Challenge MeLi

En relación al desafío planteado sobre el código morse se adjunta la siguiente solución, compuesta por:

**MorseCode**: Es una librería con un helper estático con los métodos necesarios para la conversión de morse a humano (translate2Human), humano a morse (translate2Morse) y bits a morse (decodeBits2Morse).

**MorseCodeTests**: Son unos tests unitarios que arme para las primeras construcciones y pruebas de los métodos implementados en **MorseCode**.

**ApiMorseCode**: Es la Web Api que expone los métodos 2text y 2morse, como se solicitaba en el ejercicio, aunque no esta subida al app engine.

**Telegrafo**: Es una aplicación web con MVC y jquery que implementa una UI simple para la generación de código morse. El mismo funciona generando a través de un botón las señales en formato de bits. Esos bits luego pueden transformarse a código morse o lenguaje humano a través de unos métodos incluidos en la misma que utilizan **MorseCode**. No están consultando a la Web Api directo porque esta parte del ejercicio entendí era independiente de la generación o no de la web api.

**Consideraciones generales de la solución:**

Tomando en cuenta la definición del código morse que pude leer en distintos sitios, establecí que el código más allá de quien lo origine debería cumplir cierta frecuencia que permita interpretarlo. Lo definido para el mismo en la información es que nos encontramos con puntos, rayas, pausa entre elementos de un mismo código, pausa entre códigos y pausa entre lo que serían las palabras.

Estas definiciones estiman que un punto es la menor representación.

La raya es aproximadamente el equivalente a 3 puntos.

La pausa entre elementos de un mismo código es equivalente a un punto, la pausa entre códigos es el equivalente a tres puntos y la pausa entre palabras es el equivalente a 3 rayas.

Basándome en esto y la representación en bits tome como punto de partida:

1 = .

111 = -

0 = separación entre elementos del mismo código

000 = separación entre códigos (letras)

000000000 = separación entre palabras

Adicionalmente y considerando lo mencionado en el desafío sobre la variedad de frecuencias por operador, pero a la vez la garantía de que cada mensaje manejara la misma frecuencia durante lo que dura el mensaje, es que la rutina lo que busca es evaluar las señales y del conjunto determinar cuales son puntos y cuales rayas. Asumiendo que si solo llegara 111 seria considerada raya, a menos que luego aparezca otra señal con una longitud de 3 veces mas (aproximadamente), en este caso los 111 serian interpretados como punto y la longitud mas grande como raya.

En cuanto a la interfaz web, arme una página donde se empieza a generar el código luego de iniciar haciendo click en START.

Una vez iniciado, podemos clickear sobre el circulo manteniendo presionado el mouse para indicar el tipo de señal que queremos generar.

Para ser un poco más claro, el botón muestra un punto pero si la longitud de la señal se hace mas larga, cambiara por una raya para indicar que estamos en condiciones de ya transmitir una raya.

Cada vez que se suelta el botón, se mostrara en la sección de BITS los bits generados hasta el momento y se pondrá en funcionamiento el cronometro que evaluara la longitud de la pausa.

Del mismo modo, dividí la transición del color del circulo de forma que si la pausa es corta (separación entre elementos de un mismo código) se mostrara un color, luego cuando la pausa es un poco más larga (separación entre códigos) cambiara a otro color y por último cuando es más larga (Separación entre palabras) cambia a otro color.

Luego de que la pausa sea mas larga que las mencionadas en el párrafo anterior, la interfaz da por finalizada la generación del mensaje y mostrara un botón para traducir el mensaje (SEND).

El botón de SEND en esta ocasión esta haciendo un post al servidor y utilizando la librería MorseCode, para devolver el resultado de la conversión de los bits a Morse y a Humano para mostrarse en los paneles correspondientes.