



# Sectrics - Civil Engineering Software –Part B – HSC MAJOR PROJECT

By Shaan Khan



# Contents

System Documentation.....	6
Purpose.....	6
System Overview.....	6
Capabilities .....	6
Design Constraints.....	7
Requirements.....	7
Software Requirements .....	8
Hardware Requirements.....	8
Recommended Specifications .....	9
Operational Requirements.....	10
Final Algorithm Design .....	11
Detailed Design.....	11
Project: SectricsV2.....	11
Code File: aboutMenus.cs.....	11
Namespace: Sectrics_V2 .....	11
Class: aboutMenus .....	11
Code File: aboutMenus.Designer.cs.....	22
Namespace: Sectrics_V2 .....	22
Class: aboutMenus .....	22
Code File: bridgeData.cs .....	27
Namespace: Sectrics_V2 .....	27
Class: Nodes.....	27
Class: connectedMembers.....	27
Class: forces .....	28
Class: degreesOfFreedom .....	28
Class: BridgeData .....	28
Code File: buildBridgeMenu.cs .....	29
Namespace: Sectrics_V2 .....	29
Class: buildBridgeMenu.....	29
Code File: buildBridgeMenu.Designer.cs .....	58
Namespace: Sectrics_V2 .....	58
Class: buildBridgeMenu.....	58

Code File: confirmNewBridgeMenu.cs .....	.68
Namespace: Sectrics_V2 .....	.68
Class: confirmNewBridgeMenu.....	.68
Code File: confirmNewBridgeMenu.Designer.cs.....	.81
Namespace: Sectrics_V2 .....	.81
Class: confirmNewBridgeMenu.....	.81
Code File: exitMenus.cs.....	.85
Namespace: Sectrics_V2 .....	.85
Class: exitMenus .....	.85
Code File: exitMenus.Designer.cs.....	.97
Namespace: Sectrics_V2 .....	.97
Class: exitMenus .....	.97
Code File: generalFunctions.cs.....	.101
Namespace: Sectrics_V2 .....	.101
Class: generalFunctions .....	.101
Code File: loadsMenus.cs.....	.109
Namespace: Sectrics_V2 .....	.109
Class: loadsMenus .....	.109
Code File: loadsMenus.Designer.cs.....	.136
Namespace: Sectrics_V2 .....	.136
Class: loadsMenus .....	.136
Code File: loadsMenusTable.cs.....	.142
Namespace: Sectrics_V2 .....	.143
Class: loadsMenusTable .....	.143
Code File: loadsMenusTable.Designer.cs.....	.160
Namespace: Sectrics_V2 .....	.160
Class: loadsMenusTable .....	.160
Code File: materialProperties.cs.....	.165
Namespace: Sectrics_V2 .....	.165
Class: materialProperties .....	.165
Code File: materialProperties.Designer.cs.....	.185
Namespace: Sectrics_V2 .....	.185
Class: materialProperties .....	.185
Code File: materialPropertiesTable.cs.....	.191
Namespace: Sectrics_V2 .....	.192
Class: materialPropertiesTable .....	.192

Code File: materialPropertiesTable.Designer.cs.....	211
Namespace: Sectrics_V2 .....	211
Class: materialPropertiesTable .....	211
Code File: Maths.cs.....	216
Namespace: Sectrics_V2 .....	216
Class: Maths .....	216
Code File: members.cs .....	240
Namespace: Sectrics_V2 .....	240
Class: members.....	240
Code File: members.Designer.cs.....	263
Namespace: Sectrics_V2 .....	263
Class: members.....	263
Code File: membersTable.cs.....	269
Namespace: Sectrics_V2 .....	270
Class: membersTable .....	270
Code File: membersTable.Designer.cs.....	287
Namespace: Sectrics_V2 .....	287
Class: membersTable .....	287
Code File: nodes.cs .....	292
Namespace: Sectrics_V2 .....	292
Class: nodes.....	293
Code File: nodes.Designer.cs .....	315
Namespace: Sectrics_V2 .....	315
Class: nodes.....	315
Code File: nodesTable.cs .....	321
Namespace: Sectrics_V2 .....	322
Class: nodesTable .....	322
Code File: nodesTable.Designer.cs.....	340
Namespace: Sectrics_V2 .....	340
Class: nodesTable .....	340
Code File: solveMenus.cs.....	345
Namespace: Sectrics_V2 .....	346
Class: solveMenus .....	346
Interface: IMLSharpPython.....	367
Class: MLSharpPython.....	368
Code File: solveMenus.Designer.cs.....	371

Namespace: Sectrics_V2 .....	371
Class: solveMenus .....	371
Code File: splashScreen.cs .....	376
Namespace: Sectrics_V2 .....	376
Class: splashScreen .....	376
Code File: splashScreen.Designer.cs .....	379
Namespace: Sectrics_V2 .....	379
Class: splashScreen .....	379
Code File: startMenu.cs .....	382
Namespace: Sectrics_V2 .....	382
Class: startMenu .....	382
Code File: startMenu.Designer.cs .....	394
Namespace: Sectrics_V2 .....	394
Class: startMenu .....	394
Code File: Program.cs .....	398
Namespace: Sectrics_V2 .....	398
Class: Program .....	398
Code File: AssemblyInfo.cs .....	401
Code File: supportMenus.cs .....	401
Namespace: Sectrics_V2 .....	401
Class: supportMenus .....	401
Code File: supportMenus.Designer.cs .....	425
Namespace: Sectrics_V2 .....	425
Class: supportMenus .....	425
Code File: supportMenusTable.cs .....	431
Namespace: Sectrics_V2 .....	432
Class: supportMenusTable .....	432
Code File: supportMenusTable.Designer.cs .....	450
Namespace: Sectrics_V2 .....	450
Class: supportMenusTable .....	450
Code File: tutorialMenu.cs .....	455
Namespace: Sectrics_V2 .....	455
Class: tutorialMenu .....	455
Code File: tutorialMenu.Designer.cs .....	466
Namespace: Sectrics_V2 .....	466
Class: tutorialMenu .....	466

Code File: Resources.Designer.cs .....	472
Namespace: Sectrics_V2.Properties .....	472
Class: Resources .....	473
Code File: Settings.Designer.cs .....	475
Namespace: Sectrics_V2.Properties .....	475
Class: Settings .....	475

# System Documentation

## Purpose

The purpose of this System Design Document is to provide a description for how the Sectrics application will be constructed. The Systems Design Document was created to ensure that the Sectrics application will meet the requirements specified in the planning & designing phase. This document provides a description of the system architecture used, software & hardware requirements and additional requirements regarding the scope & construction of the program.

## System Overview

As many truss analysis software follow the finite element method and finding simple truss analysis questions requiring method of joints or method of sections being scarce, most civil engineers when starting their education find that these questions are scarce and far between, especially with answers available. To fix this issue, Sectrics, a civil engineering software has been created.

The Sectrics application goal is to create trusses 2D trusses in a simplified finite element method. This allows beginners to easily create their own trusses without worrying about air resistance or incredibly complicated integrals and allowing them to create questions or validate answers for questions by inputting the specified truss into the program. In addition, by utilization of the finite element method requiring the stiffness matrix, it has allowed this software to be incredibly fast, giving outputs of results instantly with an incredibly high degree of accuracy. This further aid user as it allows them to quickly determine answers of questions and hastily develop new questions without delay.

## Capabilities

The Sectrics tool in addition is also compatible with Microsoft Excel intercepting CSV files in the correct format as bridges thus allowing the quick utilization of CASE tools or external tools to quickly create bridges. The Sectrics application in addition will provide the following capabilities:

- Numerical calculation capabilities providing users calculations of the internal forces affecting members determining the magnitude & direction of tensile / compressive forces
- Numerical calculations providing users calculations of the resultant reaction forces affecting the truss on node supports
- Data import capabilities which allows users to import various data created by the Sectrics application or Microsoft Excel as trusses
- Data export capabilities which allow users to export data to various tools for a simplified reporting & presentation capability including but not limited to; exporting the built truss in a CSV file for use later throughout the program or exporting a picture of the truss in a .png picture file
- Enhanced & additional user interfaces which provides users with a much simpler way for data entry as well as other capabilities

## **Requirements**

### **High-Level Requirements**

The civil engineering software, Sectrics has been approved for development. In order for it to satisfactorily fit the target markets specification, there are several requirements that must be met. Sectrics is a program targeted towards students and therefore must successfully address the requirements of students. By instead of listing complicated data outside of the scope of students' studies but only listing the bare essentials thus being a tool for question validation rather than focus on real-life applications such as proper civil engineering we prevent liability as well as save time and money due to further focusing the scope. For a successful execution of this project, several requirements must be met these include:

- Being able to easily allow users to enter data such as coordinates of nodes, coordinates & magnitude of forces etc.
- Being able to easily be able to receive & interpret output results given by the program. I.e. output of reaction forces in each support
- Being able to quickly respond to user feedback with a tactile response indicating that the program is still functioning whilst adding the factor of responsiveness into the program
- Having the ability to respond to users enter incorrect data via data validation and catching these errors and preventing them from occurring

### **Boundaries**

The “Sectrics” project includes all work regarding planning, designing, building & releasing the product in a commercial environment. This project includes; Identification of the problem, functionality requirements, feasibility study, UI & screen designs, proposed algorithm design & implementation, refined system modelling & technical requirements regarding the software. Items **NOT** included in this project include:

- The full implementation of cross-platform or multi-platform functionality
- Future software support & patches
- Any ongoing patches
- Commercial popularity & campaign
- Collection & utilization of user feedback
- Compatibility for all hardware setups, emphasizing on the minimum hardware requirements
- Compatibility requirements for windows drivers' configurations leading to incompatibility
- Collection and utilization of private user data
- Control & rerouting of network traffic regarding what a system does preventing control of data from other systems
- Certification & reliability of Sectrics

### **Design Constraints**

During the planning & designing phase of the Sectrics application, several constraints have been identifying that will impact and limit the design of the tool. These constraints are beyond the scope of the Sectrics project however must be carefully factored into the system design. To date, the following constraints have been identified.

- The Sectrics application must be compatible for Windows 10 or higher and must have relatively low skill required to install
- The Sectrics application must have incredibly high accuracy and low margin of error as the requirements of this software for truss analysis require so otherwise the consequences may be dire even though this is advertised as a student and not actual construction program.
- The Sectrics application must be compatible with a range of hardware going from low spec computers to high-spec ones. To ensure this we've let the software be relatively small with custom libraries created for speed, however in big analysis jobs the time taken to calculate will increase as the size of the truss increases.

## ***Limitations***

The Sectrics application will be limited by the following:

- The specifications of the hardware
- The wide range of hardware used all having different setups making it more inefficient to optimize for a specific setup and having to compromise and generally optimize for all setups
- The wide ranges of software setups on computers leading to some software interfering or straight up being incompatible with Sectrics
- The requirement of .NET Core, Python as well as NumPy & SciPy
- The size of the bridge will be limited by the limit of how many characters can be in command prompt. This is due to the command prompt requiring the entire bridge to be passed to the prompt to calculate reaction forces

In addition, the Sectrics application will also be limited by:

- Speed: The software will be limited by speed as the calculation section of the program is programmed in Python which is a dynamic language. Dynamic languages are generally easier to program in than static languages however lack in efficiency and for big bridges the matrices required to be inverted and solved can be in the 100s requiring more than 100 factorial operations for the inverse of the function alone. This will cause for lower end PCs to slow down due to the processing power required to process such a matrix and may limit the speed of the program.
- Lack of fool-proof program: As this software is designed for Civil Engineers in mind as well as the finite element method being used mainly by professionally certified Civil Engineers. The margin of error for data entry for such trusses are quite low and a simple error can lead to critical failure for the program. To mitigate the chances of this occurring a fail fast data procedure has been chosen and a numerous of data validation checks are made prior to the official entry of all coordinates. In addition, if an error does occur in the program, a numerous amount of catch statements has been implemented to avoid catastrophic failure of the program causing it to crash and instead will output an error for the user to diagnose or developers to isolate.

## ***Software Requirements***

This application was coded in C# with windows 10 in mind. As this program was made specifically for windows 10 some of the libraries may not be existent on other platforms leading to incompatibility. Due to this to run the Sectrics application Windows 10 is a required pre-requisite software. In addition, Python 2.7 or higher is required with NumPy & SciPy installed.

### **Minimum:**

- Python 2.7 with libraries including:
  - o NumPy
  - o SciPy
- .NET Core

### **Recommended:**

- Python 3.7.3 with libraries including:
  - o NumPy
  - o SciPy
- .NET Framework

## ***Hardware Requirements***

The minimum hardware specifications for this application to run requires the minimum hardware requirements for Windows 10 to run. Due to this the minimum hardware requirement are:

- **Processor:** 1 gigahertz (GHz) or faster processor or SoC
- **RAM:** 1 gigabyte (GB) for 32-bit or 2 GB for 64-bit
- **Hard disk space:** 16 GB for 32-bit OS 20 GB for 64-bit OS
- **Graphics card:** DirectX 9 or later with WDDM 1.0 driver
- **Display:** 1920x1080

### ***Recommended Specifications***

Depending on the user's bridge requirements the specifications of the software varies, for commercial applications such as calculating the forces on a real-world bridge with hundreds to thousands of nodes and members, the utilization of more powerful hardware is required, for programs requiring such processing power the recommended specifications are:

- **Processor:** Intel® Core™ i7-6700K @ 4.4GHz-Hi-Perf
- **Memory:** 16GB DDR4
- **Graphics:** NVIDIA® GeForce® GTX 1080 OC 8GB GDDR5X VRAM PCIe
- **Storage:** 512GB Solid State Drive NVMe Samsung® 950 PRO M.2 PCIe 3.0 x4 2500/1500MB/s
- **Display:** 1920x1080

## ***Operational Requirements***

The operational requirements for this program are non-critical as it is just a game and failure will not result into financial and or physical damage to a user or hardware. The only issue if this program fails is the dissatisfaction of a user from having an enjoyable game experience. The reliability of this system is non-vital however if failure was to occur it would result into annoyance, and if failure was to occur consistently this would lead to a failed product. Steps taken to avoid failure and to increase reliability is include:

- Having error exception catchers, catching if the user inputs a value outside of the range and asking the user to repeat his/her selection
- Containing a simple foolproof GUI as well as minimal setting options leading to the user having a minimal chance of causing failure
- Requesting testers from all age groups to test the product and ask for user feedback and documentation and repeating errors finding the cause and eliminating them from the final product

## ***Maintenance***

After the commercial release of the software, the software solution will be officially supported for an approximate minimum lifespan of 12 months. If the Sectrics application is a success the official support life of the program may be extended and a revised version may be developed. The customer support however will vary being on a range of tiers. For the “student edition” version of Sectrics, support will be at minimal and will not be given, requiring users to read documentation as well as if determining a bug to report it to be fixed at a later date. For the commercial edition of the software or beta testers a high level of support will be given having a direct line with customers ensuring that bugs are patched and that user manuals are up to code as well as individually catering for business’s needs.

# Final Algorithm Design

## Detailed Design

In this section the detailed design of the SectricsV2 is presented.

### **Project: SectricsV2**

This document contains documentation for the project SectricsV2, and its contents.

