〔様式第４号の１〕

令和５年　3月　18 日

令和５年度　学生自主研究成果報告書

教　育　本　部　長　様

学生自主研究グループ名	Next Revolution	
研究課題名	地震発生を知らせるアプリ作り	
研究代表者（学生）	学籍番号	B25P038
	氏　名	山村蓮
指導教員	学　科	情報工学科
	氏　名	草苅良至

学生自主研究の報告書を別紙のとおり提出します。

〔様式第４号の２〕

地震発生を知らせるアプリ作り

システム科学技術学部　情報工学科　1年 山村蓮
システム科学技術学部　情報工学科　准教授　草苅良至

1. はじめに（研究目的）

私の出身は福島県であり，福島県は，過去 20 年間で震度 5 弱以上の揺れが発生している回数が日本一の県でもある．※1 そのような場所で育ち，3.11 の経験もある私は，地震の発生に特に敏感になった．それは私だけでなく，福島県民や，宮城・岩手県民，更には，“大地震”を経験した地域の住民に共通することであろう．小さな揺れでも気になる中で，自分の周辺地域で起きた地震を即座にスマホに知らせるアプリを作りたいと考えた．スマホの位置情報を活用し，自分のいる位置から半径〇km で発生した地震をセレクトしてプッシュ通知を行い，さらに画面上にそれを表示させるアプリを作ることをゴールに設定した．

※１　気象庁地震・震度データベースによる．

2. 開発方法（環境）

Mac book に，Apple が提供している統合開発環境（IDE）である X Code をインストールし，Apple 社が開発したオープンソースのプログラミング言語である Swift を用いて，開発を行った．私自身，Swift を用いたプログラミングの経験がなかった為，書籍『たった 2 日でマスターできる iPhone アプリ開発集中講座』を参考に，アプリの開発を行った．

また，地震の詳細な情報は，独立研究開発法人防災科学技術研究所（以下　防災科研）がインターネットで公開している，Hi-net　好感度地震観測網から引用した．

3. 実際に開発したアプリ

実際に作成したアプリは以下の通りである．

図１　作成したアプリのアイコン

アプリ名　地震速報
アイコン　右図（図１）
完成したアプリを実際のスマホのホーム画面から見ると
右図(図２)のようになる．

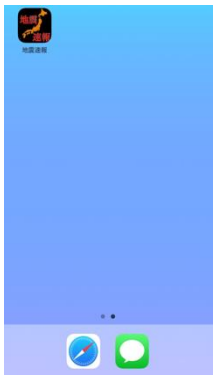

図２ ホーム画面から見たアプリ

なお，ファイルの読み書きをマスターする事に膨大な時間を要し，
設定したゴールには到達できなかった事は初めに記述しておく．

このアプリについて，以下の 5 つの項目に分けて説明する．

（１）　取り扱う問題
（２）　利用する方法
（３）　作成したプログラムと使用法

（4） 作成したプログラムの設計と実装
（5） 作成したプログラムの考察

（1） 取り扱う問題

上記にもある通り，地震の発生を通知するアプリを作る．防災科研が公開している高感度地震観測網 Hi -net には，地震発生日時，震源時誤差，緯度，南北誤差，経度，東西誤差，震源の深さ，深さ誤差，マグニチュードの 8 個のデータがあり，CSV ファイルでダウンロードが可能である．ダウンロードしファイルを読み取り，出力する作業を行う．

（2） 利用する方法

Hi-net でダウンロードしてきた csv ファイルを txt ファイルとして保存し，プロジェクト内に保存する．その後，ファイル読み取りを行う，そして，読み取ったファイルを X Code シュミレータ上の iPhone 上（図 3）に表示する．

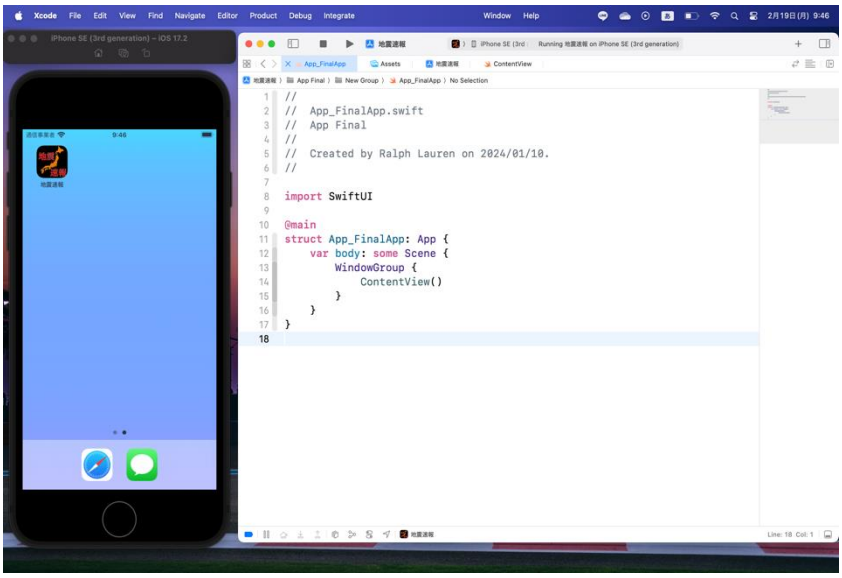

図 3　XCode のシュミレータ（画面左）とプロジェクト画面（画面左）

また実行ボタンを押す毎にファイルを表示させる為，条件分岐を用いた．
用いたプログラムは以下である（

```
        if a==1{
                Text(self.readFromFile())
               }
          }
   Button("実行") {
        a=(a+1)%2
        print("【ファイル内容】\(self.readFromFile())")
   }
```

a が 1 の時，ファイルから読み込んだテキストが表示されることになる．
a は 6 行目の式により，a を 0 と 1 の時で場合わけができる．

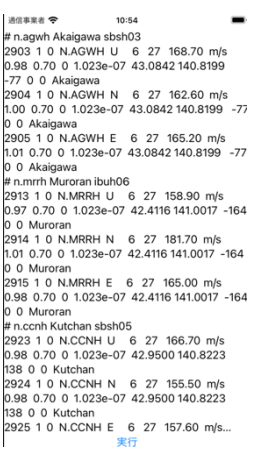

図 4 表示される画面

（3） 作成したプログラムと使用法

アプリの利用方法はシンプルである．まず，アプリを開き，中央にある実行ボタンをクリックする．
そうすると防災科研 Hi-net にある情報が表示される．（図 4 ）

