

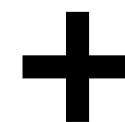
Створення інструменту для автоматичної адаптації елементів UI до навігації за допомогою ігрових контролерів в iOS

Виконав : студент КН-4 Столяров Владислав
Науковий керівник : ст. викладач Франків О.О.



Актуальність проблеми

- Сучасні тенденції: Завдяки потужним процесорам, таким як Apple A17 Pro, iPhone здатні запускати високобюджетні ігри, що збільшує попит на підтримку ігрових контролерів.
- Труднощі: На даний момент інтеграція контролерів в додатки iOS – це складний процес, а налаштування навігації вимагає написання багато додаткового коду



Постановка задачі

Завдання: Розробити фреймворк для інтеграції ігрових контролерів для навігації елементами інтерфейсу в iOS-додатках.

Вимоги:

1. Спростити процес налаштування ігрових контролерів для розробників.

2. Створити єдину систему навігації для різних елементів інтерфейсу.

3. Забезпечити сумісність з існуючими iOS фреймворками та ігровими контролерами.



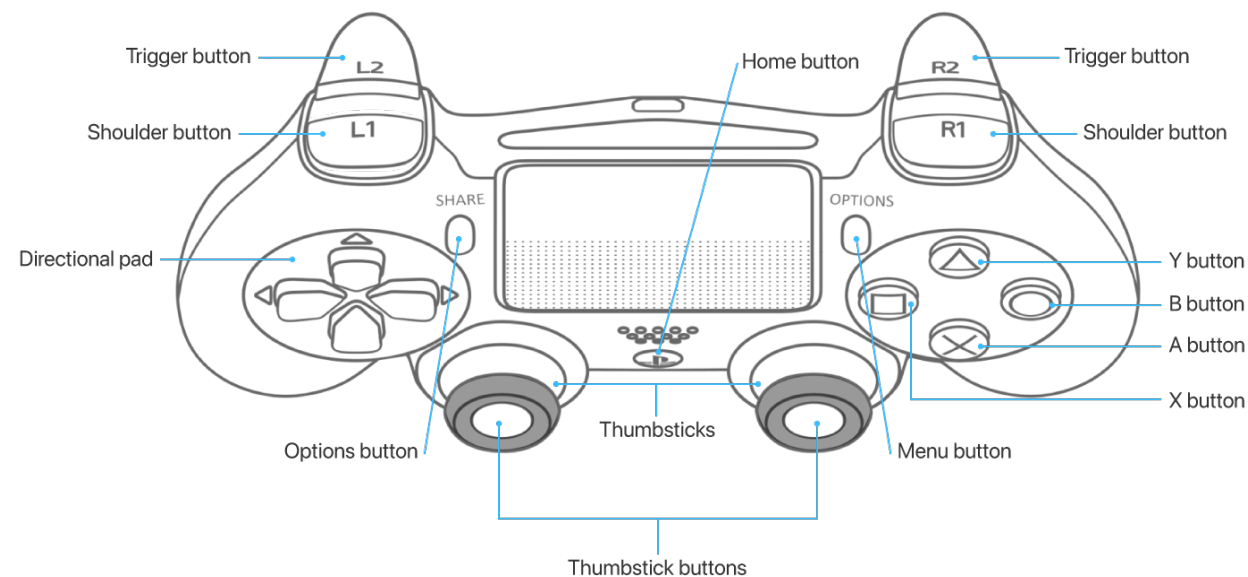
Swift та фреймворки

- Swift - це сучасна протокольно-орієнтована мова, розроблена компанією Apple для розробки під iOS, macOS
- Фреймворки для iOS надають необхідні інструменти та API для розробки додатків, що робить вкрай важливим розуміння їхніх можливостей



Ігрові КОНТРОЛЕРИ





Існуючі рішення

- Фреймворк Game Controller від Apple забезпечує базову підтримку ігрових контролерів, але є обмеженим і складним .
- Для заохочення використання ігрових контролерів потрібне зручне рішення для безперешкодної інтеграції ігрових контролерів.



Фреймворк GameControllerBinde r

GameControllerBinder спрощує процес інтеграції ігрових контролерів у додатки для iOS.

Ключові компоненти: GameControllerBinder, FocusManager, протокол Focusable.

Фреймворк дозволяє легко прив'язувати входи контролера до дій в інтерфейсі, покращуючи взаємодію з користувачем.



Публікація фреймворк у



The screenshot shows the GitHub repository page for **GameControllerBinder**. At the top, it displays the repository name, the user **vladshakhtar**, and the commit hash **f6fbc2e** from 3 days ago with 19 commits. Below this is a commit history table:

File	Commit Message	Time
.build	Initial Commit	5 months ago
.swiftpm/xcode	Initial Commit	5 months ago
Sources/GameControllerBinder	Update controllerConnected and controllerDisconnected ...	3 days ago
Tests/GameControllerBinderTests	Initial Commit	5 months ago
GameControllerBinder.podspec	Added cocoaPods v1.1.1	3 days ago
LICENSE	Create LICENSE	5 months ago
Package.swift	Add FocusManagementSystem and UIElements+Focusabl...	3 days ago
README.md	Update README.md	3 days ago

Below the commit history, there are tabs for **README** and **MIT license**. The main content area shows the **GameControllerBinder** title and a description: "A Swift framework for easily integrating game controller inputs with UI elements in iOS applications." It includes sections for **Overview** and **Features**.

Overview: The GameControllerBinder framework simplifies the process of binding game controller inputs to UI elements and actions within your iOS app. With support for various controller types, including PlayStation and Xbox controllers, this framework allows for quick setup and configuration of controller inputs to enhance the gaming experience on iOS devices.

