

Práctica 2: Divide y Vencerás

1 La Universidad de la Sabiduría

En las tierras de Málaga, donde las olas del Mediterráneo acarician las costas y las palmeras se mecen con gracia en la brisa, se encuentra la antigua Universidad de la Sabiduría. Esta institución, envuelta en siglos de conocimiento, ha sido el faro de la educación y la erudición.

Sin embargo, en las profundidades de los bosques circundantes, se agita una oscura amenaza: los Goblins de la Ignorancia. Una tribu maligna que busca apoderarse de los tesoros del saber que yacen resguardados en los anaqueles de la universidad. Guiados por un líder astuto y codicioso, los goblins han tramado una estratagema para infiltrarse en la universidad y saquearla de sus invaluable conocimiento.

Con el peligro acechando, los sabios y eruditos de la universidad han convocado a los más valientes y astutos estrategias para proteger los pasillos del saber. Armados con ingenio y sabiduría, estos defensores están listos para enfrentar a la horda de goblins y mantener el legado de conocimiento a salvo.

Los estrategas deberán utilizar con destreza las torretas para bloquear los caminos de los goblins y mantener a raya la marea de ignorancia que amenaza con inundar la universidad.

El destino de la Universidad de la Sabiduría yace en manos de aquellos que se alzarán contra la oscuridad de la ignorancia, con la esperanza de preservar el fulgor del conocimiento para las generaciones venideras. ¿Serás tú quien guíe a la victoria, o caerá la universidad en la sombra del desconocimiento? ¡El desafío comienza ahora!

2 Reglas del juego

El terreno de juego (ver Fig. 1) está compuesto por un tablero de celdas cuadradas, algunas transitables y otras no transitables. Algunas de las celdas del borde izquierdo del tablero se marcan como celdas de salida de los goblins. De igual forma, de entre todas las celdas del borde derecho se seleccionarán algunas como celdas de salida.

El objetivo de los goblins es llegar a una de las salidas y escapar con el conocimiento robado. El objetivo de los defensores es impedir que los goblins alcancen las salidas. Para ello cuenta con una serie de torretas defensivas que puede colocar en las celdas no transitables del tablero. Cada torreta tiene diferentes características:

