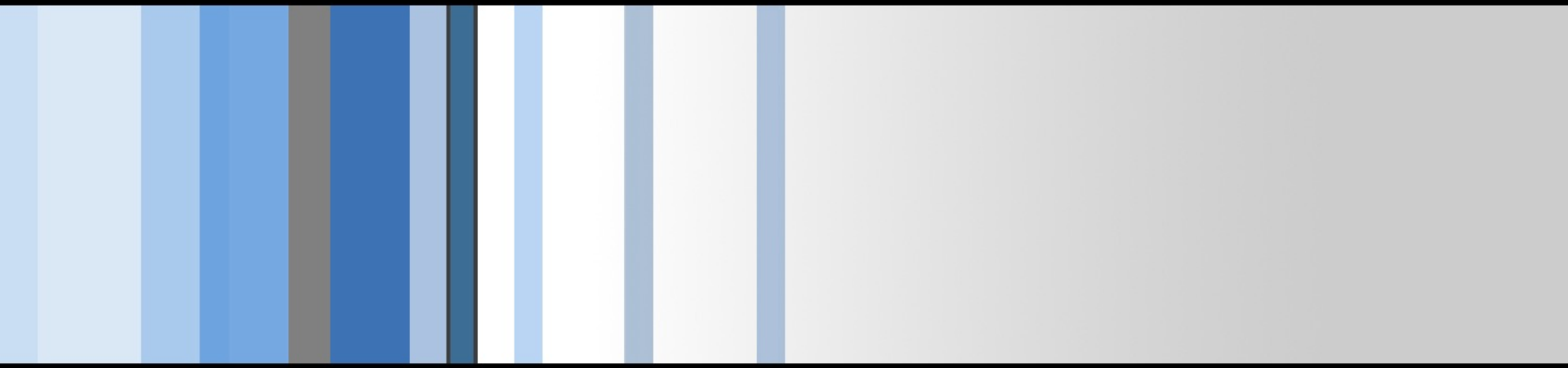
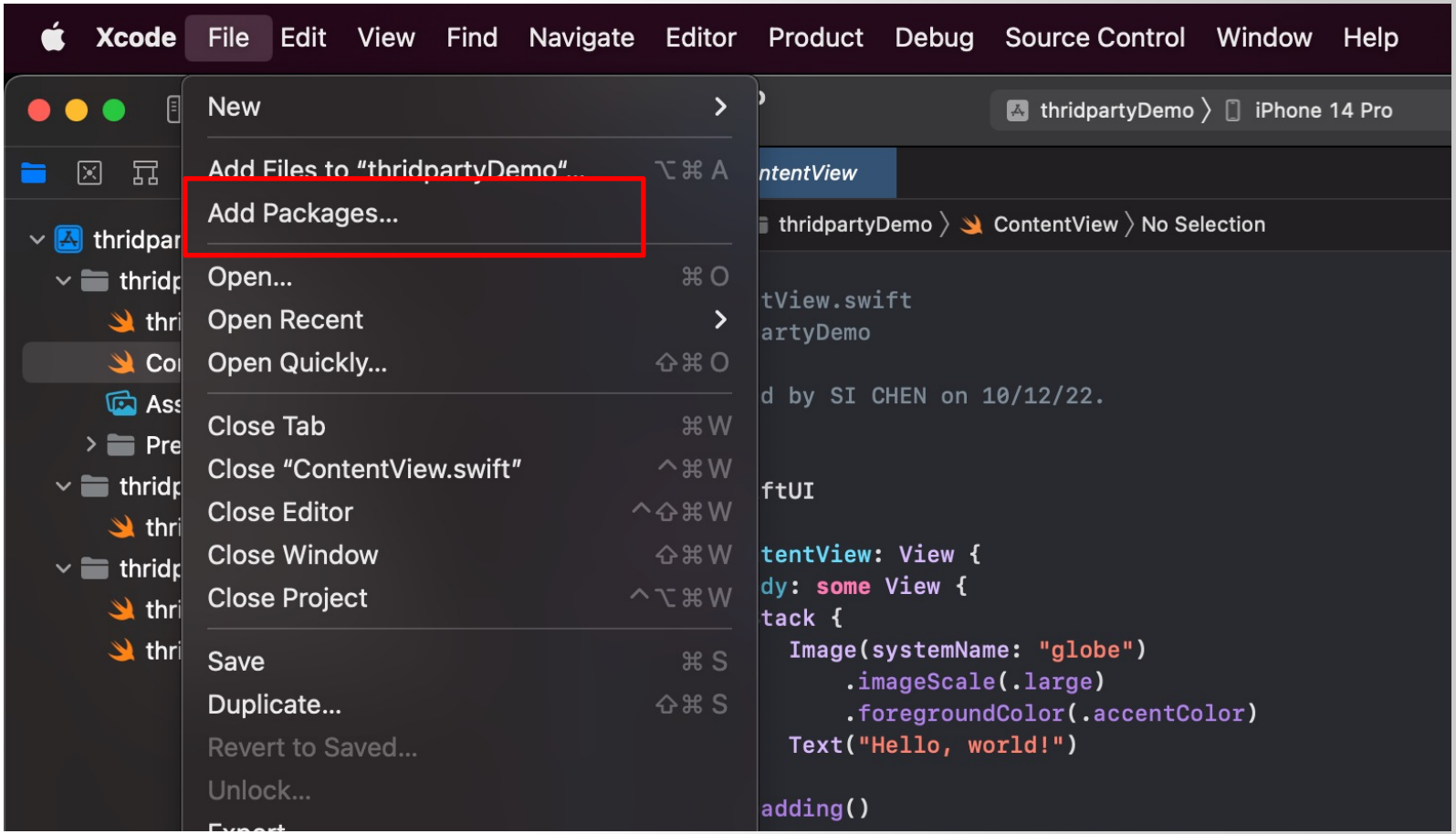
Class 16

