

# 2024 年【科學探究競賽-這樣教我就懂】

## 普高組 成果報告表單

題目名稱：生活中的好幫手---DC BOT

### 一、摘要

本文概述了一項研究項目，重點是開發一款名為 Discord Bot ( DC Bot ) 的創新功能，以增強其多功能性。該研究涉及理解 DC Bot 的原理，擴展其功能，並將其與現有市場解決方案進行比較。該研究包括測試各種 DC Bot 配置、收集用戶反饋，並使用深度學習模型集成多模態數據以提高性能。通過文獻回顧和比較分析，評估了不同平台上的不同機器人，突出它們的優勢和局限性。開發過程被記錄下來，包括編程挑戰和定制以滿足用戶需求。未來的方向建議集成人工智能以提供更加個性化的服務，並解決安全問題。對機器人開發的建議強調理解 Discord API、編程語言，並進行功能增強的測試。

### 二、探究題目與動機

在這個充滿日新月異的時代，機器人已成為我們生活的不可或缺的一部分。它們不僅能應用在日常事務，如打掃和飲食，還可以放在社交媒體平台上來運用，例如 Line、Discord 等。並且透過自動化服務，機器人能夠更有效節省時間和金錢，使管理員能更輕鬆地管理伺服器。除了以上一些基本功能外，機器人還提供了娛樂性的功能，例如小遊戲、音樂播放，甚至可以測運勢。這樣的多元功能使機器人成為社交平台上的有趣陪伴，為用戶提供更豐富的使用體驗。而為了讓本研究的自製 DC 機器人更為創新和多元化，我們計劃新增一些獨特的功能。例如加入即時翻譯功能，以促進跨文化交流。

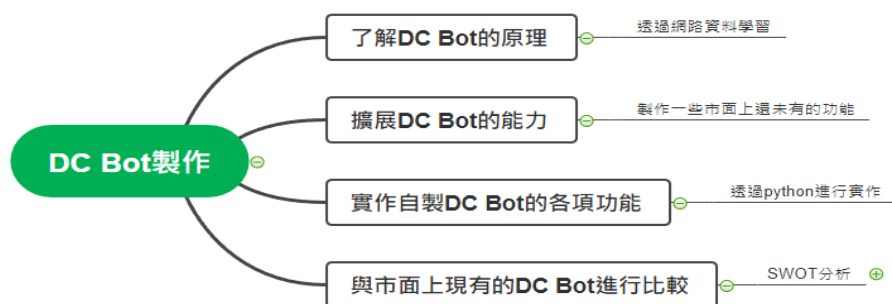
透過這些創新的功能，我們期望可以打造一個更為互動和有趣的機器人，帶來更加豐富的社交體驗並且同時提升機器人在各項領域的實用性。

### 三、探究目的與假設

- 一、了解 DC Bot 的原理
- 二、研究如何擴展 DC Bot 的能力
- 三、實作自製 DC Bot 的各項功能
- 四、與市面上現有的 DC Bot 進行比較

### 四、探究方法與驗證步驟

(一) 研究架構圖



圖一：研究架構圖

## (二) 研究方法

- 1、分析比較法：分析自己與市面上所提供產品兩者間互相的優勢和劣勢，並透過圖表呈現出來。
- 2、實作分析法：透過自己實際操作執行在系統上，這種分析方式可以用來確保實際操作過程能夠達到預期的成果。

## (三) 系統開發流程



圖二：系統開發流程圖

## (四) DC Bot 功能

### 1、啟動 DC Bot

如果想要讓 DC Bot 可以成功執行功能的話，必須要確認 DC Bot 已經被正確地啟動，而啟動 DC Bot 的程式碼正如下圖三所示。

```
1 import discord
2 from discord.ext import commands
3
4 intents = discord.Intents.all()
5 bot = commands.Bot(command_prefix="#", intents = intents)
6
7 @bot.event
8 async def on_ready():
9     print("Bot is ready")
```

