

Chapter Goals

- Define the key terms in **object-oriented design**
- Apply **object-oriented design** methodology to develop a collection of interacting objects to solve a problem
- Discuss the following threads as they relate to problem solving: **information hiding**, **abstraction**, **naming things**, and **testing**

6-2

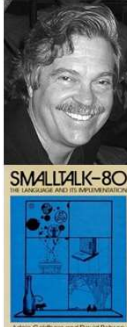
Problem Solving

- Problem solving** The act of finding a solution to a perplexing, distressing, vexing, or unsettled question
- Why we need Object-Oriented Methods?

6-3

Alan Curtis Key & Smalltalk

- 思想萌芽
 - 1968年秋，Key参观MIT人工智能实验室Seymour Papert发明的LOGO语言，看到中学生在使用“海龟绘图”的方式编程，图形界面编程方式给了他深刻的印象。
- “面向对象”的诞生
 - 1971年，Key在施乐PRAC中心开始设计Smalltalk。Smalltalk基于单独个体（即“细胞”）生物学模型来设计的，个体之间可通过彼此发送“信息”交流，来解决问题。
 - 1972年9月，Smalltalk设计完成。Key坚信，“Smalltalk将引发**儿童教育方式的革命**”。并为此开发了图形教学系统。



6-4

对象编程的模样

- Kay请人编写了图形界面的交互式程序，交孩子编程。
- 让我们看一些简单的场景
 - 孩子在屏幕上放了两个Object，一个叫Joe，一个叫Kay
 - 输入：Joe turn 30, 它旋转30度
 - 输入：Joe go 15, 它走15步
 - 当它撞倒Kay是，Kay会说“撞疼我啦！”，Joe会回答“对不起”并停下。
- 为什么可以这样编程？
 - 海龟内部藏了它的行为逻辑。我们只需要告诉它做什么，而不需要知道怎么做
 - 当碰撞（onCollision）事件发生，它们交换消息，并产生用户期望的行为



有心栽花未成，无意插柳成荫。

6-5

Object-Oriented Design

- A problem-solving methodology that produces a solution to a problem in terms of self-contained entities called **objects**
- Object** A thing or entity that makes sense within the context of the problem
For example, a student

6-6

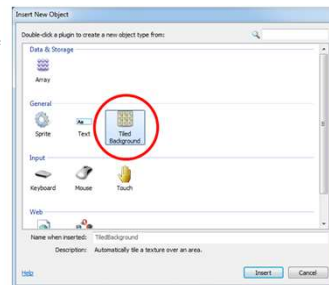
Object-Oriented Design

- A group of similar objects is described by an **object class**, or **class**
- A class contains fields that represent the properties and behaviors of the class
 - A **field** can contain data value(s) and/or methods (subprograms)
 - A **method** is a named algorithm that manipulates the data values in the object

6-7

识别Class

Object Type


<https://www.scirra.com/tutorials/37/construct2/zh-tw/page-1>

6-8

对象属性 (properties)

- 给游戏中添加了一个背景对象
- 对象必须的属性
 - **Name**, 每个对象都有一个唯一的代号
- 它有那些属性呢
 - 角度 (Angle)
 - 尺寸 (Size)
 - 图片 (Texture)
 -

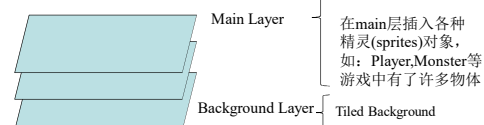
Object Type Properties	
Name	TiledBackground
Plugin	Tiled Background
Common	
Layer	Layer 0
Angle	0
Opacity	100
Position	0, 0
Size	1280, 1024
Edit variables	
Add / edit	

修改属性, 对象产生了对应的变化

6-9

游戏对象及关系

Layout (画布) 对象包含一个或多个层 (Layer) 对象



在main层插入Mouse, Keyboard对象, 该层可以 and 用户交互了

在main层插入各种精灵(sprites)对象, 如: Player, Monster等 游戏中有了许多物体

<https://www.scirra.com/tutorials/37/construct2/zh-tw/page-2>

6-10

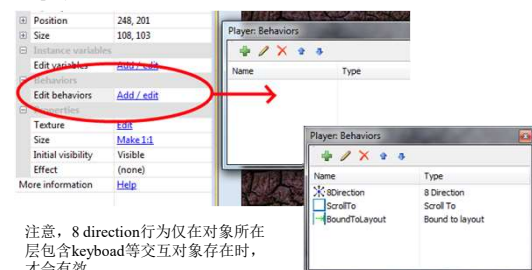
Relationships Between Classes

- Containment
 - “**part-of**”
 - An address class may be part of the definition of a student class
- Inheritance
 - Classes can inherit data and behavior from other classes
 - “**is-a**”

6-11

为对象添加行为 (behaviors / methods)

选择player, 在属性表中添加一些行为 (系统预定义了常用行为)



