**helloword – Readme**

* **程式架構**

**MainActivity.java**

**MainFragment.java**

**ButtonDraw.java**

**TimeLine.java**

CircularSeekBar

* **元件內容**

**MainActivity.java**

中繼站的角色，連接各個元件然後負責幫忙傳遞訊息到正確的元件上面。

**功能:** 切換自由移動、切換3D環場

**Functions :**

For切換3D環場

// 傳輸所需資料給ThreeDMainActivity.java

public void RunLineInfo(Vector<RunBag> in\_RunLine)

public void P1Info(Player in\_P1)

public void P2Info(Player in\_P2)

public void P3Info(Player in\_P3)

public void P4Info(Player in\_P4)

public void P5Info(Player in\_P5)

public void BInfo(Player in\_B)

For設定SeekBar

// 將seekbar上的起始時間傳到MainFragment，等錄製完路徑後存進runbag

public void seekBarStartTime(int startTime){ seekBarCallbackStartTime = startTime;

mainfrag.pass\_start\_time(startTime);

}

// 將seekbar上的持續時間傳到MainFragment，等錄製完路徑後存進runbag

public void seekBarDuration(int duration){

seekBarCallbackDuration = duration;

mainfrag.pass\_duration(duration);

}

// 將seekbar的id傳到MainFragment，等錄製完路徑後跟TimeLine layout裡的元件綁在一起

public void seekBarId(int id){

seekBarCallbackId = id;

mainfrag.pass\_id(id);

}

For設定錄製狀況

// 因為record按鍵是裝在ButtonDraw.java裡面，所以目前錄製與否要從ButtonDraw傳到MainFragment裡面做使用

public void setRecordCheck(boolean in\_recordcheck)

For清除筆跡

public void setClean(){

mainfrag.clear\_paint();

}

**MainFragment.java**

程式主體，主要負責球員路徑錄製功能與創造TimeLine＋mySeekBar元件。

**功能:** 錄製路徑、載入/儲存戰術。

**Functions :**

// assign layout的元件給程式裡的變數

public void onActivityCreated(Bundle savedInstanceState) {

….

ViewTreeObserver vto2 = circle.getViewTreeObserver();

vto2.addOnGlobalLayoutListener(new OnGlobalLayoutListener() {……

這裡面的東西是先初始化一些變數，跟先使用他們一次，不然程式在執行的時候會有一點問題。(例如: 第一次畫路徑的時候會畫不出來，畫第二次才畫出來，所以才要在這邊先畫一次，我也不知道這個bug是為何)

….}

}

// socket的thread在這裡開

public void onStart()

// 設定最大時間

public void Mainfrag\_Set\_Total\_time(int inputTime)

// 設定IP跟port

public void Mainfrag\_Set\_UDP\_IP(InetAddress IP,int port)

// 傳送資料給UE4

public void Mainfrag\_SendtoUE4()

// 把跑動資料(packet)擺進buffer裡面

private ByteBuffer putPacketIntoSocket(ByteBuffer bBuffer)

// mainfrag 的 callbackinterface在這裡創

public void onAttach(Activity activity)

// 設定player的ImageView為無持球狀態

public void set\_player\_to\_no\_ball()

// 設定正在播放

public void set\_playing(int input)

// 傳送球員資料給3D環場使用的function

public void pass\_RunLine\_Player\_Info()

// 設定seekBar的起始時間，用於存進runbag

public void pass\_start\_time(int input)

// 設定seekBar的持續時間，用於存進runbag

public void pass\_duration(int input)

// 設定seekBar的id，用於綁定TimeLine裡的layout

public void pass\_id(int input)

// 檢查是否錄製中

public void pass\_recordcheck(boolean input)

// call from timeline的function ，方便設定runbag

public void set\_seekBar\_to\_RunBag(Vector <Integer> input)

// 設定progresslow，就是start time

public void setMainFragProLow(int Low\_in)

// 取得progresslow(start time)

public int getMainFragProLow()

