

MQTT LAB 亞瑟王與黑暗魔王的決戰



🌟背景故事：

在遙遠的中世紀王國中，亞瑟王以無畏的勇氣與卓越的智慧成為了世人敬仰的領袖。傳說中的聖劍「石中劍」，這把劍不僅象徵著王者的正統，更賦予了他無與倫比的力量和勇氣。自古以來，只有那些命中注定的英雄才能拔出石中劍，並以此作為王權的象徵。亞瑟王便是那位命定的王者，帶領著他的國家邁向繁榮與強大。

在亞瑟王的統治下，不列顛王國終於迎來了長久的和平與穩定。手握聖劍「石中劍」，他一一擊退了所有敵人，無論是強大的外敵，還是內部的叛亂勢力。亞瑟王的名字如雷貫耳，無人不知無人不曉。他的每一場戰鬥，無論結果如何，都是一個傳奇。他是無數戰場上的英雄，打敗所有敢於挑戰他的對手，世間已無人能與他匹敵。亞瑟王的王國、人民、王座，都在他的掌控之中，安穩無比。

然而，就在亞瑟王剛剛統一不列顛群島，天下無敵之時，黑暗的勢力悄然甦醒。命運彷彿在向亞瑟王開了一個天大的玩笑。這位曾經無敵於天下的英雄，未曾享受過多少和平，卻又面對了最為致命的敵人——這一次，敵人來自天上，一位來自天外的黑暗魔王。這位黑暗魔王，傳說中的邪惡存在，力量強大無比，足以顛覆整個世界的秩序。他的到來，將讓這片原本安穩的土地再次陷入無盡的黑暗與毀滅。

「天下無敵，天上來敵」，亞瑟王低語著，手中的聖劍散發出耀眼的光芒。這句話，猶如一記警鐘，在他的心頭迴響。無論曾經如何無敵，這一次，亞瑟王知道，自己將面對的是一個從天而降的強大對手。他的命運，將在這一戰中決定。

為了守護王國，亞瑟王召集了所有圓桌騎士，準備發起一場攸關生死的終極對決。這一場戰鬥不僅是力量的碰撞，更是智慧與勇氣的對決。他們將運用古老的魔法與現代的神秘技術，透過一種特殊的魔法通訊系統（象徵 MQTT 協議），使得每一位騎士的行動都能即時回應，將戰鬥進程同步。每一位騎士，攻擊後，都會收到來自黑暗魔王的回應——魔王會以陰森的語氣告訴他們，自己剩下多少力量。

作為其中一位圓桌騎士，你（玩家）的任務也已經明確：

- 揮動聖劍，向黑暗魔王發起攻擊！
- 每次攻擊後，魔王會回應你，告訴你他所剩的血量。
- 當魔王受到重創，他的憤怒將激發出更強的反擊！
- 用你的智慧與勇氣，一回合一回合地削弱魔王的力量，直到最終將他擊倒！

這場戰鬥，將是一場智慧與勇氣的考驗。唯有堅持到底，並充分發揮你的戰術與決心，才能粉碎黑暗的陰霾，迎來王國的光明！

✂作業目標

簡報連結：https://www.canva.com/design/DAGjrm_9Dq4/UK7Qiu9WOh1NvpGTJhXOkA/view?utm_content=DAGjrm_9Dq4&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utm_id=h78870d41d4

1. 理解 MQTT 的通訊原理 (Publisher/Subscriber 模式)。
2. 學會使用 Mosquitto broker 作為 MQTT 的伺服器端進行測試。
3. 實作 MQTT 通訊的 Publisher 與 Subscriber 程式邏輯，模擬遊戲中「玩家攻擊 Boss」的情境。

程式作業說明：



請記得安裝環境 !!!，到 student 資料夾下，進行 npm install
若環境仍有誤，或不幸刪掉 package.json，可手動安裝指令：

- `npm install mqtt`

- 同時這次作業需要更改的部分都有標註起來 (標示如下)

```
// === 請實作這裡(以下) ===
```

```
// === 請實作這裡(以上) ===
```

- 你需要觀察 Boss 的回應格式，以及 MQTT 主題的運作，來完成對應的處理。

作業提醒：

- 請先啟動 `Game.js`，再執行 `Player.js`，否則無法順利接收上線通知或回合更新。
- 若 MQTT broker 未啟動 (未執行 `mosquitto`)，會導致無法連線，請務必確認。
- 請先把最下面 Mosquitto 的安裝和測試介紹看完後，再開始作業 !!