#### **Code File: aboutMenus.cs**

```
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

#### **Class: aboutMenus**

```
public partial class aboutMenus : Form
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

#### **Procedure: Constructor**

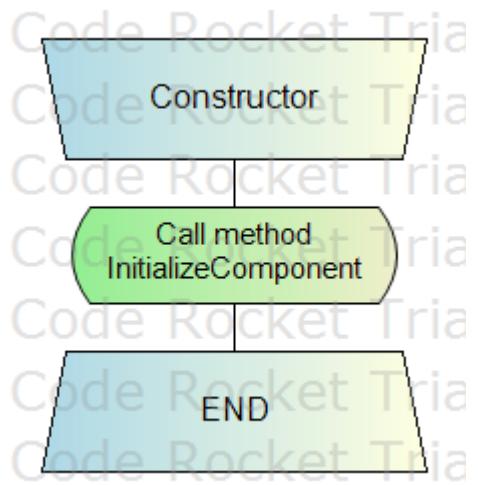
```
public aboutMenus()
```

Returns aboutMenus

#### **Pseudocode**

```
Call method InitializeComponent
```

Flowchart



**Procedure: WndProc**

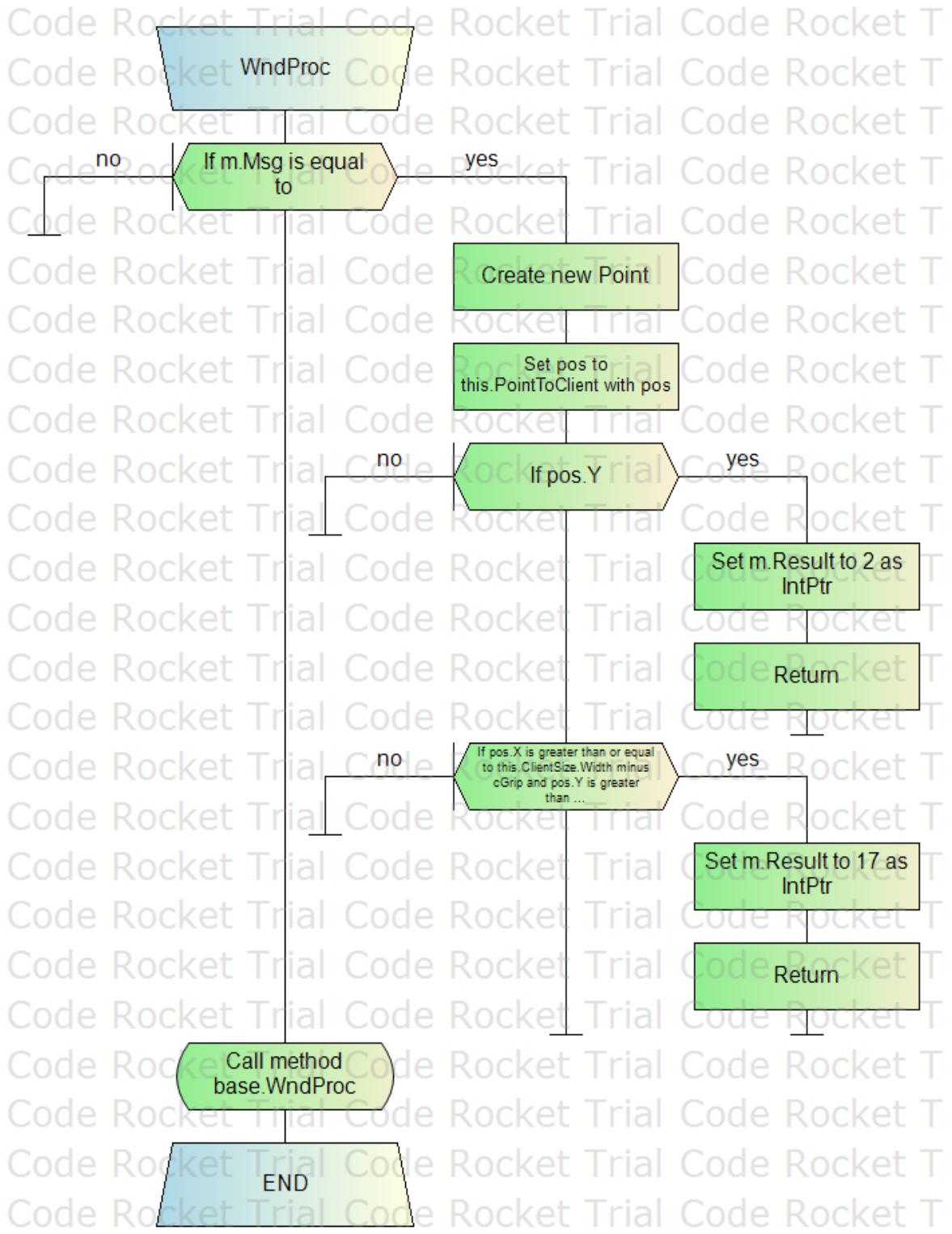
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

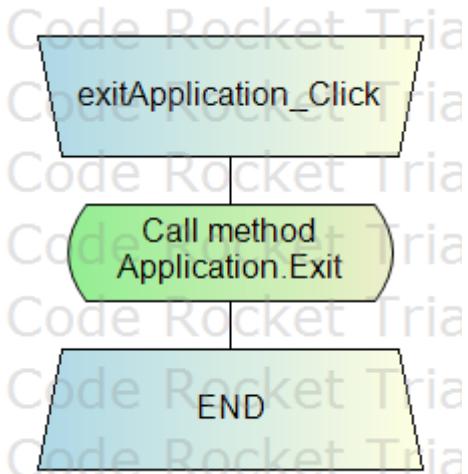
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

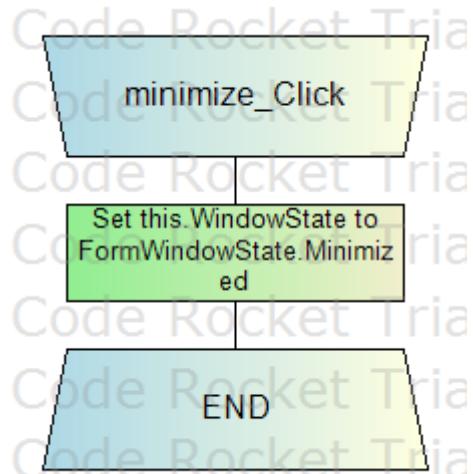
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

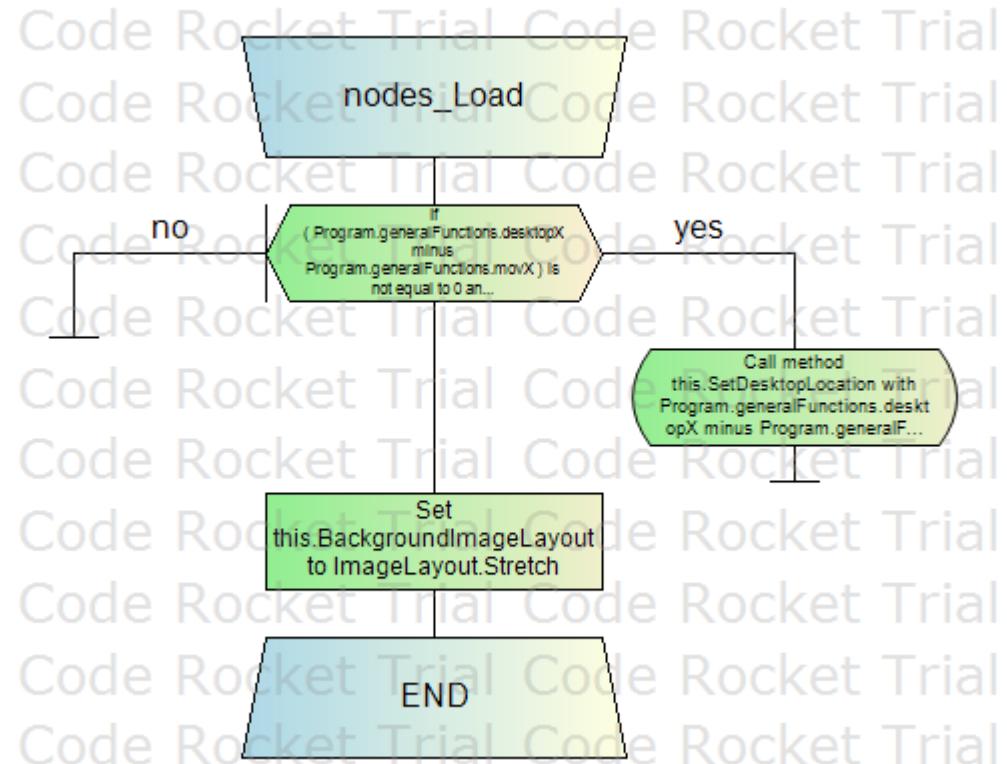
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

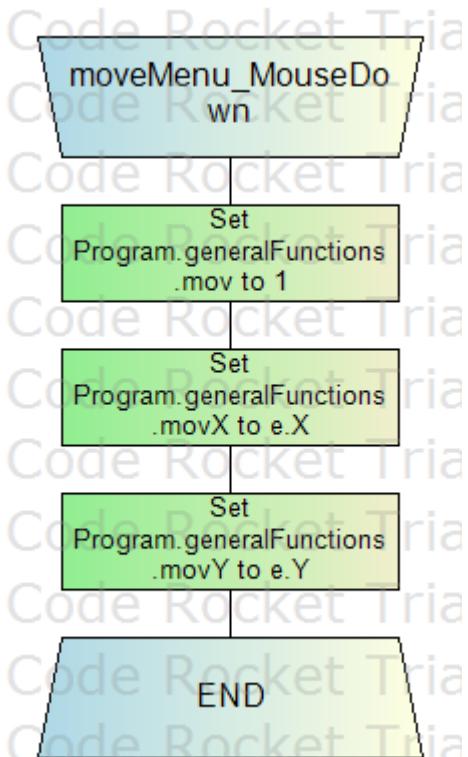
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

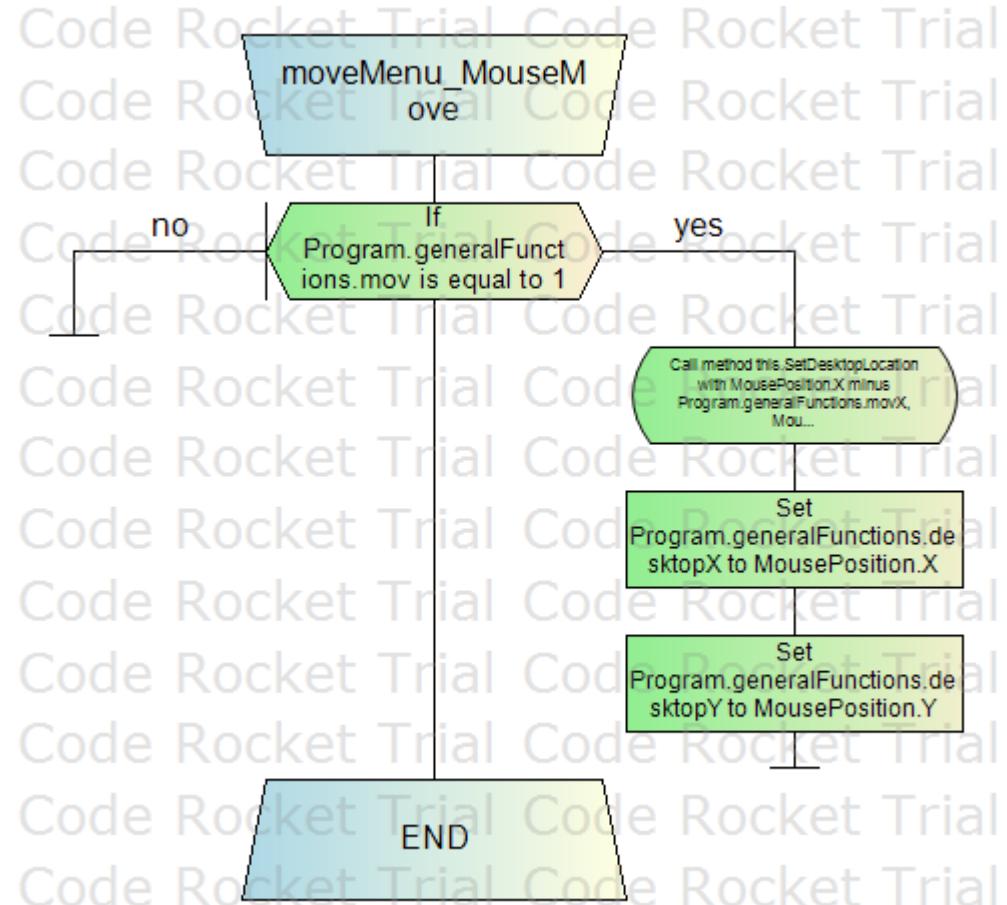
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



*Procedure: moveMenu\_MouseUp*

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

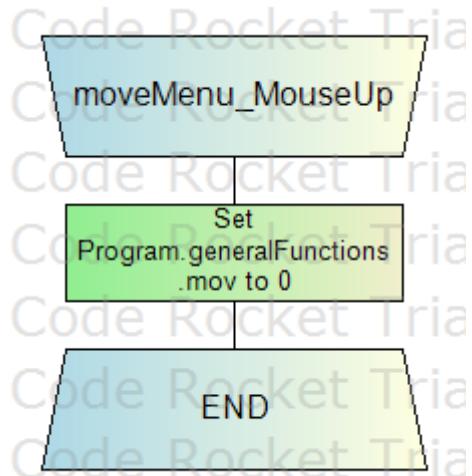
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

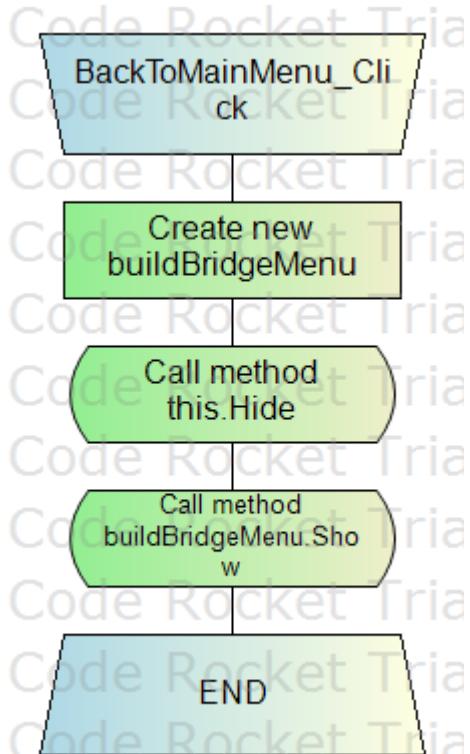
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



*Code File: aboutMenus.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

**Class: aboutMenus**

partial class aboutMenus

**Attribute: components**

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure: Dispose**

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

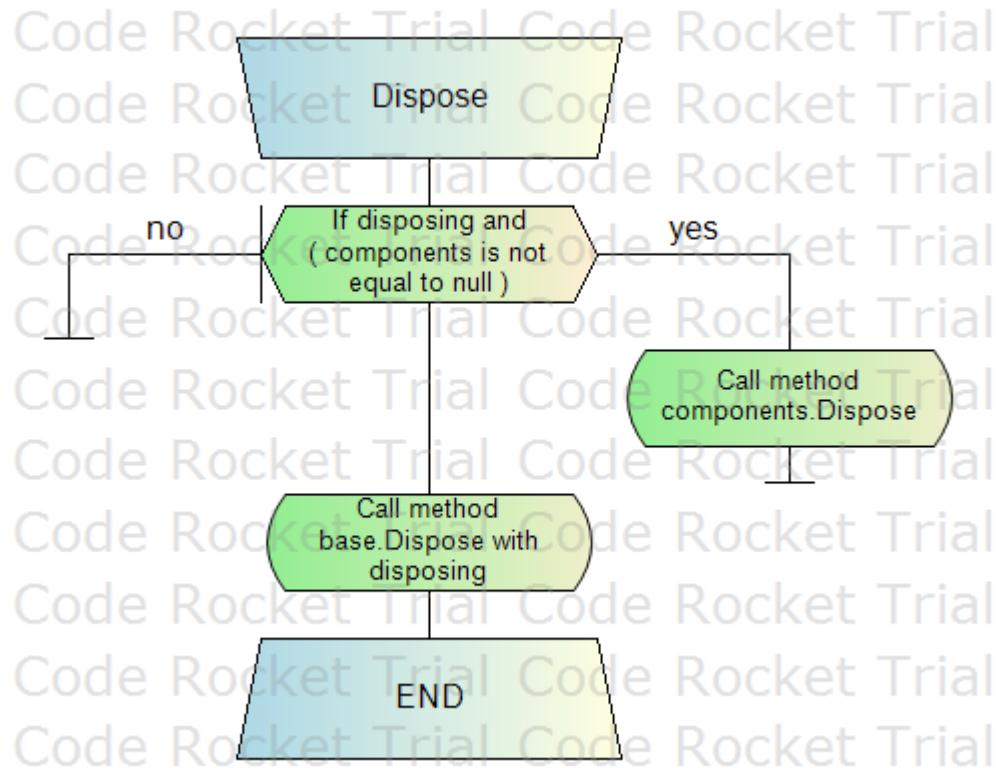
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

