

Metodología de Juego

- El Jugador 1 ingresa su nombre y elige su héroe
- El Jugador 2 ingresa su nombre y elige su héroe
- Se da inicio al juego con el botón ¡COMENZAR! Y a continuación se muestra el tablero.
- Se muestran las manos de ambos jugadores. Para seleccionar una carta y ponerla en el tablero, es decir, para ponerla en juego, se debe hacer click sobre ella. Luego aparecerá en la arena.
- El turno del jugador termina cuando este presiona el botón *Terminar Turno*.
- A medida que avanza el juego y haya más cartas en el tablero, se podría eventualmente atacar las cartas o el héroe del otro jugador. Para esto hay que:
 - o **Seleccionar la carta que desea que ataque con 1 click**
 - o **Seleccionar la carta que desea atacar o bien al héroe contrario con el botón *Atacar Héroe*, también con un click**
 - o **Para llevar a cabo el ataque, hay que presionar el botón *Atacar***
- El juego finaliza cuando la vida de uno de los 2 héroes llega a 0
- Agregamos también la función de comunicarse. Al igual que en el juego original, un jugador puede comunicarse con otro independiente de si es o no su turno.
- El botón *Habilidad* ejerce la función de hacer la habilidad propia del héroe que se eligió en un comienzo. Utiliza 2 de maná y solo se puede hacer uso de ésta 1 vez por turno al igual que en el juego original.
- El jugador puede rendirse con el botón *Rendirse*, con lo que su vida llegará a 0 instantáneamente y se cerrará el juego.

Supuestos

- Para determinar que jugador jugará primero haremos un random.
- El jugador que va primero saca 3 cartas del mazo
- El jugador que va segundo saca una carta extra llamada *La Moneda*.
- Cada jugador saca una carta al principio de cada turno
- A cada jugador se le agrega un cristal de mana por turno
- A cada jugador se le rellena todo el mana al inicio de cada turno
- El máximo cristal de mana que un jugador puede tener es 10
- El máximo de cartas que un jugador puede tener en la mano es 10
- Cada jugador puede tener solo una weapon a la vez
- Cada jugador puede tener un máximo de 7 minions en el tablero
- Cada jugador puede tener máximo 2 cartas iguales en mazo
- Cada jugador puede tener solo una carta legendaria de cada tipo en el mazo
- Un héroe solo puede atacar si tiene una weapon equipada
- Si el jugador tiene una weapon en el tablero y ocupa otra weapon esta reemplaza la anterior
- Si al jugador se le acaban las cartas del mazo cada vez que este trate de sacar una carta incluyendo la carta inicial que se saca al principio de cada turno a este se le restará uno de vida y se incrementará en uno por cada carta sacada
- Cada mazo debe tener exactamente 30 cartas
- Cuando hay 10 cartas en la mano, se **quema** la última carta que se saca
- Entraremos en detalle sobre el tiempo de cada turno cuando lo usemos más adelante en el Main (80 segundos).

- Con el nombre de una carta podemos identificar todos los atributos de la carta (habilidad, mana, etc).
- Se asume que el usuario inserta los **números** de la opción dada en consola cuando se le ofrece (Solo escribe *String* para ingresar su nombre).
-

Atributos/Métodos:

1. Creamos el método **Refresh**, que actualiza las manos de los jugadores
2. Agregamos un atributo booleano **Actividad** en la clase Carta que cuando retorna *True* es que está en el tablero, y *False* cuando está en la mano, en el mazo o muerta. Asimismo, agregamos el atributo booleano **Visibilidad**, que indica cuando podemos ver literalmente la carta en el juego; es decir que retorna *True* cuando la carta está en la mano o en juego, y *False* cuando está muerta o en el mazo.
3. Agregamos el método **Devolver Carta** en la interfaz Acciones de Juego, que recibe como parámetro el nombre de la carta y que cambia la *Actividad* y *Visibilidad* de ésta a *False* si es que se devuelve al mazo.
4. **Habilidad** en la clase Héroe es la habilidad especial que tiene cada héroe (ésta es única en cada uno y se diferenciarán por el nombre del héroe).
5. Hicimos también el método **Describir** que lo hace es dar las características de la carta en juego (Equivalente a poner el mouse sobre la carta en el juego). Este método lo heredan todas las cartas.
6. **Habilidad** en la clase Carta es el efecto que realiza la carta en el juego (*Provocación, inflige 2 de daño, etc*).
7. Creamos el método **Cambiar Actividad** que cambia el estado de la Carta para ver si está en el mazo, en la mano, en juego o muerta. Asimismo, creamos el método **Cambiar Visibilidad**.
8. En la clase Mazo agregamos el método **Agregar Carta** tiene como funcionalidad agregar las cartas ya creadas al mazo para poder jugar, por lo mismo, usa como parámetro el nombre de la carta que se está agregando. Asimismo, el método **Barajar Mazo** se hace antes de la partida y genera un random en el orden de las cartas de éste.
9. **Fatiga** es un método que posee un contador que cada vez que el jugador “saque” una carta del mazo y este está vacío, se le resta la cantidad de vidas que indique el contador. Este irá en aumento de 1 en 1 cada vez que esto suceda.
10. **Jugar Carta** es la acción de jugar la carta en sí. Para esto la carta seguirá con su **Visibilidad** en *True* y su **Actividad** pasará de *False* a *True* ya que ya no estará en la mano si no que en el tablero.
11. El método **Dar Carta** en la clase Mazo es el hecho de pasar una carta del mazo a la mano (Equivalente a robar una carta).
12. **Quemar Mano** es el método que cuando la lista de la clase Mano posee 10 cartas, la próxima carta que se robe, se quema; por lo que su *Actividad* y *Visibilidad* quedan en *False*.
13. **Terminar Turno** suma 1 a un contador que tendremos en el Main, el cual determina que jugador está jugando: Número impares juega el jugador 1 y números pares juega el segundo.
14. El método **Comunicarse** es el equivalente a poder “enviar” ciertos mensajes predeterminados en el juego. Esto lo haremos más adelante con un *Console.WriteLine*.

15. El método **Rendirse** es la opción voluntaria del jugador de perder. El método consiste en llevar el contador de vidas del Héroe a cero, por lo tanto, recibe como parámetro el atributo *Vida* de éste.
16. **Atacar** es un método propio de las clases **Weapon** y **Minion**, lo que hace es restar el ataque del atacante menos la vida del atacado, y restar el ataque del atacado con la vida del atacante; cambiando las variables de vida de ambas cartas. Si la vida de cualquiera de las cartas afectadas es menor o igual a 0, su **Actividad** y **Visibilidad** cambian a *False*.