' Sets the player's image to "female2.gif"
' Sets the player's sex to female
Private Sub FemaleImage2_Click()
    Frame.Left = 537
    Frame.Visible = True
    CurrentPlayer.Picture = "female2.gif"
    CurrentPlayer.Sex = 1
End Sub

' Centers the frame around the selected image
' Sets the player's image to "female3.gif"
' Sets the player's sex to female
Private Sub FemaleImage3_Click()
    Frame.Left = 661
    Frame.Visible = True
    CurrentPlayer.Picture = "female3.gif"
    CurrentPlayer.Sex = 1
End Sub

' Sets up the size and location of all the objects
' on the form
' Initializes all the global variables
Private Sub Form_Load()
    CurrentPlayer.Picture = "Temp"
    GameForm.Cur_x = 61
    GameForm.Cur_y = 34
    NameLabel.Left = 400 - ((NameLabel.Width + NameTextBox.Width + 10) / 2)
    NameLabel.Top = 50
    NameTextBox.Left = NameLabel.Left + NameLabel.Width + 10
    NameTextBox.Top = NameLabel.Top
    StatLabel.Left = 400 - ((StatLabel.Width + 15 + StatPointsLabel.Width) / 2)
    CharacterLabel.Left = 400 - (CharacterLabel.Width / 2)
    FemaleImage1.Left = 413
    FemaleImage2.Left = 537
    FemaleImage3.Left = 661
    MaleImage1.Left = 40
    MaleImage2.Left = 165
    MaleImage3.Left = 289
    OkayCommand.Top = 525
    OkayCommand.Left = 400 - (OkayCommand.Width / 2)
    ' Paints the backgroud to the screen
    For i = 0 To 5
        For j = 0 To 7
            NewPlayerForm.PaintPicture NewPlayerForm.Picture, j * 100, i * 100, 100, 100
        Next j
    Next i
    StatPointsLabel.Caption = 40
    LastStrengthValue = StrengthPoints.Caption
    LastAgilityValue = AgilityPoints.Caption
    LastFitnessValue = FitnessPoints.Caption
    LastHealthValue = HealthPoints.Caption
    LastIntelligenceValue = IntelligencePoints.Caption
    LastPersonalityValue = PersonalityPoints.Caption
End Sub

' Centers the frame around the selected image
' Sets the player's image to "male1.gif"
' Sets the player's sex to male
Private Sub MaleImage1_Click()
    Frame.Left = 40
    Frame.Visible = True
NewPlayerForm - 4

CurrentPlayer.Picture = "male1.gif"
CurrentPlayer.Sex = 0
End Sub

' Centers the frame around the selected image
' Sets the player's image to "male2.gif"
' Sets the player's sex to male
Private Sub MaleImage2_Click()
    Frame.Left = 165
    Frame.Visible = True
    CurrentPlayer.Picture = "male2.gif"
    CurrentPlayer.Sex = 0
End Sub

' Centers the frame around the selected image
' Sets the player's image to "male3.gif"
' Sets the player's sex to male
Private Sub MaleImage3_Click()
    Frame.Left = 289
    Frame.Visible = True
    CurrentPlayer.Picture = "male3.gif"
    CurrentPlayer.Sex = 0
End Sub

' First checks to see that the player has finished
' setting up his character; if so, then the new
' player is created
Private Sub OkayCommand_Click()
    ' make sure the player entered a name
    If NameTextBox.text = "" Then
        MsgBox ("Please enter a name for your character.")
    ' make sure the player chose an image
    ElseIf CurrentPlayer.Picture = "Temp" Then
        MsgBox ("Please select an image for your character.")
    ' make sure player distributed all of his points
    ElseIf StatPointsLabel.Caption <> "0" Then
        MsgBox ("Please finish distributing your statistics points.")
    ' Set up new player
    Else
        CurrentPlayer.Name = NameTextBox.text
        CurrentPlayer.Age = 18
        CurrentPlayer.HitPoints = 100
        CurrentPlayer.Health = 100
        CurrentPlayer.Fatigue = 100
        CurrentPlayer.Strength = StrengthPoints.Caption
        CurrentPlayer.Agility = AgilityPoints.Caption
        CurrentPlayer.Fitness = FitnessPoints.Caption
        CurrentPlayer.Health = HealthPoints.Caption
        CurrentPlayer.Intelligence = IntelligencePoints.Caption
        CurrentPlayer.Personality = PersonalityPoints.Caption
        Calculate_HitBonus
        Calculate_DamageBonus
        Calculate_FatigueBonus
        Calculate_HitPointBonus
        CurrentPlayer.Armor = 1
        ' Add a monk's robe to the player's inventory
        Call GameForm.Aad(ItemArray(48))
        ' Add a pair of sandals to the player's inventory
        Call GameForm.Add(ItemArray(47))
        ' Add the special dagger to the player's inventory
NewPlayerForm - 5

Call GameForm.Add(ItemArray(19))
' Add a loaf of bread to the player's inventory
Call GameForm.Add(ItemArray(26))
' Add a hunk of cheese to the player's inventory
Call GameForm.Add(ItemArray(27))
' Add some wine to the player's inventory
Call GameForm.Add(ItemArray(29))
' Let the player wear the monk's robe
Set CurrentPlayer.Body = ItemArray(48)
' Let the player wear the sandals
Set CurrentPlayer.Feet = ItemArray(47)
' Let the player wield the special dagger
Set CurrentPlayer.Wielded = ItemArray(19)
' Put nothing on player's head, shield, about body (used to avoid errors)
Set CurrentPlayer.Head = ItemArray(55)
Set CurrentPlayer.Shield = ItemArray(55)
Set CurrentPlayer.Aboutbody = ItemArray(55)
' Give the player some money
CurrentPlayer.Money = 5
' Set up the flags
CurrentPlayer.Poisoned = False
CurrentPlayer.Bandaged = False
CurrentPlayer.Begged = 0
CurrentPlayer.Performed = 0
' Assign height and weight depending
' on the player's sex
If CurrentPlayer.Sex = 0 Then
    CurrentPlayer.Height = 18
    CurrentPlayer.Weight = 150
Else
    CurrentPlayer.Height = 16
    CurrentPlayer.Weight = 120
End If
' Unload the form
Unload Me
End If
End Sub

Private Sub StrengthPoints_Change()
If LastStrengthValue < StrengthPoints.Caption Then
    If StrengthPoints.Caption > 8 And StrengthPoints.Caption < 19 Then
        If StatPointsLabel.Caption < 0 Then
            StatPointsLabel.Caption = StatPointsLabel.Caption + 1
        StrengthPoints.Caption = LastStrengthValue
    End If
    ElseIf StrengthPoints.Caption > 18 And StrengthPoints.Caption < 23 Then
        If StatPointsLabel.Caption < 0 Then
        StrengthPoints.Caption = LastStrengthValue
    End If
    ElseIf StrengthPoints.Caption > 21 And StrengthPoints.Caption < 27 Then
        If StatPointsLabel.Caption < 0 Then
        StrengthPoints.Caption = LastStrengthValue
    End If
    ElseIf StrengthPoints.Caption > 25 And StrengthPoints.Caption < 31 Then
        If StatPointsLabel.Caption < 0 Then

    End If
End If
NewPlayerForm - 6

StrengthPoints.Caption = LastStrengthValue
End If
End If

Else If LastStrengthValue > StrengthPoints.Caption Then
    If StatPointsLabel.Caption = 40 Then
        Else
            If AgilityPoints.Caption > 7 And AgilityPoints.Caption < 18 Then
                StatPointsLabel.Caption = StatPointsLabel.Caption + 1
            ElseIf AgilityPoints.Caption > 17 And AgilityPoints.Caption < 22 Then
            ElseIf AgilityPoints.Caption > 21 And AgilityPoints.Caption < 26 Then
            ElseIf AgilityPoints.Caption > 25 And AgilityPoints.Caption < 30 Then
            End If
        End If
    End If
End If
LastStrengthValue = StrengthPoints.Caption
End Sub

Private Sub AgilityPoints_Change()
If LastAgilityValue < AgilityPoints.Caption Then
    If AgilityPoints.Caption > 8 And AgilityPoints.Caption < 19 Then
    ElseIf AgilityPoints.Caption > 18 And AgilityPoints.Caption < 23 Then
    ElseIf AgilityPoints.Caption > 21 And AgilityPoints.Caption < 27 Then
    ElseIf AgilityPoints.Caption > 25 And AgilityPoints.Caption < 31 Then
    End If
ElseIf AgilityPoints.Caption > 18 And AgilityPoints.Caption < 23 Then
ElseIf AgilityPoints.Caption > 21 And AgilityPoints.Caption < 27 Then
ElseIf AgilityPoints.Caption > 25 And AgilityPoints.Caption < 31 Then
ElseIf AgilityPoints.Caption > 7 And AgilityPoints.Caption < 18 Then
    StatPointsLabel.Caption = StatPointsLabel.Caption + 1
ElseIf AgilityPoints.Caption > 17 And AgilityPoints.Caption < 22 Then
ElseIf AgilityPoints.Caption > 21 And AgilityPoints.Caption < 26 Then
ElseIf AgilityPoints.Caption > 25 And AgilityPoints.Caption < 30 Then
End If
ElseIf LastAgilityValue > AgilityPoints.Caption Then
    If StatPointsLabel.Caption = 40 Then
        Else
            If AgilityPoints.Caption > 7 And AgilityPoints.Caption < 18 Then
                StatPointsLabel.Caption = StatPointsLabel.Caption + 1
            ElseIf AgilityPoints.Caption > 17 And AgilityPoints.Caption < 22 Then
            ElseIf AgilityPoints.Caption > 21 And AgilityPoints.Caption < 26 Then
            ElseIf AgilityPoints.Caption > 25 And AgilityPoints.Caption < 30 Then
            End If
        End If
    End If
End If
End If
LastAgilityValue = AgilityPoints.Caption
End Sub

Private Sub FitnessPoints_Change()
If LastFitnessValue < FitnessPoints.Caption Then
  If FitnessPoints.Caption > 8 And FitnessPoints.Caption < 19 Then
  ElseIf FitnessPoints.Caption > 18 And FitnessPoints.Caption < 23 Then
  ElseIf FitnessPoints.Caption > 21 And FitnessPoints.Caption < 27 Then
  ElseIf FitnessPoints.Caption > 25 And FitnessPoints.Caption < 31 Then
  End If
ElseIf LastFitnessValue > FitnessPoints.Caption Then
  If StatPointsLabel.Caption = 40 Then
    Else
      If FitnessPoints.Caption > 7 And FitnessPoints.Caption < 18 Then
        StatPointsLabel.Caption = StatPointsLabel.Caption + 1
      ElseIf FitnessPoints.Caption > 17 And FitnessPoints.Caption < 22 Then
      ElseIf FitnessPoints.Caption > 21 And FitnessPoints.Caption < 26 Then
      ElseIf FitnessPoints.Caption > 25 And FitnessPoints.Caption < 30 Then
      End If
    End If
  End If
End If
LastFitnessValue = FitnessPoints.Caption
End Sub

Private Sub HealthPoints_Change()
If LastHealthValue < HealthPoints.Caption Then
  If HealthPoints.Caption > 8 And HealthPoints.Caption < 19 Then
  ElseIf HealthPoints.Caption > 18 And HealthPoints.Caption < 23 Then
  End If
ElseIf LastHealthValue > HealthPoints.Caption Then
  If StatPointsLabel.Caption = 40 Then
    Else
      If HealthPoints.Caption > 7 And HealthPoints.Caption < 18 Then
        StatPointsLabel.Caption = StatPointsLabel.Caption + 1
      ElseIf HealthPoints.Caption > 17 And HealthPoints.Caption < 22 Then
      End If
    End If
  End If
End If
LastHealthValue = HealthPoints.Caption
End Sub
End If
ElseIf HealthPoints.Caption > 21 And HealthPoints.Caption < 27 Then
If StatPointsLabel.Caption < 0 Then
HealthPoints.Caption = LastHealthValue
End If
ElseIf HealthPoints.Caption > 25 And HealthPoints.Caption < 31 Then
If StatPointsLabel.Caption < 0 Then
HealthPoints.Caption = LastHealthValue
End If
ElseIf LastHealthValue > HealthPoints.Caption Then
    If StatPointsLabel.Caption = 40 Then
        Else
        And HealthPoints.Caption < 18 Then
            StatPointsLabel.Caption + 1
        ElseIf HealthPoints.Caption > 17 And HealthPoints.Caption < 22 Then
        ElseIf HealthPoints.Caption > 21 And HealthPoints.Caption < 26 Then
        ElseIf HealthPoints.Caption > 25 And HealthPoints.Caption < 31 Then
        End If
    End If
End If
ElseIf LastHealthValue > HealthPoints.Caption Then
    If StatPointsLabel.Caption = 40 Then
        Else
        End Sub

Private Sub Intelligencepoints_Change()
If LastIntelligenceValue < IntelligencePoints.Caption Then
    If IntelligencePoints.Caption > 8 And IntelligencePoints.Caption < 19 Then
    If StatPointsLabel.Caption < 0 Then
        StatPointsLabel.Caption = StatPointsLabel.Caption + 1
    IntelligencePoints.Caption = LastIntelligenceValue
End If
ElseIf IntelligencePoints.Caption > 18 And IntelligencePoints.Caption < 23 Then
If StatPointsLabel.Caption < 0 Then
    IntelligencePoints.Caption = LastIntelligenceValue
End If
ElseIf IntelligencePoints.Caption > 21 And IntelligencePoints.Caption < 27 Then
If StatPointsLabel.Caption < 0 Then
    IntelligencePoints.Caption = LastIntelligenceValue
End If
ElseIf IntelligencePoints.Caption > 25 And IntelligencePoints.Caption < 31 Then
If StatPointsLabel.Caption < 0 Then
    IntelligencePoints.Caption = LastIntelligenceValue
End If
ElseIf LastIntelligenceValue > IntelligencePoints.Caption Then
    If StatPointsLabel.Caption = 40 Then
        Else
If IntelligencePoints.Caption > 7 And IntelligencePoints.Caption < 18 Then
  StatPointsLabel.Caption = StatPointsLabel.Caption + 1
ElseIf IntelligencePoints.Caption > 17 And IntelligencePoints.Caption < 22 Then
ElseIf IntelligencePoints.Caption > 21 And IntelligencePoints.Caption < 26 Then
ElseIf IntelligencePoints.Caption > 25 And IntelligencePoints.Caption < 30 Then
End If
End If