**Flowchart**



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.PictureBox
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.Button
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackColor to System.Drawing.Color.Transparent
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.TabStop to false
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackColor to System.Drawing.Color.Transparent
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.TabStop to false
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
label1
Set this.label1.AutoSize to true
Set this.label1.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 12
Set this.label1.Text to "YEAR 12 SOFTWARE DEVELOPMENT & DESIGN HSC MAJOR BY SHAAN
KHAN 2019"

```

```

label2
Set this.label2.AutoSize to true
Set this.label2.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label2.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label2.Name to "label2"
Create new System.Drawing.Size
Set this.label2.TabIndex to 13
Set this.label2.Text to "EMAIL: SHAANKHAN101@GMAIL.COM"
label3
Set this.label3.AutoSize to true
Set this.label3.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label3.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label3.Name to "label3"
Create new System.Drawing.Size
Set this.label3.TabIndex to 14
Set this.label3.Text to "GITHUB: GITHUB.COM/SHAANCODING"
label4
Set this.label4.AutoSize to true
Set this.label4.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label4.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label4.Name to "label4"
Create new System.Drawing.Size
Set this.label4.TabIndex to 15
Set this.label4.Text to "WEBSITE: WWW.SHAANCODING.COM"
pictureBox1
Set this.pictureBox1.BackColor to System.Drawing.Color.Transparent
Set this.pictureBox1.Image
Create new System.Drawing.Point
Set this.pictureBox1.Name to "pictureBox1"
Create new System.Drawing.Size
Set this.pictureBox1.TabIndex to 16
Set this.pictureBox1.TabStop to false
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 39
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent
Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ), ( ( int ) ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 40
Set this.BackToMainMenu.TabStop to false

```

```

Set this.BackToMainMenu.Text to "BACK"
Set this.BackToMainMenu.UseVisualStyleBackColor to false
This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
aboutMenus
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.BackToMainMenu
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.pictureBox1
Call method this.Controls.Add with this.label4
Call method this.Controls.Add with this.label3
Call method this.Controls.Add with this.label2
Call method this.Controls.Add with this.label1
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.DoubleBuffered to true
Set this.ForeColor to System.Drawing.SystemColors.ControlText
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "aboutMenus"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitializeInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout

```

## Flowchart



```
#endregion  
Attribute: minimize  
  
private System.Windows.Forms.Button minimize;  
Attribute: exitApplication  
  
private System.Windows.Forms.Button exitApplication;  
Attribute: label1  
  
private System.Windows.Forms.Label label1;  
Attribute: label2  
  
private System.Windows.Forms.Label label2;  
Attribute: label3  
  
private System.Windows.Forms.Label label3;  
Attribute: label4  
  
private System.Windows.Forms.Label label4;  
Attribute: pictureBox1  
  
private System.Windows.Forms.PictureBox pictureBox1;  
Attribute: moveMenu  
  
private System.Windows.Forms.Panel moveMenu;  
Attribute: BackToMainMenu  
  
private System.Windows.Forms.Button BackToMainMenu;
```

**Code File:** bridgeData.cs

```
using System;  
using System.Collections.Generic;  
using System.Text;  
Namespace: Sectrics_V2
```

namespace Sectrics\_V2

**Class: Nodes**

```
public class Nodes
```

**Property: NodeX**

```
public double NodeX { get; set; }
```

**Property: NodeY**

```
public double NodeY { get; set; }
```

**Class: connectedMembers**

```
public class connectedMembers
```

*Property: toConnection*

```
public int toConnection { get; set; }
```

*Property: fromConnection*

```
public int fromConnection { get; set; }
```

**Class: forces**

```
public class forces
```

*Property: xMagnitudeForces*

```
public double xMagnitudeForces { get; set; }
```

*Property: yMagnitudeForces*

```
public double yMagnitudeForces { get; set; }
```

**Class: degreesOfFreedom**

```
public class degreesOfFreedom
```

*Property: xDegreeOfFreedom*

```
public int xDegreeOfFreedom { get; set; }
```

*Property: yDegreesOfFreedom*

```
public int yDegreesOfFreedom { get; set; }
```

**Class: BridgeData**

```
class BridgeData
```

*Attribute: supportType*

```
public List<string> supportType = new List<string>();
```

*Attribute: supportNode*

```
public List<int> supportNode = new List<int>();
```

*Attribute: nodes*

```
public List<Nodes> nodes = new List<Nodes>();
```

*Attribute: degreesOfFreedom*

```
public List<degreesOfFreedom> degreesOfFreedom = new  
List<degreesOfFreedom>();
```

*Attribute: memberConnection*

```
public List<connectedMembers> memberConnection = new  
List<connectedMembers>();
```

*Attribute: restrainedDegreesOfFreedom*

```
public List<int> restrainedDegreesOfFreedom = new List<int>();
```

**Attribute:** forces

```
public List<forces> forces = new List<forces>();
```

**Attribute:** stiffness

```
public List<double> stiffness = new List<double>();
```

**Attribute:** areas

```
public List<double> areas = new List<double>();
```

**Attribute:** ndof

```
public int ndof;
```

**Attribute:** nodesIndex

```
public int nodesIndex = 0;
```

**Attribute:** memberIndex

```
public int memberIndex = 0;
```

**Attribute:** materialPropertiesIndex

```
public int materialPropertiesIndex = 0;
```

**Code File:** buildBridgeMenu.cs

```
using Sectrics_V2.Properties;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

**Namespace:** Sectrics\_V2

```
namespace Sectrics_V2
```

**Class:** buildBridgeMenu

```
public partial class buildBridgeMenu : Form
```

**Attribute:** zoom

```
float zoom = 1f;
```

*Attribute: xMouseOffset*

```
double xMouseOffset;
```

*Attribute: yMouseOffset*

```
double yMouseOffset;
```

*Attribute: cGrip*

```
private const int cGrip = 16;
```

*Attribute: cCaption*

```
private const int cCaption = 32;
```

*Procedure: Constructor*

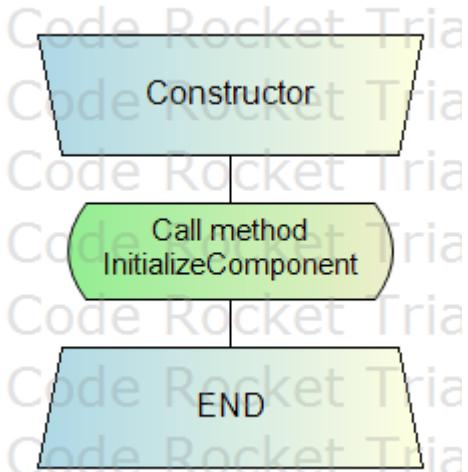
```
public buildBridgeMenu()
```

Returns buildBridgeMenu

**Pseudocode**

```
Call method InitializeComponent
```

Flowchart



**Procedure: WndProc**

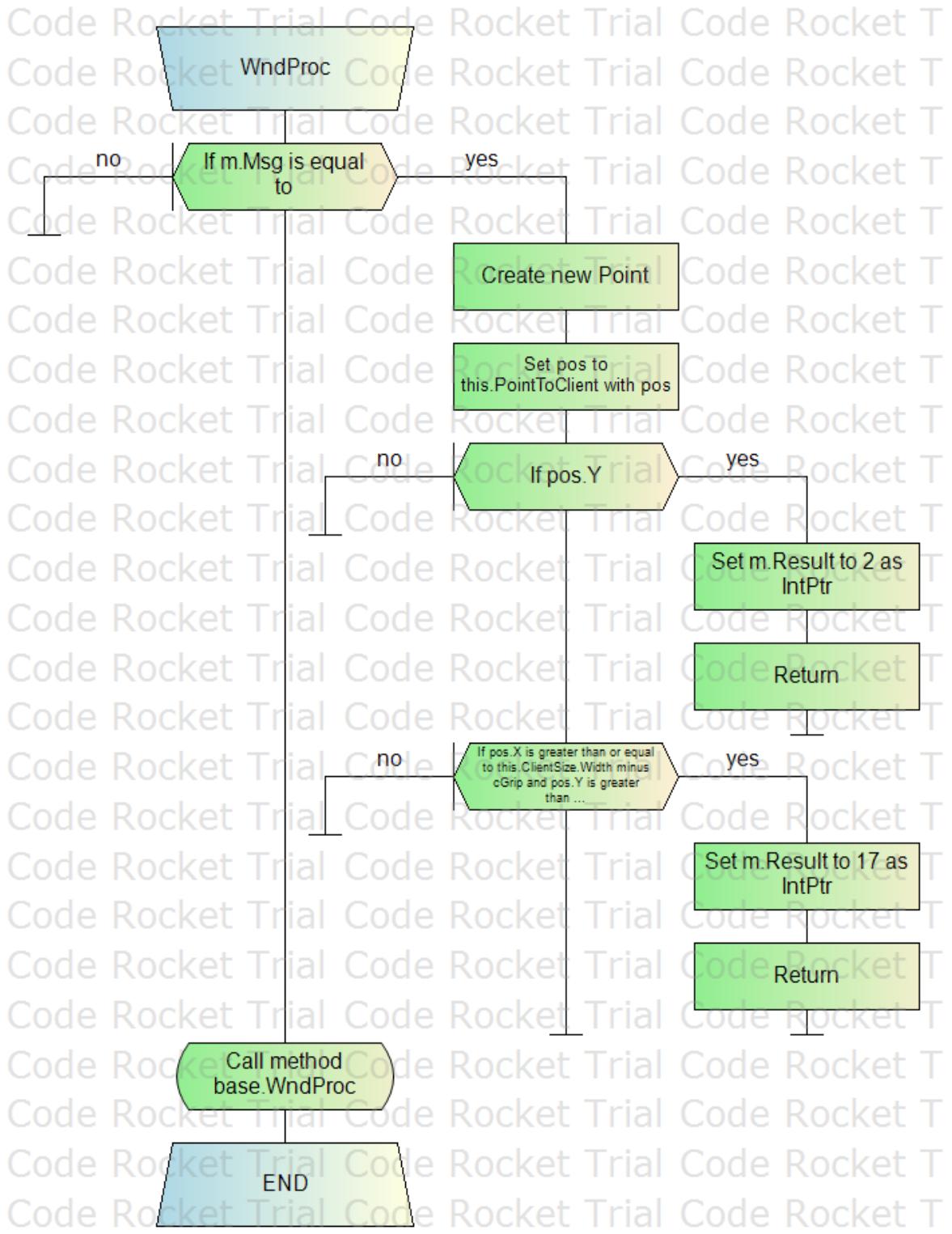
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

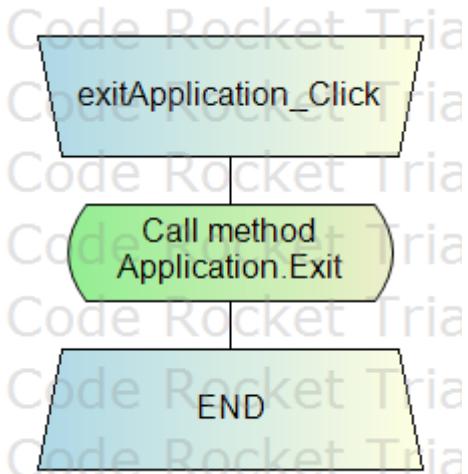
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

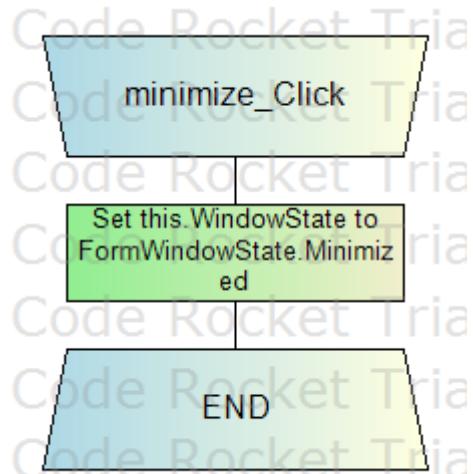
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodesMenu\_Click**

```
private void nodesMenu_Click(object sender, EventArgs e)
```

**Parameters**

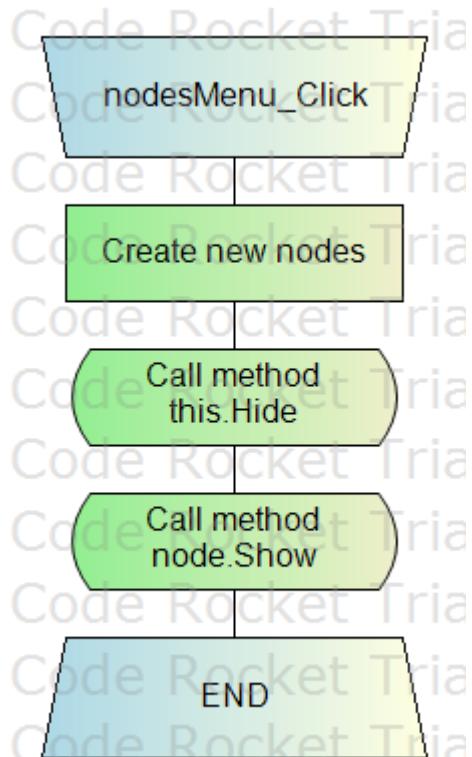
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new nodes  
Call method this.Hide  
Call method node.Show
```

**Flowchart**



**Procedure: membersMenu\_Click**

```
private void membersMenu_Click(object sender, EventArgs e)
```

**Parameters**

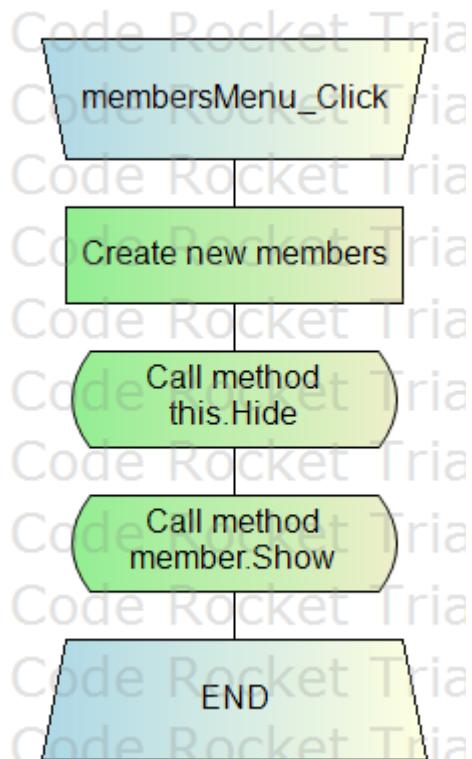
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new members  
Call method this.Hide  
Call method member.Show
```

**Flowchart**



**Procedure: materialPropertiesMenu\_Click**

```
private void materialPropertiesMenu_Click(object sender, EventArgs e)
```

**Parameters**

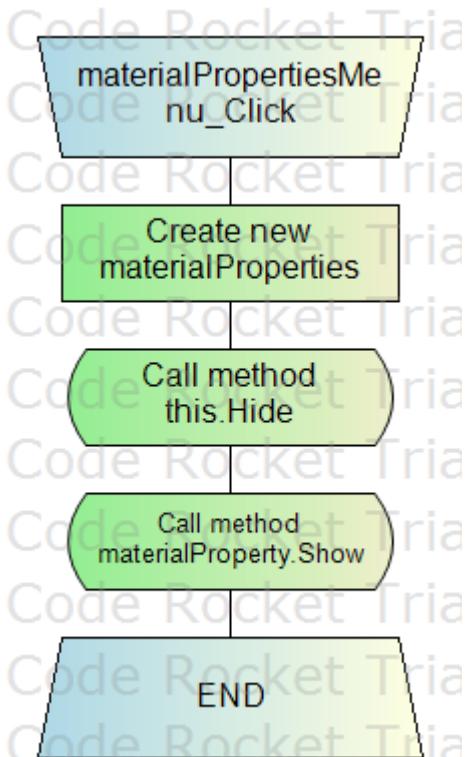
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new materialProperties  
Call method this.Hide  
Call method materialProperty.Show
```

**Flowchart**



*Procedure: supportsMenu\_Click*

