Pseudocode

Public partial class Form 1

Initialize variables

* Button Play;
* Button Instruction;
* ComboBox Levels;
* Button Back;
* 8 IMAGES for MARIO
* 2 IMAGES for GUMBA
* 2 IMAGES for Bowser
* Player rectangle
* BowserPlatfrom rectangle;
* Bowser Rectangle;
* BFireBalls Rectangle;
* ARRAY OF Rectangle for GummbaPlatfrom[] of 3 platform   
  ARRAY OF Rectangle for Enemies [] of 4 gumbass
* Timer MarioTim;
* Timer Fireball
* Timer for BowserPlatform
* Bool MFacingRight;
* Bool MJumpingRight;
* Bool Jump;
* Int force, gravity;
* Int ylimit;
* int runningpic= 0;
* int fireballshot = 0;

Private void Form Load

Add a play button

Add an instruction button

Add a ComboBox level;

Change the background

MarioTIm = new timer

BowserPlatform = new timer

Fireball = new timer

Fireball.Interval = 2000;

BowerPlatform = 100

MarioTim.Interval = 10;

Key down Method

Key Up Method

Paint Method

Have a For Loop to assign and initiallze to every array of rectangle so that they are not null

Start all the timer;

This.Doublebuffered.True

ylimit = this.Height - 40;

Jump = False

Combobox Method;

**Paint Method**

**If (Fireball %== 0)**

**Change the bowser image to an opened mouth**

**If (Fireball % ==2)**

**Change the bowser image to a closed mouth**

**If (runningpic%2 = 0)**

**Change the gumba rectangle to a picture of gumbaa**

**If (runningpic%2 = 1)**

**Change the gumba rectangle to a picture of a different gumba**

**if (dx equals 2 )**

**if (runningpic%2 = 0)**

**change the rectangle picture to MarioMiddlePicture**

**if (running&2 = 1)**

**change the rectangle picture to MairoRunningPicture**

**if (dx equals -2)**

**if (runningpic%2 = 0)**

**change the rectangle picture to MarioMiddlePicture**

**if (running&2 = 1)**

**change the rectangle picture to MairoRunningPicture**

**if (MfacingRight == false)**

**change the rectangle picture to MarioStandingLeft**

**if (MfacingRight == true)**

**change the rectangle picture to MarioStandingRight**

**if (MjumingRight == false)**

**change the rectangle picture to MarioJumpingLeft**

**if(MjumpingRight ==True)**

**change the rectangle picture to MarioJumpingRight**

**Key Up Method**

if (e.KeyCode = Key.Left)

set dx to 0

set MFacingRight = false

if(e.KeyCode = Key.Right)

set dx to 0

set MFacingRight = true

Key Down Method

if (Left && Up are pressed)

set MjumpingRight = false;

if (Right && Up are pressed)

set Mjumping Right = true

if (Right is pressed)

set the dx value to 2

if (Left is pressed)

set the dx value to -2

if (up is pressed)

if (Jump is not true)

jump is true

force is 14

gravity is 3

IN Bowser Platform timer

DO THIS BEFORE THE FIREBALL

Have a rectangle Platform going up and down on the left side of the form

Have another rectangle that has the picture of Bowser moving up and down at the same speed of the rectangle platform.

Set the x location of Fireball as the x location of the bowser and set it = to 0;

Have the Fireball.Y moving with the bowser.Y +10

runningpic ++

In Fireball Timer

Fireballshot++;

IN MarioTim Method

Player.Invalidate;

Player.X = Player.X + dx

If (jumping)

if (player's buttom is less than or equal to the ylimit)

Jumping = true

player.Y += force

player.Y -= gravity

if(player's buttom is >= y.Limit)

jumping = false;

player.Y = yLimit - player.Height;

force and gravity is equal to 0;

BUNCH OF IF STATEMENT THAT PRVENTS PLAYER FROM MOVING OUT OF THE SCREEN.

IF (Fireball.X >= 0)

if (FireBall% 2 == 0)

this.Invalidate

FireBallrecatalgeX = FirBallRectanlge.X -10

FireBallrectanlgeY = FieBallRectanlgeY

if (FireBall.X < 0)

FireBall.X = BowserRectanlge.X

FireBall.Y = BowserRectanlge.Y +10;

When Play button is clicked:

Hide the play button

Hide the instruction button

Hide the combobox button

Change the background to the game background.

Start all the timer.

player = new rectangle (20, ylimit, 25, 25)

INTIALIZE ALL the rectangles and all the array of rectangles.

Platfrom[0].location = new point (set the appropriate points) and do this for all 3 platform

Platfrom[1].location = new point (set the appropriate points)…

Platfrom[2].location = new point (set the appropriate points)…

For loop ( runs when I = 0, I < Platfrom.Lenght, I++)

Gumbaa[i].Location = new point ( Platfrom[i].X, Platfrom[i].Y + 5)

When Instruction button is clicked

Hide play button

Hide instruction button

Hide ComboBox level button.

Change the background.

Add back button on a certain location of the from.

When Back button is clicked

Change the background

Remove the back button.

Add a play button

Add an instruction button.

Add a ComboBox level button.

Meathod that checks for valid Name.

-Need a textbox in the main screen

-Check for input

-When valid place it name besides his life cound

Sound everytime Mario runs/ Jumps.

* When up key is clicked then play the sound

Track his health and deaths.

-When he intersects with any blocks then life goes down by one.

-When the value reaches to 0 lives

-Display a message that you lose

-Close the Application

When Mario intercects reachs the key on the top.

Bowser disappears and the fireball timer stops

And the rectangle of fireball becomes null

When Mario intercects with princess peach

Show a message that Mario has won the game Application.Exit

When Mario intercepts with the left or the right side of the gumba

Mario gets respawned and his life decreases by one

If Mario interests with the top part of the Gummba then set the gumba rectangle to null and change the picture to nothing