

# InterfaceBuilderすごいよ

Hiroyuki-Fujikawa.(cqa02303)

# IBの性質

- MVCモデルの中でViewとControllerを結ぶ
- 操作はGUIで行う
- Xcodeとリアルタイムに連携する

# 欠点

## － 遅い

- ・ ファイル検索と読み込み速度のせい？
- ・ まとめると良いらしいよ

## － 読みにくい

- ・ 独自UIで複数の窓やメニューに分かれてる

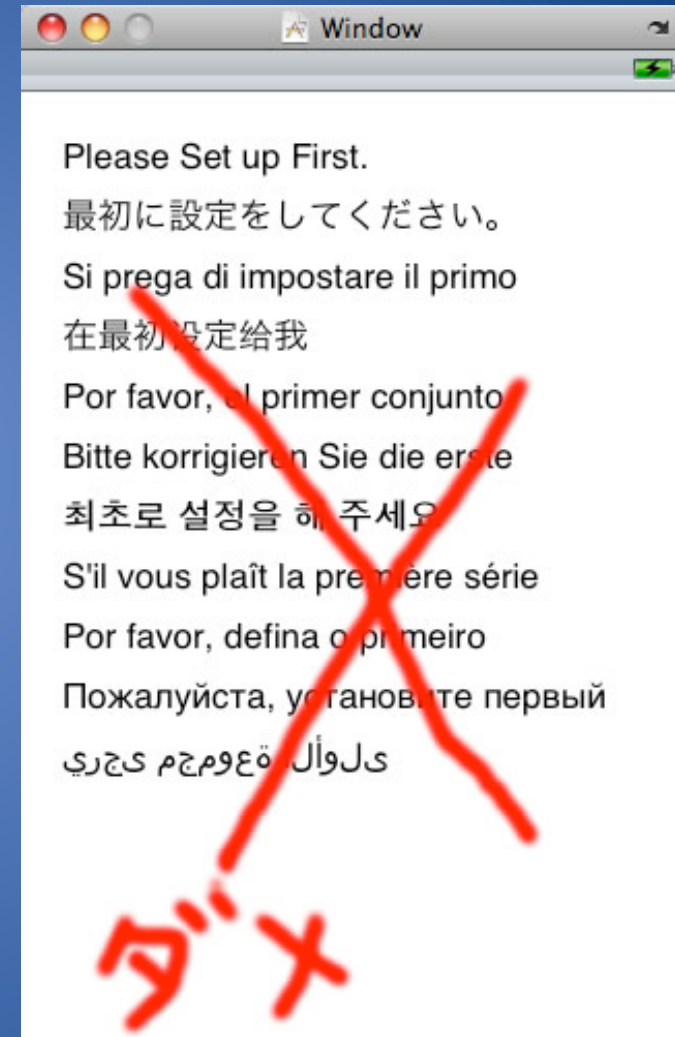
## － でも、利点もあるよ

# InterfaceBuilderのいいところ

- 実行イメージがつかみやすい
- コードをシンプルに保つ
- プロトタイピング時に使える切れの良いナイフ
