IOS 應用開發其中作業

姓名：游亮濤 112368402

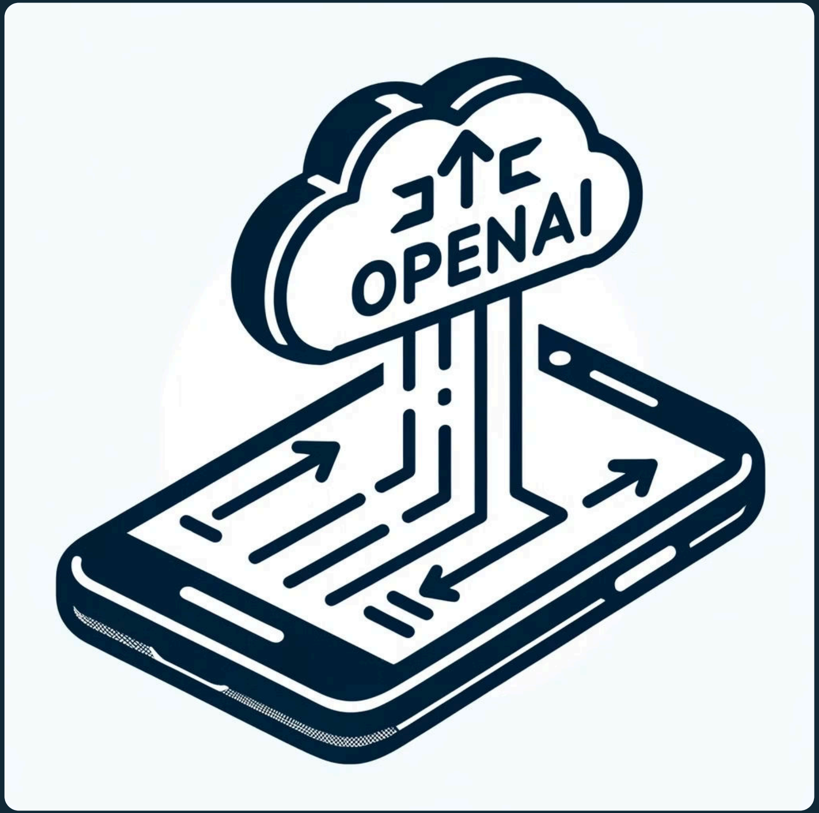
班級：電子碩一

Idea

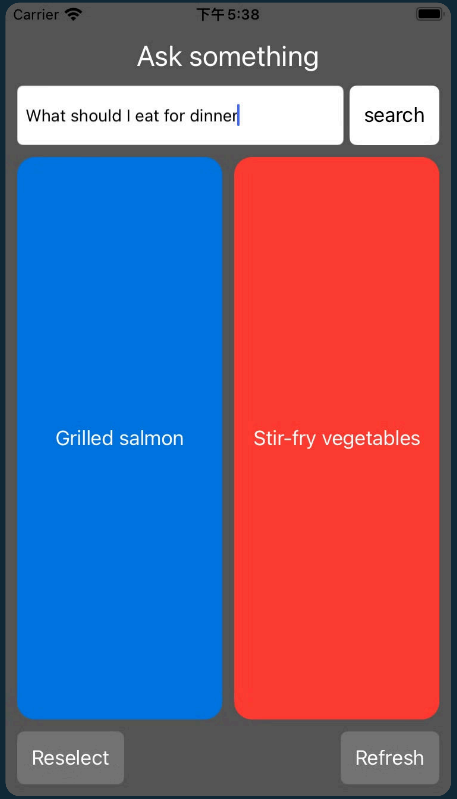
做選擇是很耗費盡力的事情，生活上會有很多瑣事的小決策（如：晚餐吃什麼，要穿什麼衣服）。由於經常被這些生活上的小決策所困擾所以決定開發一個簡易的APP幫助有選擇困難的人。

