

Indicaciones específicas:

- Esta evaluación contiene 3 páginas (incluyendo esta página) con 1 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta.
 - p1.cpp
- Deberás subir estos archivos directamente a www.gradescope.com, uno en cada ejercicio. También puedes crear un .zip

Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	20	
Total:	20	

1. (20 points) **Sistema Jugador vs. Jugador de un Videojuego**

Eres un desarrollador de videojuegos trabajando en un juego de enfrentamientos. Tu tarea es implementar un sistema para gestionar los jugadores en línea, lucha, emparejamiento y rankings. Utilizando Programación Orientada a Objetos, Estructuras de Datos, Pilas/Colas y Patrones de Diseño implemente las siguientes características del videojuego:

- La creación de un personaje debería realizarse de acuerdo a su estilo de ataque:
 - Cuerpo a cuerpo: atacan con hachas o espadas, y usan armaduras pesadas (de malla o acero).
 - A distancia: atacan con flechas o rifles, y usan armaduras ligeras (de tela o cuero).
- El campo de batalla es el lugar donde se dan los enfrentamientos entre dos jugadores. El jugador que gana se lleva *puntos de batalla* (que depende directamente proporcional al nivel de los personajes) los cuales le permiten subir de nivel poco a poco. Al jugador que pierde, se le quita puntos de batalla—directamente proporcional a su nivel. Los campos de batalla deberían estar en un grafo.
- El emparejamiento se da la siguiente manera. Si un jugador busca una partida, se ubica al jugador en un campo de batalla de su nivel y espera hasta que otro jugador del mismo nivel también busque una partida. Cuando en un campo de batalla existen dos jugadores, el enfrentamiento comienza. Si otros jugadores del mismo nivel buscan una partida, tendrán que esperar hasta que el enfrentamiento acabe. Sin embargo, podrían realizarse varios enfrentamientos de distintos niveles al mismo tiempo.
- Justo antes de empezar el enfrentamiento, cada jugador tendrá la opción de elegir el conjunto de armas y armaduras con las que enfrentará a su oponente.
- Los puntos de batalla y el nivel del personaje se deberían de actualizar automáticamente cada vez que el personaje termina una batalla. Cada jugado debería poder ver su historial de enfrentamientos. Indicando con quien se enfrentó y si ganó o perdió puntos de batalla.

Requisitos básicos del programa:

- Integrar al menos dos patrones de diseño. El estudiante puede elegir el escenario en el cual aplicar el patrón de diseño.
- Integrar al menos una pila o una cola en su implementación
- Integrar un grafo o árbol binario de búsqueda en su implementación.
- Implementar la interfaz de su programa en la función principal (-2pts).
- Comente las características y requisitos importantes de su programa (-1pts).
- Adjunte su código en un solo archivo .cpp (-1pt).

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Patrones de Diseño	Selección adecuada de los patrones que permiten la solución de algún problema de diseño con una adecuada justificación y descripción a través de un diagrama de clases (6pts).	Selección de un patrón que permite la solución de alguno de los problemas de diseño con una justificación y descripción a través de un diagrama de clases (4pts).	Selección de un patrón que permite la solución de alguno de los problemas de diseño pero no se tiene una justificación y descripción inadecuada del diagrama de clases (2pts).	Selección de un patrón pero no se justificó y descripción incorrecta del diagrama de clases. (1pts)
Pilas y Colas	Implementación correcta y completa de una pila y/o cola con operaciones básicas usadas de manera adecuada y relevante en un contexto práctico que integra los contenedores para resolver un problema complejo. (6pts).	Implementación mayormente correcta de una pila y/o cola con algunos errores menores. Uso de ambos contenedores en un contexto práctico con integración aceptable pero con algunas deficiencias. (4pts)	Implementación parcial de una pila y/o cola con varios errores. Uso de ambos contenedores en un contexto práctico pero con problemas significativos en la integración. (2pts).	Implementación incorrecta o incompleta de una pila y/o cola con la mayoría de las operaciones básicas fallidas. Uso inadecuado o irrelevante de los contenedores en un contexto práctico. (1pts).
Algoritmo de Grafos	Implementación correcta y completa de los algoritmos de grafos. Cada algoritmo se utiliza adecuadamente en un contexto práctico relevante, demostrando una clara comprensión de sus propósitos y aplicaciones. (8pts).	Implementación mayormente correcta y completa de los algoritmos de grafos. Los algoritmos se usan en un contexto práctico relevante pero con algunas deficiencias en la integración o aplicación. (5pts)	Implementación parcial de algoritmos de los grafos con varios errores. Los algoritmos se usan en un contexto práctico pero con problemas significativos en la integración y aplicación. (3pts).	Implementación incorrecta o incompleta de los algoritmos de grafos con la mayoría de las operaciones fallidas. Uso inadecuado o irrelevante de los algoritmos en un contexto práctico. (1pts).