

Pierwsze kroki z SharePlay

Firma Apple wprowadziła SharePlay podczas WWDC 2021, która umożliwia użytkownikom dzielenie się doświadczeniami poprzez integrację aplikacji z FaceTime za pomocą struktury Group Activities. Zaimplementujesz SharePlay dla przykładowej aplikacji, dodając do niej obsługę działań grupowych. Zaczyniesz od nauki działania SharePlay. Następnie zapoznasz się z aplikacją, do której dodasz obsługę SharePlay, korzystając ze struktury działań grupowych. Następnie dowiesz się, jak utworzyć niestandardową aktywność grupową dla tej aplikacji i jak zarządzać sesją aktywności grupowej. Na koniec przetestujesz działanie SharePlay w aplikacji na dwóch urządzeniach z systemem iOS. Pod koniec dowiesz się, jak działa SharePlay i jak aktualizować własne aplikacje, aby z niego korzystać. Omówione zostaną następujące tematy:

- Zrozumienie SharePlay
- Poznawanie aplikacji ShareOrder
- Tworzenie niestandardowej aktywności grupowej
- Zarządzanie sesją zajęć grupowych
- Testowanie SharePlay w aplikacji ShareOrder

Zrozumienie SharePlay

SharePlay został wprowadzony przez Apple podczas WWDC 2021. Umożliwia współdzielenie doświadczeń użytkownika dla uczestników sesji FaceTime. Na przykład użytkownik może chcieć oglądać wideo razem z innym użytkownikiem. Wszystko, co użytkownik musi zrobić, to FaceTime z innym użytkownikiem, uruchomić aplikację wideo i zainicjować SharePlay. Ta sama aplikacja zostanie uruchomiona dla drugiego użytkownika i odtworzy to samo wideo, a SharePlay zapewnia synchronizację wideo między oboma użytkownikami. Możesz także tworzyć niestandardowe doświadczenia SharePlay. Przykładem tego jest aplikacja DrawTogether zademonstrowana podczas WWDC 2021. Podczas demonstracji trzech użytkowników początkowo dołączyło do sesji FaceTime. Jeden użytkownik uruchomił aplikację DrawTogether i zainicjował sesję SharePlay w aplikacji. Inni użytkownicy otrzymali monit SharePlay zawierający przycisk Dołącz. Po naciśnięciu przycisku Dołącz, aplikacja DrawTogether została uruchomiona dla innych użytkowników, a wszystko, co użytkownik narysował na swoim ekranie, pojawiło się na ekranach innych użytkowników. SharePlay jest zasilany przez ramy działań grupowych. Ta struktura wykorzystuje FaceTime do synchronizowania działań w aplikacji i zapraszania innych uczestników do dołączenia do tych działań. Obiekty reprezentujące wspólne działania muszą być zgodne z protokołem GroupActivity. Po rozpoczęciu działania grupy obiekt GroupSession jest używany do synchronizowania zachowania aplikacji między wszystkimi uczestnikami. Tu zaimplementujesz strukturę działań grupowych w przykładowej aplikacji o nazwie ShareOrder, wykonując następujące kroki:

1. Dodaj funkcję Działania grupowe do aplikacji ShareOrder.
2. Utwórz i skonfiguruj strukturę ShareOrder zgodną z protokołem GroupActivity. Ta struktura będzie zawierać metadane opisujące aktywność grupy.
3. Skonfiguruj interfejs użytkownika aplikacji ShareOrder za pomocą przycisku, aby aktywować aktywność grupy.
4. Zaimplementuj obiekt GroupSession, który umożliwia aplikacji dołączenie do sesji aktywności grupowej.