// 清除筆跡

public void clear\_paint()

// 清除錄製內容(清除路徑)

public void clear\_record()

// 計算畫圖曲線

public int lagrange(Vector<Integer> px,Vector<Integer> py, float x)

// 移動全部球員到initial position

public void Player\_move\_to\_start\_position()

// 設定全部球員為無持球狀態

public void Player\_change\_to\_no\_ball()

// 推薦戰術

public void Do\_Recommend()

// 載入預設戰術

private Vector <String> load\_Default\_Strategies()

// 推薦戰術的按鈕

private OnClickListener recommend\_button\_Listener

// 做DTW的函數

public double DTW(Vector<Integer> road\_a,Vector<Integer> road\_b)

// 算出並顯示EDT的推薦threshold(少於多少threshold才會被推薦)

public double EDR\_threshold(int road1\_size,int road2\_size)

// 算EDR的函數

public double EDR(Vector<Integer> road\_a,Vector<Integer> road\_b)

// 把input的路徑把零去掉之後，做normalize

public Vector<myPoint> process\_simi\_road(Vector<Integer> input\_road)

// 做EDR的函數，如果要做推薦要呼叫這個，這個會先處理input road後再餵給EDR()去做出結果

public double Do\_EDR(Vector<Integer> road\_a,Vector<Integer> road\_b)

// 做EDR的第二種方法(recursive)

public double EDR2(Vector<myPoint> road\_a,Vector<myPoint> road\_b)

// 計算點跟點之間的距離 (int type)

public double Dist(int xi,int yi,int xj,int yj)

// 計算點跟點之間的距離 (double type)

public double Dist(double xi,double yi,double xj,double yj)

// 做normalize

public Vector<Double> normalization(Vector<Integer> road)

For 儲存、載入戰術(按鈕的判斷在ButtonDraw.java裡面)

// 儲存戰術

public void save\_dialog()(開啟選取檔案)

// 將Data存入file

public void writeToFile(File fout, String data)

// 載入戰術 (開啟選取檔案)

public void load\_dialog()

// 讀取文字檔

private String readFromFile(File fin)

// 將文字檔資料轉換成程式所需的資訊

private String readFromFile(File fin)

// 依據input值來設定球員的轉向，轉哪一名球員是看rotate\_which\_player這個變數，這個變數的值會依照你點到哪一名球員就變成相對應的值。P1 = 1，P2 = 2，…B= 6，D1=7，D2=8…..以此類推。

public void rotatePlayer(int input)

// 開 thread 來倒數時間，並每一秒呼叫一次checkRunLine

public void playButton()

// 開thread來搜尋 RunLine裡面有沒有RunBag的start time 等於目前時間，有的話就呼叫相應的player執行play

protected void checkRunLine(final int in\_time, final int in\_RunLineSize)

// 控制球員移動的thread，讓球員跟著runbag裡存的road 座標移動

protected void play(final int speed, final Handler play\_handler, final int in\_k, final int in\_j)

// Bitmap\_ontouch，接收球場內的點擊事件，之前是用來做如果點擊了球員路徑，就會產生rmbutton在路徑旁邊，然後就可以點擊rmbutton來刪除該路徑，但因為偵測路徑的點擊事件不太好用，所以現在是註解掉的狀態。

private OnTouchListener bitmap\_ontouch = new OnTouchListener()

// 控制所有球員的觸控事件

private OnTouchListener player\_Listener = new OnTouchListener()

// 刪除button的Listener ，刪除路徑用的，目前是註解掉的

private OnClickListener rm\_button\_Listener = new OnClickListener()

// 刪除一條路徑(根據ID) (此按鈕會產生在TimeLine layout上面)

public void Mainfrag\_remove\_one\_path(int seekbarId)

// 排序目前的路徑數字，刪除路徑後使用，因為每條路徑會有自己的一個順序的數字，所以如果刪除某一條路徑，其他路徑都要重新算過自己的順序數字。

public void Mainfrag\_sort\_pathnum()