C# Tutorial - Create a simple PAC MAN game in Visual Studio

start a new project

Choose windows form application (visual C#)

Name it missPACMAN and Click OK





In the form properties change the following

Slze - 625, 570

Text - Miss PACMAN

Since we are going to use the miss pacman animated gif in the game its only fitting that we name it Miss Pacman.

Notice after doing the above the size of the form and title text will change.

Miss PACHAN	
a restriction	
	1

Rules of the Game -

- 1. Player can move up down left and right
- 2. Player will collect coins collect all coin and you WIN the game
- 3. Player cannot touch Wall or Ghost. If they do GAME OVER
- 4. 2 (Red and Yellow)ghost will have static left to right movement
- 5. 1 (PINK) ghost will have a random movement which will scale across the form.
- 6. When the game is over there will be a game over text or when the player won the game it will show You WIN.

Since we have a player who can move left right up and down, therefore we have created a pac man image which is rotated left right up and down. This is the main reason we picked Miss Pacman because she has a bow and beauty spot which is distinctive enough to recognise which way is which.

Pacman GIF is the sequence of images below



They will all play together, and making it look like the games original player.

Game Assets	Description
¢.	This is the up directional image we named this file up.GIF since they all are animated GIF files. Download the resource from above the tutorial or get it from MOOICT.COM
¢	This is the down directional image named down.GIF
•	This is the right directional image named right.GIF

	This is the left directional image named left.GIF
•	This is the orange ghost
	This is the rotating coin This image will be multiplied and spread across the form. Once the player has collected them all they win the game.
••	This is the crazy pink ghost who will haunt your dreams.
••	This is the red ghost image.

Making The walls

We will be using 5 picture boxes for the walls.



Find the picture box component in the tool box and drag and drop it to the form.

(Name)	pictureBox1
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
Anchor	Top, Left
BackColor	MidnightBlue 🔹
BackgroundImage	(none)

In the properties window of the picture box find the option **BackColor** and change it to **MidNight Blue**.

SIZEMOUE	Normai
Tag	wall
UseWaitCursor	False

There is also a very important option called **TAG**. Enter **wall** for this option. Make sure its lower case.

now copy and paste it 4 times until it looks like below



Now its time to make the ghosts.



Drag and drop another picture box to the screen.

Ŧ	Padding	0, 0, 0, 0
Ð	Size	36, 38
	SizeMode	StretchImage
	Tag	ghost
	UseWaitCursor	False

Change the size to 36, 38, Size mode to stretch image and tag to ghost.

Now we need to import all of the images from our assets folder to the project. In order to do so, click on the small triangle on the top right corner of the small picture box and click on Choose Image.

Make sure the project resource file is selected and click on IMPORT



Once you clicked on import find the folder with all of the images in it.



Select all images and click OPEN



Select the red guy from the image list. and CLICK OK



Make a copy of the red guy, follow the same steps to change it to the yellow guy and place him between the bottom walls.



Now we have two ghosts, time for the crazy pink one. Now you can make the final copy of the ghost and change the image to the pink ghost.



All done placing the ghosts, now its time to make some changes to the

properties of the ghost picture boxes.

Click on the red guy first



Now to create the pac man

Drag and drop a new picture box to the form. Click on it and change the following in the properties window

(Name)

Name it pacman

pacman

Đ	Size	40, 40	
	SizeMode	StretchImage	
	-		size 40,40 and size mode stretch image.

Change the image to right



Good thing we imported all of them AM I RIGHT.

Finally we need to create the coin

Once again drag and drop another picture box to the form.

Đ	Size	20, 20
	SizeMode	StretchImage
	Tag	coin
	UseWaitCursor	False

Change the size to 20,20. Change the size mode to Stretch Image and apply a tag "coin" to this picture box.



Select the coin image and Click OK



now you can copy and paste as many coins as you like. let's do 30.

🔛 Miss PACMAN							_ D ×		
				0					
			9						
								-	
610	_	670	670	CTC	 _				
	•					••			
								This will	d

Now we need to two labels to the form.

۳ ا	imagelist		
Α	Label		
٨	rala ele el		
🔛 Miss F	PACMAN		
label1		label2	

Click on label 1 and check the properties window for the FONT option. Click on the ... three dots to the right.

	i la coloria	o contour o	100
Ð	Font	Microsoft Sans Serif, 8.25pt	

It will open this window and apply these changes to Label 1



BOLD and Size 12 CLICK ok

🔛 Miss PACMAN	_ _ _×
Pabel1	label2

Now click on Label 2 and apply the following changes

Font			×		
Font: Microsoft Sans Serf Minion Pro Hidaul Modern No. 30 Monotype Coraiva	Font style: Bold Regular Oblique Bold Bold Oblique	Size: 20 22 24 26 28 36 48 ▼	OK Cancel		
Effects Strikeout Underline	Sample AaBbYy Script: Western	yZz •			
				BOLD, SIZE	20
🔜 Miss PACMAN					_DX

