Game Center

Bonus Lecture #7
Overview

• GameKit Framework
• Game Center
  • Getting Started
  • Checklists
• Quick (^block) review
• Resources
  • Peer-to-Peer and Voice Chat
GameKit
Overview

• The Framework
• Game Center
• Peer-to-Peer Connectivity
  • Wireless
  • Bluetooth
• Voice Chat
Game Center
Overview

Authentication
• Leaderboards
• Achievements
• Multiplayer
  • Matchmaking
• Peer-to-peer networking
Game Center

Checklist

Authenticating a player
• A View Controller to manage your interface
• Asynchronous Game Center operation(s)
• Callbacks are block objects
• Handling network failure (E.G. resending)
  • Handling partial data
Game Center

Getting Started

• Enable Game Center in iTunes Connect.
• Configure the Bundle Identifier for your game.
• Link to the Game Kit framework.
• Import the GameKit/GameKit.h header.
• Determine whether your game requires Game Center.
• Add the Game Center key to list of IC capabilities
• IF not, weak link to the Game Kit framework
• Authenticate the player as soon as possible on launch
Game Center
iTunes Connect

• Register your App
  • This includes the App ID
• Create leaderboard(s)
• Create achievements
• Match your Bundle ID in your Xcode project
Game Center
Leaderboards (checklist)

• Authenticate the local player
• Decide how you want your game to calculate scores
• Go to iTunes Connect and configure your leaderboards
• Add code to report scores to Game Center
• Add code to handle reporting problems
• Add code to allows a player to view a leaderboard
Game Center
Achievements (checklist)

• Authenticate the local player
• Go to iTunes Connect and configure the achievements
• Report player’s progress to Game Center
• Load the player’s previous progress from Game Center
• Add code to allow the player to view achievements
Game Center
Multiplayer: Matchmaking (checklist)

• Authenticate the local player
• Display the matchmaking screen to the user
• Add code that processes invitations
• Optionally
  • Add code to programmatically find matches
  • Add advanced functionality to your game
    • E.G. Matchmaking “groups” or restrictions
Game Center

Multiplayer (checklist)

• Use Game Center’s matchmaking service
• Design your network messages (GKMatch)
• Write code to send the match object data
• Match delegate to handle the events
  1. Connection notifications
  2. Data received notifications
  3. Disconnection notifications
• Optionally
  • Add support for voice between the match participants.
Game Center
Voice Chat (checklist)

• How many channels does your game need?
• Configure an audio session to enable the microphone.
• Call the match object’s `voiceChatWithName:` method
• Call the voice chat object’s `start` method
• Enable the microphone for a channel
• Allow the user to enable and disable voice chat.
• Push-to-talk model or continuously sample?
• Optionally
  • Implement an update handler
    • Player connects, disconnects, starts or stops speaking, etc.
Game Center
Peer-to-Peer (reference)

Game Kit Programming Guide
Peer-to-Peer Connectivity

These slides are licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.
Game Center
Voice Chat (reference)

Game Kit Programming Guide
Part III: In-Game Voice
Blocks
Quick Review

Why they for?

• Completion handlers
• Notification handlers
• Error handlers
• Enumeration
• View animation and transitions
• Sorting
• etc., etc., etc…
Blocks

Quick Review

```c
int (^Multiply)(int, int) = ^(int num1, int num2)
{
    return num1 * num2;
}

int result = Multiply(7, 4); // result is 28
```
Blocks
Quick Review

```cpp
int (^Multiply)(int, int) = ^(int num1, int num2)
{
    return num1 * num2;
};

int result = Multiply(7, 4); // result is 28
```
Blocks
Quick Review

```plaintext
int (^Multiply)(int, int) = ^(int num1, int num2)
{
    return num1 * num2;
}

int result = Multiply(7, 4); // result is 28
```
Blocks
Quick Review

```c
int (^Multiply)(int, int) = ^(int num1, int num2)
{
    return num1 * num2;
};

int result = Multiply(7, 4); // result is 28
```
Blocks
Quick Review

int (^Multiply)(int, int) = ^(int num1, int num2)
{
    return num1 * num2;
};

int result = Multiply(7, 4); // result is 28
Blocks
Quick Review

```cpp
int (^Multiply)(int, int) = ^(int num1, int num2)
{ return num1 * num2;
};

int result = Multiply(7, 4); // result is 28
```
Blocks
Quick Review

```c
int (^Multiply)(int, int) = ^(int num1, int num2)
{
    return num1 * num2;
};

int result = Multiply(7, 4); // result is 28
```
Blocks
Quick Review

- (void)viewDidLoad
{
    [super viewDidLoad];
    [[NSNotificationCenter defaultCenter]
     addObserverForName:UIKeyboardWillShowNotification
     object:nil
     queue:[NSOperationQueue mainQueue]
     usingBlock: ^(NSNotification *notif)
     {
         // Notification-handling code goes here.
     }];
}

These slides are licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.
To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.
Blocks
Quick Review

- (void)viewDidLoad
{
    [super viewDidLoad];
    [[NSNotificationCenter defaultCenter]
     addObserverForName:UIKeyboardWillShowNotification
     object:nil
     queue:[NSOperationQueue mainQueue]
     usingBlock: ^(NSNotification *notif)
     {
         // Notification-handling code goes here.
     }];
}
Resources

Game Kit Programming Guide

Game Center Overview

A Short Practical Guide to Blocks

iTunes Connect Developer Guide