iOS: Blocks and GCD

Or

How to keep your
UI Responsive!

@techieGary

Blocks

- Objective-C Code
- Group of Statements that can be passed to methods
- Blocks can execute in separate thread

Why?

- Increase responsiveness of your UI
- Increases readability of Code
- “ninja move” = Can replace Delegates and Notifications in certain situations

```
NSDictionary *myDictionary = [[NSDictionary alloc]
initWithObjectsAndKeys:
      @"90",@"Test1",
      @"89",@"Quiz1",
      @"96",@"Test2",
      nil];
```

```
NSDictionary *myDictionary = [[NSDictionary alloc] initWithObjectsAndKeys:
@"90",@"Test1",
@"89",@"Quiz1",
@"96",@"Test2",
nil];
```

Method

```
[myDictionary
enumerateKeysAndObjectsUsingBlock:
^(id key, id obj, BOOL *stop)
 {
    NSLog(@"%@ is equal to %@", key, obj);
 }];
```

Parameters

Block

```
>Test1 is equal to 90
>Quiz1 is equal to 89
>Test2 is equal to 96
```

Method

```
[myDictionary enumerateKeysAndObjectsUsingBlock:
  ^(id key, id value, BOOL *stop)
{
  NSLog(@“Value for key %@ is %@”, key, value);
    if ([@“EOF” isEqualToString:key])
    {
      *stop = YES;
    }
}];
```

MobileDevNJ

Arguments

```
[myDictionary enumerateKeysAndObjectsUsingBlock:
  ^(id key, id value, BOOL *stop)
{
  NSLog(@“Value for key %@ is %@”, key, value);
    if ([@“EOF” isEqualToString:key])
    {
      *stop = YES;
    }
}];
```

MobileDevNJ

Parameters can be writeable

```
NSDictionary *myDictionary = [[NSDictionary ...];

[myDictionary enumerateKeysAndObjectsUsingBlock:
  ^(id key, id value, BOOL *stop)
{
  NSLog(@“Value for key %@ is %@”, key, value);
  if ([@“EOF” isEqualToString:key])
  {
    *stop = YES;
  }
}];
```

MobileDevNJ

Variables

- local (Read only)
- __block and Instance variables (Read/Write)
- Objects: Make sure objects don't go out of scope!

```
double stopValue = 99;
[myDictionary enumerateKeysAndObjectsUsingBlock:
  ^(id key, id value, BOOL *stop)
  {
    NSLog(@“value for key %@ is %@”, key, value);
      if ([value doubleValue] == stopValue) // READ ONLY
      {
        *stop = YES;
      }
  }
];
```

MobileDevNJ

```
__block double stopValue = 99;
[myDictionary enumerateKeysAndObjectsUsingBlock:
  ^(id key, id value, BOOL *stop)
  {
    NSLog(@“value for key %@ is %@”, key, value);
      if ([value doubleValue] == stopValue)
      {
        *stop = YES;
        stopValue = 0; // Reset
      }
  }];
NSLog (@”The value of stopValue is %f”, stopValue);
```

```
NSDictionary *myDictionary = [[NSDictionary alloc]
initWithObjectsAndKeys:
   @"90",@"Test1",
   @"100",@"Quiz1",
   @"96",@"Test2",
   nil];

__block NSString *stopKey = [NSString
stringWithFormat:@"100"];
    [myDictionary enumerateKeysAndObjectsUsingBlock:
     ^(id key, id value, BOOL *stop)
{
    NSLog(@"value for key %@ is %@", key, value);
    if ([value isEqualToString:stopKey])
    {
        *stop = YES;
        stopKey = [NSString stringWithString:key];
    }
}];
NSLog (@"The value of stopValue is %@", stopKey);
```

Another Example

```
[UIView animateWithDuration:0.5
    delay:1.0
    options: UIViewAnimationCurveEaseOut
 animations:
             ^{
  self.basketTop.frame = topFrame;
  self.basketBottom.frame = bottomFrame;
                }
      completion:
                ^(BOOL finished)
                {
                  NSLog(@"Done!");
                }];
```

MobileDevNJ

What does this Code Do?

```
- (void)viewWillAppear:(BOOL)animated
  {
    NSURL *url = [NSURL
    URLWithString:@"http...huge.jpg"];
    NSData *imageData = [NSData
        dataWithContentsOfURL:url];

    UIImage *image =[UIImage imageWithData:imageData];
    self.imageView.image = image;
    self.imageView.frame = CGRectMake(0, 0, 320, 480);

    [self startGameLoop];
  }
```

