



國立聯合大學  
NATIONAL UNITED UNIVERSITY

# 資 訊 管 理 學 系

## 實務專題報告書

### 帕金森症衛教遊戲

指導老師：楊宗珂 老師

專題組長：U1033023 李文秀

專題組員：U1033026 蔡宇捷

中 華 民 國 1 1 4 年 3 月 1 日



國立聯合大學  
NATIONAL UNITED UNIVERSITY

# 資 訊 管 理 學 系

## 實務專題審訂書

### 帕金森症衛教遊戲

專題組長：U1033023 李文秀

專題組員：U1033026 蔡宇捷

指導老師簽名：\_\_\_\_\_

中 華 民 國 114 年 3 月 1 日

## 摘要

本研究旨在開發一款帕金森症衛教遊戲，透過遊戲化學習的方式，使玩家能夠掌握帕金森症的相關知識與治療方法。遊戲設計以一名帕金森症患者的視角展開，透過互動式學習讓玩家理解疾病的特性，進而提升自我管理能力。

系統設計考量遊戲化學習的原則，確保遊戲具有清晰的學習目標、適度的挑戰性與即時的反饋機制。玩家在遊戲過程中將面臨不同的日常生活挑戰，透過完成任務獲取有關帕金森症的醫療資訊與應對策略。此外，遊戲還設有互動式教學與測驗模組，讓玩家能夠測試自身對疾病知識的理解程度。

技術層面上，本系統採用 Unity 遊戲引擎進行開發，搭配 C# 語言撰寫遊戲邏輯，並使用 Blender 製作 3D 模型，以確保遊戲畫面的流暢性與沉浸感。此外，系統內建數據分析模組，並利用 MySQL 資料庫存儲各式數據，能夠記錄玩家的學習進度與遊戲表現。

此外，我們將持續蒐集使用者回饋，進行系統調整與功能改進，期望能夠讓這款遊戲成為帕金森症患者及其照護者的重要學習工具。透過這樣的創新方式，我們希望能夠為帕金森症患者提供更多支持，幫助他們提升生活品質，並在學習與遊戲中獲得更多信心與動力。

# 目 錄

壹、 動機與目的 . . . . .	1
貳、 相關研究 . . . . .	2
參、 研究方法 . . . . .	5
肆、 系統開發工具與技術 . . . . .	6
伍、 系統介紹 . . . . .	6
陸、 成果說明 . . . . .	9
柒、 結論 . . . . .	16
捌、 參考文獻 . . . . .	17
玖、 附錄 . . . . .	20
壹拾、 影像授權書 . . . . .	39

# 壹、前言

## 1.1 背景

根據民國 110 年健保資料統計，帕金森氏症病友人數共七萬七千多人[1]，根據臺灣癌症防治網文獻顯示，帕金森氏症好發年齡為老年人口，大約落在 55 歲至 60 歲之間，發病率隨著年齡增長而增加[2]。帕金森症死亡率約為正常人之兩倍，然而，目前發病原因仍未明，且無法治癒[3]。

## 1.2 動機與目的

隨著帕金森症罹患人數不斷增加，帕金森症對我們來說是一種常見卻陌生的疾病。由於帕金森症無法治癒，患者容易認為治療沒有效果，因此部分患者不願配合治療，而加重病情惡化，而我們認為這是由於患者對此疾病未有深刻理解。醫學研究表明，只要患者配合治療其實能有效緩解病情惡化。因此，當疾病發生時，深入了解病情必定可有效改善惡化速度，享受正常的生活。然而當我們深入蒐集此方面的資訊時卻發現，目前可搜尋到的帕金森症衛教只停留在網頁，透過圖文、影音檔去宣導帕金森症的相關資訊，知識性內容通常複雜且較難理解。而帕金森患者好發年齡為老年人口，衛教資訊顯然以簡潔易懂是更加的適合。

此外，遊戲領域在近年來蓬勃發展，研究顯示平均年齡的提高也表示了遊戲受眾的增加，遊戲化學習一詞出現，更是形成一種新興趨勢。因此，若衛教資訊是通過簡潔易懂且富含趣味性的遊戲，相信能更好的使患者了解及管理自身的健康狀況；同時，也更容易向公眾宣傳此疾病的相關知識，促進社會對這一疾病的理解和關注。

## 貳、相關研究

### 一、疾病介紹

#### 1. 背景

帕金森症(PD)在 1817 年首次由 James Parkinson 發現，它是第二常見之神經退化疾病，僅次於阿茲海默症(AD)。當人們大腦中的黑質致密部多巴胺神經元(SN pars compacta)停止工作或死亡，就會引起 PD。其中，從神經系統易受損核的細胞首次受損到臨床症狀的出現之間存在很長的潛伏期。通常在已經喪失了 70-80%的多巴胺神經元之後，PD 的症狀和徵兆才會出現[12]。研究顯示，PD 患者非運動症狀特徵會在運動症狀前就開始，包含憂鬱和認知變化等。

#### 2. 年齡與性別關係

研究提到，美國每年約有 6 萬人被診斷出患有 PD，PD 患者發病平均年齡約為 60 歲，60 歲及以上人群中 PD 的盛行率約為 1%，在 80 歲以上年齡中該盛行率增至 1%至 3%[14]。儘管 PD 主要為老年人的疾病，年輕族群仍有機率患病[29]。而年齡越大，男性的患病機率高於女性會來的越顯著，約為 1.5 倍[18]。

#### 3. 病因

許多研究表明，PD 的病因與危險因子與基因突變有顯著的關係[14][28][15][19][13][6]。了解病因變可為 PD 作些預防，以下大致列幾點做介紹：

- A. 環境：長時間處在有毒環境下，如二硫化碳、農藥、甲醇、氟化物等，皆會增加風險。暴露於含有重金屬的空氣顆粒（例如在工業區域）同等。
- B. 飲食：高熱量攝取。
- C. 年齡：風險隨著年齡的增長而增加，診斷平均年齡為 65 歲。
- D. 性別：男性面臨更高的風險。
- E. 家族：有近親患有這種疾病會增加風險。
- F. 頭部外傷：即使在數年後，外傷性腦損傷也會增加 PD 的風險。
- G. SNCA：最常見的突變基因。PD 患者腦細胞中，此蛋白質聚集成為路易氏體的團塊。
- H. PINK1：第二常見的突變基因。
- I. LRRK2：突變取代了 dardarin protein 中的單一胺基酸，從而影響了該蛋白的結構和功能[20]。

### 二、症狀

1. 運動症狀：顫抖、肌肉僵硬、姿勢不穩定、發聲功能減退、書寫困難、吞嚥困難、憂鬱、認知問題、失智、恐懼焦慮、睡眠問題、快速動眼（REM）行為障礙、直立性低血壓、便秘、出汗等。

