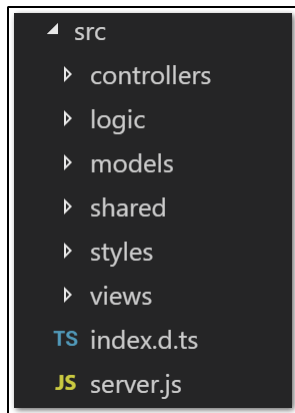


Trabajo Practico 1 – Fuera luces

Prerequisitos

1. Instalar NodeJS LTS version ([link](#))
2. Descomprimir el código. También es posible clonar el repositorio de GitHub ([link](#))
3. Abrir una terminal de CMD/Powershell/Git bash y navegar a la carpeta raíz del proyecto
4. Ejecutar `npm install` para instalar las dependencias necesarias
5. Ejecutar `npm start` para compilar el código e iniciar el servidor local
6. Abrir un navegador y dirigirse a <http://localhost:8080>

Estructura Inicial



- **Controllers:** contiene las clases que va a interactuar con la interfaz
- **Logic:** contiene clases de ayuda a utilizar por el controlador
- **Models:** contiene los modelos de las entidades
- **Shared:** contiene clases de ayuda para el proyecto en general
- **Styles:** contiene las clases de CSS para dar estilo a las vistas
- **Views:** contiene la vista inicial y los templates que se agregaran dinámicamente a la misma
- **Index.d.ts:** define una interfaz para extender la clase `HTMLElement`
- **Server.js:** inicia un servidor local que escucha en el puerto 8080

El proyecto

Controllers – AppController

Se encarga de conectar y actualizar de forma dinámica la interfaz. Al iniciarse agrega los controles iniciales a la vista y define los eventos de los mismos, tales como generar el tablero al clicar el botón de jugar. Tiene una instancia de `BoardManager`, clase que se ocupa de toda la lógica relacionada al tablero.

Logic – Http

Esta clase encapsula las llamadas HTTP GET de la clase `XMLHttpRequest` dentro de una `Promise`, con el objetivo de hacer el código más limpio y reutilizable.

Logic – BoardManager

Esta clase se ocupa de toda la lógica relacionada a la creación y manejo del tablero, así como del conteo de movimientos. Contiene una instancia de TileManager, clase que se ocupa de manejar los eventos relacionados a las Casillas tales como reaccionar al click de un usuario; y de GameSolver, clase que se ocupa de calcular y proveer las pistas.

Logic – TileManager

Pensada como dependencia de BoardManager, esta clase maneja los eventos relacionados a las casillas, tales como cambiar el estado de las especificadas por BoardManager cuando el usuario clickea o llamar a BoardManager al terminar un turno para saber si el juego termino y actualizar la interfaz en consecuencia.

Logic – GameSolver

Créditos para http://perfectweb.org/ddo/solver/vale_puzzle.html por proveer la lógica algebraica responsable de calcular la siguiente casilla a clickear. Parte del código de esa página fue copiado, refactorizado y adaptado para este proyecto.

Provee la lógica necesaria para calcular la próxima casilla a clickear.

Models – CustomElement

Implementación de la interfaz definida en index.d.ts para permitir agregar un child HTMLElement definido como string a la clase HTMLElement.

Models - Tile

Modelo de una casilla.

Shared – Startup

Crea una instancia de AppController al abrirse la página.

Shared – Constants

Contiene los IDs de los elementos, de forma que solamente sea necesario cambiarlos acá y no en todo el código si alguno es modificado.

Styles – Index.css

Estilos generales de la aplicación.

Styles – Tile.css

Contiene los estilos relacionados a las casillas.

Views – Index.html

Página inicial de la aplicación. Todos los componentes de la UI serán añadidos dinámicamente dentro del elemento *app*.

Views – Resto de los archivos

Son los templates de los controles que van a ser añadidos dinámicamente a la UI. Poseen ids únicos que, en el caso de los elementos no únicos, como las filas y las casillas, son asignados por el código de negocio al momento de crearlos. Esto permite tener elementos reutilizables y fácilmente modificables, así como una UI altamente dinámica.

Funcionamiento

La idea general es tener plantillas de HTML para cada componente la interfaz, que tienen IDs únicos mediante los cuales son conectadas a la lógica de negocio por medio de event listeners. Al crearse el ApplicationController, este agrega los controles iniciales a la interfaz.

El usuario puede elegir jugar con un tablero de entre 3 y 6 casillas por fila especificando la cantidad en el input inicial. Al clicar en JUGAR, la instancia de BoardManager se ocupa de generar el tablero, el resto de los elementos necesarios para el juego y agregar los evento listeners. Todo el proceso es reutilizable.

Luego la instancia de TileManager del BoardManager se encarga de crear cada casilla y agregar los eventos necesarios para reaccionar al click del usuario, tales como el cambio de color.

Cuando el usuario clickea una casilla, BoardManager agrega uno al contador de movimientos, calcula cuales casillas son las que tienen que cambiar, le pasa la información a TileManager y modifica la matriz que contiene del estado de las casillas. A su vez, los eventos definidos en TileManager actualizan la interfaz.

Clickear en PISTA llama al método *solve* de la instancia de GameSolver, que calcula la casilla que se debe clicar y devuelve su ID. Luego el BoardManager actualiza el tablero para iluminar la casilla a clicar.

Cuando todas las casillas están apagadas, se setea el record y se muestra el mensaje de felicitaciones. Clickear en RESET te lleva a la pantalla inicial, reseteando los movimientos y guardando el record.