

Requerimientos TI 2

Se requiere realizar un juego de dos jugadores con temática de Rick y Morty consistente en un tablero con representación en una lista enlazada que cuenta con enlaces adicionales los cuales permiten ir de una casilla a otra más rápidamente. El objetivo del juego es conseguir más súper semillas, ubicadas en casillas aleatorias, que el otro jugador en el menor tiempo posible para obtener un puntaje mayor y ganar el juego.

En el turno de cada jugador, se va a poder tirar el dado, ver el tablero, ver los enlaces de portales y el marcador.

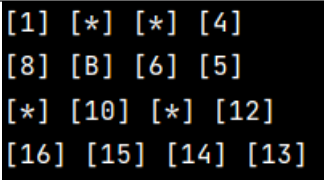
A continuación, se especificará el problema y se enunciarán cada uno de los requerimientos en detalle.

Especificación del Problema

Cliente	Domiciano Rincón
Usuario	<ul style="list-style-type: none">• Jugadores
Requerimientos Funcionales	<ol style="list-style-type: none">1. Generar el tablero2. Desplegar menú por turno3. Tirar el dado4. Ver el tablero5. Ver enlaces del tablero6. Ver el marcador del juego7. Determinar ganador y guardar puntajes
Contexto del problema	Se deben tener en cuenta desde un principio las columnas y filas para generar el tablero, además de la cantidad de semillas y portales que se van a generar. Las posiciones de los jugadores, portales y semillas son aleatorias, haciendo necesario el uso de la clase Random de Java.

	Se deben utilizar algoritmos recursivos para recorrer listas enlazadas de este tipo, y búsqueda binaria para encontrar y actualizar registros repetidos en la tabla de puntajes.
Requerimientos no Funcionales	<ol style="list-style-type: none"> 1. Representar el tablero mediante una lista enlazada de doble enlace con enlaces adicionales aleatorios que representan los portales, estos deben ser menores a la mitad del total de casillas y no deben conectar más de dos casillas. 2. Se debe usar recursividad para recorrer el tablero, no se permiten bucles. 3. El tablero de resultados debe usar arreglos, listas o colecciones para usar algoritmos de ordenamiento y búsqueda.

Requerimiento funcional 1

Nombre o identificador	R1: Generar el tablero		
Resumen	El sistema debe poder generar un tablero, ubicar a los jugadores, semillas y portales al azar para comenzar el juego.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	numColumnas	int	
	numFilas	int	
	numSemillas	int	Cantidad de semillas menor que $(\text{numColumnas} * \text{numFilas}) - 2$
	numPortales	int	Cantidad de portales menor que $(\text{numColumnas} * \text{numFilas}) / 2$
Actividades generales necesarias para obtener los resultados	<ol style="list-style-type: none"> 1. Ubicar primero a los jugadores para evitar que caigan en un portal desde un comienzo 2. Crear el tablero de forma que todas las casillas estén consecutivas 		
Resultado o Postcondición	 <p>Un mapa mostrando donde están las semillas y los jugadores</p>		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	mapa	String	N/A

Requerimiento funcional 2

Nombre o identificador	R2: Desplegar menú por turno		
Resumen	En cada turno de los jugadores, el sistema debe desplegar un menú y pedir la acción a realizar.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	acción	int	Verificar que sea una opción posible
Actividades generales necesarias para obtener los resultados	1. El usuario tendrá que elegir entre las opciones que se le muestran en pantalla		
Resultado o Postcondición	Dependiendo de la acción de la persona se van a hacer diferentes cosas, ya sea tirar un dado, ver el mapa, ver los portales, ver los puntos y hasta finalizar el juego.		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	N/A	N/A	N/A

Requerimiento funcional 3

Nombre o identificador	R3: Tirar el dado		
Resumen	Si el jugador tira el dado, el sistema debe preguntar si avanza o retrocede para efectuar el movimiento del jugador y determinar su posición final y semillas recolectadas.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	avanza	int	La opción tendrá que ser 1 o 2
Actividades generales necesarias para obtener los resultados	1. Se recibirá la opción y se moverá al usuario que esta tirando en el turno hacia la derecha o izquierda dependiendo de lo que haya seleccionado		
Resultado o Postcondición	Nodo del comienzo con el atributo de arraylist con un usuario menos		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	N/A	N/A	N/A

Requerimiento funcional 4

Nombre o identificador	R4: Ver el tablero		
Resumen	Si el jugador decide ver el tablero, el sistema mostrará una representación ASCII del tablero en donde “R” será donde está ubicado Rick, “M” donde está ubicado Morty, “*” donde estén ubicadas las semillas y de resto serán números en las casillas.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	N/A	N/A	N/A
Actividades generales necesarias para obtener los resultados	<ul style="list-style-type: none">- Verificar donde se encuentran los usuarios y las semillas para poder remplazar el numero de la casilla con * si hay semillas, “M” si esta Morty, “R” si esta Rick, y “B” si están ambos		
Resultado o Postcondición	Mapa creado		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	mapa	String	N/A

Requerimiento funcional 5

Nombre o identificador	R5: Ver enlaces del tablero		
Resumen	Si el jugador decide ver los enlaces, el sistema mostrará una representación ASCII de los enlaces usando pares de letras.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	N/A	N/A	N/A
Actividades generales necesarias para obtener los resultados	1. Imprimir la lista, pero primero verificando si el atributo de portal tiene alguna letra, en caso de no ser así se imprime el numero		
Resultado o Postcondición	Mapa de los portales		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	mapa	String	N/A

Requerimiento funcional 6

Nombre o identificador	R6: Ver el marcador del juego		
Resumen	Si el jugador decide ver el marcador, se mostrarán las semillas recolectadas por Rick y Morty.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	N/A	N/A	N/A
Actividades generales necesarias para obtener los resultados	<ol style="list-style-type: none">1. Se busca donde están los jugadores en ese momento2. Se usa el método para saber el número de semillas recolectadas3. Y se muestra en pantalla el numero de semillas por cada usuario		
Resultado o Postcondición	Se sabe cuantas semillas tiene cada uno		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	numSemillas	int	N/A

Requerimiento funcional 7

Nombre o identificador	R7: Determinar ganador y guardar puntajes		
Resumen	Al final del juego ganará el jugador que haya recolectado más semillas y se tendrá en cuenta su puntaje para el tablero de resultados usando la fórmula: $\text{puntaje} = \text{semillas recolectadas} * 120 - \text{tiempo en segundos}$. Los puntajes deben ser guardados en un archivo externo para su posterior uso y referencia.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	N/A	N/A	N/A
Actividades generales necesarias para obtener los resultados	<ol style="list-style-type: none">1. Recibir el numero de semillas por cada jugador2. Usar la formula3. Comprobar si queda en el top 5 de los mejores jugadores4. Guardar la información en el json		
Resultado o Postcondición	El ganador del juego tendrá un puntaje para saber si esta en el top 5		
Salidas	Nombre salida	Tipo de dato	Condición de selección o repetición
	Jugadores guardados	JSON	N/A
	Top 5 mejores jugadores	.txt	N/A