🔛 Miss PACMAN		
label1	label2	

Now for the final component TIMER

Find the timer component in the tool box

	ani	техирох		
	Ö	Timer		
		T louis		Drag and drop the timer on the form.
Dron	ortion			* 1 ¥
Prop	berues			
tim	ner1 Syst	em.Windows.Forms.	Timer	•
	2↓	9		
E ((Application	nSettings)		
((Name)		timer1	
E	Enabled		True	
(GenerateM	1ember	True	
1	Interval		20	
ľ	Modifiers		Private	
1	Tag			

Click on the Timer and apply the changes, **Enabled TRUE** and **Interval to 20**.

Adding Events to the game

Click on the form and look at the form properties window.

Properties	•	Ŧ	X
Form1 System.Windows.Forms.Form			•

Click on the lightning bolt icon on the form.

KeyDown	keyisdown
KeyPress	
KeyUp	keyisup
Lawout	

Find the key down and key up events and type in the following event names. Each time you type in the name it will take you to the code view, make sure to come back to the design view and type the keyisup event.

We need one more event for our game and its not in the list here.

Double click on the timer icon on bottom of the form.

Double click this

🖄 timer1



We have 3 total events in this now, keyisdown, keyisup and timer1_Tick

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace missPACMAN
{
   public partial class Form1 : Form
    {
        // start the variables
        bool goup;
        bool godown;
        bool goleft;
        bool goright;
        int speed = 5;
        //ghost 1 and 2 variables. These guys are sane well sort of
        int ghost1 = 8;
        int ghost2 = 8;
        //ghost 3 crazy variables
        int ghost3x = 8;
        int ghost3y = 8;
        int score = 0;
```

