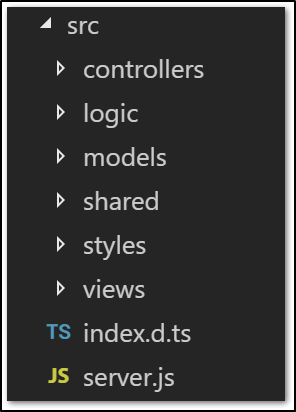
Trabajo Practico 1 – Fuera luces

Prerequisitos

1. Instalar NodeJS LTS version ([link](https://nodejs.org/es/))
2. Descomprimir el codigo. Tambien es posible clonar el repositorio de GitHub ([link](https://github.com/Pierca7/tp1-programacion-3))
3. Abrir una terminal de CMD/Powershell/Git bash y navegar a la carpeta raiz del proyecto
4. Ejecutar *npm install* para instalar las dependencias necesarias
5. Ejecutar npm start para compilar el codigo e iniciar el servidor local
6. Abrir un navegador y dirigirse a <http://localhost:8080>

Estructura inicial



* Controllers: contiene las clase que va a interactuar con la interfaz
* Logic: contiene clases de ayuda a utilizer 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 las vista inicial y los templates que se agregaran dinamicamente 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

Controllers – AppController

Se encarga de conectar y actualizar de forma dinamica la interfaz. Al iniciarse agrega los controles iniciales a la vista y define los eventos de los mismos, tales como generar el tablero al clickear el boton de jugar. Tiene una instancia de BoardManager, clase que se ocupa de toda la logica 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 codigo mas limpio y reutilizable.

Logic – BoardManager

Esta clase se ocupa de toda la logica relacionada a la creacion y manejo del tablero, asi 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 calculary y proveer las pistas.

Logic – TileManager

Pensada como dependecia 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

Creditos para <http://perfectweb.org/ddo/solver/vale_puzzle.html> por proveer la logica algebraica responsable de calcular la siguiente casilla a clickear. Parte del codigo de esa pagina fue copiado, refactorizado y adaptado para este proyecto.

Provee la logica necesaria para calcular la proxima casilla a clickear.

Models – CustomElement

Implementacion 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 pagina.

Shared – Constants

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

Styles – Index.css

Estilos generales de la aplicacion.

Styles – Tile.css

Contiene los estilos relacionados a las casillas.

Views – Index.html

Pagina inicial de la applicacion. Todos los componentes de la UI seran añadidos dinamicamente dentro del elemento “app”.

Views – Resto de los archivos

Son los templates de los controles que van a ser añadidos dinamicamente a la UI. Poseen ids unicos que en el caso de los elementos no unicos, como las filas y las casillas, son asignados por el codigo de negocio al momento de crearlos. Esto permite tener elementos reutilizables y facilmente modificables asi como una UI altamente dinamica.