Laboratorio N° 2 snake-race-thread-concurrency

Santiago Cárdenas Amaya

Juan Pablo Fonseca

Escuela Colombiana de ingeniería Julio Garavito

Bogotá 2023

Introducción

¿Como se puede mejorar el funcionamiento de una aplicación que usa multi thread’s? Bueno la respuesta se evidencia a continuación no solo teniendo diferentes hilos que ejecuten muchas veces x función, ahora pondremos a prueba estos mismos. Usando 3 nuevas funciones para alterar el estado en como estos se comportan, por un lado, tenemos el Synchronize la cual actuara como un control de acceso para los hilos que intentan acceder a un mismo recurso, por otro tenemos la función wait capaz de poner un hilo en un estado de espera para ceder el procesador a otra funcionalidad y así mismo la función notify que despertara por decirlo de algún modo a este hilo para que siga su curso natura. Con estas 3 se logran comportamientos interesantes que se pondrán aprueba a continuación.

# Parte I -PrimeFinder

1. Descargue el proyecto PrimeFinder
2. modificar la aplicación de manera que cada t milisegundos de ejecución de los threads, se detengan todos los hilos y se muestre el número de primos encontrados hasta el momento. Luego, se debe esperar a que el usuario presione ENTER para reanudar la ejecución de los mismos. Utilice los mecanismos de sincronización provistos por el lenguaje (wait y notify, notifyAll).

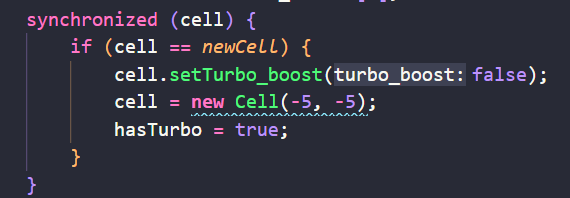
# Parte II - Snake

1. Analice el código para entender cómo hace uso de hilos para crear un comportamiento autónomo de las N serpientes.
2. Escribir en RESPUESTAS.txt (se usó este mismo documento)

* Posibles condiciones de carrera.
  + Cuando 2 o más serpientes lleguen a una misma comida, punto de teletransportación, o boost de velocidad
* Uso inadecuado de colecciones, considerando su manejo concurrente (para esto, aumente la velocidad del juego y ejecútelo varias veces hasta que se genere un error).
  + La lista de snakeBody debería ser de la librería atomic puesto que las celdas deben estar sincronizadas
* Uso innecesario de esperas activas.
  + No detectamos alguna

1. Identifique las regiones críticas asociadas a las condiciones de carrera, y haga algo para eliminarlas. Tenga en cuenta que se debe sincronizar estrictamente LO NECESARIO. En su documento de respuestas indique, la solución realizada para cada ítem del punto 2. Igualmente tenga en cuenta que en los siguientes puntos NO se deben agregar más posibles condiciones de carrera.

* Las celdas del cuerpo de la serpiente. Para esto lo que se hizo fue convertir la lista snakeBody a listas concurrentes, gracias a esto el concurrentModificationError dejó de aparecer.
* Dado que no se tiene en cuenta las colisiones entre serpientes, los otros puntos críticos son cuando una serpiente toca una celda que afectará su comportamiento, es decir cuando son: comida, punto de teletransportación, o boost de velocidad. Para solucionarlo, dentro de cada iteración de los métodos “checkIf” en los que se revisaban bloques, se sincronizaron los mismos, así como se aprecia en la imagen.



1. Haga los ajustes necesarios para que a través de botones en la interfaz se pueda Iniciar/Pausar/Reanudar el juego: iniciar el juego si no se ha iniciado aún, suspender el juego si está en ejecución, reactivar el juego si está suspendido. Para esto tenga en cuenta: Al pausar (suspender) el juego, en alguna parte de la interfaz (agregue los componentes que desee) se debe mostrar:

* La serpiente viva más larga
* La peor serpiente: la que primero murió

Recuerde que la suspensión de las serpientes NO es instantánea, y que se debe garantizar que se muestre información consistente.

Para este punto no fue posible la parte final dado que se tuvo problemas con la integración de los botones ya que cuando se pusieron a prueba, el juego no se mostraba visualmente, lo que nos generó percances, por lo cual preferimos explicar aquí lo que sucedió. En caso de que se quiera ver el intento fallido, se puede acceder a la rama features/l2/snake.

URL: https://dev.azure.com/juanpablofonsecac5/ARSW

# Conclusiones

Entender hilos es complicado, y si se le suma interfaz visual, puedes llorar todo un día. Mirando los aspectos positivos, una buena gestión de hilos permite llevar aplicaciones tradicionales a un siguiente nivel.