



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada (II/2015)

Tarea 4

1. Objetivos

- Aplicar conocimientos de Simulación de Eventos Discretos y SimPy3 para resolver un problema.

2. Introducción

El alcalde del pueblo de Progralandia se ha dado cuenta del aumento de emergencias en los últimos años. Dentro del pueblo hay cuatro ubicaciones posibles para instalar uno de cada uno de los siguientes servicios de emergencias: Estación de policía, Hospital y Estación de bomberos.

Sin embargo, no sabe cómo distribuirlos. Es por esto que ha decidido pedirle a los alumnos del curso de Programación Avanzada que determinen en cuál de estos lugares se debe hacer cada instalación. Tu labor es simular los resultados de las ubicaciones de los edificios y determinar cual combinación es la óptima.

3. Especificaciones

Crear una simulación del tránsito y actividades del pueblo de Progralandia. Para esto tendrás tantos escenarios posibles como permutaciones que puedas hacer de los tres servicios de emergencia en las ubicaciones disponibles. A modo de ayuda, se creó una interfaz gráfica que te permitirá visualizar los estados de la simulación.

3.1. Eventos:

A continuación se presentan los eventos que se desean incluir dentro de la simulación:

3.1.1. Incendios:

Existen distintos tipos de construcciones y materiales donde cada construcción está completamente compuesta de un solo material. La probabilidad de incendio de una construcción depende del dicho material. Los edificios de servicios no pueden incendiarse ya que tienen sistemas anti-incendios.

Cuando un edificio se incendia se avisa inmediatamente a los bomberos, quienes van en un carro con sirena al lugar del incidente. Los vehículos que se encuentren en el camino del carro de bomberos deben reordenarse de tal manera de dejar espacio para que pase el carro. Cuando llega al incendio se estaciona en la calle más cercana y bloquea la pista donde se estaciona hasta apagar el fuego. El tiempo que demora para apagar el fuego depende del material. Luego de apagado el incendio va de vuelta a la estación sin la sirena (es decir, a la vuelta se comporta como un automóvil normal).

3.1.2. Robos:

La seguridad de las casas del pueblo puede ser vulnerada por ladrones que intenten robar. La probabilidad de que una casa sea robada depende de su cercanía con la estación de policía.

Cuando los ladrones entran a robar a una casa se avisa inmediatamente a la estación de policía, que envía una patrulla al lugar del robo. En el camino de ida la patrulla va con su sirena encendida, por lo que los otros vehículos la dejan pasar. Si el ladrón se va antes de que llegue la patrulla significa que escapó, de lo contrario significa que el policía atrapó al ladrón. En ambos casos la patrulla vuelve a la estación de policía sin la sirena encendida.

3.1.3. Enfermos:

Dentro de las casas pueden acontecer emergencias médicas, por lo que al ocurrir alguna es necesario que asista una ambulancia para buscar al enfermo y llevarlo al hospital. Las ambulancias van y vuelven con la sirena encendida, por lo que los demás vehículos se reordenan para que pase.

3.1.4. Taxis:

Los taxis recogen y dejan pasajeros dentro de la ciudad. Para ambos procesos tienen que parar en las calles durante un tiempo determinado. Los taxis vacíos encuentran un pasajero en un tiempo aleatorio exponencial con $\lambda = \frac{1}{40}$.

Cuando un pasajero entra al taxi se elige una ubicación aleatoria de la ciudad como destino. Al momento de crear un nuevo auto existe 20% de probabilidad de que sea un taxi.

Consideraciones:

- Al localizar los tres servicios en el mapa, queda un espacio vacío. En este espacio no pueden ocurrir eventos.
- No pueden haber dos eventos simultáneos en una misma propiedad.
- Pueden haber hasta tres vehículos de emergencia de cada tipo como máximo. Si hay más emergencias, se debe esperar a que se desocupen los vehículos.

3.2. Tránsito:

Las calles del pueblo tienen dos pistas pero solo una dirección, por lo que los autos se pueden adelantar. Todas las esquinas tienen semáforos que solo funcionan a dos tiempos (*rojo* y *verde*). Al llegar a una esquina, los autos eligen la dirección hacia donde continuar de manera aleatoria y solo avanzan si se tiene luz verde y un espacio vacío al otro lado del semáforo, de manera de no quedar en la mitad de la intersección sin poder avanzar. Para simplificar el funcionamiento de los semáforos se puede asumir que se puede doblar en segunda fila.

