

Diego Luengo Gil

Videojuego

Fundamentos de la Programación

1º DAM

18/11/2024

Hoja Control de Versiones	3
Descripción del Videojuego	4
Análisis del proyecto	5
Estructura del proyecto	15
Diagrama de clases.....	16
Clases versión 1.7	17
Clases versión 1.8	17
Diagrama de flujo	18
Herramientas Utilizadas	18
Estado de implementación	19
Fuentes:	20

Hoja Control de Versiones

Versión	Fecha	Autor	Cambios realizados
1.0	Noviembre 2024	DLG	Creación del proyecto inicial
1.1	Enero 2025	DLG	Incorporación de la clase tablero
1.2	Enero 2025	DLG	Incorporación de la clase Configuración
1.3	Enero 2025	DLG	Incorporación java docs
1.4	Febrero 2025	DLG	Incorporación de herencia y polimorfismo
1.6	Marzo 2025	DLG	Eliminación herencia y comentarios actualizados
1.7	Marzo 2025	DLG	Agregación de clase Partidas y Perfiles y Modularización de la clase principal
1.8	Abril 2025	DLG	Agregación de bases de datos y sistemas de archivos

Descripción del Videojuego

Tema

El videojuego Extreme Memory consiste en un tablero con parejas de cartas ocultas que hay que revelar y emparejar

Audiencia/Estilo de juego

El juego está dirigido a un amplio rango de usuarios:

- **Audiencia:** Personas de todas las edades interesadas en juegos casuales.
- **Estilo de juego:** Casual con limite de intentos.

Reglas del juego

Las reglas del juego son sencillas y fáciles de entender, lo que lo hace accesible para cualquier usuario:

1. El tablero está compuesto por una cuadrícula de tamaño personalizable en un futuro.
2. El jugador elige dos cartas y si coincide el diseño se quedan visibles.
3. No se permite desbloquear casillas ya visibles
4. El jugador gana cuando limpia todo el tablero.

Mecánicas/Jugabilidad

El diseño del juego está orientado a proporcionar una experiencia fluida y visualmente atractiva:

1. Cada selección se comprueba que este disponible.
2. Se verifica automáticamente si el jugador a ganado después de 8 intentos.

3. La interfaz utiliza colores para distinguir entre las cartas acertadas falladas y seleccionadas.

Análisis del proyecto

Especificación de requisitos

Requisitos funcionales

A continuación, se especifican los requisitos funcionales del proyecto listado los casos de uso identificados.

Caso de Uso: Creación del Tablero

Campo	Descripción
ID	CU01
Título	Crear tablero inicial
Descripción	El sistema inicializa el tablero como una cuadrícula vacía de tamaño personalizable

Precondicion es	Ninguna.
Entrada	Tamaño.
Salida	Ninguna.
Boceto	Tablero de 3x3 representado como un array bidimensional.
Estado	Implementado.

Caso de Uso: Seleccion de cordenadas

Campo	Descripción
ID	CU02
Título	Solicitar unas cordenadas al jugador
Descripción	El jugador selecciona unas cordenadas para revelar.
Precondicion es	Las cordenadas elegidas no deben estar reveladas.
Entrada	Coordenadas de fila y columna (1-4).
Salida	Actualización del tablero con las cartas seleccionadas.
Boceto	Tablero de 4x4 con las cartas reveladas.
Estado	Implementado.

Caso de Uso: Mostrar Tablero

Campo	Descripción
ID	CU03
Título	Mostrar Tablero
Descripción	Se imprime el tablero por pantalla.
Precondicion es	El tablero se debe haber creado.
Entrada	Ninguna.
Salida	El tablero impreso.

Boceto	Líneas horizontales, verticales o diagonales completadas.
Estado	Implementado.

Caso de Uso: Verificación de victoria

Campo	Descripción
ID	CU04
Título	Verificar victoria
Descripción	El programa verifica si todas las cartas han sido reveladas.
Precondicion es	El tablero debe estar completamente lleno.
Entrada	Tablero actual.
Salida	Confirmación de victoria o continuación del juego.
Boceto	Tablero con todas las casillas ocupadas.
Estado	Implementado.

Caso de Uso: Generacion de tablero oculto

Campo	Descripción
ID	CU05
Título	Generar tablero oculto
Descripción	El programa asigna cartas con “?” A todas las posiciones del array.
Precondiciones	El tablero debe estar iniciado.
Entrada	Tablero actual.
Salida	Ninguna.
Boceto	Tablero con todas las cartas de?
Estado	Implementado.

Caso de Uso: Generación de tablero resultado

Campo	Descripción
ID	CU06
Título	Generar tablero resultado
Descripción	El programa asigna cartas aleatorias a las posiciones del array.
Precondiciones	El tablero debe estar iniciado.
Entrada	Tablero actual.
Salida	Ninguna.
Boceto	Tablero con cartas asignadas al azar.
Estado	Implementado.

Caso de Uso: Creación de Perfil

Campo	Descripción
ID	CU07
Título	Creación de Perfil
Descripción	El programa crea un perfil y lo introduce en el ArrayList perfiles
Precondiciones	
Entrada	Nombre del perfil
Salida	Confirmación
Boceto	Perfil con un arrayList de partidas
Estado	Implementado.