### 三、治療

以現有技術，此疾病無法根治，但仍舊可靠著手術、藥物及復健來延緩惡化。

#### 1. 藥物治療

治療 PD 的藥物有很多種，主要分為三大類[26]：增加大腦中多巴胺水平的藥物、緩解某些疾病症狀的藥物、有助於控制疾病的非運動症狀或不影響運動的症狀的藥物。

A. 左旋多巴：治療 PD 最有效的藥物，對於運動遲緩、震顫及僵硬產生反應。常見副作用有頭痛、頭暈、噁心及嗜睡等；較嚴重有精神錯亂、煩躁及幻想等[17]。

B. 多巴胺激動劑：透過直接刺激大腦中的多巴胺受體發揮作用。臨床試驗發現其可有效控制 PD 症狀。效果略低於左旋多巴，且副作用多。可做為早發性帕金森症患者的初始治療。

C. MAO-B 抑制劑：透過阻斷使多巴胺失活的酵素的作用來發揮作用，可讓多巴胺在被分解之前在大腦中保留更長的時間。可作為輕度運動症狀患者的初始治療。副作用則包括：噁心、頭痛和入睡困難。

D. COMT 抑制劑：主要作用於治療運動功能波動的患者及增加左旋多巴的作用，單獨服用無益。副作用包括運動障礙、幻覺、精神錯亂、噁心、腹瀉、尿液呈現橙色和站立後血壓低。

E. 抗膽鹼藥：對於 70 歲以下，行走緩慢困難不明顯的患者可建議使用來改善震顫的症狀，可單獨使用。常見的副作用包括口乾、視力模糊、便秘、噁心和心率加快等。

F. 腺苷受體拮抗劑（A2A 受體拮抗劑）：用於調節多巴胺反應以及增加多巴胺釋放的腦部相關區域。單獨服用無益。副作用包括運動障礙、便秘、噁心、幻覺等。肝臟問題的患者不適用。

#### 2. 物理治療

根據研究表明[30]，物理治療可顯著改善與運動技能相關的症狀。帕金森氏症基金會表示，每週增加體力活動 2.5 小時或更多可以幫助 PD 患者維持生活品質。

常見的治療種類有：

A. 平衡訓練：改善跌倒問題。

B. 伸展運動：提高靈活性並減少肌肉僵硬等症狀。

C. 肌力訓練：若無常規的肌肉訓練會導致肌肉無力。加強肌力可改善運動症狀。

#### 3. 手術治療

當劑末現象發生時，手術將成為可能的治療方式。有深部腦刺激術、蒼白球燒灼術等[4]。

### 四、罹患應注意

1. 飲食：均衡飲食可提供對患者有益的營養素，如  $\omega$ -3 脂肪酸。減少加工品的攝入。

2. 日常：應透過保持社交互動來保持認知能力，控制血壓和膽固醇等血管危險因素。避免跌倒[21][22]。

3. 生活習慣：運動可增加肌力、柔韌性和平衡。也可改善健康狀況，減少憂鬱焦慮。

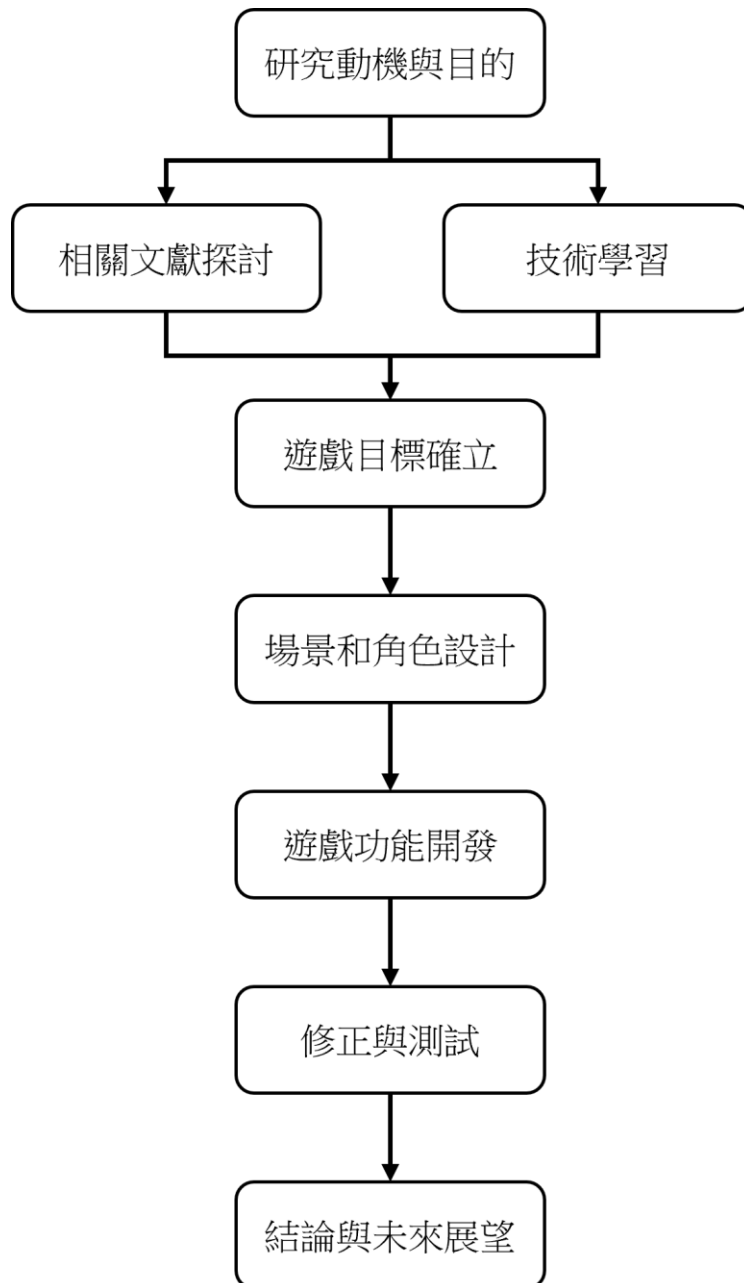
## 五、遊戲化學習

根據研究顯示，遊戲化實際上可能對學習有效[23]。遊戲化可將記憶力和回憶力提高40%，68%使用遊戲化學習時感到更有動力、更投入[27]。然而，對於不同受眾可能會影響它的效果，研究表明，參加遊戲化課程的學生表現優異，因為他們可以專注於學習。如遊戲化細胞生物學課程的大學生比聽課的學生表現高出40%[16]，然而，同時也有研究提到，其除了不會增加結果之外，還會降低快樂和動力[32][7]。即使如此，認為遊戲化是有效的還是佔據大部分的研究結果[24]，神經科學家正在越來越發現人類對此類互動設計元素的反應。這些元素可以引起感覺良好的化學反應，改變人類對刺激的反應，如增加反應時間，並在某些情況下可以改善學習、參與和動機[31]。由此，本專題將使用遊戲化學習的方式來建構衛教系統。



## 參、研究方法

本研究流程與步驟如圖所示，首先由動機與目的來確立我們的主題後，探討相關文獻與學習必要技術，接著確立遊戲目標、設計遊戲場景及角色；開發遊戲的各個功能；修正測試系統，最後做出結論。



## 肆、系統開發工具與技術



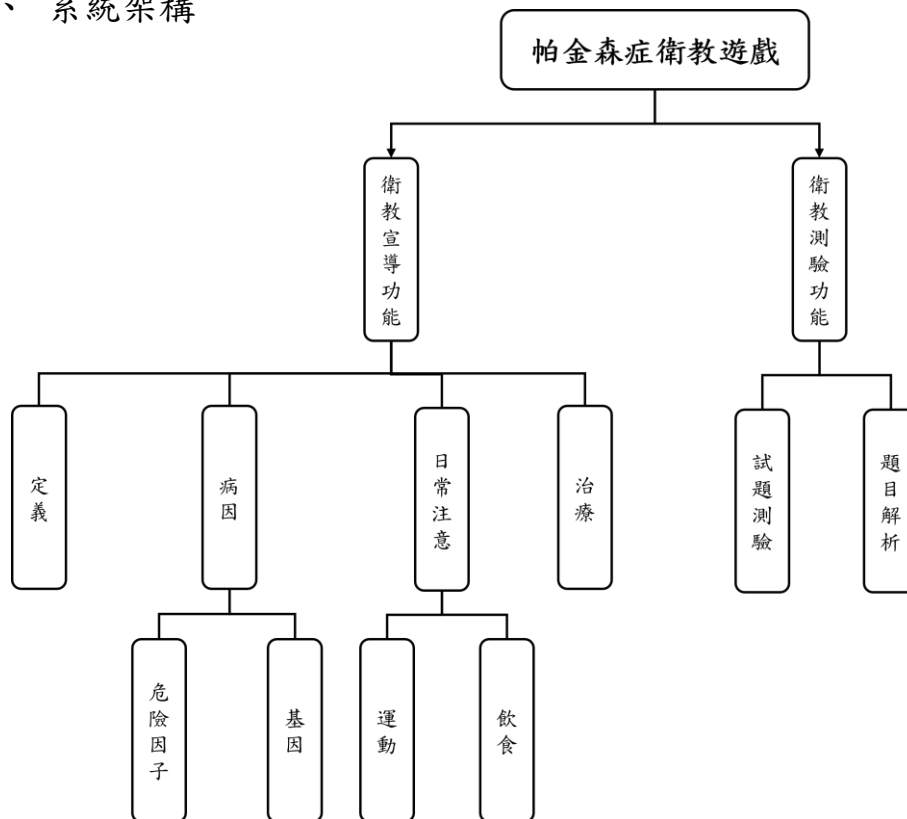
1. unity: 跨平台的遊戲引擎，用於開發 3D 遊戲。
2. C#: 主要程式語言，用於開發遊戲的邏輯、功能和遊戲性。
3. Blender: 用於創建 3D 模型，包括遊戲場景中的建築物 and 物品等。
4. PHP：用於衛教測驗功能中，存儲資料、保存遊戲數據。