Last Intelligence Value = IntelligencePoints.Caption
End Sub

Private Sub PersonalityPoints_Change()
If LastPersonalityValue < PersonalityPoints.Caption Then
  If PersonalityPoints.Caption > 8 And PersonalityPoints.Caption < 19 Then
    If StatPointsLabel.Caption < 0 Then
      StatPointsLabel.Caption = StatPointsLabel.Caption + 1
      PersonalityPoints.Caption = LastPersonalityValue
    End If
  ElseIf PersonalityPoints.Caption > 18 And PersonalityPoints.Caption < 23 Then
    If StatPointsLabel.Caption < 0 Then
      PersonalityPoints.Caption = LastPersonalityValue
    End If
  ElseIf PersonalityPoints.Caption > 21 And PersonalityPoints.Caption < 27 Then
    If StatPointsLabel.Caption < 0 Then
      PersonalityPoints.Caption = LastPersonalityValue
    End If
  ElseIf PersonalityPoints.Caption > 25 And PersonalityPoints.Caption < 31 Then
    If StatPointsLabel.Caption < 0 Then
      PersonalityPoints.Caption = LastPersonalityValue
    End If
  End If
ElseIf LastPersonalityValue > PersonalityPoints.Caption Then
  If StatPointsLabel.Caption = 40 Then
  Else
    If PersonalityPoints.Caption > 7 And PersonalityPoints.Caption < 18 Then
      StatPointsLabel.Caption = StatPointsLabel.Caption + 1
    ElseIf PersonalityPoints.Caption > 17 And PersonalityPoints.Caption < 22 Then
    ElseIf PersonalityPoints.Caption > 21 And PersonalityPoints.Caption < 26 Then
    ElseIf PersonalityPoints.Caption > 25 And PersonalityPoints.Caption < 30 Then
    End If
  End If
End If

PersonalityValue = PersonalityPoints.Caption
End Sub
SkillsForm - 1

' All functions on this form were written by
' Karen Woznick, unless otherwise noted.

' LastAnimalLoreValue As Byte
Dim LastArcheryValue As Byte
Dim LastBrawlingValue As Byte
Dim LastCampingValue As Byte
Dim LastChoppingValue As Byte
Dim LastCommerceValue As Byte
Dim LastFireBuildingValue As Byte
Dim LastFishingValue As Byte
Dim LastForagingValue As Byte
Dim LastStealthValue As Byte
Dim LastHikingValue As Byte
Dim LastHuntingValue As Byte
Dim LastItemLoreValue As Byte
Dim LastSocialValue As Byte
Dim LastWrestlingValue As Byte

' Sets up the size and location of the
' objects on the form
Private Sub Form_Load()
    ExplainLabel.Top = 50
    ExplainLabel.Left = 400 - ((ExplainLabel.Width + 10 + SkillsPointsLabel) / 2)
    SkillsPointsLabel.Top = 50
    SkillsPointsLabel.Left = ExplainLabel.Left + ExplainLabel.Width + 10
    LastAnimalLoreValue = AnimalLorePoints.Caption
    LastArcheryValue = ArcheryPoints.Caption
    LastBrawlingValue = BrawlingPoints.Caption
    LastCampingValue = CampingPoints.Caption
    LastChoppingValue = ChoppingPoints.Caption
    LastCommerceValue = CommercePoints.Caption
    LastFireBuildingValue = FireBuildingPoints.Caption
    LastFishingValue = FishingPoints.Caption
    LastForagingValue = ForagingPoints.Caption
    LastStealthValue = StealthPoints.Caption
    LastHikingValue = HikingPoints.Caption
    LastHuntingValue = HuntingPoints.Caption
    LastItemLoreValue = ItemLorePoints.Caption
    LastSocialValue = SocialPoints.Caption
    LastWrestlingValue = WrestlingPoints.Caption

    ' Paints the form's picture to the entire screen
    For i = 0 To 5
        For j = 0 To 7
            SkillsForm.PaintPicture SkillsForm.Picture, j * 100, i * 100, 100, 100
        Next j, i
        ExplainLabel.Left = 400 - ((ExplainLabel.Width + 10 + SkillsPointsLabel.Width) / 2)
        SkillsPointsLabel.Left = ExplainLabel.Left + ExplainLabel.Width + 10
        OkayButton.Left = 400 - (OkayButton.Width / 2)
        OkayButton.Top = 500
    Next
End Sub

' First checks to see if the player has finished
' distributing all of the points, then sets his
' skills rating to the values on the form
Private Sub OkayButton_Click()
    If SkillsPointsLabel.Caption <> "0" Then
        MsgBox ("Please finish distributing your skill points.")
    Else

SkillsForm - 2

CurrentPlayer.Archery = ArcheryPoints.Caption
CurrentPlayer.Brawling = BrawlingPoints.Caption
CurrentPlayer.Camping = CampingPoints.Caption
CurrentPlayer.Chopping = ChoppingPoints.Caption
CurrentPlayer.Commerce = CommercePoints.Caption
CurrentPlayer.Firebuilding = FireBuildingPoints.Caption
CurrentPlayer.Fishing = FishingPoints.Caption
CurrentPlayer.Foraging = ForagingPoints.Caption
CurrentPlayer.Stealth = StealthPoints.Caption
CurrentPlayer.Hiking = HikingPoints.Caption
CurrentPlayer.Hunting = HuntingPoints.Caption
CurrentPlayer.Itemlore = ItemLorePoints.Caption
CurrentPlayer.Social = SocialPoints.Caption
CurrentPlayer.Wrestling = WrestlingPoints.Caption
CurrentPlayer.Armor = 1
Unload Me
End If
End Sub

Private Sub AnimalLorePoints_Change()
If LastAnimalLoreValue < AnimalLorePoints.Caption Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
    If SkillsPointsLabel.Caption < 0 Then
        SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
        AnimalLorePoints.Caption = LastAnimalLoreValue
    End If
ElseIf LastAnimalLoreValue > AnimalLorePoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastAnimalLoreValue = AnimalLorePoints.Caption
End Sub

Private Sub BrawlingPoints_Change()
If LastBrawlingValue < BrawlingPoints.Caption Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
    If SkillsPointsLabel.Caption < 0 Then
        SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
        BrawlingPoints.Caption = LastBrawlingValue
    End If
ElseIf LastBrawlingValue > BrawlingPoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastBrawlingValue = BrawlingPoints.Caption
End Sub

Private Sub CampingPoints_Change()
If LastCampingValue < CampingPoints.Caption Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
    If SkillsPointsLabel.Caption < 0 Then
        SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
        CampingPoints.Caption = LastCampingValue
    End If
ElseIf LastCampingValue > CampingPoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastCampingValue = CampingPoints.Caption
End Sub
Else
  SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
End If
ElseIf LastChoppingValue > ChoppingPoints.Caption Then
  SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
  End If
ElseIf LastChoppingValue > ChoppingPoints.Caption Then
  SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
End If
LastChoppingValue = ChoppingPoints.Caption
End Sub

Private Sub CommercePoints_Change()
If LastCommerceValue < CommercePoints.Caption Then
  SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
  If SkillsPointsLabel.Caption < 0 Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    CommercePoints.Caption = LastCommerceValue
  End If
ElseIf LastCommerceValue > CommercePoints.Caption Then
  SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
End If
LastCommerceValue = CommercePoints.Caption
End Sub

Private Sub FireBuildingPoints_Change()
If LastFireBuildingValue < FireBuildingPoints.Caption Then
  SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
  If SkillsPointsLabel.Caption < 0 Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    FireBuildingPoints.Caption = LastFireBuildingValue
  End If
ElseIf LastFireBuildingValue > FireBuildingPoints.Caption Then
  SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
End If
LastFireBuildingValue = FireBuildingPoints.Caption
End Sub

Private Sub FishingPoints_Change()
If LastFishingValue < FishingPoints.Caption Then
  SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
  If SkillsPointsLabel.Caption < 0 Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    FishingPoints.Caption = LastFishingValue
  End If
ElseIf LastFishingValue > FishingPoints.Caption Then
  SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
End If
LastFishingValue = FishingPoints.Caption
End Sub
End If
ElseIf LastFishingValue > FishingPoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastFishingValue = FishingPoints.Caption
End Sub

Private Sub ForagingPoints_Change()
If LastForagingValue < ForagingPoints.Caption Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
    If SkillsPointsLabel.Caption < 0 Then
        SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
        ForagingPoints.Caption = LastForagingValue
    End If
ElseIf LastForagingValue > ForagingPoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastForagingValue = ForagingPoints.Caption
End Sub

Private Sub StealthPoints_Change()
If LastStealthValue < StealthPoints.Caption Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
    If SkillsPointsLabel.Caption < 0 Then
        SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
        StealthPoints.Caption = LastStealthValue
    End If
ElseIf LastStealthValue > StealthPoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastStealthValue = StealthPoints.Caption
End Sub

Private Sub HikingPoints_Change()
If LastHikingValue < HikingPoints.Caption Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
    If SkillsPointsLabel.Caption < 0 Then
        SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
        HikingPoints.Caption = LastHikingValue
    End If
ElseIf LastHikingValue > HikingPoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastHikingValue = HikingPoints.Caption
End Sub

Private Sub HuntingPoints_Change()
If LastHuntingValue < HuntingPoints.Caption Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
SkillsForm - 5

If SkillsPointsLabel.Caption < 0 Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    HuntingPoints.Caption = LastHuntingValue
End If
ElseIf LastHuntingValue > HuntingPoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastHuntingValue = HuntingPoints.Caption
End Sub

Private Sub ItemLorePoints_Change()
    If LastItemLoreValue < ItemLorePoints.Caption Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
    If SkillsPointsLabel.Caption < 0 Then
        SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
        ItemLorePoints.Caption = LastItemLoreValue
    End If
ElseIf LastItemLoreValue > ItemLorePoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastItemLoreValue = ItemLorePoints.Caption
End Sub

Private Sub SocialPoints_Change()
    If LastSocialValue < SocialPoints.Caption Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
    If SkillsPointsLabel.Caption < 0 Then
        SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
        SocialPoints.Caption = LastSocialValue
    End If
ElseIf LastSocialValue > SocialPoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastSocialValue = SocialPoints.Caption
End Sub

Private Sub WrestlingPoints_Change()
    If LastWrestlingValue < WrestlingPoints.Caption Then
    SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
    If SkillsPointsLabel.Caption < 0 Then
        SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
        WrestlingPoints.Caption = LastWrestlingValue
    End If
ElseIf LastWrestlingValue > WrestlingPoints.Caption Then
    If SkillsPointsLabel.Caption = 250 Then
        Else
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
    End If
End If
LastWrestlingValue = WrestlingPoints.Caption
End Sub
Private Sub ArcheryPoints_Change()
    If LastArcheryValue < ArcheryPoints.Caption Then
        SkillsPointsLabel.Caption = SkillsPointsLabel.Caption - 1
        If SkillsPointsLabel.Caption < 0 Then
            SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
            ArcheryPoints.Caption = LastArcheryValue
        End If
    ElseIf LastArcheryValue > ArcheryPoints.Caption Then
        If SkillsPointsLabel.Caption = 250 Then
            Else
                SkillsPointsLabel.Caption = SkillsPointsLabel.Caption + 1
            End If
        End If
    End If
    LastArcheryValue = ArcheryPoints.Caption
End Sub
' All functions on this form were written by
' Karen Woznick, unless otherwise noted.

