OpenTK Documentation:

**Use OpenGL API:**OpenTK wraps OpenGL in static class, so to use OpenGl api like “glViewport”, do GL.Viewport instead.

**Overrides**

**Open Game Window:**class can inherit from GameWindow, then pass options through that class

class MainWindow : GameWindow {

new MainWindow (//custom assign value params )   
 : base ( //assign initial optional window values)   
{ //assign params  
 Title += “ GL VERSION : ” …  
}

}

In game loop, call run to open window:

void Main(){  
 new MainWindow(//pass in custom params).Run(60.0, 0.0);  
 //params in Run() ensures that the game logic update runs at the same speed if machine   
 //performance varies   
}

**Resize Game Window:**when GameWindowFlag.Default = user can resize game window. To reset viewport when window is resized.

override void OnResize(EventArgs e){  
 GL.Viewport(0, 0, Width, Height); //x and y to 0, w and h of viewport to window dimens  
 //(0, 0) lower left corner of screen  
}

OnResize gets called whenever window dimension changes

**Initialize:**when the game window opens, the method OnLoad gets called. Use it to initialize

override void OnLoad(){  
 //do stuff  
}

**Updates**OnUpdateFrame updates every frame, so update the world in here

Override void OnUpdateFrame(FrameEventArgs e){  
 //update stuff  
}

**Rendering**also gets called every frame like OnUpdateFrame. Put all the rendering in here

override void OnRenderFrame(FrameEventArgs e) {  
 //render stuff  
 e.Time //from FrameEvebtArgs gives the elapsed time of the window  
  
 SwapBuffer(); //shows the rendered scene to user on screen  
}

**Keyboard Input:**using OpenTK.Input  
get state of keyboard from Keyboard class like so

var keyState = Keyboard.GetState();

which then use it to check whether is pressed or not like so,

if ( keyState.isKeyDown(Key.//any key in the enum)){  
 //do stuff  
}