## 伍、系統介紹

### 一、系統使用對象

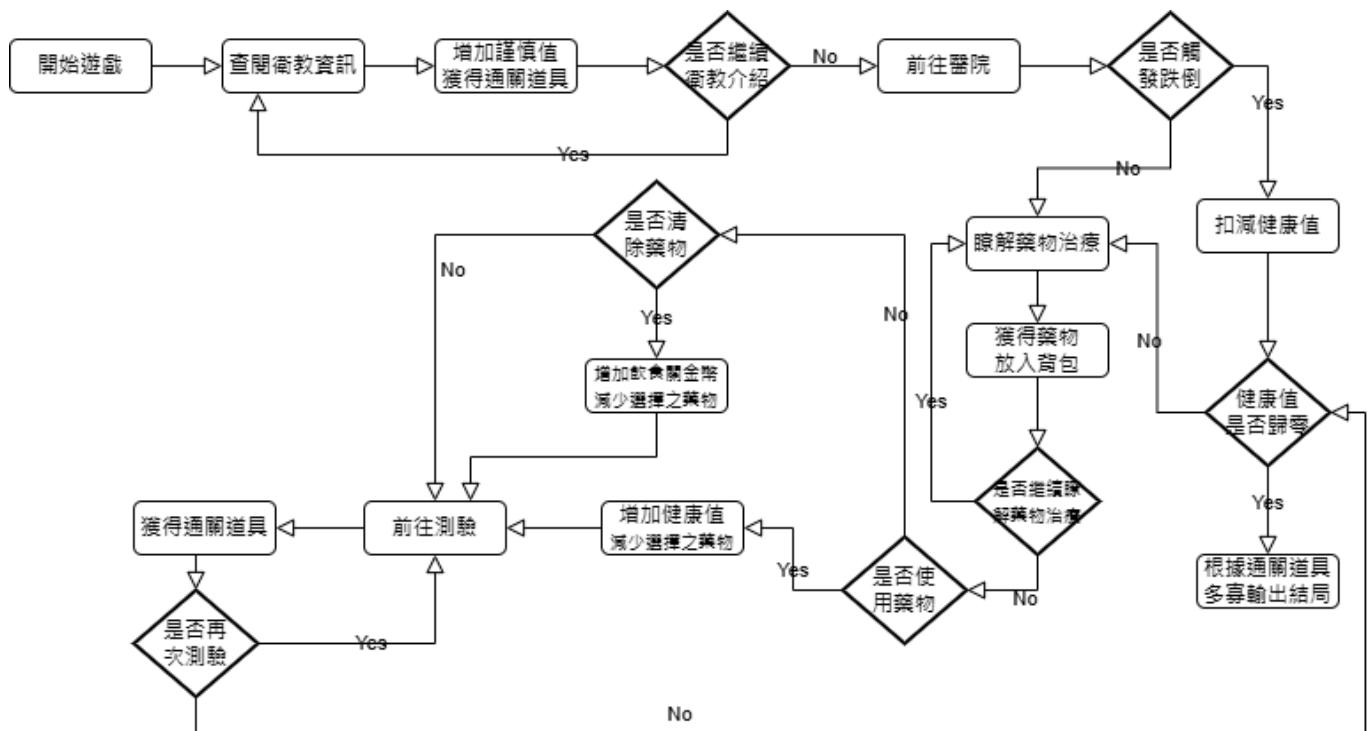
主要針對前中期之帕金森症患者。

### 二、系統架構



(圖一) 系統架構圖

### 三、遊戲流程



(圖二) 遊戲流程圖

遊戲流程說明：

1. 開始進程 → 查看衛教資訊，獲取健康知識與通關道具。
2. 醫療評估 → 決定是否進行衛教介紹，若是則前往醫院診療。
3. 診療階段：
  - 學習相關治療知識→ 獲得藥物並管理背包內物品。
4. 健康值變化：
  - 若健康值歸零 → 遊戲結束並根據通關道具數輸出角色結局。
  - 健康值未歸零 → 可依據道具進行決定是否治療。
5. 測驗 → 完成可獲取通關道具；未完成則可回到前面步驟重新學習。

## 四、功能介紹

### 1.衛教宣導功能

分為定義、病因、日常注意與治療四大部分介紹。定義部分透過對話使使用者了解帕金森症為何；病因部分透過對話與蒐集使使用者了解導致帕金森症的可能因素，並分為基因與危險因子兩部分；日常注意透過角色操控來使使用者了解若已罹患疾病須注意事項，並分為飲食及運動兩部分；治療部分透過文字與語音介紹，使使用者了解患者的常見藥品種類。

### 2.衛教測驗功能

主要功能可以分為測驗以及解析，藉由測驗來藉由測驗來讓使用者知道自己對帕金森症的了解程度，在回答問題完後給予解析，讓使用者知道該問題給予的幫助。測驗結束後可以查詢最近五筆的測驗紀錄，讓使用者可以對病症有更深入的了解。

## 五、系統特色

1. 衛教知識介紹部分以遊戲對話、語音文字互動的方式呈現，可使得資訊更容易被理解和吸收。
- 2.提供簡易的小測驗，涵蓋清晰的介面、並同步現實時間做紀錄，使用者可以獲得對帕金森症更直觀正確的資訊。

## 六、系統使用環境

Windows 環境。

## 陸、成果說明

一、為遊戲開始的主畫面，可透過按鈕決定遊戲進度，新遊戲開始或接續上次遊玩進度



二、故事劇情前情提要，使玩家對遊玩目的有一定的了解



三、清楚介紹玩法與遊戲關卡



四、定義關卡之遊玩介面，透過生動的方式介紹帕金森症為何、症狀及治療等



五、病因介紹中基因關卡之遊玩畫面，透過互動、對話及蒐集介紹常見的突變基因





六、病因介紹中危險因子關卡之遊玩畫面，透過蒐集與互動來介紹已被證實可能導致帕金森症的因素





七、飲食關卡中主畫面，介紹對患者有益或有害食物。並設置關卡介紹，清楚介紹此關玩法



八、鼠標移至食品上方能觀看相關說明，藉此認識該食物

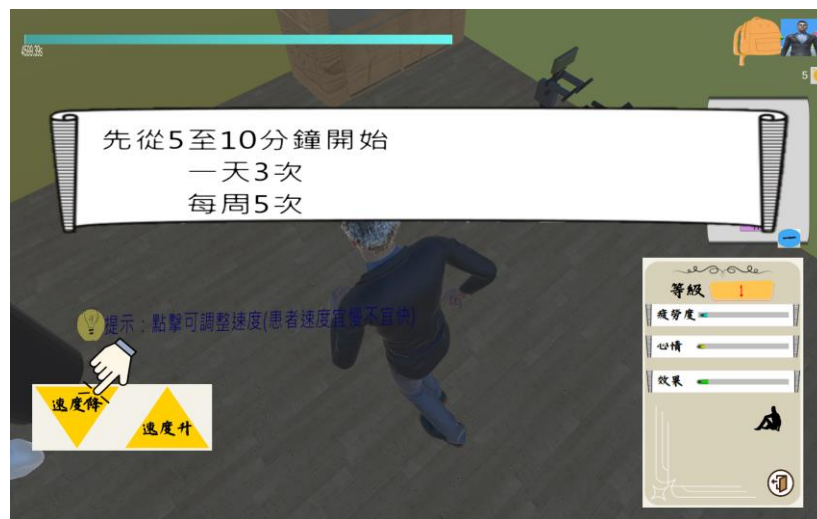


九、成功購買的食品會出現在角色前方，點擊可與其互動，若食用對患者有益之食品將增加健康值(即左上方之數值)



