

1. Introducción

Este informe detalla el análisis del proyecto Mokepon World, un juego inspirado en Pokémon, junto con una explicación de la metodología en cascada (Waterfall) y cómo se aplica al desarrollo de software. También se incluyen detalles técnicos sobre el código, la estructura del proyecto y las recomendaciones para mejorar el desarrollo del juego.

2. Metodología en Cascada

La metodología en cascada, también conocida como modelo Waterfall, es un proceso de desarrollo secuencial que sigue las siguientes etapas: 1.

****Recopilación y análisis de requisitos****: Se identifican y documentan todas las necesidades del cliente. 2. ****Diseño del sistema****: Se crea un diseño detallado del sistema, incluyendo arquitectura y componentes. 3. ****Implantación****: Se desarrolla el sistema basado en el diseño. 4. ****Pruebas****: Se realizan pruebas exhaustivas para garantizar la calidad. 5. ****Despliegue****: El sistema se entrega al cliente y se implementa en producción. 6. ****Mantenimiento****: Se realizan ajustes y actualizaciones según sea necesario. ****Características****: * Es una secuencia rígida de pasos. * Se ha aplicado con éxito en la industria de la construcción. * Es conocida por su uso en el desarrollo de software.

****Cuándo se aplica****: * Cuando el objetivo final del proyecto está bien definido. * Cuando no hay restricciones de presupuesto ni de tiempo. * Cuando los clientes no pueden cambiar el alcance del proyecto al comenzar. * Cuando no existen requisitos ambiguos. ****Ventajas****: * Produce resultados coherentes, fiables y controlados. * Ideal para proyectos con objetivos claros y requisitos bien definidos. ****Inconvenientes****: * Falta de flexibilidad para cambios durante el desarrollo. * Los errores detectados en etapas avanzadas pueden ser costosos de corregir.

3. Análisis del Proyecto Mokepon World

El proyecto Mokepon World es un juego que combina desarrollo de software, lógica de juego y animaciones. A continuación, se describen los aspectos clave del proyecto: ****Estructura del proyecto****: - El proyecto está organizado en carpetas para activos, código fuente y pruebas. - Incluye archivos para HTML, CSS y JavaScript, con subcarpetas para animaciones, lógica y controladores. * ****Lógica del juego****: - Los enemigos tienen animaciones controladas dinámicamente mediante intervalos. - El sistema de combate incluye ventajas de

Informe Detallado: Desarrollo de Mokepon World

tipo, lógica de ataques y mecanismos de reinicio. * **Clases y objetos**:

- Los Mokepons son objetos que encapsulan propiedades como tipos, ataques y posiciones.
- Se utilizan clases para definir la estructura y comportamiento de los Mokepons.
- * **Recomendaciones**:
 - Refactorizar la inicialización de Mokepons utilizando funciones de fábrica.
 - Optimizar la lógica de combate reemplazando condiciones rígidas con tablas de búsqueda.
 - Agrupar variables relacionadas con el combate en objetos para mejorar la legibilidad.

4. Detalles del Código

El código del proyecto Mokepon World incluye las siguientes características técnicas:

* **Inicialización de Mokepons**:

Los Mokepons se crean utilizando una clase que define sus propiedades y métodos. Ejemplo:

```
class Mokepon:
def __init__(self, image, input_id, label_id, name, x=0, y=0):
self.image = image
self.input_id = input_id
self.label_id = label_id
self.name = name
self.x = x
self.y = y
```

* **Animaciones de enemigos**:

Las animaciones se controlan mediante intervalos dinámicos. Ejemplo:

```
function
animateEnemy(enemyId, direction, duration) {
const intervalKey =
`enemy${enemyId}_${direction}`;
intervalsIdList[intervalKey] =
setInterval(() => {
updateEnemyPosition(enemyId, direction);
}, 50);
setTimeout(() => {
clearInterval(intervalsIdList[intervalKey]);
animateEnemy(enemyId,
-direction, duration);
}, duration);
}
```

* **Sistema de combate**:

El sistema de combate utiliza ventajas de tipo para determinar el ganador. Ejemplo:

```
const typeAdvantages = {
fire: ['ice',
'steel'],
water: ['fire', 'earth'],
earth: ['thunder', 'steel'],
};
function checkTypeAdvantage(type1, type2) {
return
typeAdvantages[type1]?.includes(type2);
}
```

5. Conclusión

El proyecto Mokepon World es un excelente ejemplo de cómo combinar desarrollo de software, lógica de juego y metodologías estructuradas como Waterfall. Las recomendaciones proporcionadas pueden ayudar a optimizar el código y mejorar la experiencia del usuario. La metodología en cascada es ideal para este tipo de proyectos debido a su enfoque estructurado y predecible.