VBugs Chapter 7 Worksheet

*Solutions*

|  |
| --- |
| **Name:** |
| **Home Room:** |

Question 1: Define the terms: object; class; constructor.

Object:

An **object** knows certain things and can do certain things. We want to create an object that knows everything about our bugs that can also create them for us on the screen when we call it.

Class:

A class is a template that defines the properties of an object and the methods used to control that object's behavior.

Constructor:

A **constructor’s** job is to intialise the objects in the class.

Question 2: Define the term: property.

Property:

A property is used in programming language to allow you to read or write to a “Private” field from somewhere outside the class.

*Exercise 1: Creating fields and a property*

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

|  |
| --- |
| Answer:  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    Public Class Bug                                End Class |

*Exercse 2: Creating a constructor*

Create a costructor for your Bug class.

|  |
| --- |
| Answer:  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  Public Class Bug  …                          End Class |

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

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

|  |
| --- |
| 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  Answer:  Public Class Bug  …                          End Class |

*Exercise 4: Creating an object*

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

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

Question 3: Why are random numbers made by computers not “truly” random?

Because the computer will use the same equation to come up with the list of numbers many times. If the computer is given the same starting point it will give the same list of numbers. However if we give it a constantly changing number like the time, the number should appear to be more random but never “truly” random.

*Exercise 5: Creating CheckCollisions() method*

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

|  |
| --- |
| Answer:  CheckCollisions()  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  Public Class Bug  …                            Public Sub Update()  If IsAlive Then    …  End Sub  End Class |

*Exercise 6: Creating CheckIfClicked() method*

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

|  |
| --- |
| Answer:  CheckCollisions()  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("splat"))  Alive = False  DeadSprite = Graphics.CreateSprite(GameImage("deadBug"), 40,10, 57, 43)  DeadSprite.X = AliveSprite.X  DeadSprite.Y = AliveSprite.Y  DeadSprite.EndingAction = SpriteEndingAction.Stop  End If  End If  End Sub  Public Class Bug                                …  Public Sub Update()  If IsAlive Then    End Sub  End Class |