APP 特點

- 以二選一的方式減少選擇範圍
- 通過OpenAI API取得問題選項
- 沒有語言限制
- 沒有問題限制



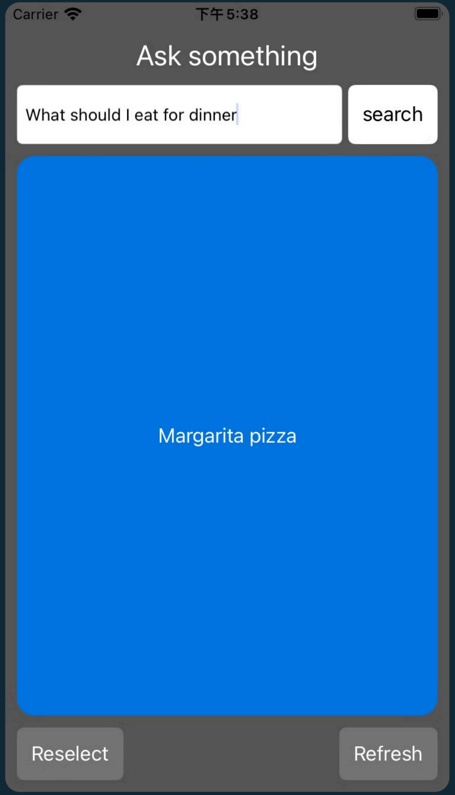
APP 功能



Ask something

What should I eat for dinner

提問問題



Ask something

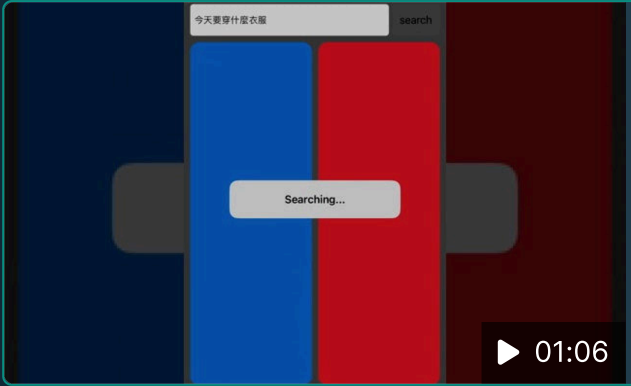
What should I eat for dinner

選擇選項

功能

- 提問問題
- 通過OpenAI API取得10個選項
- 二選一直到選出最終選項
- Reselect - 重新選擇當前全部選項
- Refresh - 清除選項和問題

