

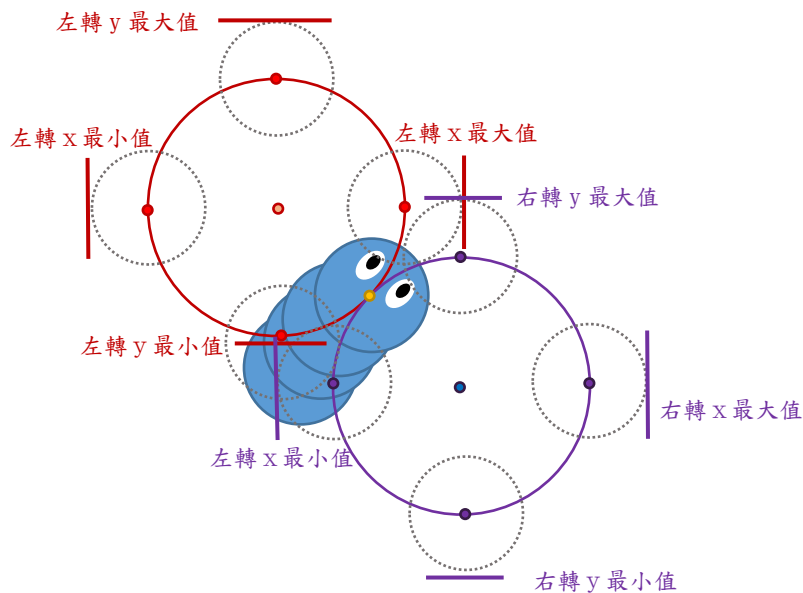
## 期末專題書面報告

因為期末專題是一個人寫，而且是最後一天趕，所以設計很簡陋QQ。AI 的設計大概分三個部分：一、避免撞到邊界，二、遇到其他隻蛇的處裡，三、吃點點，四、避免無限繞圈，以下分別說明：

### 一、避免撞到邊界：

避免撞到邊界的方法就是轉彎，重點在於何時轉彎？往哪個方向？因此我會先計算接下來往右轉360度的過程中會不會撞到邊界？如果會則左轉；不會的話則考慮接下來左轉360度會不會撞到邊界？如果會則右轉；不會則直走。計算如下：

經實測後，轉彎的半徑為113.59單位，程式碼已定義蛇的半徑為50，以此可以預測接下來轉彎的路徑：



我們可以透過頭的位置、前進方向和旋轉半徑先計算出在左轉或右轉路徑的圓心，接著計算出頭部走過的區域中，x和y的最大值和極小值。只要那個方向的最大值或最小值不在場地的範圍內，可判斷往該方向轉會碰到邊界。

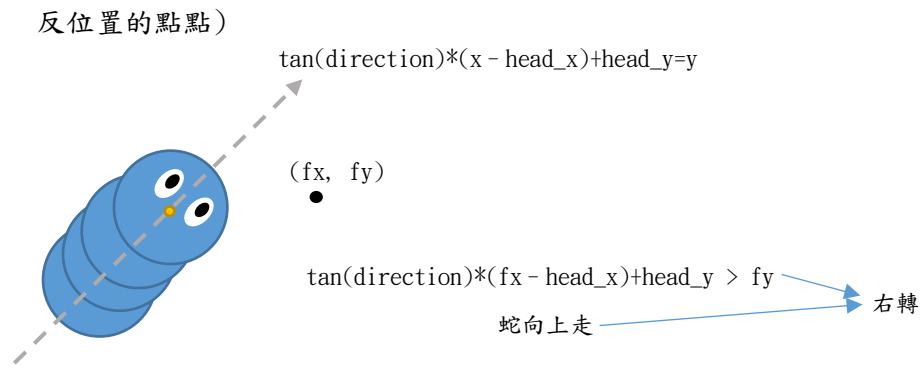
### 二、遇到其他隻蛇的處裡：

這部分比較簡單，如果離我最近的其他隻蛇的身體在我的500單位範圍以內，則不停的左轉，直到沒有蛇在我的500單位範圍內。但，如果是在邊界的話，就會依照上面的規則選擇旋轉的方向。

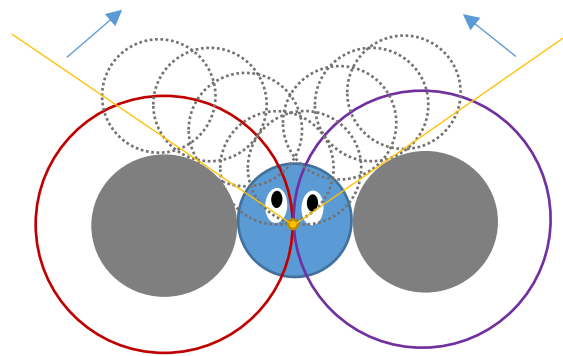
### 三、吃點點：

如果沒有快撞到邊界的危險且附近沒有其他隻蛇，蛇就會找點點吃。點點的位置全取會TLE，因此我只取100個位置，並鎖定最近的點點。鎖定最近的點點後要設計怎麼走？和時直走何時轉彎？

我們可以先沿前進方向畫直線，如果點點在線的右邊則右轉，左邊則左轉，在線上則直走(後面會提到蛇只會吃視線範圍內的點點，因此不考慮位在與前進方向相



然而如果點點在蛇吃不到的位置，就可能進行無限繞圈，因此要排除掉只往同一個方向轉彎會吃不到的點點：



如圖，因為旋轉半徑大於蛇的半徑，故出現灰色區域，即直接轉彎碰不到的區域，因此灰色區域的點點不能被鎖定為目標。為了編程上的方便，我將可鎖定的範圍內設定在兩條黃色線之間(角度約為  $2 * \arccos(\text{灰色區域半徑} / \text{旋轉半徑}) = 112$  度的可視範圍)。

#### 四、避免無限繞圈：

為了避免發生無限繞圈的情況，我建立一個計數器 counter，如果現在頭和身體第五節的距離與剛剛頭和身體第五節的距離一樣，則  $\text{counter} += 1$ ，如果是正在躲避其他蛇的狀態則 counter 歸 0，其他情況也是 counter 歸 0。如果  $\text{counter} > 1000$ ，意即連續 1000 步頭和第五節的相對距離一樣，有很大的機率是因為發生無限繞圈，因此在  $\text{counter} > 1000$  後，強制讓蛇走 300 步的 kForward，離開讓它發生無限繞圈的位置。