圖三：啟動 DC Bot 的程式碼

### 2、數字的加減乘除

在成功啟動 DC Bot 之後，可以開始實作一些基礎的功能，像是數字加減乘除，而它所對應的程式碼以及實際運行的結果，正如下圖四和五所示。

```
25 @bot.command()
26 async def add(ctx, x, y):
27     result = int(x) + int(y)
28     await ctx.send(f"{x} + {y} = {result}")
29
30 @bot.command()
31 async def minus(ctx, x, y):
32     result = int(x) - int(y)
33     await ctx.send(f"{x} - {y} = {result}")
34
35 @bot.command()
36 async def multiple(ctx, x, y):
37     result = int(x) * int(y)
38     await ctx.send(f"{x} * {y} = {result}")
39
40 @bot.command()
41 async def devide(ctx, x, y):
42     result = int(x) / int(y)
43     await ctx.send(f"{x} / {y} = {result}")
```

圖四：數字加減乘除的程式碼

```
Wei. 今天 00:33
#add 456 389

測試機器人 機器人 今天 00:33
456 + 389 = 845

Wei. 今天 00:33
#minus 968 4568

測試機器人 機器人 今天 00:33
968 - 4568 = -3600

Wei. 今天 00:33
#multiple 362 13

測試機器人 機器人 今天 00:33
362 * 13 = 4706

Wei. 今天 00:33
#devide 65 13

測試機器人 機器人 今天 00:33
65 / 13 = 5.0
```

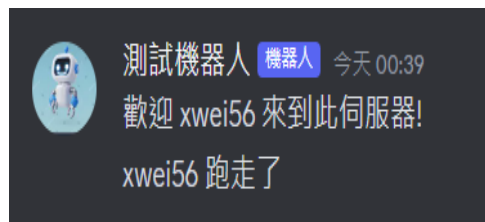
圖五：實際運行結果

### 3、成員的進出通知

當成員進入或退出伺服器時，透過系統播放是一種方便的方法，讓伺服器內的成員可以馬上知道群組成員的進出狀態。它所對應的程式碼以及實際運行的結果，正如下圖六和七所示。

```
11 @bot.event
12 async def on_member_join(member):
13     channel = bot.get_channel(1176558134660169739)
14     await channel.send(f'歡迎 {member} 來到此伺服器!')
15
16 @bot.event
17 async def on_member_remove(member):
18     channel = bot.get_channel(1176558134660169739)
19     await channel.send(f'{member} 跑走了')
```

圖六：成員進出播放的程式碼



圖七：實際運行結果

### 4、翻譯功能

Discord 上有來自世界各地的用戶，為了讓大家更方便溝通，我們加入了翻譯功能，這就代表著，使用不同語言的用戶可以更輕鬆地聊天互動，不會再受到語言的限制，而它所對應的程式碼以及實際運行的結果，正如下圖八和九所示。

```
40 @bot.command(name='translate', help='Translate text to a specified language. Usage: !translate target_language text_to_translate')
41 async def translate_text(ctx, target_language, *, text_to_translate):
42     translator = Translator()
43
44     try:
45         translation = translator.translate(text_to_translate, dest=target_language)
46         translated_text = translation.text
47         await ctx.send(f'**Translation ((target_language))**: {translated_text}')
48     except Exception as e:
49         await ctx.send(f'Error during translation: {str(e)}')
```

圖八：翻譯的程式碼



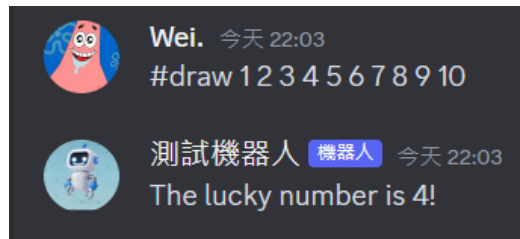
圖九：實際運行成果

### 5、抽籤功能

機器人的抽籤功能可以在 Discord 群組中增添樂趣，可用在公平抽獎、隨機選擇，更甚至是解決難題，讓互動體驗能更豐富化，而它所對應的程式碼以及實際運行的結果，正如下圖十和十一所示。

```
31 @bot.command(name='draw', help='Draw a random item from the given list. Usage: !draw item1 item2 item3 ...')
32 async def draw_lottery(ctx, *items):
33     if not items:
34         await ctx.send('Please provide a list of items to draw from.')
35         return
36
37     selected_item = random.choice(items)
38     await ctx.send(f'The lucky number is {selected_item}!')
```

