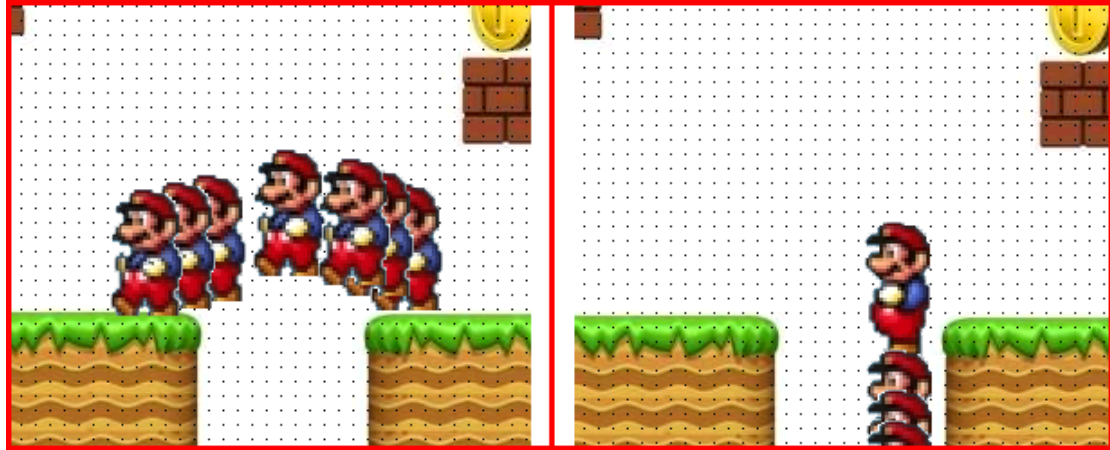


第4课 操控自如——键盘控制

任务导航

玩过电脑游戏的人都体验过用键盘控制游戏角色的快感和乐趣，而这种控制又是如何实现的呢？

本课我们将通过玛丽行走过程中越崖的控制来体验、学习用键盘控制游戏角色的方法。我们还将把玛丽游戏中的行走、坠落、越崖等分解动作有机地组合成一个初级的玛丽游戏如图 4.1 所示，体会自己编写游戏程序的乐趣。



