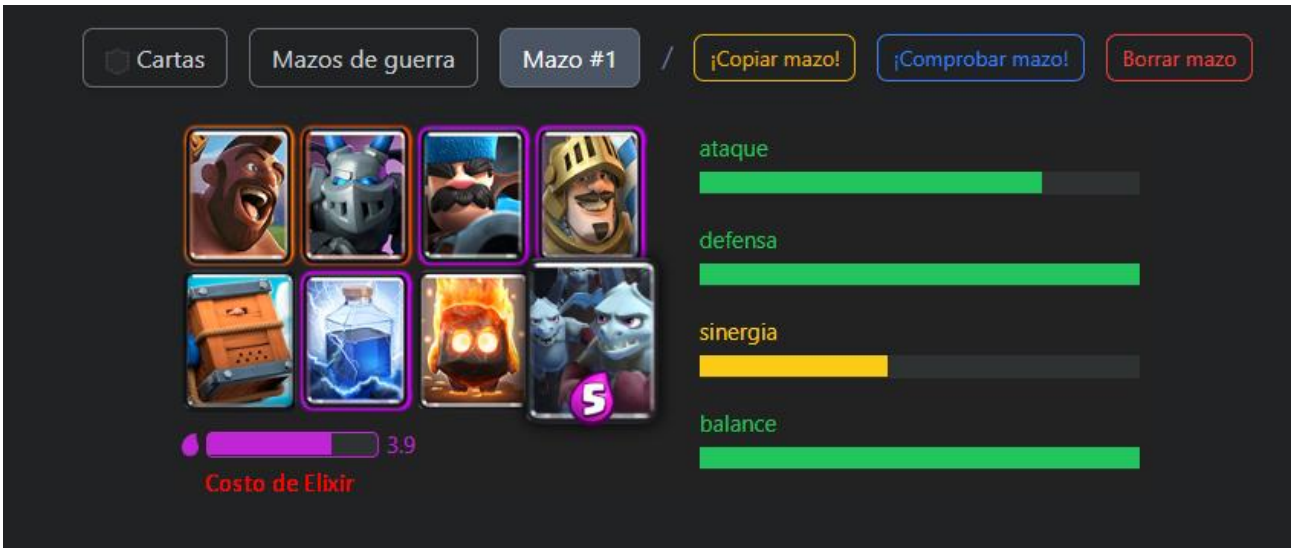
PROYECTO

Clash Royale League de SUPERCELL está por iniciar, y 2 jugadores necesitan preparase para las World Finals del 2022.

Clash Royale es un juego en base a cartas. Un mazo tiene 8 y el jugador solo puede usar 1 mazo a la vez (es decir, 8 cartas). Ninguna carta se puede repetir dentro del mismo mazo, pero 2 o más mazos pueden compartir cartas en común. Como regla general para las finales, los 4 mazos deben usar al menos 3 cartas únicas (que no se repitan en otros mazos) y las otras 5 quedan a su libre elección (pueden ser las mismas entre mazos o diferentes).

Ambos jugadores esperan recibir los 2 DECKS (mazos) que utilizarán, pero necesitan de su ayuda para seleccionar las cartas en ellos. Deberá construir un programa en C# el cual determine cual es el mejor DECK a utilizar en base a PUNTOS DE VIDA (la suma de la vida de todas las cartas que tengan vida) y DAÑO (la suma de daño que realizan todas las cartas), además de 2 propiedades de las siguientes:

- Ataque
- Defensa
- Sinergia
- Balance



Respecto a las cartas, su programa no evaluará las cartas individualmente, pero se sugiere que

elija algunos criterios para evaluar qué cartas incluirá en su mazo. Por ejemplo, algunas cartas son "áreas", es decir, que vuelan y solo pueden ser derrotadas por cartas que tenga ataque "aéreo". Podría entonces considerar una propiedad de "Preferencia de Ataque" y establecer si es Terrestre, Aéreo o Ambos. Queda a su libertad elegir esta opción u otras si considera pertinente.

Documento de Análisis y Diseño

Previo a seleccionar mazos y cartas, defina una estrategia. Pregúntese

- ¿Qué propiedades evaluaré en los mazos?
- ¿Qué propiedades son importantes de considerar sobre las cartas?
- ¿Acaso evaluaré todas las cartas o puedo descartar algunas?
- ¿Qué criterio utilizaré para descartar cartas si se requiere?

