

PROYECTO II

Parcial II

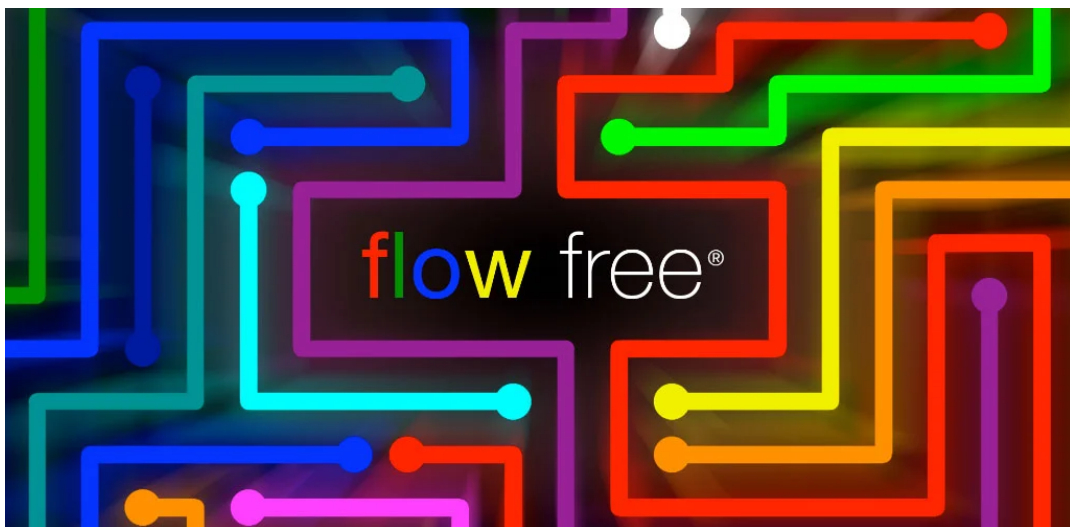
1. Introducción

El presente documento establece las pautas para la creación de un juego 2D en Java utilizando la librería libgdx u otra alternativa de desarrollo de videojuegos. Se deben implementar conceptos de archivos binarios, hilos, programación orientada a objetos (POO) y diseño abstracto. Además, el sistema debe contar con un mecanismo de registro de usuarios, almacenamiento de datos estadísticos y una progresión de niveles basada en el desempeño del jugador.

2. Selección del Juego

El desarrollador podrá elegir entre los siguientes tipos de juegos:

- **Flow Free:** Un juego de conexión de puntos con restricciones específicas.



- **Cut the Rope:** Un juego basado en física donde el usuario debe cortar cuerdas para alimentar a un personaje.



- **Sokoban:** Un juego de empuje de bloques donde el jugador debe organizar cajas en posiciones designadas.



3. Requerimientos del Proyecto

3.1 Tecnologías Utilizadas

- **Lenguaje de Programación:** Java
- **Motor de Juego:** libgdx u otra librería compatible
- **Persistencia de Datos:** Archivos binarios
- **Concurrencia:** Hilos
- **Paradigmas de Programación:** Programación Orientada a Objetos y diseño abstracto.

3.2 Registro de Usuarios

- Cada usuario de contener su propia carpeta con sus propios archivos.
- Datos:
- Los datos que se deben almacenar para cada usuario incluyen:
 - Nombre de usuario - Identificador único para cada jugador.
 - Contraseña - Almacenada de manera segura.
 - Nombre del usuario – Nombre Completo.
 - Fecha de registro - Para llevar un seguimiento del tiempo de actividad del usuario.
 - Última sesión iniciada - Para análisis del tiempo transcurrido entre sesiones.
 - Progreso del juego - Información sobre niveles completados y puntajes obtenidos.
 - Tiempo total jugado - Para estadísticas del usuario.
 - Historial de partidas - Registro detallado de intentos y logros obtenidos en cada sesión.
 - Preferencias de juego - Configuraciones personalizadas como volumen, idioma y controles.
 - Avatar o imagen de perfil - Opción para personalización del usuario.
 - Ranking o puntuación general - Para mostrar el desempeño del usuario en comparación con otros.
 - Amigos o rivales - Lista de usuarios con los que se puede comparar estadísticas o jugar en modo competitivo.
- La información de los jugadores se almacenará en archivos binarios.

- Debe existir la posibilidad de iniciar sesión con un usuario registrado para recuperar su progreso.
- Se permitirá la creación de perfiles personalizados donde el usuario podrá elegir un avatar y modificar configuraciones de juego.

3.3 Datos Estadísticos

- Se debe registrar la cantidad de partidas jugadas por usuario.
- Se almacenará la cantidad de niveles completados y el tiempo promedio por nivel.
- El sistema debe proporcionar una manera de visualizar estos datos al usuario.
- Se debe incluir un sistema de ranking basado en el desempeño del jugador para fomentar la competencia.
- Los datos estadísticos deben actualizarse dinámicamente tras cada sesión de juego.
- Se podrá implementar una opción para que los usuarios comparen sus estadísticas con las de otros jugadores.

3.4 Progresión de Niveles

- El juego debe contar con un mínimo de **siete niveles distintos y subir el nivel de dificultad**.
- Los niveles se desbloquean progresivamente a medida que el usuario supera el nivel anterior.
- Cada nivel debe incrementar en dificultad con base en parámetros específicos según el tipo de juego elegido.
- Para ganar el juego, el usuario debe completar con éxito el último nivel disponible.
- Cada nivel puede implicar diferentes mecánicas de victoria según el juego elegido:
 - **Flow Free**: Conectar todos los puntos sin dejar espacios vacíos en la cuadrícula.

- **Cut the Rope:** Lograr que el caramelo llegue al personaje recolectando la mayor cantidad de estrellas posible.
- **Sokoban:** Ubicar todas las cajas en las posiciones objetivo en el menor número de movimientos.
- Al finalizar el último nivel, el sistema debe mostrar un mensaje de victoria y guardar las estadísticas del usuario.
- Se podrá incluir una tabla de puntuaciones para que los jugadores comparen su rendimiento con otros usuarios.

4. Implementación de los Componentes

4.1 Programación Orientada a Objetos y Abstracción

- Se debe estructurar el código mediante clases y objetos.
- Se implementará una clase base abstracta Juego que será extendida por las clases específicas de los juegos seleccionados.
- Cada entidad del juego (jugador, nivel, obstáculo, etc.) Debe ser representada mediante clases específicas.

4.2 Uso de Hilos

- Se emplearán hilos para manejar eventos en tiempo real, como la detección de colisiones, la generación de elementos y la lógica del juego.
- Se debe asegurar una correcta sincronización para evitar problemas de concurrencia.

4.3 Manejo de Archivos Binarios

- Los datos de los jugadores y estadísticas se almacenarán en archivos binarios.

- Se utilizará la serialización de objetos en Java para el almacenamiento y recuperación de datos.

5. Evaluación del Proyecto

Se evaluará el correcto funcionamiento del juego de acuerdo con los siguientes criterios:

- Implementación adecuada de POO y diseño abstracto.
- Uso eficiente de hilos en la gestión del juego.
- Correcta manipulación de archivos binarios para persistencia de datos.
- Desarrollo de una experiencia de usuario fluida y progresiva con desbloqueo de niveles.
- Integración de un sistema de registro y visualización de estadísticas.