### 簡介：

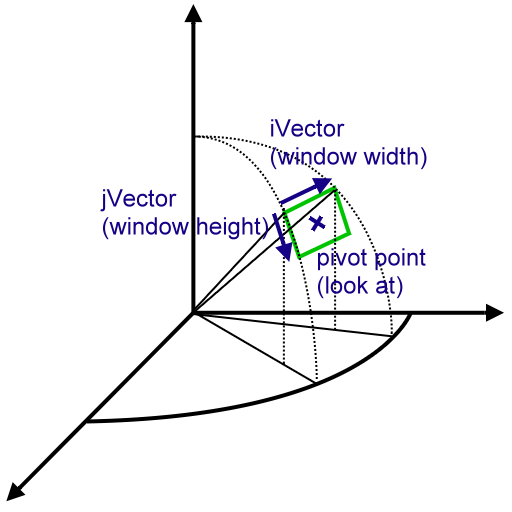
StarSky.js 是一個資料視覺化的工具，使用 processing.js 為基礎開發而成，其使用線性代數中的線性轉換矩陣，來定位各個星星在星空中的位置，以及星空的轉動。

此份文會說明星星在星空中的定位方法，以及星空的轉動的原理。

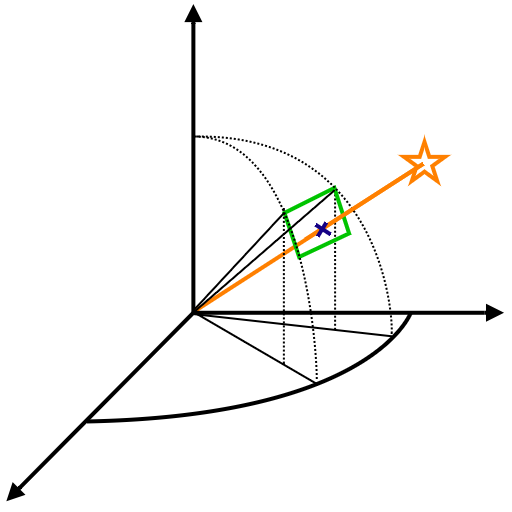
### 運作原理：

* 1. 星星位置的計算

這個程式使用圖(一)的方式記錄目前的視角座標。其中iVector 代表螢幕右邊的方向，預設是 y 軸的方向。jVector 代表螢幕下面的方向，預設是 x 軸的方向。pivotPoint 代表目前看的方向，預設是 z 軸的方向。



圖(一) 視角座標



圖(二) 2D座標對應方式

圖(二)是星星對應到2D座標的方式，詳細的計算步驟如下。

首先本程式對33的二為矩陣編號的方式如下

和數學編號的方式有一些不同

下面的矩陣代表2D畫面與3D座標之間的轉換關係：

其中 nVector 是 jVecotr 與 iVector 的外積。i, j, n 是 2D 畫面的座標，i 代表畫面往右的方向，j 代表畫面往下的方向，x, y, z 是3D空間中的座標。上面的這個矩陣是 2D 轉 3D的方式，在數學上的意義是以原點為中心進行旋轉。從 3D 空間座標轉換成 2D 畫面座標，需要經過反矩陣運算。

根據這個矩陣，先把星星 3D 座標對應到的 2D 畫面座標 n 的值算出來，如果 n 是負值或零，就代表星星的位置在視線方向的另外一個半球中，程式就不會顯示這個星星。這個分割半球的面會通過原點。

接著根據圖(二)所示，把直線與平面的交點算出來。直線使用參數式表示如下：

其中a, b, c 為星星在空間中的座標，將參數式帶入到平面方程式中：

其中n0, n1, n2 是法向量，帶入可以求出t，並進一步得到 x, y, z 的解。

3D 轉 2D 的轉換矩陣是以原點為中心的旋轉矩陣，也就是3D 座標的 (0, 0, 0) 轉出來的結果是 2D 座標的 (0, 0)，但是和預計的結果不同。結果應該是圖(二)這張圖中，綠色框框左上點的座標，旋轉成 2D 座標的結果是 (0, 0)。所以程式會把 3D空間中的這一點 (pivot point – 0.5 jVector – 0.5 iVector) 移到原點，再進行一次矩陣運算，所求的結果就是 2D 座標中的位置。