Caso de Uso: Eliminación de Perfil

Campo	Descripción
ID	CU08
Título	Eliminar Perfil
Descripción	El programa borra el perfil de la colección

Precondiciones	Un perfil creado
Entrada	Nombre del perfil
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Creación de Partidas

Campo	Descripción
ID	CU09
Título	Crear Partida
Descripción	El programa crea una partida y la añade al arrayList del perfil activo
Precondiciones	Perfil activado
Entrada	Partida
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Activar un perfil

Campo	Descripción
ID	CU10
Título	Activar un perfil
Descripción	El programa activa un perfil en el que se guardaran las partidas
Precondiciones	El perfil debe estar creado
Entrada	Nombre del usuario
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Guardar Perfil en base de datos

Campo	Descripción
ID	CU11
Título	Guardar un Perfil
Descripción	El programa guarda un perfil en la base de datos
Precondiciones	El perfil debe estar creado
Entrada	Nombre del usuario
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Borrar Perfil en base de datos

Campo	Descripción
ID	CU12
Título	Activar un perfil
Descripción	El programa borra un perfil en la base de datos
Precondiciones	El perfil debe estar creado
Entrada	Nombre del usuario
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Cargar un Perfil de la base de datos

Campo	Descripción
ID	CU13
Título	Cargar un perfil
Descripción	El programa carga un perfil de la base de datos
Precondiciones	El perfil debe estar creado
Entrada	Nombre del usuario

Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Guardar Partida en base de datos

Campo	Descripción
ID	CU14
Título	Guardar Partida
Descripción	El programa guarda una partida en la base de datos
Precondiciones	La partida debe estar iniciada
Entrada	Ninguna
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Borrar Partida en base de datos

Campo	Descripción
ID	CU15
Título	Borrar Partida
Descripción	El programa borra una partida en la base de datos
Precondiciones	La partida debe estar guardada
Entrada	Id de la partida
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Cargar Partida de la base de datos

Campo	Descripción
ID	CU16
Título	Cargar Partida

Descripción	El programa carga una partida de la base de datos
Precondiciones	La partida debe estar guardada
Entrada	Id de la partida
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Guardar Perfil en ficheros

Campo	Descripción
ID	CU17
Título	Guardar un Perfil
Descripción	El programa guarda un perfil en un fichero
Precondiciones	El perfil debe estar creado
Entrada	Nombre del usuario
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Borrar Perfil en ficheros

Campo	Descripción
ID	CU18
Título	Borrar un perfil
Descripción	El programa borra un perfil en ficheros
Precondiciones	El perfil debe estar creado
Entrada	Nombre del usuario
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Cargar un Perfil de los ficheros

Campo	Descripción
ID	CU19
Título	Cargar un perfil
Descripción	El programa carga un perfil de los ficheros
Precondiciones	El perfil debe estar creado
Entrada	Nombre del usuario
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Guardar Partida en ficheros

Campo	Descripción
ID	CU20
Título	Guardar Partida
Descripción	El programa guarda una partida en un fichero
Precondiciones	La partida debe estar iniciada
Entrada	Ninguna
Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Borrar Partida en los ficheros

Campo	Descripción
ID	CU21
Título	Borrar Partida
Descripción	El programa borra una partida de los ficheros
Precondiciones	La partida debe estar guardada
Entrada	Id de la partida

Salida	Ninguna.
Boceto	
Estado	Implementado.

Caso de Uso: Cargar Partida de los ficheros

Campo	Descripción
ID	CU22
Título	Cargar Partida
Descripción	El programa carga una partida de los ficheros
Precondiciones	La partida debe estar guardada
Entrada	Id de la partida
Salida	Ninguna.
Boceto	
Estado	Implementado.

Requisitos no funcionales

1. Interfaz de Usuario (UX):

- o Mensajes claros y uso de colores para mejorar la experiencia del usuario.
- o **Estado:** Implementado.

2. Portabilidad:

- o Compatible con sistemas Windows, macOS y Linux.
- o **Estado:** No comprobado.

3. Escalabilidad:

- o Estructura preparada para ampliar el tamaño del tablero.
- o **Estado:** No implementado.

4. **Fiabilidad:**

- o Manejo correcto de entradas inválidas.
- o **Estado:** Implementado.

