

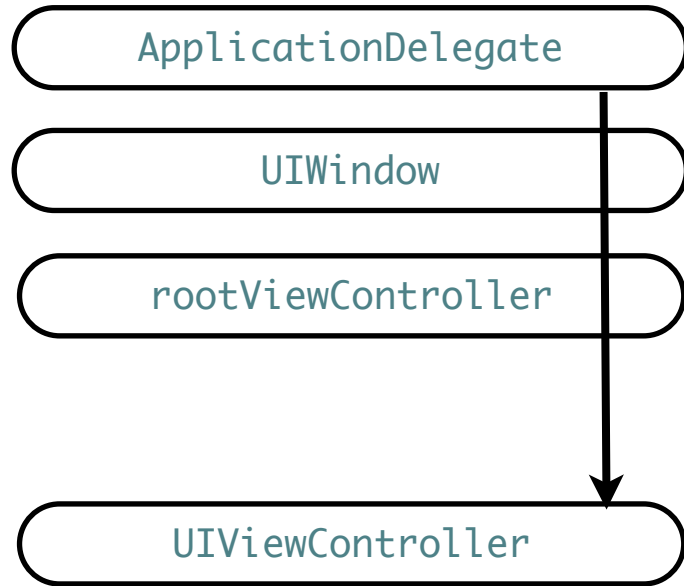
iPhone Game Development

UINavigationController, using UIViewController
UIPickerView, using UIPickerViewDelegate

UIViewController

```
navBarController *NVC=[[navBarController alloc] init];  
self.window.rootViewController=NVC;
```

```
-(id) init{  
    self =[super init];  
    if(self){  
    }  
    return self;  
}
```

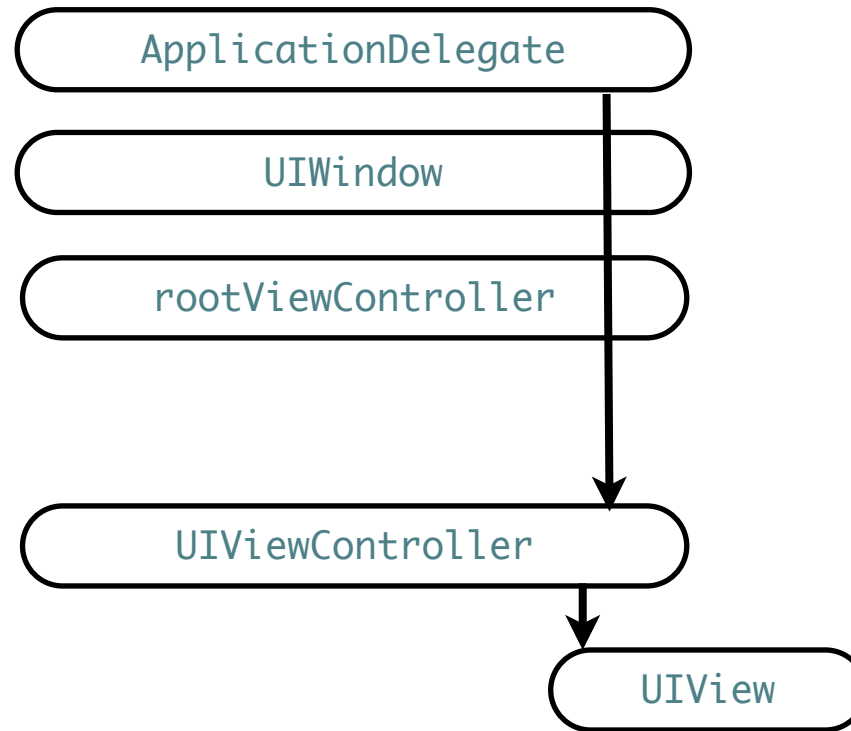


AppDelegate

UIWindow

rootViewController

UIViewController



```
navigationBar *NV=[[navigationBar alloc] initWithFrame:  
                    CGRectMake(0,0,320,460)];  
self.window.rootViewController.view=NV;
```




```
-(void) addNavBarToView:(UIView*) theView{
    UIToolbar *myToolBar=[[UIToolbar alloc] initWithFrame:
        CGRectMake(0,405,320,55)];
    NSMutableArray *myToolBarItems=[[NSMutableArray alloc] init];
    [myToolBarItems addObject:[[UIBarButtonItem alloc] initWithImage:
        [UIImage imageNamed:@"tab_homeicon.png"]
        style:UIBarButtonItemStylePlain
        target:theView action:@selector(doActionA:)]];
    [myToolBar setItems:myToolBarItems];
    [myToolBar setNeedsLayout];
    [myToolBar setNeedsDisplay];
    [myToolBar setBarStyle:UIBarStyleBlack];
    [theView addSubview:myToolBar];
    [theView setNeedsLayout];
    [theView setNeedsDisplay];
}
```