CSC 496: iOS App Development
Beyond the Standard Library

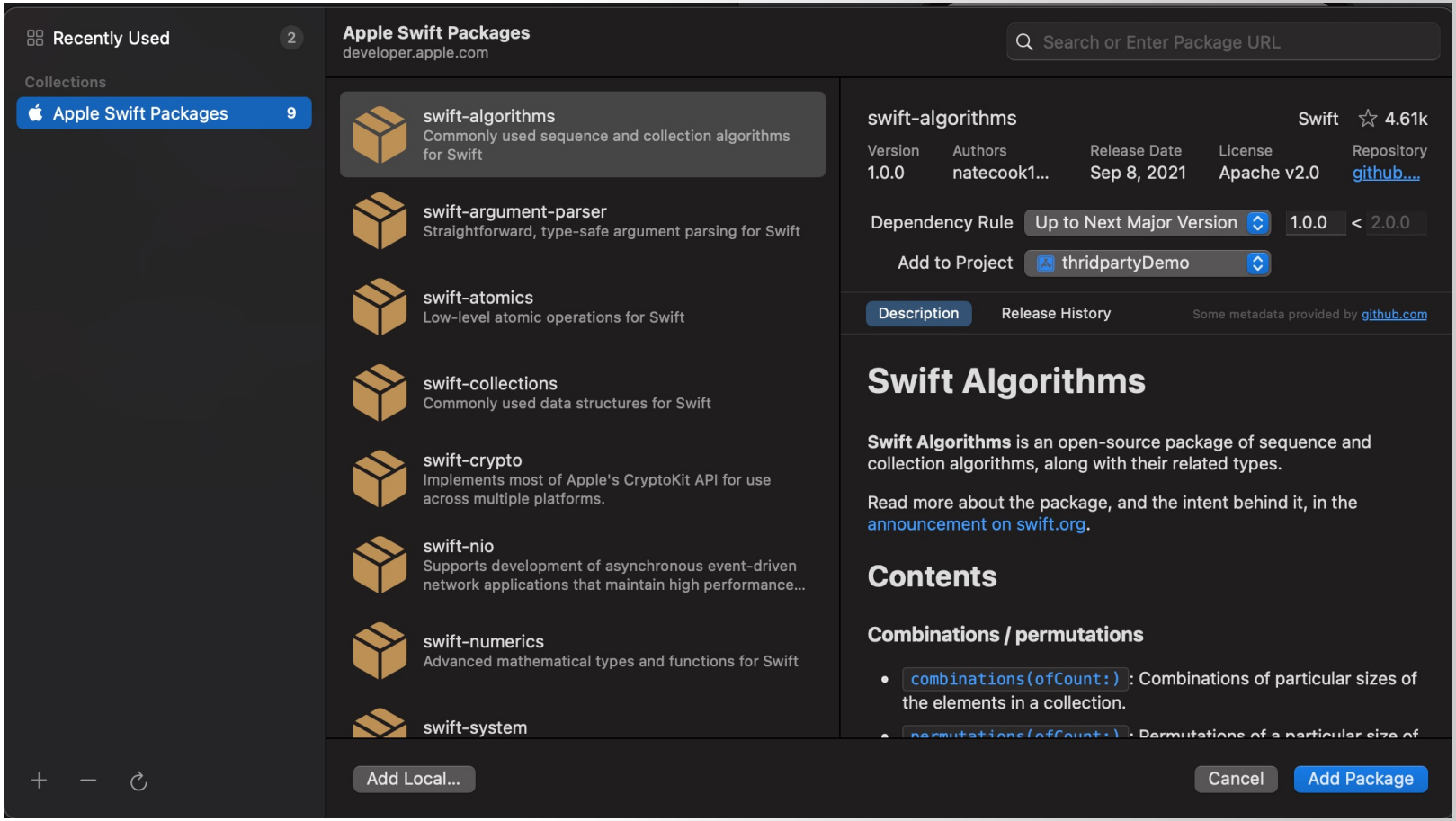
Si Chen (schen@wcupa.edu)



