Managing Selections

Selections in Table Views

1. A few human-interface guidelines to keep in mind when dealing with cell selection in table views:

=> You should never use selection to indicate state.

=> When the user selects a cell, you should respond by deselecting the previously selected cell.

=> If you respond to the the selection of a cell by pushing a new view controller onto the navigation controller’s stack, you should deselect the cell when the view controller is popped off the stack.

Responding to Selections

1. Responding to a row selection

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:
(NSIndexPath *)indexPath
 {
    [tableView deselectRowAtIndexPath:indexPath animated:NO];
    BATTrailsViewController *trailsController = [[BATTrailsViewController
alloc] initWithStyle:UITableViewStylePlain];
    trailsController.selectedRegion = [regions
objectAtIndex:indexPath.row];
    [[self navigationController] pushViewController:trailsController
animated:YES];
}
```

2. Setting a switch object as an accessory view and responding to its action message

```
- (UITableViewCell *)tableView:(UITableView *)tv
cellForRowAtIndexPath:(NSIndexPath *)indexPath {
      UITableViewCell *cell = [tv
dequeueReusableCellWithIdentifier:@"CellWithSwitch"];
      if (cell == nil) {
          cell = [[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:@"CellWithSwitch"];
          cell.selectionStyle = UITableViewCellSelectionStyleNone;
          cell.textLabel.font = [UIFont systemFontOfSize:14];
      }
      UISwitch *switchObj = [[UISwitch alloc]
initWithFrame:CGRectMake(1.0, 1.0, 20.0, 20.0)];
```

```
        switchObj.on = YES;
        [switchObj addTarget:self action:@selector(toggleSoundEffects:)
forControlEvents:(UIControlEventValueChanged |
UIControlEventTouchDragInside)];
        cell.accessoryView = switchObj;

        cell.textLabel.text = @"Sound Effects";
        return cell;
}

- (void)toggleSoundEffects:(id)sender {
    [self.soundEffectsOn = [(UISwitch *)sender isOn];
    [self reset];
}
```

3. Managing a selection list—exclusive list

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:
(NSIndexPath *)indexPath {

    [tableView deselectRowAtIndexPath:indexPath animated:NO];
    NSInteger catIndex = [taskCategories
indexOfObject:self.currentCategory];
    if (catIndex == indexPath.row) {
        return;
    }
    NSIndexPath *oldIndexPath = [NSIndexPath
indexPathForRow:catIndex inSection:0];

    UITableViewCell *newCell = [tableView
cellForRowAtIndexPath:indexPath];
    if (newCell.accessoryType == UITableViewCellAccessoryNone) {
        newCell.accessoryType = UITableViewCellAccessoryCheckmark;
        self.currentCategory = [taskCategories
objectAtIndex:indexPath.row];
    }

    UITableViewCell *oldCell = [tableView
cellForRowAtIndexPath:oldIndexPath];
    if (oldCell.accessoryType == UITableViewCellAccessoryCheckmark) {
        oldCell.accessoryType = UITableViewCellAccessoryNone;
    }
}
```

4. Managing a selection list—inclusive list

```
- (void)tableView:(UITableView *)theTableView
        didSelectRowAtIndexPath:(NSIndexPath *)newIndexPath {

    [theTableView deselectRowAtIndexPath:[theTableView
indexPathForSelectedRow] animated:NO];
    UITableViewCell *cell = [theTableView
cellForRowAtIndexPath:newIndexPath];
    if (cell.accessoryType == UITableViewCellAccessoryNone) {
        cell.accessoryType = UITableViewCellAccessoryCheckmark;
        // Reflect selection in data model
    } else if (cell.accessoryType ==
UITableViewCellAccessoryCheckmark) {
        cell.accessoryType = UITableViewCellAccessoryNone;
        // Reflect deselection in data model
    }
}
```

Programmatically Selecting and Scrolling

Programmatically selecting a row

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:
(NSIndexPath *)newIndexPath {
    NSIndexPath *scrollIndexPath;
    if (newIndexPath.row + 20 < [timeZoneNames count]) {
        scrollIndexPath = [NSIndexPath
indexPathForRow:newIndexPath.row+20
inSection:newIndexPath.section];
    } else {
        scrollIndexPath = [NSIndexPath
indexPathForRow:newIndexPath.row-20
inSection:newIndexPath.section];
    }
    [theTableView selectRowAtIndexPath:scrollIndexPath animated:YES
                        scrollPosition:UITableViewScrollPositionMiddle];
}
```