圖十：抽籤的程式碼



圖十一：實際運行成果

## 6、圈圈叉叉遊戲

在製作遊戲時，我們發現遊戲並不能直接在 DC 裡直接執行，因此我們想到其他替代方案，透過 DC Bot 可以連接 Javascript 的功能，當我們在 DC 裡輸入小遊戲名子時，透過機器人回應給我們的連結，前往別的網站來執行遊戲，小遊戲所對應的程式碼以及實際運行的結果，正如下圖十二和十三所示。

```
function checkDiagonals() {
    return (board[0][0] === currentPlayer &&
    board[1][1] === currentPlayer && board[2][2] ===
    currentPlayer) ||
    (board[0][2] === currentPlayer &&
    board[1][1] === currentPlayer && board[2][0] ===
    currentPlayer);
}

function isBoardFull() {
    for (let i = 0; i < 3; i++) {
        for (let j = 0; j < 3; j++) {
            if (board[i][j] === '') {
                return false;
            }
        }
    }
    return true;
}
```

圖十二：圈圈叉叉程式碼



圖十三：實際運行成果

## 7、貪吃蛇遊戲

跟上面圈叉遊戲相同，透過輸入小遊戲名子，讓機器人回應給我們相對應的連結，來去執行遊戲。貪吃蛇遊戲所對應的程式碼以及實際運行的結果，正如下圖十四和十五所示。

```
function gameStart() {
    snake = {
        body: [
            { x: BLOCK_COUNT / 2, y: BLOCK_COUNT / 2 }
        ],
        size: 5,
        direction: { x: 0, y: -1 }
    };
    putApple();
    updateScore(0);
    updateGameLevel(1);
}

function updateGameLevel(newLevel) {
    level = newLevel;
    if (gameInterval) {
        clearInterval(gameInterval);
    }
    gameInterval = setInterval(gameRoutine, 1000 / (10
    + level));
}

function movesnake() {
    var newBlock = {
        x: snake.body[0].x + snake.direction.x,
        y: snake.body[0].y + snake.direction.y
    };
    snake.body.unshift(newBlock);
    while (snake.body.length > snake.size) {
        snake.body.pop();
    }
}

function updateCanvas() {
    var canvas = document.getElementById("canvas_id");
    var context = canvas.getContext("2d");
    context.fillStyle = "black";
    context.fillRect(0, 0, canvas.width, canvas.height);
}

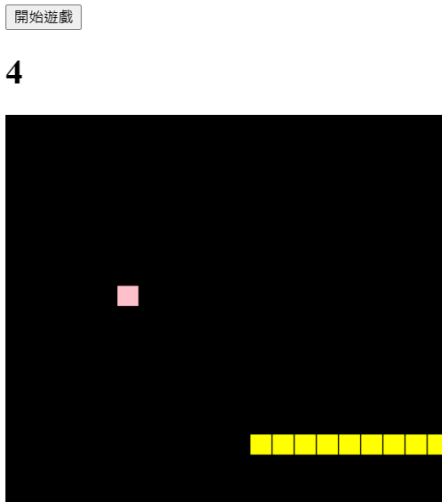
function gameRoutine() {
    movesnake();
    if (snakeIsDead()) {
        return;
    }
    if (snake.body[0].x === apple.x &&
    snake.body[0].y === apple.y) {
        eatApple();
    }
    updateCanvas();
}

function snakeIsDead() {
    if (snake.body[0].x < 0) {
        return true;
    } else if (snake.body[0].x >= BLOCK_COUNT) {
        return true;
    } else if (snake.body[0].y < 0) {
        return true;
    } else if (snake.body[0].y >= BLOCK_COUNT) {
        return true;
    }
}

function updateScore(newScore) {
    score = newScore;
    document.getElementById("score_id").innerHTML =
    score;
}

function putApple() {
    apple = {
        x: Math.floor(Math.random() * BLOCK_COUNT),
        y: Math.floor(Math.random() * BLOCK_COUNT)
    };
    for (var i=0; i<snake.body.length; i++) {
        if (snake.body[i].x === apple.x &&
        snake.body[i].y === apple.y) {
            putApple();
            break;
        }
    }
}
```

