

ALGO 2 MENDEZ

75.41 Algoritmos y Programación II Curso 4

TP2 - La Venganza De Arya

15 de abril de 2019



Fecha de presentación: 15/04/2019

1. Introducción

(SPOILER ALERT)

Arya Stark es la segunda de cinco hermanos, hijos de Lord Eddard Stark y su esposa, Lady Catelyn Stark. Tiene una hermana, Sansa; tres hermanos, Robb, Brandon y Rickon, y un medio-hermano bastardo, Jon Nieve, al que aprecia mucho. Como algunos de sus hermanos, ella es una cambiapielos, pudiendo entrar en la piel de su loba huargo, Nymeria.

Es una niña voluntariosa, independiente, testaruda y con interés en las actividades típicamente masculinas como el combate y la exploración, a menudo la confunden con un niño. No tiene una buena relación con su hermana Sansa, cuyos intereses son tradicionalmente femeninos, actividades en las que Arya no tiene habilidades.

Es animada en sus gustos por su medio-hermano Jon, con quien tiene una relación cercana. A través de sus viajes, Arya demuestra gran ingenio, astucia y la capacidad firme de aceptar las necesidades, por duras que sean. Se dice que su temperamento es similar al de su tía Lyanna.

Maneja una espada llamada *Aguja*, un regalo de su medio hermano, Jon Snow, y está entrenada en el estilo Braavosi de lucha de espadas. A lo largo de sus viajes, Arya muestra gran ingenio y astucia.

Las situaciones dolorosas que le tocaron vivir a lo largo de su corta vida (como la ejecución de su padre o presenciar la masacre que ocurrió en la boda roja) hicieron que se transforme en una persona cada vez más despiadada.

Luego de salvarle la vida a un asesino llamado Jaqen H'ghar, este se ofrece a llevar a Arya a Braavos para unirse al Gremio de *Asesinos de Hombres Sin Rostro*.

Una vez allí, es aceptada en la Casa de Blanco y Negro y comienza a entrenar para convertirse en una asesina, quien toma los rostros de sus víctimas como trofeo y los utiliza como una herramienta de infiltración en sus próximas misiones.

Suele ocultar su verdadera identidad y ser conocida por muchos nombres. Entre los que ha utilizado están:

- Arya de la Casa Stark, su verdadero nombre
- Arya Caracaballo, un apodo dado por Jeyne Poole en referencia a rostro alargado.
- Arya Entrelospíes, usado por las personas de Invernalía en referencia a su hábito de estar siempre en el camino de la gente, en vez de donde una dama debe estar.
- Arry, un huérfano del Lecho de Pulgas mientras viaja con Yoren.
- Chichones, apodo dado por los niños reclutados por Yoren.
- Comadreja, una sirvienta de Harrenhal, toma el nombre de la niña que había viajado con ella hasta allí.
- Nymeria o Nan, la copera de Roose Bolton.
- Salina, viajera a bordo de la Hija del Titán de camino a Braavos.
- Nadie

Repite los nombres de las personas que quiere muertas todos los días antes de dormir. Algunos se agregan sobre la marcha, mientras otros son abandonados por la muerte de la persona o porque se ha desarrollado una relación con ella. Lo llama plegaria y termina diciendo las palabras Valar Morghulis. Ellos son:

- Rey Joffrey, quien ordenó la ejecución de su padre.
- El Perro, quien mató a su amigo Mycah y se rió de ello.
- Ser Amory Lorch, quien atacó a su grupo, matando a Yoren e impidiendo que llegara a Invernalía.
- Ser Gregor Clegane, por las atrocidades de su banda.
- Ser Ilyn Payne, quien decapitó a su padre.

- Ser Meryn Trant, quien presume que asesinó a su maestro de danza, Syrio Forel.
- Reina Cersei, quien estuvo detrás de la caída de su familia.
- Dunsen, soldado de Ser Gregor que robó el casco de Gendry.
- Raff el Dulce, soldado de Ser Gregor, quien mató a Lommy Manosverdes.
- Los Frey, por traicionar y asesinar a su madre y a su hermano mayor. Arya quería agregarlos a la lista, pero no sabía sus nombres.
- Entre otros...