Las calles de la ciudad tienen ciertas entradas y salidas por las que los autos pueden entrar o salir de la ciudad. Para que el número de autos se mantenga más o menos constante tiene que asegurarse de que el flujo de salida y entrada sea similar. Además el número de autos no puede pasar cierto límite especificado más adelante.

Los vehículos de emergencia pueden pasar los semáforos en roja o verde y los autos le dan la preferencia. Además, taxis, bomberos, patrullas y ambulancias usan el camino más corto al objetivo sin ir contra el tránsito.

Cuando un vehículo con sirena circula por la ciudad, debe hacer que los vehículos de adelante dejen el espacio desocupando una pista. Si dos vehículos de emergencia se encuentran en una esquina, se debe dar preferencia al que viene con sirena. Si ambos van con sirena, el orden de prioridad es calle que sube, calle que baja, calle hacia la derecha, calle hacia la izquierda.

4. Valores:

4.1. Casas:

- Hay cuatro tipos de materiales para las casas: *Madera*, *Ladrillo*, *Hormigón* y *Metal*. El tiempo transcurrido entre dos incendios distribuye *exponencial*(1/10) en horas. Recuerde que puede usar `random.expovariate(lambda)` para obtener una distribución exponencial. La localización del incendio es aleatoria, pero cada construcción tiene distinto peso probabilístico dependiendo del material. Cuando ocurre un incendio, éste puede ocurrir en cualquier casa.

$$\text{Probabilidad_incendio} = \text{peso_material_casa} / \text{suma_pesos_todas_las_casas}.$$

Los pesos de los materiales son:

- Madera: 10
- Ladrillo: 7
- Hormigón: 4
- Metal: 2

El tiempo que demoran los bomberos en apagar el incendio según el material distribuye uniforme entre:

- Madera: 30 y 120 minutos
 - Ladrillo: 40 y 100 minutos
 - Hormigón: 60 y 80 minutos
 - Metal: 30 y 40 minutos
- El tiempo entre robos distribuye *exponencial*(1/4) horas, y la casa robada es aleatoria. La probabilidad de que el robo haya sido en una casa determinada depende de la distancia a la comisaría. El peso probabilístico del robo en una casa es $10 + x + y$ donde x e y son la distancia en dirección x e y de la comisaría. Por ejemplo, si una casa está tres cuadrados más arriba que la comisaría y dos cuadrados a la derecha, entonces $x = 2$ e $y = 3$ y el peso es 15.
 - El tiempo que demora el ladrón en robar la casa depende de cada casa y está especificado en el archivo con el mapa.
 - El tiempo que transcurre entre la aparición de dos personas enfermas que necesiten ambulancia distribuye *exponencial*(1/2) en horas, y la localización es aleatoria con igual probabilidad en todas las casas.

4.2. Vehículos:

- Los autos tienen velocidad de *cuadrados por segundo*. Si el auto del cuadrado siguiente va más lento, su velocidad se iguala para no chocar. La velocidad del auto se determina al momento de aparecer y distribuye uniforme entre 0,5 y 1 cuadrados por segundo.
- Los autos de emergencia circulan a 1 cuadrado por segundo con sirena y a 0,5 cuadrados por segundo sin sirena.

- Los autos prefieren la pista derecha y solo se cambian a la pista izquierda si la derecha está ocupada por un auto que va más lento que ellos. El cambio de pista demora 0,5 segundos, en los cuales deja de avanzar.
- Los taxistas tardan entre 5 y 15 segundos en recoger un pasajero y entre 10 y 20 segundos en dejarlo.
- Los semáforos tardan 20 segundos por tiempo.
- El límite de vehículos en las calles de la ciudad es de la mitad de los espacios de calles.

