Course: EGDF20
Module: EGE202 Application Programming

Practical 4: Race Cart Application: Developing an Interactive and Animated Application

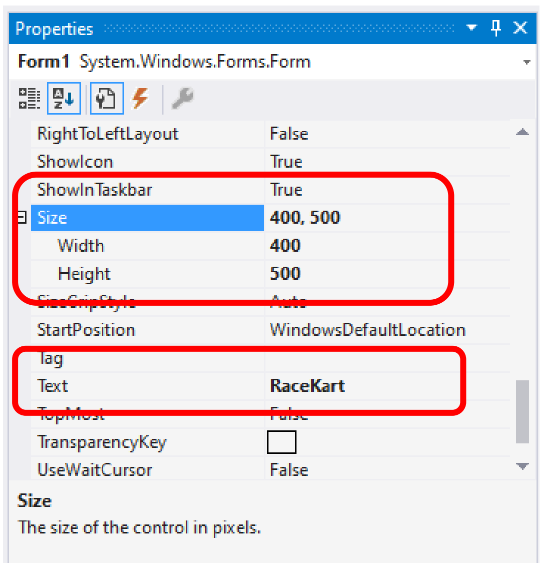
Objectives: At the end of this lab, the student should be able to learn the techniques for creating animation and implement more advanced interaction in a GUI software application.

<u>Exercise 1 – Building a Race Cart Application</u>

Part 1: Creating GUI with Animation (Game Update)

1. Under the ***File*** menu, click ***New Project*** or use the ***New Project*** button to create a new project. Alternatively, use the ***Create New Project*** link in the ***Get Started*** *popup* dialog.
2. From the pop-up dialog, select **"C#"** for the ***Language filter***, **"Windows"** for ***the Platform filter*** and **"Desktop"** for the ***Project type filter***.
3. Then choose ***Windows Forms App (.Net Framework)*** and click the ***Next*** button.
4. Type the name of your new project as ***RaceKart*** and keep the Solution name the same as Project name.
5. **Do not** tick on the check-box of [☐ **Place solution and project in the same directory**].
6. Click the **Create** button to start your project.

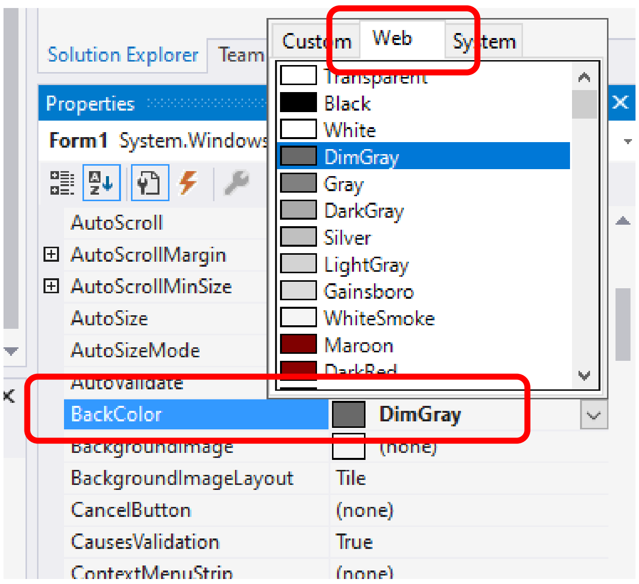
7. In the ***Properties*** window of the ***Form*** control, change the ***TopMost*** property of the ***Form1*** to 'True'.
8. Double click on "*Form1.cs*" ***Solution Explorer*** window to launch the ***Form Designer*** tab.
9. Change the ***Text*** property of the ***Form*** from '*Form1*' to '*RaceKart*' , ***Size (Width & Height)*** to 400 by 500 and **BackColor** to '*DimGray*' (Web).



Properties
Form1 System.Windows.Forms.Form

RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	400, 500
Width	400
Height	500
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	RaceKart
TopMost	False
TransparencyKey	☐
UseWaitCursor	False

Size
The size of the control in pixels.



