

GAMEKIT

@ S U M I H I R O

GameKitとは

- ✿ 複数台のiPhone/iPod touchをBlueTooth(BT)を用いて接続するフレームワーク
- ✿ iPhoneOS 3.0以降で利用可能
- ✿ GKPeerPickerControllerを使用した場合は1対1の接続が可能

大まかな流れ

- GKPeerPickerController起動
- [GKPeerPickerControllerDelegate peerPickerController:sessionForConnectionType:]でGKSessionを作成
- [GKPeerPickerControllerDelegate peerPickerController:didConnectPeer:toSession:]で接続が完了
- 以後は[GKSession sendData:toPeers:withDataMode:error:]と[GKSession receiveData:fromPeer:inSession:context:]でデータを取りする

IMPLEMENT

実装

GKPeerPickerControllerDelegateのメソッドを実装

- ✿ [GKPeerPickerControllerDelegate peerPickerController:sessionForConnectionType:]
- ✿ [GKPeerPickerControllerDelegate peerPickerController:didConnectPeer:toSession:]

セッションを作成

```
- (GKSession *)peerPickerController:(GKPeerPickerController *)picker  
    sessionForConnectionType:(GKPeerPickerControllerConnectionType)type  
{  
    GKSession *session = [[GKSession alloc]  
        initWithSessionID:@"TESTGK"  
        displayName:nil  
        sessionMode:GKSessionModePeer];  
    return [session autorelease];  
}
```

- ※ initWithSessionID: はセッションのID。別プロジェクトでも、この値が同じならば接続可能。
- ※ displayName:nil とすると、iTunesに表示されるiPhoneの名前が使用される。
- ※ sessionMode: でP2P接続かサーバ-クライアント接続かを設定する。

接続完了時の処理

```
- (void)peerPickerController:(GKPeerPickerController *)picker
    didConnectPeer:(NSString *)peerID
        toSession:(GKSession *)session
{
    self.gamePeerID = peerID;

    self.gameSession = session;
    self.gameSession.delegate = self;
    [self.gameSession setDataReceiveHandler:self withContext:nil];

    [picker dismiss];
    picker.delegate = nil;
}
```

- ※ peerID と session を保持する。
- ※ GKSessionDelegate とデータ受信を処理するインスタンスを設定する。
- ※ picker の表示を終了し、delegate を解除する。

データ受信

```
- (void) receiveData:(NSData *)data
              fromPeer:(NSString *)peer
            inSession:(GKSession *)session
             context:(void *)context
{
    [self processTheData:data];
}
```

- ✿ [GKSession setDataReceiveHandler:withContext:]で指定したインスタンスに対して、データ受信時に呼ばれる。
- ✿ データは全てNSDataで送受信される。

データ送信

```
- (void)sendData:(NSData*)data {
    [self.gameSession sendData:data
        toPeers:[NSArray arrayWithObject:self.gamePeerID]
        withDataMode:GKSendDataReliable
        error:nil];
}
```

- ✿ withDataMode: に GKSendDataReliable を指定すると確実にデータを転送する。 GKSendDataUnreliable を指定するとデータの転送は保証されない。

GKPeerPickerControllerの起動

```
- (void)startPicker {
    GKPeerPickerController* picker;
    picker = [[GKPeerPickerController alloc] init];
    picker.delegate = self;
    [picker show];
}
```

