

ALGORITMOS Y ESTRUCTURAS DE DATOS

2023/2

Proyecto 2

El presente documento entrega la descripción del escenario relacionado con el problema a resolver de la Unidad 2.

Objetivos

- Realizar el análisis de un problema y aplicar una solución considerando estructuras de datos no lineales y las operaciones comunes sobre ellas.
- Realizar la solución utilizando el lenguaje de **programación C++**.
- Realizar una correcta declaración de variables según corresponda.
- Realizar una correcta implementación de algoritmos que involucren decisiones y bucles.
- Realizar una correcta implementación de entrega de resultados por la salida estándar.

Requerimientos de entrega

El Software desarrollado debe cumplir con los siguientes requerimientos:

- El programa debe estar completamente funcional, con ausencia de errores y alertas (warnings).
- El programa debe estar bien documentado para lo cual se necesita:
 - La inclusión de un archivo README donde se explica la solución desarrollada considerando el formato recomendado para este tipo de archivos.¹
 - Presencia de comentarios que expliquen el código, considerando la descripción de estructura, clases, métodos y funciones.
- La inclusión de un archivo INSTALL donde se explica el procedimiento para instalar el programa (compilar).
- El programa debe encontrarse en un repositorio en su cuenta Github.

El programa debe ser implementado a través del uso del lenguaje de programación C++ la utilización de listas, árboles y grafos además de la entrada/salida de archivos y menús interactivos en un contexto práctico relacionado con el problema entregado.

¹ [Formato README](#)

The guardians battles

Lircay Hub está interesado en desarrollar un piloto de un juego para lo cual es importante realizar la validación de algunos conceptos asociados a este. El juego se llama "The guardians battles" que corresponde a un juego donde los guardianes son héroes con habilidades y destrezas únicas que protegen a las ciudades a lo largo de toda la tierra siendo estas los últimos asentamientos de la humanidad.

Las batallas permiten a los guardianes obtener reconocimiento, premios y viajar por el mundo con el propósito de entrar al selecto grupo de los guardianes del reino. Solo existen 3 puestos para ser guardianes del reino los cuales siguen las órdenes de Stormbreaker, the judge. Específicamente lo que se necesita es trabajar activamente en la sección de duelos o batallas, recorrido por el mundo abierto e inventario de ítems.

Existen tres aspectos generales a completar en la implementación:

- **Los guardianes y su posicionamiento para aspirar a los guardianes del reino:** Los guardianes constantemente luchan para mejorar sus habilidad y poder escalar por sobre otros para llegar a formar parte de uno de los 3 puestos más importantes.
- **El reino y los viajes:** Un guardián puede ser aprendiz y maestro al mismo tiempo, esto definido por la jerarquía que existe entre ellos. A través del reino y viajando los guardianes tiene la posibilidad de enfrentarse en duelos, obtener recompensas y mayor conocimiento para subir en la jerarquía.

En el juego debemos realizar la implementación de:

- **Guardar la lista de candidatos:** El ranking es la posición en la cual se encuentra un guardián la cual puede cambiar una vez finalizado el resultado de un combate.
- **La jerarquía de los guardianes:** La relación de maestro y aprendices que puede existir entre ellos.
- **El mapa del mundo:** La relación que existe entre los guardianes y las ciudades pertenecientes al reino y a los rivales que puede tener un guardián en cada uno de sus viajes.

Las fuentes de información son:

- Carga inicial de información de guardianes y ciudades.
- Carga de información de resultados de los viajes y las batallas.
- Opciones que puede seleccionar el usuario para validar el funcionamiento. Opciones a través de un menú para validar y realizar cambios.

