Hundir la flota

El programa está completamente realizado con el lenguaje de programación TypeScript según los estándares ECMAScript 2015.

El programa está creado con el SDK de TypeScript proporcionado por Microsoft con la API de TypeScript: https://www.typescriptlang.org/docs/

AVISO: Para ejecutar el proyecto hay que montarlo en un sistema de gestión de servidor web como puede ser Apache HTTP Server por que Typescript usa las sentencias *import* como esta:

```
import {Jugador} from './actores/Jugador.js';
```

Esto es debido a que el navegador usa peticiones http para recuperar los módulos requeridos por las sentencias import del código compilado JavaScript y si el sistema no está montado en un servidor, el navegador no es capaz de recuperarlos.

Transpilación de códigos TypeScript

TypeScript es un lenguaje que se transpila a JavaScript y tiene plena inter operatividad con códigos JavaScript, es decir puedes hacer que en medio de un código Typescript haya código JavaScript sin problemas de compatibilidad siempre y cuando se tenga bien configurado el proyecto.

Para poder transpilar el proyecto hay que usar Node.Js instalando a Node el transpilador de Typescript (tsc) por medio del comando `npm install typescript --save-dev`

Para una correcta transpilación hay que crear y configurar de manera correcta en los ficheros package-lock.json, package.json y tsconfig.json.

No voy a entrar en detalle de como configurar estos ficheros.

Estructura del proyecto:

- La estructura de ficheros tiene los directorios habituales de un proyecto front web como los directorios ./css, ./img, ./sound y el fichero index.html que es el punto de ejecución de la aplicación.
- Pero también hay ficheros de configuración de Node.js antes mencionados: package-lock.json, package.json y tsconfig.json y el directorio ./node_modules.
- El código fuente está en el directorio ./src donde se encuentran los códigos fuente y packages Typescript.
- El código de producción JavasScript transpilado de los códigos fuente TypeScript se encuentra en el directorio ./dist y son estos códigos fuente los que está linkeados con el index.html

Por Enrique Sánchez Vicente, estudiante de 2ºDAM Colegio Santísima Trinidad, Salamanca.

TypeScript Doc

Para generar la documentación de los códigos fuente he usado la herramienta TypeDoc que corre sobre Node.Js y tiene múltiples opciones de configuración de la documentación de métodos, funciones, clases, contantes, interfaces etc.

Me ha parecido la herramienta más completa que he encontrado para documentar un proyecto de TypeScript, la única pega que le pongo es que es una herramienta de terminal y hay que mirar bien la documentación para usarla correctamente.

Link: https://typedoc.org

Análisis del proyecto

Todo el proyecto está diseñado según el paradigma de la programación orientada a objeto exceptuando el punto de entrada al programa, el index.ts que sigue una programación orientada a eventos y procedural.

La arquitectura de diseño que he seguido para conformar los artefactos del programa han sido el diseño por componentes, lo que se pretende es que todas las entidades gráficas tengan una representación de clase en código para poder manejar estos elemento gráficos como instancias, lo que hace al sistema muy modular y extensible.

A su vez los componentes del sistema pueden estar conformados por más componentes como asociaciones de agregación o asociaciones de composición con sus componentes padre.

