Fredrik Sjöberg
fresj034@student.liu.se
2014-09-21
725G60

Laboration 3 – Förberedelseuppgifter

1. Förklara skillnaden mellan begreppen "frame" och "bounds" för en vy.

Skillnaden mellan en frame och en bound för en vy är att en frame för en vy är en ”structure” som definierar storleken av vyn och dess position i vyns supervy (superview’s coordinate system) medans bounds definierar storleken av vyn och dess position i vyns egna koordinat system (inte superview’s coordinate system)

2. Hur gör man för att skapa en ny vy? Skriv kod som skapar en vy på position (100, 150) med bredden 80 och höjden 90 samt lägger till den som sub-vy till foobar (antag att foobar är en variabel av typen UIView).

För att skapa en ny vy används följande kod:

```
UIView *vy = [[UIView alloc]
initWithFrame:CGRectMake(100, 150, 80, 90)];

[foobar addSubview:vy];
```

3. För att koppla samman komponenter i en storyboard och koden används nyckelordet IBOutlet. Hur används IBOutlet? Betyder nyckelordet någonting speciellt i koden? Hur gör du för att koppla en vy till en property i koden? Vilka attribut bör propertyn ha? Ge exempel på hur en sådan property kan se ut.

Outlets används för att koppla samman ett objekt som refererar till ett annat objekt. Denna referens görs i interface builder med hjälp av IBOutlet tagen så att interface builder vet att propertyn är en outlet kan koppla samman den med interfacet. Ett exempel på hur en sådan property kan se ut:

```
@property (weak) IBOutlet NSArray *keywords;
```

Här används attributet weak vilket är ett måste för att hindra strong reference cycles.

4. En vy-kontroller har en stark referens till sin huvudvy (view propertyn). Placerar man en knapp i den vyn och vill koppla knappen till koden ska den kopplas med en svag referens. Förklara skillnaden utifrån minneshantering.

Som jag nämnde ovan vill man använda en svag referens för att inte skapa strong reference cycles. Att inte använda en svag referens kan skapa strong reference cycles (retain cycles) eftersom objektet inte tas bort från minnet, vilket innebär problem när minnet tar slut.

5. På samma sätt som du kopplar samman en vy med en property så kopplar du ihop händelser med metoder, så att till exempel en viss metod anropas när en knapp trycks ned. För att göra detta i Interface Builder används nyckelordet IBAction. Hur används IBAction? Betyder nyckelordet någonting? Hur gör du sammankopplingen mellan en händelse och en metod? Beskriv hur du gör både via Interface Builder och i kod. Vilken händelse används vanligen för ett vanligt tryck på skärmen?

IBAction används för att deklarera metoder som reagerar på ”actions” i interfacet där IBAction använder sig av sender parametern som skickar en referens till objektet i fråga.

```
- (IBAction)doSomething:(id)sender;
```

Det är enkelt att göra en koppling mellan en händelse (action) och en metod i xcode, genom att högerklicka på ett objekt i interface buildern och sedan dra en pil till metoden i koden som ska hantera händelsen. Om man inte använder interface buildern och vill göra så en knapp reagerar på ett tryck med endast kod behövs följande kod

```
[myButton addTarget:self action:@selector(myMethod:)
forControlEvents:UIControlEventTouchUpInside];
```

Touch up inside heter den händelse som vanligen används för ett tryck på skärmen

6. Vilken metod i UIViewController använder du i allmänhet för att initiera egna variabler och data?

Den heter:

```
viewDidload
```

7. När en iPhone roteras är det möjligt att få användargränssnittet att rotera med och övergå till ett landskapsformat eller porträttformat, beroende på hur telefonen ursprungligen hölls. När telefonen roteras kommer den vy-kontroller som för närvarande visas få veta att rotationen skett och avgöra om rotationen är tillåten (alltså om skärmen ska rotera med). Vilken metod i UIViewController måste du överlagra om du vill att en vy ska kunna roteras? Hur kan du skriva om du vill att porträtt och landskapsläge (båda höger och vänster), men inte upp-och-nervänt porträttläge ska hanteras av en vy-kontroller? Ge kod för hela metoden.

```
-(NSUInteger) supportedInterfaceOrientations {
return UIInterfaceOrientationMaskLandscapeRight |
UIInterfaceOrientationMaskLandscapeLeft;
}
```