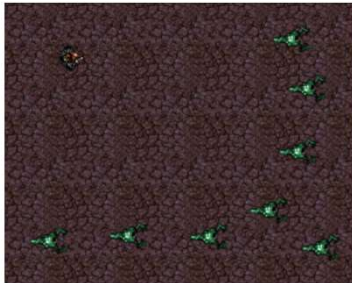
注意, 8 direction行为仅在对象所在层包含keyboard等交互对象存在时, 才会有效

<https://www.scirra.com/tutorials/37/construct2/zh-tw/page-3>

6-12

实例化 (instance)

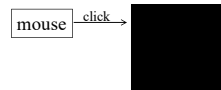
先按教程
理解
Instance



6-13

事件与编程 (1)

- 事件编写指南 (由外向内方法)
 - 将游戏看成一个黑盒子
 - 列出外部对象的事件与目标 ()
 - System
 - Every Tick
 - 将player指向鼠标
 - Mouse
 - Left Button Click
 - 发射子弹
 - Keyboard
 - 实现外部对象的事件
 - System -> Every Tick



<https://www.scirra.com/tutorials/37/construct2/zh-tw/page-4>

6-14

事件与编程 (2)

- 事件编写指南
 - 列举所有sprites的碰撞, 变化条件和目标
 - Player
 - 碰撞Monster
 - 生命值-1
 - 生命值<=0
 - Game over
 - Monster
 - 生命值<=0
 - 销毁
 - Bullet
 - 碰撞Monster
 - 销毁
 - Bullet爆炸

6-15

事件与编程 (3)

- 事件编写指南
 - 用伪代码表示复杂的目标实现过程

我们来添加碰撞事件, 用来消灭敌人。

添加事件如下:

条件: Bullet->On collision with another object->pick Monster.

```
1 | //子弹->于其他对象碰撞->选择Monster怪物。
```

动作: Monster->Destroy

```
1 | //怪物->消灭
```

动作: Bullet->Spawn another object->Explosion, layer 1

```
1 | //在图层1 碰撞的位置产生新对象-爆炸特效
```

动作: Bullet->Destroy

```
1 | //子弹销毁
```

Information Hiding/Abstraction

- Information Hiding** and **Abstraction** are two sides of the same coin.
 - Information Hiding** The practice of hiding the details of a module with the goal of controlling access to the details of the module.
 - Abstraction** A model of a complex system that includes only the details essential to the viewer.

6-17

Information Hiding/Abstraction

- Abstraction is the result with the details hidden
 - Data abstraction** Separation of the logical view of data from their implementation.
 - Procedural abstraction** Separation of the logical view of actions from their implementation.
 - Control abstraction** Separation of the logical view of a control structure from its implementation.

6-18

面向对象问题求解：图书馆找书



9-19

问题求解过程

- 识别问题中的对象和类
- 识别对象和类的
 - 属性
 - 行为
- 研究对象（类）之间的关系
 - 包含（part-of）
 - 继承（is-a）
- 由外向内逐步实现
 - 事件，行为

6-20

问题相关知识与应用场景

- 图书馆的藏书排架方法
 - 读者借书时一定会发现在图书的书脊上有一个标签,标签上有两组号码,上面的一组为分类号,下面的一组为书次号,这两组号码就构成了索书号.索书号是确定一本书排架的依据.
 - 索书号是D452.62/4052。“D”前面的部分是中国图书馆图书分类法的分类号。其中，“D”表示“政治、法律”大类，后面的数字是细分的小类。“/”后面的数字应该是表示这种书是该图书馆的“政治法律”类图书的第4052种。
- 在藏书架上查书的基本操作
 - 用书名查找图书
 - 用作者查找图书
 - 用索书号查找图书

9-21

对象 / 属性 / 方法

- 领域对象（domain objects）
 - CLASS Bookshelf
 - book index: from-number, to-number
 - container: sorted books by book-index
 - CLASS Book
 - name
 - author
 - book-index
- 行为集
 - FindByName (book-name)
 - FindByAuthor (author-name)
 - FindByIndex (book-index)

6-22

总结面向对象的设计过程


- 识别游戏中的sprites
 - 列出sprite的名称
 - 识别sprite的属性（变量）
 - 识别sprite的行为
- 在游戏中添加背景，层，Input对象（建关系）
 - 添加Input对象
 - 添加sprites对象
- 将游戏看成黑盒子，从外向内编程
 - 主体—条件—goal—实现
 - 复杂行为或事件使用伪代码表示

6-23

游戏设计讨论



6-24



作业

- 根据后面的素材，识别类和对象。
阅读“牧师与恶魔过河”（<http://www.17yy.com/f/69854.html>）的游戏介绍并玩游戏。然后回答问题
 - 游戏涉及哪些类。请列表说明
 - 游戏中有哪些对象，各几个。例如，船（一个）
 - 类和对象的区别是什么？举一个例子说明
 - 游戏中的魔鬼，有哪些属性和方法？
 - 假设魔鬼被鼠标点中，会执行onclick事件，请用文字（伪代码）描述这个事件中魔鬼与其他对象沟通的过程。
 - 类或对象会是动词吗？

6-25