a 按下空格键越崖

b 崖边未及时按下空格键坠落

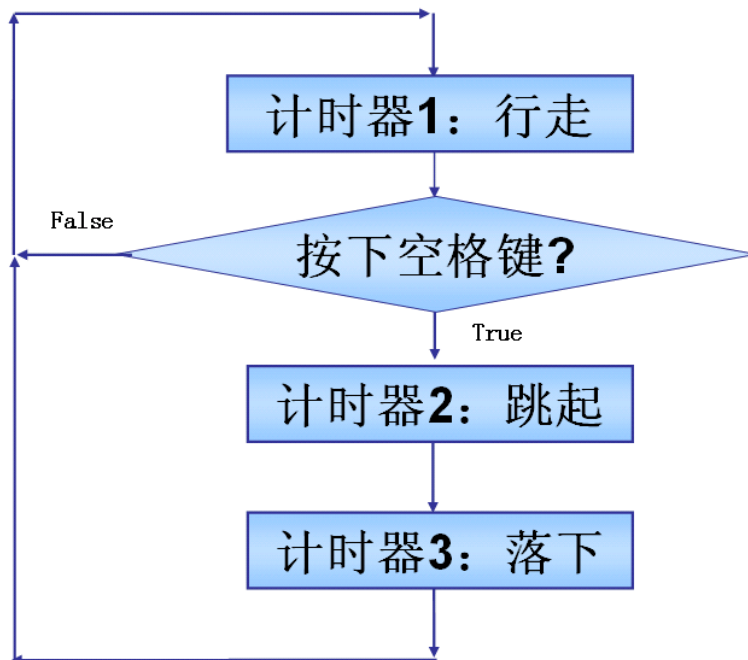
4.1 键盘控制效果

Follow Me

任务分析

不难看出，“我的玛丽游戏”是由三个动作组成的，即“行走”、“越崖”和“坠崖”，而这三个动作我们是通过四个计时器的 Timer 事件实现的。

当在玛丽行走的过程中按下空格键时玛丽跳起后落下的程序执行步骤可以用流程表示如图 4.2 所示。



4.2 “玛丽”行走、跳崖的程序流程

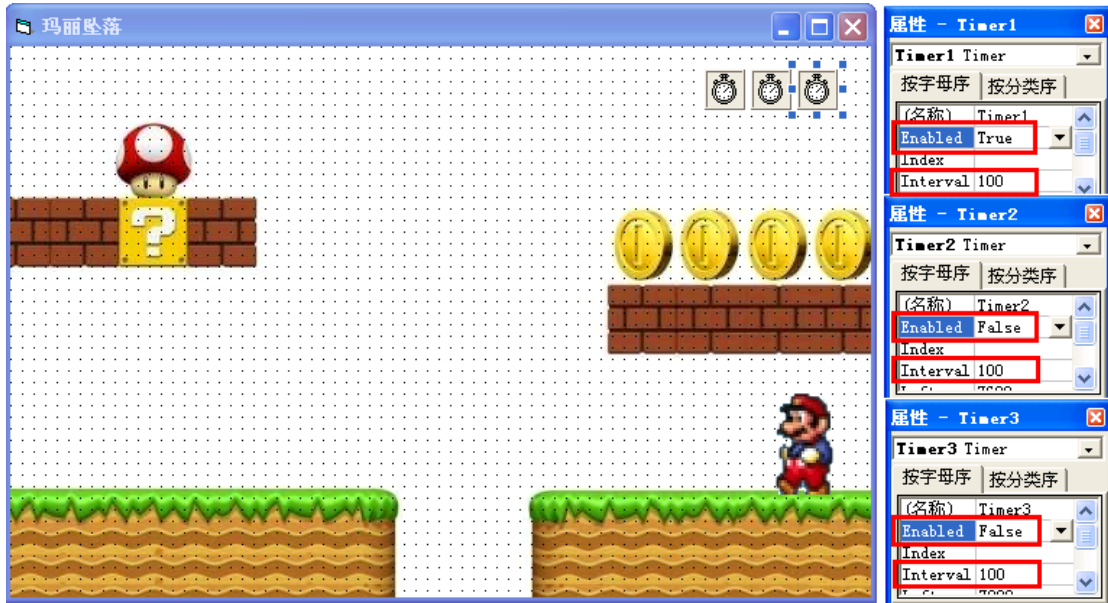
1. 玛丽行走、跳跃的程序设计

在窗体上如图 4.3 所示，设置三个计时器，分别为：

Timer1 功能：行走的控制

Timer2 功能：按下空格键后跳起的控制

Timer3 功能：跃起后落下、行走的控制



4.3 添加三个计时器并设置属性

2. 玛丽行走、越崖的程序设计

步骤 1：编写“玛丽行走”的事件过程代码。

DIY

填写向左移动的代码

Rem 玛丽行走

```
Private Sub Timer1_Timer()
```

```
_____ ' 向左移动
```

```
End Sub
```

步骤 2：编写“玛丽越崖”的事件过程代码

我们可以将“玛丽越崖”的动作分为两段程序代码来实现，即跳起和落下计时事件过程。

DIY：读懂下面的程序

Rem 玛丽跳起

```
Private Sub Timer2_Timer()
```

```
If Image1.Top > 3200 Then
```

```
Image1.Left = Image1.Left - 150 ' 向左移动
```

```
Image1.Top = Image1.Top - 100 ' 向上移动
```

```
Else
```

```
Timer2.Enabled = False ' 跳起计时器不可用
```

```
Timer3.Enabled = True ' 落下计时器可用
```

```
End If
```

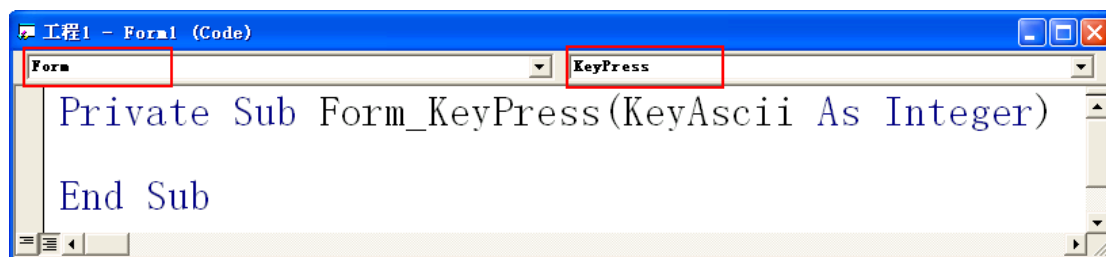
```
End Sub
```

Rem 玛丽落下

```
Private Sub Timer3_Timer()  
    If Image1.Top <= 3550 Then  
        Image1.Left = Image1.Left - 150    ' 向左移动  
        Image1.Top = Image1.Top + 80      ' 向下移动  
    Else  
        Timer3.Enabled = False          ' 落下计时器不可用  
        Timer1.Enabled = True           ' 行走计时器可用  
    End If  
End Sub
```

