este contiene se implementaron los métodos:

- **Change_Land_Characteristic**
- **Delete_Land_Characteristic**
- **Update_Land_Characteristic_Value**
- **Add_Land_Dependences**
- **Delete_Land_Dependences**
- **Change_Land_Dependences_Value**
- **Add_Land_Influences**
- **Delete_Land_Influences**
- **Change_Land_Influences_Value**

y para trabajar con las sociedades de los terrenos del mapa se implementan los métodos:

- **Change_Society_Characteristic**
- **Delete_Society_Characteristic**
- **Update_Society_Characteristic_Value**
- **Add_Society**
- **Delete_Society** Para cada uno de estos métodos se reciben parámetros necesarios para ejecutar métodos de igual nombre en los terrenos o sociedades y se ejecutan

Por último se tienen las interdependencias que nos son más que dependencias entre dos entidades de terrenos diferentes, para el trabajo con las mismas se puede usar:

- **Add_Inter_Dependences**: Recibe posición, entidad y características de las dos características relacionadas seguido de un valor c y las funciones de suma y multiplicación y añade la dependencia a la lista inter_dependencies
- **Delete_Inter_Dependences**: Recibe posición, entidad y características de las dos características relacionadas y elimina la dependencia a la lista inter_dependencies
- **Change_Inter_Dependences_Value**: Recibe posición, entidad y características de las dos características relacionadas y cambia su

valor.

```
In [ ]: simulacion.Add_Society(0,0,"habanero","Humano")
simulacion.Add_Society(1,0,"pinareño","Humano")
simulacion.Add_Inter_Dependence([0,0],"habanero","Poblacion",[1,0],"pinareño","Poblacion",10)
```

Simulación de un día

Para avanzar un día en la simulación se tiene el método **Move_One_Day_All** que lo que hace es mover un día en cada terreno, seguidamente por cada interdependencia.

Para ello no se hace más que por cada *dependencia*, *influencia*, *interdependencia* ejecuta apollándonos en las funciones de distribución de cada característica, y las funciones de suma y multiplicación de cada dependencia para calcular los resultados.

```
In [ ]: simulacion.Move_One_Day_All()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js