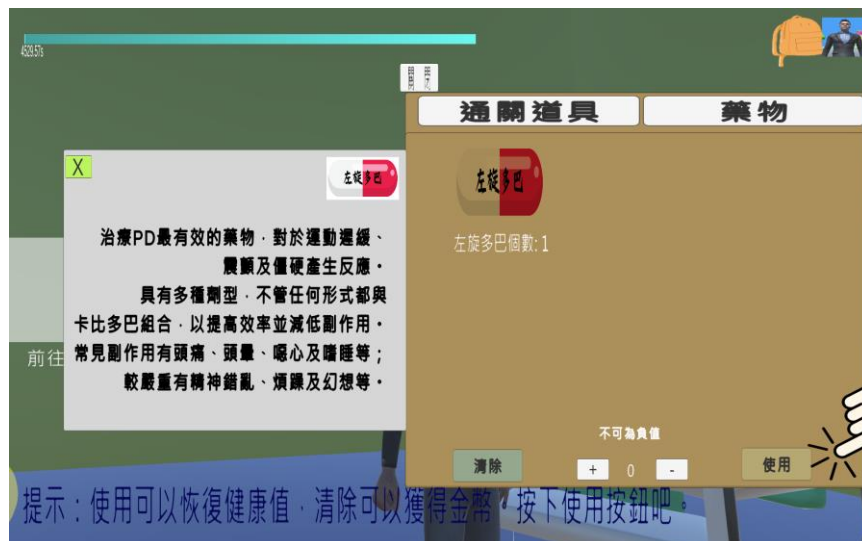


十、運動關卡將介紹幾種對患者有益之日常運動(如：慢跑、跳舞、坐式腳踏車、阻力訓練等)，並介紹運動頻率等



十一、透過與醫生角色的對話獲取藥物，介紹幾種帕金森症常見之治療藥品，使用藥品可增加健康值

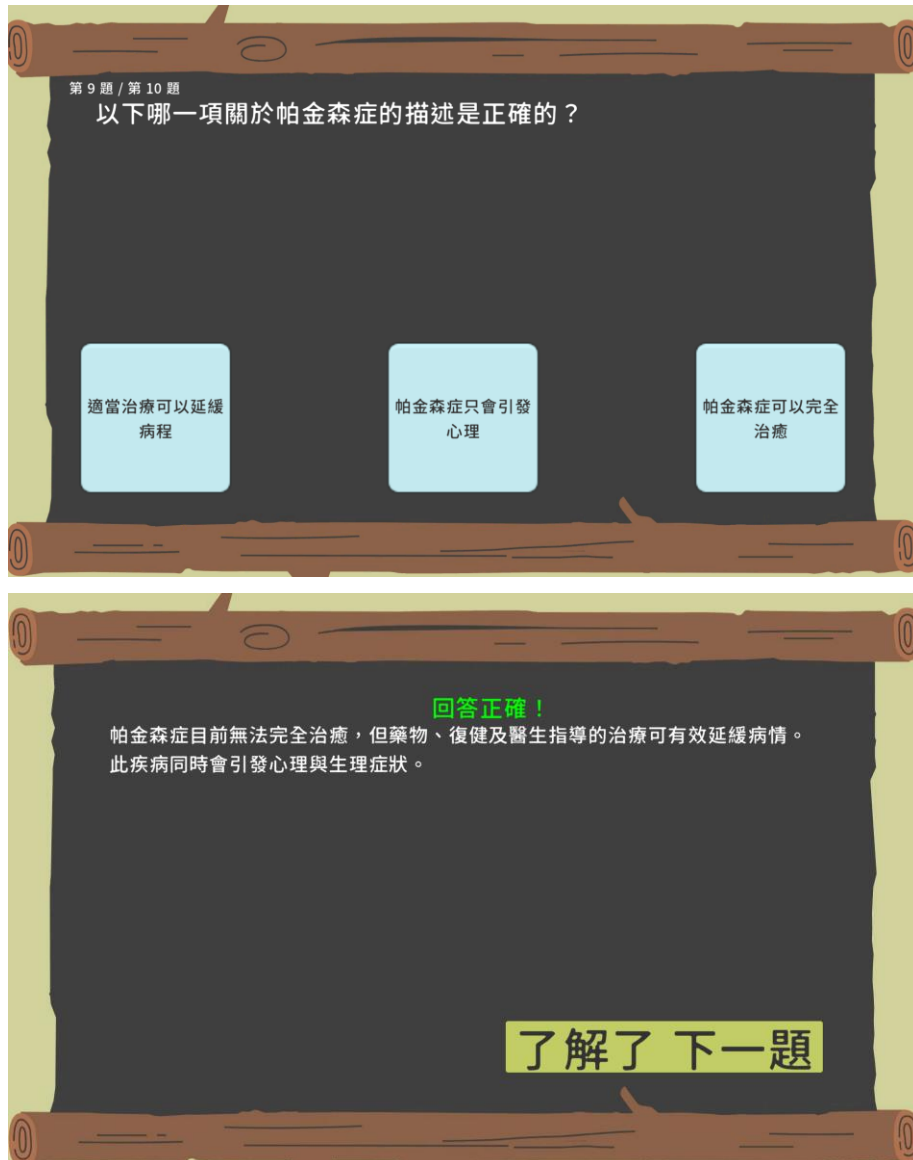




十二、為使使用者了解已學習到多少衛教知識而設置測驗關卡，此測驗關卡之主畫面，提供三按鈕：查看紀錄、開始測驗、離開

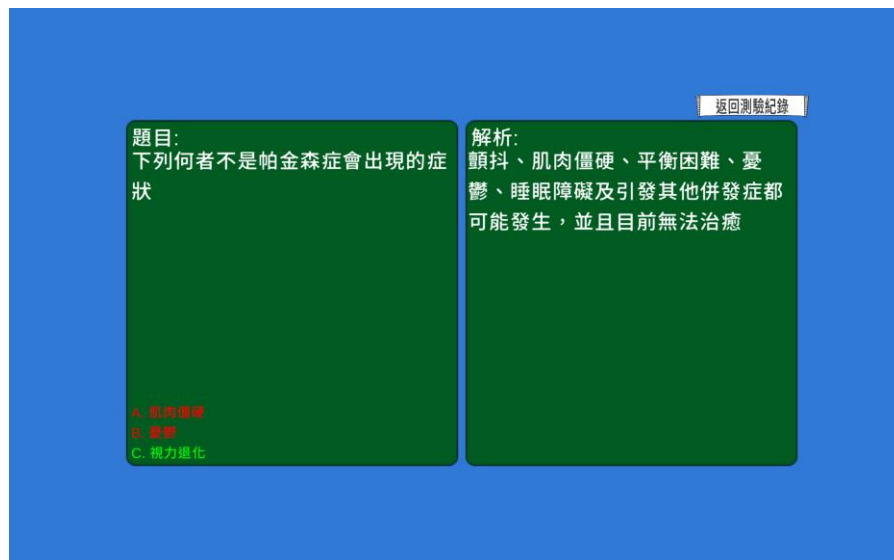


### 十三、測驗進行中之畫面，並提供解析

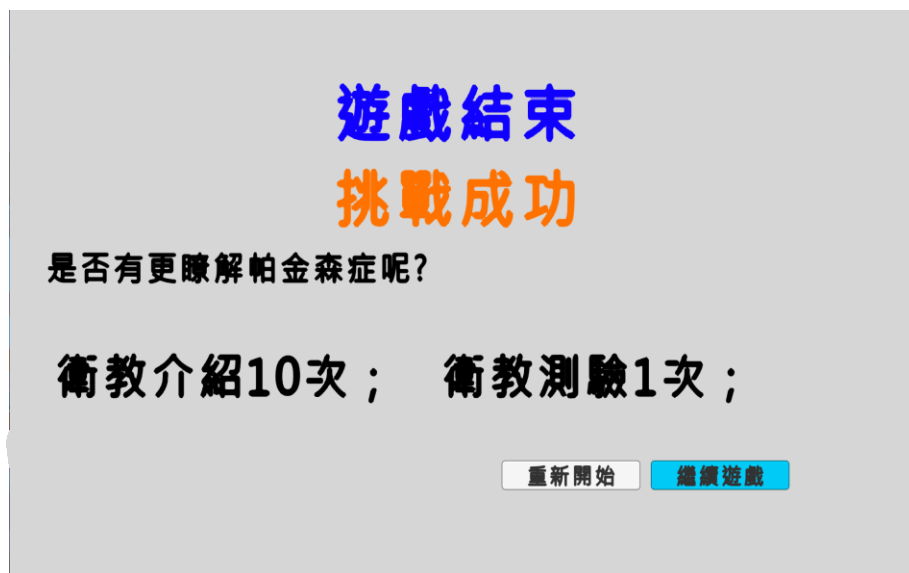


### 十四、測驗紀錄之畫面，紀錄可點擊進入查看相關資訊，包括測驗題目及解析等

近五次測驗紀錄		返回測驗選單
測驗日期: 2025-03-05 17:06:47, 答對: 6 題		
測驗日期: 2024-12-10 16:32:01, 答對: 3 題		
測驗日期: 2024-12-06 09:53:55, 答對: 5 題		
測驗日期: 2024-12-06 08:41:45, 答對: 2 題		
測驗日期: 2024-12-06 08:17:41, 答對: 4 題		
錯誤次數前五高題目		
題目ID: 34 錯誤次數: 18		
題目ID: 36 錯誤次數: 16		
題目ID: 39 錯誤次數: 16		
題目ID: 33 錯誤次數: 15		
題目ID: 24 錯誤次數: 14		



十五、此為通關之畫面



## 柒、結論

透過互動式學習，將嚴肅的醫療知識以生動且淺顯易懂的方式呈現。患者與其家屬能夠在遊戲過程中理解帕金森症相關衛教知識。相較於傳統的文字與影音衛教，我們提供更具有沉浸感的學習體驗，讓使用者在遊戲中累積知識，並能夠檢視自身的學習成效。

成員們透過此專案深化了跨領域的協作能力，從需求分析、文獻蒐集與整理、系統設計、遊戲開發到測試與調整，皆需密切溝通與分工，確保專案順利進行。技術層面上，我們學習到 unity 引擎的開發、C# 及 php 語言的實務操作、建模軟體的摸索等等，這對於組員的成長都提供莫大的幫助。

未來，我們希望進一步優化本系統，並透過與醫療機構或病友團體合作，讓更多帕金森症患者能透過這款遊戲獲得實質的幫助，透過持續改進與拓展應用場景，我們相信本系統將能發揮極大的影響力。

## 捌、參考文獻

- [1]謝承恩(2022)巴金森病友年增 2 千人 左旋多巴藥物使用量最多 |聯合報  
<https://health.udn.com/health/story/122253/6191815>
- [2]葉篤學(2019) 帕金森氏症治療現況 癌症新探 89 期癌症免疫與細胞治療專輯|台灣癌症防治網 <http://web.tccf.org.tw/lib/addon.php?act=post&id=4500>
- [3]黃慧玫(2020)防帕金森氏症？！最新研究發現，帕金森氏症發病與此疾有關|醫療 KingNet 國家網路醫藥 <https://www.kingnet.com.tw/news/single?newId=44708>
- [4] 張智傑(2012) 帕金森氏症病人復健輔助系統 An Assistive System for Patients with Parkinson's Disease[碩博士論文] 國立中央大學資訊工程學系
- [5] Alexis Elbaz & E. Ray Dorsey & Emma Nichols & Foad Abd-Allah(October 2018) Global, regional, and national burden of Parkinson's disease, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016 DOI:10.1016/S1474-4422(18)30295-3
- [6] Antonina Kouli & Kelli M. Torsney & Wei-Li Kuan Parkinson's Disease: Etiology, Neuropathology, and Pathogenesis Thomas B. Stoker & Julia C. Greenland(Eds) Parkinson's Disease: Pathogenesis and Clinical Aspects (2018/ 12/ 21) Codon Publications  
<https://www.ncbi.nlm.nih.gov/books/NBK536722/>