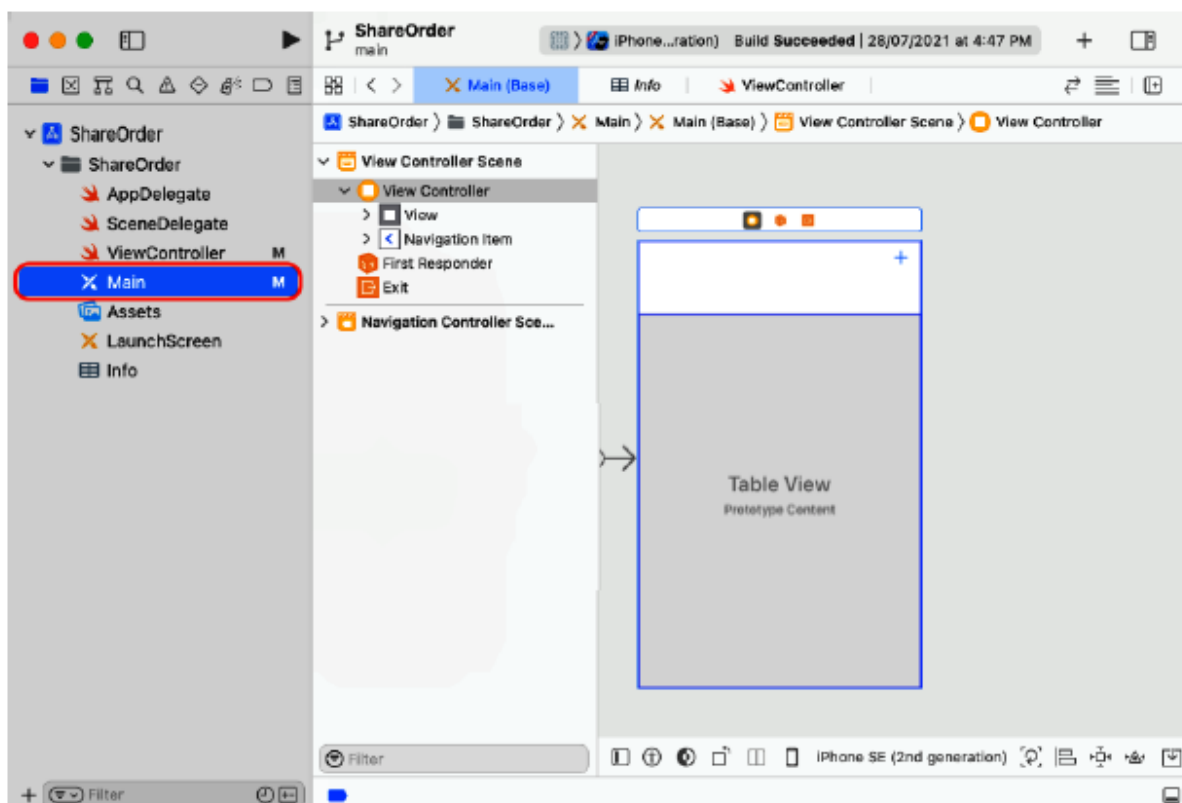
5. Zaimplementuj obiekt GroupSessionMessenger, który umożliwia aplikacji wysyłanie i odbieranie komunikatów. Te wiadomości służą do synchronizowania czynności wykonywanych przez użytkowników w aplikacji.

Zanim to zrobisz, zobaczmy, jak działa aplikacja ShareOrder w następnej sekcji.

Poznanie aplikacji ShareOrder

Aplikacja, nad którą będziesz pracować, ShareOrder, to prosta aplikacja, która rejestruje i wyświetla to, co chcesz zamówić w restauracji. Zbudujmy i uruchom tę aplikację, aby zobaczyć, jak to działa. Wykonaj następujące kroki:

1. Jeśli jeszcze tego nie zrobiłeś, pobierz folder Chapter25 pakietu kodu dla tej książki pod tym linkiem: <https://github.com/PacktPublishing/iOS-15-Programming-for-Beginners-Sixth-Edition>.
2. Otwórz folder Chapter25, a zobaczysz dwa foldery, ShareOrder-start i ShareOrder-complete. Pierwszy folder zawiera aplikację, którą będziesz modyfikować podczas tej lekcji, a drugi zawiera ukończoną aplikację.
3. Otwórz folder startowy ShareOrder i otwórz projekt ShareOrder Xcode. Kliknij plik głównego scenorysu w nawigatory projektu. Powinieneś zobaczyć przycisk + na pasku nawigacyjnym i widok tabeli wypełniający resztę ekranu.



Po uruchomieniu aplikacja wyświetli ekran z pustym widokiem tabeli. Naciśnięcie przycisku + spowoduje wyświetlenie okna dialogowego, w którym można wprowadzić zamówienie, które następnie pojawi się w widoku tabeli.

4. Kliknij plik ViewController w nawigatory projektów. Powinieneś zobaczyć następujący kod w obszarze edytora:

```
import UIKit

class ViewController: UIViewController {

var orders: [String] = []

@IBOutlet var tableView: UITableView!

override func viewDidLoad() {
super.viewDidLoad()

title = "ShareOrder"

tableView.register(UITableViewCell.self,
forCellReuseIdentifier: "orderCell")
}

@IBAction func addOrder(_ sender: UIBarButtonItem)
{
let alert = UIAlertController(title: "New
Order", message: "Add a new order",
preferredStyle: .alert)

let saveAction = UIAlertAction(title: "Save",
style: .default) {
[unowned self] action in
guard let textField =
alert.textFields?.first,
let orderToSave = textField.text else {
return
}

self.orders.append(orderToSave)

self.tableView.reloadData()
}

let cancelAction = UIAlertAction(title:
"Cancel", style: .cancel)

alert.addTextField()

alert.addAction(saveAction)

alert.addAction(cancelAction)
```

```

present(alert, animated: true)
}
}

extension ViewController: UITableViewDataSource {
func tableView(_ tableView: UITableView,
numberOfRowsInSection section: Int) -> Int {
orders.count
}

func tableView(_ tableView: UITableView,
cellForRowAt indexPath: IndexPath) ->
UITableViewCell {
let cell = tableView.dequeueReusableCell
(withIdentifier: "orderCell", for: indexPath)
cell.textLabel?.text = orders[indexPath.row]
return cell
}
}

```

Rozbijmy to:

```
var orders: [String] = []
```

Ta właściwość zawiera tablicę zamówień, które są typu String. Ta tablica będzie źródłem danych dla widoku tabeli w aplikacji.

```
@IBOutlet var tableView: UITableView!
```

To gniazdo jest połączone z widokiem tabeli w scenie kontrolera widoku w pliku głównego scenorysu.

viewDidLoad() method:

Kod w tej metodzie ustawia tytuł na pasku nawigacyjnym na ShareOrder i rejestruje identyfikator ponownego użycia orderCell dla komórek widoku tabeli.

addOrder(_:) method:

Ta metoda jest połączona z przyciskiem + na pasku nawigacyjnym. Tworzy alert z polem tekstowym i dwoma przyciskami, Zapisz i Anuluj. Dotknięcie przycisku + spowoduje wyświetlenie alertu. Następnie możesz wpisać swoje zamówienie w polu tekstowym. Kliknięcie Zapisz doda zamówienie do tablicy zamówień i ponownie załaduje widok tabeli. Dotknięcie Anuluj spowoduje odrzucenie alertu.