Add Third-party packages



Add Third-party packages



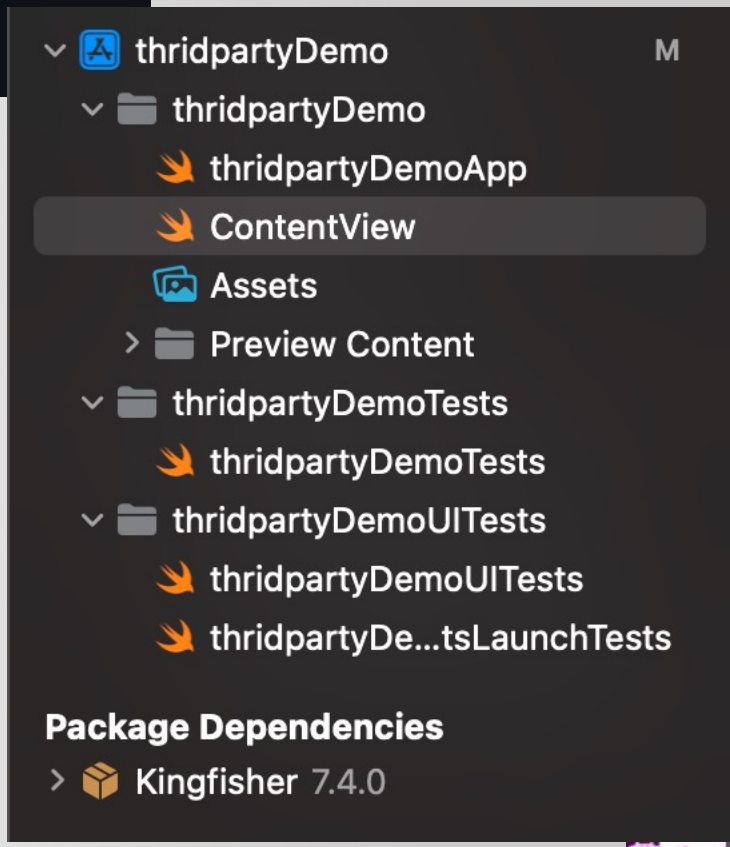
Kingfisher

https://github.com/onevcat/Kingfisher

Kingfisher

Swift Package Manager

- File > Swift Packages > Add Package Dependency
- Add `https://github.com/onevcat/Kingfisher.git`
- Select "Up to Next Major" with "7.0.0"