  - 必要なコードを書く事に集中出来るよ

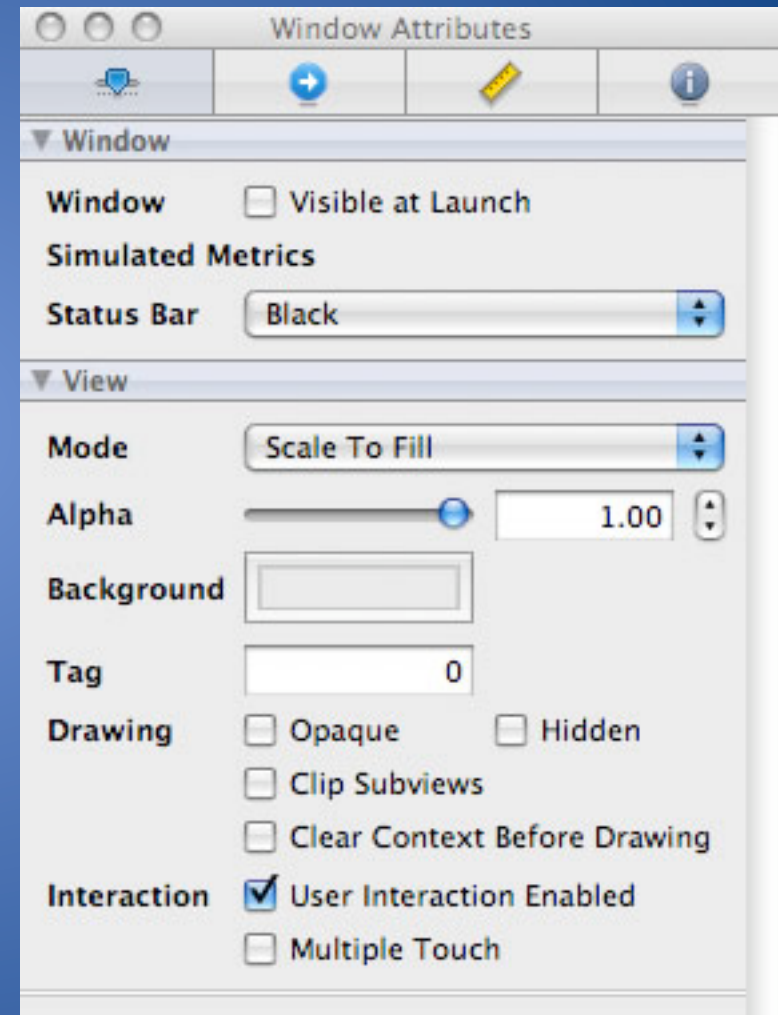
# 国際化もできる

- ダメな例
  - 埋め込み文字列の国際化
    - コードで国際化してください
- 私のおすすめ
  - 言語毎のUI最適化
    - 言語フォントのサイズ差異
    - 文化の違いを吸収



# リソースの作り方、意味付け

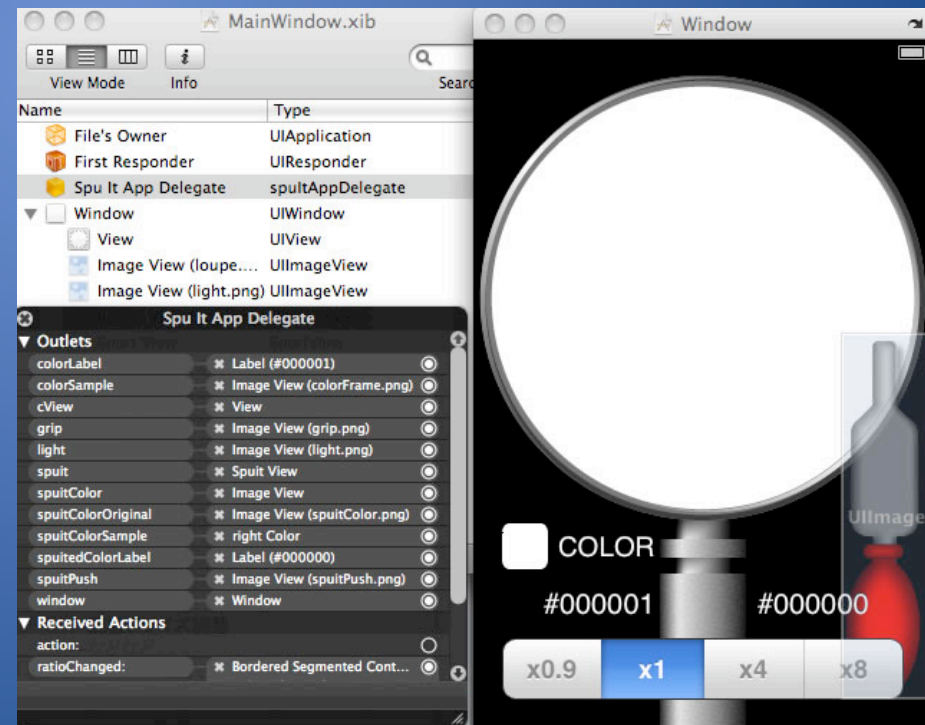
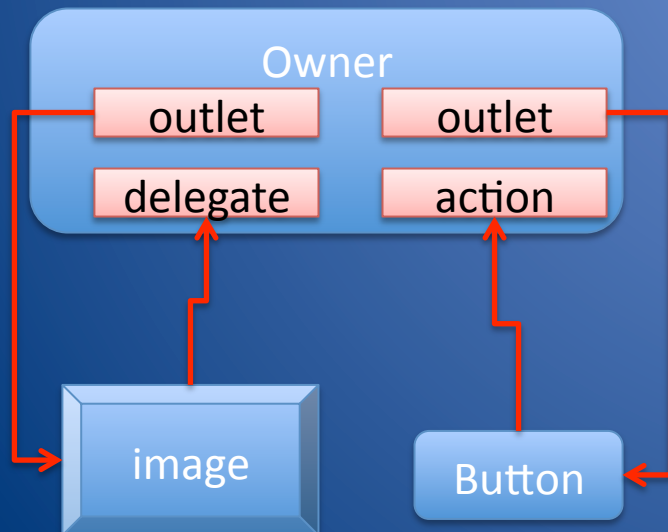
- プロパティを指定する
  - 固有プロパティ
  - 色
  - フォント
  - 画像
  - 配置、サイズ調整



