為了處理 codabar 這個一維條碼的掃描問題

先去看了一下 html5-qrcode 套件中是有能設定 codabar 這個編碼的，  
所以很快樂的引用github最新版本並設定最新的<script>最新的連結。  
範例  
<script src="<https://unpkg.com/html5-qrcode>"></script>  
<style>

.QRCode-content{

display: flex;

justify-content: space-between;

align-items: flex-end;

flex-wrap: wrap;

}

@media (max-width: 540px){

.QRCode-content{

justify-content: center;

}

}

</style>

<div id="reader"> </div>

<script type="text/javascript">  
 Html5Qrcode.getCameras().then(devices => {

if (devices && devices.length) {

var cameraId = devices[0].id;

const html5QrCode = new Html5Qrcode("reader");

const qrCodeSuccessCallback = (decodedText, decodedResult) => {

//這裡做 掃到時的處理

html5QrCode.stop();//掃描到即終止, 否則會一直重複掃描callback

};

const config = {

fps: 25,

qrbox: {//掃描內層框大小設定

width: 200,

height: 200

},

aspectRatio:1.333,//比例大小4:3=>1.333 , 16:9=>1.7777778

formatsToSupport: [

Html5QrcodeSupportedFormats.QR\_CODE,

Html5QrcodeSupportedFormats.CODE\_39, // 支援 Code 39

Html5QrcodeSupportedFormats.CODE\_128 // 支援 Code 128

Html5QrcodeSupportedFormats.CODABAR // 支援 CODABAR

],

rememberLastUsedCamera:true//只跟使用者要一次權限(safari本身安全機制要去手機系統設定)

};

html5QrCode.start({

facingMode: "environment"

}, config, qrCodeSuccessCallback);

}

}).catch(err => {

// handle err

});  
</script>

依範例說明：  
Html5Qrcode.getCameras() 這個步驟其實是非必要的，目的只是為了能讓使用者選擇鏡頭做使用而已。  
只是範例沒有做使用而已，給之後如果要做的話可以使用，後面會說明在哪裡使用。  
  
if (devices && devices.length) 這裡的目的只是為了知道 有沒有鏡頭、有沒有權限而已。  
  
const html5QrCode = new Html5Qrcode("reader"); 指定html中哪個DOM來展開套件的功能  
  
const qrCodeSuccessCallback 包裝一個掃到資料後要執行的動作  
  
config 個別的設定

fps 每秒掃描幀數

qrbox 定義掃描框範圍 範例 { width: 200, height: 200 }

aspectRatio 設定相機畫面寬高比例 比例大小4:3=>1.333 , 16:9=>1.7777778

rememberLastUsedCamera 權限設定 但實測沒什麼用

formatsToSupport 使用的編碼設定 [Html5QrcodeSupportedFormats.QR\_CODE,Html5QrcodeSupportedFormats.CODE\_39]  
能設定的項目為  
enum Html5QrcodeSupportedFormats {

QR\_CODE = 0,

AZTEC,

CODABAR,

CODE\_39,

CODE\_93,

CODE\_128,

DATA\_MATRIX,

MAXICODE,

ITF,

EAN\_13,

EAN\_8,

PDF\_417,

RSS\_14,

RSS\_EXPANDED,

UPC\_A,

UPC\_E,

UPC\_EAN\_EXTENSION,

}

videoConstraints 自定義攝像頭的約束條件，精確指定攝像頭的配置  
範例：  
{ videoConstraints: { facingMode: "environment" } } // 使用後置相機

{ videoConstraints: { facingMode: "user" } } // 使用前置相機

{ videoConstraints: { width: { ideal: 1280 }, height: { ideal: 720 } } } // 設定解析度

後續測試這邊設定時，會有跑版問題，就不設定了(設定解析度)

html5QrCode.start({

facingMode: "environment"

}, config, qrCodeSuccessCallback);  
} 啟動說明  
html5QrCode.start(  
啟動ID是上面說的 var cameraId = devices[0].id 取得的資料，但可以指定固定前鏡頭或後鏡頭，測試如果只有一個鏡頭他就會開那個唯一的鏡頭了，做法就我上面範例的做法，要指定的話就{facingMode: "environment"} 變成 devices[0].id 或 cameraId  
,  
自己設定的項目，但不一定要包起來成一個常數 可以直接設定在裡面{fps: 25}  
,  
掃瞄後的成果會執行的地方(decodedText, decodedResult) => {}  
,  
掃描失敗的處理 (errorMessage) => {}  
)  
處理成功與失敗是非必要  
  