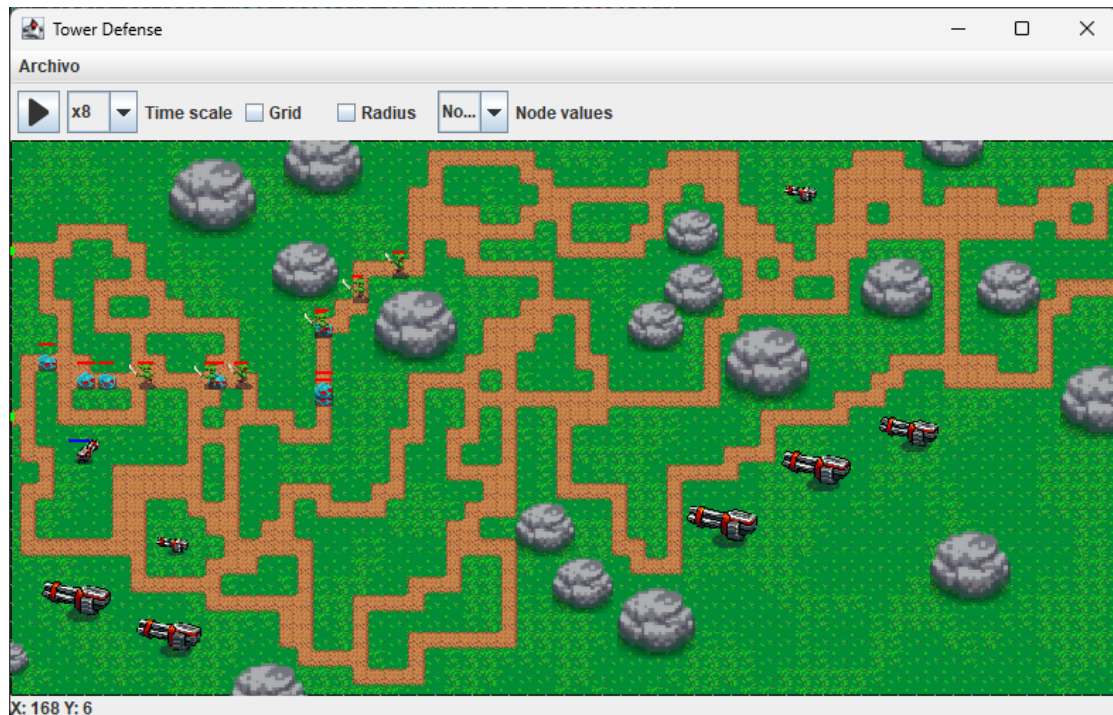


Figure 1: Interfaz gráfica del juego

- Radius: es el diámetro de la circunferencia que ocupa la torreta en el mapa (circunferencia roja en Fig. 2). No puede colocarse una torreta en una posición tal que colisione con otra entidad colocada previamente en el mapa o cuya superficie se solape con caminos transitables.
- Range: la distancia máxima a la que puede disparar un proyectil (circunferencia amarilla en Fig. 2).
- Damage: la cantidad de daño que produce en el punto de impacto del proyectil.
- Cooldown: la cadencia entre disparos.
- Dispersion: la distancia máxima a la que inflige daño el impacto de un proyectil. La cantidad de daño es proporcional a la distancia al punto del impacto.
- Cost: el coste de la torreta.

Dentro del mapa, y siempre fuera de los caminos transitables, también pueden encontrarse obstáculos. Las torretas han de colocarse siempre de tal forma que no colisiones con los obstáculos, también con un radio propio.

Los goblins avanzan, en busca de alguna salida, a través de los caminos transitables. Un goblin que se sale de los caminos transitables es automáticamente eliminado. Al igual que las torretas, los goblins también tienen sus propias características:

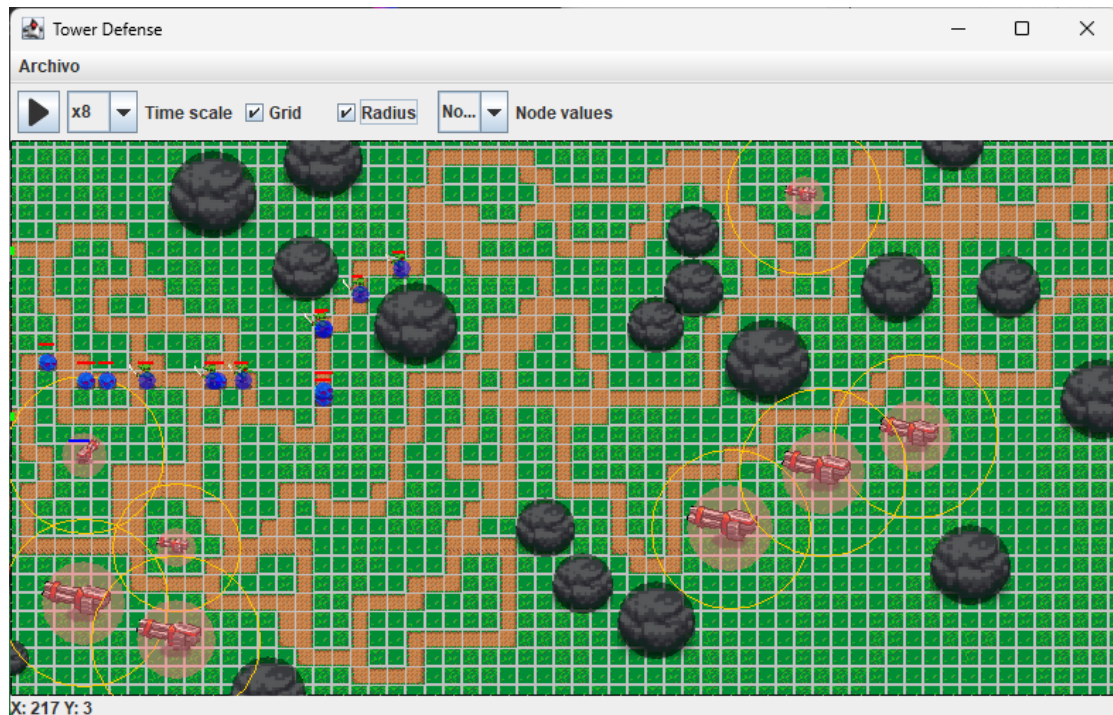


Figure 2: Características de las entidades

- Radius: es el diámetro de la circunferencia que ocupa el goblin en el mapa (circunferencia azul en Fig. 2).
- Speed: unidades de espacio que puede recorrer el goblin en un segundo.
- Health: daño que es capaz de absorber antes de ser destruido.