```
private void supportsMenu_Click(object sender, EventArgs e)
```

**Parameters**

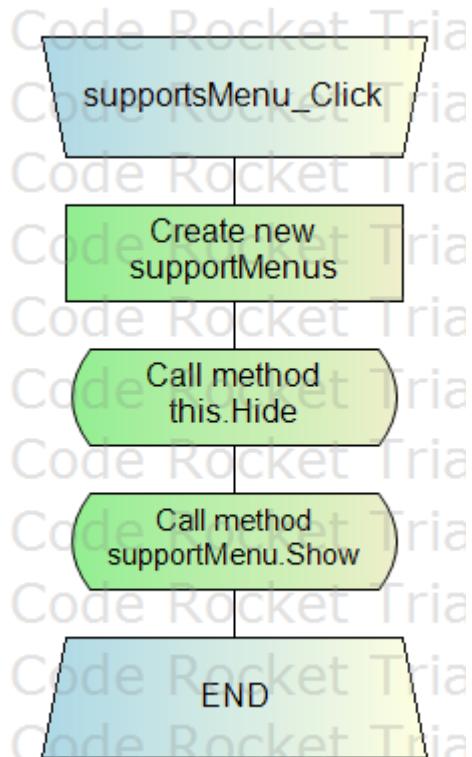
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new supportMenus  
Call method this.Hide  
Call method supportMenu.Show
```

**Flowchart**



**Procedure: loadsMenu\_Click**

```
private void loadsMenu_Click(object sender, EventArgs e)
```

**Parameters**

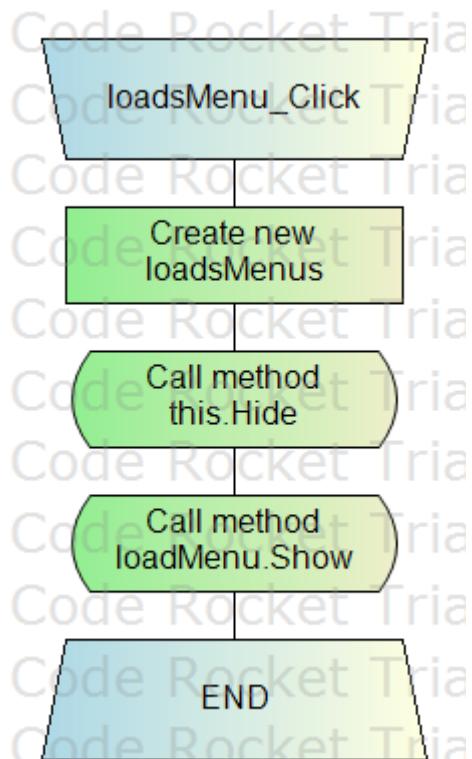
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new loadsMenus  
Call method this.Hide  
Call method loadMenu.Show
```

Flowchart



**Procedure: solveMenu\_Click**

```
private void solveMenu_Click(object sender, EventArgs e)
```

**Parameters**

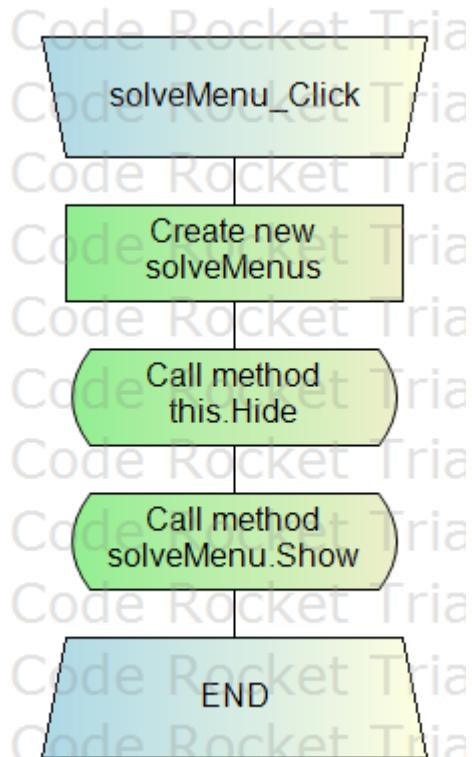
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new solveMenus  
Call method this.Hide  
Call method solveMenu.Show
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

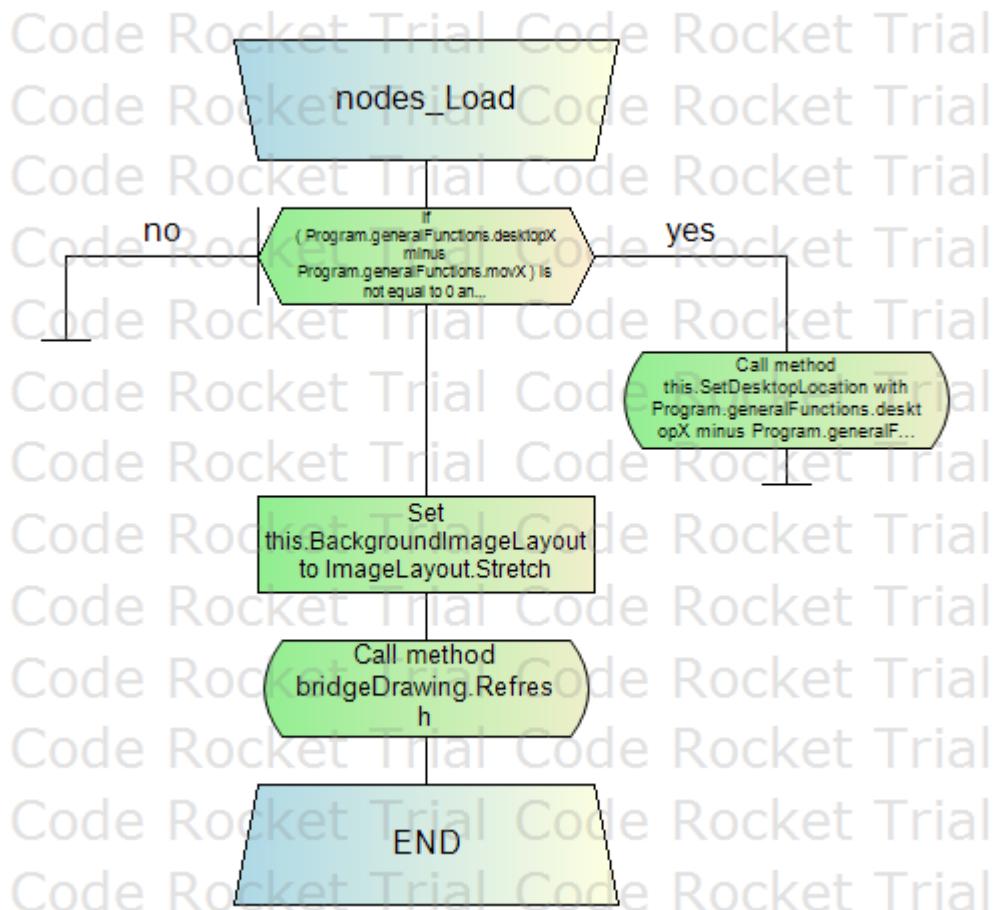
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch  
Call method bridgeDrawing.Refresh
```

**Flowchart**



**Procedure: aboutMenu\_Click**

```
private void aboutMenu_Click(object sender, EventArgs e)
```

**Parameters**

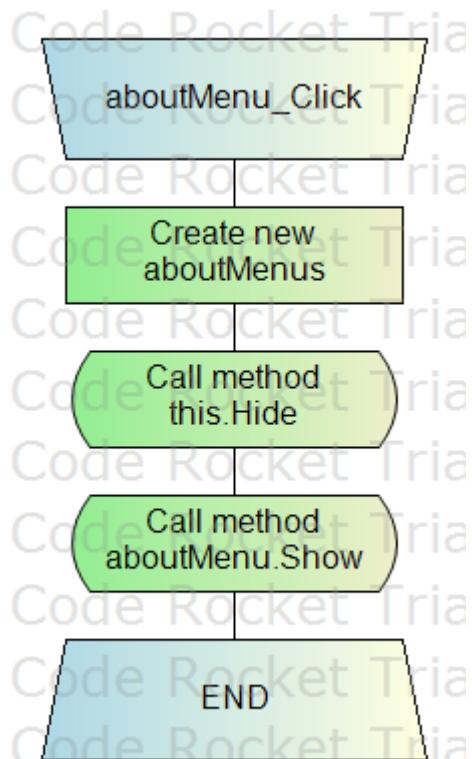
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new aboutMenus  
Call method this.Hide  
Call method aboutMenu.Show
```

Flowchart



**Procedure: exitMenu\_Click**

```
private void exitMenu_Click(object sender, EventArgs e)
```

**Parameters**

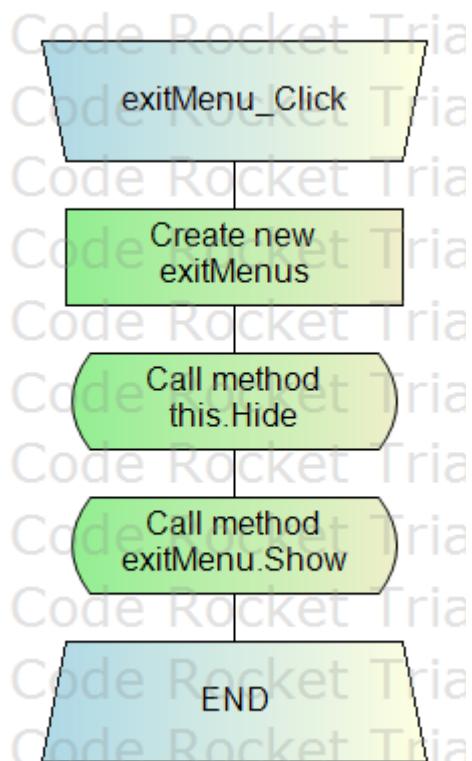
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new exitMenus  
Call method this.Hide  
Call method exitMenu.Show
```

Flowchart



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

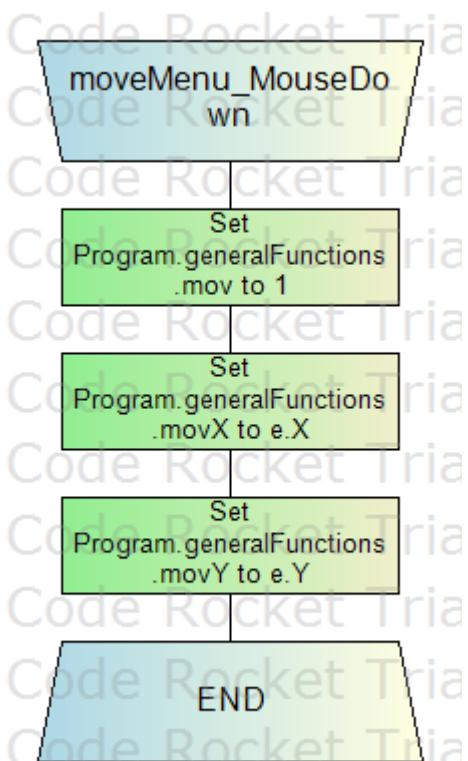
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

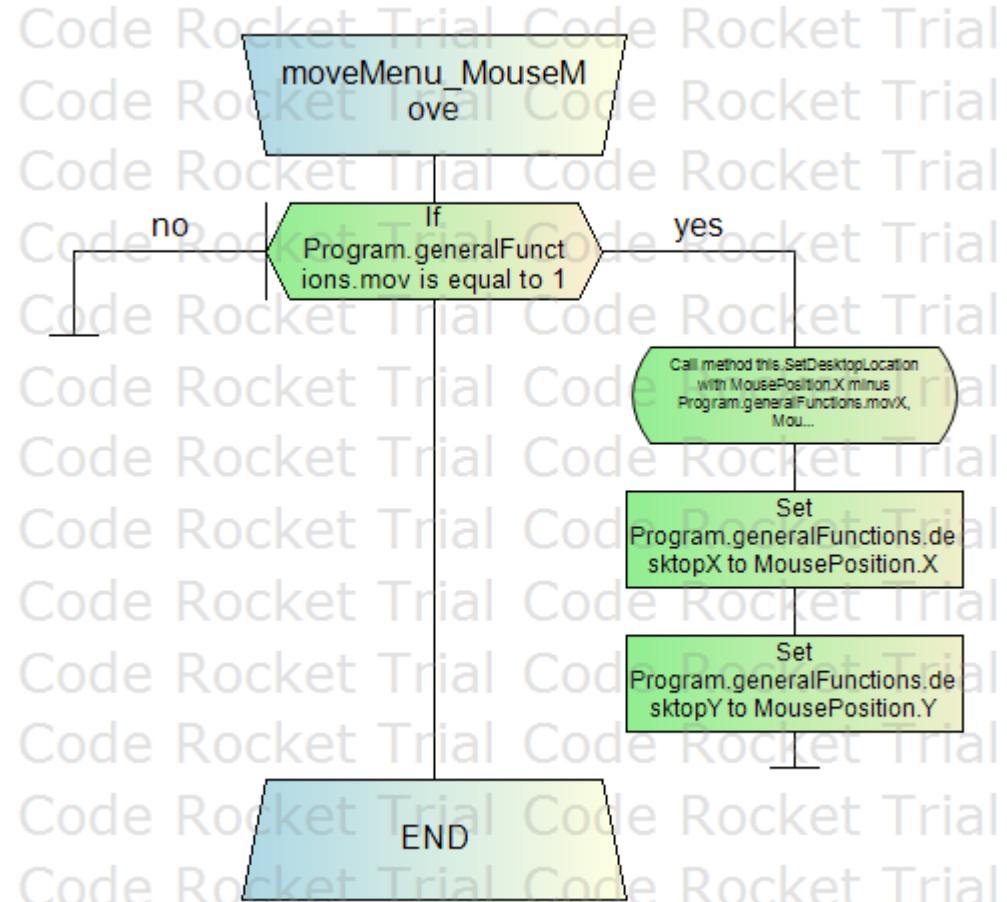
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



*Procedure: moveMenu\_MouseUp*

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

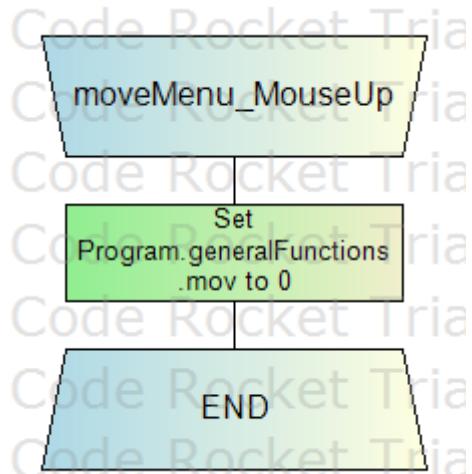
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

