VBugs Chapter 8 Worksheet

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

Question 1: Define the term “list”.

A list in programming is a collection of similar things. A List is a class that has features that you can use to manage a number of objects in our case Bugs!

Exercise 1: *Creating a list*

1. Create a list of bugs in your program. Write the code you used to achieve this in the area below:

|  |
| --- |
| Answer:  listBugs = New List(Of Bug)  Private listBugs As List(Of Bug)  Module GameLogic      …  Public Sub Main()  …  LoadResources()  Input.ShowMouse(False)  Randomize()      … |

1. Write the code that will allow each bug in your program to draw and update itself. Write the code you used to achieve this in the area below:

|  |
| --- |
| Answer:  Public Sub CleanUp()  DeadSprite.Dispose()  AliveSprite.Dispose()  End Sub  For Each Bug As Bug In listBugs  Bug.Draw()  Bug.Update()  Next  …  SwinGame.Graphics.ClearScreen(Color.White)            … |

1. Create the function that will free the sprites. Write the code you used to achieve this in the area below(Delete the lines of code we don’t need any more)::

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

Question 2: Why do we have to free our sprites after we have finished with them?

We we need to free our sprites the end of the levels and the end of the game to frees up our memory so it can be used to store the bugs for the next level or something else.

*Exercise 2:* Variables and Level Set Up

1. Declare the five new variables we need for our game.

|  |
| --- |
| Private level As Integer  Private gameTimer As Timer  Private endLevelAt As Integer  Public bugsKilled As Integer  Public score As Integer  Public time As Integer  Private listBugs As List(Of Bug) |

1. Label and letter (a-e) the correct functions of the parts of LevelSetUp() below:
   1. Adds the bugs to the list
   2. Frees each sprite
   3. Prevents a level from being less than 500 milliseconds
   4. Stops then restarts the timer.
   5. Calculates how long the level goes for.

|  |
| --- |
| -e. Calculates how long the level will go  -d. Restarts the timer  -d. Stops the timer  -c. Prevents the level from being shorter than 500 milliseconds  -b. Frees each sprite  -a. Add the bugs to the list.  Public Sub LevelSetUp()  Core.StopTimer(gameTimer)    For Each Bug As Bug In listBugs  Bug.CleanUp()  Next  listBugs.Clear()  For i As Integer = 1 To level \* 2  listBugs.Add(New Bug)  Next  endLevelAt = 10000 - 500 \* (level - 1)  If endLevelAt < 500 Then  endLevelAt = 500  End If  Core.StartTimer(gameTimer)  End Sub |

Question 3: Using the logic above how long (in seconds) will level 6 be? Show your working.

10,000 – 500 \* 6 is 10,000 – 3,000 = 7,000 or **7 seconds**

Question 4: Using the logic above how may bugs will be on the screen in level 6? Show your working.

Level 1 = 2, Level2 = 4, Level 3 = 8, Level 4 = 16, Level 5 = 32, Level 6 = **64 bugs**

Question 5: Using the logic above how long (in seconds) will level 20 be? Show your working.

10,000 – 500 \* 20 is 10,000 – 10,000 = 0 however if < 500 then it will default to 500 so it will be 500 or **0.5 Seconds**

Exercise 3: Ending the Level and Initializing the Game Values

Create the code for ending the level and initializing the game values.

|  |
| --- |
| gameTimer = Core.CreateTimer()  level = 1  LevelSetUp()  score = 0  If EndOfLevel() Then  level = level + 1  LevelSetUp()  End If  Public Function EndOfLevel() As Boolean  For Each myBug As Bug In listBugs  If myBug.IsAlive Then  Return False  End If  Next  Return True  End Function  Module GameLogic                  End Module  …  Do  SwinGame.Graphics.ClearScreen(Color.White)          Loop Until SwinGame.Core.WindowCloseRequested() = True  …  listBugs = New List(Of Bug) |

Question 6: What are the two conditions that will cause the level to end?

A level is over if either the time has run out or all the bugs are dead.

Exercise 4: Setting up the score and score penalty

Implement scoring and score penalties in your game. Debug your game after you do.

|  |
| --- |
| bugsKilled = bugsKilled + 1  score = score + time  If time < 0 Then  scorePenalty = 100 \* level  score = score - scorePenalty  Core.StopTimer(gameTimer)  Core.StartTimer(gameTimer)  End If  DeadSprite.EndingAction = SpriteEndingAction.Stop      End If  …  Bug.Update()  Next            Loop Until SwinGame.Core.WindowCloseRequested() = True |

Question 7: A player has a score of 1050, they are on level 4 of the game and the time runs out before they kill all the bugs. What will there score be after the score penalty? Use the logic above to answer this question. Show your working.

scorePenalty = 100 \* 4 which is 400

1050 – 400 = **650**

A level is over if either the time has run out or all the bugs are dead.