' Adjusts the size and location of the objects on
' the form
Private Sub Form_Load()
    OkayCommand.Top = 600 - 15 - OkayCommand.Height
    OkayCommand.Left = 400 - (OkayCommand.Width / 2)
    Page2.Top = OkayCommand.Top
    Page2.Left = OkayCommand.Left
    Page3.Top = OkayCommand.Top
    Page3.Left = OkayCommand.Left
    StoryTextBox.Top = 15
    StoryTextBox.Left = 15
    StoryTextBox.Width = 800 - 30
    StoryTextBox.Height = 600 - 30 - 10 - OkayCommand.Height
    StoryTextBox.LoadFile (CurrentImageDrive + "Vale1.rtf")
    ' Paints the form's picture to the screen
    For i = 0 To 2
        For j = 0 To 4
            StoryForm.PaintPicture StoryForm.Picture, j * 200, i * 200, 200, 200
        Next j, i
    Next
End Sub

' Unloads the current form
Private Sub OkayCommand_Click()
    Unload Me
End Sub

' Loads the next page of the story
Private Sub Page2_Click()
    Page2.Visible = False
    StoryTextBox.text = ""
    StoryTextBox.LoadFile (CurrentImageDrive + "Vale2.rtf")
    Page3.Visible = True
End Sub

' Loads the next page of the story
Private Sub Page3_Click()
    Page3.Visible = False
    StoryTextBox.text = ""
    StoryTextBox.LoadFile (CurrentImageDrive + "Vale3.rtf")
    OkayCommand.Visible = True
End Sub
'TownForm - 1

' All functions on this form were written by
' Karen Woznick, unless otherwise noted

Option Explicit
' Keeps track of what type of commands the
' player want to use
Dim TownState As String
' Array of items available to buy in this town
Dim BuyItemArray(30) As Item
' Keeps track of the next available index in the BuyItemArray
Dim Index As Integer

' Adds an item to the player's inventory and the
' items/equipments combo boxes on both the TownForm
' and GameForm
Public Sub Add(NewItem As Item)
' First checks to see if the inventory is full.
If CurrentPlayer.NextIndex >= 99 Then
    MsgBox ("You may not add any more items to your inventory.")
Else
    Set CurrentPlayer.Inventory(CurrentPlayer.NextIndex) = NewItem
    Items.AddItem (CurrentPlayer.Inventory(CurrentPlayer.NextIndex).Name)
    GameForm.Equipment.AddItem (CurrentPlayer.Inventory(CurrentPlayer.NextIndex).Name)
    CurrentPlayer.NextIndex = CurrentPlayer.NextIndex + 1
End If
End Sub

' Adds an item to the combo box which shows what items
' are available to buy in the current town
Private Sub AddBuyItem(NewItem As Item)
    Set BuyItemArray(Index) = NewItem
    BuyItems.AddItem (NewItem.Name)
    Index = Index + 1
End Sub

' Appraising
' Allows the player to use his appraising skill to determine
' an approximate value of an item in his inventory.
Private Sub Appraising()
' First checks to see if the player is holding something
If Items.text = "" Then
    MsgBox ("You must be holding something in order to appraise it.")
Else
    ' Determine a value for the item
    Dim Value As Integer
    Value = CurrentPlayer.Inventory(Items.ListIndex).Value * (CurrentPlayer.Appraising / 100)
    MsgBox ("You have appraised the item for " + CStr(Value) + " silver pieces.")
End If
' Give the player a chance to improve his skill
If CurrentPlayer.Appraising < GameForm.RandomNumber() Then
    CurrentPlayer.Appraising = CurrentPlayer.Appraising + GameForm.Improve_Skill
End If
End Sub

' Begging
' Allows the player to beg from the characters for bread
' or money, depending on his skill rating
Public Sub Begging(ByVal Name As String)
' Check to see if the player has already begged 3 times today
If CurrentPlayer.Begged < 3 Then
    ' Determine a modifier based on the player's social skill
    Dim modifier As Byte
    If CurrentPlayer.Social > 0 And CurrentPlayer.Social < 15 Then
        modifier = 1
    ElseIf CurrentPlayer.Social > 14 And CurrentPlayer.Social < 30 Then
        modifier = 2
    ElseIf CurrentPlayer.Social > 29 And CurrentPlayer.Social < 60 Then
        modifier = 4
    ElseIf CurrentPlayer.Social > 59 Then
        modifier = 6
    End If
    ' Make sure player has enough fatigue
    If CurrentPlayer.Fatigue < 2 Then
        MsgBox ("You are too tired.")
    ' If player passes skill check
    ElseIf (CurrentPlayer.Begging + modifier) > GameForm.RandomNumber() Then
        ' Determine what player receives from begging
        If CurrentPlayer.Begging < 30 And CurrentPlayer.Begging > 0 Then
            Call Show_MessageBox(CurrentImageDrive + ItemArray(26).Picture, "The " & Name & " has given you some bread.")
            ' Add bread to player's inventory
            Add (ItemArray(26))
            ' Increment bread
            CurrentPlayer.Begged = CurrentPlayer.Begged + 1
            ' Pass some game time
            GameForm.PassTime (5)
            ' Subtract fatigue points
        ElseIf CurrentPlayer.Begging < 60 And CurrentPlayer.Begging > 29 Then
            Dim temp As Byte
            temp = (GameForm.RandomNumber() Mod 5) + 1
            MsgBox ("The " & Name & " has given you " & temp & " silver pieces.")
            ' Add money
            ' Increment begged
            CurrentPlayer.Begged = CurrentPlayer.Begged + 1
            ' Pass some game time
            GameForm.PassTime (5)
            ' Subtract fatigue
        Else
            temp = (GameForm.RandomNumber() Mod 10) + 1
            MsgBox ("The " & Name & " has given you " & temp & " silver pieces.")
            ' Add money
            ' Increment begged
            CurrentPlayer.Begged = CurrentPlayer.Begged + 1
            ' Pass some game time
            GameForm.PassTime (5)
            ' Subtract fatigue
        End If
    Else
        ' Display appropriate message
        MsgBox ("You were unsuccessful at begging from the " & Name & ".")
        ' Give player a chance to improve his skill
        If CurrentPlayer.Social < GameForm.RandomNumber() Then
TownForm - 3

End If
End If
Else
MsgBox ("You may only beg 3 times each day.")
End If
End Sub

' Shows the Buy Frame and fills the combo box
' with items that are available in this town.
Private Sub BuyCommand_Click()
BuyFrame.Left = 400 - (BuyFrame.Width / 2)
BuyFrame.Top = 300 - (BuyFrame.Height / 2)
BuyFrame.Visible = True
Index = 0
BuyItems.ListIndex = -1
If CurrentTown = 0 Then
  ' add leather armor
  Call AddBuyItem(ItemArray(5))
  ' add buckler
  Call AddBuyItem(ItemArray(0))
  ' add dagger
  Call AddBuyItem(ItemArray(11))
  ' add knife
  Call AddBuyItem(ItemArray(12))
  ' add pants and shirt
  Call AddBuyItem(ItemArray(43))
  ' add dress
  Call AddBuyItem(ItemArray(45))
ElseIf CurrentTown = 1 Then
  ' add chain mail
  Call AddBuyItem(ItemArray(6))
  ' add helmet
  Call AddBuyItem(ItemArray(4))
  ' add long sword
  Call AddBuyItem(ItemArray(9))
  ' add bastard sword
  Call AddBuyItem(ItemArray(10))
  ' add mace
  Call AddBuyItem(ItemArray(13))
ElseIf CurrentTown = 2 Then
  ' add warhammer
  Call AddBuyItem(ItemArray(14))
  ' add axe
  Call AddBuyItem(ItemArray(15))
  ' add bow
  Call AddBuyItem(ItemArray(17))
  ' add plate boots
  Call AddBuyItem(ItemArray(3))
  ' add small shield
  Call AddBuyItem(ItemArray(1))
ElseIf CurrentTown = 3 Then
  ' add plate mail
  Call AddBuyItem(ItemArray(7))
  ' add bastard sword
  Call AddBuyItem(ItemArray(10))
  ' add large shield
  Call AddBuyItem(ItemArray(2))
  ' add plate boots
  Call AddBuyItem(ItemArray(3))
  ' add helmet
Private Sub BuyItemCommand_Click()
    ' Check to make sure player has selected an item
    If BuyItems.text = "" Then
        MsgBox("Please select an item.")
    Else
        Dim cost As Integer
        Dim discount As Single
        discount = 0
        GameForm.PassTime (2)
        ' Determine the discount based on the player's commerce and barter skills
        If CurrentPlayer.Commerce < 51 Then
            discount = 0.02
        ElseIf CurrentPlayer.Commerce > 50 And CurrentPlayer.Commerce < 75 Then
            discount = 0.04
        ElseIf CurrentPlayer.Commerce > 74 Then
            discount = 0.06
        End If
        If CurrentPlayer.Barter < 21 Then
            discount = discount + 0.02
        ElseIf CurrentPlayer.Barter > 20 And CurrentPlayer.Barter < 51 Then
            discount = discount + 0.05
        ElseIf CurrentPlayer.Barter > 50 And CurrentPlayer.Barter < 75 Then
            discount = discount + 0.1
        ElseIf CurrentPlayer.Barter > 74 Then
            discount = discount + 0.15
        End If
        ' Determine the cost of the item
        cost = GameForm.Price(ItemArray(Index)) * discount
        ' Store the cost in the player's item
        Player.AddItem(cost, ItemArray(Index))
        ' Display the cost in the UI
        MsgBox("Cost: ", cost)
    End If
End Sub
cost = BuyItemArray(BuyItems.ListIndex).Value - (BuyItemArray(BuyItems.ListIndex).Value * discount)
' Check to see if player has enough money
If CurrentPlayer.Money >= cost Then
' Decrease money
' Update the label which displays the player's current amount of money
Update_Silver
' Add item to inventory
Call Add(BuyItemArray(BuyItems.ListIndex))
Call Show_MessageBox(CurrentImageDrive + BuyItemArray(BuyItems.ListIndex).Picture, "You have purchased the " + BuyItemArray(BuyItems.ListIndex).Name & " for " & cost & " silver pieces.")
' If player does not have enough money
Else
    MsgBox ("You do not have enough money to buy this item.")
End If
End If
End Sub

' Removes the Buy Frame from the screen
Private Sub CancelItemCommand_Click()
    BuyFrame.Visible = False
    TownState = "Normal"
End Sub

' Removes the Sell Frame from the screen
Private Sub CancelSellCommand_Click()
    SellFrame.Visible = False
    TownState = "Normal"
End Sub