Kingfisher

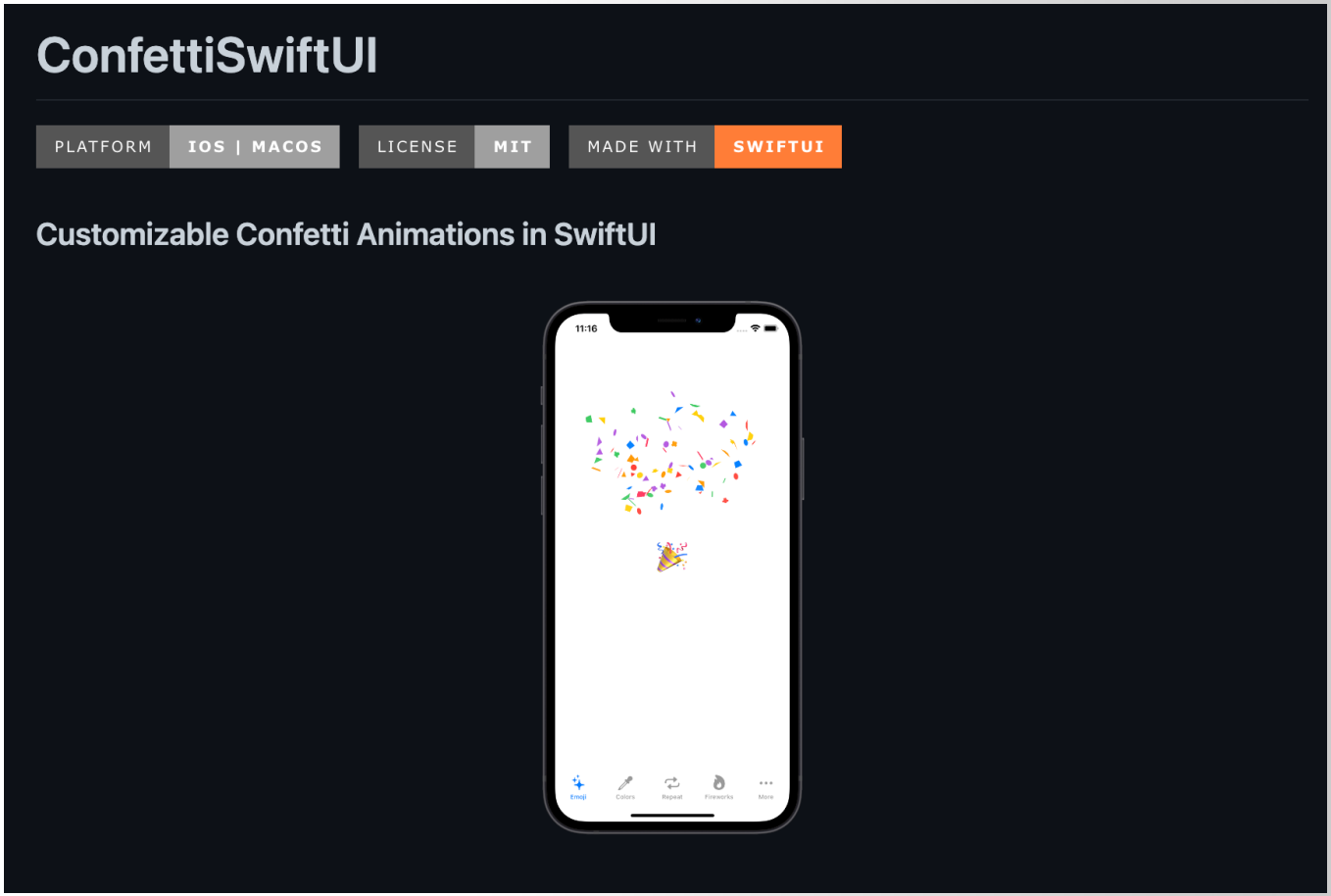
```swift
import SwiftUI
import Kingfisher

struct ContentView: View {
    var body: some View {
        VStack {
            Image(systemName: "globe")
                .imageScale(.large)
                .foregroundColor(.accentColor)
            Text("Hello, world!")
        }
        .padding()

        KFImage(URL(string:
            "https://www.wcupa
            .edu/_resources/includes/brand/images/headerLogo1.png")!)

    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

PokemonAPI

