

change UIViewController class to GADBannerView!

UIViewController.swift

import UIKit

import GoogleMobileAds

class ViewController : UIViewController {

override var preferredStatusBarStyle : UIStatusBarStyle { return .lightContent }

@IBOutlet weak var bannerView : GADBannerView!

override func viewDidLoad() {

super.viewDidLoad()

bannerView.adUnitID = "-----"

bannerView.rootViewController = self

bannerView.load(GADRequest())

bannerView.delegate = self

}

}

extension ViewController : GADBannerViewDelegate {

func adViewDidReceiveAd(_ bannerView: GADBannerView) {

print("received ad")

}

func adView(_ bannerView: GADBannerView, didFailToReceiveAdWithError error: GADRequestError) {

print(error)

3

3

IAP purchases

IAPPurchase.swift

import Foundation

enum IAPPurchase : String {

case consumable = "com.Ujia.-----"

case nonConsumable = "-----"

case autoRenewingSubscription = "-----"

case nonRenewingSubscription = "-----"

3

IAPService.swift

import Foundation

import StoreKit

```
class IAPService : NSObject {
    private override init() {}
    static let shared = IAPService()
    var products = [SKProduct]()
    let paymentQueue = SKPaymentQueue.default()
```

```
func getProducts() {
    let products: Set = [IAPProduct.autoRenewingSubscription.rawValue,
                         IAPProduct.consumeable.rawValue,
                         IAPProduct.nonConsumable.rawValue,
                         IAPProduct.nonRenewableSubscription.rawValue]
    let request = SKProductsRequest(productIdentifiers: products)
    request.delegate = self
    request.start()
    paymentQueue.add(self)
```

}

```
func purchase(product: IAPProduct) {
    guard let productToPurchase = products.filter({ $0.productIdentifier == product.rawValue }).first
    else { return }

    let payment = SKPayment(product: productToPurchase)
    paymentQueue.add(payment)
```

?

```
func restorePurchases() {
    paymentQueue.restoreCompletedTransactions()
```

?

```
extension IAPService : SKProductsRequestDelegate {
    func productsRequest(_ request: SKProductsRequest, didReceive response: SKProductsResponse) {
        self.products = response.products
```

```
        for product in response.products {
```

?

?

```
extension IAPService : SKPaymentTransactionObserver {
```

```
    func paymentQueue(_ queue: SKPaymentQueue, updatedTransactions transactions: [SKPaymentTransaction]) {
        for transaction in transactions {
            print(transaction.transactionState.rawValue, transaction.payment.productIdentifier)

            switch transaction.transactionState {
                case .purchasing: break
```

default: queue.finishTransaction(transaction)

}

}

extension SKPaymentTransactionState {

func status() -> String {

switch self {

case .deferred: return "deferred"

case .failed: ...

case .purchased: ...

case .purchasing: ...

case .restored: ...

}

}

}

Money VC.swift

import UIKit

class MoneyVC: UITableViewController {

override func viewDidLoad() {

super.viewDidLoad()

IAPService.shared.getProducts()

}

override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {

switch indexPath.row {

case 0: IAPService.shared.purchase(product: .consumable)

case 1: ...

case 2: ...

case 3: ...

default: IAPService.shared.restorePurchases()

}

}

11/11/2018

Handwriting practice sheet