Ultimo proyecto.

Temporizador:

El juego debe tener un temporizador, inicia cuando inicia la partida, pausa si se sale de la partida (se deber quedar almacenado el avance) y se termina el temporizador algunos de los jugadores ganan la partida.

* Un temporizador.

Especificaciones del temporizador:

* 1. Inicia cuando inicia la partida
  2. Para el tiempo cuando: el usuario gana, cuando el usuario sale de la partida, para cuando pierde el juego el usuario.

Almacenaje de usuario (jugador):

Se le solicita al usuario que inicie sesión o que crea se registre al iniciar una partida, para almacenar su estadística.

* Atributos va tener un usuario: id, nombre, contraseña.
* Para ingresar un usuario debe iniciar sesión con su nombre y su contraseña.

Por parte de código: planteo almacenarlo como un objeto, usando claramente POO, podría ser una matriz el almacenamiento, al momento que cada partida, automáticamente se modifican los puntos o los jugos jugados.

* Otro punto a tomar en cuenta es, el orden de los usuarios, deben estar en orden, del jugador que tiene más punteo decreciente.

Registros por jugador:

Los registros esta ligados claramente al jugador, deben ser almacenados al jugador que correspondan.

Los registros a tomar en cuenta son:

* Total, de partidas jugadas por jugador.
* Total, de partidas ganadas por jugador.
* Total, de partidas perdidas por jugador.
* Total, de movimientos realizados por jugador a lo largo de las partidas.
* Promedio de movimientos realizados por jugador a lo largo de las partidas.
* Victoria más corta por jugador (partida con la que se requirió la menor cantidad de movimientos para ganarla). -> Desplegar en una columna el total de movimientos y en otra el tiempo en minutos y segundos que duró la partida.

Ventana de inicio de sesión:

Al momento de abrir, debe contener: 2 cuadros de texto y un botón.

Acción: El primer cuadro de texto se deber ingresar el nombre, y en el segundo se debe ingresar la contraseña, el botón validara datos ingresados.

Se tiene que analizar si los cuadros de textos que no estén vacíos

Por parte de código: al momento de darle clic al botón, debe tomar el dato ingresado en el primer cuadro de texto, buscar el dato ingresado en la base de datos, si este coincide con algún dato, entonces procede a tomar el segundo dato en el segundo cuadro de texto compara con la contraseña registrada si estos dos coinciden, entonces un usuario habrá iniciado sesión. De lo contrario si el primer no coincide con los datos almacenados, simplemente volverá a pedir otra vez los datos.

Si un usuario logra iniciar sesión, sus datos quedaran listos para cualquier modificación, ya se dé puntos que obtiene en el juego o que decide editar sus datos.

El usuario tiene de cerrar sesión para que sus dados dejen de ser modificados.

Registro de un nuevo usuario:

Al momento de que un usuario elija dicha opción, deber abrir la ventana correspondiente a dicha a acción, la venta debe contener: 2 cuadros de texto, y un botón.

Se tiene que analizar si los cuadros de textos que no estén vacíos

Acción: al instante de darle clic al botón, este tomar el dato ingresado en el primer cuadro de texto para buscar si hay alguna coincidencia con los datos almacenado, si no hay, entonces se procede a tomar el dato que se ingresó en el segundo cuadro de texto para proceder su almacenamiento. De lo contrario, si se encuentra algún nombre que coincide con los datos almacenados, este se procederá a solicitar de cambiar de nombre.

Juego de damas:

Que debe tener el JFrame:

* Un temporizador.

(Reutilización del temporizador)

* Contador de turno

Especificaciones:

* 1. Cada que el usuario realizar un movida de un peón o una Dama, este va contando.
* La tabla para jugar:

Especificaciones:

* 1. Tiene casillas, un total de 64, (8\*8)

Especificaciones atributos:

* + 1. Activo (contine una ficha)
    2. Desactivado (no puede contener ficha)
    3. Tipo de ficha contenida (roja o negra)
  1. Tiene color.
  2. Contiene fichas.
  3. Contiene eventos.
* Un menú:

Especificaciones

Tabla para jugar: planteo usar 32 botones para las casillas, donde se pueden mover los peones en casillas negras;

Estos botones tienes la habilidad de contener el peón y también de analizar las casillas a tacar o a mover.

Diagrama UML

