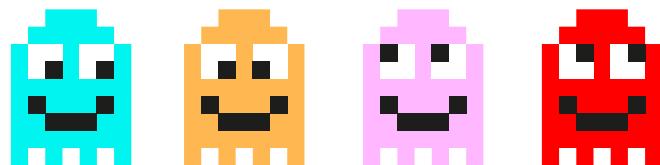


UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA ESCUELA DE
CIENCIAS Y SISTEMAS
Introducción a la Programación y Computación 1
Sección F
Ing. William Estuardo Escobar Argueta
Aux. José Leonel López Ajvix



PAC MAN

Informe de Desarrollo



20220731 - Jenifer Archila

Informe de Desarrollo

Problemas durante la implementación

Soluciones

Conclusiones

Informe de Desarrollo

Problemas durante la implementación

Durante el desarrollo del código, hubo un par de partes lógicas que costó hacer que funcionaran bien:

- El menú se cerraba solo: Al principio de la práctica, el programa solo corría una vez.
- Error con la unificación de los premios, siendo especial y el normal para que entre ambos tuvieran un porcentaje del 40%.
- Al tener tantos ciclos (do-while y while) corriendo al mismo tiempo para validar, enviar a imprimir algunos datos como "datos incorrectos" o "posición inválida" se perdían en la pantalla, se imprimían varias veces de golpe o aparecían donde no debían.

Soluciones

- *Ciclo en el menú principal:* Para que el juego no se cerrara, se envolvió todo el menú adentro de un ciclo do-while. Ahora el programa evalúa constantemente si la opción elegida es diferente de 3.
- Se utilizó la función Math.max(1, ...) para obligar al programa a que mínimo pusiera 1 premio especial. Finalmente, a los premios normales se les asignó el resto de esa resta matemática.
- Se separaron las validaciones en ciclos do-while independientes (uno para el tamaño del mapa y otro para la posición del personaje). Además, se ajustaron los System.out.println para que las alertas de error solo se impriman justo después de que el programa lo detecte.

Conclusiones

- Manejar bien los ciclos do-while es clave no solo para validar que el jugador ingrese las letras o números correctos, sino para evitar que la consola se llene de mensajes repetidos que confunden a la hora de jugar.
- Al trabajar con porcentajes en Java, mezclar decimales (double) y enteros (int) puede dar problemas de redondeo o dejar variables en cero. Apoyarse en funciones matemáticas como Math.max() es una forma segura de garantizar que siempre aparezca al menos un objeto especial en la matriz sin arruinar el total del porcentaje.
- Intentar programar toda la lógica (pedir datos, hacer cálculos e imprimir) en un solo bloque de código hace que encontrar errores sea casi imposible. Separar cada validación en su propio ciclo hace que el programa sea más predecible y mucho más fácil de arreglar.