**Try**

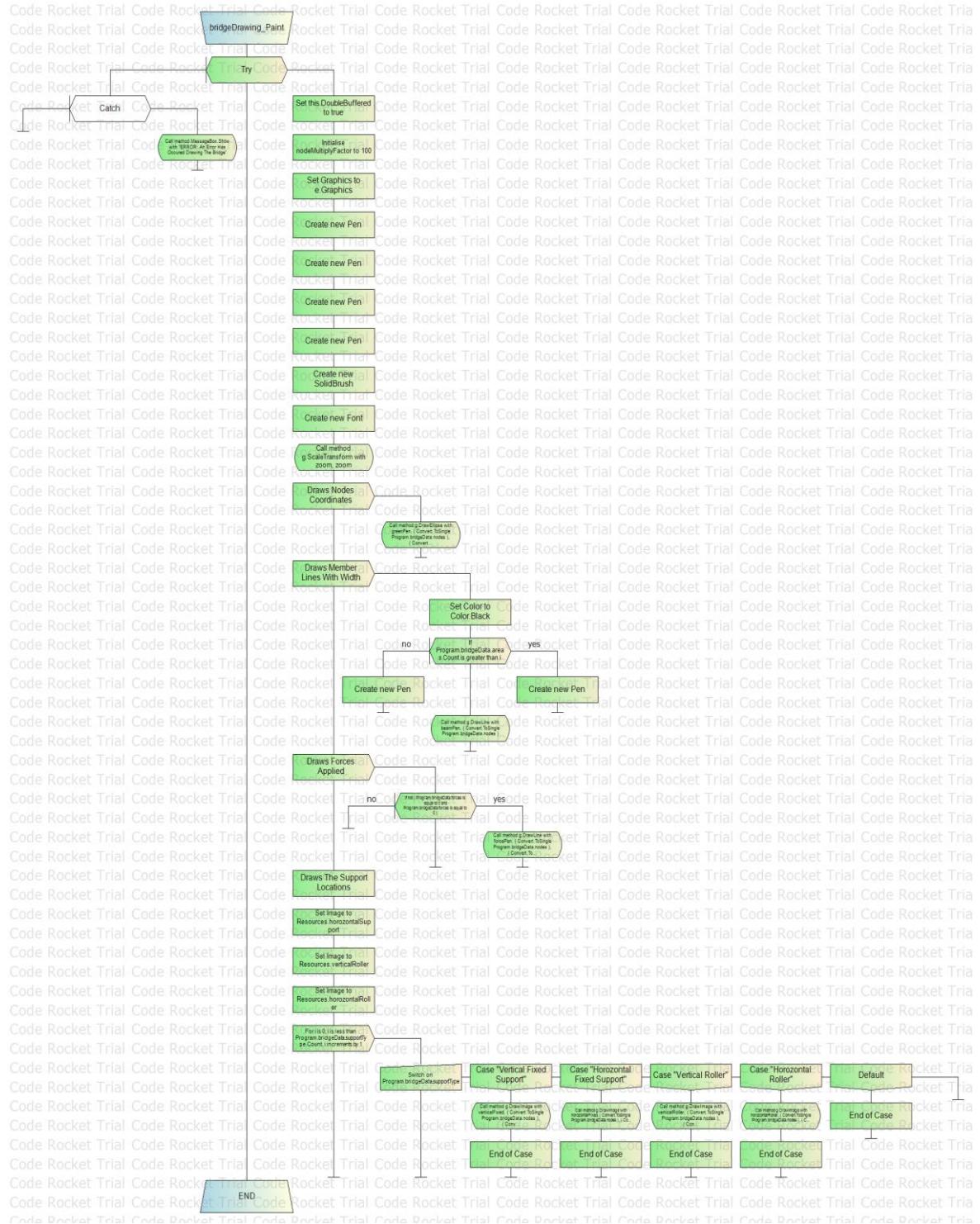
```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```

        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry

```

### Flowchart



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

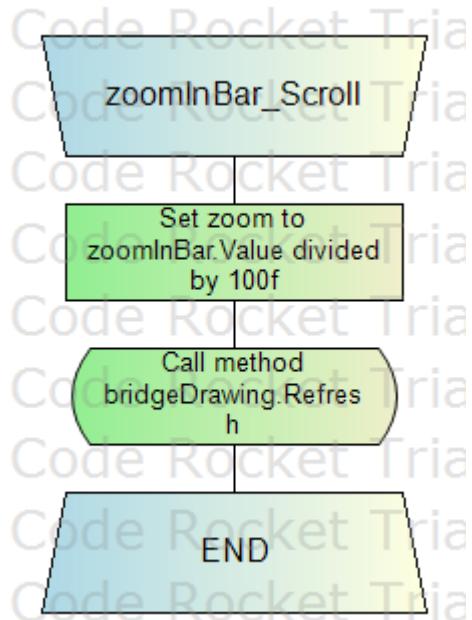
Returns Void

**Pseudocode**

Set zoom to zoomInBar.Value divided by 100f

Call method bridgeDrawing.Refresh

**Flowchart**



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

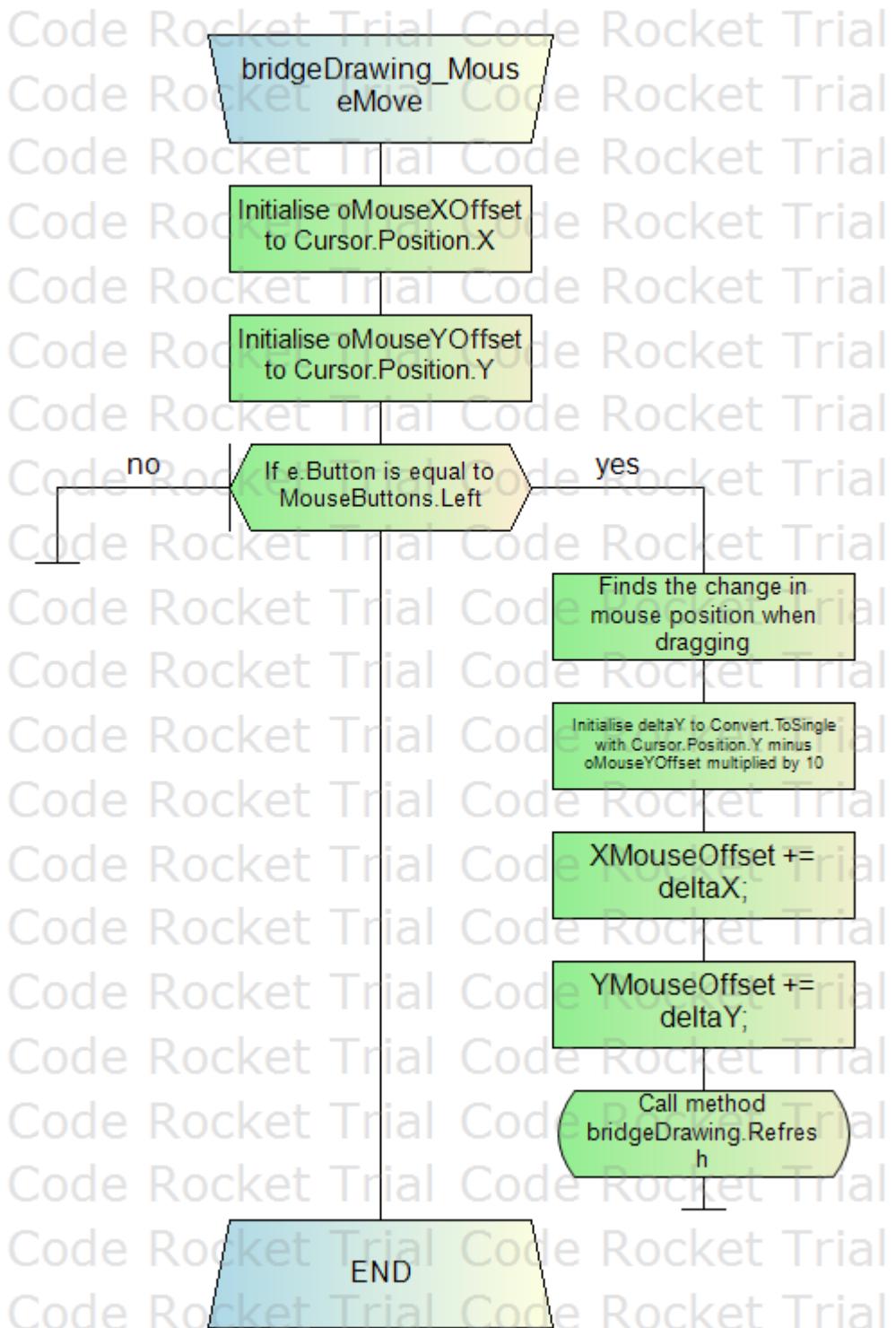
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



**Procedure: label5\_Click**

```
private void label5_Click(object sender, EventArgs e)
```

**Parameters**

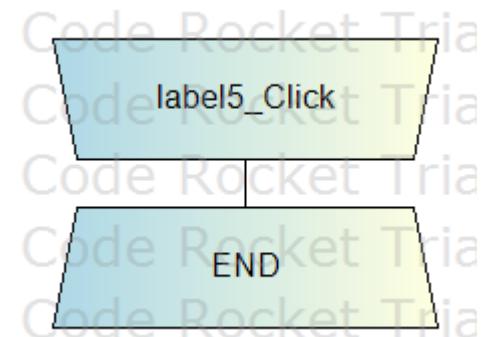
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: loadBridge\_Click**

```
private void loadBridge_Click(object sender, EventArgs e)
```

**Parameters**

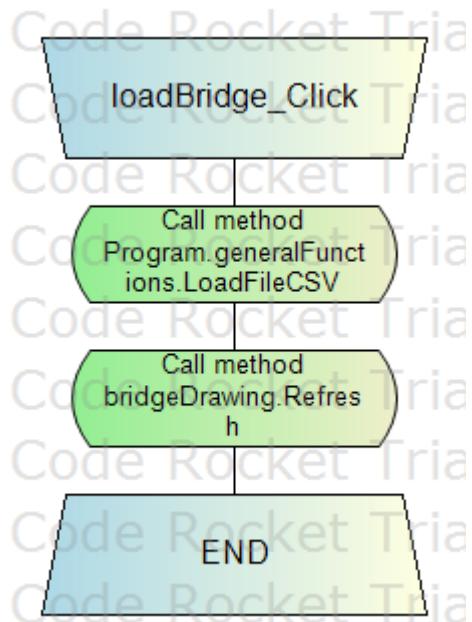
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Program.generalFunctions.LoadFileCSV  
Call method bridgeDrawing.Refresh
```

**Flowchart**



**Procedure: saveBridge\_Click**

```
private void saveBridge_Click(object sender, EventArgs e)
```

**Parameters**

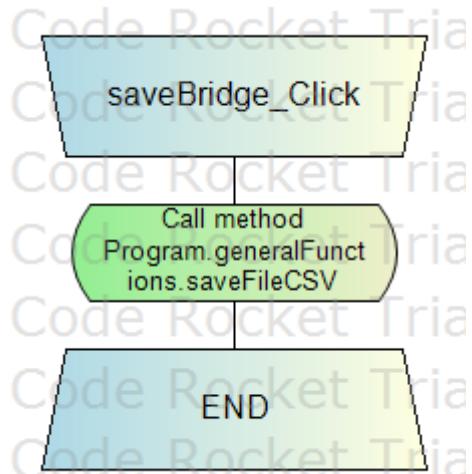
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Program.generalFunctions.saveFileCSV
```

**Flowchart**



**Procedure: newBridge\_Click**

```
private void newBridge_Click(object sender, EventArgs e)
```

**Parameters**

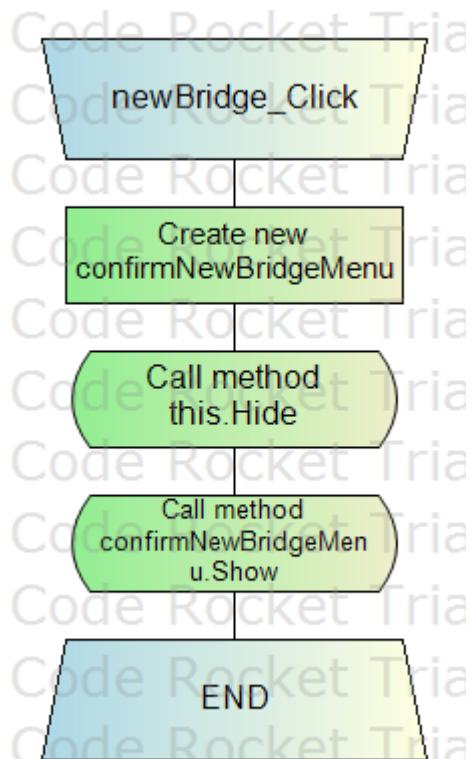
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new confirmNewBridgeMenu  
Call method this.Hide  
Call method confirmNewBridgeMenu.Show
```

**Flowchart**



**Procedure: savePictureOfBridge\_Click**

```
private void savePictureOfBridge_Click(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

**Try**

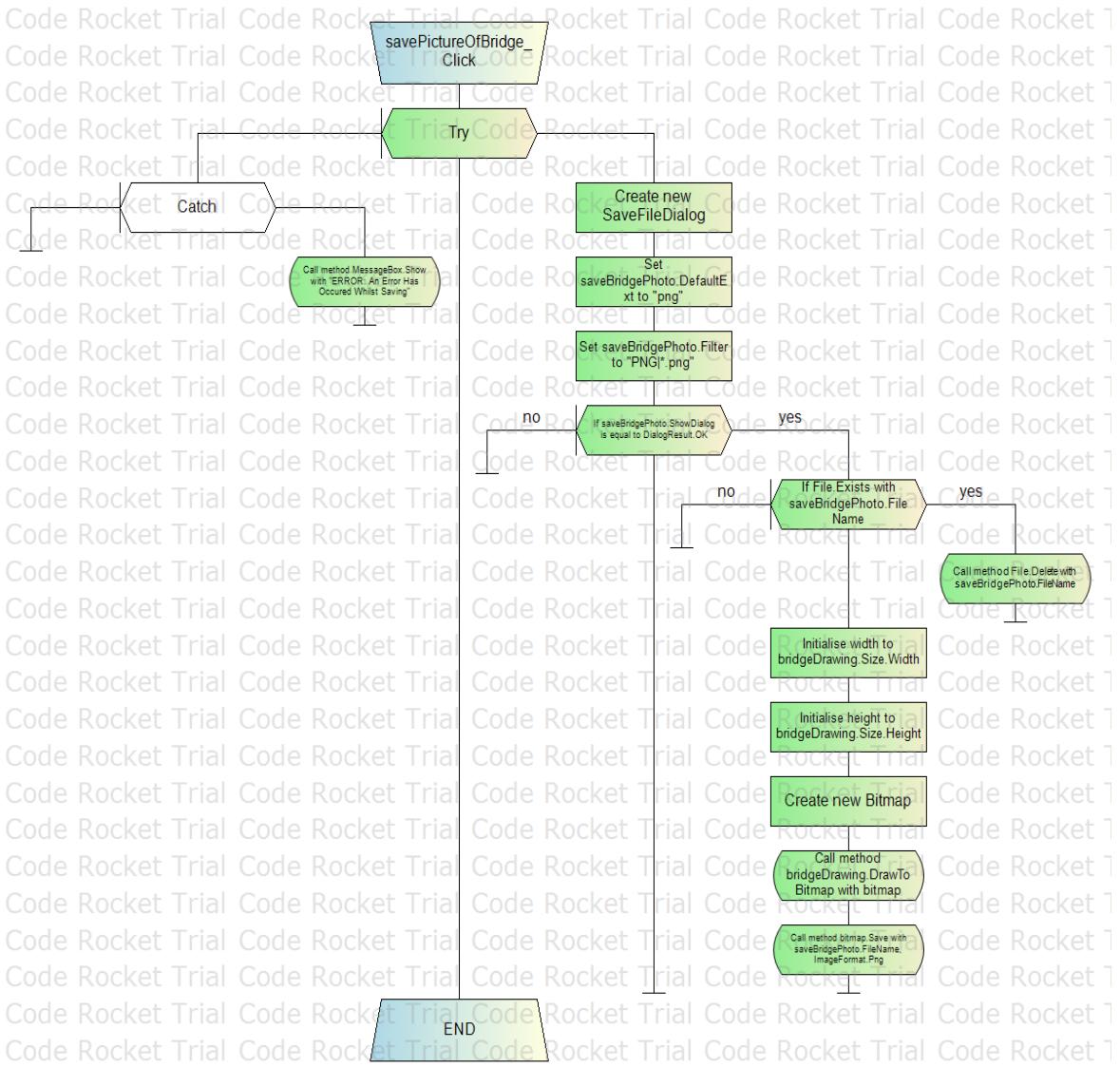
```
    Create new SaveFileDialog  
    Set saveBridgePhoto.DefaultExt to "png"  
    Set saveBridgePhoto.Filter to "PNG|*.png"  
    If saveBridgePhoto.ShowDialog is equal to DialogResult.OK  
        If File.Exists with saveBridgePhoto.FileName  
            Call method File.Delete with saveBridgePhoto.FileName  
        EndIf  
        Initialise width to bridgeDrawing.Size.Width  
        Initialise height to bridgeDrawing.Size.Height  
        Create new Bitmap  
        Call method bridgeDrawing.DrawToBitmap with bitmap  
        Call method bitmap.Save with saveBridgePhoto.FileName, ImageFormat.Png  
    EndIf
```

**Catch**

```
    Call method MessageBox.Show with "ERROR: An Error Has Occured Whilst Saving"
```

**EndTry**

**Flowchart**



*Code File: buildBridgeMenu.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

*Class: buildBridgeMenu*

partial class buildBridgeMenu

*Attribute: components*

private System.ComponentModel.IContainer components = null;

Required designer variable.

*Procedure: Dispose*

protected override void Dispose (bool disposing)

Clean up any resources being used.

