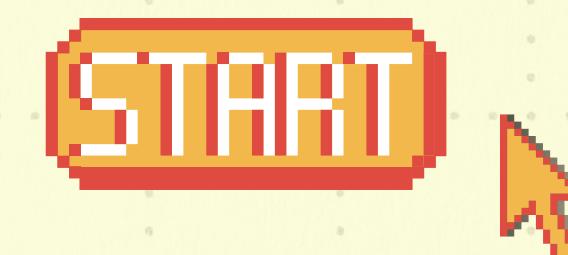




# Solo\_Training

第四組

110401525 劉璨瑤  
109303574 楊哲睿  
109303041 翁偉綸



109303518 孫颯馬  
109303543 郭又維  
110403503 謝蒿安





# 目錄

一、動機與目的 ······	3	四、組員分工 ······	11
二、產品介紹 ······	4	五、時程規劃 ······	12
三、成果演示 ······	9	六、預期結果 ······	13
四、使用技術 ······	10		

飲食習慣及作息改變

養成運動習慣

肥胖人數增加

維持健康的體態

青壯年運動量不足

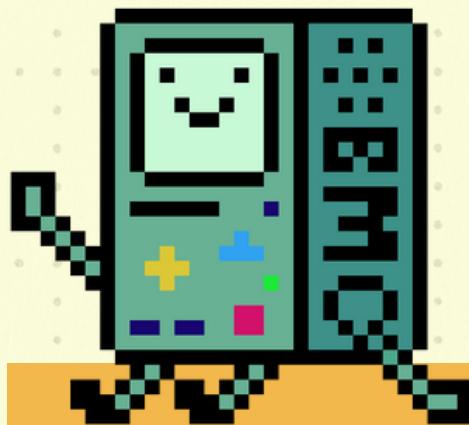
提高人們運動的意願

動機  
與  
目的





# 產品介紹



# 產品概念介紹



目標族群：15-60歲

三大功能：

1. 人物角色
2. 健身運動
3. 遊戲任務

主要內容：

以健身結合遊戲的方式做出一個電腦程式，程式內有四種健身運動，並搭配小遊戲的獎勵制度，最後記錄使用者每天健身的動作與次數。



## 運算各部位角度

- 在偵測四種運動動作時，會使用到手肘、臀部以及肩膀三部位之間的夾角。
- 在各個部位都使用三個點來計算夾角。

```
float calculateAngle(List<Vector3> firstPoint, List<Vector3> secondPoint, List<Vector3> thirdPoint)
{
    //1~2點的向量
    Vector2 firstToSecond = secondPoint[0] - firstPoint[0];
    //2到3點的向量
    Vector2 secondToThird = thirdPoint[0] - secondPoint[0];
    float magnitudeFirstToSecond = firstToSecond.magnitude;
    float magnitudeSecondToThird = secondToThird.magnitude;
    // 計算兩個向量的點積
    float dotProduct = Vector2.Dot(firstToSecond, secondToThird);
    // 計算餘弦值
    float cosTheta = dotProduct / (magnitudeFirstToSecond * magnitudeSecondToThird);
    // 計算夾角 (弧度)
    float angleInRadians = Mathf.Acos(cosTheta);
    // 將弧度轉換為角度
    float angleInDegrees = angleInRadians * Mathf.Rad2Deg;
    return angleInDegrees;
}
```

# TRAIN：影像辨識動作#程式碼原理

## 計算動作完成程度

- 使用字典將每個動作的完整偵測角度分為五等分，若有經歷這個角度則+1，上升與下降狀態會分開增加。
- 若有一個角度區間沒有經歷到則回傳值，可用來判斷動作的完成度。

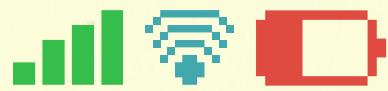
```
Preference
void UpdateStatusPushup(float angle)
{
    switch (angle)
    {
        case float a when a > 0 && a <= 20:
            statuspushup[1] = statuspushup[1] < 2 ? statuspushup[1] + 1 : statuspushup[1];
            break;
        case float a when a > 20 && a <= 40:
            statuspushup[2] = statuspushup[2] < 2 ? statuspushup[2] + 1 : statuspushup[2];
            break;
        case float a when a > 40 && a <= 60:
            statuspushup[3] = statuspushup[3] < 2 ? statuspushup[3] + 1 : statuspushup[3];
            break;
        case float a when a > 60 && a <= 80:
            statuspushup[4] = statuspushup[4] < 2 ? statuspushup[4] + 1 : statuspushup[4];
            break;
        case float a when a > 80 && a <= 100:
            statuspushup[5] = statuspushup[5] < 2 ? statuspushup[5] + 1 : statuspushup[5];
            break;
    }
}

for (int i = 1; i < statuspushup.Length; i++)
{
    if (statuspushup[i] != 2)
    {
        PushUpFinish = false;
        Debug.Log("Index " + i + " has value " + statuspushup[i] + " which is not 2.");
        break;
    }
}
```

