

MEMORIA DEL PROYECTO

Stranger Things: Rescata a Will

Fecha: Diciembre 2025

Asignatura: Programación en Java

autores: sergio sierra, helen katherin, diego carnicer

LINK REPOSITORIO GIT:

<https://github.com/byserch1154/STRANGER-THINGS/tree/main>

1. Introducción

Bueno, para este proyecto he decidido hacer un juego de aventuras basado en Stranger Things porque me gusta la serie y creo que era una forma entretenida de aplicar todo lo que hemos visto en clase. La idea era hacer algo que no fuera el típico ejercicio aburrido y que realmente tuviera sentido jugarlo.

El objetivo principal es que el jugador tenga que rescatar a Will del Upside Down tomando decisiones que afecten al resultado final. He intentado que haya varios caminos posibles y que no sea solo seguir una línea recta.

ABSTRACT

This project consists of developing a text-based adventure game programmed in Java, based on the Stranger Things series. The main objective is to rescue the character Will Byers from the Upside Down through decision that affects the story development. An interactive scene system, inventory management, mini-games hangman, map system and multiple endings have been implemented. Development has been organized using structured programming with methods, arrays and control structures. The results show a functional game that allows the user to have different experiences based on their choices, meeting the educational objectives of applying Java programming knowledge in a practical and entertaining way.

2. Desarrollo del proyecto

2.1 Estructura general

Lo primero que hice fue pensar en cómo organizar todo el código. Al final me decidí por usar un sistema de escenas numeradas porque me parecía la forma más sencilla de controlar

por dónde va el jugador. Cada escena tiene sus opciones y según lo que elijas vas a una escena u otra.

He usado variables estáticas para todo (el inventario, las estadísticas, el mapa...) porque así puedo acceder a ellas desde cualquier método sin complicarme la vida pasando parámetros por todos lados. Sé que quizás no es la forma más elegante pero funciona bien para este proyecto.

2.2 Sistema de escenas

Las escenas son básicamente métodos que se llaman escena1(), escena2(), etc. Cada una muestra un texto describiendo la situación y luego te da opciones numeradas. He hecho 14 escenas principales más algunas especiales como el ahorcado o el mapa interactivo.

Para controlar a qué escena ir uso la variable `escenaActual` que voy cambiando según lo que elige el jugador. El bucle principal en `jugar()` se encarga de ir ejecutando las escenas hasta que llegas a uno de los finales.

2.3 Inventario

El inventario lo he hecho con un array de Strings básico. Sé que hay formas más modernas como usar ArrayList pero quería hacerlo con lo que hemos dado en clase. Llevo la cuenta de cuántos items hay con `cantidadItems` y voy añadiendo objetos con el método `agregarInventario()`.

Los objetos importantes son:

- La linterna (la encuentras en el bosque)
- La brújula (te la da Hopper)
- La llave (necesaria para el mejor final)
- La foto de Will (te la da Joyce)

He puesto un método `tieneItem()` para comprobar si tienes algo antes de dejar hacer ciertas acciones. Por ejemplo, si no tienes la llave no puedes conseguir el final bueno.

2.4 Mini-juego del Ahorcado

Esta parte me costó un poco más. El ahorcado es necesario para entrar al laboratorio y he usado palabras relacionadas con la serie (demogorgon, hawkins, eleven...).

Lo que hace es:

1. Elige una palabra aleatoria del array
2. Crea un array de caracteres con guiones bajos
3. Va pidiendo letras y comprobando si están en la palabra
4. Dibuja el muñeco del ahorcado según los fallos
5. Si aciertas todas las letras ganas, si llegas a 6 fallos pierdes

Para dibujar el ahorcado he guardado los diferentes estados en un array de Strings y voy mostrando el que corresponde según los intentos que quedan.

2.5 Sistema de mapas

El mapa lo tengo como una matriz de caracteres donde cada letra representa un sitio:

- H = Hawkins
- L = Laboratorio
- B = Bosque
- P = Portal
- X = Posición del jugador

Lo muestro cuando el jugador elige la opción de ver mapa y también hay una escena especial (la 80) donde los chicos te enseñan el mapa y puedes elegir a dónde ir directamente.

2.6 Sistema de finales

Hay tres finales posibles dependiendo de lo que hagas (que exista la escena 100 no quiere decir que haya 100 escenas):

Final bueno (escena 98): Si tienes la llave y la usas en el portal, rescatas a Will y todo acaba bien.

Final neutro (escena 99): Si entras al portal sin la llave, ves a Will pero no puedes sacarlo. Te salvas tú pero él sigue atrapado.

Final malo (escena 100): Si fallas el ahorcado o tomas malas decisiones aparece el Demogorgon y es game over.

Al acabar siempre muestro las estadísticas: tiempo jugado, items recogidos, mapas vistos y decisiones tomadas.

2.7 Validación de entrada

Para que no crashee si el usuario escribe algo raro he hecho varios métodos de validación:

- `leerOpcion()`: Se asegura de que introduces un número válido dentro del rango
- `esNumeroValido()`: Comprueba que el String sea realmente un número
- `esLetra()`: Para el ahorcado, verifica que sea una letra minúscula

Si introduces algo incorrecto simplemente te vuelve a pedir la opción, no se rompe el programa.

3. Dificultades encontradas

Lo que más me ha costado ha sido:

- Hacer que el flujo entre escenas tenga sentido y no te pierdas
- El ahorcado, sobre todo la parte de ir revelando las letras correctamente
- Controlar todos los casos donde el jugador puede hacer cosas raras (meter letras donde hay que meter números, etc)

También al principio tenía problemas con el inventario porque se duplicaban items pero lo arreglé comprobando primero si ya lo tienes antes de añadirlo.

4. Posibles mejoras

Si tuviera más tiempo me gustaría:

- Añadir más escenas y hacer la historia más larga
- Meter algún mini-juego más además del ahorcado
- Hacer que el mapa sea más interactivo, tipo que puedas moverte por él
- Guardar partidas para poder continuar después
- Poner algún combate contra el Demogorgon con sistema de vida y ataques

Pero bueno, para el tiempo que tenía creo que ha quedado bastante completo.

5. Conclusión

En general estoy contento con cómo ha salido el proyecto. He podido aplicar prácticamente todo lo que hemos visto: arrays, bucles, condicionales, métodos, validación de datos...

Lo que más me ha gustado es que no es solo código por código, sino que realmente puedes jugar y entretenerse un rato.

He aprendido bastante, sobre todo a organizar mejor el código cuando es un programa más grande. También me he dado cuenta de lo importante que es validar bien las entradas del usuario y el estado en ejecución de código estamos porque si no el programa se rompe por cualquier tontería.