作業情境介紹：

這是一個簡單的 MQTT 應用情境，模擬**玩家攻擊 Boss**的遊戲機制：

- **Boss (Game.js)：**
 - 訂閱玩家的攻擊行動，收到攻擊後減少血量，並廣播最新回合資訊與血量。
 - `Game.js`：Boss 的邏輯 (訂閱攻擊、扣血、回覆狀態) ✅ 已實作好。
- **玩家 (Player.js)：**
 - 連線後發起攻擊，訂閱 Boss 的狀態更新，直到 Boss 被擊敗。
 - `Player.js`：玩家邏輯 (上線通知、發動攻擊、接收狀態) 🚧 部分需學生實作
- 輸出樣式說明 (`console.log` 參考用)：
 - **玩家 (Player.js)：**

```
Boss 已連線至 MQTT broker
```

```
對方玩家在線，遊戲開始！
```

```
[ 第 1 回合 ]
```

```
Boss 受到 Player 的攻擊，損失 715 HP，剩餘 HP：4292
```

```
...
```

```
恭喜 🎉🎉🎉 ~~~ Player 攻擊成功，Boss 被擊敗 !!!
```

◦ **Boss (Game.js) :**

Boss 已連線至 MQTT broker

對方玩家在線，遊戲開始！

[第 1 回合]

Boss 受到 Player 的攻擊，損失 715 HP，剩餘 HP：4292

...

Boss 被 Player 擊敗 !! Boss 哭泣 😭😭

測驗題：



測驗題寫入 README (題號 + 答案)

1. 在 MQTT 中，負責中介轉發訊息的元件稱為？
 - A. Client
 - B. Broker
 - C. Publisher
 - D. Subscriber
2. 在本次作業中，Boss 與 Player 之間溝通是透過下列哪個機制？
 - A. REST API
 - B. 發佈與訂閱主題 (topic)
 - C. HTTP GET 請求
 - D. WebSocket 的 binary 資料傳輸
3. 以下哪個動作不屬於 MQTT Client 的基本能力？
 - A. 連線 Broker
 - B. 訂閱 Topic
 - C. 發佈訊息
 - D. 刪除其他 Client 的資料
4. 哪一種情況最有可能需要使用 MQTT？
 - A. 線上影音串流
 - B. 高頻交易
 - C. 低頻、低功耗的感測器數據傳輸
 - D. 伺服器管理後台介面
5. 在本次作業中，battle/action topic 的主要用途是什麼？
 - A. 玩家登入通知
 - B. 傳送遊戲結束訊息
 - C. 玩家發送攻擊，Boss 回傳回合資訊
 - D. 記錄伺服器狀態
6. Boss 需要 被攻擊幾次才會被擊敗？

Ans：

作業提交方式

請提交：

1. **game.js** 程式檔。
2. **player.js** 程式檔。 (✅需實作)
3. 測驗題寫入 README (格式：題號+答案)。

- 測驗題格式: (不是答案)

1. A
2. B
3. C
4. D
5. A
6. 100次

評分方式：

- 程式題：player.js 70分
- 測驗題：每題 6 分 (共6題)
- 一共 $70 + 36 = 106$ 分

MQTT 安裝&測試說明

安裝 MQTT 伺服器 mosquitto

- **Windows：**
 - 請到 <https://mosquitto.org/download/> 下傳並安裝適合的版本

Windows

- [mosquitto-2.0.21a-install-windows-x64.exe](#)
- [mosquitto-2.0.21a-install-windows-x86.exe](#)

- 系統預設安裝 Mosquitto 到 `C:\program files\mosquitto\` 下。
- 安裝過程中，建議 **不要勾選** install as service。

- **Mac：**
 - 請先確定您有安裝 **homebrew** (參考 <https://brew.sh/>)
 - 安裝完 brew 後，請開啟終端機，輸入 `brew install mosquitto`
- 嘗試多次仍安裝失敗的人：
 - 可使用公用測試伺服器 mqtt
 - test.mosquitto.org - test.mosquitto.org:1883

啟動與測試MQTT 伺服器

1. 啟動MQTT伺服器

- **Windows：**開啟命令列視窗 (cmd)
 - `cd` 至 Mosquitto 目錄 (`C:\Program Files\mosquitto`)
 - 執行 `mosquitto.exe` (**指令執行** or **雙擊程式執行** 都行)
- **Mac：**開啟終端機，並按以下步驟執行

1. brew services list
2. brew link mosquitto

3. brew services start mosquitto -d
4. brew services restart mosquitto

2. 測試 **Listener**：新增一個 mqtt listener 來訂閱「EVENT」這個 topic

- **Windows**：另開一個命令列視窗 (cmd)
 - `cd` 至 Mosquitto 目錄 `C:\Program Files\mosquitto` 下
 - 執行 `mosquitto_sub -t EVENT`
- **Mac**：另開一個終端機
 - 執行 `mosquitto_sub -t EVENT`

3. 測試 **Publisher**：

- 新增一個 **mqtt message publisher**，發佈訊息到 **EVENT** 這個 topic。
 - **Windows**：另開一個命令列視窗 (cmd)
 - `cd` 至 Mosquitto 目錄 `C:\Program Files\mosquitto` 下
 - 執行 `mosquitto_pub -t EVENT -m hello`
 - 💡 此時 mqtt listener 中，應該會收到 hello 訊息。
 - **Mac**：另開一個終端機
 - 執行 `mosquitto_pub -t EVENT -m hello`
 - 💡 此時 mqtt listener 中，應該會收到 hello 訊息。
-