步骤 3: 设置“键盘控制”的事件过程

从“行走”到“越崖”的动作是通过按下空格键来连接运行的,即玛丽在行走过程中系统如果发现按下了“空格键”,则玛丽越崖,越崖落地后玛丽继续行走。在键盘上按下空格键的事件过程如图 4.4 所示。



4.4 敲击键盘触发的事件过程

智多星

计算机键盘使用的 ASCII 码是目前计算机中用得最广泛的字符集编码。键盘上的每一个字符都对应一个 ASCII 码,以上事件过程中的 KeyAscii As Integer 为按下的字符键定义了一个存放该键字符 ASCII 码的整型变量。当按下键盘上的某个字符键时,事件被触发,程序将获取的按键信息,即按下字符键的 ASCII 码存入整型变量 KeyAscii 中,以便在该事件过程中使用。

例如当按下空格键后,空格键对应的 ASCII 码 32 就被存放于 KeyAscii 中了。

步骤 4: 编写按下空格键的事件过程代码

Rem 键盘控制

```
Private Sub Form_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 32 Then    ' 是否按下空格键  
        Timer2.Enabled = True    ' 跳起计时器可用  
        Timer1.Enabled = False    ' 行走计时器不可用  
    End If  
End Sub
```

以上事件过程被按下的键触发后执行,程序执行时首先对按下的键是否为空格键做出判断,如果按下的键是空格键,则让跳起动作生效,行走动作失效。从而实现了行走到跳跃动作的转换。

DIY

将以上四个事件过程写在一个窗体文件中进行调试:

```
Private Sub Form_KeyPress(KeyAscii As Integer)    ' 键盘控制
```

```

End Sub
Private Sub Timer1_Timer() ' 玛丽行走
    _____
End Sub
Private Sub Timer2_Timer() ' 玛丽跳起
    _____
End Sub
Private Sub Timer3_Timer() ' 玛丽落下
    _____
End Sub

```

调试程序的过程中需要注意，同一时刻只应有一个计时器设置为可用，其它设置为不可用。每个计时器触发的时间间隔不能为 0。否则可能会出现一些不给出错误提示（也称逻辑错误）的错误。

3. 玛丽“行走”、“越崖”与“坠崖”的程序设计

任务分析：

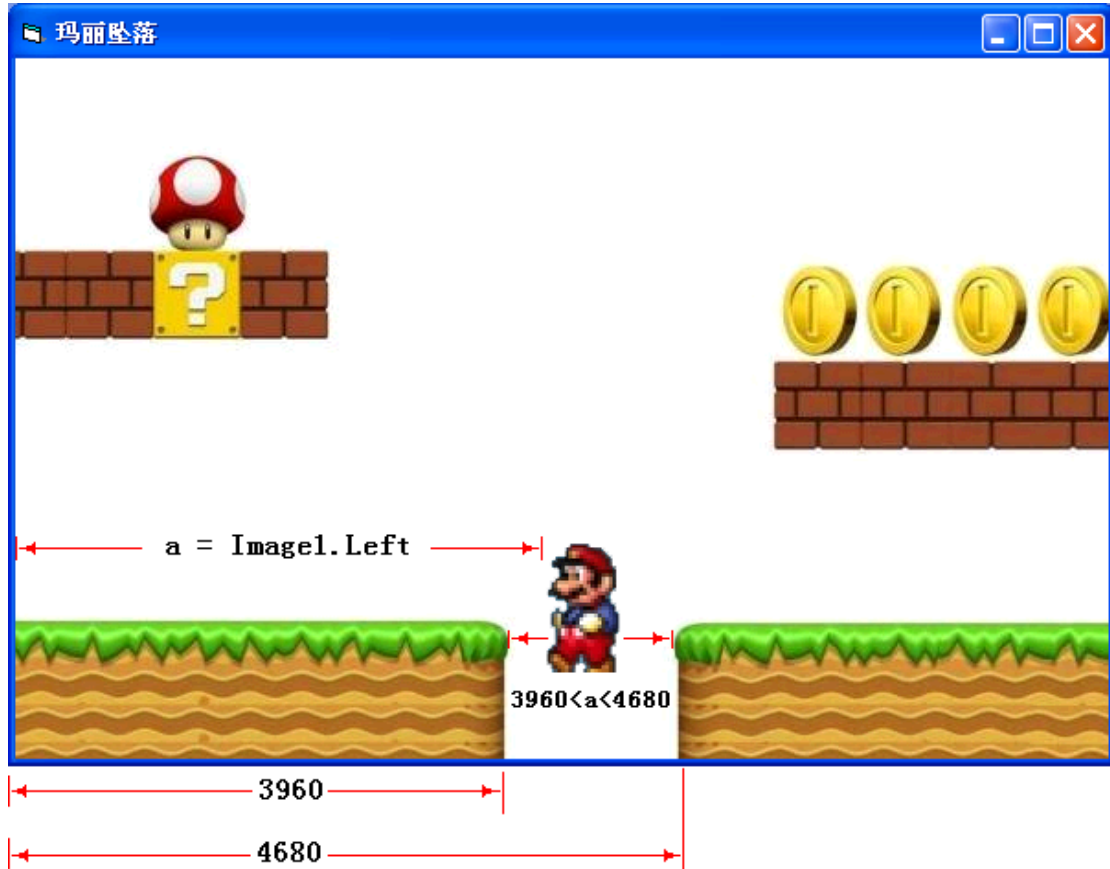
至此为止，玛丽已经可以在行走的过程中完成“越崖”的动作了，但是当按下空格键的时间滞后或提前，都有可能使玛丽坠入悬崖，为了使初级玛丽的游戏更加好玩，我们有必要将玛丽坠崖的动作加入到以上游戏过程中。

```

Rem 坠崖
Private Sub Timer4_Timer()
    Image1.Top = Image1.Top + 150 ' 坠落
End Sub

