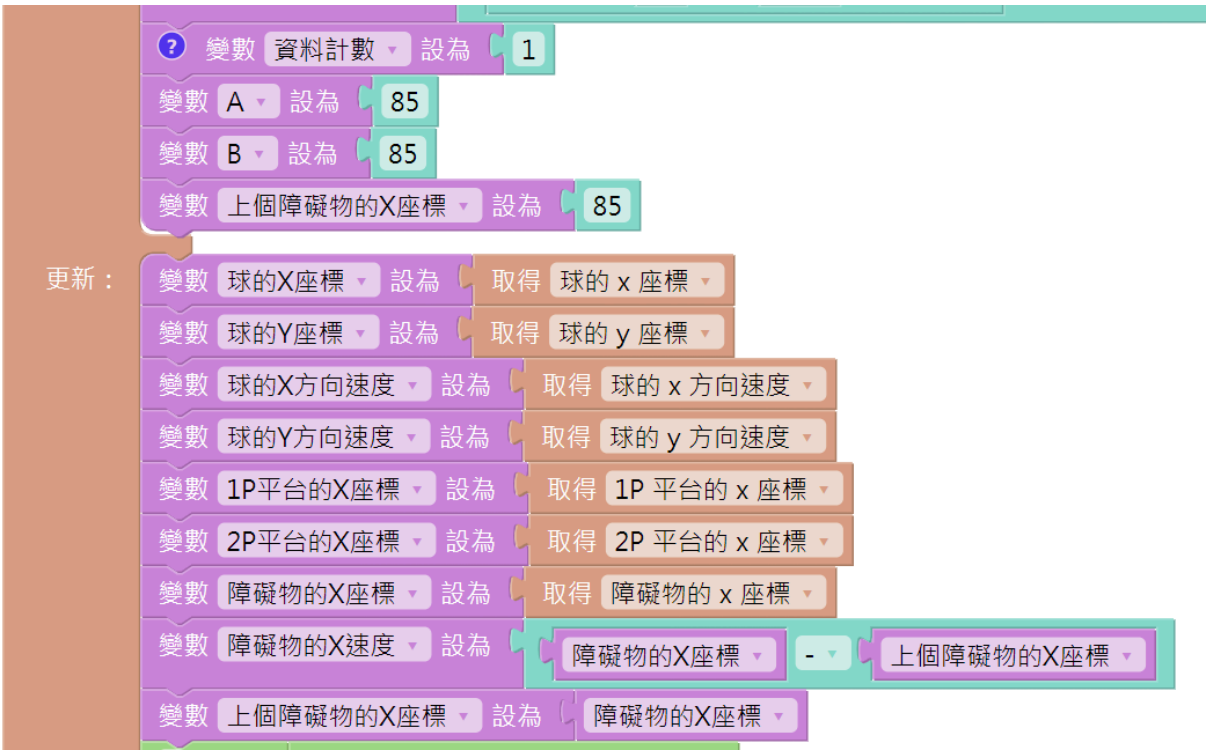
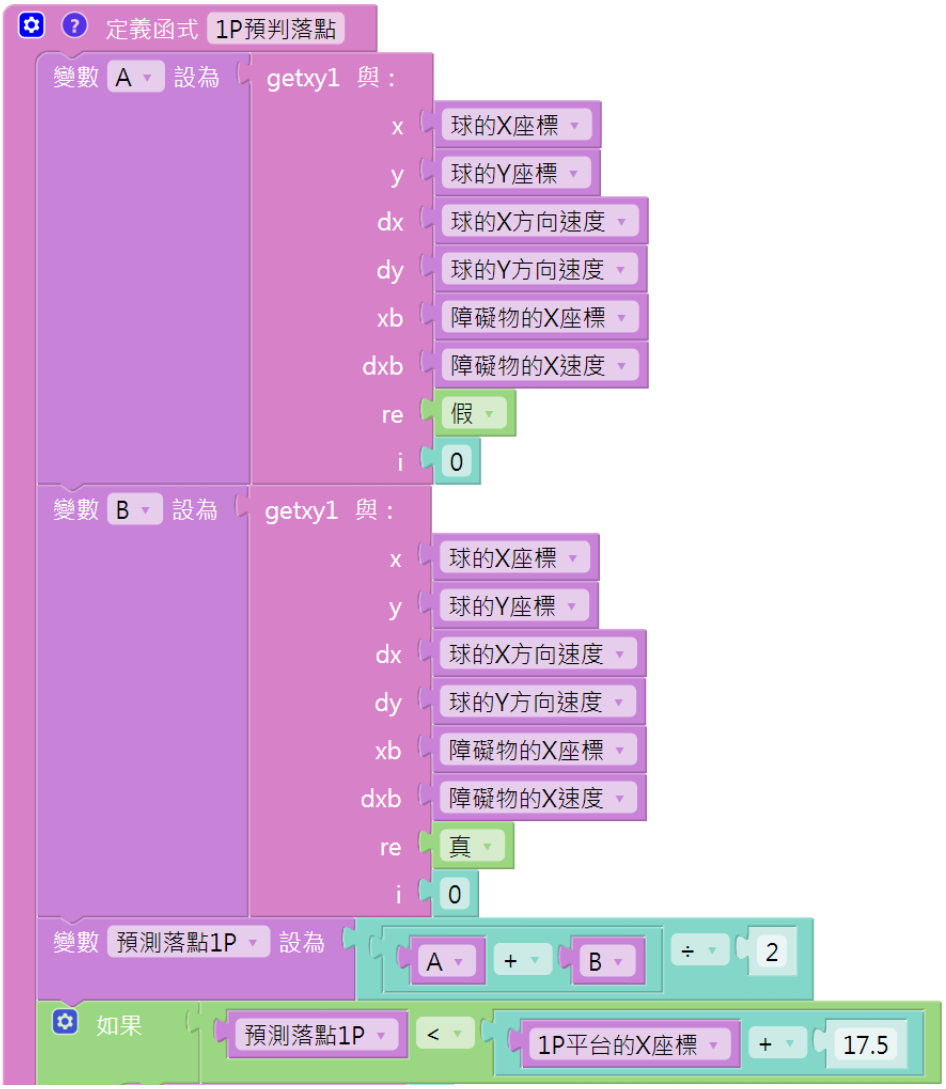
※ 1P auto.xml