' Determines what player wants to do, then
determines which character was selected
and calls the appropriate subroutine
Private Sub Character_Click(Index As Integer)
    If TownState = "Talk" Then
        Select Case CurrentTown
        Case 0
            Select Case Index
            Case 0 ' Village Councilmember
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(0).Picture, CharacterArray(0).Speech)
                Pause (2)
                If CurrentPlayer.Social < 60 Then
                    MsgBox ("You have acquired additional knowledge of social skills from the " + CharacterArray(0).Name + ").")
                    GameForm.PassTime (60)
                Else
                    MsgBox ("You are too knowledgeable in social skills to learn any more here.")
                End If
            Case 1 ' Peasant Boy
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(1).Picture, CharacterArray(1).Speech)
                Pause (2)
                If CurrentPlayer.Chopping < 60 Then
                    MsgBox ("You have acquired additional knowledge of chopping from the " + CharacterArray(1).Name + ").")
                End If
            Case 2 ' Hunter
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(2).Picture, CharacterArray(2).Speech)
                Pause (2)
                If CurrentPlayer.Hunting < 60 Then
                    MsgBox ("You have acquired additional knowledge of hunting from the " + CharacterArray(2).Name + ").")
                End If
            Case 3 ' Shepherd
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(3).Picture, CharacterArray(3).Speech)
                Pause (2)
                If CurrentPlayer.Shepherding < 60 Then
                    MsgBox ("You have acquired additional knowledge of shepherding from the " + CharacterArray(3).Name + ").")
                End If
            Case 4 ' Blacksmith
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(4).Picture, CharacterArray(4).Speech)
                Pause (2)
                If CurrentPlayer.Blacksmithing < 60 Then
                    MsgBox ("You have acquired additional knowledge of blacksmithing from the " + CharacterArray(4).Name + ").")
                End If
            Case 5 ' Carpenter
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(5).Picture, CharacterArray(5).Speech)
                Pause (2)
                If CurrentPlayer.Carpentry < 60 Then
                    MsgBox ("You have acquired additional knowledge of carpentry from the " + CharacterArray(5).Name + ").")
                End If
            Case 6 ' Tailor
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(6).Picture, CharacterArray(6).Speech)
                Pause (2)
                If CurrentPlayer.Tailoring < 60 Then
                    MsgBox ("You have acquired additional knowledge of tailoring from the " + CharacterArray(6).Name + ").")
                End If
            Case 7 ' Herbalist
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(7).Picture, CharacterArray(7).Speech)
                Pause (2)
                If CurrentPlayer.Herbalism < 60 Then
                    MsgBox ("You have acquired additional knowledge of herbalism from the " + CharacterArray(7).Name + ").")
                End If
            Case 8 ' Alchemist
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(8).Picture, CharacterArray(8).Speech)
                Pause (2)
                If CurrentPlayer.Alchemy < 60 Then
                    MsgBox ("You have acquired additional knowledge of alchemy from the " + CharacterArray(8).Name + ").")
                End If
            Case 9 ' Magician
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(9).Picture, CharacterArray(9).Speech)
                Pause (2)
                If CurrentPlayer.Magic < 60 Then
                    MsgBox ("You have acquired additional knowledge of magic from the " + CharacterArray(9).Name + ").")
                End If
            Case 10 ' Warrior
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(10).Picture, CharacterArray(10).Speech)
                Pause (2)
                If CurrentPlayer.Warrior < 60 Then
                    MsgBox ("You have acquired additional knowledge of warrior skills from the " + CharacterArray(10).Name + ").")
                End If
            Case 11 ' Archer
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(11).Picture, CharacterArray(11).Speech)
                Pause (2)
                If CurrentPlayer.Archer < 60 Then
                    MsgBox ("You have acquired additional knowledge of archery from the " + CharacterArray(11).Name + ").")
                End If
            Case 12 ' Fisher
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(12).Picture, CharacterArray(12).Speech)
                Pause (2)
                If CurrentPlayer.Fishing < 60 Then
                    MsgBox ("You have acquired additional knowledge of fishing from the " + CharacterArray(12).Name + ").")
                End If
            Case 13 ' Barber
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(13).Picture, CharacterArray(13).Speech)
                Pause (2)
                If CurrentPlayer.Barbering < 60 Then
                    MsgBox ("You have acquired additional knowledge of barbering from the " + CharacterArray(13).Name + ").")
                End If
            Case 14 ' Cafe Owner
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(14).Picture, CharacterArray(14).Speech)
                Pause (2)
                If CurrentPlayer.Cafe_Owner < 60 Then
                    MsgBox ("You have acquired additional knowledge of cafe management from the " + CharacterArray(14).Name + ").")
                End If
            Case 15 ' Gardener
                Call Show_SpeechBox(CurrentImageDrive + CharacterArray(15).Picture, CharacterArray(15).Speech)
```vbnet
GameForm.PassTime (60)
Pause (2)
If CurrentPlayer.Chopping = 0 Then
  GameForm.Skills.AddItem ("Chopping")
End If
Else
  MsgBox ("You are too knowledgeable in chopping to learn any more here.")
End If
CurrentPlayer.Chopping = CurrentPlayer.Chopping + GameForm.Learn_Skill
If CurrentPlayer.Chopping > 10 Then
  MsgBox ("You have acquired additional knowledge of carving from the " + CharacterArray(1).Name + ".")
  GameForm.PassTime (60)
  If CurrentPlayer.Carving < 60 Then
    GameForm.Skills.AddItem ("Carving")
  End If
Else
  MsgBox ("You are too knowledgeable in carving until you are more knowledgeable in chopping.")
End If
End Select
Case 1
  Call Show_SpeechBox(CurrentImageDrive + CharacterArray(3).Picture, CharacterArray(3).Speech)
  Pause (2)
  If CurrentPlayer.Hiking < 60 Then
    GameForm.PassTime (60)
    MsgBox ("You have acquired additional knowledge of hiking from " + CharacterArray(3).Name + ".")
  Else
    CurrentPlayer.Hiking = CurrentPlayer.Hiking + GameForm.Learn_Skill
  End If
Case 1
  Call Show_SpeechBox(CurrentImageDrive + CharacterArray(4).Picture, CharacterArray(4).Speech)
  Pause (2)
  If CurrentPlayer.Commerce < 60 Then
    GameForm.PassTime (60)
    MsgBox ("You have acquired additional knowledge of commerce from the " + CharacterArray(4).Name + ".")
    CurrentPlayer.Commerce = CurrentPlayer.Commerce + GameForm.Learn_Skill
  Else
    MsgBox ("You are too knowledgeable in commerce to learn any more here.")
  End If
Case 1
  MsgBox ("You are too knowledgeable in item lore to learn any more here.")
End Select
```
Case 2 'Young boy
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(6).Picture, CharacterArray(6).Speech)
Pause (2)
If CurrentPlayer.Stealth > 10 Then
  If CurrentPlayer.PickPocketing < 60 Then
    GameForm.PassTime (60)
    MsgBox ("You have acquired additional knowledge of pick pocketing from the " + CharacterArray(6).Name + ").")
  End If
  If CurrentPlayer.PickPocketing = 0 Then
    GameForm.Skills.AddItem ("PickPocketing")
    Skills.AddItem ("PickPocketing")
  End If
Else
  MsgBox ("You are too knowledgeable in pick pocketing to learn any more here.")
End If
Else
  MsgBox ("You may not learn pick pocketing until you are more accomplished in stealth.")
End If
Case 3 'Drunken man
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(7).Picture, CharacterArray(7).Speech)
Pause (2)
If CurrentPlayer.Commerce > 10 Then
  If CurrentPlayer.Barter < 60 Then
    GameForm.PassTime (60)
    MsgBox ("You have acquired additional knowledge of barter from the " + CharacterArray(7).Name + ").")
  End If
Else
  MsgBox ("You are too knowledgeable in barter to learn any more here.")
End If
Else
  MsgBox ("You may not learn barter until you are more knowledgeable in commerce.")
End If
If CurrentPlayer.Brawling < 60 Then
  GameForm.PassTime (60)
  MsgBox ("You have acquired additional knowledge of brawling from the " + CharacterArray(7).Name + ").")
Else
  MsgBox ("You are too knowledgeable in brawling to learn any more here.")
End If
End Select ' index
Case 2
Select Case Index
  Case 0 ' Tavern Girl
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(11).Picture, CharacterArray(11).Speech)
Pause (2)
If CurrentPlayer.Social < 60 Then
  GameForm.PassTime (60)
  MsgBox ("You have acquired additional knowledge of social skills from th
e " + CharacterArray(11).Name + ".")
   Else
     MsgBox ("You are too knowledgeable in social skills to learn any more here.")
   End If
Case 1 ' Bartender
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(12).Picture, CharacterArray(12).Speech)
Pause (2)
Dim choice As VbMsgBoxResult
choice = MsgBox("Would you like to buy the crosspiece from the bartender?", vbYesNo)
If choice = vbYes Then
   Dim price As Byte
   price = 100 - (CurrentPlayer.Social \ 10) - (CurrentPlayer.Barter / 10) - (CurrentPlayer.Commerce \ 10)
   If CurrentPlayer.Money >= price Then
     Call Show_MessageBox(CurrentImageDrive + ItemArray(22).Picture, "You have purchased the crosspiece for ", price, " silver pieces.")
     Update_Silver
     Call GameForm.CheckItems
   Else
     MsgBox ("You don't have enough money to buy the crosspiece from the greedy bartender.")
   End If
End If
End If
Pause (1)
If CurrentPlayer.Performance < 60 Then
  GameForm.PassTime (60)
  MsgBox ("You have acquired additional knowledge of performance from the ", CharacterArray(12).Name + ".")
  CurrentPlayer.Performance = CurrentPlayer.Performance + GameForm.Learn_Skill
Else
  MsgBox ("You are too knowledgeable in performance to learn any more here.")
End If
Case 2 ' Adventurer2
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(13).Picture, CharacterArray(13).Speech)
Pause (2)
If CurrentPlayer.AnimalLore < 60 Then
  GameForm.PassTime (60)
  MsgBox ("You have acquired additional knowledge of animal lore from the ", CharacterArray(13).Name + ".")
Else
  MsgBox ("You are too knowledgeable in animal lore to learn any more here.")
End If
Pause (1)
If CurrentPlayer.Foraging < 60 Then
  GameForm.PassTime (60)
  MsgBox ("You have acquired additional knowledge of foraging from the ", CharacterArray(13).Name + ".")
  CurrentPlayer.Foraging = CurrentPlayer.Foraging + GameForm.Learn_Skill
Else
  Pause (1)
MsgBox ("You are too knowledgeable in foraging to learn any more here.")
End If
If CurrentPlayer.Hunting < 60 Then
GameForm.PasTime (60)
MsgBox ("You have acquired additional knowledge of hunting from the " + CharacterArray(13).Name + ")."
Pause (1)
Else
MsgBox ("You are too knowledgeable in hunting to learn any more here.")
End If
If CurrentPlayer.Butchery > 10 Then
If CurrentPlayer.Skinning < 60 Then
If CurrentPlayer.Skinning = 0 Then
GameForm.Skills.AddItem ("Skinning")
End If
GameForm.PasTime (60)
MsgBox ("You have acquired additional knowledge of skinning from the " + CharacterArray(13).Name + ")."
CurrentPlayer.Skinning = CurrentPlayer.Skinning + GameForm.Learn_Skill
Else
MsgBox ("You are too knowledgeable in skinning to learn any more here.")
End If
Else
MsgBox ("You may not learn skinning until you are more knowledgeable in butchery.")
End If
If CurrentPlayer.Knifefighting > 10 And CurrentPlayer.Swordfighting > 10 Then
GameForm.PasTime (60)
MsgBox ("You have acquired additional knowledge of fencing and parrying from the " + CharacterArray(13).Name + ")."
CurrentPlayer.Fencing = CurrentPlayer.Fencing + GameForm.Learn_Skill
CurrentPlayer.Parrying = CurrentPlayer.Parrying + GameForm.Learn_Skill
Else
MsgBox ("You may not learn fencing or parrying until you are more knowledgeable in knifefighting and swordfighting.")
End If
Case 3 ' Artist
Call Show_SpeechBox(CharacterArray(14).Picture, CharacterArray(14).Speech)
If CurrentPlayer.Commerce > 10 Then
If CurrentPlayer.Begging < 60 Then
GameForm.PasTime (60)
MsgBox ("You have acquired additional knowledge of commerce from the " + CharacterArray(14).Name + ")."
Else
MsgBox ("You are too knowledgeable in begging to learn any more here.")
End If
Else
MsgBox ("You may not learn begging until you are more knowledgeable in commerce.")
End If
End Select ' index
Case 3
Select Case Index
Case 0 ' Courier
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(16).Picture, CharacterArray(16).Speech)

If CurrentPlayer.Hiking > 10 Then
    If CurrentPlayer.Jogging < 60 Then
        GameForm.PassTime (60)
        MsgBox("You have acquired additional knowledge of jogging from the " + CharacterArray(16).Name + ".")
    Else
        MsgBox("You are too knowledgeable in jogging to learn any more here.")
    End If
Else
    MsgBox("You may not learn jogging until you are sufficiently knowledgeable in hiking.")
End If
Case 1 ' Servant
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(17).Picture, CharacterArray(17).Speech)

If CurrentPlayer.Stealth < 60 Then
    GameForm.PassTime (60)
    MsgBox("You have acquired additional knowledge of stealth from the " + CharacterArray(17).Name + ".")
    CurrentPlayer.Stealth = CurrentPlayer.Stealth + GameForm.Learn_Skill
    Pause (1)
    Else
    MsgBox("You are too knowledgeable in stealth to learn any more here.")
End If
End Select ' index

Case 4
Select Case Index
Case 0 ' Merchant
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(18).Picture, CharacterArray(18).Speech)

If CurrentPlayer.Barter > 10 Then
    If CurrentPlayer.Appraising < 60 Then
        GameForm.PassTime (60)
        MsgBox("You have acquired additional knowledge of appraising from the " + CharacterArray(18).Name + ".")
        If CurrentPlayer.Appraising = 0 Then
            GameForm.Skills.AddItem ("Appraising")
            Skills.AddItem ("Appraising")
        End If
        CurrentPlayer.Appraising = CurrentPlayer.Appraising + GameForm.Learn_Skill
        Pause (1)
    Else
    MsgBox("You are too knowledgeable in appraising to learn any more here.")
    End If
Else
    MsgBox("You may not learn appraising until you are more knowledgeable in barter.")
End If

If CurrentPlayer.Jogging > 10 Then
    If CurrentPlayer.Mountaineering < 60 Then
        GameForm.PassTime (60)
        MsgBox("You have acquired additional knowledge of appraising from the " + CharacterArray(18).Name + ".")
        CurrentPlayer.Mountaineering = CurrentPlayer.Mountaineering + GameFo
Else 
MsgBox ("You are too knowledgeable in mountaineering to learn any more here.")
End If
Else 
MsgBox ("You may not learn mountaineering until you are more knowledgeable in jogging.")
End If
Case 1 ' Lady
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(19).Picture, CharacterArray(19).Speech)
Case 2 ' Drunk 
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(20).Picture, CharacterArray(20).Speech)
If CurrentPlayer.Fencing > 10 And CurrentPlayer.Parrying > 10 Then
If CurrentPlayer.Blindfighting < 60 Then 
GameForm.PassTime (60) 
MsgBox ("You have acquired additional knowledge of blindfighting from the " + CharacterArray(20).Name + ").")
Pause (1) 
Else 
MsgBox ("You are too knowledgeable in blindfighting to learn any more here.")
End If 
Else 
MsgBox ("You may not learn blindfighting until you are more knowledgeable in fencing and parrying.")
End If 
Case 3 ' Common Lady
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(21).Picture, CharacterArray(21).Speech)
Case 4 ' Girl 
Call Show_SpeechBox(CurrentImageDrive + CharacterArray(22).Picture, CharacterArray(22).Speech)
End Select ' index 
End Select ' TownNumber 
MousePointer = 0 
TownState = "Normal"
ElseIf TownState = "Begging" Then 
Select Case CurrentTown 
Case 0 
Select Case Index 
Case 0: Call Begging("village councilmember") 
Case 1: Call Begging("peasant boy") 
End Select 
Case 1 
Select Case Index 
Case 0: Call Begging("young man") 
Case 1: Call Begging("cloth merchant") 
Case 2: Call Begging("young boy") 
Case 3: Call Begging("drunken man") 
End Select 
Case 2 
Select Case Index 
Case 0: Call Begging("tavern girl") 
Case 1: Call Begging("bartender") 
Case 2: Call Begging("adventurer") 
Case 3: Call Begging("artist") 
End Select
Case 3
Select Case Index
Case 0: Call Begging("courier")
Case 1: Call Begging("servant")
End Select
Case 4
Select Case Index
Case 0: Call Begging("merchant")
Case 1: Call Begging("lady")
Case 2: Call Begging("drunk")
Case 3: Call Begging("common lady")
Case 4: Call Begging("girl")
End Select
End Select
Update_Silver
TownState = "normal"
ElseIf TownState = "PickPocketing" Then
Select Case CurrentTown
Case 0
Select Case Index
Case 0: Call PickPocketing("village councilmember")
Case 1: Call PickPocketing("peasant boy")
End Select
Case 1
Select Case Index
Case 0: Call PickPocketing("young man")
Case 1: Call PickPocketing("cloth merchant")
Case 2: Call PickPocketing("young boy")
Case 3: Call PickPocketing("drunken man")
End Select
Case 2
Select Case Index
Case 0: Call PickPocketing("tavern girl")
Case 1: Call PickPocketing("bartender")
Case 2: Call PickPocketing("adventurer")
Case 3: Call PickPocketing("artist")
End Select
Case 3
Select Case Index
Case 0: Call PickPocketing("courier")
Case 1: Call PickPocketing("servant")
End Select
Case 4
Select Case Index
Case 0: Call PickPocketing("merchant")
Case 1: Call PickPocketing("lady")
Case 2: Call PickPocketing("drunk")
Case 3: Call PickPocketing("common lady")
Case 4: Call PickPocketing("girl")
End Select
End Select
Update_Silver
TownState = "Normal"
ElseIf TownState = "Identify" Then
Select Case CurrentTown
Case 0
Select Case Index
Case 0: Call Identify(CharacterArray(0))
Case 1: Call Identify(CharacterArray(1))
End Select
Case 1
Select Case Index
Case 0: Call Identify(CharArray(3)) '("young man")
Case 1: Call Identify(CharArray(4)) '("cloth merchant")
Case 2: Call Identify(CharArray(6)) '("young boy")
Case 3: Call Identify(CharArray(7)) '("drunken man")
End Select