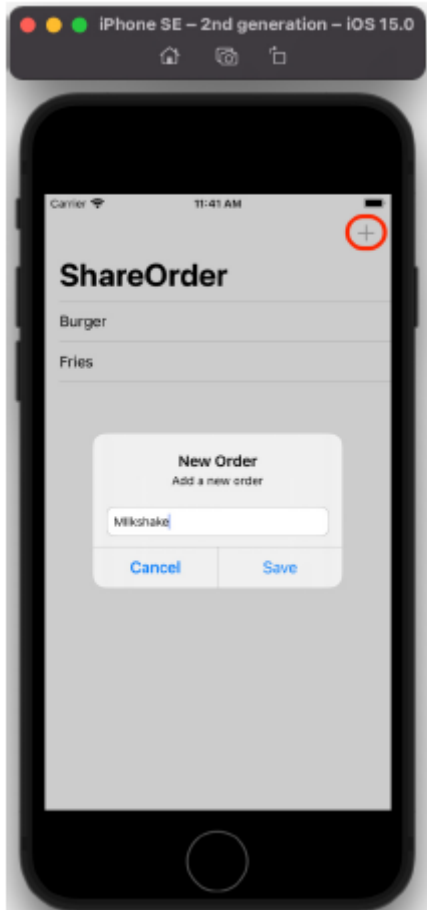
```
extension ViewController: UITableViewDataSource { ...
}

```

To rozszerzenie zawiera metody źródła danych dla widoku tabeli. Liczba wierszy do wyświetlenia jest taka sama jak liczba zamówień w tablicy orders. Każda komórka w widoku tabeli wyświetli odpowiedni ciąg w tablicy zamówień.

To rozszerzenie zawiera metody źródła danych dla widoku tabeli. Liczba wierszy do wyświetlenia jest taka sama jak liczba zamówień w tablicy orders. Każda komórka w widoku tabeli wyświetli odpowiedni ciąg w tablicy orders.

Zbuduj i uruchom aplikację. Dotknij przycisku +, wprowadź tekst w polu tekstowym alertu i dotknij Zapisz. Pojawi się w widoku tabeli, jak pokazano poniżej:



Teraz, gdy znasz już aplikację ShareOrder i sposób jej działania, dodasz do niej obsługę działań grupowych. Po wykonaniu tej czynności dodanie zamówienia podczas sesji SharePlay spowoduje, że zamówienie pojawi się na ekranie wszystkich uczestników. Zaczniemy od utworzenia niestandardowej aktywności grupowej dla aplikacji ShareOrder w następnym sekcji.

Tworzenie niestandardowej aktywności grupowej

Widziałeś, że aplikacja ShareOrder umożliwia dodawanie zleceń, które będą wyświetlane na ekranie. Dodasz aktywność grupową dla tej aplikacji, która pozwoli uczestnikom dodawać zamówienia podczas sesji SharePlay, która pojawi się na ekranie każdego uczestnika. Do reprezentowania tego działania wymagany jest obiekt niestandardowy. Kroki wymagane do realizacji tego są następujące:

- Dodaj uprawnienie Działania grupowe do aplikacji ShareOrder.
- Utwórz nową strukturę o nazwie ShareOrder, która jest zgodna z

Protokół GroupActivity i skonfiguruj metadane aktywności grupy.

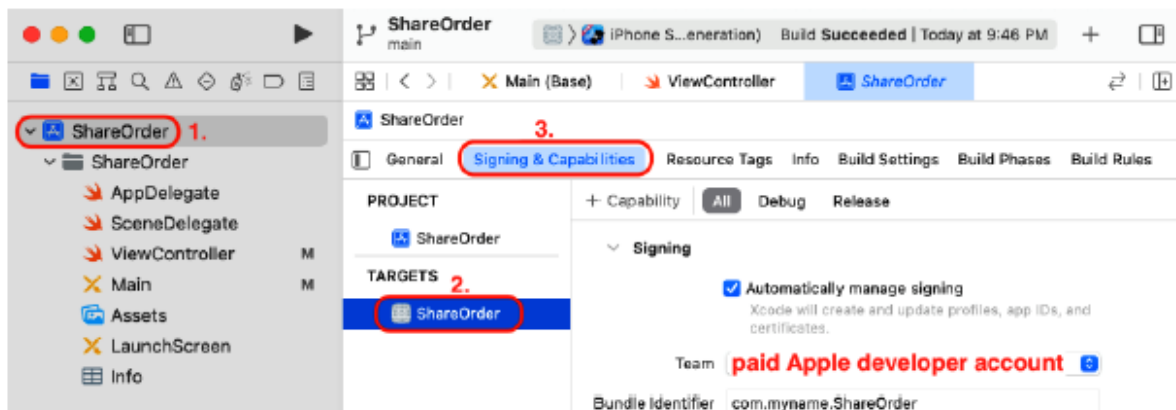
- Dodaj przycisk do interfejsu użytkownika aplikacji ShareOrder i dodaj akcję dla tego przycisku, aby aktywować aktywność grupy.