*Parameters*

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

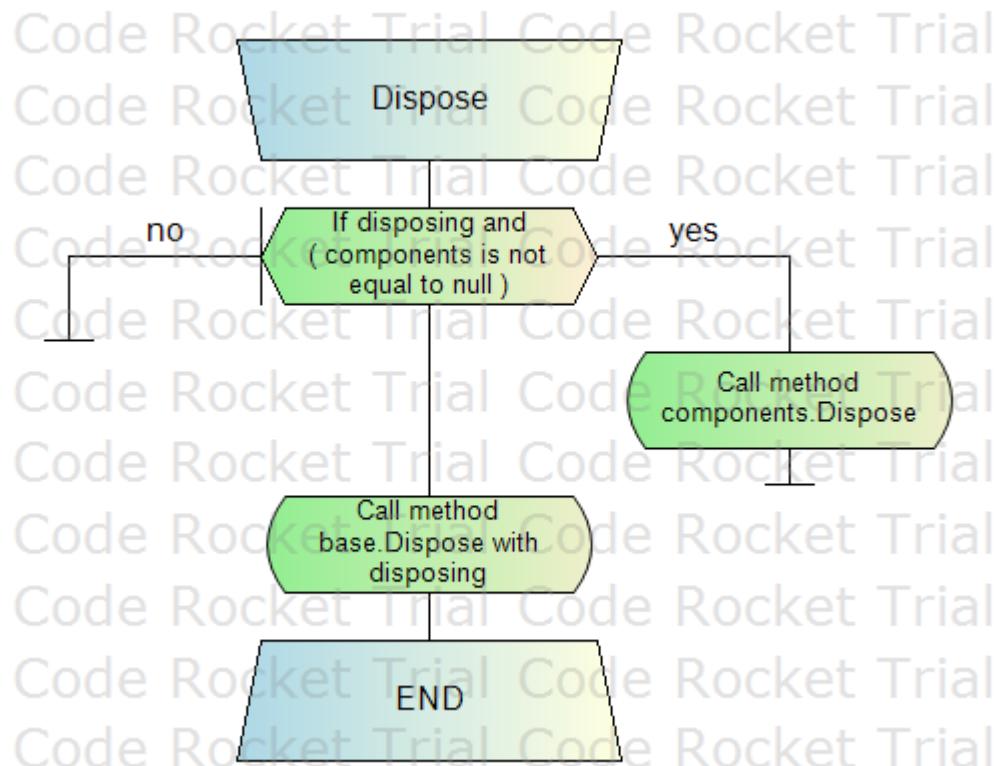
*Pseudocode*

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

*Flowchart*



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.Label
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.UseVisualStyleBackColor to false

```

```

This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
nodesMenu
Set this.nodesMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.nodesMenu.FlatAppearance.BorderSize to 2
Set this.nodesMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.nodesMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.nodesMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.nodesMenu.Name to "nodesMenu"
Create new System.Drawing.Size
Set this.nodesMenu.TabIndex to 4
Set this.nodesMenu.Text to "NODES"
Set this.nodesMenu.UseVisualStyleBackColor to false
This.nodesMenu.Click += new System.EventHandler(this.nodesMenu_Click);
membersMenu
Set this.membersMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.membersMenu.FlatAppearance.BorderSize to 2
Set this.membersMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.membersMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.membersMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.membersMenu.Name to "membersMenu"
Create new System.Drawing.Size
Set this.membersMenu.TabIndex to 5
Set this.membersMenu.Text to "MEMBERS"
Set this.membersMenu.UseVisualStyleBackColor to false
This.membersMenu.Click += new System.EventHandler(this.membersMenu_Click);
supportsMenu
Set this.supportsMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.supportsMenu.FlatAppearance.BorderSize to 2
Set this.supportsMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.supportsMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.supportsMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.supportsMenu.Name to "supportsMenu"
Create new System.Drawing.Size
Set this.supportsMenu.TabIndex to 6
Set this.supportsMenu.Text to "SUPPORTS"
Set this.supportsMenu.UseVisualStyleBackColor to false
This.supportsMenu.Click += new System.EventHandler(this.supportsMenu_Click);
loadsMenu
Set this.loadsMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.loadsMenu.FlatAppearance.BorderSize to 2
Set this.loadsMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.loadsMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.loadsMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.loadsMenu.Name to "loadsMenu"
Create new System.Drawing.Size
Set this.loadsMenu.TabIndex to 7
Set this.loadsMenu.Text to "LOADS"
Set this.loadsMenu.UseVisualStyleBackColor to false
This.loadsMenu.Click += new System.EventHandler(this.loadsMenu_Click);
solveMenu
Set this.solveMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.solveMenu.FlatAppearance.BorderSize to 2

```

```

Set this.solveMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.solveMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.solveMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.solveMenu.Name to "solveMenu"
Create new System.Drawing.Size
Set this.solveMenu.TabIndex to 8
Set this.solveMenu.Text to "SOLVE"
Set this.solveMenu.UseVisualStyleBackColor to false
This.solveMenu.Click += new System.EventHandler(this.solveMenu_Click);
aboutMenu
Set this.aboutMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.aboutMenu.FlatAppearance.BorderSize to 2
Set this.aboutMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.aboutMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.aboutMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.aboutMenu.Name to "aboutMenu"
Create new System.Drawing.Size
Set this.aboutMenu.TabIndex to 9
Set this.aboutMenu.Text to "ABOUT"
Set this.aboutMenu.UseVisualStyleBackColor to false
This.aboutMenu.Click += new System.EventHandler(this.aboutMenu_Click);
exitMenu
Set this.exitMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.exitMenu.FlatAppearance.BorderSize to 2
Set this.exitMenu.FlatAppearance.MouseOverBackColor to System.Drawing.Color.FromArgb
with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.exitMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.exitMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.exitMenu.Name to "exitMenu"
Create new System.Drawing.Size
Set this.exitMenu.TabIndex to 10
Set this.exitMenu.Text to "EXIT"
Set this.exitMenu.UseVisualStyleBackColor to false
This.exitMenu.Click += new System.EventHandler(this.exitMenu_Click);
materialPropertiesMenu
Set this.materialPropertiesMenu.FlatAppearance.BorderColor to
System.Drawing.Color.White
Set this.materialPropertiesMenu.FlatAppearance.BorderSize to 2
Set this.materialPropertiesMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.materialPropertiesMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.materialPropertiesMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.materialPropertiesMenu.Name to "materialPropertiesMenu"
Create new System.Drawing.Size
Set this.materialPropertiesMenu.TabIndex to 11
Set this.materialPropertiesMenu.Text to "MATERIAL PROPERTIES"
Set this.materialPropertiesMenu.UseVisualStyleBackColor to false
This.materialPropertiesMenu.Click += new
System.EventHandler(this.materialPropertiesMenu_Click);
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Set this.moveMenu.ForeColor to System.Drawing.Color.Transparent
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"

```

```

Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 23
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 25
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 24
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
label1
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 26
Set this.label1.Text to "STRUCTURE"
label2
Set this.label2.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label2.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label2.Name to "label2"
Create new System.Drawing.Size
Set this.label2.TabIndex to 27
Set this.label2.Text to "_____"
label3
Create new System.Drawing.Font
Set this.label3.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label3.Name to "label3"
Create new System.Drawing.Size
Set this.label3.TabIndex to 28
Set this.label3.Text to "LOAD AND SOLVE"
label4
Set this.label4.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label4.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label4.Name to "label4"
Create new System.Drawing.Size
Set this.label4.TabIndex to 29
Set this.label4.Text to "_____"

```

```

label5
Create new System.Drawing.Font
Set this.label5.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label5.Name to "label5"
Create new System.Drawing.Size
Set this.label5.TabIndex to 30
Set this.label5.Text to "MISC"
This.label5.Click += new System.EventHandler(this.label5_Click);
label6
Set this.label6.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label6.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label6.Name to "label6"
Create new System.Drawing.Size
Set this.label6.TabIndex to 31
Set this.label6.Text to " "
newBridge
Set this.newBridge.BackColor to System.Drawing.Color.Transparent
Set this.newBridge.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.newBridge.FlatAppearance.BorderSize to 2
Set this.newBridge.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ), ( ( int ) )
Set this.newBridge.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.newBridge.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.newBridge.Name to "newBridge"
Create new System.Drawing.Size
Set this.newBridge.TabIndex to 47
Set this.newBridge.TabStop to false
Set this.newBridge.Text to "NEW BRIDGE"
Set this.newBridge.UseVisualStyleBackColor to false
This.newBridge.Click += new System.EventHandler(this.newBridge_Click);
loadBridge
Set this.loadBridge.BackColor to System.Drawing.Color.Transparent
Set this.loadBridge.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.loadBridge.FlatAppearance.BorderSize to 2
Set this.loadBridge.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ), ( ( int ) )
Set this.loadBridge.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.loadBridge.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.loadBridge.Name to "loadBridge"
Create new System.Drawing.Size
Set this.loadBridge.TabIndex to 46
Set this.loadBridge.TabStop to false
Set this.loadBridge.Text to "LOAD BRIDGE"
Set this.loadBridge.UseVisualStyleBackColor to false
This.loadBridge.Click += new System.EventHandler(this.loadBridge_Click);
saveBridge
Set this.saveBridge.BackColor to System.Drawing.Color.Transparent
Set this.saveBridge.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.saveBridge.FlatAppearance.BorderSize to 2
Set this.saveBridge.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ), ( ( int ) )
Set this.saveBridge.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.saveBridge.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.saveBridge.Name to "saveBridge"
Create new System.Drawing.Size

```

```

Set this.saveBridge.TabIndex to 45
Set this.saveBridge.TabStop to false
Set this.saveBridge.Text to "SAVE BRIDGE"
Set this.saveBridge.UseVisualStyleBackColor to false
This.saveBridge.Click += new System.EventHandler(this.saveBridge_Click);
savePictureOfBridge
Set this.savePictureOfBridge.BackColor to System.Drawing.Color.Transparent
Set this.savePictureOfBridge.FlatAppearance.BorderColor to
System.Drawing.Color.White
Set this.savePictureOfBridge.FlatAppearance.BorderSize to 2
Set this.savePictureOfBridge.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.savePictureOfBridge.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.savePictureOfBridge.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.savePictureOfBridge.Name to "savePictureOfBridge"
Create new System.Drawing.Size
Set this.savePictureOfBridge.TabIndex to 48
Set this.savePictureOfBridge.TabStop to false
Set this.savePictureOfBridge.Text to "SAVE PICTURE"
Set this.savePictureOfBridge.UseVisualStyleBackColor to false
This.savePictureOfBridge.Click += new
System.EventHandler(this.savePictureOfBridge_Click);
label7
Create new System.Drawing.Font
Set this.label7.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label7.Name to "label7"
Create new System.Drawing.Size
Set this.label7.TabIndex to 49
Set this.label7.Text to "BRIDGE INFO"
label8
Set this.label8.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label8.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label8.Name to "label8"
Create new System.Drawing.Size
Set this.label8.TabIndex to 50
Set this.label8.Text to "_____"
buildBridgeMenu
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.label7
Call method this.Controls.Add with this.label8
Call method this.Controls.Add with this.savePictureOfBridge
Call method this.Controls.Add with this.newBridge
Call method this.Controls.Add with this.loadBridge
Call method this.Controls.Add with this.saveBridge
Call method this.Controls.Add with this.label5
Call method this.Controls.Add with this.label6
Call method this.Controls.Add with this.label3
Call method this.Controls.Add with this.label4
Call method this.Controls.Add with this.label1
Call method this.Controls.Add with this.label2
Call method this.Controls.Add with this.bridgeDrawing
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.materialPropertiesMenu
Call method this.Controls.Add with this.exitMenu
Call method this.Controls.Add with this.aboutMenu

```

```
Call method this.Controls.Add with this.solveMenu
Call method this.Controls.Add with this.loadsMenu
Call method this.Controls.Add with this.supportsMenu
Call method this.Controls.Add with this.membersMenu
Call method this.Controls.Add with this.nodesMenu
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "buildBridgeMenu"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout
```

### Flowchart



```
#endregion  
Attribute: minimize  
  
private System.Windows.Forms.Button minimize;  
Attribute: exitApplication  
  
private System.Windows.Forms.Button exitApplication;  
Attribute: nodesMenu  
  
private System.Windows.Forms.Button nodesMenu;  
Attribute: membersMenu  
  
private System.Windows.Forms.Button membersMenu;  
Attribute: supportsMenu  
  
private System.Windows.Forms.Button supportsMenu;  
Attribute: loadsMenu  
  
private System.Windows.Forms.Button loadsMenu;  
Attribute: solveMenu  
  
private System.Windows.Forms.Button solveMenu;  
Attribute: aboutMenu  
  
private System.Windows.Forms.Button aboutMenu;  
Attribute: exitMenu  
  
private System.Windows.Forms.Button exitMenu;  
Attribute: materialPropertiesMenu  
  
private System.Windows.Forms.Button materialPropertiesMenu;  
Attribute: moveMenu  
  
private System.Windows.Forms.Panel moveMenu;  
Attribute: bridgeDrawing  
  
private System.Windows.Forms.Panel bridgeDrawing;  
Attribute: zoomInBar  
  
private System.Windows.Forms.TrackBar zoomInBar;  
Attribute: label1  
  
private System.Windows.Forms.Label label1;  
Attribute: label2  
  
private System.Windows.Forms.Label label2;
```

*Attribute: label3*

```
private System.Windows.Forms.Label label3;
```

*Attribute: label4*

```
private System.Windows.Forms.Label label4;
```

*Attribute: label5*

```
private System.Windows.Forms.Label label5;
```

*Attribute: label6*

```
private System.Windows.Forms.Label label6;
```

*Attribute: newBridge*

```
private System.Windows.Forms.Button newBridge;
```

*Attribute: loadBridge*

```
private System.Windows.Forms.Button loadBridge;
```

*Attribute: saveBridge*

```
private System.Windows.Forms.Button saveBridge;
```

*Attribute: savePictureOfBridge*

```
private System.Windows.Forms.Button savePictureOfBridge;
```

*Attribute: label7*

```
private System.Windows.Forms.Label label7;
```

*Attribute: label8*

```
private System.Windows.Forms.Label label8;
```

**Code File: confirmNewBridgeMenu.cs**

```
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: confirmNewBridgeMenu**

```
public partial class confirmNewBridgeMenu : Form
```

