

### 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](http://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) **Gestión de enemigos, navegación, y armaduras en un Videojuego**

Eres un desarrollador de videojuegos trabajando en un juego de aventura. Tu tarea es implementar un sistema para gestionar enemigos, navegación y el sistema de armaduras del personaje/personajes. Utiliza Programación Orientada a Objetos, Estructuras de Datos, Pilas/Colas y Patrones de Diseño para implementar las siguientes características del videojuego:

- Los enemigos deben crearse con atributos a elección del desarrollador. Lo importante es que cada enemigo tenga un nivel. Cuanto más nivel tenga el enemigo, sus atributos se incrementarán significativamente, haciendo que sea más difícil vencerlo. Almacene los enemigos en una estructura de modo que el jugador los enfrente en base a su nivel (de menor a mayor).
- La creación de un personaje puede realizarse con algunas armaduras por defecto: hombros, botas, pantalones, guantes, casco, pechera, etc. Las armaduras no son obligatorias, es decir, el jugador podría incluso jugar con un personaje sin ninguna armadura. Además el jugador tiene la opción de comprar armaduras especiales con monedas del juego o dinero real.
- Cada vez que el jugador derrote a un enemigo tendrá la opción de guardar la partida. Además el jugador podría alertar a los jugadores en línea de su logro (el hecho de vencer al enemigo).
- El mapa del videojuego debe ser un grafo, donde cada nodo será un lugar importante en el videojuego: un lugar para guardar partidas, la ubicación de un enemigo, o para conversar con un personaje amistoso del videojuego (guía). Dada una posición, el jugador debería poder elegir a que lugar desea ir, el destino debería ser un lugar cercano al lugar actual del personaje.

Requisitos básicos del programa:

- Integrar al menos dos patrones de diseño. El estudiante puede elegir el escenario en el cual integrar 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).