

Using More Object-Oriented Programming

Marc Zeedar
Publisher and Editor
xDev Magazine
& **xDevLibrary**



Audience Poll

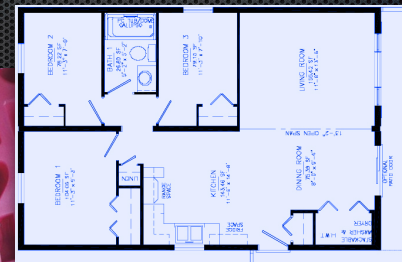
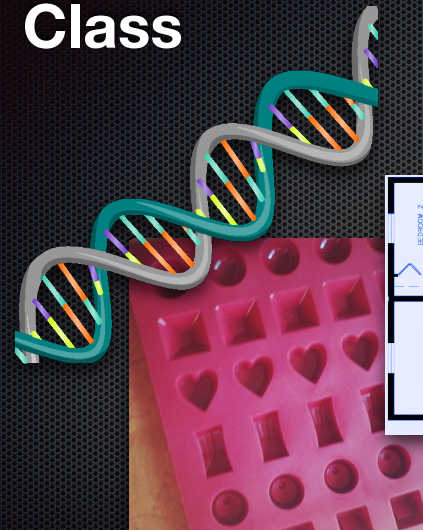
Why Don't You Use More
Object-Oriented Programming?



OOP Basics



Class



Instance (object)



Class vs. Instance

Class	Instance
DNA	Life Form
Die / Mold	Manufactured Item
Blueprint	House



Tip: Class Naming

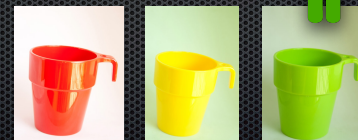
Confusing Way:

```
dim anApple as new apple
```

Clear Way:

```
dim apple as new AppleClass  
dim apple as new AppleType
```

Object's Properties



Class vs. Instance vs. Properties

Class	Instance	Properties
DNA	Life Form	Species, unique characteristics, etc.
Blueprint	House	Family in the house, carpet color, etc.
Die / Mold	Manufactured Item	Color, materials, etc.