- [7] Amon Rapp & Frank Hopfgartner & Juho Hamari(July 2019) Strengthening gamification studies: Current trends and future opportunities of gamification research. International Journal of Human-Computer Studies 127:1-6, DOI:10.1016/j.ijhcs.2018.11.007
- [8] Angelica Bottaro(September 23, 2021) 10+ Facts About Parkinson's Disease Plus common misconceptions about what it's like. Verywell Health
- [9] Chih-Ching Liu & Chung-Yi Li & Pei-Chen Lee & and Yu Sun(Received 25 September 2015; Accepted 15 December 2015) Variations in Incidence and Prevalence of Parkinson's Disease in. Hindawi Publishing Corporation, Volume 2016, Article ID 8756359, 8 pages,  
<http://dx.doi.org/10.1155/2016/8756359> Taiwan: A Population-Based Nationwide Study
- [10]Cleveland Clinic Parkinson's Disease  
<https://my.clevelandclinic.org/health/diseases/8525-parkinsons-disease-an-overview#symptoms-and-causes>
- [12] Fatemeh Nouri Emamzadeh & Andrei Surguchov (30 August 2018) Parkinson's Disease: Biomarkers, Treatment, and Risk Factors Neurosci Volume 12 - 2018 |  
<https://doi.org/10.3389/fnins.2018.00612>
- [13] Frédéric Moisan<sup>1</sup> & Sofiane Kab<sup>1</sup> & Fatima Mohamed & Marianne Canonico & Morgane

Le Guern1& Cécile Quintin & Laure Carcaillon & Javier Nicolau & Nicolas Duport& Archana Singh-Manoux & Marjorie Boussac-Zarebska & Alexis Elbaz1 (December 23, 2015.)

Parkinson disease male-to-female ratios increase with age: French nationwide study and meta-analysis. French nationwide study and meta-analysisJournal of Neurology, Neurosurgery & Psychiatry 2016;87:952-957.

[14] George DeMaagd, PharmD & BCPS & Ashok Philip, PhD Parkinson's Disease and Its Management. NIH(National Library of Medicine),P T. 2015 Aug; 40(8): 504-510, 532.

[15]JOHNS HOPKINS MEDIDINE The Genetic Link to Parkinson's Disease.  
<https://www.hopkinsmedicine.org/health/conditions-and-diseases/parkinsons-disease/the-genetic-link-to-parkinsons-disease>

[16] Jun Chen1 & Mo Liang(10 October 2022) Play hard, study hard? The influence of gamification on students' study engagement. Volume 13 - 2022,  
<https://doi.org/10.3389/fpsyg.2022.994700>

[17] Kelvin L Chou, MD & Amy Ferng, MD(Nov 30, 2023) Patient education: Parkinson disease treatment options — medications (Beyond the Basics).