```
// end of listing variables
public Form1()
{
    InitializeComponent();
    label2.visible = false;
}
private void keyisdown(object sender, KeyEventArgs e)
{
    private void keyisup(object sender, KeyEventArgs e)
{
    private void timer1_Tick(object sender, EventArgs e)
    {
    }
}
```

Add the highlighted code above the Form1() function. These are the global variables we will need for this game.

bool goup is a Boolean. This can either be true or false. There are 4 Booleans all together, we are going to use them to detect whether the player should go in one of 4 directions.

int speed is the speed integer. This variable contains numbers and it only holds the number 5. This will be used as the speed which our player moves across the game.

int ghost1 is the red ghosts speed. int ghost2 is the yellow ghost speed. Since they will only move in one direction we don't two variables for them.

int ghost3x is the X direction or horizontal directional speed for the pink ghost.

int ghost2y is the Y direction or vertical directional speed for the pink ghost.

int score is the integer which will keep track of the score in which the player collections the coins and it will increase by 1. We gave it default value of 0.

label2.visible = false;

This line above is big label we added earlier, this will show when the game is over, either when the player touched the wall or the ghosts or when the player collected all of the coins.

KEYISDOWN function

<pre>private void keyisdown(object sender, KeyEventArgs e)</pre>	In this function we are tracking 4 keys.
{	Up, down, let and right. When either of
<mark>if (e.KeyCode == Keys.Left)</mark>	these keys are pressed by the user we
{	change the directional images of pacman
<mark>goleft = true;</mark>	dynamically so resemble the movement.
<pre>pacman.Image = Properties.Resources.Left;</pre>	
<mark>}</mark>	IF key code is LEFT
	Go left is set to true
<pre>if (e.KeyCode == Keys.Right)</pre>	Change pac man image to LEFT
{	

TE kov ande is DICUT
IF KEY COUE IS RIGHT
Go right is set to true
Change pac man image to RIGHT
IF key code is UP
Go up is set to true
Change pac man image to UP
IF key code is DOWN
Go down is set to true
Change pac man image to down
Make sure you pay attention to the if
statement curly brackets
TE (condition goes here)
(condition goes here)
{ < open curty bracket
Instructions nere
<pre>} < close curly bracket</pre>

KEYISUP function

TIMER1_TICK function

private void timer1_Tick(object sender, EventArgs e)	This is the timer function. This is
	the main function or event that makes
label1 Text = "Scope: " + scope:	
	our game run.
// show the score on the board	
	First we are going to link the score
	integer to the label 1. We are calling
	label 1 text property and adding the
	score variable to it. So when the
	player gets a point it will update it.
<pre>//player movement codes start</pre>	This is where the player is moving.
if (goleft)	The reason we used Booleans is because
{	when the Boolean is true we can move
<pre>pacman.Left -= speed;</pre>	the player and when its false it will
//moving player to the left.	stop.
}	
if (goright)	To go left we are going to set the
	pacmans left property of the picture
pacman.Left += speed:	box minus equals to the speed

//moving player to the right	variable.
<pre> } if (goup) { pacman.Top -= speed; //moving to the top } </pre>	By taking away the 5 each time the timer runs from left of the picture box we are going to dynamically move it towards the left of the screen. Thats why we are using -= sign.
<pre>if (godown) { pacman.Top += speed; //moving down } //player movements code end</pre>	To go to the right we are going to plus equals to speed variable. To go up we are using the pacmans top property and taking away 5 each time it runs. The speed variable contains
<pre>//moving ghosts and bumping with the walls padchast loft is about 1</pre>	<pre>the number 5. To go down we are adding 5 each time it runs. Got the red ghost and yellow ghost we are accidenting a to the their perpective</pre>
<pre>// if the red ghost hits the picture box 4 then we reverse the speed if (redGhost.Bounds.IntersectsWith(pictureBox4.Bounds)) { ghost1 = -ghost1; } // if the red ghost hits the picture box 3 we reverse the speed place if (redGhost Bounds IntersectsWith(nictureBox2 Bounds))</pre>	since we want the ghosts to bump into the wall and go the other way we are going to use the bounds property. With this we can check if the ghost hit the wall and we will change the positive ghost1 variable to negative ghost1 variable so it will go loft instead of
<pre>// if the yellow ghost hits the picture box 1 then we reverse the speed if (yellowGhost.Bounds.IntersectsWith(pictureBox1.Bounds)) { ghost2 = -ghost2;</pre>	We are doing the same for the yellow ghost. Once it hits the wall we will change the positive variable to negative and so on.
<pre>// if the yellow chost hits the picture box 2 then we reverse the speed else if (yellowGhost.Bounds.IntersectsWith(pictureBox2.Bounds)) { ghost2 = -ghost2; } //moving ghosts and bumping with the walls end</pre>	
<pre>//for loop to check walls, ghosts and points foreach (Control x in this.Controls) {</pre>	First we are running a foreach loop. Inside the loop we are giving the condition to loop through all of the controls in the form. The loop always starts with the open
<pre>if (((PictureBox)x).Bounds.IntersectsWith(pacman.Bounds) score == 30) {</pre>	curly brackets { Then we are stating an if statement, in the statement we are looking for x variable which we declared on the loop earlier to see is X is a type of picture box and it has the tag of "wall" OR "ghost". Remember when it was mentioned the picture boxes tags are important, this is why.
<pre>// (x is riccure by the second of the s</pre>	Instead of listing through all of the picture boxes we are going to loop through all of them. Using a loop we can do more with less code.
<pre>this.Controls.Remove(x); //remove that point score++; // add to the score } }</pre>	If we hit any of the wall or ghost then we are re-arranging the pac mans position (left and top) and stopping the timer, then we are showing that GAME OVER text on the label 2.