Solution Explorer | Team
Properties
Form1 System.Windows

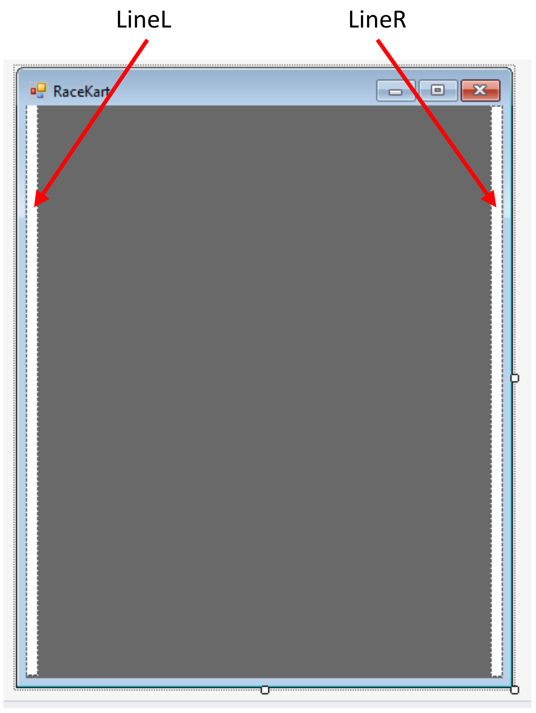
Custom | Web | System

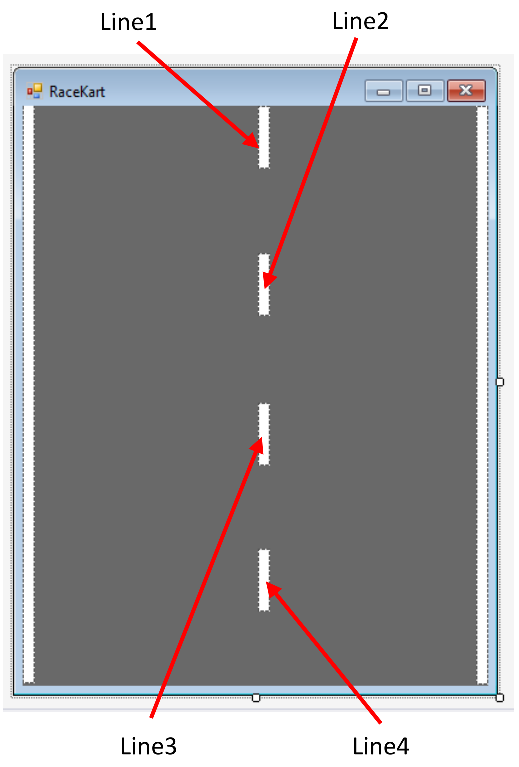
Transparent, Black, White, DimGray, Gray, DarkGray, Silver, LightGray, Gainsboro, WhiteSmoke, Maroon, DarkRed

AutoScroll	
AutoScrollMargin	
AutoScrollMinSize	
AutoSize	
AutoSizeMode	
AutoValidate	
BackColor	DimGray
BackgroundImage	(none)
BackgroundImageLayout	Tile
CancelButton	(none)
CausesValidation	True
ContextMenuStrip	(none)

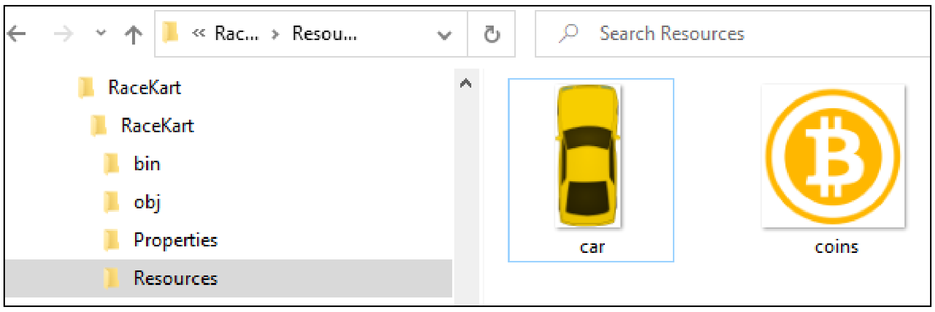
10. From the ***Toolbox***, drag in 2 *PictureBox* controls and and named them '*lineL*' and *lineR*' respectively. Modify the properties with ***Size (Width & Height)*** to 10 by 460 and **BackColor** to '*White*' (Web).

