RPG Program Code

Either replace all code from your existing program with what follows or create a new project RPG2 and copy and paste this code. You will also have to copy the map and imap text files to the debug folder.

**Copy and Paste Everything Below this Point**

Imports System.IO

Public Class Form1

Dim MapWidth As Integer = 14 'Controls the number of tiles in each row of your map - You can change this to match the width of your map files

Dim MapHeight As Integer = 14 'Controls the number of tiles in each column of your map - You can change this to match the height of your map files

Dim TileWidth As Integer = 32 'Width, in Pixels of each tile - You could change this if you wanted smaller or larger sized tiles, but I would make certain it holds the same value as TileHeight

Dim TileHeight As Integer = 32 'Height, in Pixels of each tile - - You could change this if you wanted smaller or larger sized tiles, but I would make certain it holds the same value as TileWidth

Dim NumLevels As Integer = 2 'Number of Levels in your Dungeon - If you increase this, you need to have both a map and imap text file for all levels or program will not be able to function

Dim background(MapWidth, MapHeight) As PictureBox 'Array of pictureboxes used to draw the walls and grass and any other background textures

Dim items(MapWidth, MapHeight) As PictureBox 'Array of pictureboxes, placed on top of background, which show interactive objects like keys, doors, monsters, etc...

Dim CurrentLevel As Integer = 1 'Stores the current level of the Dungeon hero is on - We start at level 1 and it goes up when we go up stairs, it goes down when we go down stairs

Dim map(MapWidth, MapHeight, NumLevels + 1) As String 'All of our map text files get loaded into this 3D array, which stores all background codes in (X, Y, Level) map(3, 5, 2) stores the background texture for (3, 5) for Level 2

Dim imap(MapWidth, MapHeight, NumLevels + 1) As String 'All of our item files get loaded into this 3D array, which stores all item codes in (X, Y, Level) imap(10, 7, 5) stores the item texture for (10, 7) for Level 5

Dim x, y, z As Integer 'Use in for loops to set starting values

Dim hero As PictureBox 'our hero!

Dim HeroX, HeroY As Integer 'Keeps track of Hero's current (X,Y) position on board

Dim NumKeys As Integer = 0 'Number of keys the hero currently holds

Dim HeroDirection As String 'Direction the hero is moving in

Dim MapFileName As String 'Stores the filename of our map text files. Make sure all map files are saved as "mapx.txt", where x is the level

Dim ItemFileName As String 'Stores the filename of our item text files. Make sure all item files are saved as "imapx.txt", where x is the level

Dim Mapfiles(NumLevels + 1) As StreamReader 'We use StreamReader to read our various map text files. We create an array of streamreaders, since each one holds a different map file

Dim ItemFiles(NumLevels + 1) As StreamReader 'We use StreamReader to read our various item text files. We create an array of streamreaders, since each one holds a different item file

Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Me.Size = New Size(465, 482)

Me.BackColor = Color.ForestGreen

LoadMapsAndItems() 'Loads all map files and items files into the map and imap Arrays

LoadHero() 'Creates our hero picturebox and places him at (1,1)

DrawLevel() 'Draws both background and items pictureboxes

End Sub

Private Sub Form1\_KeyDown(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles Me.KeyDown

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Here you can add different image files for when he moves in different directions and add code for taking alternating steps using two image files for each direction

If e.KeyCode = Keys.K Then 'K messageboxes the number of keys you currently have

MsgBox("You have " & NumKeys & " keys.")

ElseIf e.KeyCode = Keys.W And (map(HeroX, HeroY - 1, CurrentLevel) <> "w" And map(HeroX, HeroY - 1, CurrentLevel) <> "f") Then 'Move hero up if the tile above him isn't a wall

HeroY = HeroY - 1

HeroDirection = "up"

ElseIf e.KeyCode = Keys.S And map(HeroX, HeroY + 1, CurrentLevel) <> "w" Then 'Move hero down if the tile above him isn't a wall

HeroY = HeroY + 1

HeroDirection = "down"