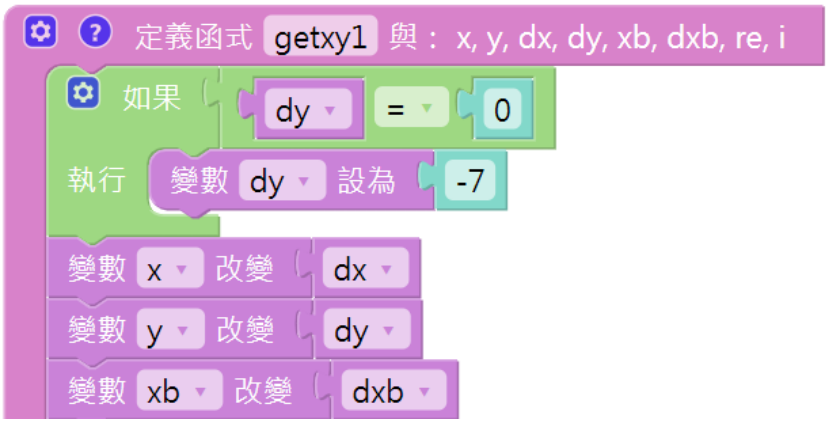
Step1.初始化



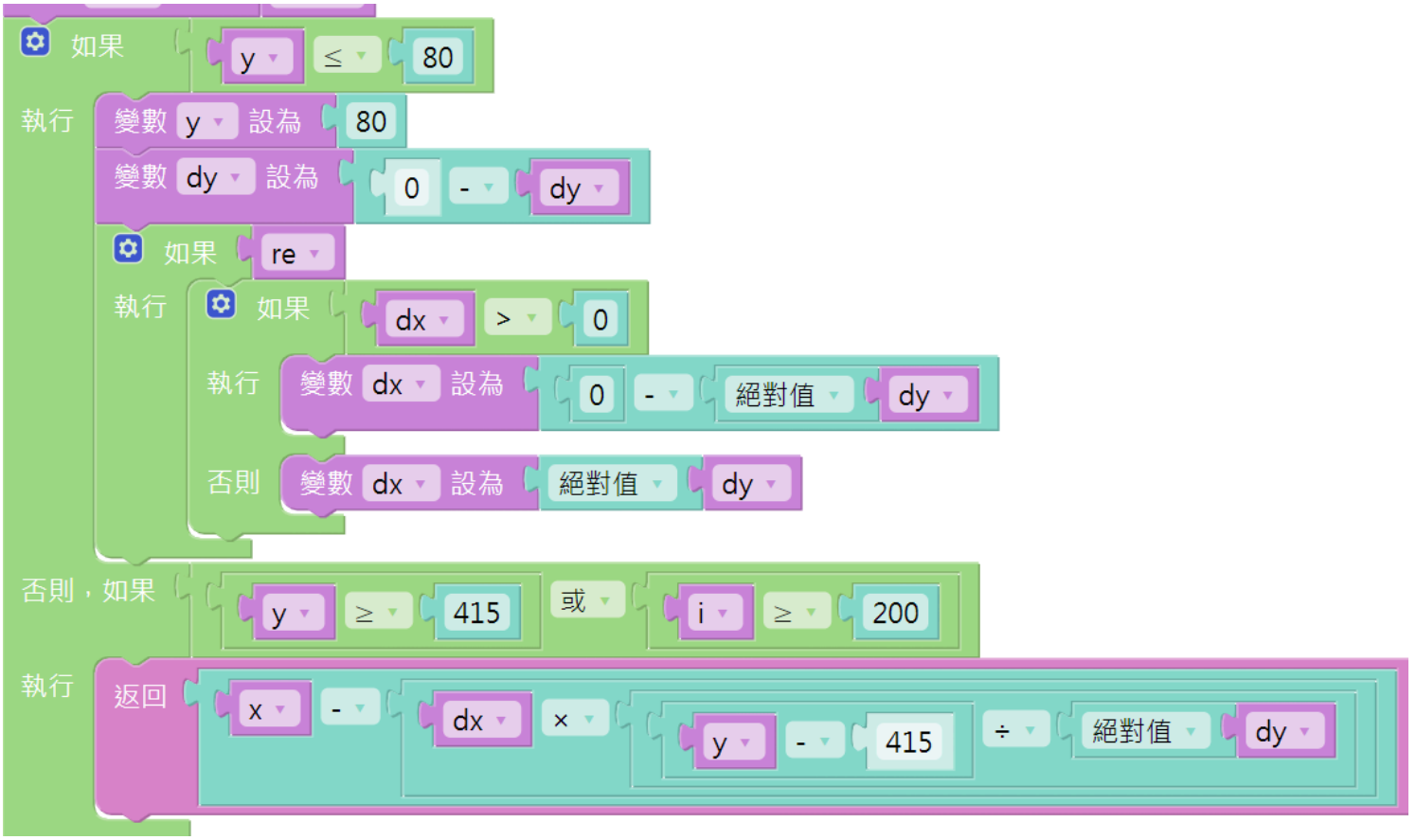
Step2.設定1p函式 及 AB點計算


定義函式 1P預判落點
變數 A 設為 getxy1 與：
x 球的X座標
y 球的Y座標
dx 球的X方向速度
dy 球的Y方向速度
xb 障礙物的X座標
dxb 障礙物的X速度
re 假
i 0
變數 B 設為 getxy1 與：
x 球的X座標
y 球的Y座標
dx 球的X方向速度
dy 球的Y方向速度
xb 障礙物的X座標
dxb 障礙物的X速度
re 真
i 0
變數 預測落點1P 設為 A + B ÷ 2
如果 預測落點1P < 1P平台的X座標 + 17.5

