



Algoritmos y Programación II

Trabajo Práctico N 3: “Nanobot”

Objetivo

Desarrollar un programa que, basándose en los Trabajos Practicos 1 y 2, cree los objetos de entrada, los visualice dentro del entorno de juego y permita el uso del simulador Nanobot.

Descripción

El simulador NanoBot

Los NanoBots se manejan a control remoto y son realmente un recurso escaso, con lo cual necesitamos maximizar la eficiencia de los mismos. Para ello, es menester contar con un grupo de excelentes pilotos a distancia y por lo tanto se requiere un mecanismo de entrenamiento. Es por eso que nuestro objetivo es desarrollar un simulador en el cual los pilotos puedan entrenarse en el uso de estas nuevas tecnologías. La aplicación a desarrollar debe simular el entorno de un organismo infectado previamente descrito en los TPs 1 y 2 y debe permitir al jugador manejar un NanoBot con todas las funcionalidades mencionadas sobre el mismo. El manejo del juego debe ser a través del teclado, respetando la siguiente configuración de teclas:

Flechas arriba/abajo/izquierda/derecha: para hacer avanzar el NanoBot en la dirección de la flecha.

Teclas 1 y 2: inyecta una dosis de suero A y B, respectivamente.

Tecla 3: atrapa un anticuerpo (si la nave está en contacto con el mismo).

Barra espaciadora: libera un anticuerpo atrapado (si la nave tenía uno).

Son requerimientos obligatorios los siguientes:

- 1) Al inicio del juego debe haber una cantidad de células (sanas y células X, Y, Z) formando la red de citoplasma en las posiciones correspondientes. Esta información debe leerse de la manera descrita en el TP1. La proporción de células sanas debe ser mayor que las de células Z. De igual forma se lee la cantidad de dosis de cada tipo de suero del NanoBot y los anticuerpos.

- 2) Las células deben estar fijas, no así el NanoBot que debe moverse con el accionar de teclas, ni los anticuerpos que deben moverse por sí solos.
- 3) El impacto de una dosis de suero contra una célula debe respetar el resultado detallado en el TP2 en función del tipo de suero que contenga.
- 4) Los anticuerpos deben moverse aleatoriamente por la red. Se debe respetar también la interacción entre los anticuerpos y el NanoBot explicada anteriormente. Lo mismo con la interacción entre los anticuerpos y las células Z.
- 5) Si en algún momento del juego, la proporción de células Z supera a la de células sanas, el juego se pierde, y si en algún momento del juego se destruyen todas las células Z, el juego se gana.

La implementación a entregar debe cumplir como mínimo con todos los requerimientos obligatorios planteados arriba. Adicionalmente, se pueden agregar las siguientes funcionalidades:

Dos jugadores: Agregar al juego la opción de jugar de a dos.

Minas de suero: El NanoBot puede dejar minas explosivas con suero (A o B). Estas minas podrían activarse y afectar a todas las células que estén dentro de un cierto radio de alcance. La explosión gráfica de las dosis ya está implementada con las teclas A y B.

Varios niveles de dificultad: Al ganar el juego, en lugar de terminar el mismo, se puede pasar a nuevos niveles que sean más difíciles que el anterior.

Implementación base

Junto con este enunciado, se les entrega el paquete “tpNanobot.zip” que contiene la clase Entorno. Esta clase es capaz de encargarse de la interfaz gráfica (incluyendo armado de la red de citoplasma) y de la interacción con el usuario. Así, el grupo solo tendrá que encargarse de la implementación de la “inteligencia” del juego. Para más información sobre la interfaz gráfica revise el “manualTPNanobot.pdf”.

Datos de entrada

Se deberá leer el archivo de texto de la forma descripta en el TP N 1 y crear los objetos en las cantidades y con las características que describa dicho archivo.

Salida

La aplicación debe, luego de procesar el archivo de entrada, crear todos los objetos involucrados, ubicándolos al momento de su creación en las posiciones indicadas dentro del escenario de juego y permitir la simulación tal como se describe.

Aclaraciones

- Las mismas consideradas en los TPs 1 y 2.
- La definición de detalles en el comportamiento de los objetos, queda a libre decisión del equipo de trabajo.
- Para el desarrollo de este trabajo se debe considerar el armado de la red citoplasma utilizando una lista general de elementos en la cual cada nodo debería vincularse con otra lista con los elementos adyacentes con los que se conecta.
- El criterio de armado de la red a medida que se leen del archivo los distintos tipos de células es: tomar de a 3 células y hacerlas adyacentes entre sí, luego, tomar la tercera hacerla adyacente con las siguientes 2 y así sucesivamente.

Normas de entrega

Se deberá subir un archivo comprimido al campus, en un link que se habilitará para esta entrega. Este archivo deberá tener un nombre formado de la siguiente manera:

NombreGrupo-TP3

Deberá contener los archivos fuente y la documentación.

La fecha de entrega vence el día lunes 18/11/2019 a las 23.55hs.

Se evaluará: funcionalidad, eficiencia, algoritmo utilizado, buenas prácticas de programación (como nombres, sangrías, etc.), modularización, comentarios.

Puntaje: 20 puntos.