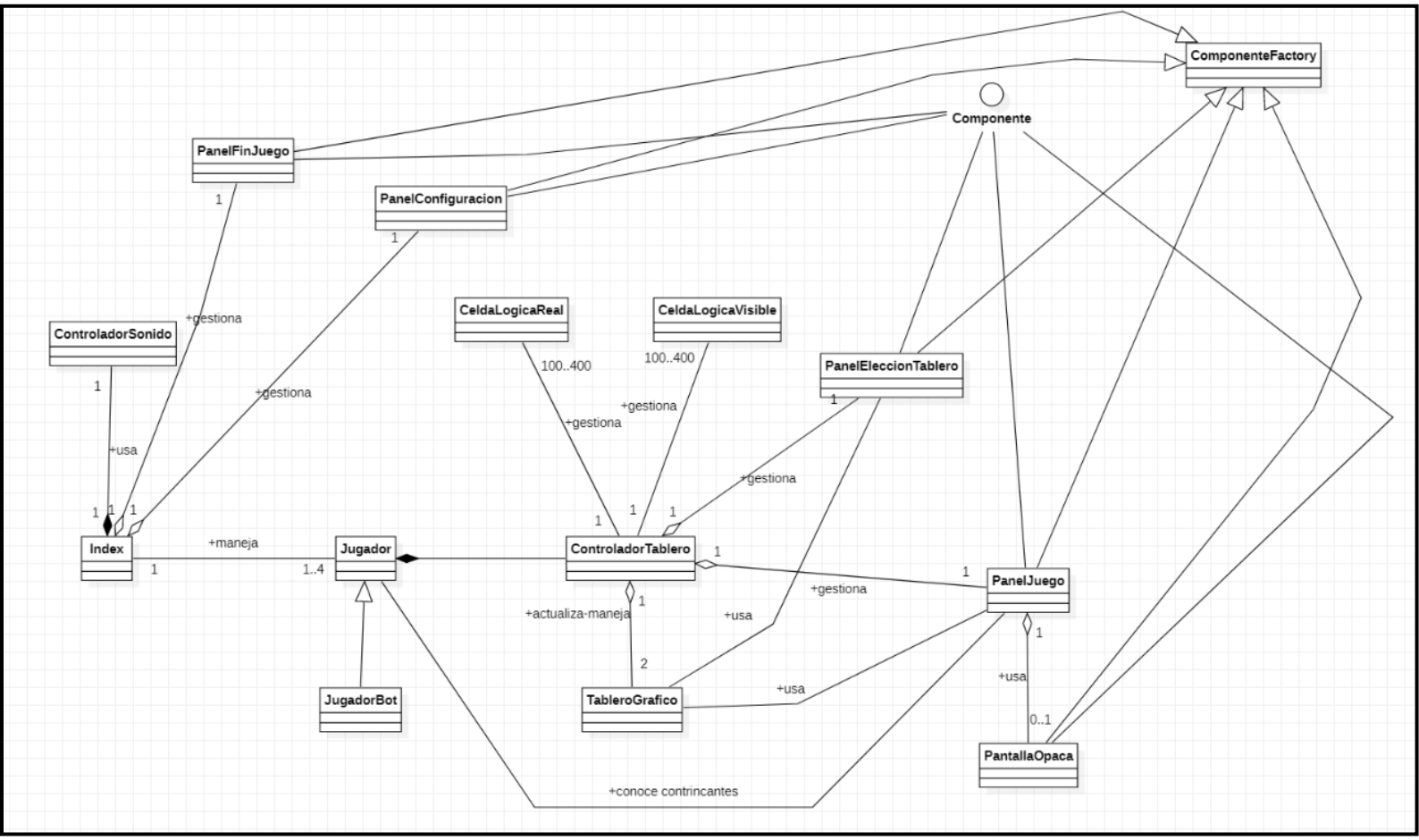
En cierta medida el diseño del programa funciona como un framework para la gestión de componentes gráficos que facilita la manipulación del DOM y de los elementos gráficos que se van a manejar. La metodología que sigue esta arquitectura de componentes es que todo componente padre es el encargado de gestionar en el DOM a todos los componentes hijos con los que se asocia.

Todas las clases componente implementan la interfaz Componente que obliga a implementar una interfaz de comunicación estándar con cualquier componente para realiza las acciones genéricas de despliegue y eliminación sobre los nodos del DOM.

Cualquier componente puede o no heredar de la clase ComponenteFacotory en función de la necesidad que tenga, esta clase sirve para poder crear sub elementos en el DOM que van a conformar la composición del componente, de esta manera un componente que es un panel gráfico de opciones tiene la capacidad de generar y adherirse subcomponentes del DOM como inputs de texto o botones de manera autosuficiente sin usar a otros artefactos del sistema.

Esta clase intenta tener métodos genéricos de creación de subcomponentes que permita a las clases específicas que hereden de ella generar estos subcomponentes con capacidad de personalización suficiente para satisfacer los requisitos de cada especificación

Diagrama de clases



Recuento de requisitos

Por falta de tiempo en el desarrollo no he podido concluir con todos los puntos de los requisitos planteados para el sistema pero he añadido bastantes requisitos extraordinarios.

Requisitos satisfechos:

- Sistema de sonido.
- Diseño por componentes.
- Uso de un lenguaje derivado de JavaScript.
- Multijugador hasta 4 jugadores.
- Uso de transiciones CSS.
- Configuración de diferente número de casillas para los tableros.

Los requisitos que no se han realizado son:

- Sistema de puntuación para los jugadores dependiendo del número de barcos hundidos (en este caso gana el jugador que quede vivo).
- Listado del orden de puntuación de los jugadores en el panel de finalización de una partida.
- CSS responsive.

Por Enrique Sánchez Vicente, estudiante de 2ºDAM Colegio Santísima Trinidad, Salamanca.