Case 2
Select Case Index
Case 0: Call Identify(CharArray(11)) '("tavern girl")
Case 1: Call Identify(CharArray(12)) '("bartender")
Case 2: Call Identify(CharArray(13)) '("adventurer")
Case 3: Call Identify(CharArray(14)) '("artist")
End Select

Case 3
Select Case Index
Case 0: Call Identify(CharArray(16)) '("courier")
Case 1: Call Identify(CharArray(17)) '("servant")
End Select

Case 4
Select Case Index
Case 0: Call Identify(CharArray(18)) '("merchant")
Case 1: Call Identify(CharArray(19)) '("lady")
Case 2: Call Identify(CharArray(20)) '("drunk")
Case 3: Call Identify(CharArray(21)) '("common lady")
Case 4: Call Identify(CharArray(22)) '("girl")
End Select
End If
End Sub

' Layer is exiting town
' Unload the town form
Private Sub ExitCommand_Click()
    CurrentTown = -1
    Unload Me
End Sub

' Sets up all the objects on the form
Private Sub Form_Load()
    Picture1.Left = 75 ' 400 - (Picture1.Width / 2)
    Picture1.Top = 15
    TownLabel.Left = 0
    TownLabel.Top = 0
    PlayerName.Caption = " " + CurrentPlayer.Name
    PlayerName.Left = Picture1.Width - 225
    PlayerName.Top = 0
    Update_Silver
    SilverLabel.Left = PlayerName.Left
    SilverLabel.Top = PlayerName.Top + PlayerName.Height + 5
    BuyCommand.Top = Picture1.Top + Picture1.Height + 12
    BuyCommand.Left = 75
    SellCommand.Top = BuyCommand.Top
    SellCommand.Left = 162
    RestCommand.Top = BuyCommand.Top
    RestCommand.Left = 249
    Items.Top = Picture1.Top + Picture1.Height + 4
    Items.Left = 336
    Skills.Top = Items.Top + Items.Height + 3
    Skills.Left = 336
    TalkCommand.Top = BuyCommand.Top
    TalkCommand.Left = 476
IdentifyCommand.Top = BuyCommand.Top
IdentifyCommand.Left = 563
ExitCommand.Top = BuyCommand.Top
ExitCommand.Left = 650
Dim i As Integer
For i = 1 To 19
    Load TileArray(i)
    If (i <> 0) And (i <> 1) And (i <> 2) And (i <> 3) And (i <> 4) And (i <> 15) And (i <> 16) And (i <> 18) And (i <> 19)) Then
        TileArray(i).Picture = Image1.Picture
    Else
        TileArray(i).Picture = Image2.Picture
    End If
    TileArray(i).Visible = True
    TileArray(i).Width = 130
    TileArray(i).Height = 130
Next i
TileArray(0).Picture = Image2.Picture
For i = 0 To 19
    TileArray(i).Left = 130 * (i Mod 5)
    TileArray(i).Top = 130 * (i \\ 5)
Next i
' Load player image onto entrance of town
PlayerImage.Left = TileArray(17).Left
PlayerImage.Top = TileArray(17).Top
PlayerImage.Picture = LoadPicture(CurrentImageDrive + CurrentPlayer.Picture)
' Display the buildings and characters in the current town
Select Case CurrentTown
Case 0: TownLabel.Caption = " Village"
    InnImage1.Left = TileArray(9).Left
    InnImage1.Top = TileArray(9).Top
    ShopImage1.Left = TileArray(5).Left
    ShopImage1.Top = TileArray(5).Top
    WellImage1.Left = TileArray(12).Left
    WellImage1.Top = TileArray(12).Top
    Character(0).Picture = LoadPicture(CurrentImageDrive + "councilmember.jpg")
    Character(0).Left = TileArray(8).Left
    Character(0).Top = TileArray(8).Top
    Character(1).Picture = LoadPicture(CurrentImageDrive + "peasant.gif")
    Character(1).Left = TileArray(11).Left
    Character(1).Top = TileArray(11).Top
Case 1: TownLabel.Caption = " Southern City"
    InnImage1.Left = TileArray(5).Left
    InnImage1.Top = TileArray(5).Top
    InnImage2.Left = TileArray(9).Left
    InnImage2.Top = TileArray(9).Top
    InnImage2.Visible = True
    ShopImage1.Left = TileArray(10).Left
    ShopImage1.Top = TileArray(10).Top
    ShopImage2.Left = TileArray(14).Left
    ShopImage2.Top = TileArray(14).Top
    ShopImage2.Visible = True
    WellImage1.Left = TileArray(7).Left
    WellImage1.Top = TileArray(7).Top
    Character(0).Picture = LoadPicture(CurrentImageDrive + "youngman.jpg")
    Character(0).Left = TileArray(6).Left
    Character(0).Top = TileArray(6).Top
    Character(1).Picture = LoadPicture(CurrentImageDrive + "townsmen.jpg")
    Character(1).Left = TileArray(13).Left
    Character(1).Top = TileArray(13).Top
    Character(2).Picture = LoadPicture(CurrentImageDrive + "boy.jpg")
Case 2: TownLabel.Caption = "Desert City"

InnImage1.Left = TileArray(6).Left
InnImage1.Top = TileArray(6).Top
ShopImage1.Left = TileArray(8).Left
ShopImage1.Top = TileArray(8).Top
WellImage1.Left = TileArray(5).Left
WellImage1.Top = TileArray(5).Top
WellImage2.Left = TileArray(9).Left
WellImage2.Top = TileArray(9).Top
WellImage2.Visible = True
Character(0).Picture = LoadPicture(CurrentImageDrive + "tavern.gif")
Character(0).Left = TileArray(11).Left
Character(0).Top = TileArray(11).Top
Character(1).Picture = LoadPicture(CurrentImageDrive + "bartender.gif")
Character(1).Left = TileArray(7).Left
Character(1).Top = TileArray(7).Top
Character(2).Picture = LoadPicture(CurrentImageDrive + "townadven.gif")
Character(2).Left = TileArray(13).Left
Character(2).Top = TileArray(13).Top
Character(3).Picture = LoadPicture(CurrentImageDrive + "artist.gif")
Character(3).Left = TileArray(10).Left
Character(3).Top = TileArray(10).Top

Case 3: TownLabel.Caption = "OceanSide City"

InnImage1.Left = TileArray(5).Left
InnImage1.Top = TileArray(5).Top
InnImage2.Left = TileArray(9).Left
InnImage2.Top = TileArray(9).Top
InnImage2.Visible = True
ShopImage1.Left = TileArray(12).Left
ShopImage1.Top = TileArray(12).Top
WellImage1.Left = TileArray(7).Left
WellImage1.Top = TileArray(7).Top
CottageImage.Left = TileArray(10).Left
CottageImage.Top = TileArray(10).Top
CottageImage.Visible = True
Character(0).Picture = LoadPicture(CurrentImageDrive + "courier.gif")
Character(0).Left = TileArray(11).Left
Character(0).Top = TileArray(11).Top
Character(1).Picture = LoadPicture(CurrentImageDrive + "servant.jpg")
Character(1).Left = TileArray(13).Left
Character(1).Top = TileArray(13).Top

Case 4: TownLabel = "StoneWall"

InnImage1.Left = TileArray(7).Left
InnImage1.Top = TileArray(7).Top
ShopImage1.Left = TileArray(10).Left
ShopImage1.Top = TileArray(10).Top
ShopImage2.Left = TileArray(14).Left
ShopImage2.Top = TileArray(14).Top
ShopImage2.Visible = True
WellImage1.Left = TileArray(12).Left
WellImage1.Top = TileArray(12).Top
Character(0).Picture = LoadPicture(CurrentImageDrive + "townsmen.jpg")
Character(0).Left = TileArray(11).Left
Character(0).Top = TileArray(11).Top
Character(1).Picture = LoadPicture(CurrentImageDrive + "lady.gif")
Character(1).Left = TileArray(8).Left
Character (1).Top = TileArray(8).Top
Character (2).Picture = LoadPicture(CurrentImageDrive + "drunk.jpg")
Character (2).Left = TileArray(6).Left
Character (2).Top = TileArray(6).Top
Character (3).Picture = LoadPicture(CurrentImageDrive + "common.gif")
Character (3).Left = TileArray(13).Left
Character (3).Top = TileArray(13).Top
Character (4).Picture = LoadPicture(CurrentImageDrive + "girl.gif")
Character (4).Left = TileArray(9).Left
Character (4).Top = TileArray(9).Top
End Select
' Initialize the TownState
TownState = "Normal"
' Update the label to display the player's current amount of money
Update_Skills
For i = 0 To CurrentPlayer.NextIndex - 1
   Items.List(i) = GameForm.Equipment.List(i)
Next i
End Sub
' Displays a message describing the character that
' was selected
Private Sub Identify(ByVal Person As Character)
   If CurrentPlayer.Itemlore < 30 Then
      MsgBox ("This is " + Person.Description1 + ".")
   ElseIf CurrentPlayer.Itemlore > 29 And CurrentPlayer.Itemlore < 65 Then
      MsgBox ("This is " + Person.Description2 + ".")
   ElseIf CurrentPlayer.Itemlore > 64 Then
      MsgBox ("This is " + Person.Description3 + ".")
   End If
   MousePointer = 0
   TownState = "Normal"
End Sub
' Changes state to normal if the state was previously
' identify
' Changes state to identify if the state was anything other
' than identify
' Updates the mousepointer accordingly
Private Sub IdentifyCommand_Click()
   If TownState = "Identify" Then
      TownState = "Normal"
      MousePointer = 0
   Else
      TownState = "Identify"
      MousePointer = 99
      MouseIcon = LoadPicture(CurrentImageDrive + "bullseye.ico")
   End If
End Sub
' Player is using or wearing an item in his inventory
Private Sub Items_Click()
   If Items.text <> "" Then
      ' If the player wants to identify the item
      If TownState = "Eye" Then
         TownState = "Normal"
         Call GameForm.Identify_Item(CurrentPlayer.Inventoty(Items.ListIndex))
      ' If player wants to sell the item
      ElseIf TownState = "Sell" Then
         SelectedItemLabel = CurrentPlayer.Inventoty(Items.ListIndex).Name
Else
    Select Case CurrentPlayer.Inventory(Items.ListIndex).ItemType
        ' If item is Food
        Case 8:
            MsgBox ("You have eaten the " + CurrentPlayer.Inventory(Items.ListIndex).Name + ".")
            ' Remove item from inventory
            Remove (Items.ListIndex)
            CurrentPlayer.Fatigue = CurrentPlayer.Fatigue + 2
        ' Elseif item is Clothing
        Case 7:
            Select Case CurrentPlayer.Inventory(Items.ListIndex).Wear_Position
                Case 1: Set CurrentPlayer.Head = CurrentPlayer.Inventory(Items.ListIndex)
                    MsgBox ("You are now wearing " & CurrentPlayer.Inventory(Items.ListIndex).Name & " on your head.")
                    MsgBox ("You are now wearing " & CurrentPlayer.Inventory(Items.ListIndex).Name & " on your body.")
                Case 3: Set CurrentPlayer.Feet = CurrentPlayer.Inventory(Items.ListIndex)
                    MsgBox ("You are now wearing " & CurrentPlayer.Inventory(Items.ListIndex).Name & " on your feet.")
                Case 4: Set CurrentPlayer.Aboutbody = CurrentPlayer.Inventory(Items.ListIndex)
                    MsgBox ("You are now wearing " & CurrentPlayer.Inventory(Items.ListIndex).Name & " about your body.")
                Case 6: Set CurrentPlayer.Shield = CurrentPlayer.Inventory(Items.ListIndex)
                    MsgBox ("You are now wearing " & CurrentPlayer.Inventory(Items.ListIndex).Name & " as a shield.")
            End Select
            GameForm.Check_Armor
        ' Elseif item is Armor
        Case 6:
            Select Case CurrentPlayer.Inventory(Items.ListIndex).Wear_Position
                Case 1: Set CurrentPlayer.Head = CurrentPlayer.Inventory(Items.ListIndex)
                    MsgBox ("You are now wearing " & CurrentPlayer.Inventory(Items.ListIndex).Name & " on your head.")
                    MsgBox ("You are now wearing " & CurrentPlayer.Inventory(Items.ListIndex).Name & " on your body.")
                Case 3: Set CurrentPlayer.Feet = CurrentPlayer.Inventory(Items.ListIndex)
                    MsgBox ("You are now wearing " & CurrentPlayer.Inventory(Items.ListIndex).Name & " on your feet.")
                Case 4: Set CurrentPlayer.Aboutbody = CurrentPlayer.Inventory(Items.ListIndex)
                    MsgBox ("You are now wearing " & CurrentPlayer.Inventory(Items.ListIndex).Name & " about your body.")
                Case 6: Set CurrentPlayer.Shield = CurrentPlayer.Inventory(Items.ListIndex)
                    MsgBox ("You are now wearing " & CurrentPlayer.Inventory(Items.ListIndex).Name & " as a shield.")
            End Select
            GameForm.Check_Armor
        ' Elseif item is a Weapon
        Case 1 To 5:
            Set CurrentPlayer.Wielded = CurrentPlayer.Inventory(Items.ListIndex)
MsgBox("You are now wielding " & CurrentPlayer.Inventory(Items.ListIndex).Name & ".")
End Select
End If
End If
End Sub