APP 操作展示

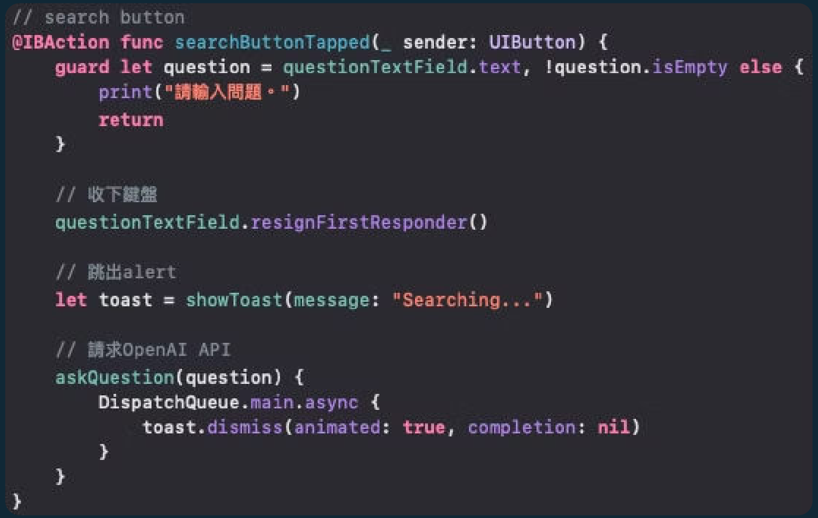


今天要穿什麼衣服

YouTube

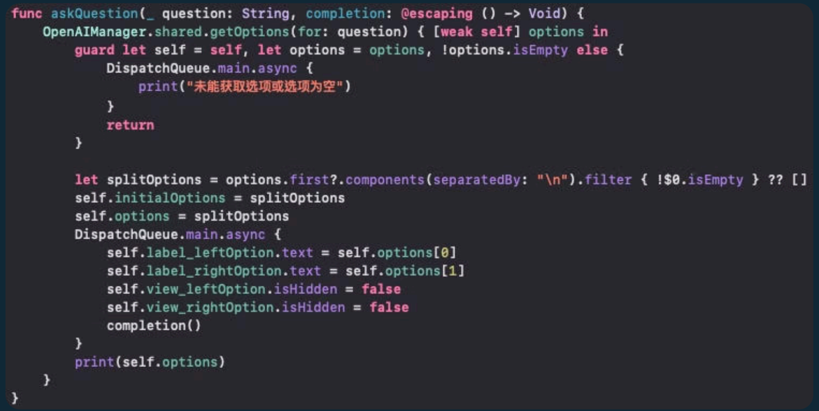
IOS 應用開發期中報告

程式碼



```swift
// search button
@IBAction func searchButtonTapped(_ sender: UIButton) {
    guard let question = questionTextField.text, !question.isEmpty else {
        print("請輸入問題。")
        return
    }

    // 收下鍵盤
    questionTextField.resignFirstResponder()

    // 跳出alert
    let toast = showToast(message: "Searching...")

    // 請求OpenAI API
    askQuestion(question) {
        DispatchQueue.main.async {
            toast.dismiss(animated: true, completion: nil)
        }
    }
}
```

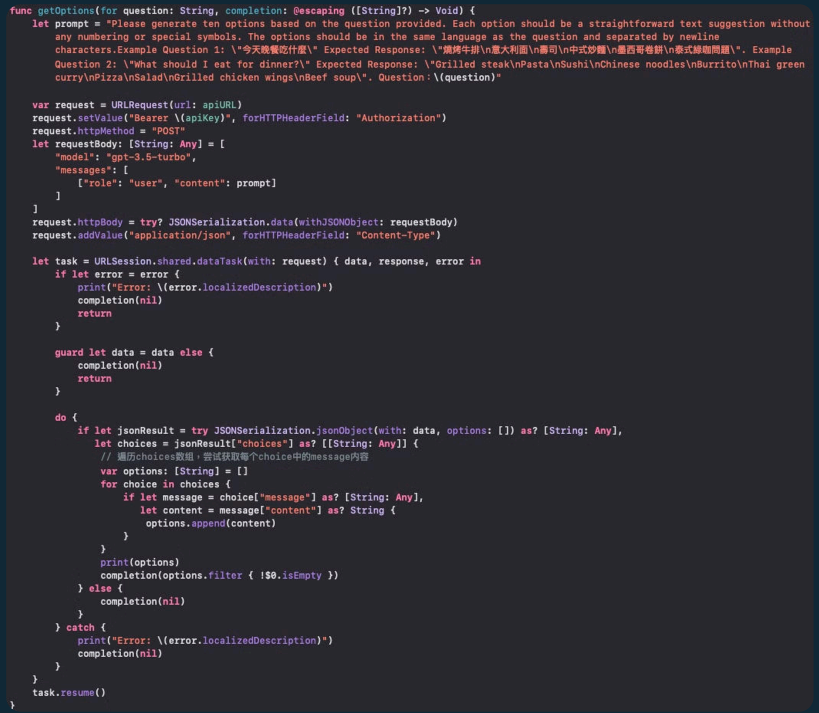
搜尋按鈕



```swift
func askQuestion(_ question: String, completion: @escaping () -> Void) {
    OpenAIManager.shared.getOptions(for: question) { [weak self] options in
        guard let self = self, let options = options, !options.isEmpty else {
            DispatchQueue.main.async {
                print("未能获取选项或选项为空")
            }
            return
        }

        let splitOptions = options.first?.components(separatedBy: "\n").filter { !$0.isEmpty } ?? []
        self.initialOptions = splitOptions
        self.options = splitOptions
        DispatchQueue.main.async {
            self.label_leftOption.text = self.options[0]
            self.label_rightOption.text = self.options[1]
            self.view_leftOption.isHidden = false
            self.view_rightOption.isHidden = false
            completion()
        }
        print(self.options)
    }
}
```

