

CS 696 Mobile Phone Application Development
Fall Semester, 2010
Doc 22 Game Kit
Nov 18, 2010

Copyright ©, All rights reserved. 2010 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

Game Kit Programming Guide, Apple Documentation

More iPhone 3 Development: Tackling iPhone SDK3, Mark & LaMarche, Apress, 2009
Chapter 8 Peer-to-Peer Over Bluetooth Using GameKit

GameKit

Game Center



- Authentication
- Friends
- Leaderboards
- Achievements
- Auto-matching
- Invitations
- Peer-to-Peer Networking
- In-Game Voice Chat

Peer-to-peer Connectivity



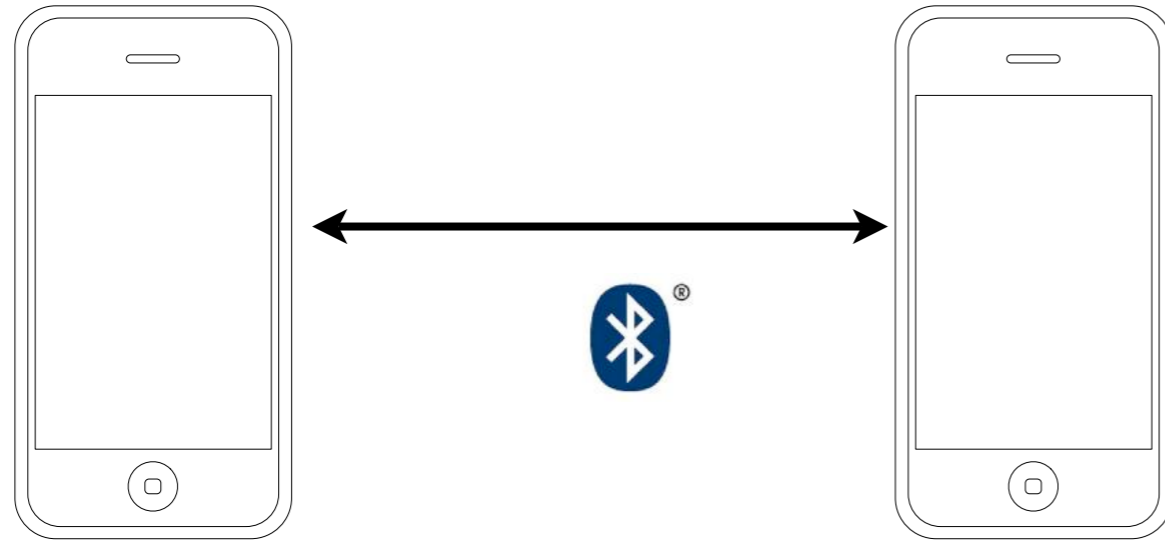
- Bluetooth Local Wireless

In-Game Voice



- Voice Chat Services

Peer-to-Peer



Peer-to-Peer

Server

- Advertises a service

- Service has a session ID

Client

- Searches for a service

- Search for given session ID

Peer

- Advertises like server

- Searches like client

- At same time

Peer-to-Peer Classes

GKSession

Manages streams of data between peers

Sends to one peer or

Broadcast to all

GKSessionDelegate

GKPeerPickerController

UI for selecting peer/server

GKPeerPickerControllerDelegate

Sending Data

You can send any data you want

Sender puts it in NSData

Receiver has to extract data from NSData object

Receiver has to know the data format

GKSession

Some Properties

Manages discovery of peers

Simple interface for
Manage peer connections
Sending/Receiving data

displayName

Peer displays to user

peerID

Unique ID, identify you to peers

sessionID

String used to identify service

GKSession - Sending data

sendData:toPeers:withDataMode:error:

Data is NSData object

sendDataToAllPeers:withDataMode:error:

setDataReceiveHandler:withContext:

Sets handler to receive data from peers

Context is sent to handler

DataReceiveHandler

```
- (void) receiveData:(NSData *)data  
  fromPeer:(NSString *)peer  
  inSession: (GKSession *)session  
  context:(void *)context;
```

GKSessionDelegate

Observing Changes to Peers

- session:peer:didChangeState:
GKPeerStateAvailable,
GKPeerStateUnavailable,
GKPeerStateConnected,
GKPeerStateDisconnected,
GKPeerStateConnecting

Connection Requests From Other Peers

- session:didReceiveConnectionRequestFromPeer:

Connection Errors

- session:connectionWithPeerFailed:withError:
- session:didFailWithError:

GKPeerPickerController

UI for connecting to peer

User can pick between "Nearby" and "Online"