## 整體完成效果

- 已成功使用C#完成動作辨識並可以準確偵測使用者骨架，預測骨架方面則取決於偵測到的資訊量。
- 成功分別計算三種動作的次數以及棒式的秒數。
- 可根據前面提到的字典來判斷四種動作的完成程度以及未完成原因。

```
else
{
    Debug.Log("手肘角度: "+PushupAngle +" 肩膀角度: "+PlateAngle+" 臀部角度: "+SquatAngle);
    if((PushupAngle>=90)&& (PushupAngle <=120))&&((SquatAngle>=160)&& (SquatAngle <=180))
    {
        timer_f += Time.deltaTime;
        timer_i = (int)timer_f;
    }
    else
    {
        Debug.Log("動作錯誤請矯正");
    }
    Debug.Log("時間: "+ timer_i); //印出持續時間
}
```

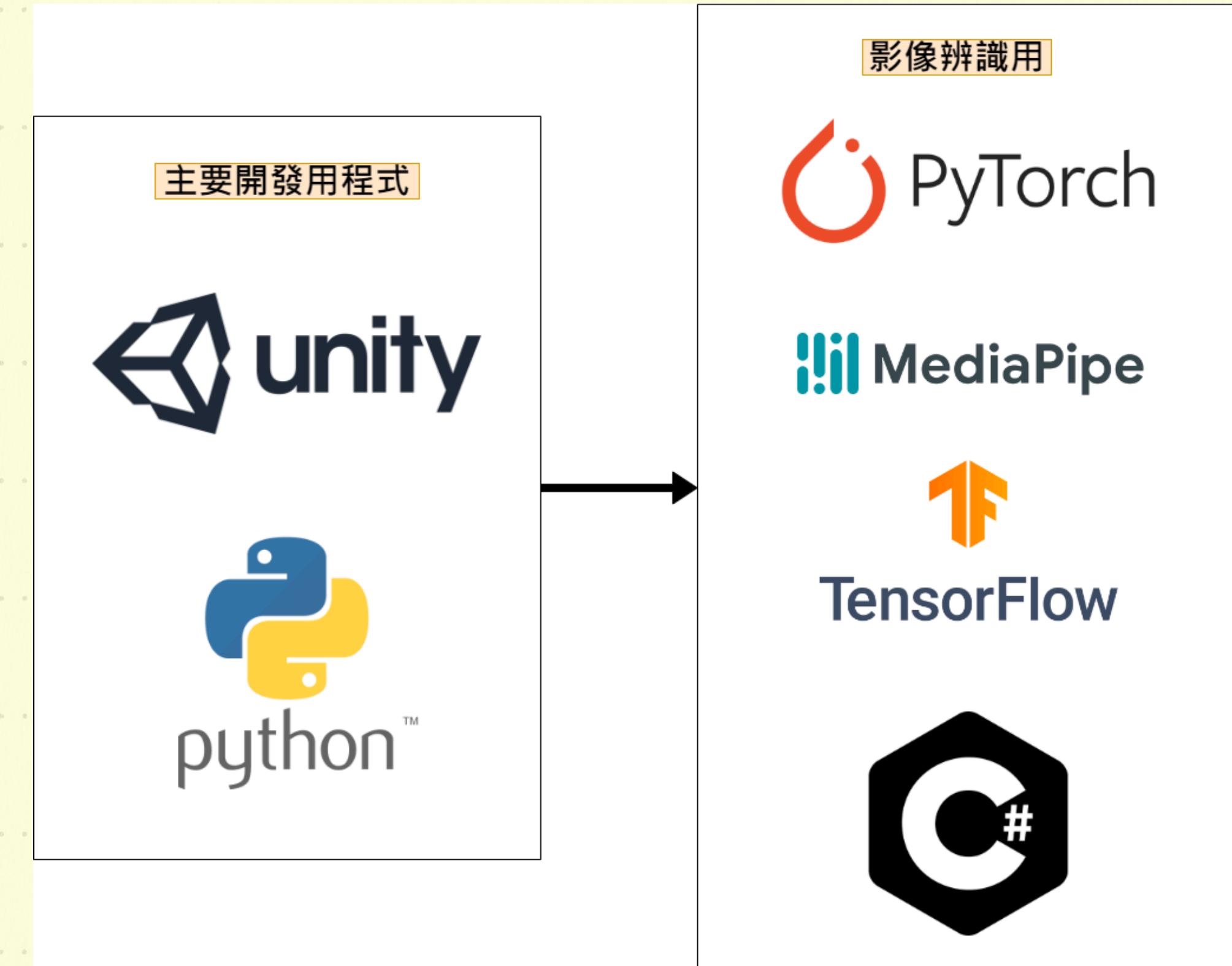


# Solo TRAINING

## 成果演示

# 使用技術

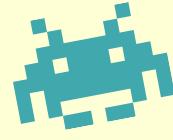
- C#
- Mediapipe
- Unity
- Tensorflow
- pytorch
- Python