ElseIf e.KeyCode = Keys.A And map(HeroX - 1, HeroY, CurrentLevel) <> "w" Then 'Move hero left if the tile to his left isn't a wall

HeroX = HeroX - 1

HeroDirection = "left"

ElseIf e.KeyCode = Keys.D And map(HeroX + 1, HeroY, CurrentLevel) <> "w" Then 'Move hero right if the tile to his right isn't a wall

HeroX = HeroX + 1

HeroDirection = "right"

End If

hero.Location = New Point(HeroX \* TileWidth, HeroY \* TileHeight) 'Moves the hero to his new Location based on his (X,Y) values

CheckItems() 'Check to see if he hit any items (keys, doors, stairs, etc..) 'Check to see if he interacts with an item after everymove

End Sub

Private Sub LoadMapsAndItems() 'Don't Change this

For Me.z = 1 To NumLevels

x = 0

y = 0

MapFileName = "map" & z & ".txt"

Mapfiles(z) = New StreamReader(MapFileName)

Using Mapfiles(z)

Do While Mapfiles(z).Peek() >= 0

For Each m As String In Mapfiles(z).ReadLine

map(x, y, z) = m

x += 1

Next

y += 1

x = 0

Loop

End Using

Mapfiles(z).Close()

Next

For Me.z = 1 To NumLevels

x = 0

y = 0

ItemFileName = "imap" & z & ".txt"

ItemFiles(z) = New StreamReader(ItemFileName)

Using ItemFiles(z)

Do While ItemFiles(z).Peek() >= 0

For Each i As String In ItemFiles(z).ReadLine

imap(x, y, z) = i

x += 1

Next

y += 1

x = 0

Loop

End Using

ItemFiles(z).Close()

Next

End Sub

Private Sub DrawLevel()

'Draw Background

For Me.x = 0 To MapWidth - 1

For Me.y = 0 To MapHeight - 1

background(x, y) = New PictureBox

background(x, y).Size = New Size(TileWidth, TileHeight)

background(x, y).Location = New Point(x \* TileWidth, y \* TileHeight)

background(x, y).Location = New Point(x \* TileWidth, y \* TileHeight)

background(x, y).SizeMode = PictureBoxSizeMode.StretchImage

Controls.Add(background(x, y))

If map(x, y, CurrentLevel) = "g" Then

background(x, y).Image = Image.FromFile("grass.png")

ElseIf map(x, y, CurrentLevel) = "w" Then

background(x, y).Image = Image.FromFile("wall.png")

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Here is where you will add additional background textures to your RPG game \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Add elseIf statements for other background textures and images you want to appear in your maze

'Make sure you use a unique character for each new background texture ("o" for ocean perhaps or "f" for forest)

'Make sure you place the image files in the debug folder

'If the new texture is impassable, like the walls, you will need to add an additional statement to each of your "if" statements in your Form1\_KeyDown

'For example, if I added a forest texture, using the "f" character, my movement if statements would change to look like:

' ElseIf e.KeyCode = Keys.W And (map(HeroX, HeroY - 1, CurrentLevel) <> "w" And map(HeroX, HeroY - 1, CurrentLevel) <> "f") Then

End If

Next

Next

'Draw Items

For Me.x = 0 To MapWidth - 1

For Me.y = 0 To MapHeight - 1

items(x, y) = New PictureBox

items(x, y).Size = New Size(TileWidth, TileHeight)

items(x, y).Location = New Point(x \* TileWidth, y \* TileHeight)

items(x, y).SizeMode = PictureBoxSizeMode.StretchImage

items(x, y).Visible = True

Controls.Add(items(x, y))

items(x, y).BringToFront()

If imap(x, y, CurrentLevel) = "k" Then

items(x, y).Image = Image.FromFile("key.png")

ElseIf imap(x, y, CurrentLevel) = "l" Then

items(x, y).Image = Image.FromFile("door.png")

ElseIf imap(x, y, CurrentLevel) = "u" Then

items(x, y).Image = Image.FromFile("stairsup.png")

ElseIf imap(x, y, CurrentLevel) = "d" Then

items(x, y).Image = Image.FromFile("stairsdown.png")

