

## Proyecto Semestre 2022-2

### Curso Técnicas y Prácticas de Programación

Implementar un Deck Builder Game en **consola**. Este género de videojuegos consiste en armar progresivamente un deck de cartas a medida que el jugador avanza por diferentes caminos en un mapa, siendo el objetivo principal sortear los obstáculos que se le presentan al jugador con ayuda de dicho deck. Normalmente, este género viene acompañado de elementos del género de Rol como bestiarios, combates por turnos, enemigos elite, jefes finales, estadísticas del jugador que dependen de un rol, mochila de ítems, entre otros. El proyecto busca que los estudiantes puedan conocer y analizar la lógica que hay detrás de estos videojuegos, por lo que se recomienda investigar títulos como **Slay the Spire**, de modo que sirvan como una base y apoyo para la implementación del mismo. A continuación se presentan los requerimientos principales de desarrollo que los estudiantes deben cumplir:

1. Al iniciar el videojuego se debe mostrar un **menú inicial** donde el jugador puede seleccionar las opciones **Juego Nuevo** o **Salir**. Si el jugador tiene una partida guardada previamente, este menú debe mostrar la opción adicional **Continuar**.
2. Si el jugador selecciona la opción **Juego Nuevo** en el menú inicial, el videojuego debe permitirle al jugador elegir un personaje de una lista de 3 opciones disponibles. Los estudiantes tienen la libertad de definir los nombres de estos personajes y el valor de sus estadísticas (**fuerza, dureza e inteligencia**). El personaje del jugador debe iniciar siempre con un deck de al menos 5 cartas.
3. Cuando la partida inicia, el videojuego debe mostrarle al jugador un **menú de juego** con las siguientes opciones:
  - a. **Caminar**: al seleccionar esta opción, el videojuego debe mostrarle al jugador los posibles caminos que puede tomar. Cada camino debe permitirle al jugador ver qué se va a encontrar si lo elige. Las opciones que se pueden tener para cada camino son: **batalla normal, batalla**

**contra un enemigo elite, batalla contra un jefe, sitio de descanso** (lugar en el que el jugador recupera el total de puntos de vida) y **tienda** (lugar donde el jugador puede comprar cartas). Cada camino solo debe tener una opción.

- b. **Deck:** al seleccionar esta opción, el videojuego debe mostrarle al jugador el total de cartas que ha coleccionado. Se debe mostrar el nombre de la carta y su respectiva descripción. El número total de cartas que puede tener un jugador es ilimitado; además puede tener cartas repetidas.
  - c. **Bestiario:** al seleccionar esta opción, el videojuego debe mostrarle al jugador una lista con todos los enemigos que ha derrotado en los diferentes combates; cada uno con su respectiva información. Esta información incluye nombre y descripción. Los estudiantes tienen la libertad de definir dichos valores para cada enemigo. El total de enemigos disponibles en el videojuego debe ser 30, incluyendo enemigos normales, enemigos de elite y jefes.
  - d. **Guardar:** al seleccionar esta opción, el videojuego debe guardar el progreso del jugador, ya sea en un archivo binario o en un archivo de texto.
  - e. **Salir:** al seleccionar esta opción, el videojuego termina y debe mostrarle al jugador el menú inicial.
- 4. Si el jugador selecciona **Continuar** en el menú inicial, el videojuego debe reanudarse en el último punto donde el jugador guardó la partida.
  - 5. Si el jugador selecciona **Salir** en el menú inicial, el videojuego termina completamente.

## Cartas

Las cartas son uno de los ejes centrales y representan los ataques y las habilidades que el jugador puede utilizar en los combates que vaya teniendo a lo largo del camino. Los estudiantes deben tener en cuenta que el total de cartas diferentes

disponibles en el videojuego debe ser 15 y que estas se dividen en tres categorías. Así mismo, los estudiantes tienen la libertad de definir los nombres, las descripciones, lo que hace cada carta en el combate y la categoría a la que pertenecen. Por último cada carta debe tener asignado un valor correspondiente al **gasto de poder** en cada turno del combate (esto se explica mejor en la sección Batallas).

Las tres categorías de cartas son:

1. **Cartas de ataque:** son cartas equivalentes a ataques físicos. El valor de daño que realizan en combate puede variar, influenciado por la estadística de fuerza del personaje.
2. **Cartas de defensa:** son cartas que aumentan la defensa física y mágica del jugador en combate, es decir, cuantos puntos de daño puede bloquear. Sus valores se ven afectados por la estadística de dureza del personaje.
3. **Cartas de habilidad:** son cartas equivalentes a ataques mágicos o magias de recuperación. Sus valores se ven afectados por la estadística de inteligencia del personaje.