等待API回傳跟新頁面和選項

程式碼

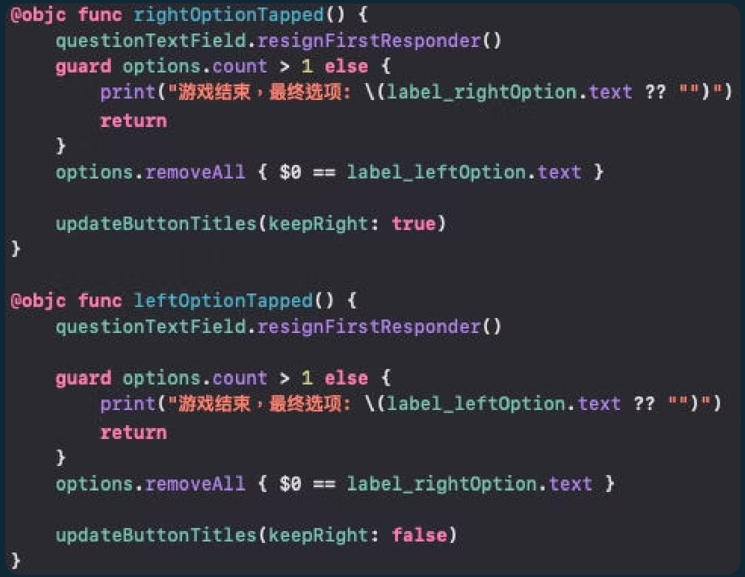


```swift
func getOptions(for question: String, completion: @escaping ([String]?) -> Void) {
    let prompt = "Please generate ten options based on the question provided. Each option should be a straightforward text suggestion without
        any numbering or special symbols. The options should be in the same language as the question and separated by newline
        characters.Example Question 1: \"今天晚餐吃什麼\" Expected Response: \"燒烤牛排\n意大利面\n壽司\n中式炒麵\n墨西哥卷餅\n泰式綠咖問題\". Example
        Question 2: \"What should I eat for dinner?\" Expected Response: \"Grilled steak\nPasta\nSushi\nChinese noodles\nBurrito\nThai green
        curry\nPizza\nSalad\nGrilled chicken wings\nBeef soup\". Question：\(question)"

    var request = URLRequest(url: apiURL)
    request.setValue("Bearer \(apiKey)", forHTTPHeaderField: "Authorization")
    request.httpMethod = "POST"
    let requestBody: [String: Any] = [
        "model": "gpt-3.5-turbo",
        "messages": [
            ["role": "user", "content": prompt]
        ]
    ]
    request.httpBody = try? JSONSerialization.data(withJSONObject: requestBody)
    request.addValue("application/json", forHTTPHeaderField: "Content-Type")

    let task = URLSession.shared.dataTask(with: request) { data, response, error in
        if let error = error {
            print("Error: \(error.localizedDescription)")
            completion(nil)
            return
        }

        guard let data = data else {
            completion(nil)
            return
        }

        do {
            if let jsonResult = try JSONSerialization.jsonObject(with: data, options: []) as? [String: Any],
               let choices = jsonResult["choices"] as? [[String: Any]] {
                // 遍历choices数组，尝试获取每个choice中的message内容
                var options: [String] = []
                for choice in choices {
                    if let message = choice["message"] as? [String: Any],
                       let content = message["content"] as? String {
                        options.append(content)
                    }
                }
                print(options)
                completion(options.filter { !$0.isEmpty })
            } else {
                completion(nil)
            }
        } catch {
            print("Error: \(error.localizedDescription)")
            completion(nil)
        }
    }
    task.resume()
}
```

OpenAIManager

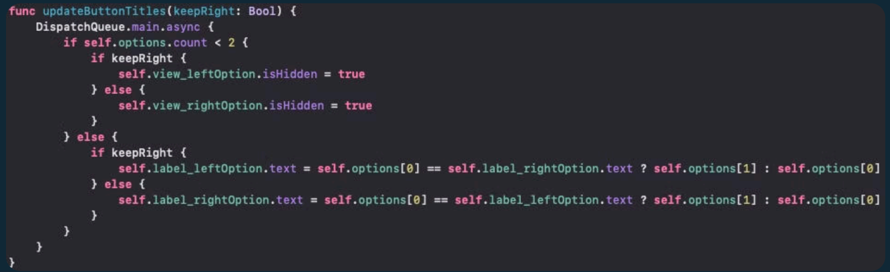
- Prompt instruction
- API reqeust
- API response 反序列化
- 取出content