11. Then arrange '*lineL*' and *lineR*' to the each of sides as shown in figure below. **(Note: You can use arrow keys to fine tune the location of the controls.)**



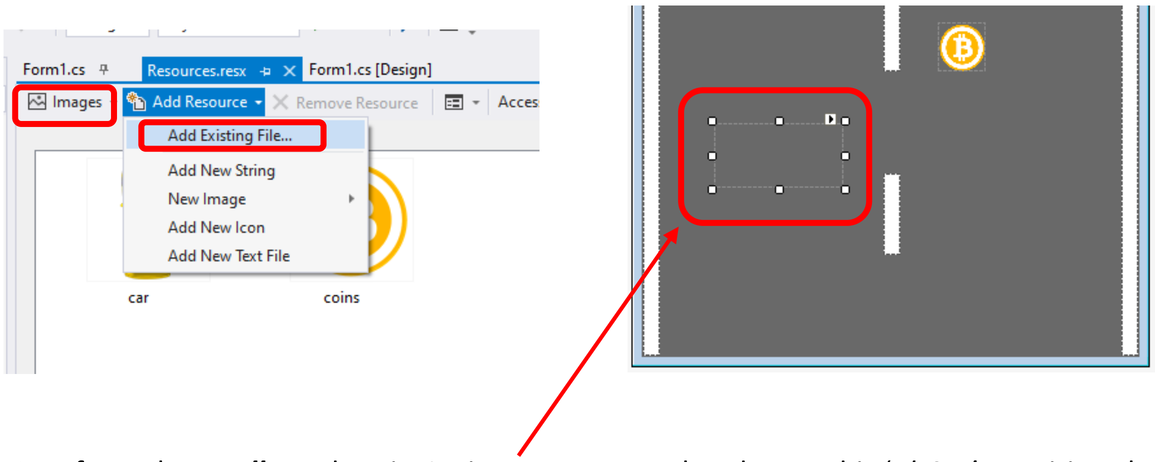


12. From the ***Toolbox***, drag in 1 *PictureBox* control and and named it '*line1*'. Modify the properties with ***Size (Width & Height)*** to 10 by 50 and **BackColor** to '*White*' (Web). Then duplicate it (copy and paste) 3 times and named them '*line2*', '*line3*' and '*line4*'. Arrange them based on figure above.
13. Download both "car.jpg" and "coins.jpg" files to your PC hard drive and place them in the **<u>Resource</u>** sub-folder of your RaceKart project folder.

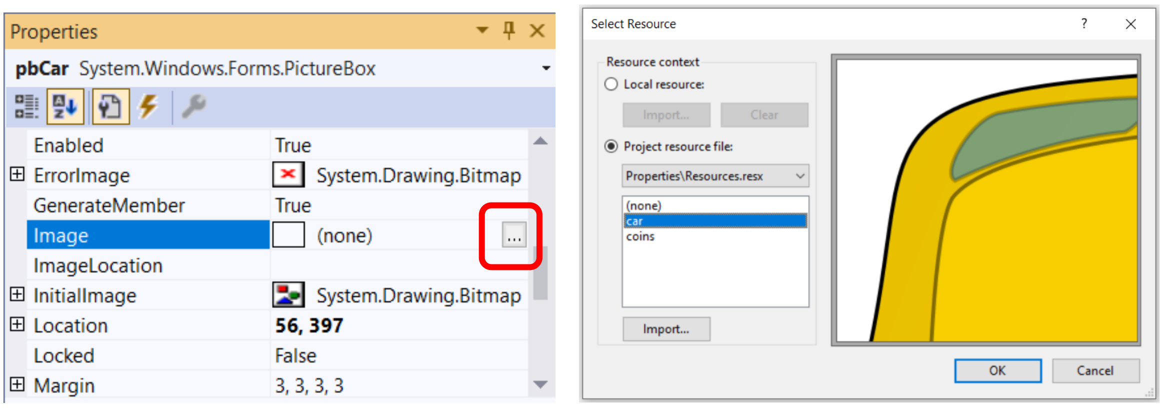


14. From the ***Solution Explorer*** window, expand the ***Properties*** node and double click on ***Resources.resx*** to open the ***Resources'*** tab.