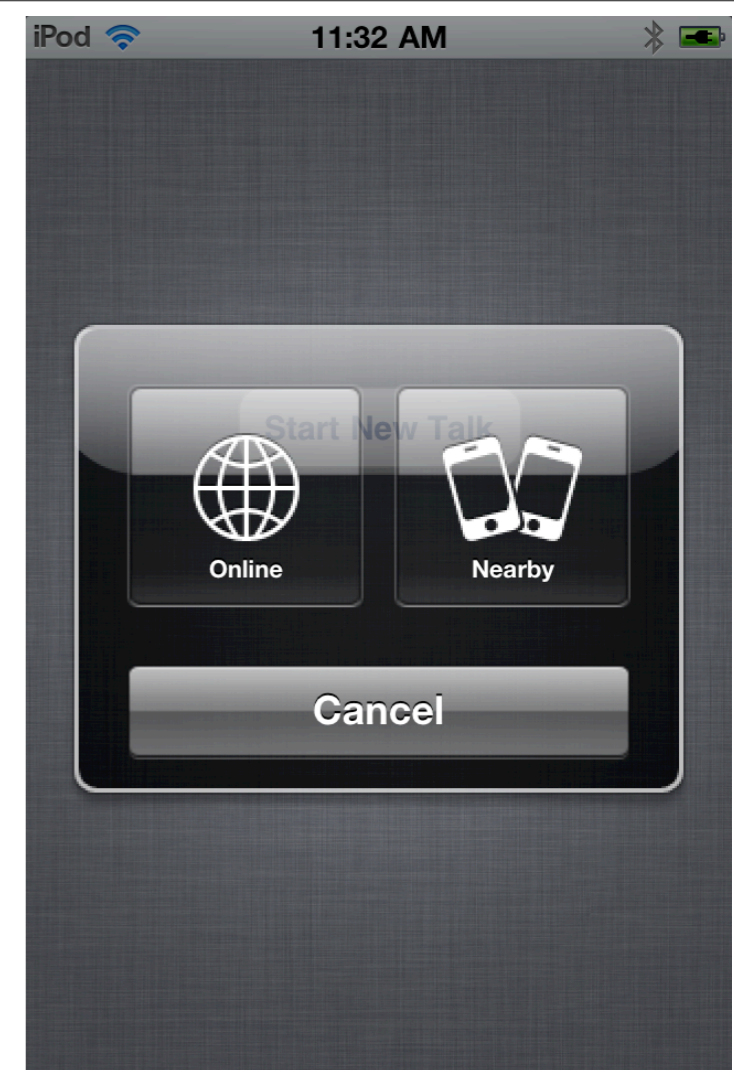
Nearby

Bluetooth

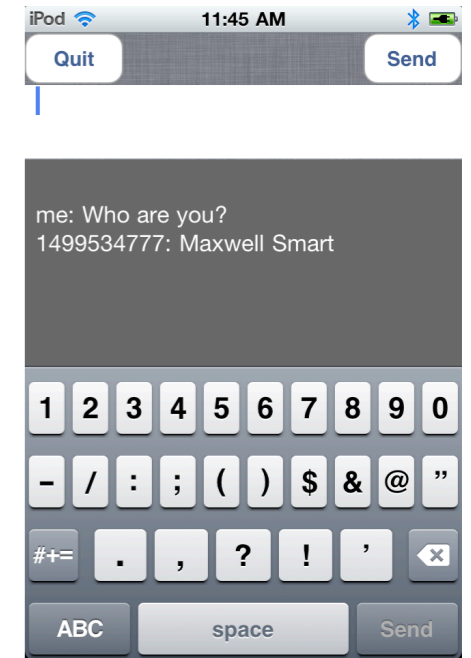
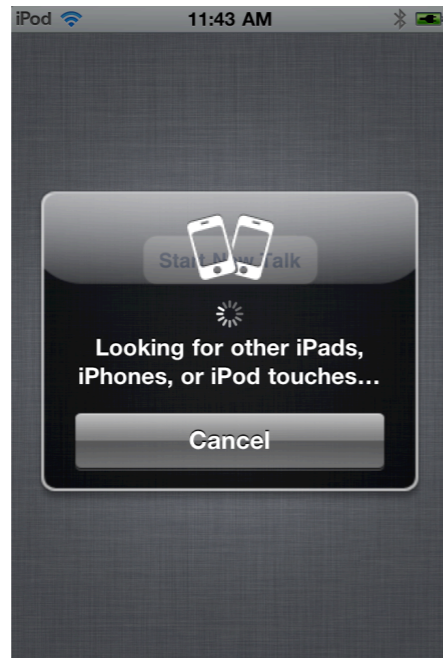
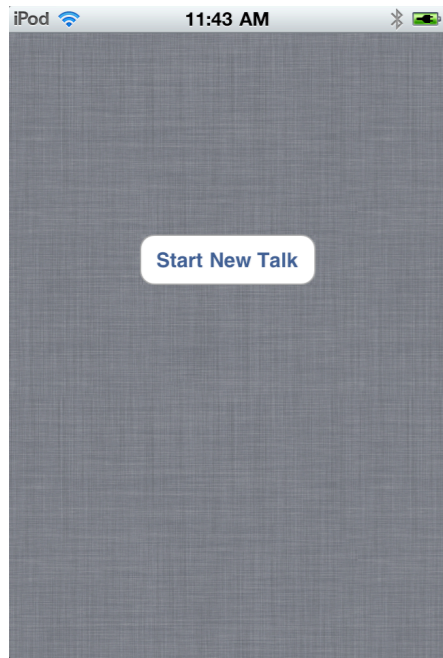
Connects to peer for you