Desarrollo de un producto de Software

El objetivo es crear un **programa en C++** que permita realizar la simulación de cómo se trabaja y actualiza esta información en el juego considerando lo señalado en la sección anterior. El programa debe contener:

- Desarrollar la declaración de estructura (s) para trabajar con los guardianes. Donde es relevante considerar la jerarquía y las ciudades.
- Desarrollar la carga inicial de información en la estructura de datos correspondiente para trabajar con ellas a través de la lectura de archivos de configuración. Como la información proviene de archivos, es importante considerar que la validación de los archivos debe poder adaptarse a un archivo distinto al entregado con ayuda de memoria.
- Desarrollar un menú que permite navegar por las distintas opciones que tiene el programa. Dentro de las opciones deben estar;
 - Ver la lista de candidatos.
 - Ver al guardián.
 - Conocer el reino.
 - Presenciar una Batalla.
 - Salir.
- Desarrollar las funcionalidades que permitan trabajar con la información del ranking a través de un árbol binario según la problemática. También se deben considerar las operaciones sobre este tipo de estructura de datos como lo son las operaciones de recorrido, búsqueda y eliminación.
- Desarrollar las funcionalidades que permitan trabajar con la información de jerarquías a través de un árbol general según la problemática. También se deben considerar las operaciones sobre este tipo de estructura de datos como lo son las operaciones de recorrido, búsqueda y eliminación.
- Desarrollar las funcionalidades que permitan trabajar con la información de ciudades presentes en el reino a través de un grafo según la problemática. También se deben considerar las operaciones sobre este tipo de estructura de datos como lo son las actualización, búsqueda de aristas y detección de camino.
- Desarrollar la implementación a través de algoritmos que permita evidencia cambios en el ranking de los guardianes según el resultado de peleas, ítems y recompensas y el recorrido por el mundo. Las reglas aquí para ser implementadas son por cada menú, son:
 - **Ver la lista de candidatos:** Se puede revisar el estado actual del ranking de guardianes para lograr llegar a ser candidatos a guardianes del reino. Solo los guardianes que tienen entre 90 y 99 puntos son catalogados como caballeros candidatos.
 - **Ver al guardián:** Ver el detalle de un guardián donde se debe conocer toda la información relacionada con él.
 - **Conocer el reino:** Viajar a través del reino conocido, además se pueden realizar nuevos caminos y consultar por los existentes.
 - **Presenciar una batalla:** Seleccionar un guardián, viajar a una ciudad desde la que se encuentra y pelear con alguno de los campeones locales. Aquí se debe considerar los siguientes puntos dentro de la aplicación:

- El guardián retador es uno seleccionado de la lista de guardianes. La única restricción es que no pueden ser seleccionados el gran maestro y los guardianes del reino.
- El resultado de la batalla al vencer a un Maestro es de 5 puntos.
- El resultado de la batalla al vencer a un aprendiz es de 3 puntos.
- Solo se puede enfrentar a los guardianes dentro de la ciudad.
- La probabilidad de ganar es a través de una proporción de 4 a 6 donde la probabilidad del guardián retador gane la pelea.
- Puede viajar a otras ciudades del reino como lo realiza en el **“Conocer el reino”**.
- **Al finalizar la batalla se debe ver reflejado en la lista de candidatos los cambios según el resultado.**

Presentación de la solución

La entrega del proyecto debe ser acompañada por un video, de no más de 5 minutos en los que se debe:

- Desarrollar un material de apoyo (presentación) que sirva como guía para el video. Debe incorporar la portada además de:
 - Mencionar el problema a resolver.
 - Exponer los aspectos claves de su solución, en este caso las estructuras, listas y funciones más importantes.
- Mostrar parte de la jugabilidad. Aquí puede realizar cortes en la grabación para que pueda presentar todo en los 5 minutos señalados.

Entregable

El entregable corresponde al link de acceso público al repositorio de la cuenta de github.
En el archivo README del repositorio debe incorporar el LINK para el video.