15. From the ***Resources.resx*** tab select ***Images*** and ***Add Resource->Add Existing File …*** to add both the files as image resources that you can use in the application.



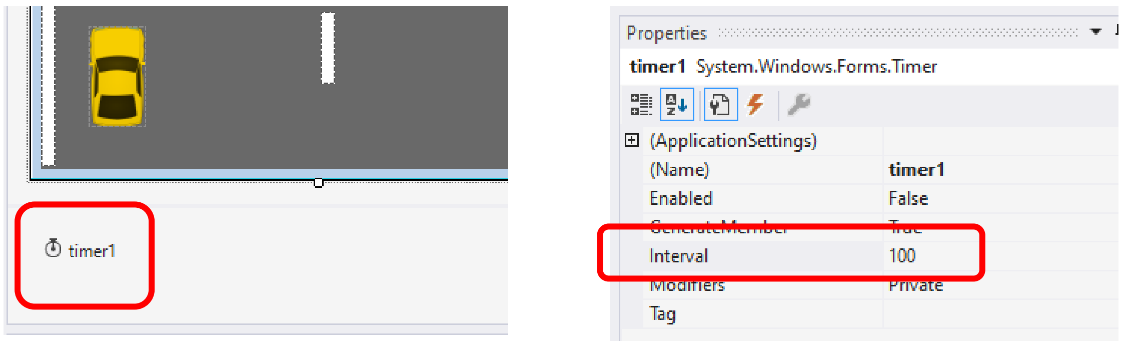
16. Next from the ***Toolbox***, drag in 1 *PictureBox* control and named it '*pbCar*'. Position the car *PictureBox* to the bottom left of the *Form* area.
17. On the **Image** property of **pbCar**, click […] and call out the **Resource panel**. Select the car image and click OK button.



Properties
pbCar System.Windows.Forms.PictureBox

Enabled	True
ErrorImage	System.Drawing.Bitmap
GenerateMember	True
Image	(none)
ImageLocation	
InitialImage	System.Drawing.Bitmap
Location	56, 397
Locked	False
Margin	3, 3, 3, 3

18. Modify the properties with ***Size (Width & Height)*** to 40 by 70, ***SizeMode*** to '*StretchImage*' and **Image** to '*car*' (Project Resource File).
19. Next, we are ready to create simple animation of a moving car (actually is the track that moves rather than the car). To do that we need a timer to periodically move objects.
20. From the ***Toolbox***, drag a *Timer* object into the Form **(must drop within *Form* area)** and the *Timer* object with default name *timer1* will be appended below the *Form.*



Properties
timer1 System.Windows.Forms.Timer

(ApplicationSettings)	
(Name)	timer1
Enabled	False
GenerateMember	True
Interval	100
Modifiers	Private
Tag	

21. Select the *timer1* object and verify at the ***Propertie**s* window that the ***Interval*** is 100. This controls the frequency of animation being updated every 100ms.

22. Double click on the timer1 object to create a ***timer1_Tick()*** event handler that will be executed every tick (100ms interval). Add the following codes:

```
private void timer1_Tick(object sender, EventArgs e)
{
    moveTrack();
}

private void moveTrack()
{
    line1.Top += gameSpeed;
    if (line1.Top > trackH)
        line1.Top = 0;

    line2.Top += gameSpeed;
    if (line2.Top > trackH)
        line2.Top = 0;

    line3.Top += gameSpeed;
    if (line3.Top > trackH)
        line3.Top = 0;

    line4.Top += gameSpeed;
    if (line4.Top > trackH)
        line4.Top = 0;
}
```

23. Next add the following variables declaration for *gameSpeed* and *trackH*. *gameSpeed* controls how fast the car(track) moves while *trackH* contains the height of the *Form* client area to control when will the white divider line moves from bottom to top again.

```
public partial class Form1 : Form
{
    int gameSpeed = 5;
    int trackW = 0, trackH = 0;

    public Form1()
    {
        InitializeComponent();
    }
```

24. At this build and verify that there's no errors. You will notice that with the timer still nothing moves. What's missing?

Timer need to be enabled! When should we enable the timer?