Estructura del proyecto

A continuación, se describe la organización del proyecto del videojuego:



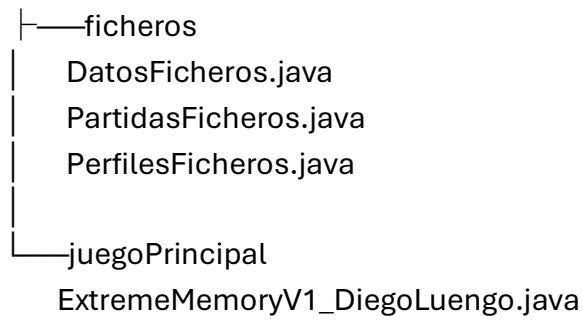


Diagrama de clases

INCLUIDO EN LA CARPETA DEL JUEGO Y EN LA ENTREGA DEL AULA VIRTUAL

TABLERO.JAVA
generarTablero
generarTableroOculto
mostrarTablero
PARTIDA.JAVA
leerPartida
leerPartidaByte
guardarPartida
guardarPartidaByte
GESTIONDEDATOS.JAVA
cargarJuego

Clase Configuración

Atributo maxIntentos: Este atributo afecta al máximo de intentos que tienes para revelar las cartas

Atributo dificultad: Este atributo afecta a si se muestra la primera carta revelada o no

Atributo dimensiones: Este atributo afecta al tamaño del tablero, aunque de momento solo soporta 4

Clases versión 1.7

Clase Perfil: Representa al usuario que está jugando, solo se guarda el nombre y una colección de las Partidas que tiene

Clase Partida: Representa a las partidas del usuario, contiene la fecha de la partida, la hora de inicio y la de final, además de datos de la partida como la situación del tablero y el número de fallos y el de aciertos

Clase colores: Contiene los colores usados en el juego y los mensajes de victoria y derrota

Clases versión 1.8

ConexionBD: Contiene todo lo necesario para la conexión con la base de datos

DatosDB: Contiene funciones para gestionar los datos del juego con la base de datos

PartidaDB: Contiene funciones para gestionar las partidas con la base de datos

PerfilesDB: Contiene funciones para gestionar los perfiles con la base de datos

DatosFicheros: Contiene funciones para gestionar los datos generales del juego con los ficheros

PartidaFicheros: Contiene funciones para gestionar las partidas con los ficheros

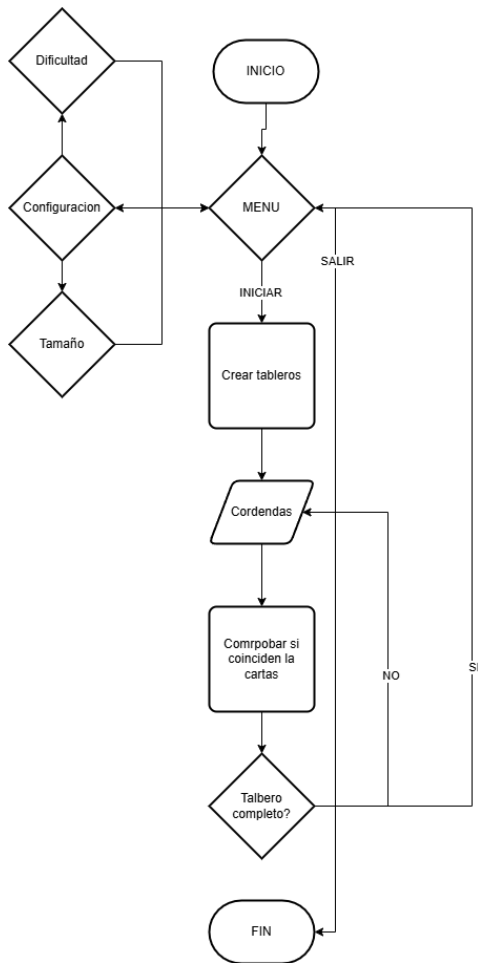
PerfilesFicheros: Contiene funciones para gestionar los perfiles con los ficheros

GestionDeDatos: Contiene funciones para gestionar los datos

GestionDePartidas: Contiene funciones para gestionar las partidas

GestionDePerfiles: Contiene funciones para gestionar los perfiles

Diagrama de flujo



Herramientas Utilizadas

1. **JDK 21:** Para la compilación y ejecución del código.
2. **Visual Studio Code:** Como entorno de desarrollo integrado (IDE).
3. **MYSQL:** Como gestor de base de datos

Estado de implementación

Caso de Uso	Estado	Detalles
Creación del Tablero	Implementado	Funciona correctamente. Todas las condiciones comprobadas.
Selección de coordenadas	Implementado	
Verificación de Victoria	Implementado	
Cambiar Cartas	Implementado	
Crear Perfil	Implementado	
Cargar Perfil	Implementado	
Borrar Perfil	Implementado	
Crear Partida	Implementado	
Guardar PartidaBD	Implementado	
Borrar PartidaBD	Implementado	
Cargar PartidaBD	Implementado	
Guardar PartidaFicheros	Implementado	
Borrar PartidaFicheros	Implementado	
Cargar Partida Ficheros		

Fuentes:

Información para intentar borrar la terminal:

<https://es.stackoverflow.com/questions/529856/limpiar-la-consola-en-java>

Joshin y Federico Meza

Texto a ASCII para generar textos:

<https://patorjk.com/software/taag/#p=display&f=Graffiti&t=>

Patrick Gillespie

<https://tremend.com/blog/enterprise-solutions/untitledserializable-java-objects-in-mysql/#:~:text=If%20you%20want%20to%20save,save%20it%20in%20the%20database.&text=Attention%20the%20table%20column%20must%20be%20BLOB%2C%20not%20varchar.>