如果依上面的範例來新增掃描功能的話，沒意外會大跑版  
所以需要設定 css 來處理  
這邊是依”樂活商米網站”的頁面來調整的css的部分就不詳說了,畢竟是亂調到能用實際還是交給熟知的人來處理比較好  
<style>

.QRCode-content{

display: flex;

justify-content: space-between;

align-items: flex-end;

flex-wrap: wrap;

}

@media (max-width: 540px){

.QRCode-content{

justify-content: center;

}

}

</style>   
<div id="reader" style="max-width: 100%;"> </div> 這邊加入 style="max-width: 100%;"  
  
後續想調整為更好掃描到的問題，就調整 config 中的 fps 和 videoConstraints調整解析度，測下來是感覺對靈敏的提升不大，反而增加了處理效能與手機耗電，就不調整了。  
後續就決定找其他的套件看能否處理 codabar 的問題。  
發現 Html5QrcodeSupportedFormats 中能設定 codabar 太好了直接拿來用。  
但掃描很久都沒成功的資料出現，於是我看到這個issues  
<https://github.com/mebjas/html5-qrcode/issues/803>  
簡單來說 codabar 掃不到的問題，但沒處理修正與新增。  
後續也發現 黑底白圖的QRcode此套件也無法掃描的issues，  
所以就先放棄這個套件後續的測試與看文件了  
以上是 html5-qrcode 看文件與查資料 測出來的東西  
  
  
接下來去看了 QuaggaJS 這個套件，起初如果要做一維/二維掃描做切換，那使用這個套件跟上面那個插件切換使用就好了，不必再做下去。  
範例  
<script src="<https://serratus.github.io/quaggaJS/dist/quagga.min.js>"></script>  
<style>

#reader {

position: relative;

}

#reader video {

width: 100%; /\* 讓視頻填滿容器 \*/

height: 100%; /\* 高度隨比例調整 \*/

top: 0;

left: 0;

}

#reader canvas {

position: absolute;

width: 100%; /\* 讓視頻填滿容器 \*/

height: 100%; /\* 高度隨比例調整 \*/

top: 0;

left: 0;

}

</style>   
<div id="reader" style="max-width: 100%;"> </div>  
<script type="text/javascript">  
 Quagga.init({

inputStream: {

name: "Live",

type: "LiveStream",

target: document.querySelector("#reader"), // 預覽框

constraints: {

facingMode: "environment" // 後置相機

}

},

decoder: {

readers: ["code\_128\_reader","code\_39\_reader","code\_39\_vin\_reader","codabar\_reader"] // 指定 Codabar 條碼格式

},

// locate: false // 禁用定位框和輔助繪圖

}, (err) => {

if (err) {

console.error("Codabar 掃描初始化失敗:", err);

return;

}

Quagga.start();

});

Quagga.onDetected((result) => {

const code = result.codeResult.code; // 條碼內容

const format = result.codeResult.format; // 條碼格式

console.log(`掃描成功！條碼內容：${code}, 格式：${format}`);

});  
</script>  
  
  
但事事難料，聽了說要不要去看上面那個套件的QRcode的解譯方式加進這個套件裡，就不用做切換了，所以在好奇心下我就去看了他原始碼有找到他做編碼的地方，但腦中沒有能插入並改寫的想法，於是先放棄的這個想像。

直到我找到了到Quagga2，  
Quagga2 是一位創作者修改Quagga的後續，原因是他聯繫不到 Quagga 原作者做git PR 所以自行包一個版本出來做使用(此創作者有在原作者裡詢問與徵求但沒響應)。  
  
其中我看到其中一個函式registerReader();，此函式看說明是能自己加編碼器到Quagga2中，前提是zxing這個底層套件有支援的格式，所以我往這邊像做新增。  
  