Zacznijmy od dodania uprawnienia Działania grupowe do aplikacji ShareOrder w następnym sekcji.

Dodawanie uprawnienia do Działań Grupowych

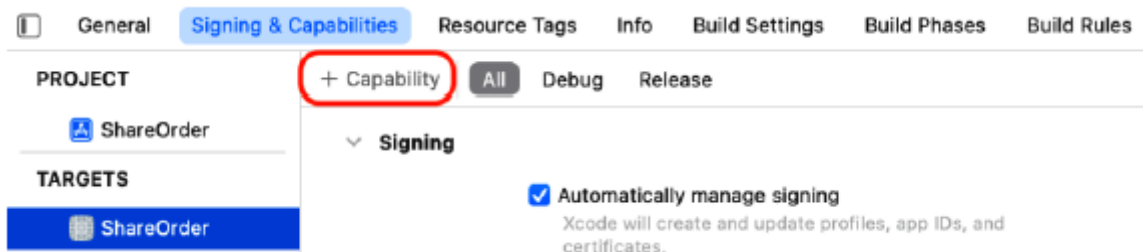
Ponieważ Twoja aplikacja będzie wchodzić w interakcje między różnymi urządzeniami, musi mieć uprawnienie `com.apple.developer.group-session`. Użyjesz Xcode, aby dodać to uprawnienie do swojej aplikacji. Pamiętaj, że potrzebujesz do tego płatnego konta programisty Apple. Wykonaj następujące kroki:

1. Kliknij projekt ShareOrder (najwyższy element w nawigacji projektów) i kliknij cel ShareOrder. Wybierz okienko Podpisywanie i możliwości w obszarze Edytor. Sprawdź, czy Team został ustawiony na płatne konto programisty Apple.

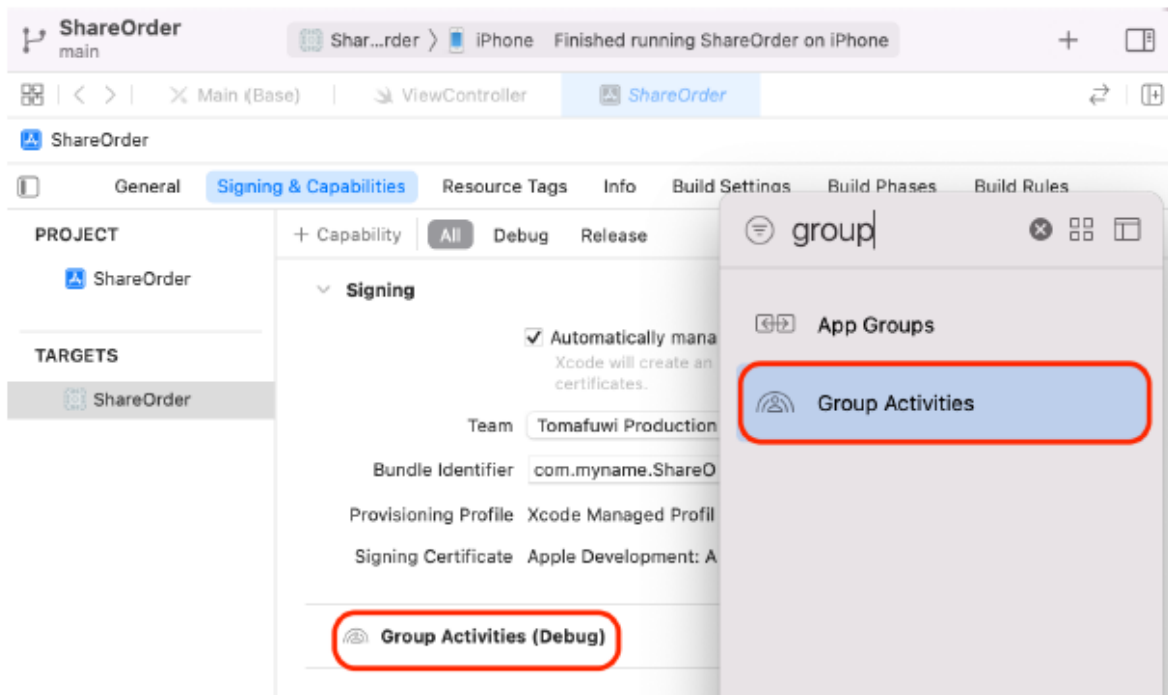


2. Podłącz jedno ze swoich urządzeń iOS z systemem iOS 15 i sprawdź, czy aplikacja ShareOrder może działać na Twoim urządzeniu.

3. W panelu Podpisywanie i możliwości kliknij przycisk +, aby dodać możliwość do aplikacji ShareOrder.



4. W wyświetlonym oknie wyszukaj i dwukrotnie kliknij opcję Działania grupowe, aby dodać ją do aplikacji ShareOrder:



Dodano uprawnienie Działania grupowe do aplikacji ShareOrder. Następnie utworzysz strukturę ShareOrder, która będzie reprezentować aktywność grupy w następnej sekcji.