' Player is performing
Public Sub Performance()
    ' Make sure player can still perform today
    If CurrentPlayer.Performed < 3 Then
        ' Calculate a modifier based on the player's social skill
        Dim modifier As Byte
        If CurrentPlayer.Social > 0 And CurrentPlayer.Social < 15 Then
            modifier = 1
        ElseIf CurrentPlayer.Social > 14 And CurrentPlayer.Social < 30 Then
            modifier = 2
        ElseIf CurrentPlayer.Social > 29 And CurrentPlayer.Social < 60 Then
            modifier = 4
        ElseIf CurrentPlayer.Social > 59 Then
            modifier = 6
        End If
        ' Make sure player has enough fatigue points
        If CurrentPlayer.Fatigue < 15 Then
            MsgBox("You are too tired.")
        End If
        ' If player passes skill check
        ElseIf (CurrentPlayer.Performance + modifier) > GameForm.RandomNumber() Then
            ' Determine what player obtains based on his performance skill rating
            If CurrentPlayer.Performance < 31 And CurrentPlayer.Performance > 0 Then
                MsgBox("The passersby throw you some bread.")
                Add (ItemArray(26))
            ElseIf CurrentPlayer.Performance > 30 And CurrentPlayer.Performance < 61 Then
                MsgBox("The passersby throw you a few coins.")
                CurrentPlayer.Money = CurrentPlayer.Money + (GameForm.RandomNumber() Mod 5) + 1
            Else
                MsgBox("The passersby throw you some coins.")
                CurrentPlayer.Money = CurrentPlayer.Money + (GameForm.RandomNumber() Mod 10) + 1
            End If
            GameForm.PassTime(30)
            CurrentPlayer.Performed = CurrentPlayer.Performed + 1
        ' If player fails the skill check
        Else
            GameForm.PassTime(20)
            CurrentPlayer.Fatigue = CurrentPlayer.Fatigue - 10
            MsgBox("You were unsuccessful at performing." & Chr(13) & "The passersby laughed and walked on.")
            CurrentPlayer.Performance = CurrentPlayer.Performance + GameForm.Improve_Skill
            If CurrentPlayer.Social < GameForm.RandomNumber() Then
            End If
        End If
    Else
        MsgBox("You can only perform 3 times a day.")
    End If
End Sub

' Remove Item from Inventory
Private Sub Remove(theIndex As Byte)
    ' Remove item from inventory
    Set CurrentPlayer.Inventory(theIndex) = Nothing
Removes item from equipment list

```vba
Items.RemoveItem (theIndex)
GameForm.Equipment.RemoveItem (theIndex)
```

+ Reset array to account for empty slot

```vba
Dim i As Byte
For i = theIndex To 99
    Set CurrentPlayer.Inventory(i) = CurrentPlayer.Inventory(i + 1)
    Set CurrentPlayer.Inventory(i + 1) = Nothing
Next i
```

- Subtract one from the next index

```vba
CurrentPlayer.NextIndex = CurrentPlayer.NextIndex - 1
```

Private Sub MessageBox_GotFocus()
    MessageBox.Visible = False
End Sub

PickPocketing

- The player is attempting to steal money from a character.

```vba
Public Sub PickPocketing(ByVal Name As String)
    ' First make sure player has enough fatigue points
    If CurrentPlayer.Fatigue < 2 Then
        MsgBox ("You are too tired."")
    Else
        ' Determine a modifier based on what character is being pickpocketed
        Dim modifier As Integer
        modifier = 0
        If (Name = "young boy") Or (Name = "young man") Or (Name = "drunk") Or (Name = "drunken man") Then
            modifier = 5
        ElseIf (Name = "cloth merchant") Or (Name = "merchant") Then
            modifier = -5
        ElseIf (Name = "adventurer") Or (Name = "town guard") Then
            modifier = -10
        End If
        ' If player passes the skill check
        If (CurrentPlayer.PickPocketing) + modifier > GameForm.RandomNumber() Then
            ' Determine how much money is stolen based on the player's skill
            Dim Money As Byte
            If CurrentPlayer.PickPocketing > 0 And CurrentPlayer.PickPocketing < 50 Then
                Money = (GameForm.RandomNumber() Mod 5) + 1
            ElseIf CurrentPlayer.PickPocketing > 49 And CurrentPlayer.PickPocketing < 75 Then
                Money = (GameForm.RandomNumber() Mod 10) + 1
            ElseIf CurrentPlayer.PickPocketing > 74 Then
                Money = (GameForm.RandomNumber() Mod 15) + 1
            End If
            ' Display an appropriate message
            MsgBox ("You have obtained " & Money & " silver coins by pickpocketing the " & Name & ".")
            ' Add the money
            ' Pass some game time
            GameForm.PassTime (2)
            ' Subtract fatigue points
        Else
            ' If player fails the skill check
            Else
                ' Pass some game time
                GameForm.PassTime (2)
```
' Subtract fatigue
' Determine if player can talk his way out of the situation
If CurrentPlayer.Social > 90 Then
    ' Display an appropriate message
    MsgBox ("You were unsuccessful at pickpocketing the " & Name & "." & Chr(13) & "
    However, due to your high social skill, the townguard has agreed to let you go.")
    ' Give the player a chance to improve his skill
End If

If CurrentPlayer.PickPocketing > 90 Then
    MsgBox ("You were unsuccessful at pickpocketing the " & Name & "." & Chr(13) & "
The " & Name & " has called the townguard!")
    Call the town guard
    CurrentMonsterNumber = 13
    Load the attack form
    AttackForm.Show vbModal, Me
    ' Give the player a chance to improve his skill
End If

Else
    ' Display an appropriate message
    MsgBox ("You were unsuccessful at pickpocketing the " & Name & "." & Chr(13) & "
End If
End If
End Sub

' Player is resting for 12 hours
Private Sub RestCommand_Click()
    Call GameForm.PassTime(60 * 12)
    Dim tempfatigue As Integer
    tempfatigue = CurrentPlayer.Fatigue + (12 * (GameForm.RandomNumber() Mod 20) + 1 + 15)
    ' Make sure we have not exceeded the player's max fatigue points
    If tempfatigue > CurrentPlayer.Max_Fp Then
    Else
        CurrentPlayer.Fatigue = tempfatigue
    End If
    ' Make sure we have not exceeded the player's max hit points
    If CurrentPlayer.HitPoints > CurrentPlayer.Max_Hp Then
    End If
    ' Update progress bars
    GameForm.HitPoints.Value = CurrentPlayer.HitPoints
    GameForm.Fatigue.Value = CurrentPlayer.Fatigue
    MsgBox ("You feel much better after 12 hours of rest.")
End Sub

' Display the Sell Frame
Private Sub SellCommand_Click()
    SelectedItemLabel.Caption = "No Item Selected"
    SellFrame.Left = 400 - (SellFrame.Width / 2)
    SellFrame.Top = 300 - (SellFrame.Height / 2)
    SellFrame.Visible = True
    TownState = "Sell"
End Sub

' Player is selling an item
Private Sub SellItemCommand_Click()
' Make sure the player has selected an item
If Items.text = "" Or SelectedItemLabel.Caption = "No Item Selected" Then
    MsgBox ("You have not selected an item to sell.")
ElseIf CurrentPlayer.InventorY(Items.ListIndex).Value = -1 Then
    MsgBox ("You cannot sell this item!"")
Else
    Dim extra As Single
    Dim sellprice As Integer
    If CurrentPlayer.Inventory(Items.ListIndex).ItemType < 8 Then
        GameForm.PassTime (2)
        ' Determine an extra bonus based on the player's commerce and barter ratings
        If CurrentPlayer.Commerce < 51 Then
            extra = 0.02
        ElseIf CurrentPlayer.Commerce > 50 And CurrentPlayer.Commerce < 75 Then
            extra = 0.04
        ElseIf CurrentPlayer.Commerce > 74 Then
            extra = 0.06
        End If
        If CurrentPlayer.Barter < 21 Then
            extra = extra + 0.02
        ElseIf CurrentPlayer.Barter > 20 And CurrentPlayer.Barter < 51 Then
            extra = extra + 0.05
        ElseIf CurrentPlayer.Barter > 50 And CurrentPlayer.Barter < 75 Then
            extra = extra + 0.1
        ElseIf CurrentPlayer.Barter > 74 Then
            extra = extra + 0.15
        End If
    ' Calculate a sell price
    ' Increase Money
    Call Show_MessageBox(CurrentImageDrive + CurrentPlayer.InventorY(Items.ListIndex).Picture, "You have sold the " & CurrentPlayer.Inventory(Items.ListIndex).Name + " for " & sellprice & " silver pieces.")
    ' Remove the item from the player's body
    If CurrentPlayer.Inventory(Items.ListIndex).Name = CurrentPlayer.Aboutbody.Name Then
        Set CurrentPlayer.Aboutbody = ItemArray(55)
    ElseIf CurrentPlayer.Inventory(Items.ListIndex).Name = CurrentPlayer.Body.Name Then
        Set CurrentPlayer.Body = ItemArray(55)
    ElseIf CurrentPlayer.Inventory(Items.ListIndex).Name = CurrentPlayer.Feet.Name Then
        Set CurrentPlayer.Feet = ItemArray(55)
    ElseIf CurrentPlayer.Inventory(Items.ListIndex).Name = CurrentPlayer.Head.Name Then
        Set CurrentPlayer.Head = ItemArray(55)
    ElseIf CurrentPlayer.Inventory(Items.ListIndex).Name = CurrentPlayer.Wielded.Name Then
        Set CurrentPlayer.Wielded = ItemArray(55)
    ElseIf CurrentPlayer.Inventory(Items.ListIndex).Name = CurrentPlayer.Shield.Name Then
        Set CurrentPlayer.Shield = ItemArray(55)
    End If
    ' Update the player's armor rating
    GameForm.Check_Armor
    ' Remove item from inventory
    Remove (Items.ListIndex)
EndIf
Or CurrentPlayer.Inventory(Items.ListIndex).Name = "wood" Or CurrentPlayer.Inventory(Items.ListIndex).Name = "wood"
Or CurrentPlayer.Inventory(Items.ListIndex).Name = "fish" Or CurrentPlayer.Inventory(Items.ListIndex).Name = "meat"
Or CurrentPlayer.Inventory(Items.ListIndex).Name = "roots"
Or CurrentPlayer.Inventory(Items.ListIndex).Name = "herbs"

GameForm.PassTime (2)
' Determine the extra bonus based on the player's
' commerce and barter skill ratings
If CurrentPlayer.Commerce < 51 Then
    extra = 0.02
ElseIf CurrentPlayer.Commerce > 50 And CurrentPlayer.Commerce < 75 Then
    extra = 0.04
ElseIf CurrentPlayer.Commerce > 74 Then
    extra = 0.06
End If
If CurrentPlayer.Barter < 21 Then
    extra = extra + 0.02
ElseIf CurrentPlayer.Barter > 20 And CurrentPlayer.Barter < 51 Then
    extra = extra + 0.05
ElseIf CurrentPlayer.Barter > 50 And CurrentPlayer.Barter < 75 Then
    extra = extra + 0.1
ElseIf CurrentPlayer.Barter > 74 Then
    extra = extra + 0.15
End If
' Calculate a sell price
' Increase Money
Call Show_MessageBox(CurrentImageDrive + CurrentPlayer.Inventory(Items.ListIndex).Picture, "You have sold the " & CurrentPlayer.Inventory(Items.ListIndex).Name & " for " & sellprice & " silver pieces.")
' Remove item from inventory
Remove (Items.ListIndex)
Else
    MsgBox ("You can not sell that item.")
End If
' Update the label displaying the player's current amount of money
Update_Silver
End Sub

' Show Message_Box
' Shows a message box with the given image and message
Public Sub Show_MessageBox(ByVal image As String, ByVal text As String)
    MessageBox.MessagePicture = image
    MessageBox.LabelText = text
    MessageBox.Top = 150
    MessageBox.Left = 400 - (MessageBox.Width / 2)
    MessageBox.Height = 225
    MessageBox.Width = 350
    MessageBox.Visible = True
End Sub

' Show Speech_Box
' Shows a speech box with the given image and message
Private Sub Show_SpeechBox(ByVal image As String, ByVal text As String)
    SpeechBox.image = image
    SpeechBox.LabelText = text
Private Sub Skills_Click()
    Select Case Skills.text
        Case "Appraising": Appraising
        Case "Begging": TownState = "Begging"
        Case "Performance": Performance
        Case "Pick Pocketing": TownState = "PickPocketing"
    End Select
End Sub

Private Sub Update_Silver()
End Sub

Private Sub Update_Skills()
    If CurrentPlayer.Appraising > 0 Then Skills.AddItem ("Appraising")
    If CurrentPlayer.Barter > 0 Then Skills.AddItem ("Barter")
    If CurrentPlayer.Begging > 0 Then Skills.AddItem ("Begging")
    If CurrentPlayer.Performance > 0 Then Skills.AddItem ("Performance")
    If CurrentPlayer.pickPocketing > 0 Then Skills.AddItem ("Pick Pocketing")
End Sub

Private Sub SpeechBox_GotFocus()
    SpeechBox.Visible = False
End Sub

Private Sub TalkCommand_Click()
    If TownState = "Talk" Then
        TownState = "Normal"
        MousePointer = 0
    Else
        TownState = "Talk"
        MousePointer = 39
        MouseIcon = LoadPicture(CurrentImageDrive + "mouth.ico")
    End If
End Sub
' Comments pertaining to entire program:
' This program is set up for a screen size of 400 x 600.

