

PRÁCTICA DE LA ASIGNATURA

LABORATORIO DE PROGRAMACIÓN II

Curso 2008/09

Ingeniería Informática
Ingeniería Técnica en Informática de Sistemas
Ingeniería Técnica en Informática de Gestión

Roberto Rodríguez Echeverría (rre@unex.es)
Encarna Sosa (esosa@unex.es)
José María Conejero (chemacm@unex.es)



PRISONBREAK

REDUX

Enunciado de la Práctica

Con el auge de la serie estadounidense Prison Break, en muchas de la cárceles norteamericanas se han producido motines y movimientos de presos con el objetivo de conseguir escapar de la misma. En concreto, en la prisión Joliet de Illinois, la dirección ha observado que los intentos de fugas han aumentado en un 75% desde que esta serie de televisión comenzó su emisión. En este sentido, la dirección ha decidido aumentar el control sobre las diferentes zonas de la prisión, para paliar estos intentos de fugas.

Para poder desarrollar un plan preventivo, la dirección de la prisión ha encargado el desarrollo de un sistema software de simulación que recree las acciones que tienen lugar de manera cotidiana en esta prisión. Este sistema podrá generar simulaciones de intentos de fuga masivos de una prisión, contrastando la fiabilidad de los sistemas de seguridad y mantenimiento de la prisión a analizar. Con este sistema, la empresa pretende obtener unos resultados estadísticos fidedignos que revelen los verdaderos problemas de la prisión, con el fin de justificar ante la administración pública del estado de Illinois la concesión de más medios técnicos y humanos.

A continuación, se realiza una especificación del sistema a desarrollar, fruto de la obtención de requisitos obtenida en diferentes reuniones con el alcaide de la prisión.

Especificación general del sistema

- La misión principal del sistema consiste en simular fugas masivas en diferentes prisiones. Las prisiones se diferenciarán en el número de plantas, sus dimensiones y su distribución, desde el punto de vista estructural. Además, cada prisión tendrá su propio equipo de guardias y, por supuesto, sus propios presos. Evidentemente, para que la simulación sea realista todos los personajes deben moverse por la prisión a la vez.
- Todas las prisiones mantienen el mismo esquema de seguridad basado en las siguientes características:
 - Cada planta sólo tiene una entrada y una salida con el fin de minizar las posibles rutas de escape.
 - Cada planta de la prisión presenta una distribución distinta de sus salas para complicar el desarrollo de una posible ruta de escape. Los diseñadores del sistema han usado el algoritmo de Kruskal para generar la distribución de cada planta de forma diferente, añadiendo una serie de atajos (tirar algunas paredes) posteriormente para mejorar el tránsito por la misma. La figura 1 representa la distribución de una planta de dimensiones 6x6.

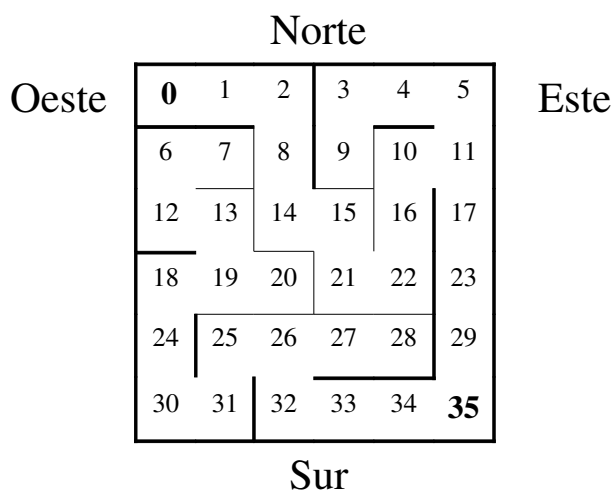


Ilustración 1: Ejemplo de la distribución de las salas de una planta

- Cada sala se puede comunicar con otras cuatro, según los cuatro puntos cardinales. Y en el caso de la sala de salida de una planta, también se comunica bajando las escaleras con la sala de entrada de la siguiente planta.
- El paso de planta a planta está protegido por una puerta de alta seguridad, que sólo puede ser abierta con la combinación de llaves adecuada. Evidentemente, la puerta se encuentra en la salida de la planta.
- Una puerta de alta seguridad se configura insertando un conjunto de llaves predefinido. Una vez configurada, se cierra. Sólo es posible abrirla usando la combinación de llaves adecuada que sea capaz de cambiar su estado a abierta. Todas las puertas de la prisión se configuran y cierran al inicio de la simulación.
- Las puertas disponen de un sistema de alarma basado en una memoria de los códigos de llave ya utilizados en la misma. Esta alarma se activa cuando detecta que un código de llave ya ha sido usado previamente. Dado lo crítico del sistema, el proceso de comprobar si un código ha sido ya utilizado o no debe estar muy optimizado para que la respuesta de la alarma sea inmediata y no dé tiempo al preso a reaccionar.
- Al menos, existe un guardia por planta haciendo una ruta de control de