Exercise 5: *Printing current score and time on the screen*

1. Load new fonts into your program.

|  |
| --- |
| NewFont("cat\_scratch", "cat\_scratch.ttf", 40)  NewFont("bear", "bear.ttf", 120)  NewFont("bear\_huge", "bear.ttf", 170)  NewFont("comic", "comic.ttf", 16)  NewFont("Courier", "cour.ttf", 16)          End Sub |

1. Draw the player status to the screen. (Test you game after you do)

|  |
| --- |
| time = (endLevelAt - Core.GetTimerTicks(gameTimer)) / 100  Text.DrawText("Level: " & level, Color.Green, GameFont("cat\_scratch"), 320, 2)  Text.DrawText("Time: " & time, Color.Red, GameFont("comic"), 2, 2)  Text.DrawText("Bugs killed: " & bugsKilled, Color.Green, GameFont("comic"), 2, 60)  Text.DrawText("Score: " & score, Color.Green, GameFont("comic"), 2, 30)  SwinGame.Graphics.ClearScreen(Color.White) |

*Question 8: Why do you think one “Level” has quatations around it in the DrawText statement above and another one in the same statement does not (level). What do you think the quotations mean? What do you think will be shown on the screen?*

“” indicate text that is drawn to screen the level without them refers to the variable level this means it will display the current level next to the words “Level: ”

*Question 9: In the same statement what do you think the & symbol means?*

& links the separate parts of the message being displayed, the variable and the text

*Question 10: If 4 seconds has passed and you are in level 3. What is being diplayed on the screen next to “Time”?*

4 seconds = 4,000 in the computer, Level 3 is 10,000 – 500 \*3 which is 10,000 – 1.500 which is 8,500. 8,500 – 4,000 is 4,500. 4,500 divide by 100 is 45. **45 will be displayed to the screen**

Exercise 7: *Printing a start and end point of the game*

1. *Build into your program the level introductions.*

|  |
| --- |
| Public Sub DrawLevelIntro()  For i As Integer = 1 To 50  Graphics.ClearScreen(Color.Black)  Text.DrawText("Level " & level, Color.Green,  GameFont("bear"), 280, 200)  Text.DrawText("Score: " & score, Color.Green,  GameFont("cat\_scratch"), 320, 300)  Core.RefreshScreen(25)  Core.ProcessEvents()  Next  End Sub  Module GameLogic                          End Module |

1. *Build into your program the loser screen.*

|  |
| --- |
| Public Sub DrawLoser()  Do  Graphics.ClearScreen(Color.White)  Text.DrawText("YOU ", Color.Green,  GameFont("bear"), 80, 150)  Text.DrawText("LOOOOSE!", Color.Red,  GameFont("bear\_huge"), 275, 125)  Text.DrawText("You Reached Level " & level,  Color.Green, GameFont("cat\_scratch"), 200, 340)  Text.DrawText("And Killed " & bugsKilled & " Bugs",  Color.Green, GameFont("cat\_scratch"), 240, 400)  Text.DrawText("Press SPACE to play again", Color.Green,  GameFont("cat\_scratch"), 120, 480)  Core.RefreshScreen(30)  Core.ProcessEvents()  Loop Until Input.WasKeyTyped(Keys.VK\_SPACE) Or  SwinGame.Core.WindowCloseRequested() = True  If Input.WasKeyTyped(Keys.VK\_SPACE) Then  score = 0  level = 1  LevelSetUp()  End If  End Sub  Module GameLogic                                                    End Module |

1. *Build into your program the penalty notice*

|  |
| --- |
| Public Sub Penalty()  For i As Integer = 1 To 25  Graphics.FillRectangle(Color.FromArgb(5, 255, 0, 0), 0,  0, 800, 600)  Text.DrawText("Too Slow!! ", Color.Black,  GameFont("bear"), 190, 200)  Text.DrawText("- " & scorePenalty, Color.Black,  GameFont("cat\_scratch"), 310, 300)  Core.RefreshScreen(25)  Core.ProcessEvents()  Next  End Sub  Module GameLogic                          End Module |

Question 11: How long does the Penalty Notice stay on the screen for(in seconds)? Show your working.

It loops 25 times and the refresh is set to 25 times per second therefor the penalty notice will stay up for **1 second.**