Step3. 變數每次迴圈都要前進，dy不能為 0

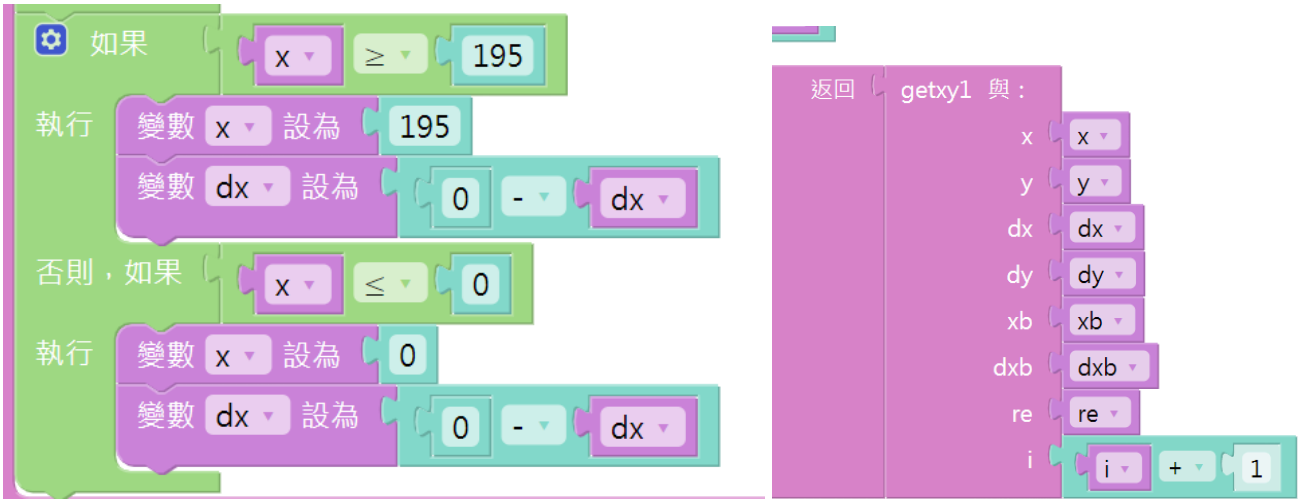


Step4.處理1P y 方向一般碰撞函數返回值，y大於 415就返回計算好的x碰撞點，
要考慮對手接球時是否有切球(re)