ElseIf imap(x, y, CurrentLevel) = "n" Then

items(x, y).Visible = False

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Here is where you will add additional items to your RPG game \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Add elseIf statements for other item codes and images you want to appear in your maze

'Make sure you use a unique character for each new item ("m" for monster perhaps or "f" for fire)

'Make sure you place the image files in the debug folder

End If

Next

Next

hero.BringToFront() 'After drawing the background and items, bring the hero to the front, so he is on top of these other pictureboxes

End Sub

Private Sub LoadHero() 'Only change you might make would be to change the image to your own hero

HeroX = 1

HeroY = 1

hero = New PictureBox

hero.Size = New Size(TileWidth, TileHeight)

hero.Location = New Point(HeroX \* TileWidth, HeroY \* TileHeight)

hero.Image = Image.FromFile("herodown1.png") 'You could replace this image with your own hero images

hero.SizeMode = PictureBoxSizeMode.StretchImage

hero.Visible = True

Controls.Add(hero)

hero.BringToFront()

End Sub

Private Sub MoveHeroBack() 'This method is used to move the hero back where he came from, used for example, when hitting a door and having no keys, shouldn't be a reason to change this

If HeroDirection = "up" Then

HeroY = HeroY + 1

ElseIf HeroDirection = "down" Then

HeroY = HeroY - 1

ElseIf HeroDirection = "left" Then

HeroX = HeroX + 1

ElseIf HeroDirection = "right" Then

HeroX = HeroX - 1

End If

hero.Location = New Point(HeroX \* TileWidth, HeroY \* TileHeight)

End Sub

Private Sub CheckItems() 'This subroutine tells the program what to do when the hero hits a specific item.

If imap(HeroX, HeroY, CurrentLevel) = "k" Then 'k means key, so we add 1 to numKeys and change the k to an n so it won't draw it next time the level loads and won't add to numkeys when we bext walk over it, we also make the tile invisible so we don't see it anymore

NumKeys = NumKeys + 1

imap(HeroX, HeroY, CurrentLevel) = "n"

items(HeroX, HeroY).Visible = False

ElseIf imap(HeroX, HeroY, CurrentLevel) = "l" And NumKeys = 0 Then 'If I hit a "l" (Locked door) and I don't have any keys, move me back

MoveHeroBack()

ElseIf imap(HeroX, HeroY, CurrentLevel) = "l" And NumKeys > 0 Then 'if I hit a locked door and I have at least 1 key, subtract 1 key and change the "l" to "n" (nothing) and make it invisible

NumKeys = NumKeys - 1

imap(HeroX, HeroY, CurrentLevel) = "n"

items(HeroX, HeroY).Visible = False

ElseIf imap(HeroX, HeroY, CurrentLevel) = "u" Then 'u means an up stairs, so add one to my Currentlevel, erase the current level and load the new background and items for the level above (CurrentLevel + 1)

CurrentLevel = CurrentLevel + 1

clearBackgroundAndItems()

DrawLevel()

hero.BringToFront()

ElseIf imap(HeroX, HeroY, CurrentLevel) = "d" Then 'd means down starts so subtract one from CurrentLevel, erase the current level and load the new background and items for the level below (CurrentLevel - 1)

CurrentLevel = CurrentLevel - 1

clearBackgroundAndItems()

DrawLevel()

hero.BringToFront()

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Here is where you will add additional items interactions to your RPG game \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Add elseIf statements for other item codes how you want the program to react.

'You could add sword objects that are kept track of it your inventory like keys

'You could add monster objects that can only be killed with swords (each monster killed subracts 1 sword

'You could have different color keys for different color doors

'You could have levers and gates

'This is where your imagine gets to come into play

End If

End Sub

Private Sub clearBackgroundAndItems() ' Don't change anything here, it erases the current background and items so we can load new ones when we go up or down a level

For Me.x = 0 To MapWidth - 1

For Me.y = 0 To MapHeight - 1

items(x, y).Dispose()

items(x, y) = Nothing

background(x, y).Dispose()

background(x, y) = Nothing

Next

Next

End Sub

End Class