' CurrentPlayer is declared here so that all functions
' and forms can access player data.
Public CurrentPlayer As New Player
Public ItemArray(75) As New Item
Public CharacterArray(50) As New Character
Public CurrentMonsterNumber As Integer
Public CurrentTown As Integer

Public Const currentdrive = "d:\"
Public Const DRIVE = currentdrive + "images\"
Public Const CurrentImageDrive = currentdrive + "game\"

Type playertype
    Name As String
    Age As Byte
    Sex As Boolean
    Height As Byte
    Weight As Byte
    Money As Integer
    HitPoints As Integer
    Fatigue As Byte
    Evade As Byte
    Armor As Byte
    Agility As Byte
    Strength As Byte
    Fitness As Byte
    Health As Byte
    Intelligence As Byte
    Personality As Byte
    Picture As String
    Hitbonus As Integer
    Appraising As Byte
    Archery As Byte
    Armorlore As Byte
    Bandaging As Byte
    Barter As Byte
    Begging As Byte
    Blindfighting As Byte
    Bluntweaponfighting As Byte
    Brawling As Byte
    Butchery As Byte
    Camping As Byte
    Carpentry As Byte
    Carving As Byte
    Chopping As Byte
    Commerce As Byte
    Cooking As Byte
    Fencing As Byte
    Firebuilding As Byte
    Fishing As Byte
    Foraging As Byte
    Handtohandcombat As Byte
    Herblore As Byte
    Hiking As Byte
    Hunting As Byte
    Itemlore As Byte
    Jogging As Byte
    Kicking As Byte
Module1 - 2

Knifefighting As Byte
Lockpicking As Byte
Max_Fp As Byte
Max_Hp As Byte
Mountaineering As Byte
Parrying As Byte
Performance As Byte
PickPocketing As Byte
Punching As Byte
Skinning As Byte
Social As Byte
Swimming As Byte
Swordfighting As Byte
Weaponlore As Byte
Wrestling As Byte
Parry As Boolean
Inventory(0 To 99) As Byte
Nextindex As Byte
Body As Byte
Aboutbody As Byte
Feet As Byte
Wielded As Byte
Shield As Byte
Head As Byte
Damagebonus As Integer
Poisoned As Boolean
Bandaged As Boolean
Stealth As Byte
Performed As Byte
Begged As Byte
Cur_x As Byte
Cur_y As Byte
Cur_minutes As Long
End Type

Sub Pause(Seconds As Single)
' Wait for a specified number of seconds.
'--------------------------------------------------------
Dim EndTime As Single
EndTime = Timer + Seconds
While Timer < EndTime
    DoEvents
Wend
End Sub
'local variable(s) to hold property value(s)
Private mvarName As String 'local copy
Private mvarIDNumber As Integer 'local copy
Private mvarPicture As String 'local copy
Private mvarDescription1 As String 'local copy
Private mvarDescription2 As String 'local copy
Private mvarDescription3 As String 'local copy
Private mvarSpeech As String 'local copy

Public Property Let Speech(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Speech = 5
    mvarSpeech = vData
End Property

Public Property Get Speech() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Speech
    Speech = mvarSpeech
End Property

Public Property Let Description3(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Description3 = 5
    mvarDescription3 = vData
End Property

Public Property Get Description3() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Description3
    Description3 = mvarDescription3
End Property

Public Property Let Description2(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Description2 = 5
    mvarDescription2 = vData
End Property

Public Property Get Description2() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Description2
    Description2 = mvarDescription2
End Property

Public Property Let Description1(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Description1 = 5
    mvarDescription1 = vData
End Property
Public Property Get Description1() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Description1
    Description1 = mvarDescription1
End Property

Public Property Let Picture(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Picture = 5
    mvarPicture = vData
End Property

Public Property Get Picture() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Picture
    Picture = mvarPicture
End Property

Public Property Let IDNumber(ByVal vData As Integer)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Ident = 5
    mvarIdent = vData
End Property

Public Property Get IDNumber() As Integer
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Ident
    IDNumber = mvarIdent
End Property

Public Property Let Name(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Name = 5
    mvarName = vData
End Property

Public Property Get Name() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Name
    Name = mvarName
End Property
Item - 1

' local variable(s) to hold property value(s)
' Name of item
Private mvarName As String 'local copy
' String representing the location of the item's picture file
Private mvarPicture As String 'local copy
' Short, vague description
Private mvarDescription1 As String 'local copy
' item's weight in stones
Private mvarWeight As Byte 'local copy
' position that the item is worn on
' following codes available
' 1 - Head
' 2 - Body
' 3 - Feet
' 4 - AboutBody
' 5 - Wielded
' 6 - Shield
Private mvarWear_Position As Byte 'local copy
' Height of item in Height
Private mvarHeight As Byte 'local copy
' type that the item is
' following codes available
' 1 - Sword
' 2 - Pierce
' 3 - Blunt
' 4 - Axe
' 5 - Range
' 6 - Armor
' 7 - Clothing
' 8 - Food
' 9 - Light
' 10 - Other
Private mvarItemType As Byte 'local copy

Private mvarSpecial1 As Byte 'local copy
Private mvarSpecial2 As Integer 'local copy
' short description
Private mvarDescription2 As String 'local copy
' long description
Private mvarDescription3 As String 'local copy
' Value in silver pieces
Private mvarValue As Integer 'local copy
' local variable(s) to hold property value(s)
Private mvarIDNumber As Byte 'local copy

Public Property Let IDNumber(ByVal vData As Byte)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.IDNumber = 5
   mvarIDNumber = vData
End Property

Public Property Get IDNumber() As Byte
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.IDNumber
   IDNumber = mvarIDNumber
End Property
Public Property Let Value(ByVal vData As Integer)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Value = 5
    mvarValue = vData
End Property

Public Property Get Value() As Integer
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Value
    Value = mvarValue
End Property

Public Property Let Description3(ByVal vData As String)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Description3 = 5
    mvarDescription3 = vData
End Property

Public Property Get Description3() As String
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Description3
    Description3 = mvarDescription3
End Property

Public Property Let Description2(ByVal vData As String)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Description2 = 5
    mvarDescription2 = vData
End Property

Public Property Get Description2() As String
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Description2
    Description2 = mvarDescription2
End Property

Public Property Let ItemType(ByVal vData As Byte)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.ItemType = 5
    mvarItemType = vData
End Property

Public Property Get ItemType() As Byte
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.ItemType
    ItemType = mvarItemType
End Property

Public Property Let Special12(ByVal vData As Integer)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Special12 = 5
    mvarSpecial12 = vData
End Property

Public Property Get Special12() As Integer
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Special12
    Special12 = mvarSpecial12
End Property
Public Property Let Special1(ByVal vData As Byte)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Special1 = 5
      mvarSpecial1 = vData
End Property

Public Property Get Special1() As Byte
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Special1
      Special1 = mvarSpecial1
End Property

Public Property Let Height(ByVal vData As Byte)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Height = 5
      mvarHands = vData
End Property

Public Property Get Height() As Byte
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Height
      Height = mvarHands
End Property

Public Property Let Wear_Position(ByVal vData As Byte)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Wear_Position = 5
      mvarWear_Position = vData
End Property

Public Property Get Wear_Position() As Byte
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Wear_Position
      Wear_Position = mvarWear_Position
End Property

Public Property Let Weight(ByVal vData As Byte)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Weight = 5
      mvarWeight = vData
End Property

Public Property Get Weight() As Byte
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Weight
      Weight = mvarWeight
End Property

Public Property Let Description1(ByVal vData As String)
' used when assigning a value to the property, on the left side of an assignment.
Item - 4

'Syntax: X.Description = 5
    mvarDescription = vData
End Property

Public Property Get Description1() As String
    'used when retrieving value of a property, on the right side of an assignment.
    'Syntax: Debug.Print X.Description
    Description1 = mvarDescription
End Property

Public Property Let Picture(ByVal vData As String)
    'used when assigning a value to the property, on the left side of an assignment.
    'Syntax: X.Picture = 5
    mvarPicture = vData
End Property

Public Property Get Picture() As String
    'used when retrieving value of a property, on the right side of an assignment.
    'Syntax: Debug.Print X.Picture
    Picture = mvarPicture
End Property

Public Property Let Name(ByVal vData As String)
    'used when assigning a value to the property, on the left side of an assignment.
    'Syntax: X.Name = 5
    mvarName = vData
End Property

Public Property Get Name() As String
    'used when retrieving value of a property, on the right side of an assignment.
    'Syntax: Debug.Print X.Name
    Name = mvarName
End Property
'local variable(s) to hold property value(s)
Private mvarName As String 'local copy name of the monster
Private mvarTotal_HitPoints As Integer 'local copy total life monster has
Private mvarCurrent_HitPoints As Integer 'local copy current life
Private mvarFatigue As Byte 'local copy unimplemented
Private mvarPicture As String 'local copy picture of the monster
Private mvarArmor As Byte 'local copy ability to soak damage
Private mvarEvade As Byte 'local copy ability to avoid being hit
Private mvarDamage As Byte 'local copy number of sides to the dice
Private mvarDamageDice As Byte 'local copy number of dice rolled
Private mvarToHit As Byte 'local copy bonus to the hit roll
Private mvarToDam As Byte 'local copy bonus to the dam roll
Private mvarLightAttackPercent As Byte 'local copy how often a light attack is used
Private mvarMediumAttackPercent As Byte 'local copy how often a medium attack is used
Private mvarFleePercent As Byte 'local copy point at which the monster will attempt to flee
Private mvarNoFlee As Byte 'local copy how often monster will attempt to keep player from fleeing
Private mvarHasSpecial As Boolean 'local copy does the monster have a special attack
Private mvarSpecialAttackPercent As Byte 'local copy how often a special is used
Private mvarSpecial As Byte 'local copy the special
Private mvarDrift As Integer 'local copy unimplemented
Private mvarLevel As Byte 'local copy level of the monster, how tough it is
Private mvarMonsterType As Byte 'local copy animal, humanoid, giant or monster

Public Property Let MonsterType(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Type = 5
    mvarType = vData
End Property

Public Property Get MonsterType() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Type
    MonsterType = mvarMonsterType
End Property

Public Property Let Level (ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Level = 5
    mvarLevel = vData
End Property

Public Property Get Level() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Level
    Level = mvarLevel
End Property

Public Property Let Drift (ByVal vData As Integer)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Drift = 5
    mvarDrift = vData
End Property

Public Property Get Drift() As Integer
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Drift
    Drift = mvarDrift
Public Property Let Special(ByVal vData As Byte) 'used when assigning a value to the property, on the left side of an assignment. 'Syntax: X.Special = 5
    mvarSpecial = vData
End Property

Public Property Get Special() As Byte 'used when retrieving value of a property, on the right side of an assignment. 'Syntax: Debug.Print X.Special
    Special = mvarSpecial
End Property

Public Property Let SpecialAttackPercent(ByVal vData As Byte) 'used when assigning a value to the property, on the left side of an assignment. 'Syntax: X.SpecialAttackPercent = 5
    mvarSpecialAttackPercent = vData
End Property

Public Property Get SpecialAttackPercent() As Byte 'used when retrieving value of a property, on the right side of an assignment. 'Syntax: Debug.Print X.SpecialAttackPercent
    SpecialAttackPercent = mvarSpecialAttackPercent
End Property

Public Property Let HasSpecial(ByVal vData As Boolean) 'used when assigning a value to the property, on the left side of an assignment. 'Syntax: X.HasSpecial = 5
    mvarHasSpecial = vData
End Property

Public Property Get HasSpecial() As Boolean 'used when retrieving value of a property, on the right side of an assignment. 'Syntax: Debug.Print X.HasSpecial
    HasSpecial = mvarHasSpecial
End Property

Public Property Let NoFlee(ByVal vData As Byte) 'used when assigning a value to the property, on the left side of an assignment. 'Syntax: X.NoFlee = 5
    mvarNoFlee = vData
End Property

Public Property Get NoFlee() As Byte 'used when retrieving value of a property, on the right side of an assignment. 'Syntax: Debug.Print X.NoFlee
    NoFlee = mvarNoFlee
End Property
Public Property Let FleePercent(ByVal vData As Byte)
    ' used when assigning a value to the property, on the left side of an assignment.
    'Syntax: X.FleePercent = 5
    mvarFleePercent = vData
End Property

Public Property Get FleePercent() As Byte
    ' used when retrieving value of a property, on the right side of an assignment.
    'Syntax: Debug.Print X.FleePercent
    FleePercent = mvarFleePercent
End Property

Public Property Let MediumAttackPercent(ByVal vData As Byte)
    ' used when assigning a value to the property, on the left side of an assignment.
    'Syntax: X.MediumAttackPercent = 5
    mvarMediumAttackPercent = vData
End Property

Public Property Get MediumAttackPercent() As Byte
    ' used when retrieving value of a property, on the right side of an assignment.
    'Syntax: Debug.Print X.MediumAttackPercent
    MediumAttackPercent = mvarMediumAttackPercent
End Property

Public Property Let LightAttackPercent(ByVal vData As Byte)
    ' used when assigning a value to the property, on the left side of an assignment.
    'Syntax: X.LightAttackPercent = 5
    mvarLightAttackPercent = vData
End Property

Public Property Get LightAttackPercent() As Byte
    ' used when retrieving value of a property, on the right side of an assignment.
    'Syntax: Debug.Print X.LightAttackPercent
    LightAttackPercent = mvarLightAttackPercent
End Property

Public Property Let ToDam(ByVal vData As Byte)
    ' used when assigning a value to the property, on the left side of an assignment.
    'Syntax: X.ToDam = 5
    mvarToDam = vData
End Property

Public Property Get ToDam() As Byte
    ' used when retrieving value of a property, on the right side of an assignment.
    'Syntax: Debug.Print X.ToDam
    ToDam = mvarToDam
End Property
Public Property Let ToHit(ByVal vData As Byte)  
"used when assigning a value to the property, on the left side of an assignment.  
' syntax: X.ToHit = 5  
mvarToHit = vData  
End Property

Public Property Get ToHit() As Byte  
"used when retrieving value of a property, on the right side of an assignment.  
'Syntax: Debug.Print X.ToHit  
ToHit = mvarToHit  
End Property

Public Property Let DamageDice(ByVal vData As Byte)  
"used when assigning a value to the property, on the left side of an assignment.  
'Syntax: X.DamageDice = 5  
mvarDamageDice = vData  
End Property

Public Property Get DamageDice() As Byte  
"used when retrieving value of a property, on the right side of an assignment.  
'Syntax: Debug.Print X.DamageDice  
DamageDice = mvarDamageDice  
End Property

Public Property Let Damage(ByVal vData As Byte)  
"used when assigning a value to the property, on the left side of an assignment.  
'Syntax: X.Damage = 5  
mvarDamage = vData  
End Property