[18] Lorraine V Kalia, MD & Dr Anthony E Lang, MD(April 19, 2015) Parkinson's disease[Seminar presentation]. VOLUME 386, ISSUE 9996, P896-912, AUGUST 29, 2015  
DOI:[https://doi.org/10.1016/S0140-6736\(14\)61393-3](https://doi.org/10.1016/S0140-6736(14)61393-3)

[19] Manabu Funayama & Kenya Nishioka, Yuanzhe Li & Nobutaka Hattori(11 July 2022)

Molecular genetics of Parkinson's disease: Contributions and global trends. Journal of Human Genetics, volume 68, pages125–130 (2023)

[20]Medline Plus of National Library of Medicine (May 1, 2012) LRRK2 gene leucine rich repeat kinase 2. <https://medlineplus.gov/genetics/gene/lrrk2/>

[21] Mayo Clinic Parkinson's disease.  
<https://www.mayoclinic.org/diseases-conditions/parkinsons-disease/symptoms-causes/syc-20376055>

[22] Molly C. Cincotta, MD (May 19, 2023) Newly Diagnosed With Parkinson's Disease? Here's What You Need to Know. Temple Health

[23] Michael Sailer(August 28, 2019) Does Gamification of Learning work?

Gamification Research Network News, discussion and resources on the use of game design in non-game contexts

[24] MEHRNOOSH KHOSHNOODIFAR, PhD & ASIEH ASHOURI, MSC student, PhD &

MAHDOKHT TAHERI, PhD(Oct 2023) Effectiveness of Gamification in Enhancing Learning and Attitudes: A Study of Statistics Education for Health School Students. 11(4): 230–239.doi: 10.30476/JAMP.2023.98953.1817

[26]National Institute of Neurological Disorders and Stroke(January 08, 2024) Parkinson's Disease <https://www.ninds.nih.gov/health-information/disorders/parkinsons-disease>

[27] Nikita Verma(February 19, 2023) How Effective is Gamification in Education? 10 Case Studies and Examples.  
<https://axonpark.com/how-effective-is-gamification-in-education-10-case-studies-and-examples/>

[28] OHSU Brain Institute. Understanding Parkinson's Disease.  
<https://www.ohsu.edu/brain-institute/understanding-parkinsons-disease>

[29] P. Chand & I. Litvan(12 February 2007.) Parkinson's Disease. Encyclopedia of Gerontology (Second Edition), 2007, Pages 322-333

[30]PAM HEALTH Types of Physical Therapy for Parkinson's Disease  
<https://pamhealth.com/company/company-updates/types-of-physical-therapy-for-parkinsons-disease>

[31] PEW RESEARCH CENTER(MAY 18, 2012) The Future of Gamification  
<https://www.pewresearch.org/internet/2012/05/18/the-future-of-gamification/>

[32] Rodrigo Smiderl & Sandro José Rig & Leonardo B. Marques & Jorge Arthur Peçanha de Miranda Coelho & Patricia A. Jaques(09 January 2020) The impact of gamification on students' learning, engagement and behavior based on their personality traits. Smart Learning Environments volume 7, Article number: 3 (2020)

[33] Shaziya Allarakha(12/22/2022) What Are the 5 Stages of Parkinson's Disease?  
[https://www.medicinenet.com/what\\_are\\_the\\_5\\_stages\\_of\\_parkinsons\\_disease/article.htm](https://www.medicinenet.com/what_are_the_5_stages_of_parkinsons_disease/article.htm)

## 玖、附錄

一、3D 角色控制器，適用於 Unity，負責角色的 移動、旋轉、跳躍，以及 偵測地面。

包含：平滑移動與旋轉、重力與跳躍機制、地面檢測

### 1. 角色移動 (SetMoveInput & FixedUpdate)

A. 使用 CharacterController 處理角色的移動。

B. SetMoveInput(Vector3 input): 設定移動輸入，當輸入大於 0.1 時才記錄移動向量。

C. FixedUpdate(): 若有移動輸入，則將角色朝著輸入方向移動。角色會朝向 `_lookDirection` + 主相機的旋轉角度 (`_mainCamera.transform.rotation.eulerAngles.y`)。

D. Quaternion.Slerp 用於平滑旋轉角色，防止瞬間轉向。

### 2. 角色旋轉 (SetLookDirection)

A. `_lookDirection` 設定角色面向方向。

B. 角色的朝向與移動方向一致，並根據主相機方向調整。

### 3. 角色跳躍 (Jump)

只有當角色在地面 (`_isGrounded`) 時，才允許跳躍。

### 4. 計算跳躍初速度：

A. `jumpVelocity = Mathf.Sqrt(2f * -_gravity * _JumpHeight);`

B. 垂直速度 `_velocity.y` 設為 `jumpVelocity`，使角色跳起。

### 5. 地面偵測 (CheckGround)

A. 使用 `Physics.SphereCast` 判斷角色是否接觸地面：

B. `_groundCheckRadius`: 接觸判定的圓形半徑，避免角色卡在坡道邊緣。

C. `_groundCheckDistance`: 判斷角色與地面的距離。

D. 若接觸地面，回傳 `true`，並更新 `_groundNormal` (地面法向量)。

### 6. 地面偵測可視化 (OnDrawGizmosSelected)

使用 Unity Gizmos 在場景中繪製紅色或綠色圓圈，來顯示地面檢測範圍。



```

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ControllerMovement3D : MonoBehaviour
{
    [Header("Movement")]

    [SerializeField] private float _moveSpeed = 10f; //移動速度

    [SerializeField] private float _turnSpeed = 10f; //旋轉速度

    [SerializeField] private Camera _mainCamera; //相機

    private float _speed = 0f;

    private bool _hasMoveInput;

    private Vector3 _moveInput;

    private Vector3 _lookDirection;

    [Header("Jumping")]

    [SerializeField] private float _gravity = -20f;

    [SerializeField] private float _JumpHeight = 2.5f;

    private Vector3 _velocity;

    [Header("Grounding")]

    [SerializeField] private float _groundCheckOffset = 0f;

    [SerializeField] private float _groundCheckDistence = 0.4f;

    [SerializeField] private float _groundCheckRadius = 0.25f;

    [SerializeField] private LayerMask _groundMask;

    private bool _isGrounded;

    private Vector3 _groundNormal;

    private CharacterController _characterController;

    private void Start()

```

```

{
    _characterController = GetComponent<CharacterController>();
    if (_mainCamera == null)
    {
        Debug.LogError("Main camera is not assigned to the ControllerMovement3D
script!");
    }
}

public void SetMoveInput(Vector3 input)
{
    _hasMoveInput = input.magnitude > 0.1f;
    _moveInput = _hasMoveInput ? input : Vector3.zero;
}

public void SetLookDirection(Vector3 direction)
{
    _lookDirection = new Vector3(direction.x,0f,direction.z).normalized;
}

public void Jump()
{
    if(!_isGrounded) return;
    float jumpVelocity = Mathf.Sqrt(2f * -_gravity * _JumpHeight);
    _velocity = new Vector3(0, jumpVelocity, 0);
}

public void FixedUpdate()
{
    _isGrounded = CheckGround();
    _velocity.y += _gravity * Time.fixedDeltaTime;
}

```

```

_characterController.Move(_velocity * Time.fixedDeltaTime);

_speed = 0;

float targetRotation = 0f;

if(_moveInput.magnitude < 0.1f)
{
    _moveInput = Vector3.zero;

    return;
}

if(_moveInput != Vector3.zero)
{
    _speed = _moveSpeed;
}

targetRotation = Quaternion.LookRotation(_lookDirection).eulerAngles.y +
_mainCamera.transform.rotation.eulerAngles.y;

Quaternion rotation = Quaternion.Euler(0,targetRotation,0);

transform.rotation = Quaternion.Slerp(transform.rotation, rotation, _turnSpeed *
Time.fixedDeltaTime);

_moveInput = rotation * Vector3.forward;

_characterController.Move(_moveInput * _speed * Time.fixedDeltaTime);
}

private bool CheckGround()
{
    Vector3 start = transform.position + Vector3.up * _groundCheckOffset;

    if(Physics.SphereCast(start, _groundCheckRadius, Vector3.down, out RaycastHit hit,
_groundCheckDistance, _groundMask))
    {
        _groundNormal = hit.normal;
    }
}