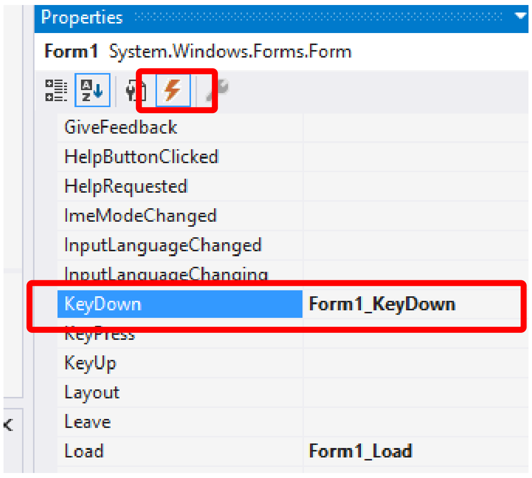
25. At the ***Form Designer***, double click on the *Form Titlebar* to create the ***Form_Load*** event. Add the following codes to the event handler.

```
private void Form1_Load(object sender, EventArgs e)
{
    trackH = this.ClientRectangle.Height;
    trackW = this.ClientRectangle.Width;
    timer1.Start();
}
```

26. Build and test the application again. You should see the tracks are moving.

Part 2: Implement Keyboard Control (Status Update)

1. Now we are going to implement keyboard control to move the player (car).
2. At the ***Form Designer*** tab, select the *Form* **object** by selecting the *Form* **Titlebar**. Then at the ***Properties*** panel event's tab to double click on the dropdown list next to ***KeyDown.***


Properties
Form1 System.Windows.Forms.Form

GiveFeedback	
HelpButtonClicked	
HelpRequested	
ImeModeChanged	
InputLanguageChanged	
InputLanguageChanging	
KeyDown	Form1_KeyDown
KeyPress	
KeyUp	
Layout	
Leave	
Load	Form1_Load

3. The action in the previous step will automatically create a ***KeyDown*** event handler. Add the following codes into the ***KeyDown*** event handler

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Left)
    {
        pbCar.Left -= 4;
        if (pbCar.Left < 0)
            pbCar.Left = 0;
    }
    else if (e.KeyCode == Keys.Right)
    {
        pbCar.Left += 4;
        if (pbCar.Right > trackW)
            pbCar.Left = trackW - pbCar.Width;
    }
}
```

4. Build and test the application. You can manually change the *gameSpeed* value in the codes to try out different speed.
5. Now that we can move the player (car), let's add some coins that the player can collect.
6. From the ***Toolbox***, drag in 2 *PictureBox* controls and named it *'pbCoin1'* and *'pbCoin2'*. Modify the properties with ***Size (Width & Height)*** to 30 by 30, ***SizeMode*** to *'StretchImage'* and **Image** to *'coins'* (Project Resource File).

7. Position the coins *PictureBox* at your choice of location within the *Form* area.
8. Next add the following codes to move the coins (similar to how we move the white divider lines)

```
private void moveCoins()
{
    pbCoin1.Top += gameSpeed;
    if (pbCoin1.Top > trackH)
    {
       pbCoin1.Location = new Point(r.Next(0, trackW - pbCoin1.Width), 0);
    }

    pbCoin2.Top += gameSpeed;
    if (pbCoin2.Top > trackH)
    {
        pbCoin2.Location = new Point(r.Next(0, trackW - pbCoin2.Width), 0);
    }
}
```

9. Noticed that in the above codes we have an undeclared variable *r* which is a random object variable so that the coins starting location will appear randomly in the game.
10. Let's declare the variable *r* in the ***Load*** event handler

```
public partial class Form1 : Form
{
    int gameSpeed = 5;
    int trackW = 0, trackH = 0;
    Random r;

    public Form1()
```

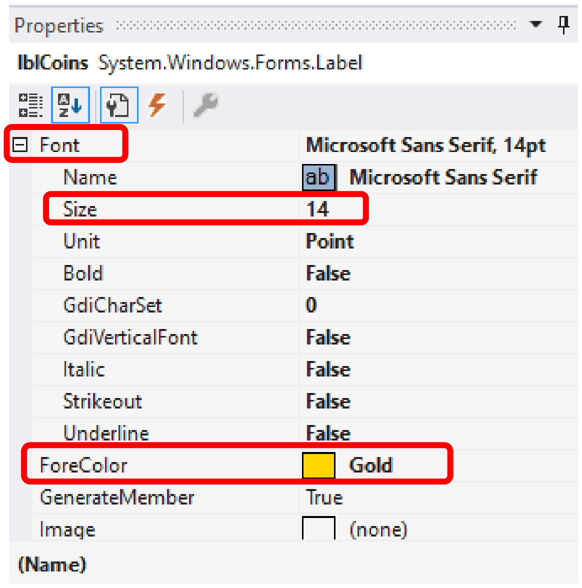
11. Next modify the ***Load*** event handler to instantiate a *Random* object and ***Tick*** event handler to call the function to move the coins