Step5.處理x 方向一般碰撞

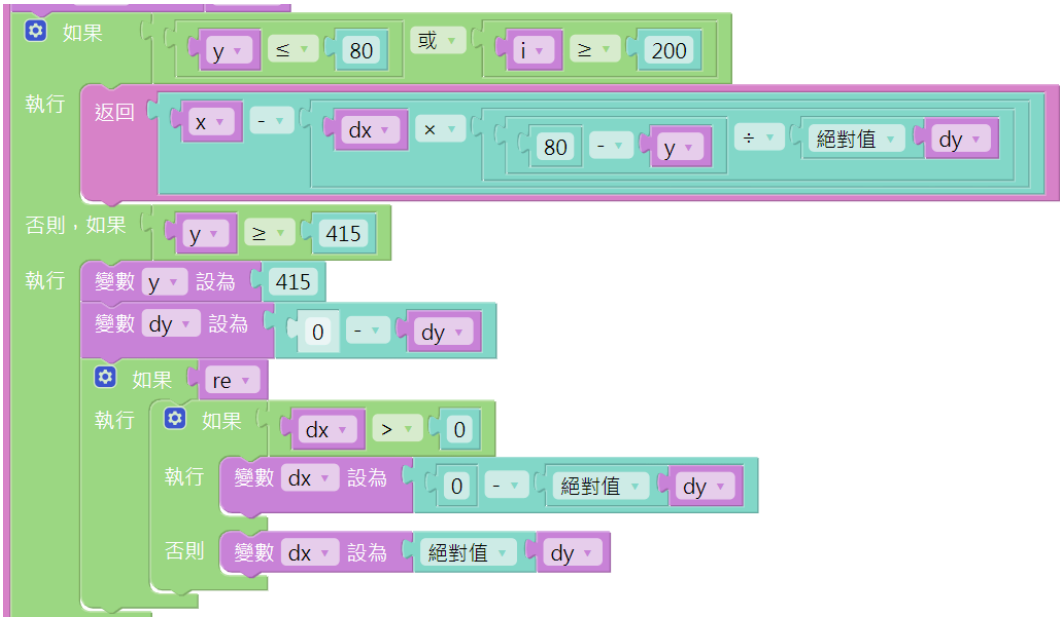
Step6.處理返回值



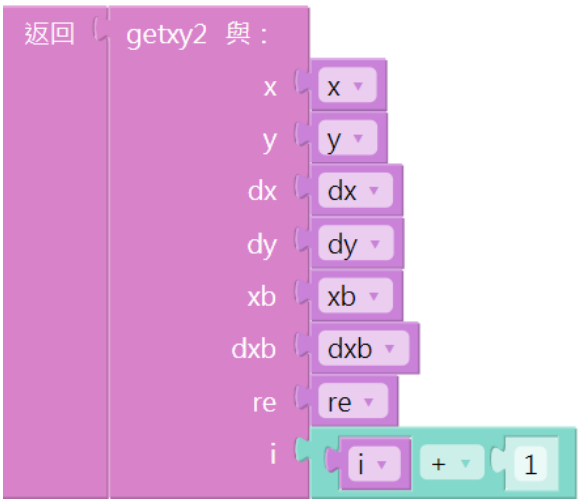
※ 2P

Step1 複製 getxy1 得到 getxy2，寫好 2P預判落點

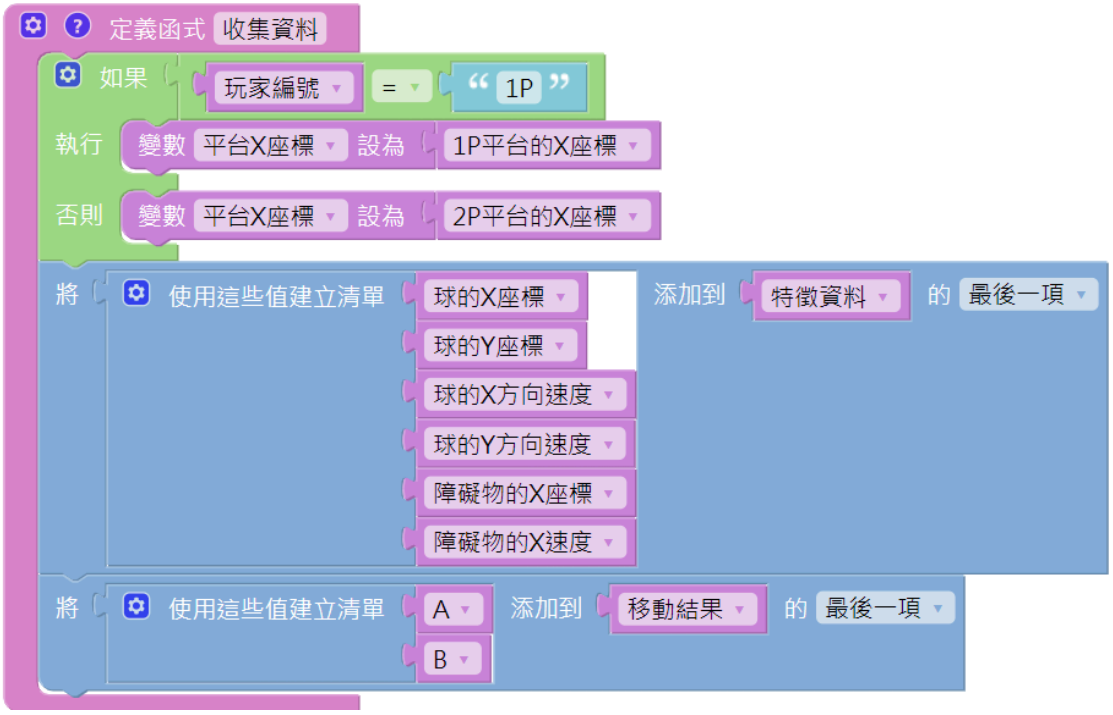
Step2.修改getxy2 邊界對調，415時反彈,80時返回，公式的部分也要調整好