*Attribute: cGrip*

```
private const int cGrip = 16;
```

*Attribute: cCaption*

```
private const int cCaption = 32;
```

*Procedure: Constructor*

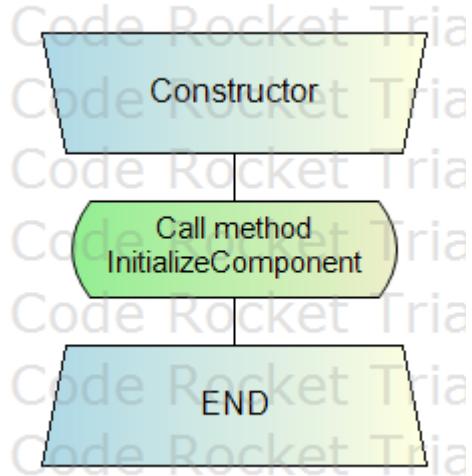
```
public confirmNewBridgeMenu()
```

Returns confirmNewBridgeMenu

**Pseudocode**

```
Call method InitializeComponent
```

Flowchart



**Procedure: WndProc**

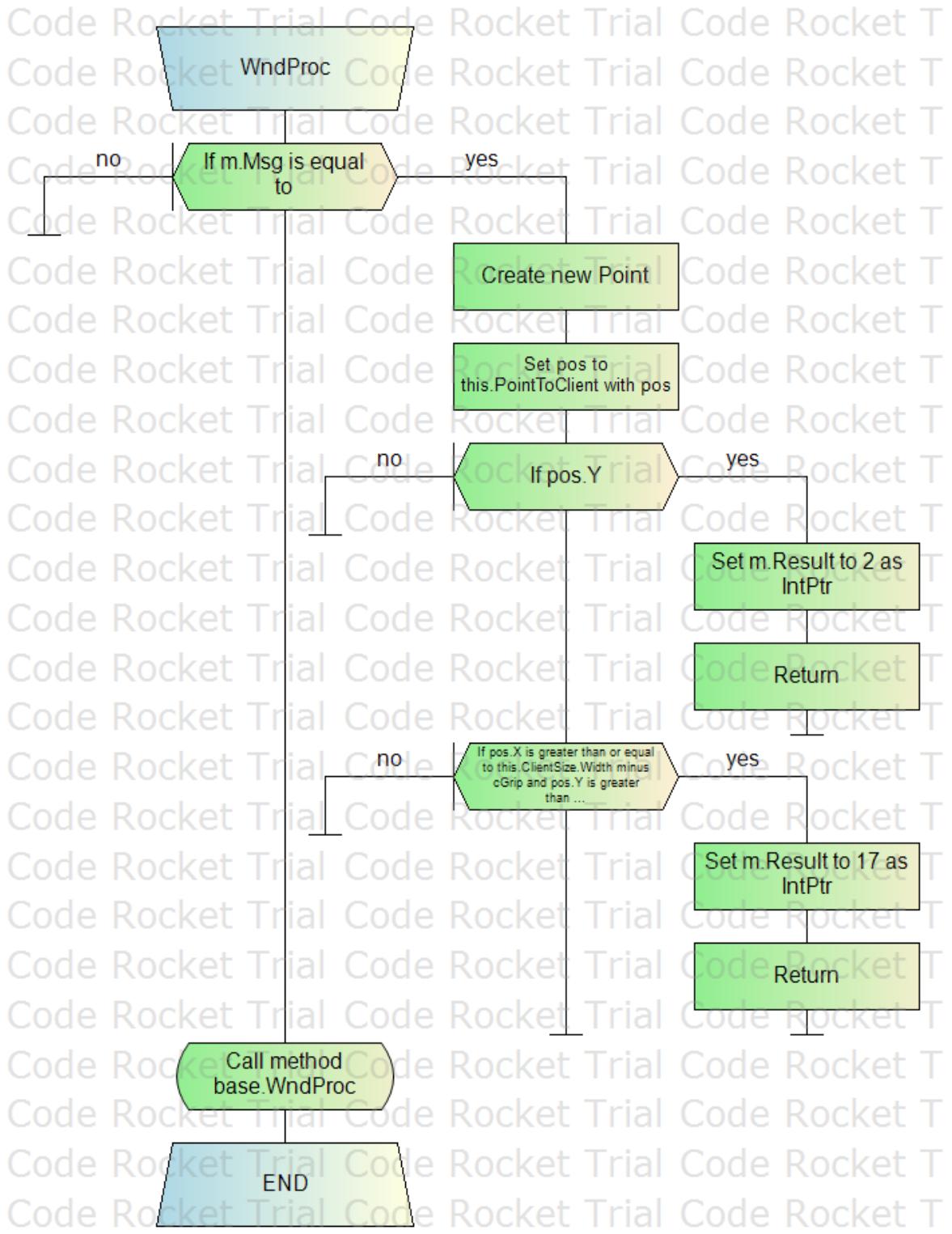
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

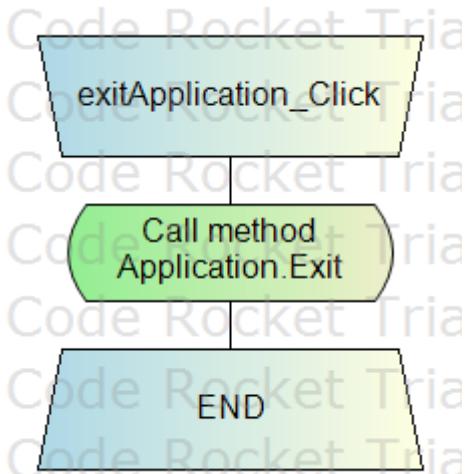
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

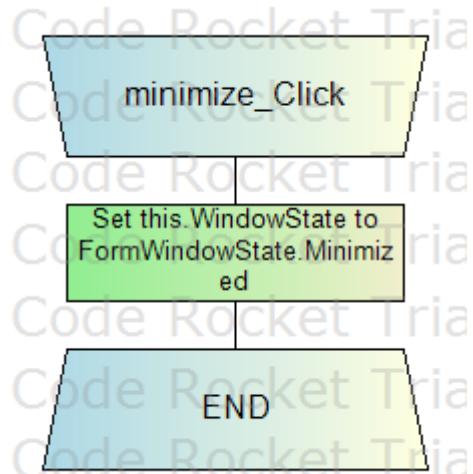
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

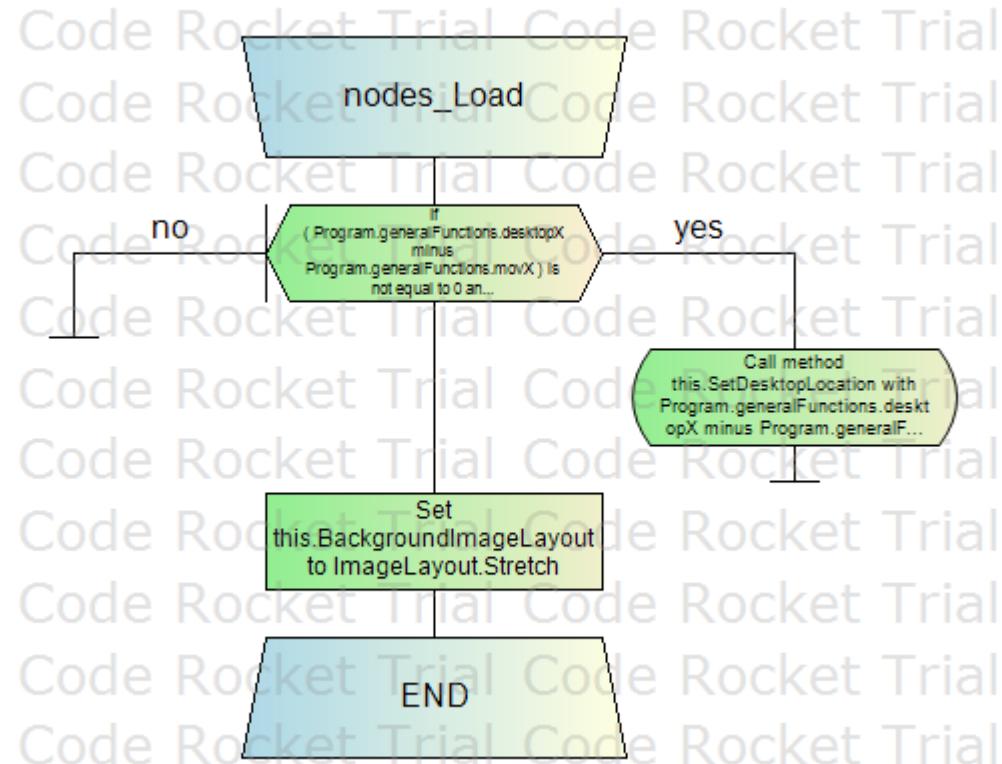
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: exitConfirm\_Click**

```
private void exitConfirm_Click(object sender, EventArgs e)
```

**Parameters**

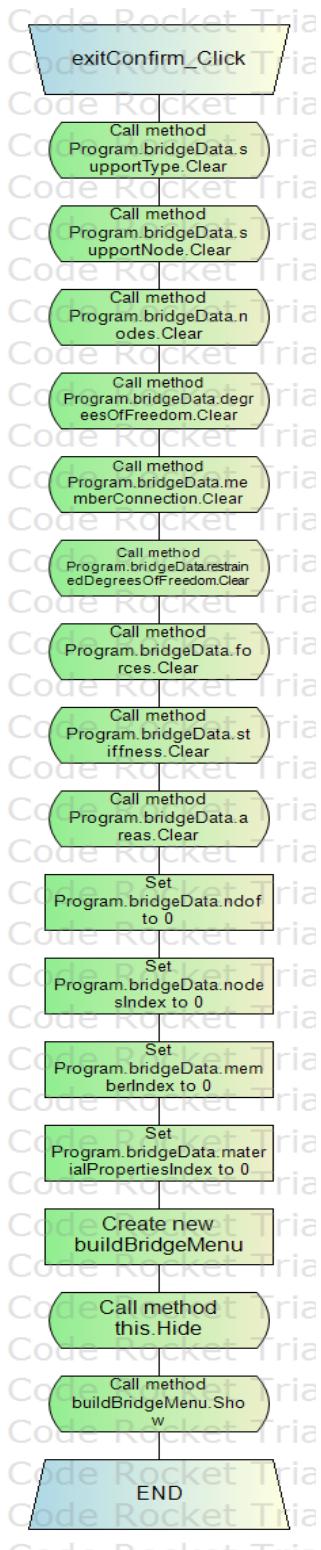
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Program.bridgeData.supportType.Clear  
Call method Program.bridgeData.supportNode.Clear  
Call method Program.bridgeData.nodes.Clear  
Call method Program.bridgeData.degreesOfFreedom.Clear  
Call method Program.bridgeData.memberConnection.Clear  
Call method Program.bridgeData.restrainedDegreesOfFreedom.Clear  
Call method Program.bridgeData.forces.Clear  
Call method Program.bridgeData.stiffness.Clear  
Call method Program.bridgeData.areas.Clear  
Set Program.bridgeData.ndof to 0  
Set Program.bridgeData.nodesIndex to 0  
Set Program.bridgeData.memberIndex to 0  
Set Program.bridgeData.materialPropertiesIndex to 0  
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



**Procedure: exitDecline\_Click**

```
private void exitDecline_Click(object sender, EventArgs e)
```

**Parameters**

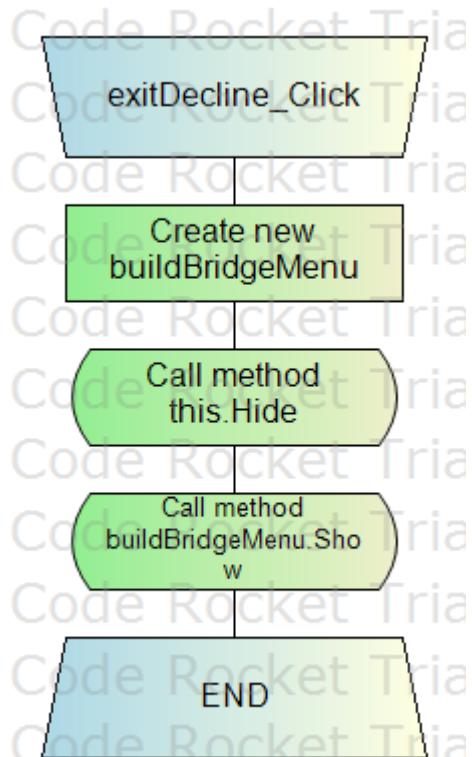
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

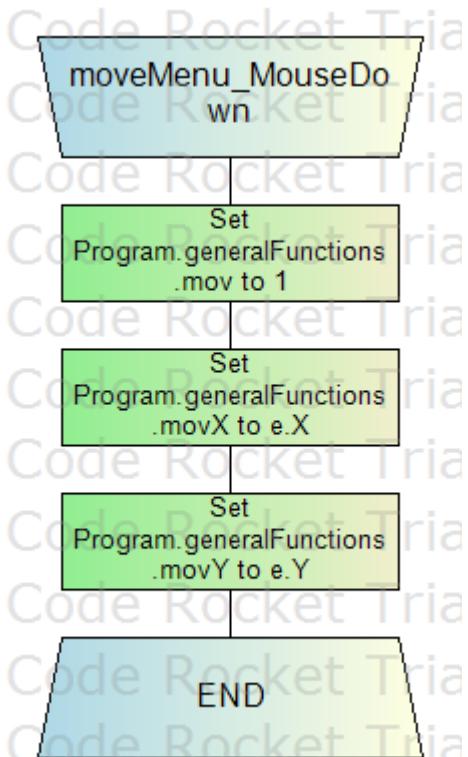
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

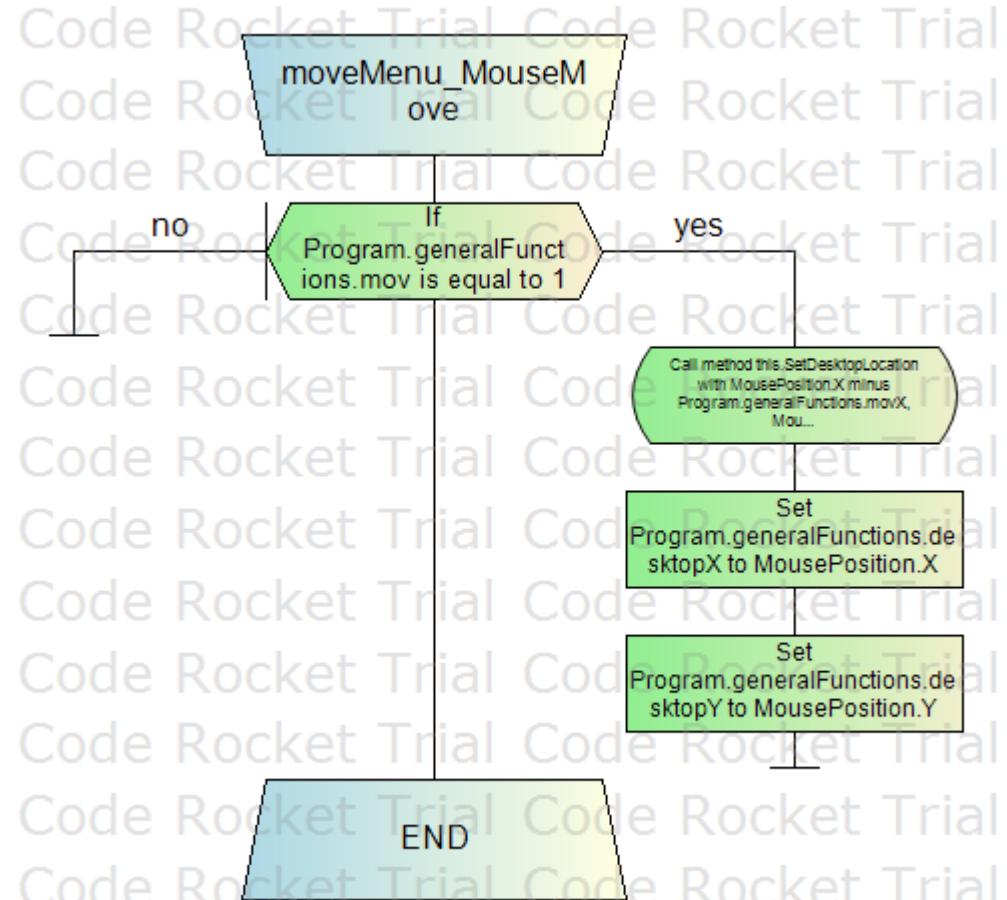
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



*Procedure: moveMenu\_MouseUp*

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

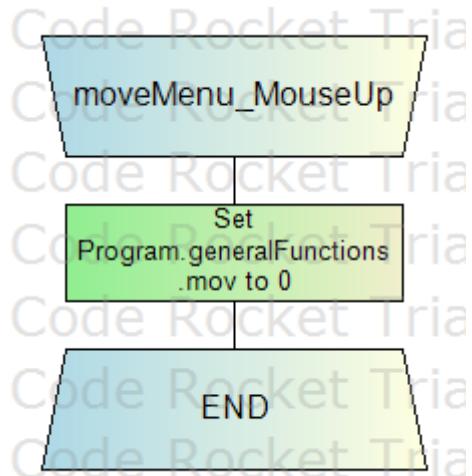
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



*Code File: confirmNewBridgeMenu.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

*Class: confirmNewBridgeMenu*

partial class confirmNewBridgeMenu

*Attribute: components*

private System.ComponentModel.IContainer components = null;

Required designer variable.

*Procedure: Dispose*

protected override void Dispose (bool disposing)

Clean up any resources being used.

