

排球遊戲

以下是要做排球遊戲可能要決定的事情

哪種遊戲：網頁、手遊、單機…

角色建立：自己畫 or 抓別人的圖…，設計角色的軟體

關卡設計：可從現實中初學者學習順序參考

各種動作在遊戲中如何實現

新手遊戲&排球知識教學(動作、規則、戰術…)

加入球隊：球隊積分、勝率…

個人&球隊成績計算方式

遊戲模式：

練習模式：單人→玩家→電腦、多人→玩家→好友

比賽模式：單人→玩家→電腦、每隊多人(2~5 人)→玩家→好友 or 系統配對、

六人對六人→系統配對 or 球隊比賽

遊戲難度：新手、簡單、中等、困難。

遊戲練習及比賽場域：室內、室外、沙灘排球…

遊戲的視角

遊戲的程式語言：前端、後端、資料庫等等

可以教我們開發遊戲的人 or 老師

市面上排球遊戲：Spike Volleyball

1. 排球遊戲目標

1.1 目標

核心目標：開發一種包括介紹排球知識、規則、戰術兼娛樂的排球遊戲

目標用戶：遊戲愛好者、排球愛好者、初學者、專業選手。

平台選擇：PC、主機（如 PS5、Xbox）、行動裝置（iOS、Android）、網頁。

1.2 遊戲概念設計

遊戲模式：

教學模式：排球基本動作（如發球、接發球、扣殺）及規則互動式教學。

職業模式：模擬職業球員訓練和比賽。

多人模式：線上對戰及本地多人合作。

休閒模式：簡化規則的排球遊戲。

遊戲核心機制：

動作捕捉系統，模擬真實排球動作。

排球物理引擎，真實模擬球體運動。

2. 系統架構

2.1 系統規格與技術需求

遊戲引擎：Unity 或 Unreal Engine（選擇基於需求）。

畫面需求：

高精度 3D 模型（球場、角色、球）。

動態燈光和環境效果。

物理模擬：

真實物理引擎（如 NVIDIA PhysX）。

網路架構：

P2P 或伺服器端多人連線系統。

2.2 軟體需求：

建模軟體：Blender、Maya。

3D 角色動畫軟體：MotionBuilder。

版本控制：Git。

2.3 系統架構設計

模組化架構：

界面模組：主選單、設置、排行榜。

遊戲模組：排球教學、比賽模擬。

網路模組：多人連線與比賽匹配。

3.開發軟體需求

3.1 原型開發

建立基本遊戲場景（排球場、角色）。

實作核心遊戲機制（如擊球物理）。

測試基礎互動功能。

3.2 核心功能開發

遊戲機制：完善球體物理、角色動作及控制。

教學系統：添加互動式教學步驟。

多人系統：實現線上遊戲配對與伺服器連線功能。

3.3 資源整合

將完成的美術資源（角色、場景、UI）導入遊戲。

動作捕捉數據整合，實現流暢動作。

排球姿勢偵測

1.目標

目標用戶：各種程度，從排球初學者到教練、專業運動員

主要功能：

偵測排球動作（如發球、接球、扣球等），記錄步伐、姿勢、跳躍高度等

提供動作準確性的即時或近即時回饋

即時數據處理與分析結果視覺化呈現

基於大數據分析提供動作改進建議

長期記錄，用於追蹤進步

其他需求

高準確率（>95% 動作辨識準確度）

低延遲（即時分析延遲 <1 秒）

系統可靠性（穩定運行 ≥99.9%）

2.系統架構

系統框架

數據收集：通過鏡頭捕捉動作

動作偵測：進行姿勢偵測、關鍵點特徵提取

動作分類：動作分類與準確度分析

動作評分：利用準確率、召回率、F1 分數和設定好的評分標準評分

回饋：即時回饋（語音/視覺提示）

儲存資料至資料庫：保存用戶歷史數據，用於進步追蹤

3.開發軟體需求

開發環境：Python

即時影像工具：OpenCV

人工智慧框架：

Yolo、OpenPose、MediaPipe Holistic(姿勢偵測)

TensorFlow、PyTorch、SVM、卷積(姿勢分類)

數據分析工具：Pandas、NumPy

視覺化回饋呈現工具：Matplotlib

版本控制：Git（GitHub, GitLab）

4.系統開發流程

4.1 資料蒐集與準備

收集排球動作數據（影像資料）

諮詢排球老師、教練等專業人士標註正確動作和錯誤範例，訂定評分標準

增強數據集：不同角度動作、影像光影變化

4.2 模型開發

姿勢偵測：

用 Yolo、OpenPose、MediaPipe Holistic 提取關鍵點

動作分類：

用 TensorFlow、PyTorch、SVM、卷積分類標註好的影像

4.3 訓練與測試

劃分訓練集、驗證集、測試集

評估指標：準確率、召回率、F1 分數