Tworzenie struktury ShareOrder

Aktywność, którą zrealizujesz w aplikacji ShareOrder, pozwoli na wyświetlanie na ekranie wszystkich uczestników zleceń dodanych podczas sesji SharePlay. Potrzebujesz obiektu do reprezentowania tej działalności. Zaimplementujesz ten obiekt, tworząc strukturę ShareOrder. Ta struktura będzie zgodna z protokołem GroupActivity i będzie zawierać metadane opisujące aktywność. Aby utworzyć i skonfigurować strukturę ShareOrder, wykonaj następujące kroki:

1. Kliknij plik ViewController w nawigatorze projektów i zaimportuj strukturę GroupActivities, jak pokazano:

```
import UIKit
```

```
import GroupActivities
```

2. Dodaj następujące rozszerzenie po ostatnim nawiasie klamrowym, aby utworzyć i skonfigurować strukturę ShareOrder:

```
extension ViewController {
    struct ShareOrder: GroupActivity {
        var metadata: GroupActivityMetadata {
            var metadata = GroupActivityMetadata()
            metadata.title = NSLocalizedString("Share
            Order", comment: "Title of group activity")
            metadata.type = .generic
        }
    }
}
```

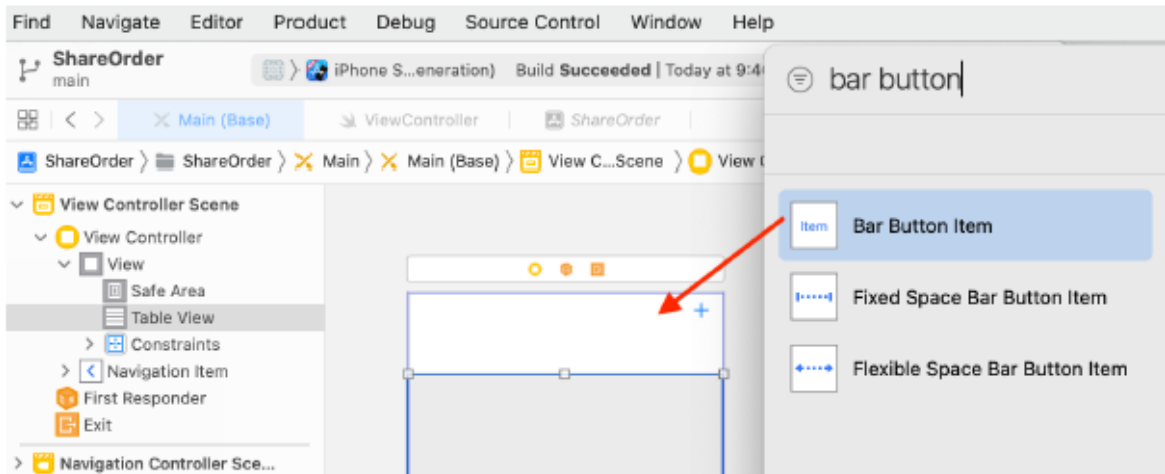
```
return metadata
}
}
}
```

Ta struktura reprezentuje środowisko udostępniania aplikacji ShareOrder. Jest zgodny z protokołem GroupActivity, który zapewnia kontekst i metadane do rozpoczęcia sesji związanej z aktywnością. Tutaj używasz obliczonej właściwości o nazwie metadata, aby ustawić tytuł metadanych i komentarz. W przypadku działań niestandardowych typ metadanych jest ustawiony na .generic. Utworzyłeś i skonfigurowałeś strukturę ShareOrder. W następnej sekcji dodasz przycisk do interfejsu użytkownika aplikacji ShareOrder i skonfigurujesz ten przycisk, aby aktywować aktywność grupy podczas sesji FaceTime.

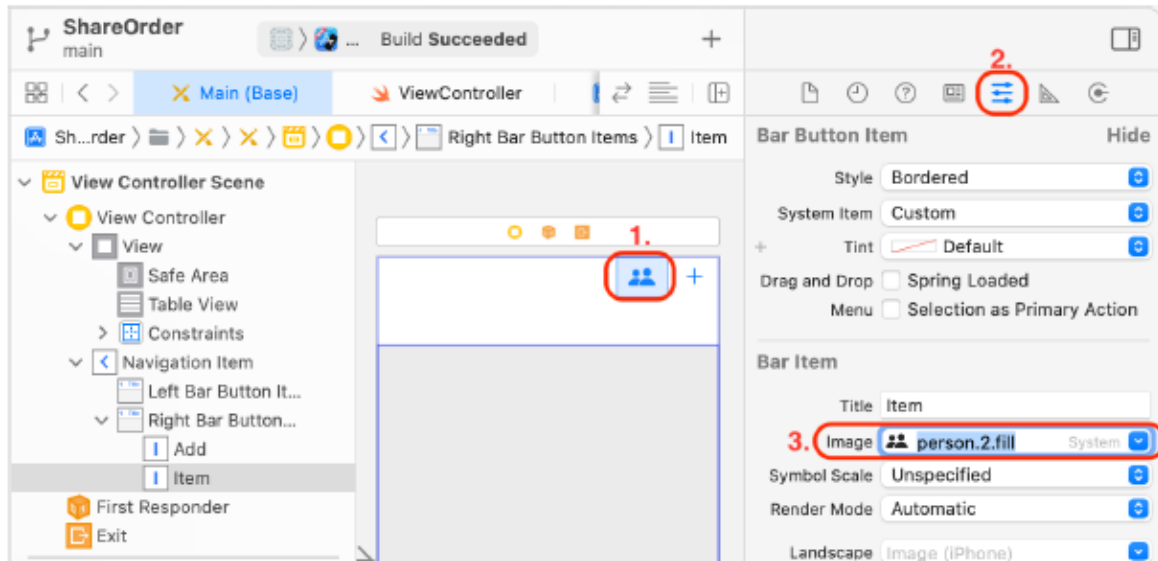
Aktywacja niestandardowej aktywności grupowej