Online

Application handles details of connection



QuickTalk - Peer-to-peer Example



Starting GKPeerPickerController

```
- (IBAction) startNewTalkPressed:(id) sender{
    GKPeerPickerController *picker;
    picker = [[GKPeerPickerController alloc] init];
    picker.delegate = self;
    picker.connectionTypesMask = GKPeerPickerControllerConnectionTypeNearby;
    [picker show];
}
```

Connection Mask Types

GKPeerPickerConnectionTypeOnline

GKPeerPickerConnectionTypeNearby

GKPeerPickerConnectionTypeOnline | GKPeerPickerConnectionTypeNearby

GKPeerPickerController Delegate Method

Session for connections

```
- (GKSession *)peerPickerController:(GKPeerPickerController *)picker
    sessionForConnectionType:(GKPeerPickerControllerConnectionType)type {

    GKSession *theSession = [[GKSession alloc] initWithSessionID:kSessionID
                            displayName:nil
                            sessionMode:GKSessionModePeer];
    return [theSession autorelease];
}
```


Session Modes

GKSessionModeServer,
GKSessionModeClient,
GKSessionModePeer,

GKPeerPickerController Delegate Method

When Peer connects

```
- (void)peerPickerController:(GKPeerPickerController *)picker
    didConnectPeer:(NSString *)thePeerID
    toSession:(GKSession *)theSession {
    self.peer = thePeerID;
    self.session = theSession;
    self.session.delegate = self;
    [self.session setDataReceiveHandler:self withContext:NULL];
    [picker dismiss];
    picker.delegate = nil;
    [picker autorelease];

    [self startTalk];
}
```

GKPeerPickerController Delegate Method

Not needed in example

- (void)peerPickerControllerDidCancel:(GKPeerPickerController *)picker {
}
- (void)peerPickerController:(GKPeerPickerController *)picker
 didSelectConnectionType:(GKPeerPickerConnectionType)type {

}

Sending Message

Called by Send button

```
- (void) send{
    [self sendPacket:self.entryTextView.text];
    self.entryTextView.text = @"";
}

- (void) sendPacket:(NSString *)message {

    NSData* data = [message dataUsingEncoding:NSUTF8StringEncoding];

    NSError *error = nil;
    if (![session sendDataToAllPeers:data withDataMode:GKSendDataReliable error:&error]) {
        NSLog(@"Error sending data: %@", [error localizedDescription]);
    } else {
        self.historyTextView.text = [NSString stringWithFormat:@"%@\n%@: %@",
            self.historyTextView.text,@"me",message];
    }
}
```

Receiving Data

Data Receiver Handler

```
- (void) receiveData:(NSData *)data
  fromPeer:(NSString *)peer
  inSession:(GKSession *)theSession
  context:(void *)context {

    NSString* message = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];

    self.historyTextView.text = [NSString stringWithFormat:@"%@\n%@: %@",
                                self.historyTextView.text, self.peer, message];
    [message release];
}
```

GKSessionDelegate Method

```
- (void)session:(GKSession *)theSession didFailWithError:(NSError *)error {
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle: @"Error Connecting!"
                                                       message: @"Unable to establish the
connection."
                                                       delegate:self
                                                       cancelButtonTitle: @"Ok"
                                                       otherButtonTitles:nil];

    [alert show];
    [alert release];
    theSession.available = NO;
    [theSession disconnectFromAllPeers];
    theSession.delegate = nil;
    [theSession setDataReceiveHandler:nil withContext:nil];
    self.session = nil;
}
```

GKSessionDelegate Method

```
- (void)session:(GKSession *)theSession peer:(NSString *)peerID
    didChangeState:(GKPeerConnectionState)inState {

    if (inState == GKPeerStateDisconnected) {
        UIAlertView *alert = [[UIAlertView alloc] initWithTitle: @"Peer Disconnected!"
                                                            message:@"Your party has disconnected"
                                                            delegate:self
                                                            cancelButtonTitle: @"Ok"
                                                            otherButtonTitles:nil];

        [alert show];
        [alert release];

        theSession.available = NO;
        [theSession disconnectFromAllPeers];
        theSession.delegate = nil;
        [theSession setDataReceiveHandler:nil withContext:nil];
        self.session = nil;
    }
}
```