* 1. 座標軸的旋轉

程式會儲存旋轉位置矩陣 absRotationMatrix，以及 iVector, jVector一開始的初始值，iVector, jVecotr的初始值在程式的運作中永遠不會改變，座標軸旋轉時會改變的是旋轉位置矩陣。

當進行座標軸的旋轉時，程式會把旋轉的角度乘到旋轉位置矩陣中，之後再將旋轉矩陣乘以 iVector, jVector 的初始值，計算出 iVector, jVector旋轉之後的數值，再進行星星位置的計算。

其中以 x 軸為軸心的旋轉，是將旋轉位置矩陣，乘以以下的矩陣：

以 y 軸為軸心的旋轉，是乘以以下的矩陣：

以 z 軸為軸心的旋轉，是乘以以下的矩陣：

* 1. 滑鼠左鍵拖曳的計算

滑鼠左鍵的功能是控制視線看的方向，拖曳時坐標軸旋轉的方向會和目前的 jVector, iVector 有關係。

例如在 jVector 為 x 方向時，向右邊拖曳，座標軸就會以 x 為軸心進行旋轉，同理如果為 y 方向，就會以 y 為軸心旋轉。

iVector 也是類似的情況，如果 iVector 為 y 方向，往上面拖曳，座標軸就會以 y 為軸心旋轉。

所以在決定轉的方向時，會讓 jVector 與單位向量進行內積，算出 jVector 在 x, y, z 軸的投影量，再根據投影量決定 x, y, z 軸要旋轉的角度。同理 iVector 也是以相同的方式處理。

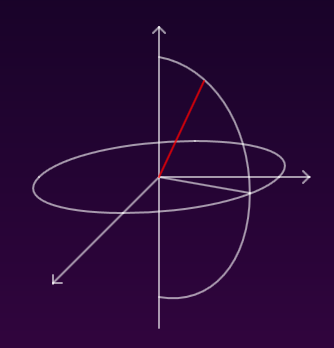
* 1. 滑鼠右鍵拖曳的計算

和上面類似，滑鼠右鍵的功能是，讓座標軸以視線方向為軸心旋轉。因此，程式將 pivot point 與單位向量進行內積，再以投影量決定各個軸的旋轉角度。

* 1. 視野放大縮小的計算

視野放大縮小是以改變 iVector, jVector 的長度達到功能的，當星星放大時，視野會縮小，假設放大倍率是兩倍，iVector, jVector 的值就同時除以2，就可以達到放大縮小的功能。

* 1. 地球儀的繪製



圖(三) 地球儀

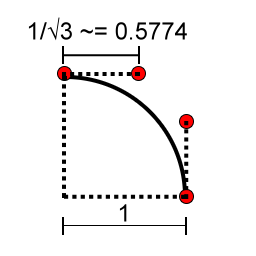
畫面左下方的地球儀，也是使用線性轉換的方式，將 3D 座標轉換成 2D 的方式顯示，其中單位向量的關係如下：

其中 , , 代表 3D 空間的座標，, 是 2D 平面的座標，i 在畫面上是向右為正，j向下為正。

寫成矩陣的方式如下：

使用這個矩陣可以計算出 3D座標的向量，在地球儀上面對應到的 2D 畫面的向量。

在畫面上出現的圓弧形是利用貝茲曲線，繪製多個 1/4 圓弧。圖(四) 是一個正圓形的 1/4 圓弧，用貝茲曲線繪製的畫法。



圖(四) 貝茲曲線 1/4 正圓形圓弧

在3D空間中這些圓弧是正圓形，而 3D 到 2D 的轉換是線性轉換，所以只要轉換圖(四)中，標出紅點的位置，再根據得到的結果繪製貝茲曲線，就會是正確的圓弧了。