圖十四：貪吃蛇程式碼



圖十五：實際運行成果

## 8、天氣預報功能

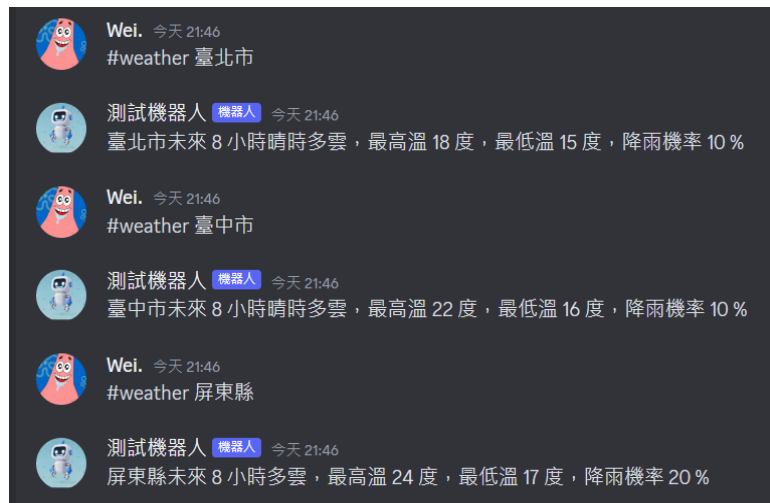
天氣預報如今一直都是生活中不可或缺的一部分，而為了讓 DC 用戶可以及時得到未來天氣的走向，因此我們加入了天氣預報功能，得知各縣市未來 8 小時的天氣內容。天氣預報所對應的程式碼以及實際運行的結果，正如下圖十六和十七所示。

```

56 @bot.command()
57 async def weather(ctx, city):
58     url = f'https://opendata.cwa.gov.tw/fileapi/v1/opendataapi/F-C0032-001?Authorization=CWA-D9C289A3-72AF-49A8-9CBA-60D09927B103&downlo
59     data = requests.get(url)
60     data_json = data.json()
61     location = data_json['cwaopendata']['dataset']['location']
62
63     for i in location:
64         if i['locationName'].lower() == city.lower():
65             wx8 = i['weatherElement'][0]['time'][0]['parameter']['parameterName']
66             maxt8 = i['weatherElement'][1]['time'][0]['parameter']['parameterName']
67             mint8 = i['weatherElement'][2]['time'][0]['parameter']['parameterName']
68             ci8 = i['weatherElement'][3]['time'][0]['parameter']['parameterName']
69             pop8 = i['weatherElement'][4]['time'][0]['parameter']['parameterName']
70
71             message = f'{city}未來 8 小時{wx8}，最高溫 {maxt8} 度，最低溫 {mint8} 度，降雨機率 {pop8} %'
72             await ctx.send(message)
73             return
74
75     await ctx.send("抱歉，找不到該城市的天氣資訊")

```

圖十六：天氣預報程式碼



圖十七：實際運行成果

## 9、SWOT 分析

如今市場上有著許多提供著許多不同功能的 DC Bot，因此我們把自己做的 DC Bot 放到市場上做比較，並透過 SWOT 分析並比較他們互相的優劣，正如下表一所示。

表一：SWOT 分析

	自己的機器人	Mee6	Avery
優點	1. 提供加減乘除運算 2. 可播放成員進出 3. 可提供抽籤以及翻譯功能 4. 有小遊戲可遊玩	1. 擁有聊天等級 2. 可透過表情符號來獲得身分	可防止被詐騙
缺點	功能較簡陋	若需要進階功能要付費	提供功能種類較單一

## 五、結論與生活應用

在我們自行開發 DC Bot 的過程中，我們成功地克服了多項技術挑戰，包括語言處理和程式語言撰寫等方面。通過持續學習、發現漏洞並進行修正和優化，我們順利解決了這些問題，同時進一步強化了 DC Bot 的功能。尤其值得一提的是，由於我們的開發方式是自主的，我們能夠根據自身需求製作出符合我們期望的 DC Bot，從而實現了客製化與應用場景的完美結合。例如，我們成功實現了翻譯功能，使得使用者可以方便地在外語和中文之間切換。這種客製化設計賦予了 DC Bot 更大的彈性和適應性，使其更好地滿足各種使用者需求。在未來的發展過程中，我們將持續擴展 DC Bot 的功能，並注重提升使用者的體驗。通過收集用戶反饋並做出相應改進，我們期望 DC Bot 能夠不斷進步，並與使用者需求高度契合。

## 參考資料

smallshawn95 (2023 年 8 月 24 日)。Python Discord Bot 基礎教學。https://hackmd.io/@smallshawn95/python\_discord\_bot\_base

邦卡 (2023 年 8 月 29 日)。Discord 多功能機器人 MEE6 的三個實用免費功能介紹。https://banka.com.tw/three-useful-functions-in-mee6-bot/#t-1686731126053