

TI2-snakes-and-ladders

Requerimientos funcionales y diagrama de clase

2021

Daniela Olarte G1
Santiago Gutiérrez G2

Requerimientos funcionales

RF1. Crear un tablero de juego. Para esto se necesita el número de columnas, el número de filas, el número de serpientes, el número de escaleras, el número de jugadores y el nombre del jugador(es). Todos estos valores son leídos de una sola línea y tienen que estar exactamente en el orden anterior separados por un espacio. Las serpientes (identificadas con letras mayúsculas del abecedario) y las escaleras (identificadas con números) serán asignadas de forma aleatoria en cualquier cuadrícula del tablero y tendrá tanto un inicio como un final. Para identificar las filas se utilizarán números ascendentes de izquierda a derecha, sin embargo, cuando se identifique que la línea ha terminado, este sube y se desplaza hacia la izquierda, y así mismo cuando la línea haya terminado, este sube y se desplaza hacia la derecha; y así sucesivamente. Asimismo, el tablero se crea satisfactoriamente si ninguna escalera inicia en la casilla 1, ninguna serpiente inicia en la casilla $n \times m$ y ninguna casilla de inicio o fin de la escalera o serpiente coincide con otro inicio o fin de escalera o serpiente. Posteriormente, se imprime el tablero en consola después de un salto de línea con sus casillas llenas: enumeración de casillas, serpientes y escaleras.

RF2. Empezar a jugar. Una vez se da un salto de línea, el tablero solo muestra los símbolos de los jugadores y el número de escaleras y serpientes aleatorias indicadas por el usuario. En adelante solo se podrá jugar ingresando un salto de línea para que juegue el jugador a quien le corresponde su turno. Por lo que, cada vez que el jugador juegue, un dado aleatorio (con un mínimo de movimientos de 1 y un máximo de movimientos de 6) juega internamente en el programa. Posteriormente, se imprime en consola el símbolo del jugador y el puntaje que saca el dado, e inmediatamente, se muestra el tablero con la nueva posición.

RF3. Caer en la posición en la que una serpiente o escalera se encuentra. Si el jugador cae en un inicio de serpiente, este baja hacia el fin de la serpiente y si el jugador cae en un fin de una escalera, este sube hacia el inicio de la escalera. Posteriormente, se muestra el tablero con la nueva posición.

RF4. Digitar palabras específicas. Si el jugador escribe la palabra **num** y luego se da salto de línea, el programa mostrará la misma cuadrícula que muestra al inicio, con las casillas numeradas, las serpientes y las escaleras. Esperará entonces un salto de línea para mostrar el tablero actual (con las serpientes, escaleras y posición actual de los jugadores) y continuar con el juego. Si el jugador escribe la palabra **simul** y luego se da salto de línea, el programa empezará en modo simulación, que consiste en ir mostrando lo que cada jugador a su turno juega, con el tablero correspondiente de cada nueva posición, esperando 2 segundos entre cada jugada, pero sin esperar ningún salto de línea. Si el jugador escribe la palabra **menu** y luego se da salto de línea, el juego se corta sin terminar y el programa regresa al menú principal, mostrándose sus opciones.

RF5. Calcular el puntaje de un jugador. El juego termina cuando uno de los jugadores llegue a la última casilla (casilla cuya numeración es $n \times m$). De modo que, para calcular el puntaje se multiplica la cantidad de movimientos y la cantidad total de casillas del tablero.

RF6. Registrar un jugador. Para esto se necesita el apodo, su símbolo y el puntaje que sacó en el juego. El jugador se guarda en un árbol binario ordenado inversamente por el puntaje y luego se imprime en consola el nombre o apodo del jugador ganador y la cantidad de veces que el jugador lanzó el dado en el juego.

RF7. Guardar la información de los jugadores en un archivo serializado de forma automática cada vez que se registre un jugador.

RF8. Recuperar la información de los jugadores del archivo serializado al iniciar la aplicación.

RF9. Mostrar los puntajes y los demás datos (nombres/apodos y sus símbolos) de todos los jugadores de forma inversa en una lista vertical enumerada.

Diagrama de clases

