Project 5 Dance Game 技術文件

(一)、 專案

使用 Github 專案:

https://github.com/digital-standard/ThreeDPoseUnityBarracuda

使用 unity 的 Berracuda 之 ResNet 吃 RGB 影像來分析出 3D 動作。這個專案可以實時辨識影片或相機照到的畫面然後讓角色作出動作。

(二)、 比對動作

NPC 做出動作後，紀錄每一個 frame 的關節旋轉，在玩家做動作的時候，紀錄玩家每個 Frame 的動作，然後去比對。

```
for (int i = 0; i < player.boneFrames.Count; i++)
{
    total = 0;
    // 找最接近
    foreach (KeyValuePair<int, Transform> pair in npc.boneFrames[0])
    {
        Transform npcTransform = pair.Value;
        Transform playerTransform = player.boneFrames[i][pair.Key];
        float distance = ComputeRotationDistance(playerTransform, npcTransform);
        total += distance;
    }
    if (total < min)
    {
        min = total;
        index = i;
    }
}
```

玩家可能做了一連串的動作，所以先對玩家的動作中找出最接近 NPC 動作的 frame，從那個 frame 開始去對接下來的每個 frame 比對。

```
total = 0;
for (int j = 0; j < npc.boneFrames.Count; j++)
{
    if (j + index >= player.boneFrames.Count)
        return 0;
    float total2 = 0;
    foreach (KeyValuePair<int, Transform> pair in npc.boneFrames[j])
    {
        Transform npcTransform = pair.Value;
        Transform playerTransform = player.boneFrames[j + index][pair.Key];
        float distance = ComputeRotationDistance(playerTransform, npcTransform);
        float score = 0;
        if (distance < 40)
            score = 1;
        else if (distance < 20)
            score = 2;
        else if (distance < 15)
            score = 3;
        else if (distance < 10)
            score = 4;
        total2 += score;
    }
    total += total2;
}
Debug.Log("total: " + total);
return total;
```

因為逐個 frame 比對很可能因為玩家做的動作速度快慢而造成沒辦法比對成功，所以這邊對 distance 放寬了條件，只要有部分比對成功就算過。分數的部分是單純依照動作相似度給予 1~4 分不等，當成給玩家的回饋。

(三)、 判斷玩家不在範圍內

判斷部分主要是透過抓取 Ground Truth 的人體部位比例來得知。

```
List<float> partRatio = GetPartLenRatio(Skeletons);
float partLenErrorSum = 0;
for (int i = 0; i < partRatio.Count; i++)
{
    partLenErrorSum += Mathf.Abs(GTRatio[i] - partRatio[i]);
}
float errorAverage = GetRecentAverageErrorRatio(partLenErrorSum);
IsError = partLenErrorSum > 9f;
IsErrorAverage = errorAverage > 9f;
if (IsError)
    Debug.Log("Part len ratio error:" + errorAverage);
```

當玩家離開相機範圍內時，因 Heatmap 少了 Confidence 較高的點(身體部位)，導致剩下的 Pixel confidence 都頗低，以至於辨識出來的人體骨架非常詭異，而骨架詭異的同時，各個身體部位的比例也會變得很詭異，因此演算法就簡單地透過人體比例來判定。

```
private List<float> GetPartLenRatio(List<Skeleton> skeletons)
{
    List<float> ratio = new List<float>();
    for (int i = 0; i < skeletons.Count; i++)
    {
        ratio.Add(
            Vector3.Distance(skeletons[i].start.Pos3D, skeletons[i].end.Pos3D) /
            Vector3.Distance(skeletons[(int)SkelotonName.Spine].start.Pos3D, skeletons[(int)SkelotonName.Spine].end.Pos3D)
            );
    }
    return ratio;
}
```