Definir una estrategia implica que deberá describir bajo qué criterios usted ha considerado ciertas cartas sobre otras para construir los mazos. Asuma que, al seleccionar algunas propiedades para dicha estrategia, las demás son menos relevantes, pero no desaparecen por completo.

Verificación de mazos (Decks)

Ingrese a https://www.deckshop.pro/es/card/list donde dispone de toda la información de todas las cardas, organizada por múltiples categorías.

Una vez que su estrategia está clara y tenga algunas cartas como opción, ingrese a https://www.deckshop.pro/es/deck-builder/clan-wars/war-decks para construir los dos mazos para cada jugador. Evalúe si las propiedades que usted selecciono (ataque, defensa, sinergia, o balance) se ajustan a su estrategia al construir el mazo. Ajuste de ser necesario.

Para su documento:

- Cree un documento de Excel y hoja de cálculo en Google.
- Agregue una Hoja llamada "Estrategia".
    - Describa las 2 propiedades que evaluó sobre los mazos para construirlos, y como estos se relacionan con los PUNTOS DE VIDA y el DAÑO.
    - Defina también como "decidirá" que mazo es mejor que otro en base a las 2 estrategias seleccionadas además de los PUNTOS DE VIDA y el DAÑO.
- Agregue una hoja por mazo y jugador. (Jugador 1 – Mazo 1, Jugador 1 – Mazo 2, Jugador 2 – Mazo 1, Jugador 2 – Mazo 2)
    - Liste las cartas a utilizar en cada mazo
        - Incluya
            - Nombre
            - Foto
            - Costo de Elixir
            - Daño (si aplica o 0 en caso contrario)
            - Puntos de Vida (si aplica o 0 en caso contrario)
    - Nombre el mazo de alguna forma creativa
    - Calcule los valores para las propiedades del mazo

        - DAÑO: suma de los daños individuales de cada carta
        - PUNTOS DE VIDA: suma de los puntos de vida individuales de cada carta en el mazo
        - Para las 2 propiedades que elija en su estrategia: alto (verde), medio (amarillo), bajo (rojo).
    - Describa la mejor situación de cada mazo según su estrategia. Por ejemplo, si las propiedades que usted eligió son ataque y sinergia, puede explicar que ha construido un mazo que permite una ataque rápido y muy eficiente contra mazos que no tengan tan buena defensa. Sea creativo, usted define los límites.

Programación C#

Una vez que disponga de toda la información para construir sus mazos, es momento de crear una aplicación que permita validar su estrategia. Cree un proyecto de Windows Forms en Visual Studio. Recuerde trabajar en C# .NET Framework.

Agregue los mazos a su código:

Deberá ingresar los 4 DECKS al sistema:

- Cree una clase Carta
    - La case carta tendrá las propiedades publicas
        - Nombre
        - Puntos de vida
        - Daño

- Crear una clase llamada Deck
    - La clase Deck tendrá las propiedades publicas
        - Cartas (se recomienda usar un array de tipo Carta, con 8 elementos)
        - Puntos de Vida
        - Daño
        - Ataque
        - Defensa
        - Sinergia
        - Balance
        - Nombre
    - Tambien tendrá los siguientes métodos:
        - GetDañoTotal() – Calcula la suma de daño individual de cada carta en el mazo.
        - GetPuntosDeVida() – Calcula la suma de los puntos de vida de cada carta del mazo

- Crear un constructor para la clase Deck que reciba el nombre del mazo a crear.

- Cree 4 instancias de la clase Deck y nómbrelas como indica en su documento de análisis y diseño.

- Cree la instancia de cada carta que vaya a utilizar en los mazos. Luego, construya los mazos asignando esas cartas a la propiedad Cartas de cada objeto Deck.