Step3. x 方向一般碰撞和1P一樣，別忘了尾部遞迴的返回函數要用getxy2



※收集資料

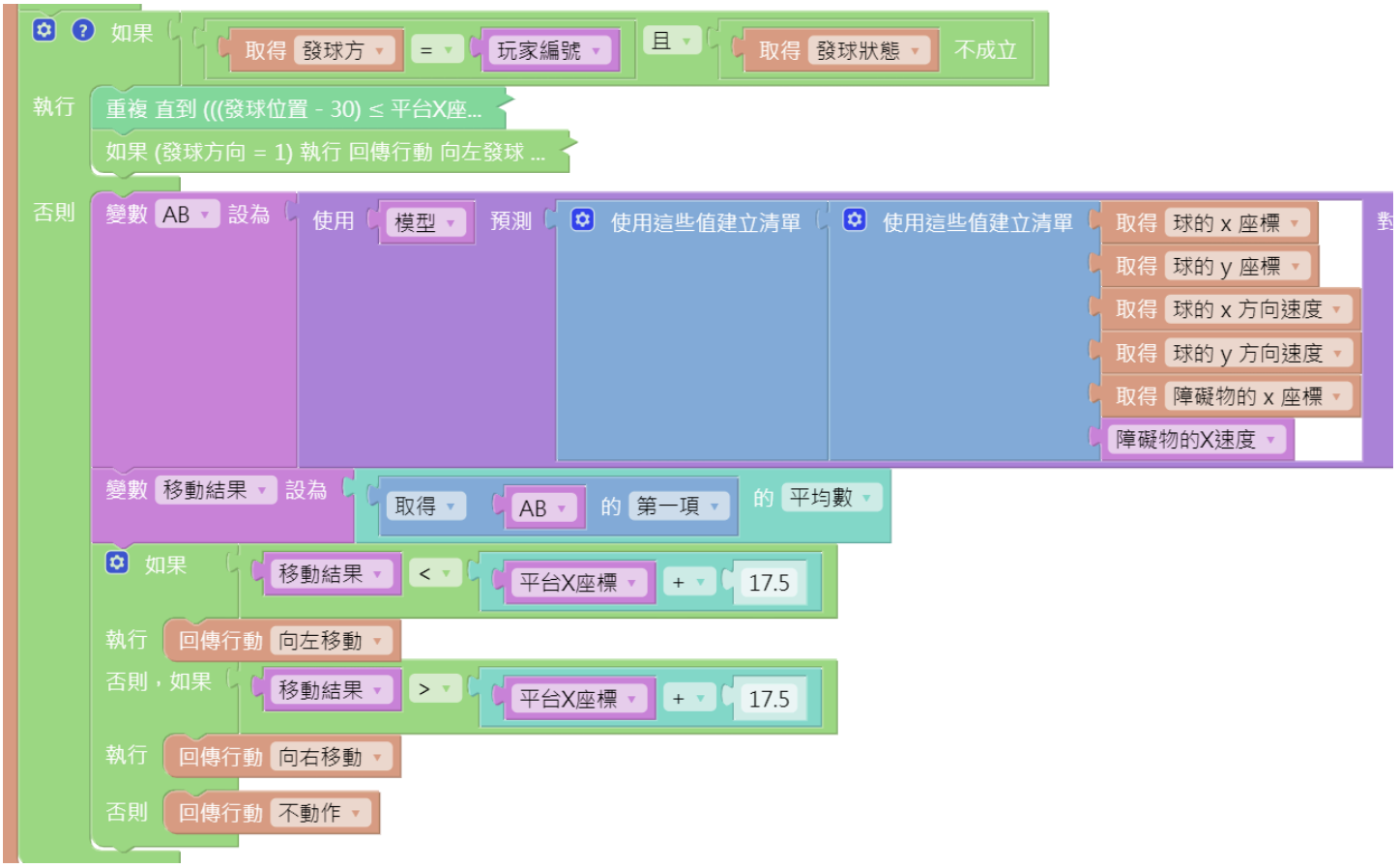


※訓練程式(截錄1P)

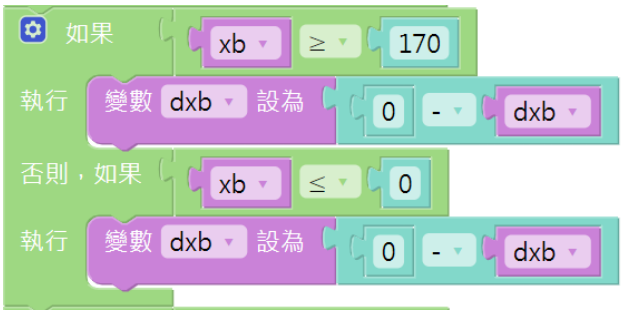
※ml_play.xml

Step1.處理障礙物x速度及模型設定

Step2. 使用模型預測得到AB點再取平均變成預測的落點(移動結果)，確認回傳行動(球往哪裡動)



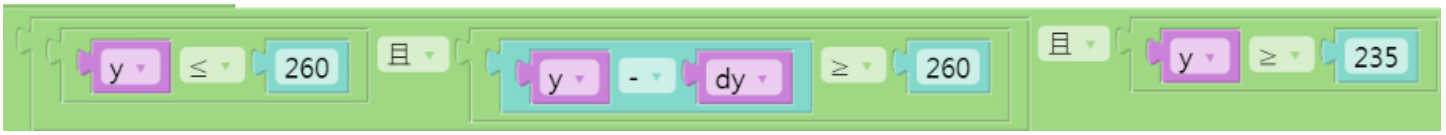
※障礙物碰左右壁



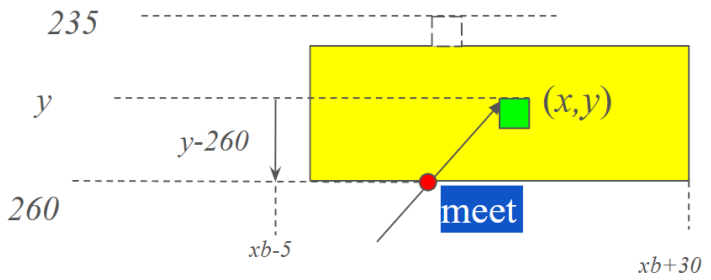
※1P加入障礙物判斷

Step1.處理往上碰障礙物 (1P加入障礙物判斷)

前提：dy<0,235≦y≦260 且y-dy≧235　　(y-dy即上個y)