# 動作を指定する

- IBOutletに結びつける
- delegateを設定
- actionを指定

-- 設定イメージ --



# IBOutletについて

- nibロード内で値を代入
  - dealloc時のrelease必要
- Propertyじゃなくて良い
  - object\_setInstanceVariable();で実装？
  - サンプルはpropertyが多いね

```
@interface spuAppDelegate : NSObject <UIApplicationDelegate>
    IBOutlet UIWindow *window;
    IBOutlet UIView *cView;
    IBOutlet UIImageView *light;
    IBOutlet UIImageView *grip;
    IBOutlet UILabel *colorLabel;
    IBOutlet UILabel *spuitedColorLabel;
    IBOutlet UIView *colorSample;
    IBOutlet UIView *spuitColorSample;
    IBOutlet UIView *spuit;
    IBOutlet UIImageView *spuitColor;
    IBOutlet UIImageView *spuitColorOriginal;
    IBOutlet UIImageView *spuitPush;
    NSObject *cameraController;
    int32_t pixels[256]; // image buffer
    CGContextRef ctx;
}
- (void)setSpuitColor;

- (void)action:(id)sender;
- (void)ratioChanged:(id)sender;

@end
```



# 複合的なxib(例1)

- チャトコメッセージCell
  - リサイズ出来る吹出し
  - hiddenプロパティ
- チャトコメッセージ画面
  - Imageリソースの使い回し
  - Imageリソース読み込みの省力化