Utworzyłeś i skonfigurowałeś strukturę ShareOrder używaną do reprezentowania niestandardowej aktywności grupy dla aplikacji ShareOrder. Teraz dodasz przycisk do paska nawigacyjnego aplikacji ShareOrder, aby aktywować tę aktywność grupową podczas sesji FaceTime z innymi uczestnikami. Wykonaj następujące kroki:

1. Kliknij plik głównego scenorysu w nawigatorze projektów i kliknij przycisk Biblioteka.
2. W polu filtra Biblioteki wyszukaj obiekt Element przycisku paska i przeciągnij go na pasek nawigacyjny obok przycisku +:



3. Po zaznaczeniu elementu przycisku paska kliknij przycisk Inspektora atrybutów. W polu Element paska ustaw Obraz na person.2.fill:



Ustawia to ikonę elementu przycisku paska.

4. Kliknij przycisk Dostosuj opcje edytora i wybierz Asystent, aby wyświetlić edytor asystenta. Zawartość pliku ViewController pojawi się w edytorze asystentów.

5. Control + przeciągnij z elementu przycisku paska, który właśnie dodałeś, tuż przed metodą `addOrder(_:)` w edytorze asystentów. Pojawi się wyskakujące okno dialogowe.

6. W polu Nazwa wpisz nazwę `ActivateGroupActivity` i kliknij Połącz:



7. Dodaj następujący kod do metody `ActivateGroupActivity(_:)`, aby aktywować aktywność grupową po dotknięciu elementu przycisku paska:

```

@IBAction func activateGroupActivity(_ sender: Any) {
Task {
do {
try await ShareOrder().activate()
} catch {
print("Unable to activate")
}
}

```

```
}
```

```
}
```

Spowoduje to wyświetlenie monitu SharePlay z przyciskiem Dołącz wszystkim uczestnikom sesji FaceTime.

8. Kliknij przycisk x, aby zamknąć edytor asystenta.

Właśnie dodałeś przycisk, który po dotknięciu aktywuje twoją aktywność grupową. W następnej sekcji dowiesz się, jak zarządzać sesją aktywności grupowej w aplikacji ShareOrder.

Zarządzanie sesją zajęć grupowych

Utworzyłeś strukturę ShareOrder, która reprezentuje aktywność grupy dla aplikacji ShareOrder, i dodałeś przycisk do paska nawigacyjnego aplikacji, aby aktywować aktywność grupy podczas sesji FaceTime. Teraz musisz dodać kod, aby umożliwić uczestnikom dołączenie do tej sesji aktywności grupowej i zapewnić wzajemną synchronizację wszystkich uczestników. Kroki wymagane do realizacji tego są następujące:

- Utwórz obiekt GroupSession, który pozwoli aplikacji dołączyć do sesji aktywności grupowej.
- Utwórz obiekt GroupSessionMessenger, który umożliwi aplikacji wysyłanie i odbieranie wiadomości w celu synchronizowania zawartości.

Zobaczmy, jak zaimplementować i skonfigurować obiekt GroupSession dla aplikacji ShareOrder w następnej sekcji.

Implementacja obiektu GroupSession

Aplikacja ShareOrder ma obecnie obiekt reprezentujący aktywność grupy oraz przycisk, który służy do aktywowania aktywności grupy podczas sesji FaceTime. Dodasz kod do swojej aplikacji, aby zaimplementować obiekt GroupSession, który umożliwi uczestnikom dołączenie do sesji aktywności grupowej. Aby zaimplementować i skonfigurować obiekt GroupSession dla aplikacji ShareOrder, wykonaj następujące kroki:

1. Kliknij plik ViewController w nawigаторze projektów. Dodaj opcjonalną właściwość po właściwości zamówienia, aby przechowywać wystąpienie sesji aktywności grupowej:

```
var orders: [String] = []
```

```
var groupSession: GroupSession<ShareOrder>?
```

2. Dodaj następujący kod do viewDidLoad(), aby utworzyć asynchroniczne zadanie odbierania sesji grupowej:

```
override func viewDidLoad() {
```

```
super.viewDidLoad()
```

```
title = "ShareOrder"
```

```
tableView.register(UITableViewCell.self,
```

```
forCellReuseIdentifier: "orderCell")
```

```
Task {
```

```

for await session in ShareOrder.sessions() {
    configureGroupSession(session)
}
}
}

```

Zobaczysz błąd, ponieważ `configureGroupSession(_:)` nie został jeszcze zaimplementowany.