Porcentajes y ponderación

#	Indicadores	Ponderación
1	Realiza la gestión del proyecto a través de un repositorio y presenta a lo menos cuatro registros de commits realizados durante el desarrollo.	2%
2	Realiza una correcta descripción del Software junto con la utilización del formato recomendado para la elaboración del archivo README.	2%
3	Realizar una correcta descripción de las instrucciones a realizar para una correcta instalación y/o compilación del producto de Software.	2%
4	Desarrolla una solución con ausencia de errores y alertas.	3%
5	Desarrolla de manera correcta comentarios en la estructura(s) y funciones.	2%
6	Desarrolla la declaración de estructura (s) para trabajar según la problemática.	3%
7	Desarrolla la función que entrega la posibilidad de realizar la carga inicial de información.	3%
8	Desarrolla la funcionalidad que entrega la posibilidad de seleccionar las opciones a través de un menú.	2%
9	Desarrolla la funcionalidad que permite trabajar con la creación de un árbol binario según la problemática.	3%
10	Desarrolla la funcionalidad que permite trabajar con la creación de un árbol general según la problemática.	3%
11	Desarrolla la funcionalidad que permite trabajar con la creación de un grafo general según la problemática.	3%
12	Desarrolla la funcionalidad de impresión del contenido de los árboles y grafos según la problemática.	5%
13	Desarrolla la implementación a través de funciones de las operaciones de; recorrido, búsqueda y eliminación de un árbol binario.	12%
14	Desarrolla la implementación a través de funciones de las operaciones de; recorrido, búsqueda y eliminación de un árbol general.	12%
15	Desarrolla la implementación a través de funciones de las operaciones de; actualización y búsqueda de aristas en el grafo.	8%
16	Desarrolla la funcionalidad que permite determinar si existen un recorrido entre dos puntos del grafo.	12%
17	Desarrolla la implementación a través de algoritmos de los cambios en los datos (pelea) y la actualización de estos en las estructuras de datos correspondientes.	15%
18	Realizar la construcción de material multimedia (video) y material de apoyo para ser expuesto en el video.	3%
19	Desarrolla una exposición exposición de la solución desarrollada utilizando un lenguaje técnico y formal	5%
	TOTAL	100%

Archivo de Ejemplo Ciudades - cities.conf

```
City,ConnectedCity
Silverstone City, Valley City
Silverstone City, Storm City
Silverstone City, Forest City
Valley City, Storm City
Valley City, Forest City
Storm City, Forest City
Capital City, Mountain City
Capital City, Desert City
Capital City, Cliff City
Mountain City, Desert City
Mountain City, Cliff City
Desert City, Cliff City
Coast City, Island City
Coast City, River City
Coast City, Hill City
Island City, Coast City
Island City, Hill City
River City, Hill City
Sunflower Fields City, River City
Tesla City, Forest City
```

Archivo de Ejemplo Guardianes - guardians.conf

```
Name,PowerLevel,Master,City
Freya,100,None,Tesla Capital
Skadi,97,Freya,Tesla Capital
Magni,92,Freya,Tesla Capital
Forseti,96,Freya,Tesla Capital
Freyr,73,Magni,Storm City
Angrboda,72,Freyr,Cliff City
Thorin,88,Skadi,Sunflower Fields City
Heimdall,84,Skadi,Desert City
Baldr,71,Angrboda,River City
Eirik,74,Heimdall,Hill City
Frigg,51,Baldr,Island City
Surt,63,Skadi,Capital City
```

Nanna,60,Baldr,Hill City
Valkyrie,74,Heimdall,Coast City
Bragi,83,Heimdall,Desert City
Fenrir,73,Bragi,Mountain City
Loki,68,Valkyrie,Coast City
Mjolnir,71,Angrboda,Capital City
Odin,62,Fenrir,Mountain City
Hermod,82,Bragi,Sunflower Fields City
Ragnar,58,Hermod,Coast City
Balder,62,Skadi,Hill City
Hel,57,Ragnar,Hill City
Vidar,53,Fenrir,Silverstone City
Eir,58,Fenrir,Coast City
Sol,57,Eirik,River City
Idunn,73,Skadi,Forest City
Hoenir,50,Eirik,Desert City
Aegir,50,Skadi,Capital City
Njord,49,Aegir,Forest City
Ullr,62,Surt,Mountain City
Sigurd,73,Eirik,Storm City
Tyr,59,Odin,Capital City
Gefjon,48,Njord,Valley City
Sif,62,Surt,Sunflower Fields City