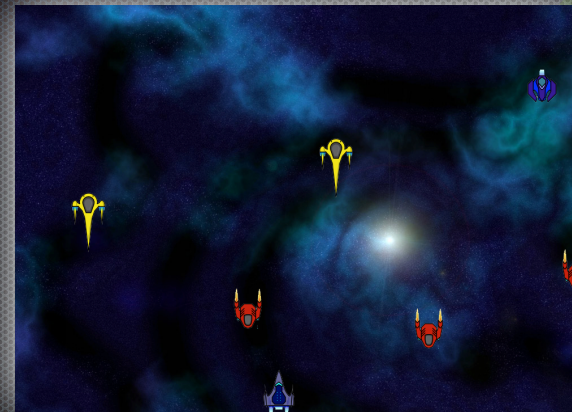
Object's Methods



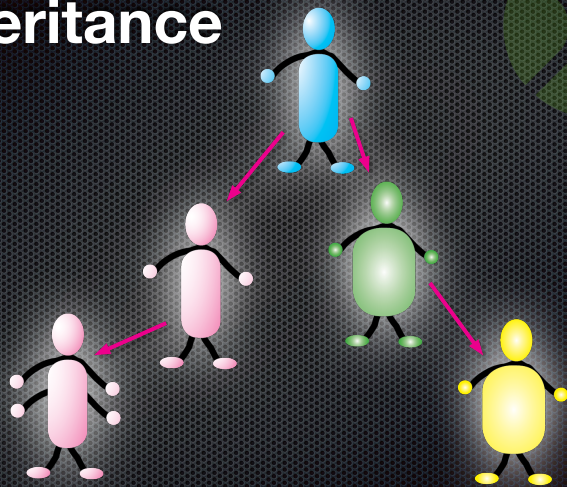
Object's Events



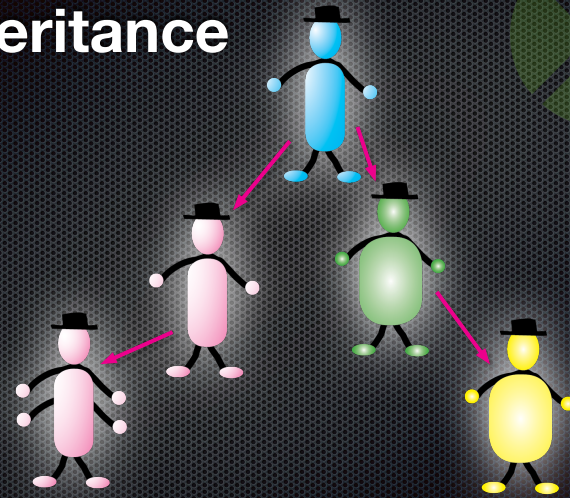
Demo: Invaders



Inheritance



Inheritance



Class Interfaces



Limited:

```
function isAnimalDeclawed(animal as  
AnimalClass) as boolean
```

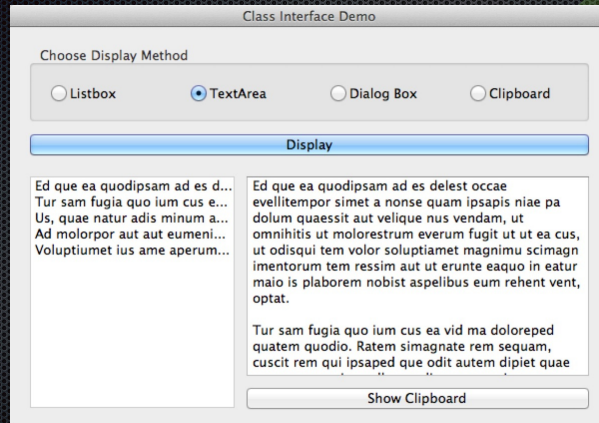
or

```
function isAnimalDeclawed(cat as  
CatClass) as boolean
```

Better (using a Class Interface):

```
function isAnimalDeclawed(animal as  
hasClawsType) as boolean
```


Demo: Interfaces



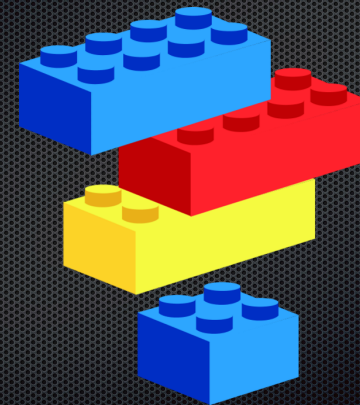
pbDisplay.Action

```
Sub Action()  
    dim aDisplay as displayInterface  
  
    if rbList.value then aDisplay = listBox1  
    if rbTextArea.value then aDisplay = textArea1  
    if rbClipboard.value then aDisplay = new  
        displayableClipboardClass  
    if rbDialog.value then aDisplay = new  
        displayableMsgBoxClass  
  
    aDisplay.display(app.kDemoText)  
End Sub
```

Benefits of OOP



Reusable Code



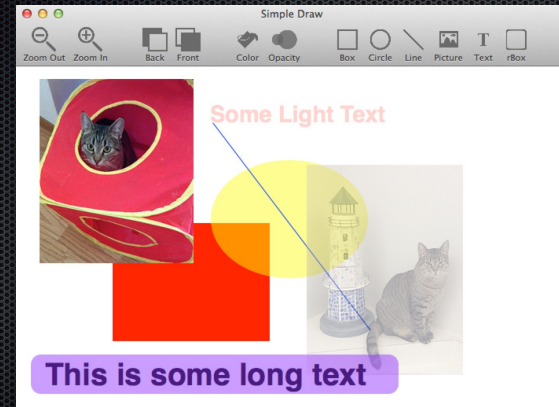


Tip: Build Standalone Modules

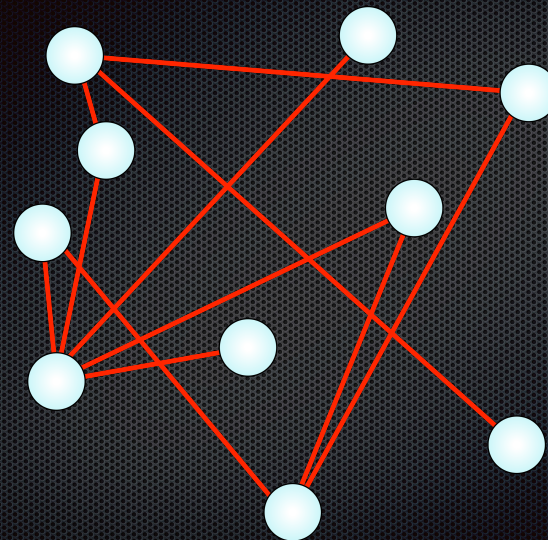
Create modules as separate Xojo projects

- Faster compiling
- Easier debugging
- Less distracting
- Ensure it's really independent

Demo: Expandable Code



Decoupled Code



Clarity



Which code is easier to understand?

```
withdraw(accountsList(kSavingAccountType),  
today, 250)
```

```
account.savings.withdrawal(today, 250)
```

