

Memoria

Generación de mapas

Carlos Bailón Pérez	carloskb24@gmail.com
Daniel Castaño Estrella	dani.kas.90@gmail.com

Contenido de la práctica (el 100% de lo que se pedía)

Laberinto

Tipo de casilla:

- Cuadrada
- Hexagonal
- Triangular

Mapa

Tipo de mapa:

- Pangea
- Islas
- Continentes

Notas de los autores

Hemos intentado ser lo más profesionales posible con esta práctica, intentando hacer un código lo más óptimo posible (en el único sitio en que no nos hemos preocupado tanto ha sido en la clase Executor con todos sus métodos prints, que son un caos, pero un caos lógico) y a la vez claro y expresado en las menos líneas posibles.

Herencias, clases abstractas, buenas prácticas de getters y setters con variables privadas o protegidas, clases finales para mejorar el rendimiento, datos genéricos, control de datos de entrada (cuando se pide un int y no lo introduces no falla el programa y por supuesto no se permite introducir números inválidos), reimplementación de métodos, métodos de test disponibles pero comentados para no perjudicar la versión “release”... Estas son algunas de las prácticas que hemos tenido en cuenta a la hora de elaborar la entrega y que más nos enorgullecen, a parte de su correcto funcionamiento, claro.

Para un entendimiento óptimo del código (y para no enrollarnos de más aquí) hemos incluido un javadoc y un UML.

Cambios frente al enunciado

- El tipo de mapa se pide antes del tamaño ya que es una elección que puede afectar a la decisión del tamaño (por ejemplo: el laberinto de triángulos queda estrecho si igualamos filas y columnas)

Consideraciones finales (y un poco de humor)

- Nos hubiera gustado poder implementar un algoritmo que no tuviera forma de árbol para el laberinto, para permitir ciclos y poder dar vueltas sobre un mismo conjunto de paredes. Para otra vez.
- Al final traducimos todos los comentarios, variables y métodos al inglés (esperamos no habernos dejado nada en castellano) porque sabemos que es una buena costumbre, sobre todo en nuestra industria donde hay mucho más trabajo fuera que aquí.
- Hemos entendido las ventajas de la recursividad y en general ha sido una práctica muy positiva.
- Lo más difícil de la práctica ha sido dibujar los hexágonos, un verdadero quebradero de cabeza, ¡teníamos que comentarlo!
- Hay un porcentaje alarmante de gente que hace tutoriales en YouTube sobre algoritmos y confunde la palabra con “logaritmo”.
- Test de última hora: porque nunca está de más testear una vez más. Se nos ocurrió poner la misma casilla de salida que de llegada y... crash. Así que pusimos una condición que tuviera en cuenta este problema.