75.41 95.15 95.12 Algoritmos y Programación II Curso 4 TP2 - Sala de escape Pokemon

14 de junio de 2022

1. Introducción

2. Desarrollo del trabajo

Vamos a reutilizar la mayor parte del TP1 implementado al principio del cuatrimestre. Para esta segunda parte se pide incorporar algunos cambios y explicar algunas cuestiones detalladas en los siguientes items:

- La definicion de la estructura de la sala fue movida al archivo sala.c.
- Haga las modificaciones que crea nacesarias a la estructura sala, de modo tal que se utilicen los TDA implementados a lo largo del cuatrimestre.
- Explique y justifique por qué utiliza los TDA seleccionados para cada caso.
- ¿Falla alguna prueba al realizar estos cambios? ¿Por qué?
- En caso de que alguna prueba falle, muestre como arreglarla.
- La definicion del enumerado de acciones fue movido al .h de sala. Recuerde que este archivo es el único archivo de cabecera visible pare el usuario de la biblioteca (el único archivo que puede ser incluído desde el o los archivos que implementen la lógica del juego.)
- Se agrega al enumerado el tipo de acción **ESCAPAR** que corresponde al caracter 'g' en los archivos de acción.
- Implemente las nuevas funciones agregadas a sala.h
- Implemente en el archivo escape_pokemon.c la lógica del juego (utilizando la sala).
- Si lo necesita, puede crear archivos extra dentro de src para separar funcionalidades. Recuerde que NO está permitido incluir los archivos de cabecera privados. Solamente está permitido hacer include de sala.h.

2.1. El juego

El juego a desarrollar debe presentar una interfaz basada en texto donde el jugador pueda ingresar comandos y así dirigir cada una de las acciones del protagonista. Existen 4 comandos básicos que deben ser implementados:

- ayuda: Muestra un mensaje de ayuda para recordarle al jugador los comandos disponibles.
- agarrar ¡objeto¡: Intenta agarrar alguno de los objetos conocidos de la sala.
- describir ¡objeto;: Describe uno de los objetos conocidos de la sala.
- salir: Sale del juego.

Además de estos comandos, se deben aceptar comandos a determinar por el archivo de interacciones que van a permitir interactuar con los diferentes objetos de la sala. Por ejemplo:

- examinar habitacion: Si el verbo examinar existe para la habitación, se ejecuta la acción especificada en el archivos de interacciones.
- usar llave cajon: Si el verbo usar existe para la llave con el objeto cajon, se ejecuta la acción especificada en el archivos de interacciones.

El formato de estas interacciones queda a determinar por el alumno y debe estar documentado tanto en el informe como en la ayuda. Por ejemplo se podrían agregar artículos para hacer la interacción mas natural (**usar la llave en el cajón**), pero no es obligatorio. Para el caso de interacciones que involucran 2 objetos, es necesario que el primer objeto esté en posesión del jugador.

Si un verbo figura mas de una vez en el archivo de interacciones para un objeto, se deben ejecutar todas las interacciones posibles (preferiblemente en orden de aparición, pero no es obligatorio que así lo sea).

Al iniciar el juego, el jugador sólo conoce el primer objeto que aparece en el archivo de objetos. Interactuando con este objeto es que se logran descubrir mas objetos con los cuales se debe interactuar para poder lograr escapar de la sala.

2.2. Acciones

Cada una de las acciones posibles presentes en el archivo de interacciones produce un efecto diferente. Cada una de las acciones tiene asociada un mensaje, este mensaje debe ser mostrado por pantalla luego de completada la acción de alguna forma a determinar por la implementación. Adicionalmente cada acción realiza una tarea específica en la sala.

- MOSTRAR_MENSAJE: No hace nada (además de mostrar el mensaje, claro).
- ELIMINAR_OBJETO: Elimina el objeto de la sala (ya sea que esté en posesión del usuario o no).
- DESCUBRIR_OBJETO: Si el objeto no es conocido, lo hace conocido.
- ESCAPAR: Cambia el estado de la sala escape exitoso (básicamente ganaste).
- REEMPLAZAR_OBJETO: Elimina el objeto parámetro de la sala y descubre uno nuevo (por ejemplo el comando usar llave cajon reemplaza cajon con otro objeto) (odio esta acción pero como ya la había puesto en el TP1, queda).

2.3. Consejos

Se recomienda, como siempre, elaborar pruebas que ayuden a verificar el correcto comportamiento de cada una de las funciones implementadas. Las pruebas de las nuevas funciones de **sala** son obligatorias (y son las que corre **chanutron**), las pruebas del resto de las funciones que no pertenecen a sala pero implementan lógica adicional como manejo de menues o entrada y manipulación de texto son opcionales y corren por cuenta del alumno.

Tenga en cuenta que al interactuar con un juego basado en texto es molesto distinguir entra mayúsculas y minúsculas o espacios múltiples. Intente que la implementación sea lo suficientemente permisiva como para que el jugador no se frustre al tener que volver a escribir todo nuevamente porque puso un espacio de más o porque escribió una palabra en minúsculas y se esperaban minúsculas, etc.

Es conveniente también que existe alguna forma de verificar la información conocida en cualquier momento (objetos disponibles para interactuar, objetos en posesión del usuario, cualquier otra información útil). La presentación de estos datos es optativa pero se recomienda incorporarlos de alguna forma al trabajo. Se puede agregar un comando para mostrar esta información o se puede dibujar en pantalla de alguna forma.

Se recomienda también agregar una pantalla de inicio con la ayuda y/o una breve descripción del juego, su objetivo y cómo jugar. Recuerde que parte de la nota de este trabajo viene de parte de la experiencia que su corrector tenga al interactuar con el juego. Mientras menos información se tenga o mas difícil sea interactuar con el juego, mas difícil va a ser obtener un mayor puntaje.

Por último, se aconseja incorporar por lo menos un escenario extra al escenario de ejemplo que viene con el enunciado.

2.4. Entrega

Una vez completada la implementación, Hay que subirla al sistema de entregas, para verificar su correcto funcionamiento. Además de la implementación, se debe entregar un informe (ya sea en formato PDF o video de 5 minutos o menos). En el informe **tenés que explicarle a tu corrector** cómo se compila y corre tu programa y se deben discutir las cuestiones pedidas en la sección **Desarrollo del trabajo**. Adicionalmente se deben agregar todas las consideraciones que sean importantes para el corrector. **El informe es parte de la nota**.