Vista de Jugadores

- Debe mostrar el Nombre y Equipo de cada jugador.
- No es necesario que el nombre y equipo pueda editarse dinámicamente (puede dejarlo quemado en su código usando variables).
- También mostrará un Nick, que se construye de la siguiente forma:
    - Iniciales de cada palabra del nombre del Equipo
    - Nombre del Jugador
    - Longitud total del nombre.
    - Por ejemplo
        - Equipo: Team Queso
        - Jugador: Joseph Soto

      El Nick seria: TQJosephSoto12
        - TQ por las iniciales del equipo.
        - El nombre sin espacios de por medio
        - 12 es la cantidad de caracteres en "TQJosephSoto". Puede usar la propiedad ".Length()" para obtener dicho número.

  Recuerde: aunque el nombre o equipo de los jugadores no cambie dinámicamente, el Nick si lo hará. Si durante su presentación se le solicita cambiar el nombre de un equipo o de un jugador en su código, el Nick se deberá ajustar automáticamente sin más cambios al programa.

Vista de Mazos Jugador X

- Debe mostrar las cartas del Mazo 1 y 2, de forma ordenada y claramente identificables, para cada jugador. Puede crear 2 vistas diferentes, una para el jugador 1 y otra para el jugador 2.
- Para las cartas, debe mostrar el Nombre de la carta únicamente.
- De cada mazo, deberá mostrar las propiedades siguientes:
    - DAÑO (calculado por su programa en GetDañoTotal())
    - PUNTOS DE VIDA (calculado por su programa en GetPuntosDeVida())
    - Propiedad 1 seleccionada (el valor lo tomara de su documento de análisis)
    - Propiedad 2 seleccionada (el valor lo tomara de su documento de análisis)
- Una sección de la vista debe mostrar
    - Qué mazo de los dos es mejor para el jugador X dada su estrategia. Utilice las condicionales IF, IF-ELSE para validar su estrategia contra los mazos y asi determinar el mejor mazo.

Recuerde: aunque los mazos estarán agregados en su código y no cambiaran dinámicamente, es importante que las propiedades calculadas si lo hagan. Si en su presentación se le solicita cambiar una carta o los valores de una carta, o de su mazo, los valores de DAÑO y PUNTOS DE VIDA se verán afectados.

Recuerde: su programa debe calcular el mejor mazo analizando los valores de sus propiedades

según su estrategia. Si se le solicita cambiar alguna propiedad en el código de su programa, el mismo debe ser capaz de mostrar el mazo ganador correcto según su estrategia para los nuevos valores de las propiedades.

Vista PvP

Su programa debe permitir comparar 1 mazo de cada jugador entre sí.

- Permita seleccionar 1 de los 2 mazos del jugador 1.
- Permita también seleccionar 1 de los 2 mazos del jugador 2.
- De cada mazo, deberá mostrar las propiedades siguientes:
    - DAÑO (calculado por su programa según las cartas del mazo)
    - PUNTOS DE VIDA (calculado por su programa según las cartas del mazo)
    - Propiedad 1 seleccionada
    - Propiedad 2 seleccionada
- Una sección de la vista debe mostrar
    - Qué mazo de los dos es mejor para el jugador X dada su estrategia. Utilice las condicionales IF, IF-ELSE para validar su estrategia contra los mazos y asi determinar el mejor mazo.

Extra (20%)

- **Mostrará las imágenes de cada carta en cada una de las vistas donde se muestren las cartas.**
- **El nombre de los jugadores y su equipo puede ser editado dinámicamente, y su Nick se ajusta en tiempo real al modificar alguno de esos dos datos.**

Rubrica

Evaluación de programación (40%)		
	Puntaje máximo	**Puntaje obtenido**
Código de proyecto	**10**	
Interfaz	**5**	
Funcionalidad	**10**	
Uso adecuado de clases y propiedades	**10**	
Repositorio	**5**	
Evaluación de documentación (50%)		

Presentación	**2**	
Calidad del informe	**2**	
Estructura y formato	**2**	
Ortografía y gramática	**2**	
Redacción y coherencia	**2**	
Análisis de las cartas, mazos y sus propiedades	**15**	
Definición de la estrategia y sus validaciones.	**15**	
Defensa	**10**	
Evaluación de exposición		
Puntualidad	**2.5**	
Dominio del Proyecto	**2.5**	
Dominio del Código Fuente	**2.5**	
Manejo de cambios	**2.5**	
TOTAL	**100**	