forma continua. El guardia tiene dos misiones básicas: capturar presos en fuga y mantener cerrada la puerta de su planta. Cuando un guardia encuentra un preso en una celda, lo detendrá, llevándolo a la celda de castigo de la prisión. Si el guardia encuentra la puerta de salida de la planta, comprobará si está abierta y, en este caso, la cerrará.

- Para simular una fuga masiva, se liberará un conjunto de presos en la planta superior de la prisión. Estos presos tendrán acceso al mapa de la prisión durante unos breves instantes previos al intento de fuga para que puedan trazar su ruta de escape de forma individual. Una vez decidida su ruta de escape, los presos se moverán por las distintas salas de cada planta, según su ruta, con el objetivo de hallar la salida de la misma, intentando evitar coincidir con los posibles guardias que patrullan por la misma. Para poder salir de una planta tendrán que abrir su puerta de alta seguridad.

- Las rutas de los personajes se especifican en función de los puntos cardinales y de la sala en la que se encuentren. En cada movimiento, el punto cardinal indica a qué sala debe moverse el personaje desde la sala en la que se encuentra en ese momento. Por ejemplo, si la ruta es “Sur, Este, Sur, Este, Oeste, ... ” y el preso se ha movido ya dos veces, en su tercer movimiento debe moverse a la sala que se encuentre al sur de la sala en la que está. La Figura 2 muestra el ejemplo de ruta comentado para un preso que ha empezado a moverse en la celda 6 y ahora se encuentra en la 13: su siguiente movimiento le llevará a la 19.

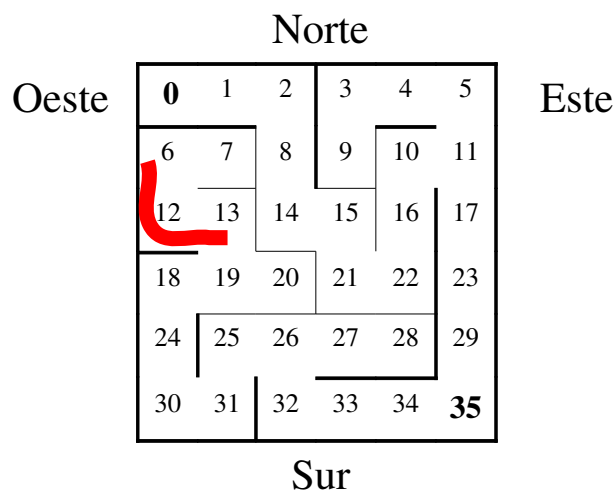


Ilustración 2: Ejemplo ruta de un personaje

- En su camino, los presos irán encontrando llaves que se han perdido o dejado olvidadas en las diferentes salas, de manera que se simule de alguna manera el robo y pérdida de llaves común en este tipo de prisiones. Cada vez que un preso entre en una sala, la revisará en busca de llaves. Si encuentra varias llaves en una sala, sólo tomará una. Cada llave dispondrá de un código que permita identificarla en el sistema. Además, las llaves son capaces de almacenar información biométrica del último individuo que la utilizó, permitiendo identificar al mismo en el estudio de análisis final de la simulación.

- Cuando alguno de los presos consiga abrir la puerta de la prisión (la de la planta inferior), se considerará cumplido el objetivo de la simulación y el sistema terminará ofreciendo los datos de registro de la simulación. Básicamente, estos datos consisten en indicar cuántos presos se han escapado, cuántos presos han sido llevados a las celdas de castigo, cuántas llaves se han utilizado...

