VBugs Worksheet 7

|  |
| --- |
| **Name:** |
| **Year Level: SOLUTIONS** |

**Answers to Part 1**

*Exercise 1: Creating fields and a property*

1. In you Bug class, create AliveSprite, DeadSprite and Alive fields and a property for Alive field.

|  |
| --- |
| Answer:  Public Class Bug  Private Alive As Boolean  Private AliveSprite As Sprite  Private DeadSprite As Sprite  Public Property IsAlive() As Boolean  Get  Return Alive  End Get  Set(ByVal value As Boolean)  Alive = value  End Set  End Property  End Class |

*Exercse 2: Creating a constructor*

1. Ceate a costructor for your Bug class.

|  |
| --- |
| Answer:  Public Class Bug  …  Public Sub New()  Alive = True  If IsAlive Then  AliveSprite = Graphics.CreateSprite(GameImage("sprite"))  AliveSprite.X = Rnd() \* (800 - AliveSprite.Width)  AliveSprite.Y = Rnd() \* (600 - AliveSprite.Height)  AliveSprite.Movement.X = Rnd() \* 2 - 1  AliveSprite.Movement.Y = Rnd() \* 2 - 1  End If  End Sub  End Class |

*Exercise 3: Creating Draw() and Update() methods*

1. Create Draw() and Update() methods inside the Bug class.

|  |
| --- |
| Answer:  Public Class Bug  …  Public Sub Draw()  If IsAlive Then  Graphics.DrawSprite(AliveSprite)  Else  Graphics.DrawSprite(DeadSprite)  End If  End Sub  Public Sub Update()  If IsAlive Then  Graphics.UpdateSprite(AliveSprite)  Else  Graphics.UpdateSprite(DeadSprite)  End If  End Sub  End Class |

*Exercise 4: Creating an object*

1. Create myBug object and make it to draw and to update itself (do not forget to add Randomize() method). Debug to see the result.

|  |
| --- |
| Answer:  …  LoadResources()  Input.ShowMouse(False)  Randomize()  Dim myBug As Bug  myBug = New Bug  …  'Clears the Screen to White (customized color)  SwinGame.Graphics.ClearScreen(Color.White)  myBug.Draw()  myBug.Update()  DrawMouse()  … |

**Answers to Part 2**

*Exercise 1: Creating CheckCollisions() method*

1. Create CheckCollisions() method inside the Bug Class. Debug to see the result.

|  |
| --- |
| Answer:  Public Class Bug  …  Public Sub Update()  If IsAlive Then  CheckCollisions()  …  End Sub  …  Private Sub CheckCollisions()  If AliveSprite.X + AliveSprite.Width >= Core.ScreenWidth Or AliveSprite.X <= 0 Then  AliveSprite.Movement.X = -AliveSprite.Movement.X  Audio.PlaySoundEffect(GameSound("hit"))  End If    If AliveSprite.Y + AliveSprite.Height >= Core.ScreenHeight Or AliveSprite.Y <= 0 Then  AliveSprite.Movement.Y = -AliveSprite.Movement.Y  Audio.PlaySoundEffect(GameSound("hit"))  End If  End Sub  End Class |

**Answers to Part 3**

*Exercise 1: Creating CheckIfClicked() method*

1. Create ChechkIfClicked() method inside the Bug class. Debug to see the result.

|  |
| --- |
| Answer:  Public Class Bug  …  Public Sub Update()  If IsAlive Then  CheckIfClicked()  …  End Sub  …  Private Sub CheckIfClicked()  Dim mousePoint As Point2D  mousePoint = Input.GetMousePosition()  If IsAlive And Physics.IsSpriteOnScreenAt(AliveSprite, mousePoint.X, mousePoint.Y) Then  If Input.MouseWasClicked(MouseButton.LeftButton) Then  Audio.PlaySoundEffect(GameSound("hit1"))  Alive = False  DeadSprite = Graphics.CreateSprite(GameImage("deadBug"), 20, 10, 57, 43)  DeadSprite.EndingAction = SpriteEndingAction.Stop  DeadSprite.X = AliveSprite.X  DeadSprite.Y = AliveSprite.Y  End If  End If  End Sub  End Class |