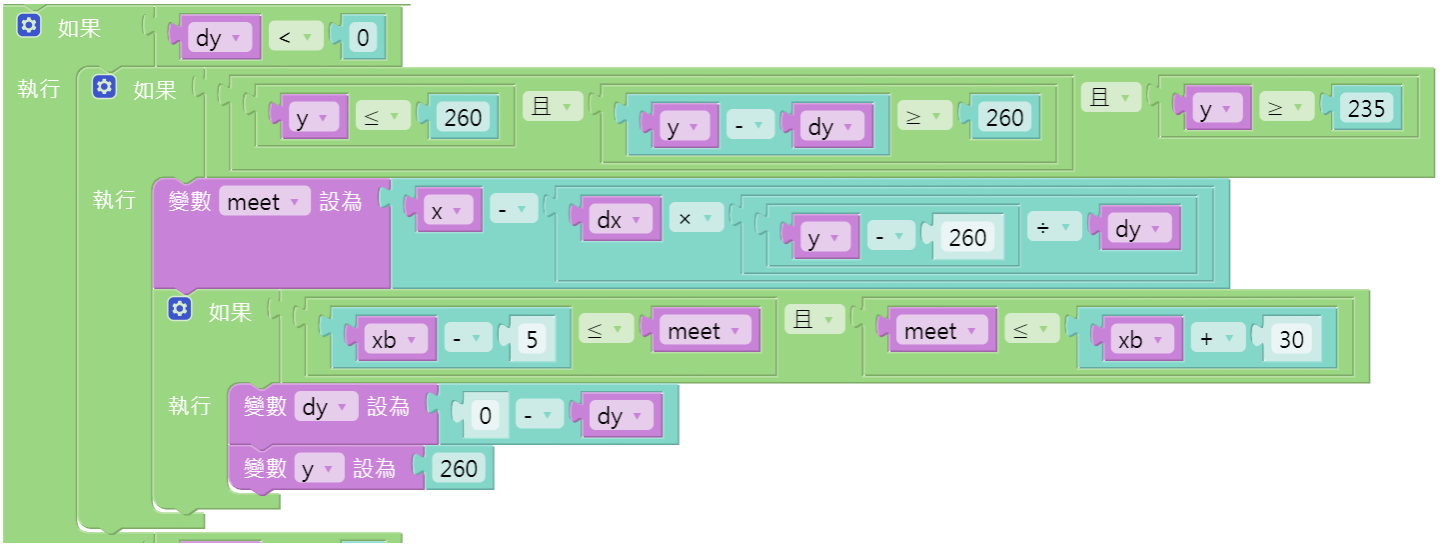
算出碰撞點 meet



$$meet = x - dx \frac{y-260}{dy}$$

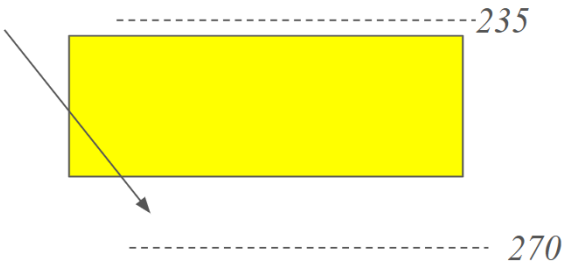
xb-5 ≦meet≦ xb+30　　(障礙物長度為30,xb:障礙物左邊界)

撞到了的話 y要向下推回 260 且y方向速度要反向

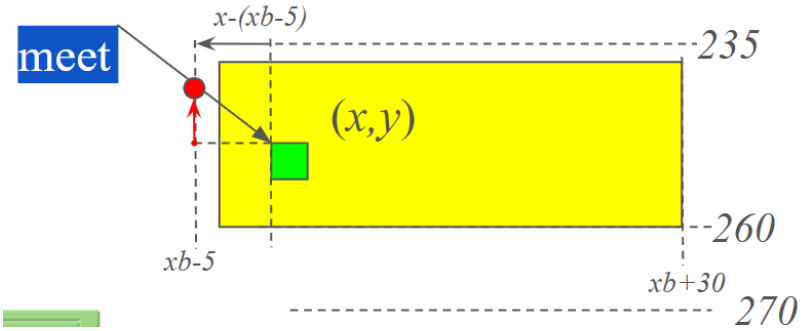


Step2.處理往右下碰障礙物 (1P加入障礙物判斷)

前提：dy>0,dx>0,260≦y≦270， x≧xb-5 且 x-dx ≦xb-5　(x-dx即上個x)



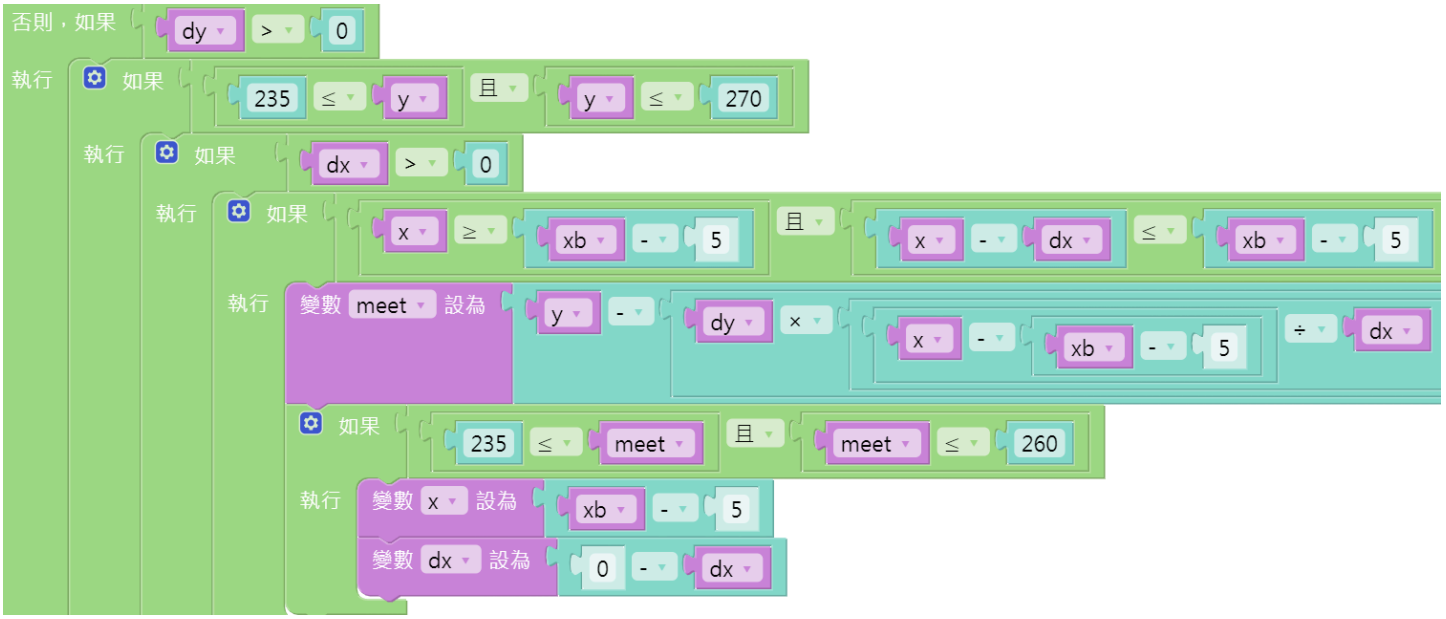
算出碰撞點 meet



$$meet = y - dy\,\frac{x-(xb-5)}{dx}$$

235 ≦meet≦ 260

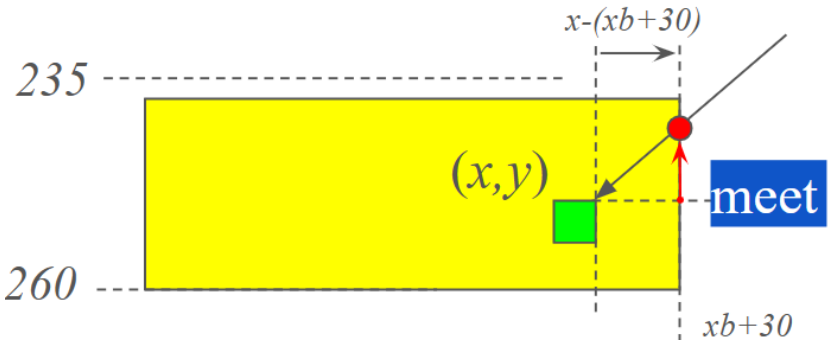
撞到了的話 x要往左推回 xb-5 且x方向速度要反向



Step3.處理往左下碰障礙物 (1P加入障礙物判斷)