2. Enunciado

Aclaración: Las acciones que se deban realizar a lo largo de esta actividad son ajenas a la moral de la cátedra. Este trabajo es sólo una representación de una obra de ficción basada en la época medieval.

En esta oportunidad se pide implementar un juego que simule el intento de Arya por eliminar a todas las personas que aparecen en su *Lista de Venganza*.

Se contará con una serie de víctimas, éstas le han causado algún daño a Arya, con lo cual su muerte está justificada.

Para ganar el juego se deben eliminar a todas las personas de dicho listado, sin embargo, para poder asesinar a cada uno de ellos, es necesario asesinar a los que aparecen antes que este en la nómina de la joven Stark.

Por ejemplo, si se desea asesinar de la tercer persona del listado, antes se debe haber asesinado a la primera y a la segunda.

Una de las habilidades de nuestra heroína es que puede coleccionar los rostros de las personas que asesina, sirviéndole éstos para asesinar a otros.

En este sentido, algunas de sus víctimas solo podrán ser liquidadas si se cuenta con el rostro indicado.

Estas personas, aún sin haberle hecho algo a Arya, por transitividad, entraron en su lista de venganza, ya que las necesita para limpiar su alma.

Inicialmente, nuestra protagonista se encuentra en Braavos y sólo podrá avanzar de a una ciudad por vez.

Una vez que se avanza, no puede retroceder. Dentro de cada una de estas ciudades se encuentra una única persona.

Cada persona puede estar en la nómina de Arya o no. En caso de no estarlo, esa persona puede ser inocente o culpable de algún crimen (ajeno a nuestra incumbencia) y al asesinarla puede beneficiar al jugador o perjudicarlo. Así se abren dos opciones:

- **Si era inocente:** se le restarán puntos de vida al usuario (indicados en el campo dano del correspondiente registro).
- **Si era culpable:** se le otorgará alguno de los siguientes beneficios:
 1. Aumenta en 10 los puntos de vida del jugador.
 2. Remueve a la próxima víctima del listado, excepto si se trata de Walder Frey, la última víctima de Arya.
 3. Otorga al usuario la llave del castillo de Walder Frey. Sin esta no se podrá eliminar a dicho Lord.

Sólo los inocentes restan puntos de vida.

Si los puntos de vida del jugador llegan a 0, pierde.

Si el usuario se encuentra frente a una de las personas del listado pero no asesinó a quien la precedía en dicha

lista, sus puntos de vida se reducen a 0 y pierde, puesto que Arya no cuenta con todos los rostros necesarios para infiltrarse con éxito y es ejecutada.

Si el usuario llega al castillo Frey pero no tiene la llave, también pierde ya que no puede entrar a acabar con la vida del Lord.

3. Objetivo

El presente trabajo práctico tiene como finalidad que el alumno se familiarice con las estructuras de datos dinámicas creadas, aplicadas a un caso de la vida cotidiana poniendo en práctica su uso.

4. Interacción con el usuario

El programa debe comunicarse con el usuario (y viceversa) a través de un Menú con las siguientes opciones:

- **I**: muestra la información de la persona que se encuentra en la ciudad actual.
- **M**: lista la ciudad actual y aquellas que no fueron visitadas.
- **A**: avanza a la próxima ciudad.
- **P**: muestra por pantalla la próxima persona en el listado.
- **C**: muestra por pantalla los nombres de todos los rostros recolectados por Arya hasta el momento.

Si la letra ingresada no corresponde a las del menú, deberá volver a pedirse hasta que sea válida. Por cada letra inválida ingresada, Arya perderá un punto de vida (porque la época medieval era muy cruel, aún con una jovencita como Arya).