# (例1)チャトコ

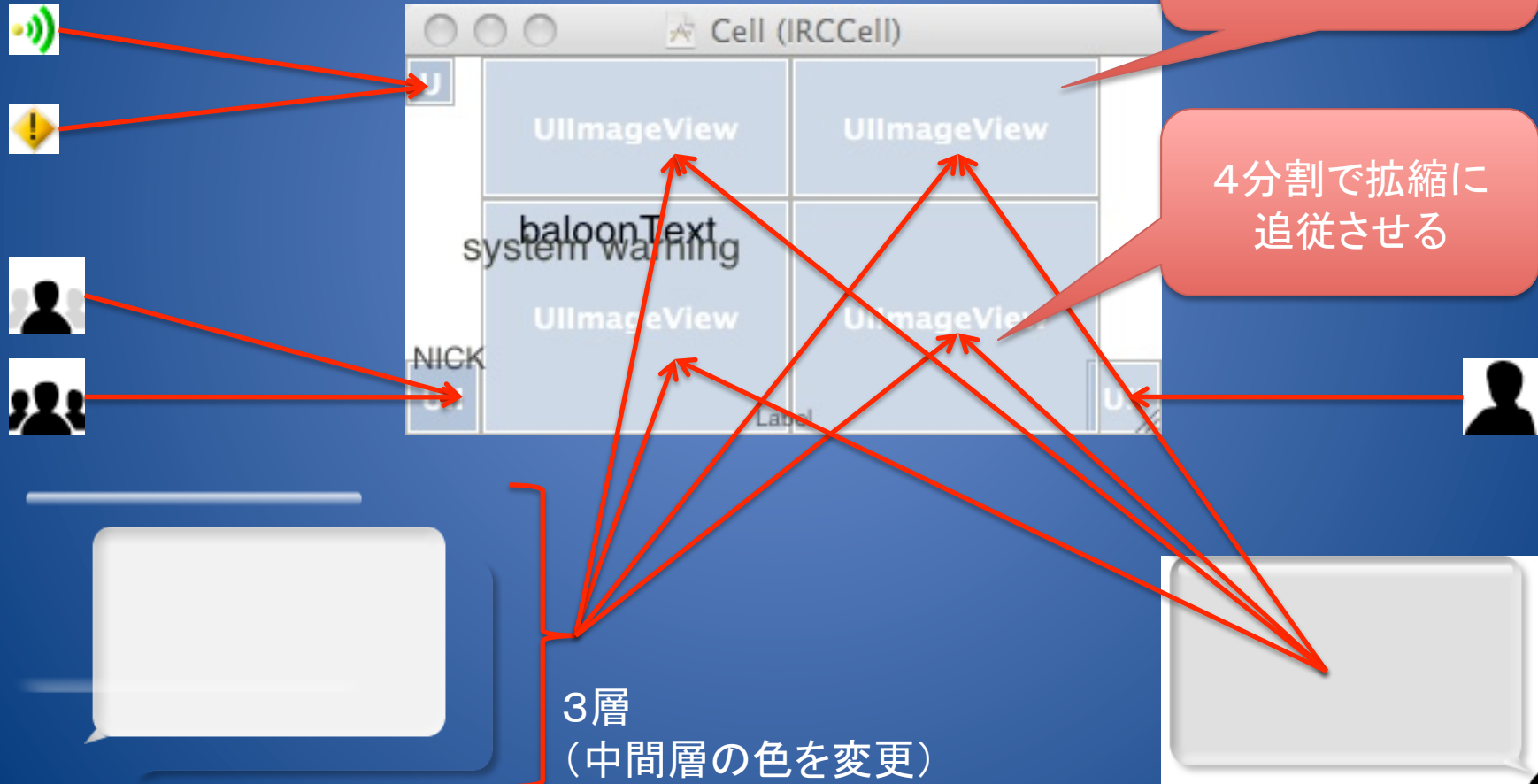
- iChatの画面面白いよね



- ショボイけど、作ってみたYO!