前提：dy>0,dx<0 ,260≦y≦270， x≦xb+30 且 x-dx ≧xb+30　(x-dx即上個x)

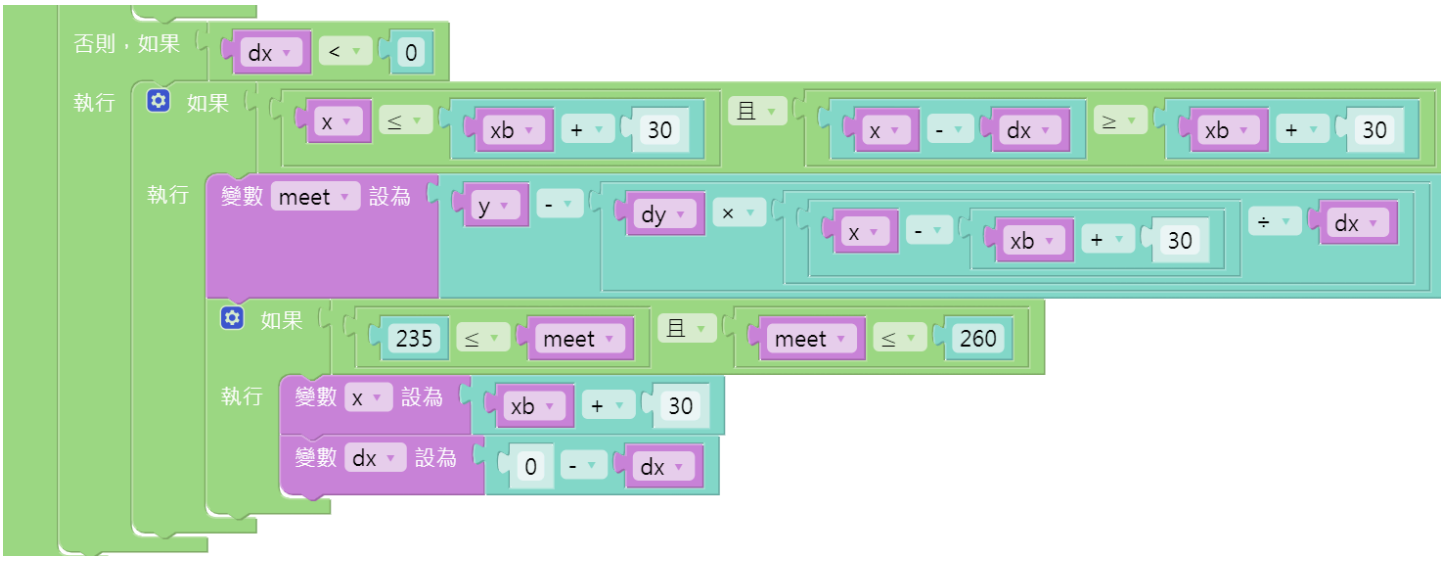
算出碰撞點 meet



$$meet = y - dy\frac{x-(xb+30)}{dx}$$

235 ≦meet≦ 260

撞到了的話 x要往左推回 xb-5 且x方向速度要反向

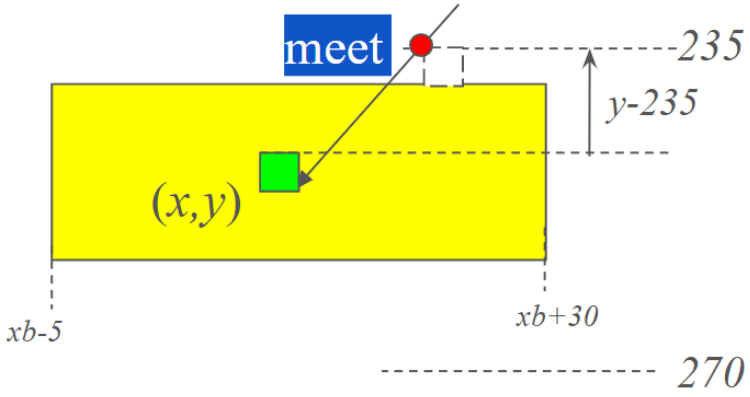


※2P加入障礙物判斷

Step1.處理往下碰障礙物 (2P加入障礙物判斷)

前提：dy>0 , 235≦y≦270 且 y-dy≦235　　(y-dy即上個y)