5. Estadísticas:

Luego de cada simulación se tienen que generar estadísticas sobre el funcionamiento de los servicios:

- Bomberos: Se tiene que calcular el tiempo promedio de los incendios, es decir, desde que comienza hasta que es apagado.
- Policías: Se tiene que calcular la cantidad de robos en los que el ladrón escapó y en los que fue atrapado.
- Ambulancias: Se tiene que calcular el tiempo promedio de rescate de los enfermos, que incluye la ida a la casa y la vuelta al hospital.

Para calcular las estadísticas anteriores, corra 50 réplicas por escenario en un horizonte de 14 días, calcule la media e intervalo de confianza al 95 % para cada variable de desempeño anteriormente mencionada.

A partir de las medias calculadas determine cuál es la mejor distribución espacial de los edificios de emergencia. El criterio que use para determinarlo a partir de las estadísticas se debe especificar en el `README.md`.

6. Reconstrucción:

El alcalde de Progralandia se ha dado cuenta de toda la destrucción que ha quedado en la ciudad luego de la simulación y él considera que es buena idea agregar un cuarto servicio: una compañía de reconstrucción.

Como ya determinó la mejor posición para la Estación de bomberos, Estación de policía y Hospital, se le solicita que en el espacio que quedó libre instale la constructora y que simule la reconstrucción de la ciudad. Debe determinar cuánto tiempo tomará esta reconstrucción.

Consideraciones:

- La simulación de la reconstrucción debe ser hecha obligatoriamente con `SimPy3`.
- Se debe considerar que el tiempo que demora la constructora en llegar desde la ubicación a la casa es equivalente a la de un vehículo normal.
- Se considera a una casa destruida si es que esta sufrió un incendio en la última simulación realizada con la distribución de edificios elegida.
- La casa se debe reconstruir del mismo material original.
- El orden de prioridad para la reconstrucción será asignado según la cercanía con la compañía.
- Luego de cada reconstrucción el vehículo deberá volver al centro de operaciones de la compañía para obtener más materiales. Luego de eso puede seguir arreglando otras casas.
- La constructora posee dos vehículos con equipos independientes. Cada uno puede hacer reparaciones por separado.

- El tiempo que toma la construcción de una casa varía según el material:
 - Madera: 40 a 60 minutos
 - Ladrillo: 50 a 80 minutos
 - Hormigón: 60 a 100 minutos
 - Metal: 60 a 200 minutos

7. Interfaz Gráfica:

Adicionalmente, tiene a su disposición el módulo `gui` con el que podrá ver a través de una interfaz gráfica el comportamiento de su simulación.

Dentro del módulo se encuentra definida la clase `GrillaSimulacion`, que consiste en una ventana con una grilla de 20 celdas de ancho por 20 de alto. Es necesario tener instalado el módulo `PyQt4` para poder ejecutar la interfaz.

Para ver ejemplos de cómo utilizar la interfaz gráfica puedes revisar el código de muestra en `main.py`. También puedes revisar directamente el código de la interfaz en `gui/gui.py`. Además está el archivo `ejemplo_gui.py` con un ejemplo comentado del uso de la interfaz gráfica.

A continuación se señala un resumen de los métodos que expone la clase `GrillaSimulacion`:

- `self.agregar_ambulancia(x, y, theta, reflection)`
- `self.agregar_enfermo(x, y)`
- `self.agregar_auto(x, y, theta, reflection)`
- `self.agregar_cuartel_bomberos(x, y)`
- `self.agregar_carro_bomba(x, y, theta, reflection)`
- `self.agregar_incendio(x, y)`
- `self.agregar_casa(x, y)`
- `self.agregar_hospital(x, y)`
- `self.agregar_robo(x, y)`
- `self.agregar_comisaria(x, y)`
- `self.agregar_patrulla(x, y, theta, reflection)`
- `self.agregar_taxi(x, y, theta, reflection)`
- `self.agregar_calle(x, y)`
- `self.quitar_imagen(x, y)`

`GrillaSimulacion` cuenta con el atributo `tiempo_intervalo`, que por defecto vale 0, y define los intervalos de tiempo con los que la interfaz gráfica actualizará los eventos de la simulación. Se reitera que para ver un ejemplo comentado del uso de la interfaz, puedes revisar el archivo `ejemplo_gui.py`.