```
import PokemonAPI

struct ContentView: View {
    @State var pokemonName = ""
    @State var pokemonWeight  = 0
    var body: some View {

        Text(pokemonName)
        Text(String(pokemonWeight))
        Button("Click"){
            PokemonAPI().pokemonService.fetchPokemon("150") { result in
                switch result {
                case .success(let pokemon):
                    self.pokemonName = pokemon.name ?? "" // bulbasaur
                    self.pokemonWeight = pokemon.weight ?? 0 //
                        bulbasaur
                case .failure(let error):
                    print(error.localizedDescription)
                }
            }
        }
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

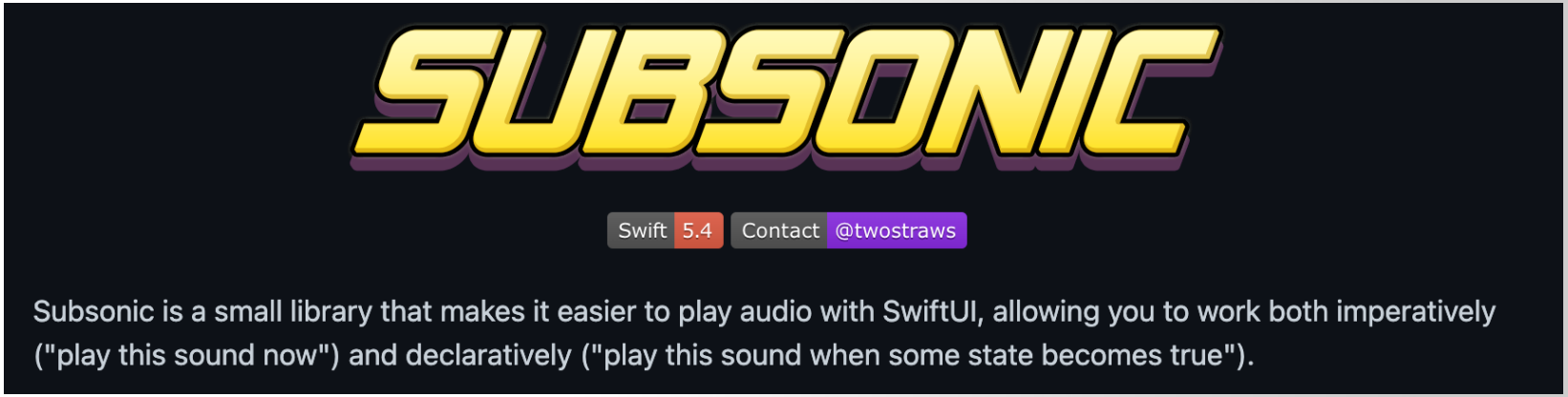
https://github.com/kinkofer/PokemonAPI

ConfettiSwiftUI



https://github.com/simibac/ConfettiSwiftUI

SubSonic



Lab3: Build an app with API and third-party library （Group Project)

- **Assignment Requirements:**

1. **API Selection**: Choose an API for integration from https://github.com/public-apis/public-apis. Alternatively, you may opt for any other API that does not necessitate authentication.
2. **Data Fetching**: Develop an application capable of fetching data from the chosen API. Your application should retrieve at least one property or data field from the API.
3. **User Interface (UI)**: Construct a user interface to display the fetched data. Implement features that allow the user to initiate new data requests. For example, if you select the Weather API, the interface should enable the user to specify a state (like PA or NY) for which to fetch data.
4. **Dynamic Imagery**: Incorporate at least one changeable image in the application's UI. The image should update based on the data retrieved from the API. For instance, the image could turn red if the fetched daily temperatures exceed a certain threshold.
5. **Library Usage**: Utilize **at least one third-party library** in the development of your application.

- **Bonus Criteria:**

- **Asynchronous Programming**: Earn a 2 points bonus if you implement asynchronous programming using async/await for data fetching and handling.

e.g., changeable Image of Weather API

Q & A