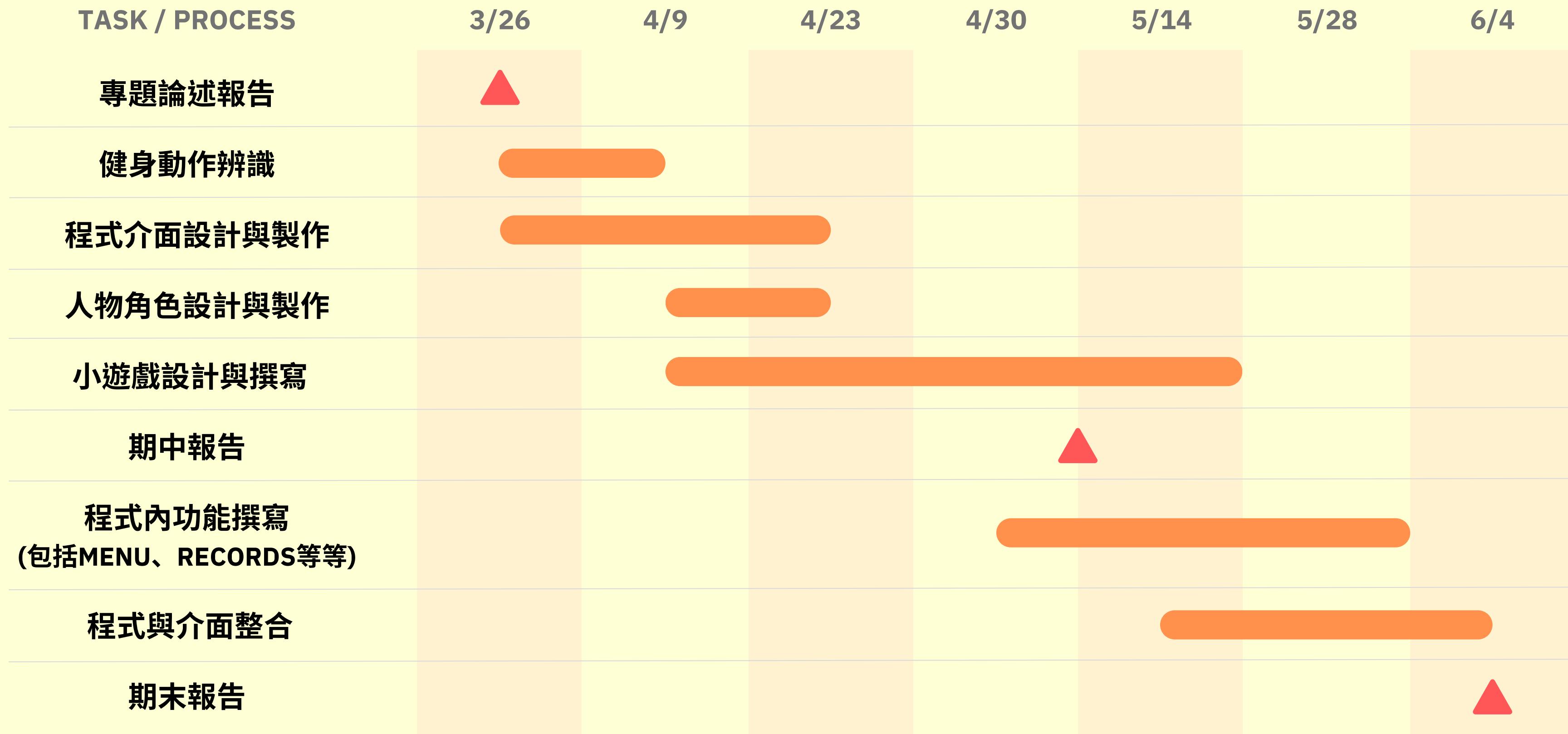
# 組員分工



工作內容	參與人員
程式介面設計	孫颯馬、郭又維、劉璨瑤、謝蒿安
影像辨識程式撰寫	楊哲睿
人物角色製作、遊戲地圖	翁偉綸
遊戲程式撰寫	孫颯馬、郭又維
簡報製作	劉璨瑤、謝蒿安



# 時程規劃

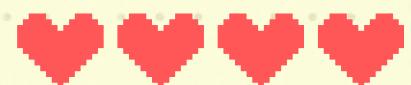


# 預期 結果

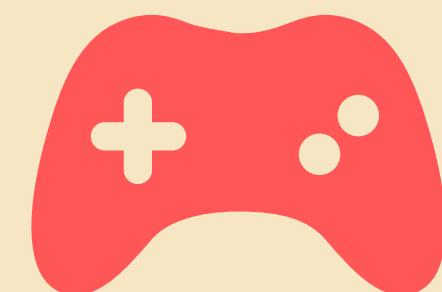
觸及50位使用者



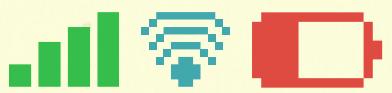
維持健康的體態



養成運動的習慣



遊戲性達到  
讓使用者喜歡的程度



# THANK YOU