Public Property Get Damage() As Byte  
"used when retrieving value of a property, on the right side of an assignment.  
'Syntax: Debug.Print X.Damage  
Damage = mvarDamage  
End Property

Public Property Let Evade(ByVal vData As Byte)  
"used when assigning a value to the property, on the left side of an assignment.  
'Syntax: X.Evade = 5  
mvarEvade = vData  
End Property

Public Property Get Evade() As Byte  
"used when retrieving value of a property, on the right side of an assignment.  
'Syntax: Debug.Print X.Evade  
Evade = mvarEvade  
End Property
Public Property Let Armor(ByVal vData As Byte)
' used when assigning a value to the property, on the left side of an assignment.
' Syntax: X.Armor = 5
    mvarArmor = vData
End Property

Public Property Get Armor() As Byte
' used when retrieving value of a property, on the right side of an assignment.
' Syntax: Debug.Print X.Armor
    Armor = mvarArmor
End Property

Public Property Let Picture(ByVal vData As String)
' used when assigning a value to the property, on the left side of an assignment.
' Syntax: X.Picture = 5
    mvarPicture = vData
End Property

Public Property Get Picture() As String
' used when retrieving value of a property, on the right side of an assignment.
' Syntax: Debug.Print X.Picture
    Picture = mvarPicture
End Property

Public Property Let Fatigue(ByVal vData As Byte)
' used when assigning a value to the property, on the left side of an assignment.
' Syntax: X.Fatigue = 5
    mvarFatigue = vData
End Property

Public Property Get Fatigue() As Byte
' used when retrieving value of a property, on the right side of an assignment.
' Syntax: Debug.Print X.Fatigue
    Fatigue = mvarFatigue
End Property

Public Property Let Current_HitPoints(ByVal vData As Integer)
' used when assigning a value to the property, on the left side of an assignment.
' Syntax: X.Current_HitPoints = 5
    mvarCurrent_HitPoints = vData
End Property

Public Property Get Current_HitPoints() As Integer
' used when retrieving value of a property, on the right side of an assignment.
' Syntax: Debug.Print X.Current_HitPoints
    Current_HitPoints = mvarCurrent_HitPoints
End Property
Public Property Let Total_HitPoints(ByVal vData As Integer)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Total_HitPoints = 5
    mvarTotal_HitPoints = vData
End Property

Public Property Get Total_HitPoints() As Integer
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Total_HitPoints
    Total_HitPoints = mvarTotal_HitPoints
End Property

Public Property Let Name(ByVal vData As String)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Name = 5
    mvarName = vData
End Property

Public Property Get Name() As String
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Name
    Name = mvarName
End Property
' Class Player
' Functions for each data member allowing them
to be set or retrieved by the program
Local variable(s) to hold property value(s)

Private mvarName As String 'local copy
Private mvarAge As Byte 'local copy
Private mvarSex As Boolean 'local copy
Private mvarHeight As Byte 'local copy
Private mvarWeight As Byte 'local copy
Private mvarMoney As Integer 'local copy
Private mvarHitPoints As Integer 'local copy
Private mvarFatigue As Byte 'local copy
Private mvarEvade As Byte 'local copy
Private mvarArmor As Byte 'local copy
Private mvarAgility As Byte 'local copy
Private mvarStrength As Byte 'local copy
Private mvarFitness As Byte 'local copy
Private mvarHealth As Byte 'local copy
Private mvarIntelligence As Byte 'local copy
Private mvarPersonality As Byte 'local copy
Private mvarPicture As String 'local copy
Private mvarHitBonus As Integer 'local copy
Private mvarAnimalLore As Byte
Private mvarAppraising As Byte
Private mvarArchery As Byte
Private mvarArmorLore As Byte
Private mvarBandaging As Byte
Private mvarBarter As Byte
Private mvarBegging As Byte
Private mvarBlindFighting As Byte
Private mvarBluntWeaponFighting As Byte
Private mvarBrawling As Byte
Private mvarButchery As Byte
Private mvarCamping As Byte
Private mvarCarpentry As Byte
Private mvarCarving As Byte
Private mvarChopping As Byte
Private mvarCommerce As Byte
Private mvarCooking As Byte
Private mvarFencing As Byte
Private mvarFireBuilding As Byte
Private mvarFishing As Byte
Private mvarForaging As Byte
Private mvarHandToHandCombat As Byte
Private mvarHerbLore As Byte
Private mvarHiking As Byte
Private mvarHunting As Byte
Private mvarItemLore As Byte
Private mvarJogging As Byte
Private mvarKicking As Byte
Private mvarKnifeFighting As Byte
Private mvarMountaineering As Byte
Private mvarParrying As Byte
Private mvarPerformance As Byte
Private mvarPickPocketing As Byte
Private mvarPunching As Byte
Private mvarSkinning As Byte
Private mvarSocial As Byte
Private mvarSwimming As Byte
Private mvarSwordFighting As Byte
Private mvarWeaponLore As Byte
Private mvarWrestling As Byte
Private mvarParry As Boolean 'local copy
Private mvarInventory(100) As Item 'local copy
Private mvarNextIndex As Byte 'local copy
Private mvarBody As Item 'local copy
Private mvarAboutBody As Item 'local copy
Private mvarFeet As Item 'local copy
Private mvarWielded As Item 'local copy
Private mvarShield As Item 'local copy
Private mvarHead As Item 'local copy
Private mvarDamageBonus As Integer 'local copy
Private mvarPoisoned As Boolean 'local copy
Private mvarBandaged As Boolean 'local copy
Private mvarStealth As Byte 'local copy
Private mvarPerformed As Byte 'local copy
Private mvarBegged As Byte 'local copy
Private mvarMax_Hp As Byte 'local copy
Private mvarMax_Fp As Byte 'local copy

Public Property Let Max_Fp(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Max_Fp = 5
       mvarMax_Fp = vData
End Property

Public Property Get Max_Fp() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Max_Fp
       Max_Fp = mvarMax_Fp
End Property

Public Property Let Max_Hp(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Max_Hp = 5
       mvarMax_Hp = vData
End Property

Public Property Get Max_Hp() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Max_Hp
       Max_Hp = mvarMax_Hp
End Property

Public Property Let Begged(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Begged = 5
       mvarBegged = vData
End Property

Public Property Get Begged() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Begged
       Begged = mvarBegged
Public Property Let Performed(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Performed = 5
       mvarPerformed = vData
End Property

Public Property Get Performed() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Performed
       Performed = mvarPerformed
End Property

Public Property Let Stealth(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Stealth = 5
       mvarStealth = vData
End Property

Public Property Get Stealth() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Stealth
       Stealth = mvarStealth
End Property

Public Property Let Bandaged(ByVal vData As Boolean)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Bandaged = 5
       mvarBandaged = vData
End Property

Public Property Get Bandaged() As Boolean
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Bandaged
       Bandaged = mvarBandaged
End Property

Public Property Let Poisoned(ByVal vData As Boolean)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Poisoned = 5
       mvarPoisoned = vData
End Property

Public Property Get Poisoned() As Boolean
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Poisoned

Player - 4

    Poisoned = mvarPoisoned
End Property

Public Property Let Damagebonus(ByVal vData As Integer)
' used when assigning a value to the property, on the left side of an assignment.
' Syntax: X.ToDamage = 5
    mvarToDamage = vData
End Property

Public Property Get Damagebonus() As Integer
' used when retrieving value of a property, on the right side of an assignment.
' Syntax: Debug.Print X.ToDamage
    ToDamage = mvarToDamage
End Property

Public Property Set Head(ByVal vData As Object)
' used when assigning an Object to the property, on the left side of a Set statement.
' Syntax: Set x.Head = Form1
    Set mvarHead = vData
End Property

Public Property Get Head() As Item
' used when retrieving value of a property, on the right side of an assignment.
' Syntax: Debug.Print X.Head
    Set Head = mvarHead
End Property

Public Property Set Shield(ByVal vData As Object)
' used when assigning an Object to the property, on the left side of a Set statement.
' Syntax: Set x.Shield = Form1
    Set mvarShield = vData
End Property

Public Property Get Shield() As Item
' used when retrieving value of a property, on the right side of an assignment.
' Syntax: Debug.Print X.Shield
    Set Shield = mvarShield
End Property

Public Property Set Wielded(ByVal vData As Object)
' used when assigning an Object to the property, on the left side of a Set statement.
' Syntax: Set x.Wielded = Form1
    Set mvarWielded = vData
End Property

Public Property Get Wielded() As Item
' used when retrieving value of a property, on the right side of an assignment.
' Syntax: Debug.Print X.Wielded
Player - 5

Set Wielded = mvarWielded
End Property

Public Property Set Feet(ByVal vData As Object)
'used when assigning an Object to the property, on the left side of a Set statement.
'Syntax: Set x.Feet = Form1
    Set mvarFeet = vData
End Property

Public Property Get Feet() As Item
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Feet
    Set Feet = mvarFeet
End Property

Public Property Set Aboutbody(ByVal vData As Object)
'used when assigning an Object to the property, on the left side of a Set statement.
'Syntax: Set x.AboutBody = Form1
    Set mvarAboutBody = vData
End Property

Public Property Get Aboutbody() As Item
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.AboutBody
    Set Aboutbody = mvarAboutBody
End Property

Public Property Set Body(ByVal vData As Object)
'used when assigning an Object to the property, on the left side of a Set statement.
'Syntax: Set x.Body = Form1
    Set mvarBody = vData
End Property

Public Property Get Body() As Item
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Body
    Set Body = mvarBody
End Property

Public Property Let NextIndex(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.NextIndex = 5
    mvarNextIndex = vData
End Property

Public Property Get NextIndex() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.NextIndex
NextIndex = mvarNextIndex
End Property

Public Property Set Inventory(ByVal vIndex As Byte, ByVal vData As Item)
' used when assigning an object to the property, on the left side of a Set statement.
'Syntax: Set x.Inventory = Form1
    Set mvarInventory(vIndex) = vData
End Property

Public Property Get Inventory(ByVal vIndex As Byte) As Item
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Inventory
    Set Inventory = mvarInventory(vIndex)
End Property

Public Property Let Parry(ByVal vData As Boolean)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Parry = 5
    mvarParry = vData
End Property

Public Property Get Parry() As Boolean
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Parry
    Parry = mvarParry
End Property

Public Property Let Hitbonus(ByVal vData As Integer)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.HitBonus = 5
    mvarHitBonus = vData
End Property

Public Property Get Hitbonus() As Integer
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.HitBonus
    Hitbonus = mvarHitBonus
End Property

Public Property Let Picture(ByVal vData As String)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Picture = 5
    mvarPicture = vData
End Property

Public Property Get Picture() As String
' used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Picture
    Picture = mvarPicture
End Property

Public Property Let Personality(ByVal vData As Byte)
' used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Personality = 5
    mvarPersonality = vData
End Property

Public Property Get Personality() As Byte
Player - 7

'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Personality
    Personality = mvarPersonality

Public Property

Public Property Let Intelligence(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Intelligence = 5
    mvarIntelligence = vData
End Property

Public Property Get Intelligence() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Intelligence
    Intelligence = mvarIntelligence
End Property

Public Property Let Health(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Health = 5
    mvarHealth = vData
End Property

Public Property Get Health() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Health
    Health = mvarHealth
End Property

Public Property Let Fitness(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Fitness = 5
    mvarFitness = vData
End Property

Public Property Get Fitness() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Fitness
    Fitness = mvarFitness
End Property

Public Property Let Strength(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Strength = 5
    mvarStrength = vData
End Property

Public Property Get Strength() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Strength
    Strength = mvarStrength
End Property

Public Property Let Agility(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Agility = 5
    mvarAgility = vData
End Property

Public Property Get Agility() As Byte
'used when retrieving value of a property, on the right side of an assignment.
```
' Syntax: Debug.Print X.Agility
    Agility = mvarAgility
End Property

Public Property Let Armor(ByVal vData As Byte)
    ' used when assigning a value to the property, on the left side of an assignment.
    ' Syntax: X.Armor = 5
    mvarArmor = vData
End Property

Public Property Get Armor() As Byte
    ' used when retrieving value of a property, on the right side of an assignment.
    ' Syntax: Debug.Print X.Armor
    Armor = mvarArmor
End Property

Public Property Let Evade(ByVal vData As Byte)
    ' used when assigning a value to the property, on the left side of an assignment.
    ' Syntax: X.Evade = 5
    mvarEvade = vData
End Property

Public Property Get Evade() As Byte
    ' used when retrieving value of a property, on the right side of an assignment.
    ' Syntax: Debug.Print X.Evade
    Evade = mvarEvade
End Property

Public Property Let Fatigue(ByVal vData As Byte)
    ' used when assigning a value to the property, on the left side of an assignment.
    ' Syntax: X.Fatigue = 5
    mvarFatigue = vData
End Property

Public Property Get Fatigue() As Byte
    ' used when retrieving value of a property, on the right side of an assignment.
    ' Syntax: Debug.Print X.Fatigue
    Fatigue = mvarFatigue
End Property

Public Property Let HitPoints(ByVal vData As Integer)
    ' used when assigning a value to the property, on the left side of an assignment.
    ' Syntax: X.HitPoints = 5
    mvarHitPoints = vData
End Property

Public Property Get HitPoints() As Integer
    ' used when retrieving value of a property, on the right side of an assignment.
    ' Syntax: Debug.Print X.HitPoints
    HitPoints = mvarHitPoints
End Property

Public Property Let Money(ByVal vData As Integer)
    ' used when assigning a value to the property, on the left side of an assignment.
    ' Syntax: X.Money = 5
    mvarMoney = vData
End Property

Public Property Get Money() As Integer
    ' used when retrieving value of a property, on the right side of an assignment.
    ' Syntax: Debug.Print X.Money
```