Al llamar cualquiera de los métodos anteriormente señalados, se actualizará la grilla de simulación, agregando el ícono asociado a ese método en la posición `(x, y)`. La grilla trabaja con un sistema de coordenadas

con origen en la esquina superior izquierda. La celda de la esquina superior izquierda tiene coordenadas (1, 1).

Los métodos asociados a vehículos cuentan con dos parámetros adicionales, **theta**, que permite indicar la rotación en contra del sentido del reloj (medida en grados) del ícono y **reflection** que toma valores booleanos y permite aplicar una reflexión en el ícono (efecto espejo) cuando vale **True**. Esto es útil cuando el elemento viaja de derecha a izquierda.

Al ejecutar un método sobre una celda ocupada con otro ícono, elimina el ícono antiguo y es reemplazado por el del nuevo método invocado. Además cuenta con el método **self.quitar_imagen(x, y)** que remueve el ícono de la celda (x, y) dejándola vacía.

8. Archivo mapa:

El archivo **mapa.txt** contiene el mapa de la ciudad con todas sus calles y casas. En la primera línea aparece la dimensión del mapa de la forma **ixj**. En las líneas siguientes están las calles, casas y espacios de construcción con el formato '**i,j objeto**', donde **objeto** puede ser **calle <dirección>**, **vacío** o **casa de <material> <rango>**.

- Si el objeto es una **calle**, la **dirección** representa el sentido de la calle (recuerde que las calles son unidireccionales), y puede ser **izquierda**, **derecha**, **abajo** y **arriba**.
- Si el objeto es **vacío**, entonces es un espacio para construir un servicio de emergencias.
- Si el objeto es una **casa**, primero dice de qué material está construida. Luego, los valores del rango son los límites de tiempo inferior y superior de los robos en la propiedad. Por ejemplo: **5,6 casa de madera [2,10]** es una casa de madera en la posición (5,6), y los robos tardan entre 2 y 10 minutos en esa casa.

9. Notas:

- Al ejecutar el programa se debe preguntar al usuario el tiempo de simulación y el periodo de tiempo entre cada imagen de la simulación que quiere ver en la interfaz.
- Además se debe generar un archivo **output_escenario_X.txt** para la primera réplica de cada escenario donde se muestre un log de eventos con lo ocurrido. El formato debe ser como el Cuadro 1 (separe con tabs). Preocúpese de ser consistente en el registro de los eventos.

Cuadro 1: Formato Reporte

Tiempo	Evento	Tipo Entidad	Id Entidad	Otro 0	Otro 1
10	Ladrón escapa	ladrón	0		
20	Se produce un incendio	casa	4	madera	
25	Apaga incendio	bombero	10	casa	4
30	Se detiene taxi	taxi	3		

- Al finalizar las simulaciones se tienen que mostrar las estadísticas y una imagen de la distribución más óptima en la interfaz.
- Está prohibido usar threading. Se debe simular usando eventos discretos para la primera parte y sólo donde se especifica SimPy3.
- Si desea usar alguna librería debe preguntar por el foro.

10. Restricciones y alcances

- Tu programa debe ser desarrollado en Python 3.4
- Esta tarea es estrictamente individual, y está regida por el Código de Honor de la Escuela: [Clickear para Leer](#).
- Su código debe seguir la guía de estilos PEP8
- Si no se encuentra especificado en el enunciado, asuma que el uso de cualquier librería Python está prohibido. Pregunte por foro si se pueden usar librerías específicas.
- El ayudante puede castigar el puntaje¹ de tu tarea, si le parece adecuado. Se recomienda ordenar el código y ser lo más claro y eficiente posible en la creación algoritmos.
- Debe adjuntar un archivo `README.md` donde comente sus alcances y el funcionamiento de su sistema (*i.e.* manual de usuario) de forma *concisa* y *clara*.
- Cree un módulo para cada conjunto de clases. Divídalas por las relaciones y los tipos que poseen en común.
- Cualquier aspecto no especificado queda a su criterio, siempre que no pase por encima de ningún otro.

11. Entrega

- **Fecha/hora:** lunes 26 de octubre a las 23:59 hrs.
- **Lugar:** GIT - Carpeta: Tareas/T04

Tareas que no cumplan con las restricciones señaladas en este enunciado tendrán la calificación mínima (1.0).

¹Hasta -5 décimas.