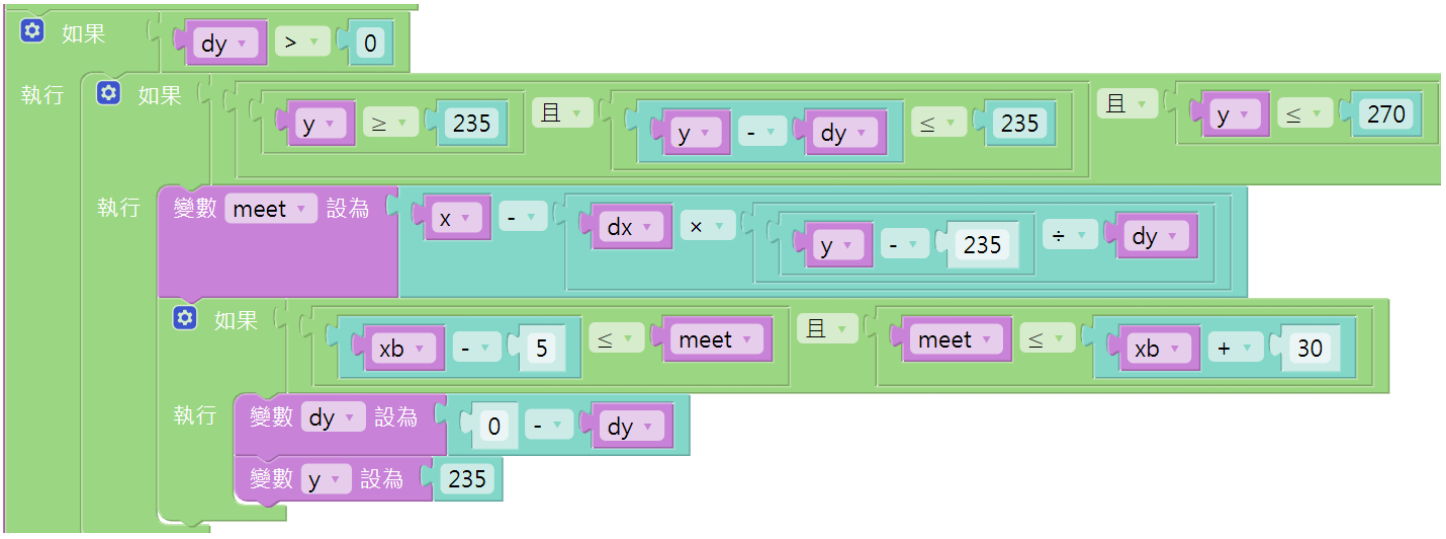
算出碰撞點 meet



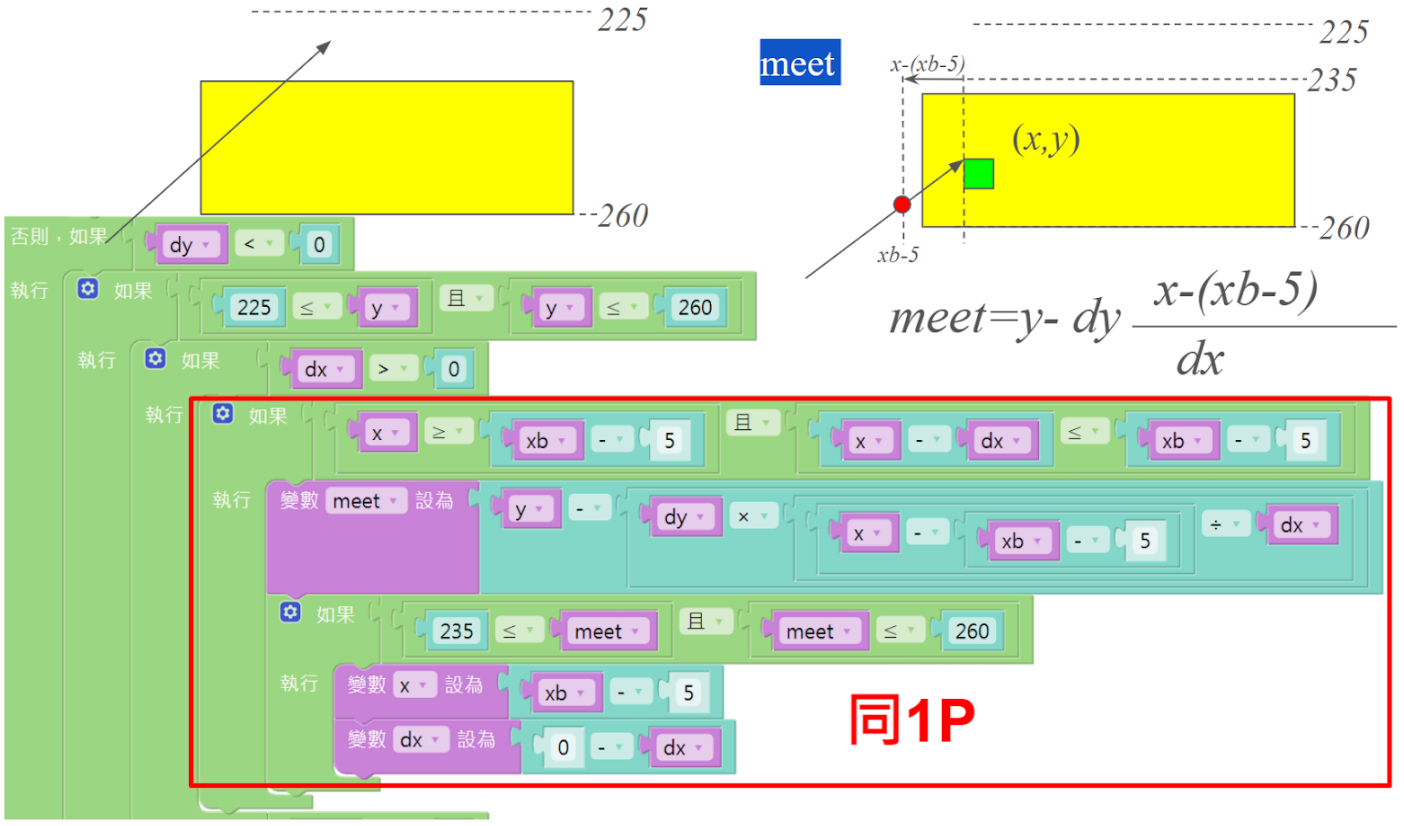
$$meet = x - dx\,\frac{y-235}{dy}$$

xb-5 ≦meet≦ xb+30　　(障礙物長度為30,xb:障礙物左邊界)

撞到了的話 y要向上推回 235 且y方向速度要反向



Step2.處理往右上碰障礙物 (2P加入障礙物判斷)



Step3.處理往左上碰障礙物 (2P加入障礙物判斷)