Features:

- Bind actions to game controller buttons and triggers
- Support for PlayStation and Xbox controllers
- Easy integration with UI elements
- Customize button actions for press and release states
- Automatic focus management for UI elements
- Directional navigation using D-pad and thumbsticks
- Simplified navigation setup for UI elements

On the right side of the repository page, there are several sections:

- Readme**: Links to the README file.
- MIT license**: Link to the license file.
- Activity**: Shows 0 stars, 1 watching, and 0 forks.
- Releases**: Shows 4 releases, with the latest one being "Added focus states" from 3 days ago.
- Packages**: States "No packages published" and provides a link to "Publish your first package".
- Languages**: A bar chart showing the language distribution: Swift 90.1% and Ruby 9.9%.
- Suggested workflows**: Based on the tech stack, it suggests workflows for Swift, Ruby on Rails, and Ruby, each with a "Configure" button.

Доступ до фреймворку

GameControllerBinder 1.1.1

By vladshakhtar

 [vladshakhtar/GameControllerBinder](#)

GameControllerBinder

A Swift framework for easily integrating game controller inputs with UI elements in iOS applications.

Overview

The GameControllerBinder framework simplifies the process of binding game controller inputs to UI elements and actions within your iOS app. With support for various controller types, including PlayStation and Xbox controllers, this framework allows for quick setup and configuration of controller inputs to enhance the gaming experience on iOS devices.

gamecontrollerbinder

Repository [github.com/vladshakhtar/GameControllerBinder](#)

Dependency Rule 1.1.1

Add to Project

GameControllerBinder

A Swift framework for easily integrating game controller inputs with UI elements in iOS applications.

Overview

The GameControllerBinder framework simplifies the process of binding game controller inputs to UI elements and actions within your iOS app. With support for various controller types, including PlayStation and Xbox controllers, this framework allows for quick setup and configuration of controller inputs to enhance the gaming experience on iOS devices.

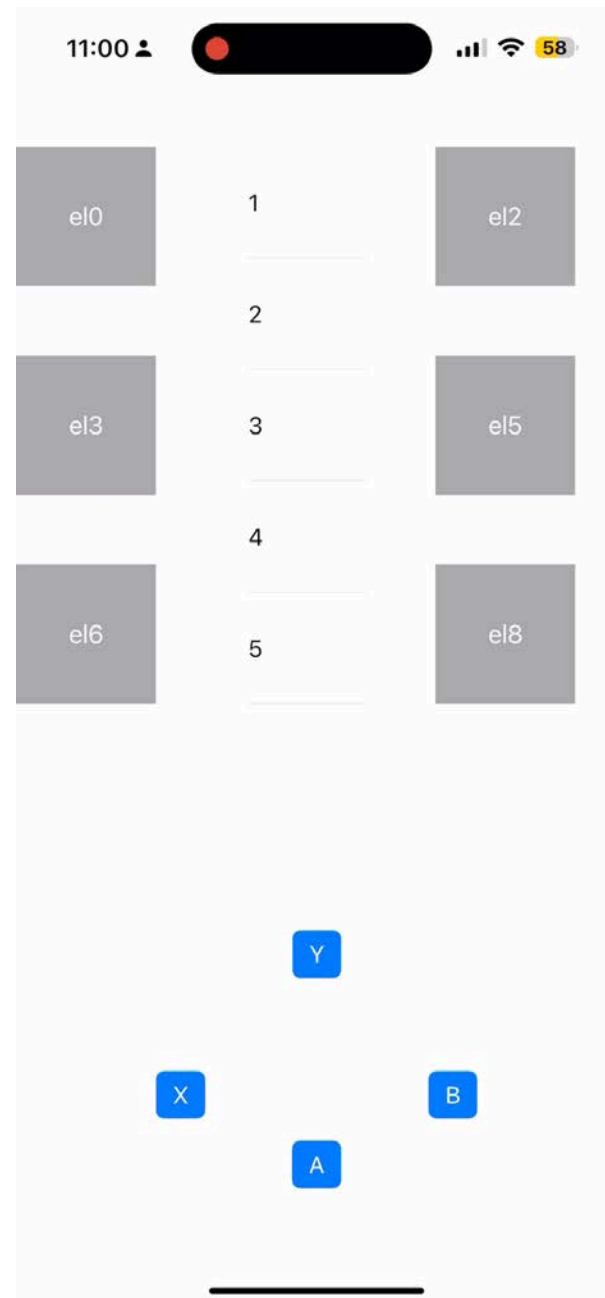
Features

- Bind actions to game controller buttons and triggers
- Support for PlayStation and Xbox controllers
- Easy integration with UI elements
- Customize button actions for press and release states
- Automatic focus management for UI elements
- Directional navigation using D-pad and thumbsticks
- Simplified navigation setup for UI elements



Приклад використання

```
override fun viewDidAppear(_ animated: Bool) {  
    super.viewDidAppear(animated)  
    controllerBinder.registerAllFocusableSubviews(from: view)  
    controllerBinder.setupDefaultDirectionalInputHandlers()  
    controllerBinder.setupTapActionInputHandler()  
    controllerBinder.bindButtonToAction(buttonName: .buttonB) {  
        self.navigationController?.popViewController(animated: true)  
    }  
}
```



Можливості для вдосконалення

Підтримка
SwiftUI

Підтримка
кастомізації
елементів у
фокусі

Підтримка
роботи
динамічного
інтерфейсу





Висновки

Створення загальнодоступного фреймворку GameController

Реалізація ефективного алгоритму для навігації

Використання протоколу Focusable для взаємодії з елементами

Запропоновані вдосконалення перетворюють фреймворк на універсальний інструмент для інтеграції контролерів



Дякую за увагу!