```

不难看出玛丽坠入悬崖的可能性有两个，其一：玛丽走到悬崖边时由于没有跳起而落入悬崖中，其二：玛丽落下时落在悬崖中。所以判断玛丽行走和落下时是否在悬崖的范围内是解决问题的关键。如果玛丽在悬崖范围内，如图 4.5 所示；则令玛丽坠崖的计时器可用，其它计时器不可用；否则令玛丽继续行走。



4.5 悬崖范围示意图

根据以上分析可以在玛丽行走的 Timer1 的计时事件过程中加入玛丽是否走到悬崖范围的判断，并且根据判断结果做出是否从行走动作转换为坠崖动作的处理，Timer1 的事件过程代码可修改如下：

Rem 玛丽行走

```
Private Sub Timer1_Timer()
```

```

    a = Image1.Left           ' 玛丽到左边框的距离
    If a > 3960 And a < 4680 Then ' 玛丽处于悬崖范围的判断
        Timer1.Enabled = False ' 行走计时器不可用
        Timer4.Enabled = True  ' 坠落计时器可用
    End If
    Image1.Left = Image1.Left - 200 ' 向左移动

```

```
End Sub
```

DIY

参照 Timer1 的修改，将 Timer3 修改为能够对玛丽是否处于悬崖范围做出判断，并能从越崖落下后转换到坠崖的动作。

运行调试修改后的程序，实现行走中按空格键的跳跃、行走中的坠崖和跳起后落下时坠崖的组合动作。

4. 优化我的玛丽游戏

有 Timer1 行走、Timer2 跳起、Timer3 落下和 Timer4 坠落，四个基本动作组成的初级玛丽游戏，还可以做以下简单的优化。

优化 1: 实现玛丽行走的动画效果

分析：前面代码中的 Image1 是一张玛丽迈步动作的图片，程序运行时动作单一，没有走路的感觉，为此我们可以将另外一张玛丽直立的图片 Image2 载入，将两张图片重合放在同一起始位置，程序运行时图片交替显示、隐藏，使玛丽在行走的过程中具有动画的视觉效果。

步骤 1：添加名称为 Image2 的图片框，并载入图片“玛丽 2.Gif”的图片。

步骤 2：将 Image2 置于起始位置与 Image1 位置重合。

步骤 3：修改用于行走功能的事件过程 Timer1_Timer() 如下（无坠崖判断）

Rem 玛丽行走动画

```
Private Sub Timer1_Timer()
    If Image2.Visible = True Then      ' 可见性判断
        Image2.Visible = False        ' 直立玛丽图像不可见
        Image1.Visible = True         ' 迈步玛丽图像可见
    Else
        Image1.Visible = False        ' 迈步玛丽图像不可见
        Image2.Visible = True         ' 直立玛丽图像可见
    End If
    Image1.Left = Image1.Left - 200   ' 迈步玛丽左移
    Image2.Left = Image2.Left - 200   ' 直立玛丽左移
End Sub
```

