

TAREA N° 3

PROGRAMACIÓN ORIENTADA A OBJETOS

DOCENTES:

Jazna Meza Hidalgo / Alberto Zenteno Acevedo

OBJETIVO

Aplicar lo aprendido en Java. El estudiante deberá aplicar los contenidos revisados en clases relacionados con los conceptos de la orientación a objetos y la implementación de clases en Java. Incluye los conceptos de herencia, manejo de archivos, sobrecarga y sobrescritura de métodos y el uso de interfaces gráficas en Java considerando controles gráficos y programación de eventos. Se incluye la investigación de temas relacionados con eventos del mouse y manejo de hilos de ejecución.

ENUNCIADO

Se requiere implementar un juego de memoria donde debe identificar parejas de imágenes midiendo el tiempo que demora el jugador en encontrar todas las parejas. Existen 2 opciones de juego: nivel básico y nivel medio; en el nivel básico se tiene un tablero de 16 imágenes, 8 pares de imágenes organizadas en 4 filas y 4 columnas y el nivel medio consiste en un tablero de 24 imágenes, 12 pares de imágenes organizadas en 6 filas y 4 columnas.

Para cada jugador se debe manejar su nombre y el conjunto de los tiempos que ha ido registrando las veces que ha jugado.

Su programa debe cargar datos de jugadores y permitir seleccionar algún jugador de la lista disponible o bien agregar un nuevo jugador. Los nombres de los jugadores deben ser únicos dentro de la lista de jugadores.

El tablero se debe completar con imágenes de 3 temas distintos. La ubicación de cada imagen (y su pareja) debe ser asignada de manera aleatoria.

Una vez que se ha armado el tablero con las imágenes ubicadas dentro del tablero de acuerdo al nivel y tema seleccionado se debe ofrecer al usuario una ventana con las imágenes disponibles para ese tema. Esta ventana solo será de referencia para que el usuario sepa cuáles son las imágenes con las cuales jugará, el orden de las imágenes va a cambiar al momento de iniciar el juego.

Una vez que se hayan encontrado todas las parejas se debe mostrar el tiempo en segundos que ha demorado en encontrar las parejas y se debe guardar en memoria para, eventualmente, ser traspasados al archivo de texto (revisar sección **SU DESAFÍO**).

Además, debe existir una opción que indica la estadística del jugador; esto es indicar el mejor tiempo, el promedio de los tiempos y el total de juegos que lleva registrados incluyendo los que ha realizado en la última sesión.

SU DESAFÍO

Como una parte inicial del proyecto ya ha sido diseñado el diagrama de clases y como se sabe que usted está aprendiendo Java y tiene el desafío de mejorar su conocimiento es que ha sido seleccionado para implementar el diagrama y construir una aplicación gráfica en Java que permita cumplir con los siguientes requerimientos:

1. Cargar desde un archivo de texto llamado Memorice.txt un conjunto de jugadores junto a sus tiempos. El formato del archivo debe ser

- líneas formadas por una cadena de caracteres (nombre del jugador) seguido de un punto y coma (;), luego vendrán números enteros que representan los tiempos registrados en los juegos, cada uno de estos valores están separados por un punto y coma (;).
2. Manejar una lista de jugadores considerando los datos contenidos en el archivo de forma de poder iniciar el juego con algún usuario registrado previamente. Esta lista es la que deberá ser usada para poder iniciar una sesión en el juego.
 3. Permitir agregar un nuevo usuario indicando su nombre y verificando que ese nombre no se encuentre registrado previamente.
 4. Permitir a un jugador iniciar un juego indicando el tema de las imágenes que desea cargar y el nivel en el que desea jugar.
 5. Un jugador puede cerrar la ventana donde aparecen las imágenes, en ese momento se debe pedir la confirmación de su deseo de salir del juego indicando la advertencia que sus tiempos no serán guardados.
 6. Cuando un jugador decide salir de la ventana de juego se debe volver a la ventana donde se pueda seleccionar un usuario, un nivel, un tema y volver a jugar.
 7. El jugador puede tener la posibilidad de guardar todos los juegos que realizó en la última sesión iniciada; esto significa que su nombre y sus tiempos se agregan al archivo de texto o se actualizan sus datos en el caso de que ya se encuentra registrado.
 8. Un jugador puede jugar las veces quiera y se debe registrar el tiempo de cada juego; esto por si decide guardar sus juegos al cerrar su sesión.
 9. Usted debe disponer de las imágenes asociadas a cada tema, en el caso de que no existan imágenes de un tema se debe advertir al usuario y evitar que comience a jugar.

