

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias
Departamento de Electrónica y Ciencias de la Computación

Parcial Práctico 2 2022-2
Curso Técnicas y Prácticas de Programación

Consideraciones para el parcial:

1. El parcial práctico 2 debe desarrollarse y presentarse en los equipos conformados al inicio del semestre.
2. Los estudiantes deben utilizar el lenguaje de programación C para desarrollar los puntos propuestos.
3. El profesor hará la presentación y explicación del parcial práctico 2 el miércoles 12 de octubre de 2022 y los estudiantes tendrán para desarrollarlo y entregarlo hasta el día viernes 14 de octubre de 2022 a las 6:00 PM.
4. Cada equipo tendrá un espacio para sustentar el parcial práctico 2 los días miércoles 19 de octubre de 2022 y lunes 24 de octubre de 2022 en el horario de la clase.
5. El desarrollo de los puntos propuestos se debe enviar en un **único** archivo .c al correo del profesor martinV.sierra@javerianacali.edu.co. Tener en cuenta lo siguiente:
 - a. Solamente debe enviar el archivo .c un integrante del equipo.
 - b. Los nombres de los integrantes del equipo deben aparecer al inicio del archivo en un comentario..

Puntos a desarrollar:

1. (2.5 puntos) El deck de cartas es uno de los ejes centrales del género Deck Builder. Con este deck, el jugador puede sortear los obstáculos que se le presenten durante su aventura. En este punto, el objetivo consiste en implementar el módulo encargado de mostrar las cartas que el jugador tiene a su disposición. Para ello se deben tener en cuenta los puntos a continuación:
 - a. Al iniciar, el programa debe mostrar un menú con las opciones **Caminar**, **Deck** y **Salir**. Esta última opción permite al jugador terminar el programa.

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias
Departamento de Electrónica y Ciencias de la Computación

- b. Si el jugador selecciona la opción **Caminar**, hay un 50% de probabilidad de encontrar una carta y un 50% de no encontrar nada. Si el usuario no encuentra nada, simplemente retorna al menú inicial.
 - c. Si el jugador encuentra una carta, el programa debe mostrarle la información de la carta y debe permitirle agregarla al deck o descartarla. Los estudiantes deben definir la información que tendrán las cartas. Esta debe incluir el nombre de la carta, la categoría a la que pertenece (ataque, defensa o habilidad) y una pequeña descripción.
 - d. Si el jugador decide agregar la carta al deck, el programa debe añadir la carta al conjunto de cartas del jugador y, posteriormente, regresar al menú inicial. Por otro lado, si el jugador decide descartar la carta, el programa debe simplemente regresar al menú inicial. El número de cartas que el jugador puede añadir a su deck es **ilimitado**.
 - e. Al seleccionar la opción **Deck** del menú inicial, el jugador puede ver la información de todas las cartas que ha recolectado. Si aún no tiene cartas, debe aparecer un texto indicando que el deck se encuentra vacío.
 - f. Cuando el jugador se encuentre en la opción **Deck**, el programa debe darle la opción de eliminar las cartas que ya no desee conservar. El jugador puede eliminar cuántas cartas desee.
 - g. Adicional al punto anterior, el programa debe darle la opción al jugador de regresar al menú inicial desde el **Deck**.
2. (2.5 puntos) Refactorizar el **Bestiario** del primer parcial para mostrar la información de los 30 enemigos que se piden en el proyecto del curso. Este **Bestiario** debe implementarse de forma tal que deje abierta la posibilidad de agregar un número **determinado** de enemigos a futuro.

Criterios de evaluación:

1. Cumplir con todos los requisitos solicitados en cada punto.
2. Notificar adecuada y detalladamente al usuario todo lo que sucede en el programa. Esto quiere decir que el contenido de los mensajes y la forma de

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias
Departamento de Electrónica y Ciencias de la Computación

mostrarlos deben ser muy claros para la persona que interactúa con el software.

3. Aplicar las buenas prácticas de programación vistas hasta el momento en el curso para tener un buen programa en C:
 - a. Buena indentación y organización del código.
 - b. Correcta definición de variables y constantes.
 - c. Correcto uso de constantes simbólicas.
 - d. Correcta declaración y uso de las estructuras de control.
 - e. Correcto uso de las cadenas de texto y la librería string.h.
 - f. Correcta definición y uso de funciones y sus prototipos.
 - g. Correcta definición y uso de **estructuras de datos**.
 - h. Correcto uso de **punteros** y de la **memoria dinámica**.
 - i. Correcta definición y uso de **arreglos dinámicos** y **estructuras dinámicas**.
 - j. Manejo de posibles problemas que puedan surgir a partir de la interacción del usuario con el programa.