|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NOMBRE | DESCRIPCION | RESULTADO | FALLA  EN | POSIBLE SOLUCION | AUTOR FUNCION | AUTOR PRUEBAS |
| Mover | Permite desplazarse entre salas | OK |  |  | CARLOS LI HU | ANTONIO AMOR |
| Coger | Permite coger objetos | ERROR | extractOB |  |
| Dejar | Permite dejar objetos | ERROR | extractOB |  |
| Encender | Permite encender objetos | OK |  |  |
| Apagar | Permite apagar objetos | OK |  |  |
| Abrir\_puerta | Permite abrir puertas | ERROR | Room\_get\_  exit\_dooor |  |
| Talk\_obj | Permite hablar con objetos | ERROR | Get\_room\_list |  |
| Talk\_pers | Permite hablar con personajes | OK |  |  |
| Examinar-ob | Permite examinar objetos | ERROR | Get\_room\_list |  |
| Examinar-es | Permite examinar espacios | OK |  |  |
| Dormir | Permite dormir | OK |  |  |
| Despertar | Permite despertar | ERROR | Get\_room\_list |  |

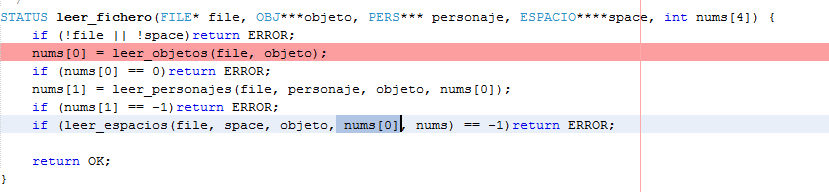
En un primer momento el cuadro de pruebas tenía la siguiente configuración:

Sin embargo si observábamos como funcionaban las funciones en las que el código colapsaba era fácil reconocer un factor común en ellas. Había un problema con las listas y este problema estaba localizado en el lector de datos del programa.

Hubo que cambiar en lect.c lo siguiente:

* En leer\_fichero: a la hora de leer que objetos contiene:

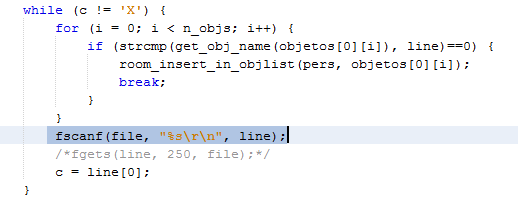
Se cambió num[1](número de personajes por el seleccionado en azul(ya corregido) número de objetos(obtenido de la línea seleccionada en rosa)



* En leer\_espacio: a la hora de leer que objetos contiene:

Se cambió fgets por el fcanf seleccionado en azul(ya corregido)

Gracias a eso el strcmp funciona correctamente y se introducen los objetos en los espacios.



Con estos cambios realizados obtenemos la siguiente configuración de la tabla de pruebas:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NOMBRE | DESCRIPCION | RESULTADO | FALLA  EN | POSIBLE SOLUCION | AUTOR FUNCION | AUTOR PRUEBAS |
| Mover | Permite desplazarse entre salas | OK |  |  | CARLOS LI HU | ANTONIO AMOR |
| Coger | Permite coger objetos | OK |  |  |
| Dejar | Permite dejar objetos | OK |  |  |
| Encender | Permite encender objetos | OK |  |  |
| Apagar | Permite apagar objetos | OK |  |  |
| Abrir\_puerta | Permite abrir puertas | OK |  |  |
| Talk\_obj | Permite hablar con objetos | OK |  |  |
| Talk\_pers | Permite hablar con personajes | OK |  |  |
| Examinar-ob | Permite examinar objetos | OK |  |  |
| Examinar-es | Permite examinar espacios | OK |  |  |
| Dormir | Permite dormir | OK |  |  |
| Despertar | Permite despertar | OK |  |  |

Sin embargo quedan cuestiones por corregir como:

Al pasar parámetros incorrectos a las funciones algunas siguen dando OK aunque trantan bien el error, lo cual puede derivar en problemas si la función llamante tiene codificado el return.

Es el caso de abrir\_puerta y talk\_pers

El resto detectan y tratan bien los errores.

En lo que respecta a la homogeneidad del código esta es buena. Todas las funciones se escriben con la misma estructura. Sin embargo, el código está muy compacto y eso puede dificultar su lectura.

La mayoría de las funciones tienen una cabecera que permite entender que hace cada una de las líneas de código pero también hay unas pocas funciones de gran complejidad en las que recomiendo comentarios en ciertos momentos del código para el mejor seguimiento de la función.

Por ultimo comentar el control de errores, salvo en las dos funciones comentadas arriba se realiza un buen control que evita violaciones de segmento…

NOTA: las funciones tienen fprintf cosa que nos dijo el profesor que no era recomendable.