<pre>// end of for loop checking walls, points and ghosts.</pre>	We are also checking if the player has collected all of the coins and scored equals to 30. To check if the player has hit one of the coins on screen we are doing the

	same. We check if the player bounds intersects with coins then we remove that coin from the display and add one to screen.
<pre>//ghost 3 going crazy here pinkGhost.Left += ghost3x; pinkGhost.Top += ghost3y;</pre>	This is the crazy PINK ghost. The last two ghosts will go horizontally only but this one will bump across the whole screen. When we start we are increasing the LEFT and TOP.
<pre>//ghost 3 bumping against the walls and form borders if (pinkGhost.Left < 1 pinkGhost.Left + pinkGhost.Width > ClientSize.Width - 2 (pinkGhost.Bounds.IntersectsWith(pictureBox4.Bounds)) (pinkGhost.Bounds.IntersectsWith(pictureBox3.Bounds)) (pinkGhost.Bounds.IntersectsWith(pictureBox1.Bounds)) (pinkGhost.Bounds.IntersectsWith(pictureBox2.Bounds))) { ghost3x = -ghost3x; }</pre>	In this LONG if statement we are checking whether the PINK ghost hits edge of screen OR the far end of the screen OR picture box 4 OR picture box 3 OR picture box 1 OR picture box 2 then we change the x direction of the PINK ghost.
<pre>if (pinkGhost.Top < 1 pinkGhost.Top + pinkGhost.Height > ClientSize.Height - 2) { ghost3y = -ghost3y; } // end of the crazy ghost movements }</pre>	In this statement we are checking if the pink ghost hits the top or the bottom of the screen then we are bumping off the screen when it does.



Hiss PACHAN				Hiss PACHAN		LIQ X
Score: 0				Score: 21	GAME OVER	
	•					
		•			🔹 😐 😐	
	•					
*						
				- - - 1	•	
•	🙆 🔹				• • • 😐	
•		0 0				

Its working. Happy Coding -

Full Code the game is below. Check and double check the code while working on it.

Keep learning and Moo On

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace missPACMAN
{
    public partial class Form1 : Form
    {
        // start the variables
        bool goup;
        bool godwn;
```

```
bool goleft;
bool goright;
int speed = 5;
//ghost 1 and 2 variables. These guys are sane well sort of
int ghost1 = 8;
int ghost2 = 8;
//ghost 3 crazy variables
int ghost3x = 8;
int ghost3y = 8;
int score = 0;
// end of listing variables
public Form1()
{
    InitializeComponent();
    label2.Visible = false;
}
private void keyisdown(object sender, KeyEventArgs e)
{
  if (e.KeyCode == Keys.Left)
    {
        goleft = true;
        pacman.Image = Properties.Resources.Left; // change the image to the right
    }
    if (e.KeyCode == Keys.Right)
    {
        goright = true;
        pacman.Image = Properties.Resources.Right; // change the image to the left
    }
    if (e.KeyCode == Keys.Up)
    {
        goup = true;
        pacman.Image = Properties.Resources.Up;
    }
    if (e.KeyCode == Keys.Down)
    {
        godown = true;
        pacman.Image = Properties.Resources.down;
    }
}
private void keyisup(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Left)
    {
        goleft = false;
    }
    if (e.KeyCode == Keys.Right)
    {
        goright = false;
    }
    if (e.KeyCode == Keys.Up)
    {
        goup = false;
    }
```

```
if (e.KeyCode == Keys.Down)
    {
        godown = false;
    }
}
private void timer1_Tick(object sender, EventArgs e)
{
    label1.Text = "Score: " + score; // show the score on the board
    //player movement codes start
    if (goleft)
    {
        pacman.Left -= speed;
        //moving player to the left.
    }
    if (goright)
    {
        pacman.Left += speed;
        //moving player to the right
    }
   if (goup)
    {
        pacman.Top -= speed;
        //moving to the top
    }
    if (godown)
    {
        pacman.Top += speed;
        //moving down
    }
    //player movements code end
    //moving ghosts and bumping witht he walls
    redGhost.Left += ghost1;
    yellowGhost.Left += ghost2;
    // if the red ghost hits the picture box 4 then we reverse the speed
    if (redGhost.Bounds.IntersectsWith(pictureBox4.Bounds))
    {
        ghost1 = -ghost1;
    }
    // if the red ghost hits the picture box 3 we reverse the speed
    else if (redGhost.Bounds.IntersectsWith(pictureBox3.Bounds))
    {
        ghost1 = -ghost1;
    }
    // if the yellow ghost hits the picture box 1 then we reverse the speed
    if (yellowGhost.Bounds.IntersectsWith(pictureBox1.Bounds))
    {
        ghost2 = -ghost2;
    }
    // if the yellow chost hits the picture box 2 then we reverse the speed
    else if (yellowGhost.Bounds.IntersectsWith(pictureBox2.Bounds))
    {
        ghost2 = -ghost2;
    }
    //moving ghosts and bumping with the walls end
    //for loop to check walls, ghosts and points
    foreach (Control x in this.Controls)
    {
        if (x is PictureBox && x.Tag =="wall" || x.Tag =="ghost")
        {
            // checking if the player hits the wall or the ghost, then game is over
```

```
if (((PictureBox)x).Bounds.IntersectsWith(pacman.Bounds) || score == 30)
        {
            pacman.Left = 0;
            pacman.Top = 25;
            label2.Text = "GAME OVER";
            label2.Visible = true;
            timer1.Stop();
        }
    }
    if (x is PictureBox && x.Tag == "coin")
    {
        //checking if the player hits the points picturebox then we can add to the score
        if (((PictureBox)x).Bounds.IntersectsWith(pacman.Bounds))
        {
            this.Controls.Remove(x); //remove that point
            score++; // add to the score
        }
    }
}
// end of for loop checking walls, points and ghosts.
//ghost 3 going crazy here
pinkGhost.Left += ghost3x;
pinkGhost.Top += ghost3y;
if (pinkGhost.Left < 1 ||</pre>
    pinkGhost.Left + pinkGhost.Width > ClientSize.Width - 2 ||
    (pinkGhost.Bounds.IntersectsWith(pictureBox4.Bounds)) ||
    (pinkGhost.Bounds.IntersectsWith(pictureBox3.Bounds)) ||
    (pinkGhost.Bounds.IntersectsWith(pictureBox1.Bounds)) ||
    (pinkGhost.Bounds.IntersectsWith(pictureBox2.Bounds))
    )
{
    ghost3x = -ghost3x;
}
if (pinkGhost.Top < 1 || pinkGhost.Top + pinkGhost.Height > ClientSize.Height - 2)
{
    ghost3y = -ghost3y;
}
// end of the crazy ghost movements
```

}

}

}