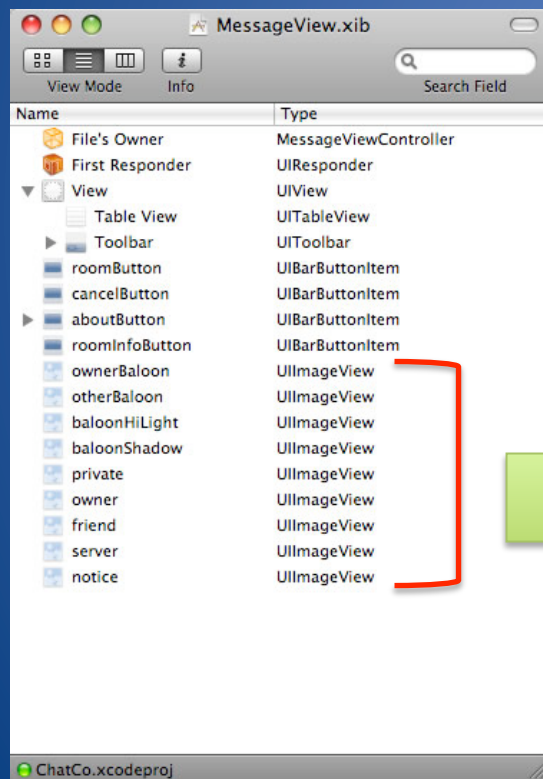
# (例1)チャトコ

- 半分は吹き出しで出来ています

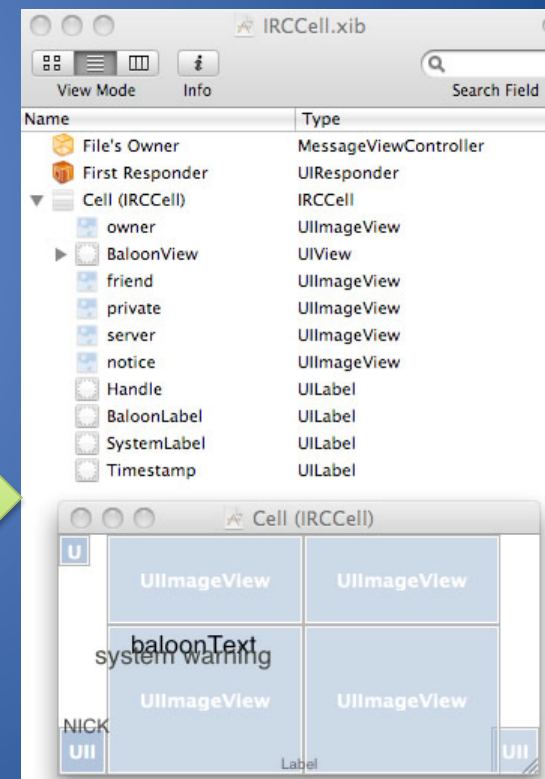
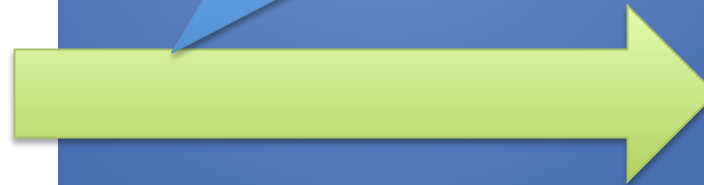


# (例1) チャトコ

- 画像元ネタは別のxibで読み込み(UITableView)



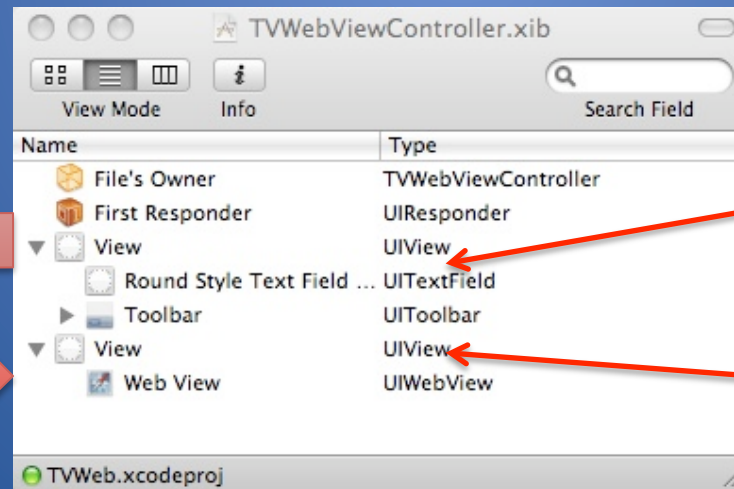
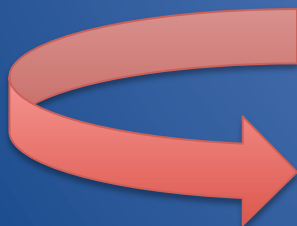
それぞれの  
UIImageView.imageを  
共有するだけ



# 複合的なxib(例2)

- TVWeb(TVでブラウザデモのコード)
  - Windows間でactionを伝達する定義
  - ソースコード記述の省力化
    - 読み返す時、処理にフォーカスしやすい