後續我也放棄用原來的’html5-qrcode’ 而是去用專心在QRcode格式上的套件’jsQR’來做加入Quagga2中作為額外套件來使用。  
  
但目前製作還沒完成 還在測試與實驗階段 後續成功在補上來

範例：  
<!-- <script src="https://cdn.jsdelivr.net/npm/@ericblade/quagga2/dist/quagga.min.js"></script> -->  
<script src="https://cdn.jsdelivr.net/npm/@ericblade/quagga2/dist/quagga.js"></script>  
<script src="<https://cdn.jsdelivr.net/npm/jsqr/dist/jsQR.js>"></script>  
<style>

#reader {

position: relative;

}

#reader video {

width: 100%; /\* 讓視頻填滿容器 \*/

height: 100%; /\* 高度隨比例調整 \*/

top: 0;

left: 0;

}

#reader canvas {

position: absolute;

width: 100%; /\* 讓視頻填滿容器 \*/

height: 100%; /\* 高度隨比例調整 \*/

top: 0;

left: 0;

}

</style>   
<script type="text/javascript">  
 class QrCodeReader {

FORMAT;

\_row;

config;

supplements;

constructor(config, supplements) {

this.\_row = [];

this.config = config || {};

this.supplements = supplements;

this.FORMAT = {

value: 'qr\_reader',

writeable: false,

};

return this;

}

decodeImage(inputImageWrapper) {

const data = inputImageWrapper.getAsRGBA();

const result = jsQR(data, inputImageWrapper.size.x, inputImageWrapper.size.y);

if (result === null) {

return null;

}

if (result.data === '') {

return null;

}

// TODO: translate result.location into same values as box/boxes from other readers?

return {

codeResult: {

code: result.data,

format: this.FORMAT.value,

}

};

}

decodePattern(pattern) {

// STUB, this is probably meaningless to QR, but needs to be implemented for Quagga, in case

// it thinks there's a potential barcode in the image

return null;

}

}

// 註冊解碼器到 Quagga2

Quagga.registerReader("qr\_reader", QrCodeReader);

</script>  
Quagga 與 Quagga2 中css修改方式是慢慢測試出來了，不一定是對的但在當下的需求上能使用的，主要是相機功能固定在規定的區塊內。  
  
有發現 Quagga2有範例 加入其他編碼器編譯的功能，但他是用 TS (TypeScript) 做製作的，  
Quagga.registerReader(“{編碼名稱}”,額外執行的class or function name)  
裡面的重點為  
class 名稱{  
constructor(config, supplements) {  
 這邊是相關設定 但沒特殊需求就沒詳細查詢代表意義 只修改有用到的 但也是用到

this.config = config || {}; 這個目前看到是都會有的  
 this.\_row = [];

this.supplements = supplements;

this.FORMAT = {

value: QR\_CODE,

writeable: false,

};

return this;

}  
decodeImage(inputImageWrapper) {  
 這邊主要就加入 其他編譯的地方了 回傳內容就是後續成功或失敗的內容  
 code = 編譯的格式或函示  
 內建的回傳格式  
 return null; 編譯失敗時，不要任何反應時就回傳null  
 codeResult: {

code: code, 編譯出來的內容

format: QR\_CODE, 顯回傳的格式

}  
}  
decodePattern(pattern) {

這裡其實 不知道作用 但可能是我編譯 QRCODE 所以沒反應吧?

return null;

}  
}

以上是 自己寫要額外編譯格式時的 class 製作方式  
詳細可以看 上面範例 做測試與操作 即可  
  
參考文件：  
1.html5-qrcode  
<https://github.com/mebjas/html5-qrcode/tree/master>  
2.quaggaJS  
<https://github.com/serratus/quaggaJS>  
3.quagga2  
<https://github.com/ericblade/quagga2>  
4.quagga2-reader-qr 加入 QRCode編碼範例  
<https://github.com/ericblade/quagga2-reader-qr>  
<https://www.npmjs.com/package/quagga2-reader-qr?activeTab=readme>  
5.php-zxing  
<https://github.com/dsiddharth2/php-zxing>  
6.zxing  
<https://github.com/zxing/zxing>

7.jsQR  
<https://github.com/cozmo/jsQR/blob/master/docs/jsQR.js>