using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Flappy\_Bird\_Windows\_Form

**{**

public partial class Form1 **:** Form

**{**

// coded for MOO ICT Flappy Bird Tutorial

// Variables start here

int pipeSpeed = 8; // default pipe speed defined with an integer

int gravity = 15; // default gravity speed defined with an integer

int score = 0; // default score integer set to 0

// variable ends

public Form1**()**

**{**

InitializeComponent**()**;

**}**

private void gamekeyisdown**(**object sender, KeyEventArgs e**)**

**{**

// this is the game key is down event thats linked to the main form

if **(**e.KeyCode == Keys.Space**)**

**{**

// if the space key is pressed then the gravity will be set to -15

gravity = -15;

**}**

**}**

private void gamekeyisup**(**object sender, KeyEventArgs e**)**

**{**

// this is the game key is up event thats linked to the main form

if **(**e.KeyCode == Keys.Space**)**

**{**

// if the space key is released then gravity is set back to 15

gravity = 15;

**}**

**}**

private void endGame**()**

**{**

// this is the game end function, this function will when the bird touches the ground or the pipes

gameTimer.Stop**()**; // stop the main timer

scoreText.Text += " Game over!!!"; // show the game over text on the score text, += is used to add the new string of text next to the score instead of overriding it

**}**

private void gameTimerEvent**(**object sender, EventArgs e**)**

**{**

flappyBird.Top += gravity; // link the flappy bird picture box to the gravity, += means it will add the speed of gravity to the picture boxes top location so it will move down

pipeBottom.Left -= pipeSpeed; // link the bottom pipes left position to the pipe speed integer, it will reduce the pipe speed value from the left position of the pipe picture box so it will move left with each tick

pipeTop.Left -= pipeSpeed; // the same is happening with the top pipe, reduce the value of pipe speed integer from the left position of the pipe using the -= sign

scoreText.Text = "Score: " + score; // show the current score on the score text label

// below we are checking if any of the pipes have left the screen

if**(**pipeBottom.Left **<** -150**)**

**{**

// if the bottom pipes location is -150 then we will reset it back to 800 and add 1 to the score

pipeBottom.Left = 800;

score++;

**}**

if**(**pipeTop.Left **<** -180**)**

**{**

// if the top pipe location is -180 then we will reset the pipe back to the 950 and add 1 to the score

pipeTop.Left = 950;

score++;

**}**

// the if statement below is checking if the pipe hit the ground, pipes or if the player has left the screen from the top

// the two pipe symbols stand for OR inside of an if statement so we can have multiple conditions inside of this if statement because its all going to do the same thing

if **(**flappyBird.Bounds.IntersectsWith**(**pipeBottom.Bounds**)** **||**

flappyBird.Bounds.IntersectsWith**(**pipeTop.Bounds**)** **||**

flappyBird.Bounds.IntersectsWith**(**ground.Bounds**)** **||** flappyBird.Top **<** -25

**)**

**{**

// if any of the conditions are met from above then we will run the end game function

endGame**()**;

**}**

// if score is greater then 5 then we will increase the pipe speed to 15

if**(**score **>** 5**)**

**{**

pipeSpeed = 15;

**}**

**}**

**}**

**}**