Las torretas atacan a los goblins que tienen a su alcance. Cuando un proyectil impacta, hace cierta cantidad de daño a los goblins alrededor del impacto. Si un goblin recibe más daño del que puede recibir, muere.

El juego se divide en rondas. En cada ronda una cantidad de goblins y de torretas son colocados en el tablero. La puntuación de cada ronda coincide con el número de goblins que han conseguido escapar en ella. El juego termina cuando finaliza la última ronda. La puntuación final obtenida será la suma de las puntuaciones de cada ronda.

3 Ejercicios

Para completar la práctica se proponen una serie de tareas, que deberán llevar a cabo cada alumno individualmente. El resultado de estas tareas dará lugar al desarrollo de una aplicación en Java y de una memoria explicativa de los resultados obtenidos. Ambos deberán entregarse a través del campus virtual antes de la fecha límite, a través de la

tarea creada al efecto y en el formato solicitado. No se corregirán productos enviados por otros medios o que se envíen en un formato distinto al solicitado.

El objetivo de la práctica consiste en desarrollar un programa que sirva para diseñar un procedimiento eficiente para la colocación de las torretas. No es el objetivo de esta práctica la colocación de esa tarea se llevará a cabo en una práctica posterior.

En el material de prácticas podrá encontrar un código de ejemplo que genera terrenos para el juego de diferentes tamaños. Dentro de esos terrenos habrá una lista de celdas no transitables en las que se deberán colocar un número dado de torretas. Antes de colocar las defensas, en esta práctica estudiaremos la eficiencia de algunos algoritmos para la selección de las mejores celdas.

Supongamos que el valor de la celda se asigna en función de la distancia al centro del terreno. Si debemos devolver las k mejores ubicaciones podemos ordenar las celdas en función de este valor, en orden creciente, y devolver las k primeras celdas de la lista ordenada. Pero, ¿qué método de ordenación puede ser el más eficiente? Pero, ¿merece la pena realizar la ordenación? Estas son las cuestiones que han de resolverse en esta práctica. Concretamente, las tareas a llevar a cabo son:

1. Descargar el archivo comprimido del campus con los códigos de ayuda y estudiar su contenido.
2. Ejecutar el archivo “compile.bat” desde una consola de comandos para compilar el código de ejemplo.
3. Ejecutar el archivo “run.bat” para ejecutar el código de ejemplo.
4. Modificar los métodos “selectBestNodesNoSort”, “selectBestNodesInsertionSort”, “selectBestNodesMergeSort” y “selectBestNodesQuickSort”, de la clase “Main”, para que devuelvan la lista de las mejores k (*count*) celdas para colocar las torretas. El método “selectBestNodesNoSort” debe encontrar las celdas sin ordenar el listado, mientras que el resto de métodos deben aplicar el algoritmo correspondiente.
5. Añadir un comentario al comienzo del fichero “Main.java” con la tabla de tiempos obtenidos. Acompañar la tabla con un texto donde se justifiquen esos resultados, indicando por qué método se decantaría finalmente para realizar la selección de las celdas.

4 Entrega

Todo el código desarrollado en esta práctica deberá estar incluido en el fichero “Main.java”. Será éste el único archivo que se envíe a través de la tarea habilitada al efecto en el campus virtual. El fichero debe enviarse sin comprimir.

5 Evaluación

Todos los programas enviados por los alumnos serán ejecutados en uno de los ordenadores del laboratorio y se anotará el número de métodos correctamente implementados por cada alumno. La calificación de la práctica se otorgará en función de la cantidad de métodos implementados correctamente y de la justificación de los resultados obtenidos.