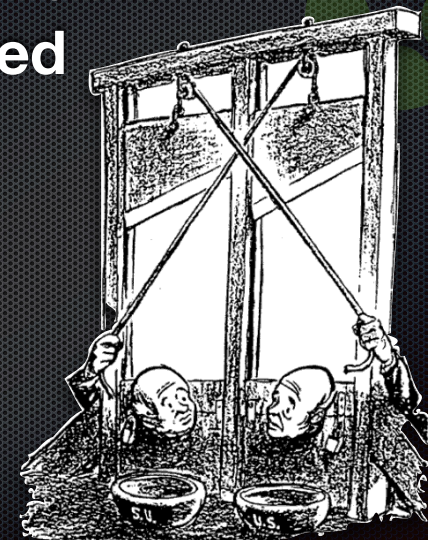
Multiple Methods:

```
convertMarkdownToHTML(text as string, useSmartyPants as  
boolean = false)  
convertMarkdownToHTML(text as string, htmlTemplate as  
string, css as string, useSmartyPants as boolean = false)  
convertMarkdownToXML(text as string, useSmartyPants as  
boolean = false)  
convertMarkdownToXML(text as string, htmlTemplate as string,  
css as string, useSmartyPants as boolean = false)
```

With a Class we have a single Method:

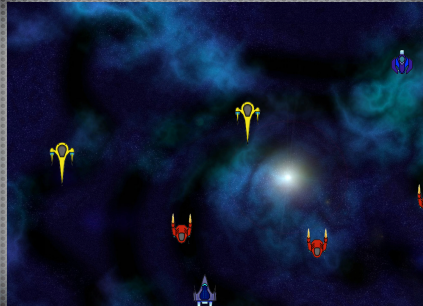
```
Markdown.convert(text as string)  
Markdown.conversionType as integer [Markdown.HTML,  
Markdown.XML, etc.]  
Markdown.useSmartyPants as boolean  
Markdown.template as string  
Markdown.css as string
```

When OOP Gets Complicated



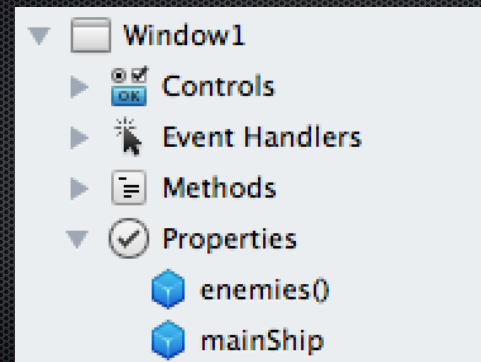
Demo: Invaders2

How do we make the Red enemy ship smarter?



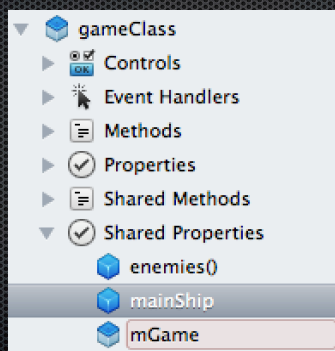
Non-OOP Way:

```
px = window1.mainShip.x
```

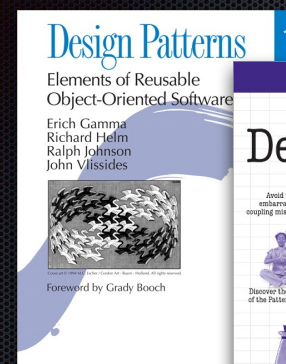


OOP Way:

```
dim game as gameClass = gameClass.getInstance  
dim targetLocation as integer =  
game.mainShipLocation
```



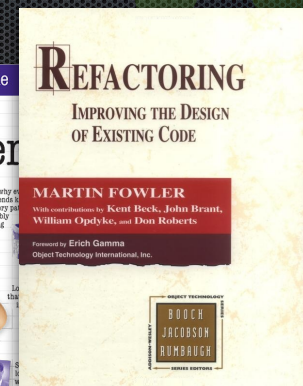
Further Learning



ISBN 0201633612



ISBN 0596007124



ISBN 0201485672

Further Learning

Order printed books, digital subscriptions, etc:

xdevmag.com



Q & A

Marc Zeedar

publisher@xdevmag.com

Example Projects Download:

xdevmag.com/downloads/xdc2018.zip

xdevmag.com/downloads/xdc2018movies.zip

Photo Credits: pdpics.com, flickr.com, wikipedia.com, openciptart.org, publicdomainpictures.net, clipshrine.com, cartoon by Bill Mauldin (1960).