```
private void Form1_Load(object sender, EventArgs e)
{
    trackH = this.ClientRectangle.Height;
   trackW = this.ClientRectangle.Width;
    timer1.Start();

    r = new Random();
}

private void timer1_Tick(object sender, EventArgs e)
{
    moveTrack();
    moveCoins();
}
```

12. Build and test the application. Verify that the coins will reappear randomly at the top after it disappear at the bottom.

Part 3: Implement Game Status (Status Update)

1. At the ***Form Designer*** tab, drag in a *Label* control and placed it at the top left corner of the *Form* area.
2. Modify the ***Name*** property to *'lblCoins'*, ***Text*** to *'Coins:'* ***ForeColor*** to *'Gold' (Web)* and ***Font->Size*** to *'14'.*



```
Properties
lblCoins  System.Windows.Forms.Label
⊟ Font              Microsoft Sans Serif, 14pt
    Name            ab Microsoft Sans Serif
    Size            14
    Unit            Point
    Bold            False
    GdiCharSet      0
    GdiVerticalFont False
    Italic          False
    Strikeout       False
    Underline       False
  ForeColor         Gold
  GenerateMember    True
  Image             (none)
(Name)
```

3. Next, we will declare a new variable called ***myCoins*** to keep track of total collected coins.

```
public partial class Form1 : Form
{
    int myCoins = 0;
    int gameSpeed = 5;
    int trackW = 0, trackH = 0;
    Random r;
```

4. Next, we are going to add a function to collect the coins by checking if the car *PictureBox* overlaps with the coins *PictureBox*.

```
private void coinsCollect()
{
    if (pbCar.Bounds.IntersectsWith(pbCoin1.Bounds))
    {
        pbCoin1.Location = new Point(r.Next(0, trackW - pbCoin1.Width), 0);
        myCoins++;
    }
    if (pbCar.Bounds.IntersectsWith(pbCoin2.Bounds))
    {
        pbCoin2.Location = new Point(r.Next(0, trackW - pbCoin2.Width), 0);
        myCoins++;
    }

    lblCoins.Text = "Coins: " + myCoins;
}
```

5. Lastly, we are going to check for overlapping between cars and coins by calling the function ***coinsCollect*** during the ***Tick*** event.