REQUERIMIENTOS DE LA APLICACIÓN GRÁFICA

1. Manejar una interfaz gráfica de usuario que permita cumplir con todos los requerimientos que se establecen.
2. Usted debe decidir el diseño de su interfaz gráfica incorporando la mayor cantidad de componentes gráficos que se puedan incluir, pensando siempre en la facilidad de uso de la aplicación de parte de los usuarios.
3. En la ventana donde se está jugando DEBE aparecer en el título el nombre del jugador que ha iniciado la sesión.

DIAGRAMA DE CLASES

Ver Anexo - DiagramaClases_T3.png

El diagrama de clases incluye las clases asociadas al negocio. La clase **Util** en este caso incluye algunas funcionalidades generales y a esa clase usted le puede agregar los métodos que estime convenientes.

INVESTIGACIÓN

Usted deberá investigar respecto de varios temas para poder desarrollar su tarea. Entre los conceptos que debe averiguar se encuentran:

MouseListener

<https://docs.oracle.com/javase/7/docs/api/java/awt/event/MouseListener.html>

SwingWorker

<https://docs.oracle.com/javase/7/docs/api/javax/swing/SwingWorker.html>
http://chuwiki.chuidiang.org/index.php?title=Ejemplo_sencillo_con_SwingWorker

RESTRICCIONES

- El programa **DEBE** considerar la implementación de código de acuerdo a lo indicado en el diagrama de clases.
- La tarea **DEBE** compilar al momento de ser entregada.
- La documentación debe responder al esquema revisado en clases.

CONDICIONES

- **Fecha de entrega.** Viernes 11 de enero hasta las 22:00
- **Forma de entrega**
 - Subir los archivos a la plataforma <http://moodleubb.ubiobio.cl>. El período para subir los archivos estará disponible el día de inicio de entrega de la tarea. SE DEBE INDICAR CLARAMENTE los integrantes del grupo y la sección.
- **Formato de entrega**
 - Proyecto NetBeans.
 - Archivo readme.txt indicando los integrantes del equipo
- **Equipo de desarrollo**
 - Dos (2) personas

FORMA DE EVALUACIÓN

- Orden y claridad en el código
- Documentación
- Compilación y ejecución correcto: Funcionamiento correcto.
- Interrogación acerca del desarrollo.

FORMA DE IDENTIFICACIÓN DE INTEGRANTES DEL EQUIPO

Se debe incluir, en la sección de **@author** de la documentación, el nombre de los integrantes e incluir un archivo README.txt que indique el nombre completo de los integrantes y sección a la cual pertenecen.

OBSERVACIONES IMPORTANTES

- Se acepta el hecho de que los grupos de trabajo puedan compartir el conocimiento, pero lo que **NO SE ACEPTA** es el hecho de que los códigos se encuentren EXACTAMENTE iguales; dicha situación será evaluada con la nota mínima SIN DERECHO A APELACIÓN.
- Las dudas que NO SE RESPONDEN son el *¿cómo se hace esto?* Las dudas que SI SE RESPONDEN son *¿traté de hacer esto, pero no me ha resultado?*
- Las tareas que sean entregadas fuera del plazo establecido serán CASTIGADAS con puntos de descuento. Estos puntos de descuento vienen dados por la siguiente forma:

1,5 puntos por día de atraso

Con respecto a esto, se amerita una importante observación: por ejemplo, si la tarea se entrega entre los siguientes rangos de tiempo, se aplican los descuentos que se muestran a continuación:

Fecha de entrega	Descuento aplicado
11 de enero a las 22:01 horas hasta el 12 de enero a las 21:59 horas	Descuento = 1.5 puntos
12 de enero a las 22:00 horas hasta el 13 de enero a las 21:59 horas	Descuento = 3 puntos
13 de enero a las 22:00 horas hasta el 14 de enero a las 21:59 horas	Descuento = 4.5 puntos
14 de enero días después de las 22:00	Calificación = 1