DIY

- 读懂修改后的 Timer1 事件过程代码。
- 将具有“行走动画”效果的 Timer1_Timer 的事件过程与具有“坠崖判断”行走的 Timer1_Timer 事件过程合二为一，使修改后的 Timer1_Timer 的事件过程既具有“坠崖判断”功能又具有“行走动画”的视觉效果
- 重新调试初级玛丽游戏

优化 2: 通过单击窗体实现初级玛丽游戏的重新运行

当单击窗体时让玛丽图像回到游戏的初始位置，并设置好四个计时器的初始设置。关于窗体的单击事件过程代码如图 4.6 所示。

```
Private Sub Form_Click()      ' 单击窗体恢复游戏初始运行
    Image1.Left = 7800
    Image2.Left = 7800
    Image1.Top = 3550
    Image2.Top = 3550
    Timer1.Enabled = True
    Timer4.Enabled = False
    Timer2.Enabled = False
    Timer3.Enabled = False
End Sub
```

4.6 关于窗体的单击事件

DIY

综合调试“初级玛丽”游戏程序，使游戏实现以下功能：

- 具有“动画行走”效果的移动；
- 按下空格键的跳跃；
- 运动到悬崖位置后的坠落；
- 单击窗体后恢复初始运行状态；
- 行走至窗体左端消失后从窗体右端出现。

知识着陆

在“初级玛丽”的游戏程序设计中，采用了“分解、综合”的设计思路，即分解设计基本动作，综合调试程序功能。

利用按下键时触发的事件过程，实现了键盘对游戏角色的控制。利用窗体的单击事件过程，实现了游戏的初始化的功能。

关于窗体的按下键的事件过程如下：

```
Private Sub Form_KeyPress(KeyAscii As Integer)
    ' 程序代码内容
End Sub
```

关于窗体的单击事件过程如下：

```
Private Sub Form_Click()
    ' 程序代码内容
End Sub
```

视野拓展

ASCII 码

ASCII 码是美国信息互换标准代码(American Standard Code for Information Interchange)的简写，是基于英语的一种编码方式，用于计算机的信息传输。

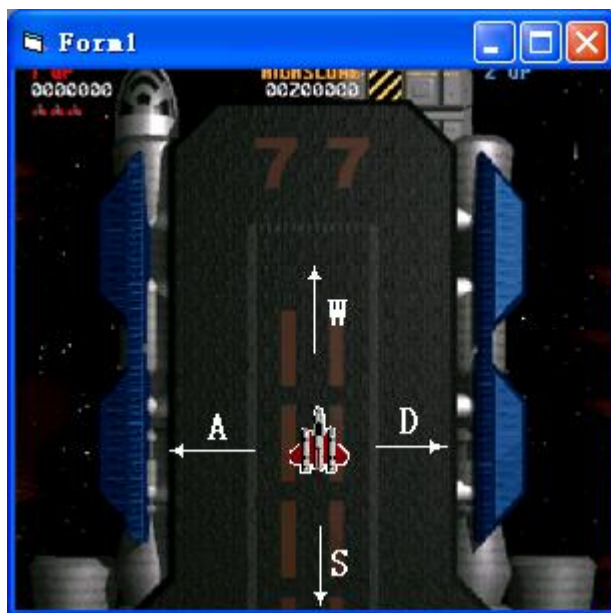
ASCII 共定义了 256 个代码(从 0-255)，从 0-32 位为控制字符(ASCII control characters)，从 33-127 位为可打印字符(ASCII printable characters)。从 0-127 是标准的 ASCII 编码，从 128-255 是扩展的 ASCII 编码。

常见字符的 ASCII 编码

十进制数	说明	十进制数	说明
8	退格	65	A
9	水平制表符	66	B
10	换行键	67	C
13	回车键	68	D
32	空格	69	E
33	!	70	F
34	"	71	G
35	#	72	H
36	\$	97	a
37	%	98	b
38	&	99	c
48	0	100	d
49	1	101	e
50	2	102	f
51	3	103	g
52	4	104	h

创作天地

1. 修改玛丽行走的程序，使玛丽能“蹦跳行走”。
2. 编程，使分别按下小写 a、d、w、s 键时飞机分别向左、右、上、下移动，如图 4.7 所示。



4.7 键盘控制移动

3. 修改以上程序，使分别按下 a、d、w、s 大小写键都能使飞机向左、右、上、下移动。

部分键盘字符 ASCII 码表			
按键	大写	小写	控制
A	65	97	左移
D	68	100	右移
W	87	119	上移
S	83	115	下移

表 4.1

4. 修改以上程序，使分别按下 a、d、w、s 大小写键都只能在窗体内移动小飞机。
5. 编个程序在标签框中显示按下的字符键的 ASCII 码。