```
private void timer1_Tick(object sender, EventArgs e)
{
    moveTrack();
    moveCoins();
    coinsCollect();
}
```

6. Build and test the application.

7. Proceed to the next page of additional challenges. You will need to modify the codes and formulars slightly to implement the outcomes.

Part 4: Additional Challenge (Changing moving speed)

1. Modify the appropriate code for the moving speed of car.
2. Use keyboard **PageUp** key to **increase** the speed of the top speed of gameSpeed = 50.
3. Use keyboard **PageDown** key to **decrease** the speed and stop at gameSpeed = 0.

Part 5: Additional Challenge (Driving the car around the race road boundary)

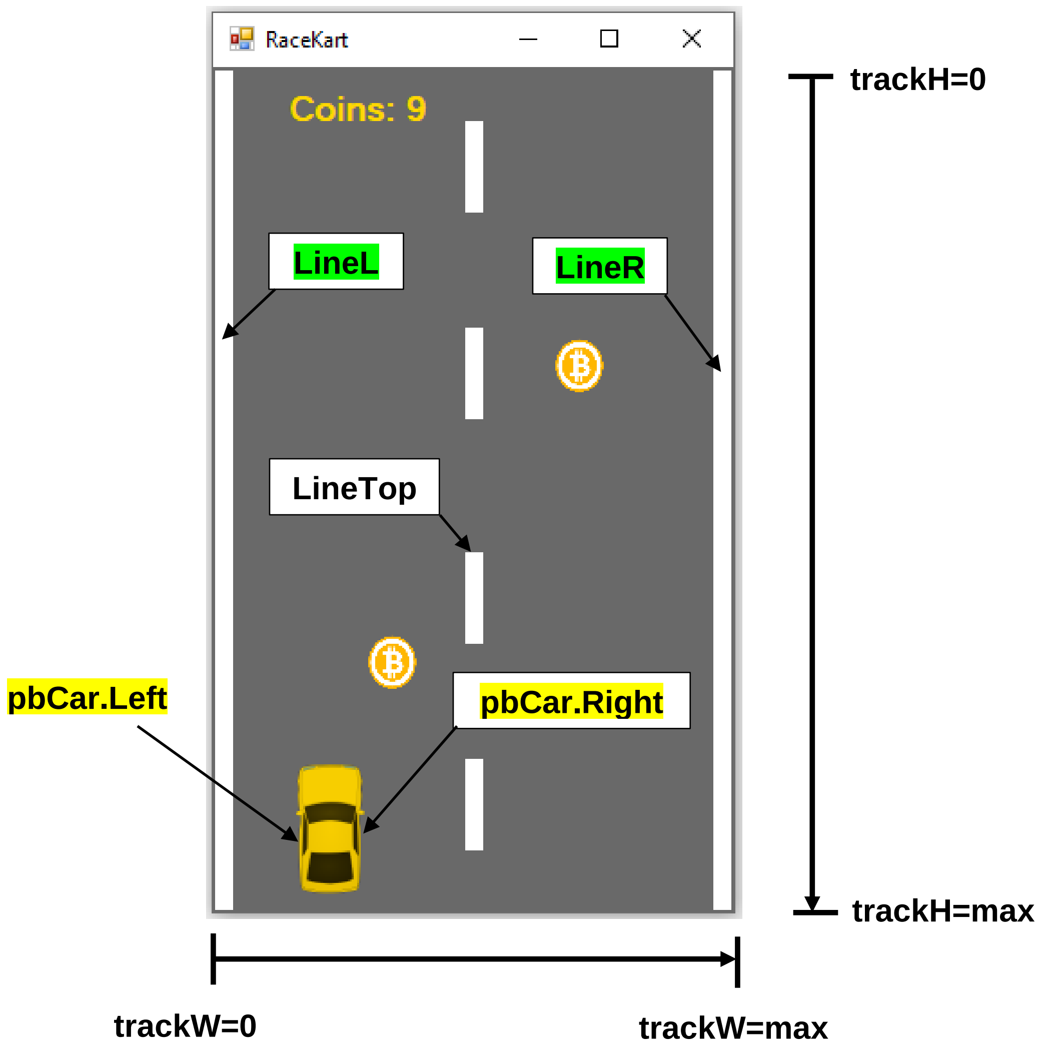
1. Add 2 more key controls to drive the car up and down the racetrack.
2. Use keyboard **Up** arrow key to move the car upwards the racetrack.
3. Use keyboard **Down** arrow key to move the car downwards the racetrack.

Part 6: Additional Challenge (Avoid car driving on Left and Right vertical White Line)

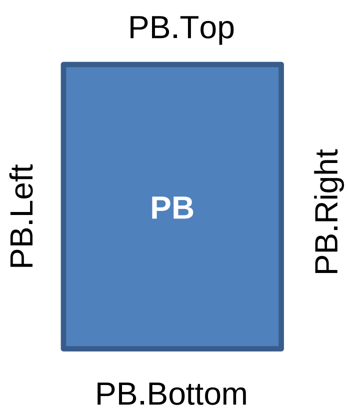
1. Based on the information provide in Figure below,
2. Modify the appropriate code to prevent driving the car on the Left and Right white line (**LineL** and **LineR** respectively).
3. The car must not drive out of the Top side and Bottom side of the visible racetrack as well.

Part 7: Additional Challenge (Design Game Goal of 10 coins collection)

1. Set the game goal to be collection of 10 coins.
2. Once the goal has been achieved, stop the racetrack.
3. Make announcement to the player “Congratulation! You have achieved your Goal”.



Note: About PictureBox (**PB**) Bounds


PB.Top
PB.Left
PB
PB.Right
PB.Bottom