3. Dodaj następujący kod do rozszerzenia po definicji struktury `ShareOrder`, aby zaimplementować metodę `configureGroupSession(_:)`:

```

func configureGroupSession(_ groupSession:
    GroupSession<ShareOrder>) {
    orders.removeAll()
    self.groupSession = groupSession
}

```

Ta metoda usuwa wszystkie zamówienia z tablicy `orders`, a następnie przypisuje odebraną sesję aktywności grupowej do właściwości `groupSession`.

4. Zmodyfikuj metodę `configureGroupSession(_:)` dodając kod, aby dołączyć do sesji aktywności grupowej:

```

func configureGroupSession(_ groupSession:
    GroupSession<ShareOrder>){
    orders.removeAll()
    self.groupSession = groupSession
    groupSession.join()
}

```

Możesz wybrać produkt | Skompiluj z paska menu Xcode, aby sprawdzić, czy w tym momencie nie ma błędów w projekcie `ShareOrder`. Po zaimplementowaniu i skonfigurowaniu obiektu `GroupSession` zaimplementujesz i skonfigurujesz obiekt `GroupSessionMessenger` dla aplikacji `ShareOrder` w następnej sekcji.

Implementacja obiektu `GroupSessionMessenger`

Gdy użytkownicy aplikacji `ShareOrder` dołączą do sesji `SharePlay`, wszelkie zamówienia, które dodadzą do aplikacji, pojawią się na wszystkich ekranach. Ta synchronizacja jest obsługiwana przez obiekt `GroupSessionMessenger`, który może wysyłać komunikaty do i odbierać komunikaty z innych urządzeń w sesji aktywności grupy. Pamiętaj, że ważne jest, aby rozmiar wiadomości był niewielki, ponieważ duże rozmiary wiadomości mogą powodować awarie aplikacji. W przypadku tej aplikacji wiadomości będą zawierały tylko ciągi zawierające złożone zamówienia, które będą dość małe. Aby zaimplementować obiekt `GroupSessionMessenger` dla aplikacji `ShareOrder`, wykonaj następujące kroki:

1. W pliku ViewController dodaj nową opcjonalną właściwość po właściwości groupSession, aby przechowywać instancję GroupSessionMessenger:

```
var groupSession: GroupSession<ShareOrder>?
```

```
var messenger: GroupSessionMessenger?
```

2. Zmodyfikuj metodę configureGroupSession(_:), aby utworzyć instancję GroupSessionMessenger i przypisać ją do właściwości komunikatora:

```
func configureGroupSession(_ groupSession:
```

```
GroupSession<ShareOrder>){
```

```
orders.removeAll()
```

```
self.groupSession = groupSession
```

```
let messenger = GroupSessionMessenger(session:
```

```
groupSession)
```

```
self.messenger = messenger
```

```
groupSession.join()
```

```
}
```

3. Dodaj asynchroniczne zadanie, aby otrzymywać wiadomości GroupSessionMessenger po przypisaniu właściwości komunikatora:

```
let messenger = GroupSessionMessenger(session:
```

```
groupSession)
```

```
self.messenger = messenger
```

```
Task.detached { [weak self] in
```

```
for await (message, _) in messenger.messages(of:
```

```
String.self) {
```

```
await self?.handle(message)
```

```
}
```

```
}
```

Odebrana wiadomość GroupSessionMessenger zawiera ciąg znaków reprezentujący zamówienie. Zaimplementujesz metodę handle(_:) i przekażesz wiadomość do tej metody w celu przetworzenia. Zobaczysz błąd, ponieważ metoda handle(_:) nie została jeszcze zaimplementowana.

4. Zaimplementuj metodę handle(_:) po metodzie configureGroupSession(_:), jak pokazano:

```
func handle(_ message: String) {
```

```
self.orders.append(message)
```

```
self.tableView.reloadData()
```

```
}
```

Ta metoda dołącza ciąg z wiadomości do tablicy orders i ponownie ładuje widok tabeli.

5. Zmodyfikuj metodę `addOrder(_:)` w następujący sposób, aby wysłać wiadomość `GroupSessionMessenger`, gdy uczestnik naciśnie przycisk +:

```
self.orders.append(orderToSave)

if let messenger = messenger {
    Task {
        do {
            try await messenger.send(orderToSave)
        } catch {
            print("Failed to send")
        }
    }
}

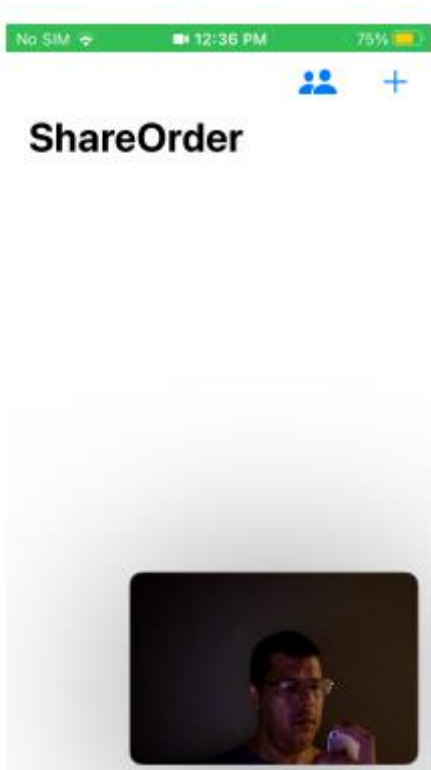
self.tableView.reloadData()
```

