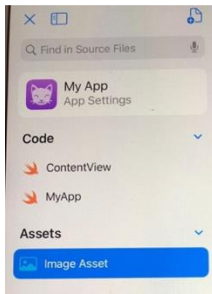# Talking App

1. Upload your own photo image onto Swift Playgrounds

To insert an image, press the icon . The following appears. Press on "Photo" to insert a new image from the Photos Albums.
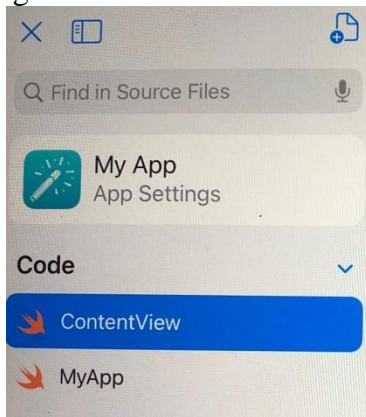
The selected image will then be shown under "Assets" in name of "Image Asset". Your image will be automatically named as "Image Asset". This name will be used in the App.
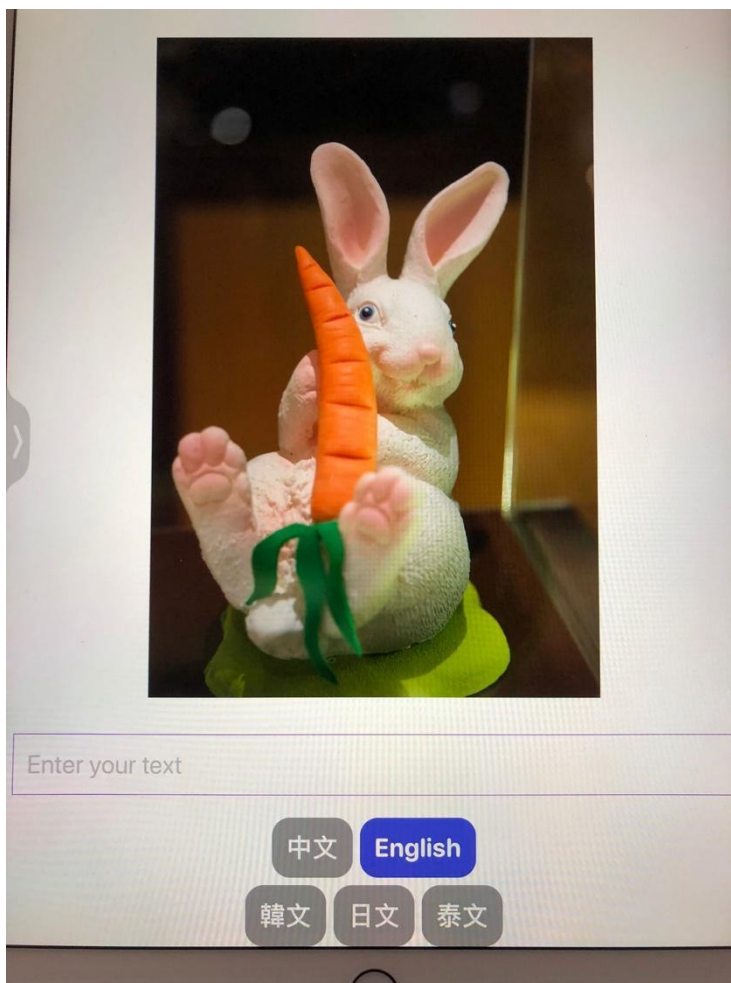


As an example, I use the following image

2. After uploading the image, copy and paste the following program into ContentView replacing all the contents inside.



3. If everything okay, after pressing the play button, the Talking App will appear. You can input your text in any language and listen to how they say in Chinese (Cantonese), English, Korean, Japanese and Thai. It is very interesting to input Emojis like □, □ to see how they say in different languages.

```swift
//
//  ContentView.swift
//  Talking App
//
//  Created by Wing Ching Tam on 28/12/2021.
//

import SwiftUI
import AVFoundation


struct ContentView: View {

    func speak(text: String, language: String) {
        let utterance = AVSpeechUtterance(string: text)
        utterance.voice = AVSpeechSynthesisVoice(language: language)
        let synthesizer = AVSpeechSynthesizer()
        synthesizer.speak(utterance)
    }

    @State var text = ""
    @State var language = "en-AU"
    @State var isFocused: Bool = true

    var body: some View {

        VStack {
            Image("Image Asset")
                .resizable()
                .scaledToFit()
                .padding()

            TextField("Enter your text", text: $text, onCommit: { isFocused = false
            }
            )
                .padding()
                .font(.system(.title, design: .rounded))
                .border(Color.purple, width:1.0)
                .padding()

            Spacer()

            HStack {
                Button {
                    language = "zh-HK"
                    speak(text:text, language: language)
                } label: {
```

```swift
                Text("中文")
                    .fontWeight(.bold)
                    .font(.system(.title, design: .rounded))
            }
            .padding()
            .foregroundColor(.white)
            .background(language == "zh-HK" ?  Color.indigo : Color.gray)
            .cornerRadius(20)
            Button {
                language = "en-AU"
                speak(text:text, language: language)
            } label: {
                Text("English")
                    .fontWeight(.bold)
                    .font(.system(.title, design: .rounded))
            }
            .padding()
            .foregroundColor(.white)
            .background(language == "en-AU" ?  Color.indigo : Color.gray)
            .cornerRadius(20)


        }

        HStack {
            Button {
                language = "ko-KR"
                speak(text:text, language: language)
            } label: {
                Text("韓文")
                    .fontWeight(.bold)
                    .font(.system(.title, design: .rounded))
            }
            .padding()
            .foregroundColor(.white)
            .background(language == "ko-KR" ?  Color.indigo : Color.gray)
            .cornerRadius(20)
            Button {
                language = "ja-JP"
                speak(text:text, language: language)
            } label: {
                Text("日文")
                    .fontWeight(.bold)
                    .font(.system(.title, design: .rounded))
            }
            .padding()
            .foregroundColor(.white)
```

```swift
                    .background(language == "ja-JP" ?  Color.indigo : Color.gray)
                    .cornerRadius(20)

                    Button {
                        language = "th-TH"
                        speak(text:text, language: language)
                    } label: {
                        Text("泰文")
                            .fontWeight(.bold)
                            .font(.system(.title, design: .rounded))
                    }
                    .padding()
                    .foregroundColor(.white)
                    .background(language == "th-TH" ?  Color.indigo : Color.gray)
                    .cornerRadius(20)
                }

            }
        }
}

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