Al momento de entrar a cada ciudad, se debe mostrar la ciudad a la que se llegó y el nombre de la persona disponible para asesinar.

```
1 Has llegado a <Nombre de la Ciudad>...
2 Ante tus ojos se encuentra <Nombre de la persona encontrada>.
```

Si el usuario ingresa la opción **I** se mostrará el nombre y la descripción de la persona en la ciudad actual. Acto seguido se le preguntará si desea terminar con su vida o perdonarla, el usuario deberá ingresar S (sí) o N (no), cualquier otra letra es inválida y deberá volver a solicitársele que ingrese otra. Por cada letra inválida ingresada Arya perderá un punto de vida (porque sí, porque sí, porque sí, porque mientras te espero por ti me muerdo).

```
1 <Nombre de la persona encontrada>: <Descripción de la persona>.
2 Deseas terminar con su vida? (S/N)
```

Ante cualquier decisión, debe mostrarse el mensaje correspondiente a la persona, en caso de decidir terminar con su vida, el mensaje de muerte, en caso de no matarlo, el mensaje de perdón.

En caso de llegar a una ciudad donde la persona debe ser ejecutada, pero no es la próxima del listado de Arya, deberá mostrarse un mensaje acorde a la situación, terminando el juego.

Se deja a criterio del alumno como mostrar al usuario los demás comandos.

Queda también a criterio del alumno la forma de mostrar la información del estado del juego al usuario, sin embargo debe estar a cada momento visible la **vida** restante. Se sugiere que se muestren más datos, como la cantidad de ciudades que quedan por visitar, si se tiene la llave o no, cantidad de víctimas que aun no han sido ejecutadas, y todo aquello que crea pertinente.

Asimismo al finalizar el juego debe informarse el resultado, y en caso de haber perdido, el motivo.

5. Implementación

El presente trabajo consiste en desarrollar la biblioteca venganza_arya (.c y .h). Para la implementación de dicha biblioteca deberán valerse de las estructuras realizadas anteriormente, en particular deberán utilizar:

- El TDA Pila para almacenar las víctimas del listado de Arya.
- El TDA Lista para almacenar las ciudades que recorrerá en su búsqueda de venganza.

- El TDA Lista para almacenar las caras recolectadas por Arya.

A su vez, se pide implementar el TDA Iterador para recorrer de forma segura, limpia y abstracta las ciudades.

Se contará con 2 archivos de entrada:

- **victimas.txt**: Es un archivo de texto donde se encuentran, línea por línea, todas aquellas personas que forman parte del listado de venganza de Arya. Esta información debe ser volcada a la pila, y estarán por orden de importancia.
- **ciudades.dat**: Es un archivo binario con registros del tipo **ciudad_t** contiene la información de todas las ciudades que recorrerá Arya y dentro de cada ciudad se encuentran el nombre de la ciudad, el nombre de la persona factible de asesinar, la descripción de la persona, los mensajes en caso de terminar con su vida o no, si está en nuestro listado o no, si es inocente o no y el tipo de beneficio.

Ejemplo de archivo **victimas.txt**:

```
1 Lord_Walder_Frey
2 Lady_Elizabeth_Bennet
3 Lord_Launderdale
4 Ser_Ronald_McDonald
5 The_Waif
6 Ser_Meryn_Trant
7 ...
```

Nótese que al ser Lord Walder Frey el más importante para Arya está primero en el listado.