- ✿ UIImagePickerController と UIAlertView を混ぜたような起動方法。

PRACTICE

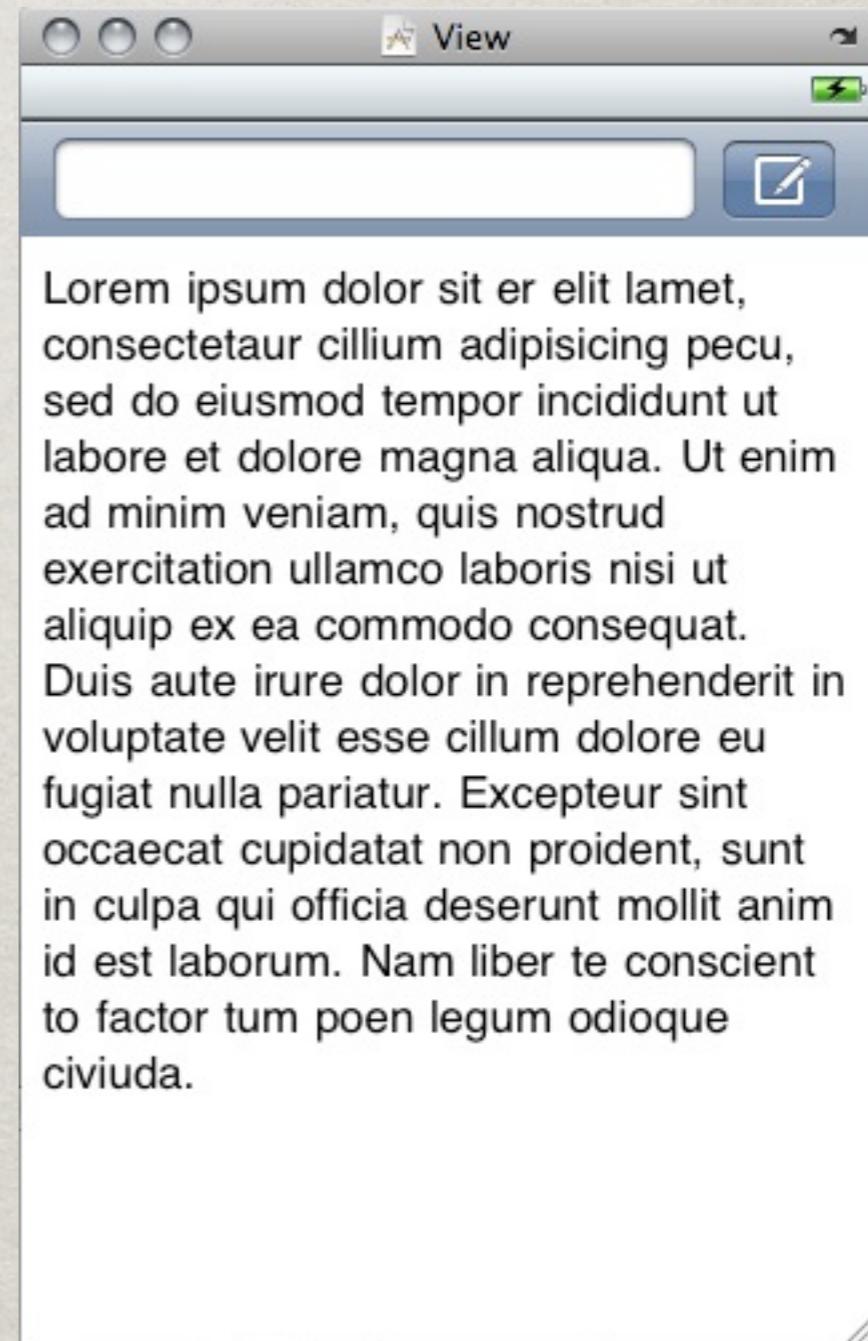
実践

テキストチャットを作る

- ✿ P2Pで接続する
- ✿ メッセージを送信する、及び、受信するだけのシンプルな機能

ユーザーアインターフェース

- ✿ ツールバー上に
TextField と
BarButtonItem を設置
- ✿ ログ表示は TextView



送信

```
- (IBAction)sendChatMessage:(id)sender {
    NSString *message = self.chatTextField.text;
    if (![message isEqualToString:@""]){
        NSMutableString *newText = [NSMutableString stringWithFormat:@"あなた: %@", message];
        [newText appendString:self.mainTextView.text];
        self.mainTextView.text = newText;
        [self sendData:[message dataUsingEncoding:NSUTF8StringEncoding]];
        self.chatTextField.text = @"";
    }
}
```

- ※ BarButtonItem が押された時に、TextField に記述された文章を送信する。
- ※ NSString は [NSString dataUsingEncoding:] で NSData に変換できる。

受信

```
- (void) receiveData:(NSData *)data
    fromPeer:(NSString *)peer
    inSession: (GKSession *)session
    context:(void *)context
{
    NSString *message = [[NSString alloc] initWithData:data
                                                encoding:NSUTF8StringEncoding];
    NSMutableString *newText = [NSMutableString stringWithFormat:@"あの人: %@
\n", message];
    [newText appendString:self.mainTextView.text];
    self.mainTextView.text = newText;
}
```

- ※ [GKSession setDataReceiveHandler:withContext:]で指定したインスタンスに
対して、データ受信時に呼ばれる。
- ※ NSData から [NSString initWithData:encoding:] で NSString に変換できる。

動作テスト

TIPS

おまけ

NSDictionary vs NSData

```
NSDictionary *dict = [NSDictionary  
    dictionaryWithObjectsAndKeys:@"firstData",@"firstKey",  
                           @"secndData",@"secondKey",nil];  
NSData *data = [NSKeyedArchiver archivedDataWithRootObject:dict];  
  
NSDictionary *dict2 =  
    (NSDictionary*)[NSKeyedUnarchiver unarchiveObjectWithData:data];
```

- ✿ NSKeyedArchiver / NSKeyedUnarchiver を
使用すると簡単に変換できる。

UIImage vs NSData

```
UIImage *image = [UIImage imageNamed:@"someImage"];
NSData *data = UIImagePNGRepresentation(image);

UIImage *image2 = [UIImage imageWithData:data];
```

- ※ UIImage は NSKeyed(Un)Archiver では NSData と変換できない。
- ※ UIImagePNGRepresentation もしくは
UIImageJPEGRepresentation を使用して画像データに変換する。

サンプルコード

✿ Apple謹製の GKTank のソースコードを読みましょう。

THANK YOU.

ありがとうございました。

@SUMIHIRO

<http://twitter.com/sumihiro>