（4）作成したプログラムの設計と実装

```swift
import SwiftUI
struct ContentView: View {
    @State var a=0
    var body: some View {
        VStack{
            if a==1{
                    Text(self.readFromFile())
                   }
             }
        Button("実行") {
            a=(a+1)%2
            print("【ファイル内容】\(self.readFromFile())")
       }
    }

    func writingToFile(text: String) {
        /// ①Documentsフォルダ URL取得
        guard let dirURL = FileManager.default.urls(for: .documentDirectory, in: .userDomainMask).first else {
            fatalError("フォルダURL取得エラー")
        }
        print(dirURL)
        /// ②対象のファイルURL取得
        let fileURL = dirURL.appendingPathComponent("output.txt")

        /// ③ファイルの書き込み
        do {
            try text.write(to: fileURL, atomically: true, encoding: .utf8)
        } catch {
            print("Error: \(error)")
        }

    }

    /// ファイル読み込みサンプル
    func readFromFile() -> String {

        /// ①Documentsフォルダ URL取得
        guard let dirURL = FileManager.default.urls(for: .documentDirectory, in: .userDomainMask).first else {
            fatalError("フォルダURL取得エラー")
        }

        /// ②対象のファイルURL取得
        let fileURL = dirURL.appendingPathComponent("output.txt")

        print(fileURL)

        /// ③ファイルの読み込み
        guard let fileContents = try? String(contentsOf: fileURL) else {
            fatalError("ファイル読み込みエラー")
        }

        /// ④読み込んだ内容を戻り値として返す
        return fileContents
    }
}
```

作成したプログラムは以上の通りである．構造体とエラーメッセージについて説明する．

＜ContentView 構造体＞

・View プロトコルに準拠している．

・@State var a = 0: アプリ内で状態を管理するための a という整数型の State プロパティであり，初期値は 0 である．

・body プロパティ: UI の構造を定義する．VStack を使用し，要素を垂直に配置している．

・Button: “実行“というテキストを表示するボタンがある．ボタンがタップされた時に実行されるクロージャ内で，a の値が変更され，readFromRile（）メソッドが呼び出される．

＜writingToTile(text:)メソッド＞

・引数としてテキストを取り，それをファイルに書き込むメソッドである．

・File Manager を使用し，ドキュメントフォルダの URL を取得し，そのフォルダ内に”output,txt”という名前のファイルを作成する．

・指定されたテキストを UTF-8 エンコーディングでファイルに書き込む．

＜readFromFile()メソッド＞

・ファイルからテキストを読み込むメソッドである．

・File Manager を使用して，ドキュメントフォルダの URL を取得，”output.txt”ファイルの URL を取得する．

・ファイルからテキストを読み取り，その内容を文字列として返す．

＜フォルダ URL 取得エラー＞

FileManager.default.urls(for:in)メソッドは，指定された検索パスに対するディレクトリの ULR を返す．しかし，何らかの理由でディレクトリが見つからない場合，guard 文により，nil が返され，それによりプログラムが fatalError(“フォルダ URL 取得エラー“)の行で停止し，エラーが発生する．例に挙げると，ファイルシステムの問題やアクセス権の不足などが原因として考えられる．

＜ファイルの書き込みエラー＞

try text.write(to:atomically:encording:)メソッドは，指定された URL にテキストを書き込む．その際エラーが発生する可能性がある．例を挙げると，ディスク容量不足，ファイルがロックされているなどが考えられ，エラーが発生した場合は catch ブロックが実行され，エラーメッセージが表示される．

＜ファイル読み込みエラー＞

String(contentsOf:)メソッドは，指定された URL から文字列を読み取る．その際エラーが発生する場合がある．例を挙げると，ファイル自体が存在しない，アクセス権の不足，ファイルが破壊されている場合などである．エラーが発生した場合には，guard ブロック内の fatalError が呼び出され，プログラムが停止し，エラーが表示される．

（５）作成したプログラムの考察

このプログラムは，処理に少し時間が掛かる為，あまり良いプログラムとはいえない．
また，時間の関係で実現できなかった機能（位置情報を利用し自分の周辺の地震を表示する，最新の地震情報を表示，プッシュ通知する　など）を考えると，さらに改善の余地は考えらえる．

4. 研究成果

目標到達点には届かなかったが，発生した地震の情報を表示させる iOS アプリの開発に成功した．

5. まとめ

ファイル読み書きに時間が大幅に掛かってしまった為，目標到達はできなかった．しかし，出来るようになったファイル読み書きを用いて作れるアプリの幅は大きく広がったと思う．今後としては，利便性を向上させた地震速報のアプリ開発と，私が一番作りたいアプリである，チラシの情報を一元化する“FLYER”の開発に取り組んでいきたい．また，iOS と Android どちらにも対応している，Dart 言語の習得を目指し，プログラミングの学習を進めていきたい．

6. 参考文献

・ SwiftUI 対応　たった 2 日でマスターできる iPhone アプリ開発集中講座