

Simulación

Ayudantia 7



¿Por qué?

Simular permite modelar un sistema y realizar experimentos sobre él.

El objetivo es validar y evaluar el comportamiento del modelo, además de comprender el comportamiento del sistema.

Simulación Síncrona

MIENTRAS el tiempo simulación no termine
aumentar tiempo en una unidad
si ocurren eventos en este intervalo de tiempo:
simular los eventos

- Ejecución lenta
- Mayoría de incrementos no cambia el estado del sistema
- Las verificaciones generan pérdida de tiempo en la CPU

Simulación DES

MIENTRAS la lista de eventos no esté vacía y el tiempo de simulación no termine
tomar un evento del principio de la lista de eventos
avanzar tiempo de simulación al tiempo del evento
simular el evento



TIPS!!!

Identificar clases, eventos y variables + Código verboso

Como siempre, un modelo apropiado nos facilitará la tarea a la hora de programar y también a la hora de leer y compartir nuestro código. Por esto, tenemos que identificar cuando estamos ante una clase, cuales son los eventos a simular y cuales son los valores a utilizar. Todo esto acompañado de una asignación apropiada de nombres de variables y de funciones, harán nuestro código muy entendible.

La primera sucursal creada del **Banco Seguritas** cuenta solo con 2 **cajas** que están disponibles para todo público. Cada caja tiene su propia cola. Los **clientes** que llegan se colocan en la cola más corta y ante iguales largos de cola prefieren a la caja 1. Cuando se produce un avance de cola, el último cliente de la otra cola verifica si puede mejorar de posición al cambiarse y si así fuera lo hace en ese mismo instante. Los clientes llegan con un tiempo entre llegadas **Uniforme(1, 3)** minutos a la sucursal. Cada cajera se demora un tiempo **Uniforme(1, 10)** minutos en atender a un cliente.

Eventos:

1. Llega cliente al sistema
2. Sale cliente de la caja 1
3. Sale cliente de la caja 2

TIPS!!!

import random

Utilizado para determinar los distintos eventos que ocurren con tasas de llegada aleatoria. La librería random tiene funcionalidad para simular distintas distribuciones.

#Escoger un elemento aleatorio de una secuencia (ej: lista)
`random.choice(secuencia)`

#Para eventos que distribuyen uniforme
`random.randint(a, b)`

#Para eventos que distribuyen exponencial (lambd
ocurrencias cada una unidad de tiempo)
`random.expovariate(lambd)`

TIPS!!!

Properties para facilitar simulación y determinar estados:

Suponga que hay grupos de personas, y dichos grupos actuarán según el tiempo más lento de alguno de sus participantes (Una cadena es tan fuerte como el eslabón más débil :())

```
class Eslabon:
```

```
    def __init__(self):  
        self.next_event = randint(1, 10)
```

```
class Cadena:
```

```
    @property  
    def next_event(self):  
        maximo = max(self.eslabones, key=lambda e: e.next_event)  
        return maximo.next_event
```

```
class Simulacion:
```

```
    @property  
    def next_event(self):  
        minimo = min(self.cadenas, key=lambda c: c.next_event)  
        return minimo.next_event
```

Código



Ejercicio

Desde el ministerio de transporte, se te ha pedido que implementes un programa que modele y simule el funcionamiento de sus autopistas, pues quiere ver en qué autopistas se debe licitar nuevas estaciones de servicio y gasolineras. Para esto, te dan las siguientes indicaciones: "El programa debe ser capaz de simular el funcionamiento de distintas rutas, en donde cada ruta tendrá un largo entre 350 y 500km (que es la distancia que posee dicha ruta para ser completada) y una probabilidad de que un conductor en auto, motocicleta o camión llegue a la ruta. Cada conductor, puede conducir a una velocidad aleatoria entre 30 y 40km/unidad de tiempo. Además, cada vehículo gasta entre 0.1 y 0.07 litros por cada kilómetro recorrido. Además de esto, los autos no tienen estanques infinitos, por lo que al momento de ingresar a la ruta, se tiene que su estanque tiene una cantidad aleatoria entre 40 y 50 litros. Como no hay estaciones de servicio, un auto puede llegar al final de la ruta, o quedarse en pana en la mitad del trayecto. Una vez que la simulación termine, se debe informar por cada ruta, cuántos autos se quedaron sin bencina, y cuántos lograron terminar la ruta exitosamente".

Se debe hacer una simulación síncrona, y otra bajo eventos discretos.