MobileDevNJ

```
- (void)viewWillAppear:(BOOL)animated
  {
    NSURL *url = [NSURL
    URLWithString:@"http...huge.jpg"];
    NSData *imageData = [NSData
    dataWithContentsOfURL:url];

    UIImage *image = [UIImage
    imageWithData:imageData];
    self.imageView.image = image;
    self.imageView.frame = CGRectMake(0, 0, 320,
    480);

    [self startGameLoop];
    }
```

```
- (void)viewWillAppear:(BOOL)animated
  {
    NSURL *url = [NSURL
    URLWithString:@"http...huge.jpg"];
    NSData *imageData = [NSData
    dataWithContentsOfURL:url];

    UIImage *image = [UIImage
    imageWithData:imageData];
    self.imageView.image = image;
    self.imageView.frame = CGRectMake(0, 0, 320,
    480);

    [self startGameLoop];
    }
```

Network Call

Updating UI

```
- (void)viewWillAppear:(BOOL)animated
  {
    NSURL *url = [NSURL
    URLWithString:@"http...huge.jpg"];
    NSData *imageData = [NSData
    dataWithContentsOfURL:url];

    UIImage *image = [UIImage
    imageWithData:imageData];
    self.imageView.image = image;
    self.imageView.frame = CGRectMake(0, 0, 320,
    480);

    [self startGameLoop];
  }
```

Network Call

Updating UI

Suffering starts here ;

MobileDevNJ

GCD

- C APIs that manages queues for iOS
- Queues run in separate threads
  - Serial and Concurrent
- If your queue blocks, only affects that queue

Creating and releasing queues

```
dispatch_queue_t dispatch_queue_create(const
char *label, NULL);
// Queue can be concurrent or serial = NULL
```

```
dispatch_release(dispatch_queue_t);
```

Putting blocks in the queue

```
typedef void (^dispatch_block_t)(void);

 void dispatch_async(dispatch_queue_t queue,
       dispatch_block_t block);
```

Defining Blocks

```
typedef double (^my_block_t)(double op);
my_block_t sqrt;

sqrt = ^(double operand) {
    // block code goes here
    return operand * operand;
}
double squareValue = sqrt(5.0);
```

MobileDevNJ

Defining Blocks

Return Value Name Parameters

```
double (^sqrt)(double op) = ^(double op)
{ return op * op; };
```

```
double square = sqrt(5.0);
```

Return value defaults to what's returned in Block

MobileDevNJ

Putting blocks in the queue

```
double (^sqrt)(double op) = ^(double op)
    { return op * op; };

dispatch_queue_t downloadQueue =
dispatch_queue_create("image downloader", NULL);

dispatch_async(downloadQueue,
    ^{
        NSLog(@"%2.f",sqrt(5.0));
    });

dispatch_release(downloadQueue);
```

Getting the current queue

```
dispatch_queue_t dispatch_get_current_queue();
dispatch_queue_retain (dispatch_queue_t);

    // keep it in the heap until dispatch_release
```

Getting the current queue

```
dispatch_queue_t dispatch_get_main_queue();
```

```
- (void)viewWillAppear:(BOOL)animated
    {
        [self startGameLoop];
    }
```

```
- (void)viewWillAppear:(BOOL)animated
   {
      dispatch_queue_t downloadQueue =
                  dispatch_queue_create("image downloader", NULL);
```

```
      [self startGameLoop];
   }
```

MobileDevNJ

```
- (void)viewWillAppear:(BOOL)animated
   {
      dispatch_queue_t downloadQueue =
                     dispatch_queue_create("image downloader", NULL);
      dispatch_async(downloadQueue, ^{

      });

      [self startGameLoop];
  }
```

MobileDevNJ

```
- (void)viewWillAppear:(BOOL)animated
   {
      dispatch_queue_t downloadQueue =
                     dispatch_queue_create(“image downloader”, NULL);
      dispatch_async(downloadQueue, ^{

         NSURL *url = [NSURL URLWithString:@"http...huge.jpg"];
         NSData *imageData = [NSData dataWithContentsOfURL:url];
```

```
      });

      [self startGameLoop];
   }
```

MobileDevNJ

```
- (void)viewWillAppear:(BOOL)animated
   {
      dispatch_queue_t downloadQueue =
                  dispatch_queue_create(“image downloader”, NULL);
      dispatch_async(downloadQueue, ^{

         NSURL *url = [NSURL URLWithString:@"http...huge.jpg"];
         NSData *imageData = [NSData dataWithContentsOfURL:url];

      dispatch_async(dispatch_get_main_queue(), ^{
            UIImage *image = [UIImage imageWithData:imageData];
            self.imageView.image = image;self.imageView.frame = CGRectMake(0, 0,
            320, 480);
         });
      });

      [self startGameLoop];
   }
```

MobileDevNJ

Let's Look @ Code

Follow:

@MobileDevNJ

@TechieGary