程式碼



```
@objc func rightOptionTapped() {
    questionTextField.resignFirstResponder()
    guard options.count > 1 else {
        print("游戏结束，最终选项: \(label_rightOption.text ?? "")")
        return
    }
    options.removeAll { $0 == label_leftOption.text }

    updateButtonTitles(keepRight: true)
}

@objc func leftOptionTapped() {
    questionTextField.resignFirstResponder()

    guard options.count > 1 else {
        print("游戏结束，最终选项: \(label_leftOption.text ?? "")")
        return
    }
    options.removeAll { $0 == label_rightOption.text }

    updateButtonTitles(keepRight: false)
}
```

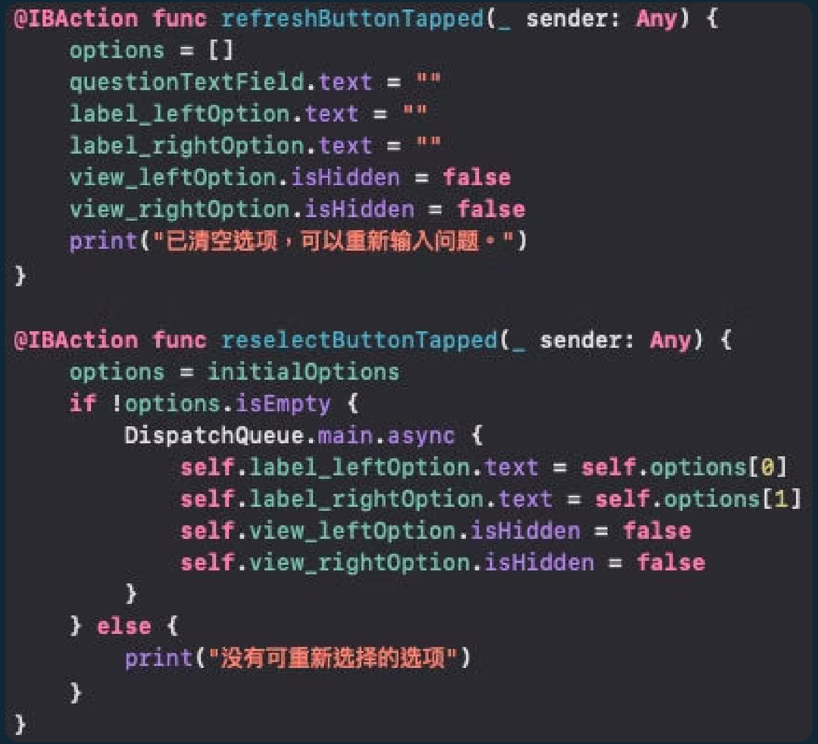
選擇選項



```
func updateButtonTitles(keepRight: Bool) {
    DispatchQueue.main.async {
        if self.options.count < 2 {
            if keepRight {
                self.view_leftOption.isHidden = true
            } else {
                self.view_rightOption.isHidden = true
            }
        } else {
            if keepRight {
                self.label_leftOption.text = self.options[0] == self.label_rightOption.text ? self.options[1] : self.options[0]
            } else {
                self.label_rightOption.text = self.options[0] == self.label_leftOption.text ? self.options[1] : self.options[0]
            }
        }
    }
}
```

更新頁面

程式碼



```swift
@IBAction func refreshButtonTapped(_ sender: Any) {
    options = []
    questionTextField.text = ""
    label_leftOption.text = ""
    label_rightOption.text = ""
    view_leftOption.isHidden = false
    view_rightOption.isHidden = false
    print("已清空选项，可以重新输入问题。")
}

@IBAction func reselectButtonTapped(_ sender: Any) {
    options = initialOptions
    if !options.isEmpty {
        DispatchQueue.main.async {
            self.label_leftOption.text = self.options[0]
            self.label_rightOption.text = self.options[1]
            self.view_leftOption.isHidden = false
            self.view_rightOption.isHidden = false
        }
    } else {
        print("没有可重新选择的选项")
    }
}
```

- 清楚選項和問題

- 重置選項

APP 優化

- 增加Slider 設定選項數量
- 記錄搜尋以便下次快速搜尋
- 記錄喜好請求相似選項