tons of different ways to customize buttons

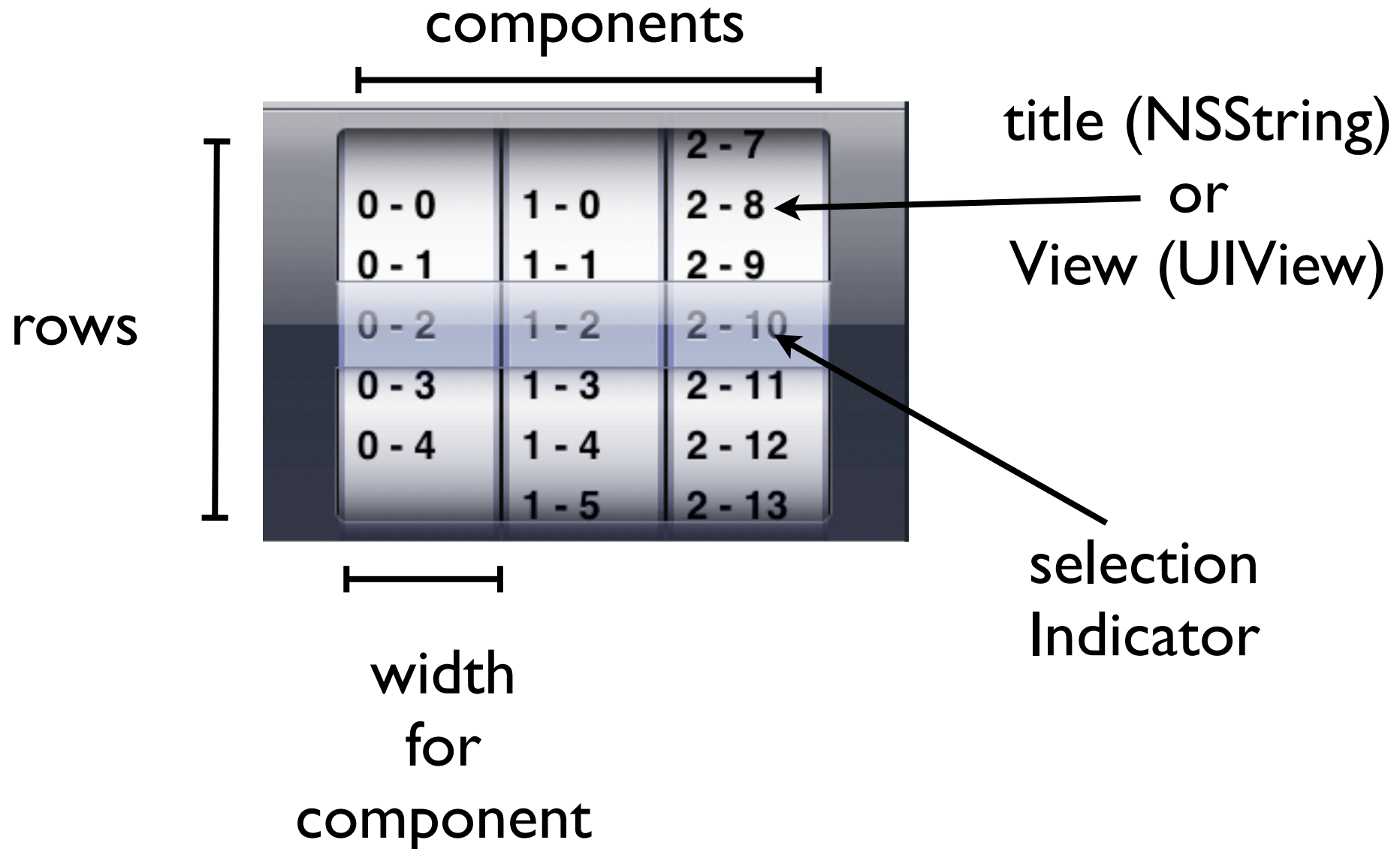
- initWithBarButtonSystemItem:target:action:
- initWithCustomView:
- initWithImage:style:target:action:
- initWithTitle:style:target:action:
- initWithImage:landscapeImagePhone:style:target:action:

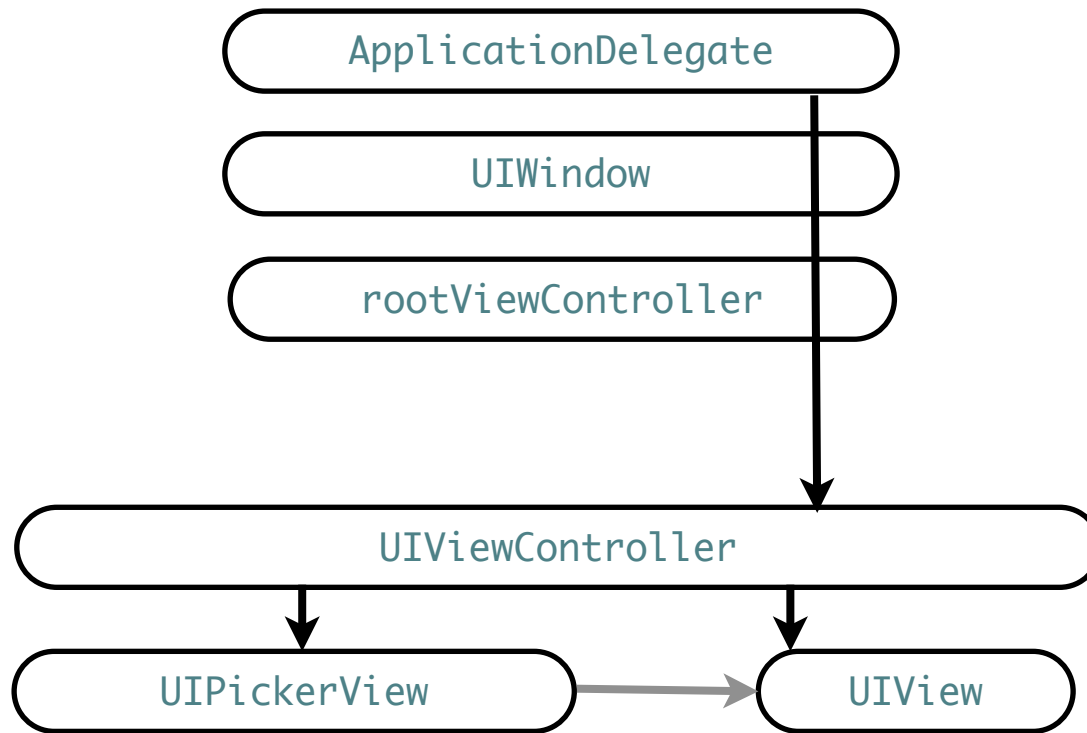
continued

tintColor property

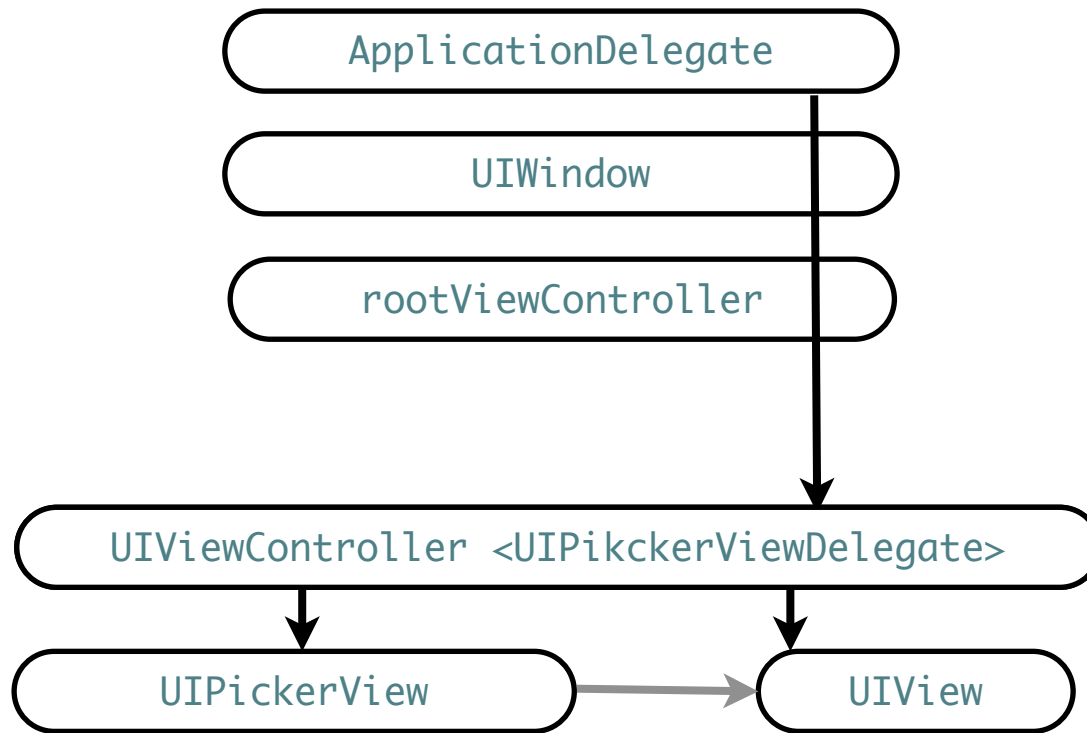
- backButtonBackgroundImageForState:barMetrics:
- backButtonBackgroundImage:forState:barMetrics:
- backButtonTitlePositionAdjustmentForBarMetrics:
- backButtonTitlePositionAdjustment:forBarMetrics:
- backButtonBackgroundVerticalPositionAdjustmentForBarMetrics:
- backButtonBackgroundVerticalPositionAdjustment:forBarMetrics:
- backgroundVerticalPositionAdjustmentForBarMetrics:
- setBackgroundVerticalPositionAdjustment:forBarMetrics:
- backgroundImageForState:barMetrics:
- setBackgroundImage:forState:barMetrics:
- titlePositionAdjustmentForBarMetrics:
- setTitlePositionAdjustment:forBarMetrics:

UIPickerView

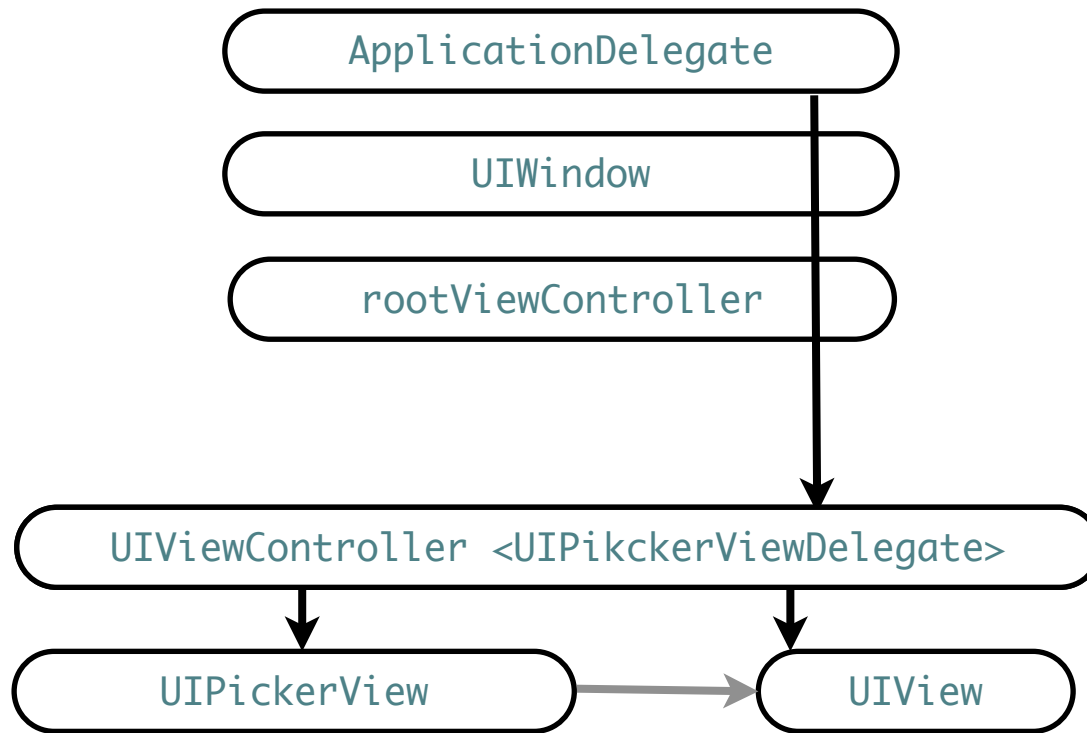




How many rows?
How many columns?
The text?
Maybe a view?



How many rows?
How many columns?
The text?
Maybe a view?



How many rows?
 How many columns?
 The text?
 Maybe a view?

- pickerView:rowHeightForComponent:
- pickerView:widthForComponent:
- optional:
- pickerView:titleForRow:forComponent:
- pickerView:viewForRow:forComponent:reusingView:
- pickerView:didSelectRow:inComponent:

in the controller (which is also the delegate):

Adding the Picker:

```
UIPickerView *P=[[UIPickerView alloc]
    initWithFrame:CGRectMake(0, 480-216, 320, 216)];
P.delegate=self;
P.showsSelectionIndicator=YES;
[theView addSubview:P];
```

implement all of these:

```
-(NSInteger) numberOfComponentsInPickerView:(UIPickerView *) pickerView;

-(NSInteger) pickerView:(UIPickerView*) pView numberOfRowsInComponent:
(NSInteger) component;

-(NSString*) pickerView:(UIPickerView *) pickerView titleForRow:(NSInteger)
row forComponent:(NSInteger) component;

- (CGFloat) pickerView:(UIPickerView *) pickerView widthForComponent:
(NSInteger) component;

-(void) pickerView:(UIPickerView *) pickerView didSelectRow:(NSInteger) row
inComponent:(NSInteger) component;
```

exercise

- use a picker to set background color of a view