```

```

        return true;
    }

    _groundNormal = Vector3.up;

    return false;
}

private void OnDrawGizmosSelected()
{
    Gizmos.color = Color.red;

    if(_isGrounded)Gizmos.color = Color.green;

    Vector3 start = transform.position + Vector3.up * _groundCheckOffset;

    Vector3 end = start + Vector3.down * _groundCheckDistance;

    Gizmos.DrawWireSphere(start, _groundCheckRadius);

    Gizmos.DrawWireSphere(end, _groundCheckRadius);

}
}

```

二、衛教介紹之醫院場景中，玩家操控醫生角色給予之藥品點擊邏輯，用於藥品介紹與互動

### 1. 藥品互動（進入範圍觸發）

OnTriggerEnter(Collider other): 玩家靠近藥品時，觸發對話面板 PDtalkPanel 顯示，並播放對話動畫。根據學習進度（ScoreBoard.MedLearn），可能會更新指引步驟（guideManage.ShowStep()）。

### 2. 藥品展示與介紹

onClickBtMed(): 玩家點擊 藥品按鈕 BtMed，開始藥品介紹。生成 3D 藥品物件 (objectPrefab)，並將材質更換為對應藥品圖片 (sprites[currentIndex])。顯示對應的藥品介紹文字 (textArray[currentIndex])。若所有圖片顯示完，則隨機顯示一張。

### 3. 分數紀錄與存檔

UpdateScoreBoard(int index): 根據選擇的藥品類型 (index)，更新 ScoreBoard 記錄藥品學習數據。使用 gameDataHandler.SaveDataMeds() 存檔玩家學習的藥品數據。

#### 4. 聲音播放

A. PlayConGreMedSound(): 播放學習成功音效 (ConGreMed)，並自動接續下一個音效 (GoKauJin)。

B. PlayPressMedBtSound(): 播放點擊藥品的音效 (PressMedBt)。

C. StopCurrentSound(): 停止當前音效，避免重複播放。

#### 5. UI 控制

A. onClickBtclose(): 點擊關閉按鈕 (Btclose)，隱藏對話面板 (PDtalkPanel)。

B. texthint、texthint2：負責顯示提示玩家靠近藥品或關閉文本的 UI。

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class MedicineManager : MonoBehaviour
{
    public GameObject PDtalkPanel;
    public Image Imtalk;
    public Button BtMed;    //藥物
    public Button Btclose; //關閉
    public Animator animator;
    public int currentIndex = 0; // 用來追蹤顯示的 sprite
    public GameObject objectPrefab; // 3D 物件的預製件
    public Sprite[] sprites; // 存放 6 張圖片
    private GameObject spawnedObject; // 動態生成的 3D 物件
    public Button BtCon; //介紹藥品按鈕
    public TMP_Text displayText; //介紹藥品文本
```

```

public TMP_Text texthint;    //提示靠近藥品和點擊按鈕

public GameObject Sign;

public TMP_Text texthint2;    //提示可關閉顯示文本

public Image ImText;

public string[] textArray;    //對應文本

public bool isentertalk = false;

private int m1 = ScoreBoard.med1;//藥品數

private int m2 = ScoreBoard.med2;

private int m3 = ScoreBoard.med3;

private int m4 = ScoreBoard.med4;

private int m5 = ScoreBoard.med5;

private int m6 = ScoreBoard.med6;

public bool israndom = false;    // 防止重複點擊

public int randomIndex;

private GameDataHandler gameDataHandler;

private GuideManage guideManage;

public AudioSource audioSource;

public AudioClip ConGreMed;

public AudioClip GoKauJin;

public AudioClip PressMedBt;

private AudioClip currentClip;

private BagManager bagManager;

private ConfiStart confiStart;

public void PlayConGreMedSound()

{

    StopCurrentSound();

```

```

        audioSource.PlayOneShot(ConGreMed);

        currentClip = GoKauJin;

        StartCoroutine(ContinueNextSound());
    }

    public void PlayPressMedBtSound()
    {
        StopCurrentSound();

        audioSource.PlayOneShot(PressMedBt);
    }

    private System.Collections.IEnumerator ContinueNextSound()
    {
        yield return new WaitForSeconds(ConGreMed.length);

        if (currentClip != null)
        {
            audioSource.clip = currentClip;

            audioSource.Play();
        }
    }

    private void OnTriggerEnter(Collider other)
    {
        if(texthint.gameObject.activeSelf)
        {
            return;
        }

        else
        {

```

```

        guideManage.HideStep();

        if(ScoreBoard.MedLearn == 0)
        {
            ScoreBoard.scheduleNum = 32;

            guideManage.ShowStep(ScoreBoard.scheduleNum);

            PlayPressMedBtSound();
        }

        PDtalkPanel.SetActive(true);

        Imtalk.gameObject.SetActive(true);

        displayText.gameObject.SetActive(false);

        animator.SetTrigger("talk");
    }
}

void Start()
{
    guideManage = FindObjectOfType<GuideManage>();
    gameDataHandler = FindObjectOfType<GameDataHandler>();
    bagManager = FindObjectOfType<BagManager>();
    confiStart = FindObjectOfType<ConfiStart>();

    PDtalkPanel.SetActive(false);

    BtMed.onClick.AddListener(onClickBtMed);

    Btclose.onClick.AddListener(onClickBtclose);

    BtCon.gameObject.SetActive(false);

    displayText.gameObject.SetActive(false);

    texthint.gameObject.SetActive(false);

    texthint2.gameObject.SetActive(false);
}

```



```

        ImText.gameObject.SetActive(false);
    }

    private void OnTriggerExit(Collider other)
    {
        Imtalk.gameObject.SetActive(false);
        animator.ResetTrigger("talk");
        animator.Play("Idle");
    }

    void onClickBtclose()
    {
        PDtalkPanel.SetActive(false);
    }

    public void onClickBtMed()
    {
        guideManage.HideStep();
        if(ScoreBoard.MedLearn == 0)
        {
            ScoreBoard.scheduleNum = 33;
            guideManage.ShowStep(ScoreBoard.scheduleNum);
            PlayConGreMedSound();
        }

        texthint.gameObject.SetActive(true);
        Sign.SetActive(true);
        animator.ResetTrigger("talk");
        animator.Play("give");
    }

```

```

if (spawnedObject == null)
{
    spawnedObject = Instantiate(objectPrefab, new Vector3(198.09f, 6.22f, -73.4f),
Quaternion.identity);
}

if (currentIndex < sprites.Length)
{
    israndom = false;

    Debug.Log("ScoreBoard: " + ScoreBoard.med1);

    Material material = spawnedObject.GetComponent<Renderer>().material;

    material.mainTexture = sprites[currentIndex].texture;

    displayText.text = textArray[currentIndex];

    currentIndex++;
}

else // 當所有圖片顯示過一次後，隨機選擇一張
{
    israndom = true;

    int randomIndex = Random.Range(0, sprites.Length);

    Material material = spawnedObject.GetComponent<Renderer>().material;

    material.mainTexture = sprites[randomIndex].texture;

    displayText.text = textArray[randomIndex];
}

Debug.Log("ScoreBoard1: " + ScoreBoard.med1);

    Debug.Log("ScoreBoard2: " + ScoreBoard.med2);

    Debug.Log("ScoreBoard3: " + ScoreBoard.med3);

    Debug.Log("ScoreBoard4: " + ScoreBoard.med4);