Kod dodany do metody `addOrder(_:)` tworzy zadanie asynchroniczne, które wysyła wiadomość zawierającą zamówienie do wszystkich innych uczestników sesji FaceTime. Gdy wiadomość zostanie odebrana, zostanie przetworzona przez metodę `handle(_:)`, która dodaje zamówienie do tablicy orders i przeładowuje widok tabeli. Zbuduj i uruchom aplikację `ShareOrder` na swoim urządzeniu z systemem iOS 15. Powinno działać tak jak wcześniej. W następnej sekcji przetestujesz `SharePlay` w aplikacji `ShareOrder`, używając dwóch urządzeń iOS w sesji FaceTime.

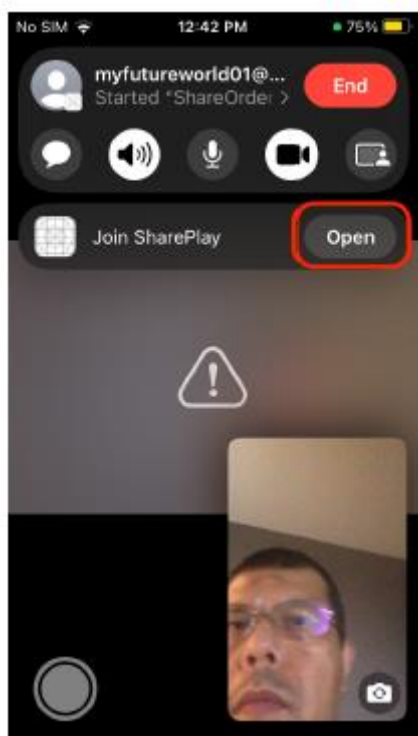
Testowanie SharePlay w aplikacji ShareOrder

Dodałeś cały kod wymagany do wdrożenia `SharePlay` w aplikacji `ShareOrder`. Aby go przetestować, potrzebujesz dwóch urządzeń z systemem iOS 15.1 lub nowszym z zainstalowaną aplikacją `ShareOrder`. Możesz także użyć komputera Mac z systemem macOS 12.1 Monterey lub nowszym jako jednego z urządzeń. Zainicjujesz sesję FaceTime między obydwoma urządzeniami i zainicjujesz sesję `SharePlay`. Powinno być możliwe dodawanie zamówień z dowolnego urządzenia, a wszystkie dodane zamówienia pojawią się na obu ekranach. Wykonaj następujące kroki:

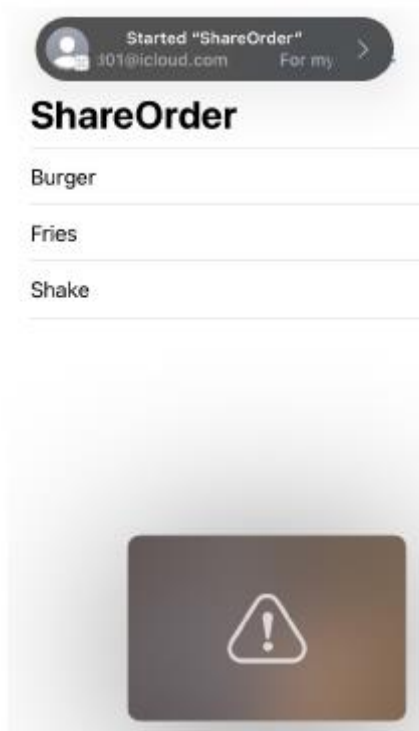
1. Zainstaluj aplikację `ShareOrder` na drugim urządzeniu z systemem iOS.
2. Rozpocznij połączenie FaceTime między dwoma urządzeniami i uruchom aplikację `ShareOrder` na pierwszym urządzeniu.



3. Dotknij przycisku, aby aktywować aktywność grupową. Powinieneś zobaczyć monit SharePlay na drugim urządzeniu.



4. Stuknij przycisk Dołącz w monicie SharePlay, a aplikacja ShareOrder uruchomi się na drugim urządzeniu. Dodaj zamówienie za pomocą przycisku +, a pojawi się ono na ekranie obu urządzeń.



Gratulacje! Właśnie zaimplementowałeś SharePlay w aplikacji ShareOrder!

Podsumowanie

Zaimplementowałeś SharePlay dla aplikacji ShareOrder, dodając do niej obsługę działań grupowych. Zacząłeś od nauki działania SharePlay. Następnie zapoznałeś się z aplikacją ShareOrder, aby zobaczyć, jak to działa. Następnie utworzyłeś niestandardową aktywność grupową dla tej aplikacji i dodałeś kod do zarządzania sesją aktywności grupowej. Na koniec przetestowałeś działanie SharePlay w aplikacji na dwóch urządzeniach z systemem iOS. Rozumiesz już podstawy SharePlay i możesz teraz dodawać niestandardowe działania grupowe do własnych aplikacji.