Registro **ciudad_t**:

```
1 typedef struct persona {
2     char    nombre[50];
3     char    descripcion[500];
4     char    msj_muerte[100];
5     char    msj_perdon[100];
6     int     en_lista;           // 1 si está, 0 en caso contrario.
7     int     culpable;           // 1 si lo es, 0 en caso contrario.
8     int     danio;              // daño proporcionado por una persona inocente.
9     int     beneficio;          // 1 aumenta vida, 2 elimina víctima, 3 llave del castillo.
10 } persona_t;
11
12 typedef struct ciudad {
13     char    nombre[50];
14     persona_t posible_victima;
15 } ciudad_t;
```

Registro **jugador_t**:

```
1 typedef struct jugador {
2     int     vida;               //inicialmente 100
3     bool    posee_llave;
4     lista_t* rostros;
5     pila_t*  victimas;
6 } jugador_t;
```

Funciones públicas a desarrollar en la biblioteca **venganza_arya**:

```
1 /*
2  * Crea un jugador y deja la lista de rostros preparada para ser utilizada.
3  * Devuelve NULL si no pudo crear algo de lo necesario.
4  */
5 jugador_t* nuevo_jugador();
6
7 /*
8  * Carga las ciudades del archivo a la lista de ciudades a utilizar.
9  * Devuelve 0 si la lista se cargo correctamente, -1 en caso contrario.
10 */
11 int cargar_ciudades(char archivo[50], lista_t* ciudades);
12
13 /*
14  * Carga las victimas del archivo a la pila de víctimas a utilizar.
15  * Devuelve 0 si la lista se cargo correctamente, -1 en caso contrario.
16 */
17 int cargar_victimas(char archivo[50], pila_t* victimas);
```

```

18
19 /*
20  * Luego de asesinar una persona, se deben actualizar los registros según corresponda.
21  * Recolectar el rostro, desapilar si es una víctima, actualizar el estado del jugador.
22  * Devuelve 0 si se pudo actualizar correctamente o -1 en caso contrario.
23  */
24 int actualizar_juego(jugador_t* jugador, persona_t persona);
25
26 /*
27  * Pasa a la siguiente ciudad.
28  * Devuelve 0 si se pudo avanzar o -1 en caso contrario.
29  */
30 int avanzar_mapa(iterador_t* it_ciudades);
31
32 /*
33  * Imprime por pantalla la ciudad actual y aquellas que no fueron visitadas.
34  * Devuelve 0 si se pudo mostrar o -1 en caso contrario.
35  */
36 void mostrar_mapa(iterador_t* it_ciudades);
37
38 /*
39  * Imprime por pantalla los rostros colectados.
40  * Devuelve 0 si se pudo mostrar o -1 en caso contrario.
41  */
42 int mostrar_rostros_recolectados(iterador_t* it_rostros);
43
44 /*
45  * Imprime por pantalla el nombre de la próxima víctima.
46  * Devuelve 0 si se pudo mostrarlo o -1 en caso contrario.
47  */
48 int mostrar_proxima_victima(pila_t* victimas);

```

6. Compilación y entrega

Se espera que el trabajo sea compilado sin errores con la siguiente línea de compilación:

```
1 gcc *.c -Wall -Werror -Wconversion -std=c99 -o la_venganza_de_arya
```

El juego debe ser ejecutado y jugado sin problemas con el comando:

```
1 ./la_venganza_de_arya
```

Y al correrse con valgrind, debe poder jugarse sin problemas ni pérdida de memoria:

```
1 valgrind --leak-check=full --track-origins=yes --show-reachable=yes ./la_venganza_de_arya
```

El trabajo será entregado por la plataforma **Kwyjibo**, donde deberá obtener la etiqueta **successful**. El archivo .zip entregado debe contener:

- La biblioteca creada.
- El juego.c.
- Separadas por carpetas, las demás bibliotecas propias utilizadas (TDAs).
- Enunciado.
- Readme explicando como se compila, ejecuta y juega, agregando casos de prueba (como ganar y perder).
- Cualquier tipo de pruebas que hayan realizado (archivos de ciudades, archivos de víctimas, etc).

Se recomienda incursionar en el armado de un script **bash** para las pruebas de este trabajo. Se adjuntará un mini script donde se compila y ejecuta un programa y luego se simula el ingreso de datos por parte del usuario, obteniendo la salida del programa y se la compara con una esperada.

Referencias

https://hieloyfuego.fandom.com/wiki/Arya_Stark