actionを接続

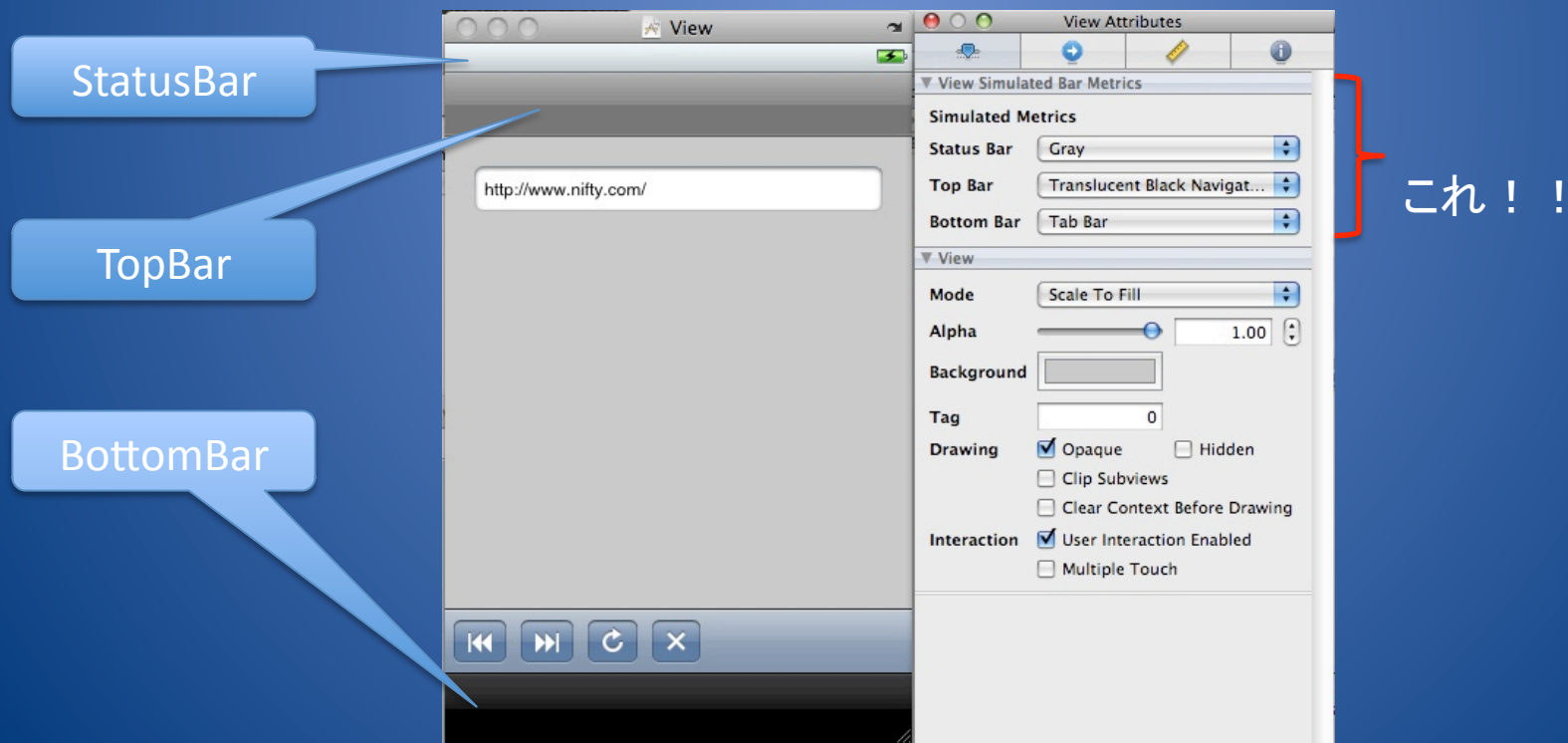


• iPhone用View

• TV出力用View

# IB利用時の注意点(1)

- Simulated Matrix
  - 実行時のViewは変更出来ない。あくまで確認用



## IB利用時の注意点(2)

- Controllerの初期化コード
  - -(void)awakeFromNib;
  - ただし, xib読み込み時の親Controllerだけ

# IBで困った事

- Controllerを使わないでviewを使う場合
  - touchとiPhoneで構造が異なる事がある

※暫定コードで解決

```
NSArray *nibs = [[NSBundle mainBundle]
                 loadNibNamed:@"IRCell" owner:self options:nil];
cell = (IRCell*)[nibs objectAtIndex:nibs.count - 1];
```

- 同じ物の複製が欲しい場合
  - ロード負荷の軽減とかどうよ？
  - NSCoderで出来そう？



ありがとうございました