## Batallas

Las batallas en este género se realizan por medio de turnos. En cada turno debe ocurrir lo siguiente:

1. En todos los turnos se debe mostrar:
  - a. La vida del jugador junto con el total de puntos de poder que puede gastar por turno.
  - b. El nombre y la vida del enemigo.
  - c. Las cartas que puede utilizar el jugador. Estas corresponden a un mini deck de 5 cartas que se eligen aleatoriamente del deck completo que tiene el jugador.
  - d. Una opción para terminar el turno del jugador.

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

2. Al principio de cada turno se debe mostrar la acción que realizará el enemigo y los efectos que tendrá dicha acción. Por ejemplo, si el enemigo decide atacar al jugador, se debe indicar que hará un ataque y mostrar el valor del daño de dicho ataque. De esta forma el jugador puede decidir qué hacer en cada turno. Los estudiantes tienen la libertad de definir las acciones que realizan los enemigos. Se debe tener en cuenta que dichas acciones se ejecutan de manera aleatoria.
3. El jugador debe seleccionar las cartas que usará en su turno. Esta selección está limitada por los puntos de poder que puede gastar el jugador en cada turno y los valores de gastos de poder asignados a cada carta. Por ejemplo: si los puntos de poder que puede gastar el jugador es 3 y este selecciona una carta cuyo gasto de poder es 2, entonces solo le quedaría un punto de poder para gastar. Cada vez que el jugador selecciona la carta, se debe ver reflejado el cambio en los puntos de poder y la carta debe removerse del mini deck del turno. Para el caso de este proyecto, el valor de puntos de poder que el jugador puede gastar en cada turno es 5.
4. Cuando un jugador selecciona una carta, el efecto de dicha carta se aplica inmediatamente. Por ejemplo: si la carta resta 4 puntos de vida al enemigo, estos 4 puntos deben restarse del enemigo al instante. Es recomendable que este efecto se muestre al jugador para que este sepa que se aplicó correctamente.
5. Cuando el jugador selecciona la opción de terminar turno, entonces el enemigo aplica la acción que había notificado anteriormente y que se describe en el punto 2 de esta sección. Una vez esta acción se realiza comienza el siguiente turno. La lógica se repite hasta que el combate termina.
6. El combate termina cuando el enemigo pierde todos sus puntos de vida o cuando el jugador pierde todos sus puntos de vida:
  - a. Si pierde el enemigo, se debe mostrar en la consola 3 cartas aleatorias de forma que el jugador pueda seleccionar **una** de ellas. Una vez el jugador complete esta selección, se debe regresar al menú de juego.
  - b. Si pierde el jugador, se debe regresar al menú inicial.

## Requerimientos adicionales y consideraciones

1. Las cartas y la lista de enemigos en el bestiario siempre deben mostrarse en **orden alfabético**.
2. La lógica de caminos debe estar implementada usando una **Estructura Dinámica**. Es importante recordar que el jugador tiene la opción de seleccionar varios caminos.
3. Todo aquel elemento que esté relacionado con datos e información debe manejarse en archivos de texto (p. ej: nombres y descripción de los enemigos). Así, al iniciar el videojuego, este debe cargar la información en las estructuras de datos correspondientes.
4. Lo más importante a tener en cuenta en el desarrollo del proyecto es que se deben aplicar los conceptos vistos durante todo el curso. Por lo tanto, además de la correcta ejecución del programa, se calificarán las buenas prácticas en el uso de:
  - a. Comentarios para explicar el código.
  - b. Variables, constantes, constantes simbólicas.
  - c. Tipos de datos: enteros, flotantes, char, arreglos, etc.
  - d. Estructuras de control.
  - e. Funciones.
  - f. Estructuras de datos.
  - g. Estructuras dinámicas como Listas.
  - h. Separación de archivos en cabeceras y archivos de implementación.
  - i. Algoritmos de búsqueda y ordenamiento.
5. Se valorará la creatividad de los estudiantes al momento de definir nombres de personajes, estadísticas, nombres de enemigos, prompts en la consola, economía del juego, etc.

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

La nota del proyecto tiene dos componentes: una calificación (0.0 a 5.0) por el funcionamiento del programa y un multiplicador (0.0 a 1.0) basado en la sustentación realizada por cada grupo.