

Mi Aldea



Realizado por:

Patricia Cantero García

Francisco Javier Díaz Puertas

Ángel Molina Santiago

Índice:

1. Explicación de nuestro proyecto
2. Patrones de diseño
3. Realización de la base de datos
4. Como hemos organizado nuestro proyecto

Explicación de nuestro proyecto

Nuestro proyecto es una aplicación web que consiste en simular batallas con acumulación de puntos con el objetivo de mejorar nuestro personaje tanto su nivel como su casa.

En nuestra aplicación una vez que el usuario se registra puede escoger un personaje el cual puede elegir entre ser arquero, guerrero o mago. Parte de una casa de paja y con un nivel de armadura de nivel 1.

Necesita llegar a 10 puntos para poder mejorar la casa o llegar a 15 puntos para poder mejorar el personaje escogido. Los puntos se consiguen batallando donde si ganas recibes 3 puntos, si empatas recibes 1 y si pierdes restas un 1 punto.

Para realizar esto en el controlador tenemos los ficheros acción y conexión.

En acción se produce la ejecución de los métodos creados en otros ficheros que el usuario necesite.

En conexión tenemos la clase Db que se encarga de realizar la conexión a la base de datos. En esta clase hemos aplicado el patrón de diseño singleton, que hace que se pueda instanciar una única vez la conexión.

En modelo tenemos los ficheros: arquero, batalla, guerrero, mago, personajes, jugadores y objetos.

En personajes tenemos la clase personaje con las variables protegidas nombre, nick, contraseña, tipo, casa, arma y la variable privada puntos. El constructor de esta clase recibe tres variables las cuales son nombre, nick y contraseña e inicia puntos = 0, casa = paja y arma = nivel 1. Luego tenemos los métodos get que lo que hacen es devolver la variable. Además, tenemos el método actualizar personaje que recibe el nick, también tenemos actualizarPersonaje, luchar y actualizarPuntos .

También tenemos las clases Arquero, Guerrero y Mago que son subclases de Personaje. En su constructor llaman al constructor del padre y asigna a la variable tipo el personaje que es.

En la clase batalla la lucha se hace un aleatorio entre 1 y 3.

Luego tenemos la clase Jugadores que en su constructor va a recibir el objeto de la clase Db, tiene los métodos consultar que se utiliza a la hora de hacer un registro ya que comprueba si existe o no el usuario, el método iniciar cuando un jugador se logea para comprobar si el nick y la contraseña son correctas.

Luego tenemos devolverDatos que lo necesitamos ya que cuando estamos en la pantalla de usuario.php solo tenemos el valor de la sesión pero necesitamos el personaje para poder usar sus métodos y esto lo que hace es devolver. Ahora tenemos insertar que se usa para crear el personaje a la hora del registro. El método actualizar

se usa para mejorar arma, mejorar casa o actualizar puntos. El método borrar que es para cuando queremos borrar un personaje y por último consultar mejora que se utiliza cuando estamos mejorando arma o casa para ver el valor que tenía antes.

La clase Objeto lo que hace es que utiliza un patrón de diseño abstract Factory la cual parte de una interfaz llamada objeto que tiene el método mejorarObjeto y que recibe un objeto de la clase Personaje.

La clase Casa y Arma implementan esta interfaz por lo que en su método mejorarObjeto mejoran el objeto (casa si llegan a 10 puntos y arma si llegan a 15 puntos) y restan los puntos que han usado al mejorar el objeto.

En vista tenemos index.php, usuario.php, crearpersonaje.php y combate.php.

En el index que te da la opción de elegir entre logearte o registrar personaje.

En crearpersonaje partiendo de la elección de un tipo de personaje realizamos el formulario y si no existe lo crea (el nick tiene que ser único)

En usuario partiendo de la sesión que ha iniciado recuperamos el objeto personaje para poder usar sus métodos.

En combate lo único que nos hace es mostrarnos el combate entre los personajes.

Patrones de diseño

Singleton: Este patrón simplemente restringe la instanciación de una clase a un objeto.

Active Record pattern: Este patrón sirve para acceder a datos de una base de datos. Una tabla o view de una base de datos se encapsula en una clase, de forma que el objeto instanciado puede manipular la tabla en función de los métodos que se hayan creado para manipularla.

El patrón Abstracta Factory nos ofrece una interfaz para crear objetos a partir de cada clase de la familia de productos. Mientras tu código cree objetos a través de esta interfaz, no tendrás que preocuparte por crear la variante errónea de un producto que no combine con los productos que ya ha creado tu aplicación.

Realización de la base de datos

La base de datos mialdea tiene una tabla que se llama jugadores que contiene las columnas Id, nombre, nick, contr, personaje, casa, arma, puntos y control.

Donde id un entero autoincrementable y es la clave primaria, nombre, nick y contraseña son varchar y nick además es unique, luego personaje, casa y arma son de

tipo enum que solo pueden recibir los datos preestablecidos, puntos es un entero y control es una columna usada para verificar si existe el usuario.

A esta tabla solo puede acceder el usuario jap desde cualquier sitio con todos los privilegios.

Como hemos organizado nuestro proyecto

Del uso de la base de datos, creación de la base de datos y conexión que son los ficheros conexión.php, jugadores.php, basededatos.sql se ha encargado Patricia.

De la creación de la clase objeto y la clase personaje que son los ficheros objetos.php y personajes.php se ha encargado Ángel.

De la creación de las subclases de personaje y la clase batalla que son los ficheros arquero.php, guerrero.php, mago.php y batalla.php y el fichero acción.php, se ha encargado Javi.

Las vistas la hemos realizado entre los tres integrantes del grupo.

Nos hemos conectado todas las tardes por Discord y hemos estado subiendo las modificaciones por Github y hemos usado el servidor Plesk proporcionado para subir el proyecto.