- Dado que los presos y los guardias van a recorrer la planta moviéndose por las

diferentes salas, la simulación establecerá un orden para que los personajes se muevan por turnos. De este modo, cada personaje sólo podrá moverse una única vez en cada turno. El orden establecido es el siguiente: primero se moverán los personajes de la planta superior, después los personajes de la planta inferior a ésta y así sucesivamente hasta llegar a la planta más baja. En cuanto al orden seguido en una misma planta, los personajes se moverán siguiendo el orden de las salas en las que se encuentren, primero los de la sala con identificador 0, seguidamente los de la sala 1 y así sucesivamente hasta haber movido a todos los personajes de todas las salas. Los personajes (tanto presos como guardias) que se encuentran en una sala se moverán siguiendo un orden histórico, es decir, aquellos que llegaron primero a la sala saldrán también los primeros de la misma. La figura 3 muestra el estado de una prisión con una planta en la ejecución de la simulación, representando la posición de los presos "X" y "@" y el guardia "P". Por simplicidad, no se muestran en esta figura la presencia de llaves ni de puertas.

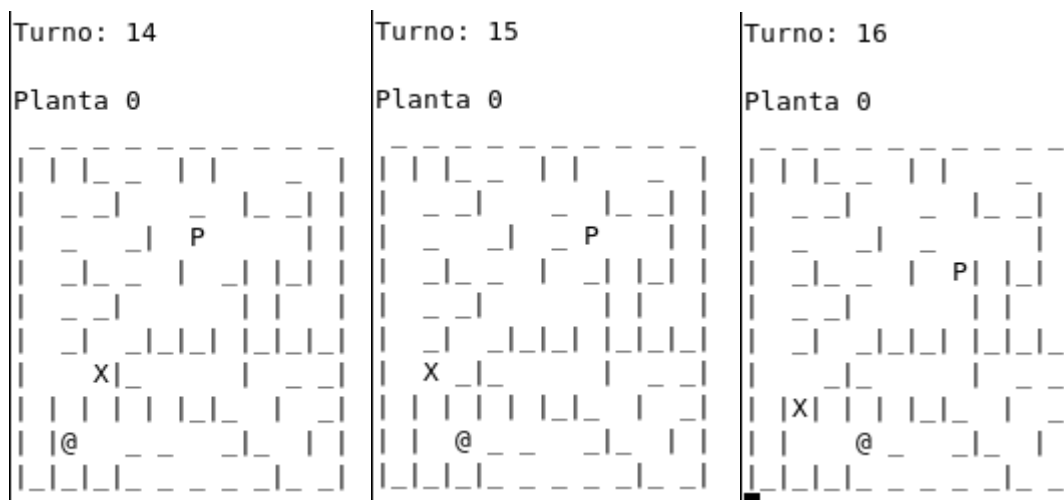


Ilustración 3: Ejemplo de presos y guardia moviéndose por una planta

Objetivos del enunciado

El objetivo principal de este enunciado es presentar una especificación general del sistema a construir sin entrar en detalles de diseño. Se trata de una especificación tipo que se obtendría de la entrevista con un cliente. A partir de los requisitos apuntados, se debe realizar una fase de análisis y diseño del sistema según las metodologías orientadas a objetos para identificar las diferentes clases que compondrán el sistema y sus relaciones. Una vez definidas las clases, sus responsabilidades y colaboraciones se comenzará con la implementación del sistema.

En cuanto a la planificación, el desarrollo del sistema se dividirá en las tres fases o entregas que se detallan a continuación:

- EC1.- Entrega de la clase Puerta y de todas las clases que la componen. El enunciado de la EC1 se publicará el martes 3 de marzo.
- EC2.- Entrega del prototipo de la Prisión, que contendrá toda la parte estructural de la misma y un personaje que permita probar su correcto funcionamiento. El enunciado de la EC2 se publicará el lunes 23 de marzo.
- EC3.- Entrega final. Extensión del prototipo con todos los personajes que forman parte de la simulación. El enunciado de la EC3 se publicará el lunes 11 de mayo.

Notas para los alumnos:

Todos los alumnos matriculados en esta asignatura en el presente curso académico serán examinados y evaluados según sus conocimientos y habilidades para analizar, diseñar, desarrollar y modificar este proyecto individualmente. Dicho trabajo se desarrollará en el lenguaje de programación C++ y bajo los entornos de desarrollo propuestos en la asignatura.

Para llevar a cabo la tarea el desarrollo de la práctica se ha de tener en cuenta que se deben utilizar las siguientes estructuras de datos: árbol, lista, pila, cola y grafo. Además se aplicarán de forma adecuada las características de la P.O.O., tales como herencia o polimorfismo, así como genericidad, sobrecarga, uso de librerías estándar (STL) y control de excepciones.

En sucesivas entregas del proyecto se irán especificando con todo lujo de detalles las características del proyecto en las que no se ha profundizado en este primer enunciado.