“Mythic Memory”

Índice

1. Resumen Ejecutivo

Memoria Mítica es un juego de memoria interactiva que fusiona elementos medievales, de fantasía y mitología. Diseñado para desafiar la agilidad mental del jugador, este juego propone encontrar pares de cartas en un entorno ambientado en mundos épicos y legendarios. Cada acierto se premia con monedas de oro, mientras que los errores generan leves penalizaciones. La mecánica única radica en que el temporizador, con una duración base de 1:20, comienza únicamente al voltear la primera carta, otorgando al jugador el control sobre el inicio del desafío.

2. Objetivos y Funcionalidades

- **Objetivo Principal:**
  Encontrar todos los pares de cartas en el menor tiempo posible y acumular monedas de oro para adquirir bonos en la tienda interna del juego.
- **Funcionalidades Clave:**
  • **Desafío de Memoria:** El jugador debe encontrar 12 pares de cartas en un tablero de 24 tarjetas.
  • **Sistema de Monedas:** Se otorgan 10 monedas por cada acierto y se descuenta 1 moneda por cada error, sin permitir que el saldo baje de cero.
  • **Temporizador Interactivo:** El reloj inicia al voltear la primera carta, permitiendo iniciar el juego cuando el jugador lo desee.
  • **Tienda de Bonus:** Al finalizar la partida, el jugador puede acceder a una tienda para comprar mejoras, como +10 segundos adicionales para la próxima partida, utilizando las monedas acumuladas.
  • **Efectos de Audio y Animaciones:** Se incorporan sonidos y transiciones visuales que enriquecen la experiencia de juego y sumergen al jugador en una atmósfera mítica.

3. Arquitectura y Organización del Proyecto

El juego se ha desarrollado siguiendo un enfoque modular que facilita su mantenimiento y escalabilidad. La estructura se divide en tres componentes principales:

3.1 HTML

- Defina la estructura de la aplicación.
- Contiene contenedores individuales para cada pantalla (inicio, juego, fin, tienda).
- Incorpora elementos multimedia (por ejemplo, etiquetas <audio>) para la retroalimentación sonora.

3.2 CSS

- Gestiona la presentación visual del juego.
- Emplea técnicas modernas como CSS Grid y Flexbox para la disposición del contenido.
- Implementa animaciones y transiciones (por ejemplo, el efecto “flip” de las cartas) que evocan una estética medieval y fantástica, utilizando colores oscuros, dorados y texturas inspiradas en pergaminos antiguos.

3.3 JavaScript

- Controla la lógica y el flujo del juego.
- Implementa el algoritmo Fisher-Yates para mezclar las cartas de forma aleatoria.
- Gestiona el temporizador, que se activa al voltea la primera carta.
- Coordina la interacción del usuario, la verificación de pares y la lógica de la tienda de mejoras.

4. Detalle de la Implementación

El código del juego ha sido cuidadosamente modularizado para facilitar su mantenimiento y futuras ampliaciones. Cada funcionalidad se encuentra implementada en funciones específicas que gestionan tareas como:

- El voltear de las cartas y el control de estados (primer y segundo clic).
- La verificación de coincidencias y el manejo de aciertos/errores.

- La activación y control del temporizador, asegurando que se iniciará únicamente cuando el jugador se sienta listo.
- La integración de la tienda, permitiendo comprar bonos y aplicar mejoras en partidas futuras.

Esta organización permite que cada parte del sistema se desarrolle y se despliegue de forma independiente, lo que simplifica tanto la identificación de errores como la incorporación de nuevas características.

5. Técnicas de Implementación Aplicadas

En Memoria Mítica se han aplicado diversas técnicas tanto en el desarrollo como en el despliegue del sistema, que incluyen:

- **Implementación por Localidades Individuales:**
  Cada componente del juego (estructura, presentación y lógica) se encuentra en archivos separados (HTML, CSS y JavaScript). Esto permite un desarrollo modular, donde cada "localidad" o módulo del sistema se desarrolla, prueba y despliega de manera independiente, facilitando actualizaciones sin afectar el resto del sistema.
- **Implementación Directa:**
  La interacción con el DOM se realiza de forma directa mediante eventos (addEventListener), lo que permite respuestas inmediatas a las acciones del usuario. Esta técnica favorece un flujo de datos claro y una manipulación directa de los elementos visuales.
- **Técnicas de Despliegue de Sistemas:**
  • **Despliegue Incremental:** Cada módulo del juego se implementa y prueba de forma individual antes de integrarlo en el sistema completo. Esto permite detectar y corregir errores en fases tempranas.
  • **Despliegue Automatizado:** Se puede integrar en un pipeline de CI/CD (Integración y Despliegue Continuos) que actualiza automáticamente la versión en producción tras la verificación de cada componente.
  • **Uso de Contenedores:** La arquitectura modular se presta para un despliegue en

contenedores (por ejemplo, Docker), facilitando la escalabilidad y la replicación del entorno de desarrollo en producción.

- **Efectos Visuales Avanzados:**
  La utilización de CSS Grid, Flexbox y transformaciones 3D permite una organización visual atractiva y coherente con la temática del juego, integrándose de manera fluida con la lógica de JavaScript.

6. Conclusiones y Posibles Mejoras

Memoria Mítica es un ejemplo de cómo se pueden combinar diversas temáticas (medieval, fantasía y mitología) para crear una experiencia de juego única y envolvente. La implementación modular y las técnicas de implementación aplicadas aseguran que el sistema sea escalable, fácil de mantener y adaptable a futuras mejoras.

Posibles Mejoras Futuras:

- Ampliar la tienda con nuevos bonos y elementos de personalización.
- Incluye modos de juego adicionales o niveles de dificultad.
- Integrar almacenamiento persistente (por ejemplo, localStorage o bases de datos) para conservar el progreso del usuario entre sesiones.
- Optimizar el despliegue utilizando contenedores y pipelines de CI/CD para automatizar actualizaciones.