    Debug.Log("ScoreBoard5: " + ScoreBoard.med5);

```

```

    Debug.Log("ScoreBoard6: " + ScoreBoard.med6);

    Intalk.gameObject.SetActive(false);

    isentertalk = true;

    bagManager.BtBag.interactable = true;

    confiStart.Confi.interactable = true;

}

public void UpdateScoreBoard(int index)
{
    switch (index)
    {
        case 0: ScoreBoard.med1++; break;
        case 1: ScoreBoard.med2++; break;
        case 2: ScoreBoard.med3++; break;
        case 3: ScoreBoard.med4++; break;
        case 4: ScoreBoard.med5++; break;
        case 5: ScoreBoard.med6++; break;
    }

    int playerId = gameDataHandler.GetCurrentPlayerID();

    gameDataHandler.SaveDataMeds(playerId, ScoreBoard.med1, ScoreBoard.med2,
ScoreBoard.med3, ScoreBoard.med4, ScoreBoard.med5, ScoreBoard.med6);

}

private void StopCurrentSound()
{
    if (audioSource.isPlaying)
    {
        audioSource.Stop();
    }
}

```

```
}  
  
}
```

### 三、測驗中歷史紀錄裡的錯題與解析邏輯

#### 1. 錯題加載與顯示：

A. LoadTopErrors()：從 top\_errors.php 伺服器端獲取錯題列表，並動態生成按鈕顯示錯題資訊（ID 和錯誤次數）。

B. SetupButton()：為每個錯題按鈕設定點擊事件，當玩家點擊時加載對應的題目詳情。

#### 2. 獲取與顯示題目解析：

A. LoadQuestionDetails(int questionID)：根據錯題 ID，向 questions.php 請求該題目的詳細內容。

B. FetchQuestionDetails()：處理伺服器返回的 JSON 數據，並顯示錯題的題目、選項與解析。

C. ShowExplanation(QuestionData questionData)：切換 UI，顯示題目、選項與詳細解析。

#### 3. 選項格式化：FormatOptionsWithColors()：根據正確答案，將選項上色（綠色為正確，紅色為錯誤），幫助玩家理解錯誤。

#### 4. 數據模型：

A. ErrorData：存儲錯題的 ID 和錯誤次數。

B. QuizQuestionData：存儲題目的詳細內容，包括問題、選項、正確答案和解析。

```
using UnityEngine;  
  
using UnityEngine.UI;  
  
using UnityEngine.Networking;  
  
using TMPro;  
  
using Newtonsoft.Json;  
  
using System;  
  
using System.Collections;
```

```

using System.Collections.Generic;

public class QuizErrorAnalysis : MonoBehaviour
{
    public GameObject resultMenu;

    public GameObject resultExplanation; // 顯示題目和解析 Canvas

    public TextMeshProUGUI questionText; // 顯示題目

    public TextMeshProUGUI explanationText; // 顯示解析

    public TextMeshProUGUI optionsText; // 顯示選項

    public Button[] resultButtons;

    private Dictionary<int, int> topErrors = new Dictionary<int, int>();

    private int selectedQuestionID;

    private const string BaseUrl = "http://localhost:8080/"; // 替換為實際的 PHP 路徑

    void Start()
    {
        StartCoroutine(LoadTopErrors());
    }

    IEnumerator LoadTopErrors()
    {
        using (UnityWebRequest request = UnityWebRequest.Get(BaseUrl +
"top_errors.php"))
        {
            yield return request.SendWebRequest();

            if (request.result != UnityWebRequest.Result.Success)
            {
                Debug.LogError("Error fetching top errors: " + request.error);

                yield break;
            }
        }
    }
}

```

```

string json = request.downloadHandler.text;

List<ErrorData> errors = JsonConvert.DeserializeObject<List<ErrorData>>(json);

if (errors.Count > 0)
{
    for (int i = 0; i < resultButtons.Length; i++)
    {
        if (i < errors.Count)
        {
            var errorData = errors[i];

            topErrors[errorData.ID] = errorData.ErrorCount;

            SetupButton(resultButtons[i], errorData);
        }
        else
        {
            resultButtons[i].gameObject.SetActive(false);
        }
    }
}
else
{
    Debug.LogWarning("No top errors available.");
}

}

void SetupButton(Button button, ErrorData errorData)
{

```

```

var textComponent = button.GetComponentInChildren<TextMeshProUGUI>();

if (textComponent != null)
{
    textComponent.text = $"題目 ID: {errorData.ID} 錯誤次數:
{errorData.ErrorCount}";
}

button.onClick.RemoveAllListeners();

button.onClick.AddListener(() => LoadQuestionDetails(errorData.ID));

button.gameObject.SetActive(true);
}

void LoadQuestionDetails(int questionID)
{
    selectedQuestionID = questionID;

    StartCoroutine(FetchQuestionDetails());
}

IEnumerator FetchQuestionDetails()
{
    using (UnityWebRequest request = UnityWebRequest.Get(BaseUrl + "questions.php"))
    {
        yield return request.SendWebRequest();

        if (request.result != UnityWebRequest.Result.Success)
        {
            Debug.LogError("Error fetching question details: " + request.error);

            yield break;
        }

        string json = request.downloadHandler.text;

        List<QuestionData> questions =

```

```

JsonConvert.DeserializeObject<List<QuestionData>>(json);

    QuestionData questionData = questions.Find(q => q.ID == selectedQuestionID);

    if (questionData != null)
    {
        ShowExplanation(questionData);
    }
    else
    {
        Debug.LogError("Question ID not found in the database.");
    }
}

void ShowExplanation(QuestionData questionData)
{
    resultMenu.SetActive(false);
    resultExplanation.SetActive(true);
    questionText.text = questionData.Question;
    explanationText.text = questionData.Explanation;
    optionsText.text = FormatOptionsWithColors(
        questionData.OptionA,
        questionData.OptionB,
        questionData.OptionC,
        questionData.Answer
    );
}

```



```
private string FormatOptionsWithColors(string optionA, string optionB, string optionC, int
correctAnswer)
```

```
{
    string colorA = correctAnswer == 0 ? "<color=green>" : "<color=red>";
    string colorB = correctAnswer == 1 ? "<color=green>" : "<color=red>";
    string colorC = correctAnswer == 2 ? "<color=green>" : "<color=red>";
    return $"{colorA}A. {optionA}</color>\n" +
        $"{colorB}B. {optionB}</color>\n" +
        $"{colorC}C. {optionC}</color>";
}
}
```

```
[System.Serializable]
```

```
public class ErrorData
```

```
{
    [JsonProperty("ID")]
    public int ID { get; set; }
    [JsonProperty("ErrorCount")]
    public int ErrorCount { get; set; }
}
```

```
[System.Serializable]
```

```
public class QuizQuestionData
```

```
{
    [JsonProperty("ID")]
    public int ID { get; set; }
    [JsonProperty("Question")]
    public string Question { get; set; }
    [JsonProperty("OptionA")]
```

```
public string OptionA { get; set; }  
[JsonProperty("OptionB")]  
public string OptionB { get; set; }  
[JsonProperty("OptionC")]  
public string OptionC { get; set; }  
[JsonProperty("Answer")]  
public int Answer { get; set; }  
[JsonProperty("Explanation")]  
public string Explanation { get; set; }  
}
```

# 國立聯合大學 資訊管理學系

## 專題影像使用授權書

級別：110 級

指導老師：楊宗珂 老師

專題名稱：帕金森症衛教遊戲

組員名單：

學號	姓名
U1033023	李文秀
U1033026	蔡宇捷

本數位影像作品，詳如專題名稱，經本人同意無償授權國立聯合大學資訊管理學系，提供本系展覽活動展出及學校之校內刊物編輯及網站公開展示使用。

本數位影像作品，悉為個人原創作品，非抄襲、複製或變造他人作品，如有不實、願自負法律責任。

授權人簽名：

(所有組員皆須簽名)

中華民國      114      年      03      月      01      日