*Parameters*

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

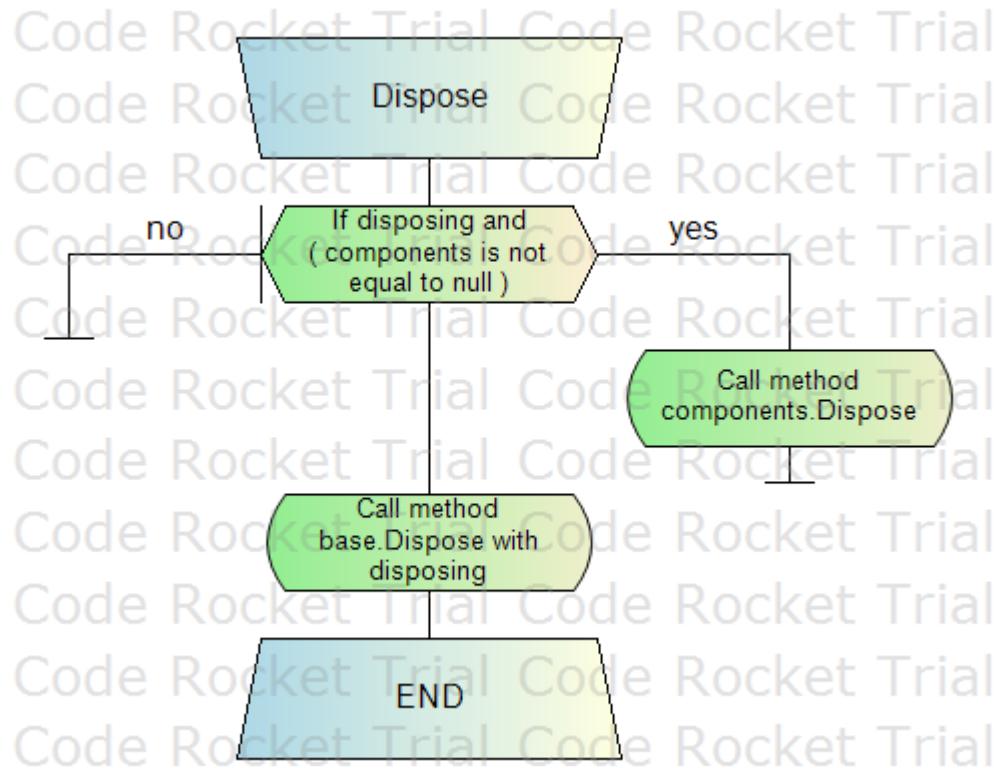
*Pseudocode*

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

*Flowchart*



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Panel
Call method this.SuspendLayout
minimize
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
exitDecline
Set this.exitDecline.BackColor to System.Drawing.Color.Transparent
Set this.exitDecline.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.exitDecline.FlatAppearance.BorderSize to 2
Set this.exitDecline.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.exitDecline.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.exitDecline.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.exitDecline.Name to "exitDecline"
Create new System.Drawing.Size
Set this.exitDecline.TabIndex to 17
Set this.exitDecline.TabStop to false
Set this.exitDecline.Text to "NO"
Set this.exitDecline.UseVisualStyleBackColor to false
This.exitDecline.Click += new System.EventHandler(this.exitDecline_Click);
exitConfirm
Set this.exitConfirm.BackColor to System.Drawing.Color.Transparent

```

```

Set this.exitConfirm.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.exitConfirm.FlatAppearance.BorderSize to 2
Set this.exitConfirm.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ) ), ( ( int ) ) )
Set this.exitConfirm.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.exitConfirm.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.exitConfirm.Name to "exitConfirm"
Create new System.Drawing.Size
Set this.exitConfirm.TabIndex to 16
Set this.exitConfirm.TabStop to false
Set this.exitConfirm.Text to "YES"
Set this.exitConfirm.UseVisualStyleBackColor to false
This.exitConfirm.Click += new System.EventHandler(this.exitConfirm_Click);
label1
Set this.label1.AutoSize to true
Set this.label1.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 15
Set this.label1.Text to "ARE YOU SURE YOU WANT TO START OVER?"
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Set this.moveMenu.ForeColor to System.Drawing.Color.Transparent
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 24
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
confirmNewBridgeMenu
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ), ( ( int ) ),
( ( int ) ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.exitDecline
Call method this.Controls.Add with this.exitConfirm
Call method this.Controls.Add with this.label1
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "confirmNewBridgeMenu"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
Call method this.ResumeLayout with false
Call method this.PerformLayout

```

## Flowchart



```
#endregion
```

*Attribute: minimize*

```
private System.Windows.Forms.Button minimize;
```

*Attribute: exitApplication*

```
private System.Windows.Forms.Button exitApplication;
```

*Attribute: exitDecline*

```
private System.Windows.Forms.Button exitDecline;
```

*Attribute: exitConfirm*

```
private System.Windows.Forms.Button exitConfirm;
```

*Attribute: label1*

```
private System.Windows.Forms.Label label1;
```

*Attribute: moveMenu*

```
private System.Windows.Forms.Panel moveMenu;
```

**Code File: exitMenus.cs**

```
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: exitMenus**

```
public partial class exitMenus : Form
```

*Attribute: cGrip*

```
private const int cGrip = 16;
```

*Attribute: cCaption*

```
private const int cCaption = 32;
```

**Procedure: Constructor**

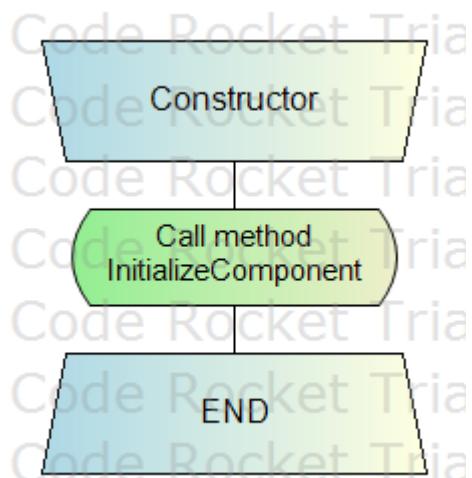
```
public exitMenus()
```

Returns exitMenus

**Pseudocode**

Call method InitializeComponent

Flowchart



**Procedure: WndProc**

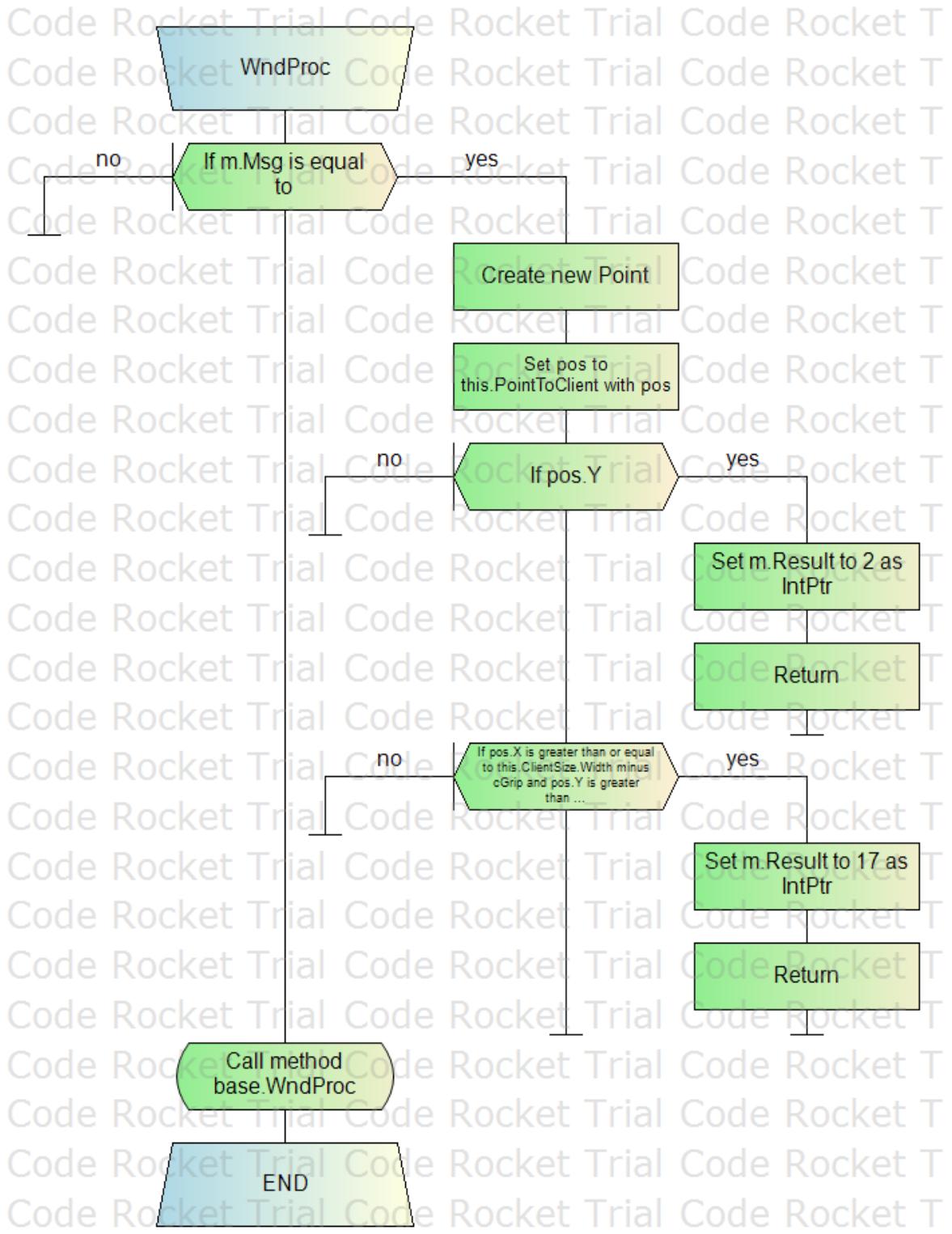
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

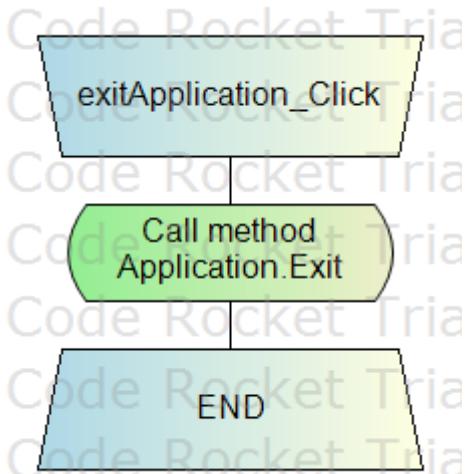
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



**Procedure: minimize\_Click**

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

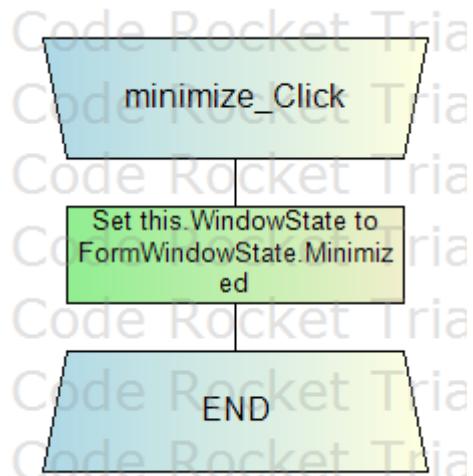
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

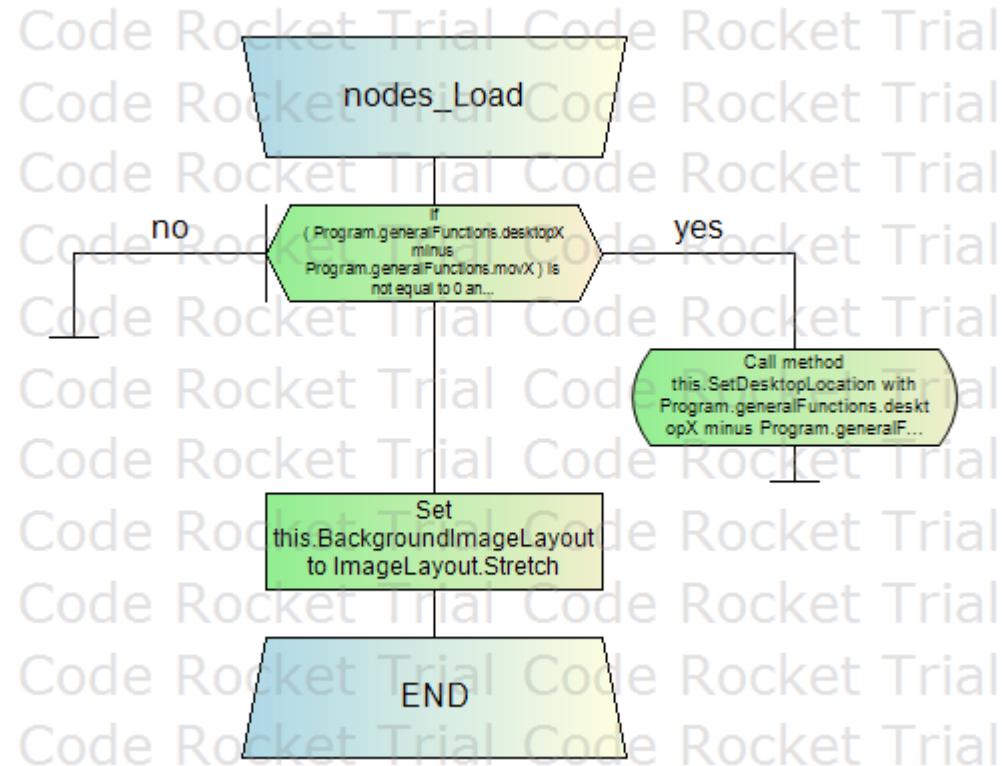
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: exitConfirm\_Click**

```
private void exitConfirm_Click(object sender, EventArgs e)
```

**Parameters**

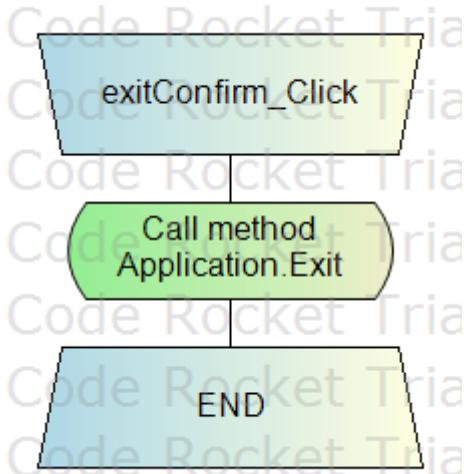
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



**Procedure: exitDecline\_Click**

```
private void exitDecline_Click(object sender, EventArgs e)
```

**Parameters**

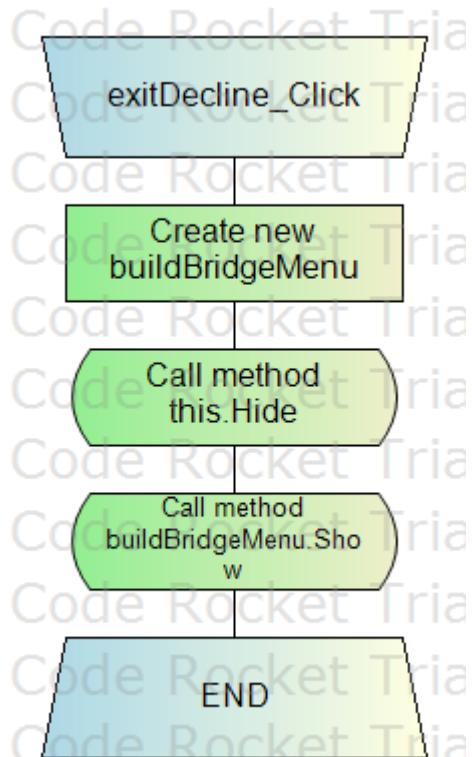
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

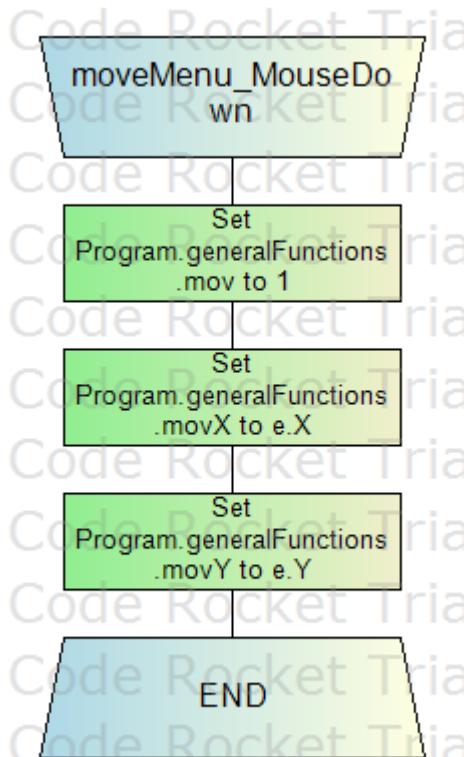
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

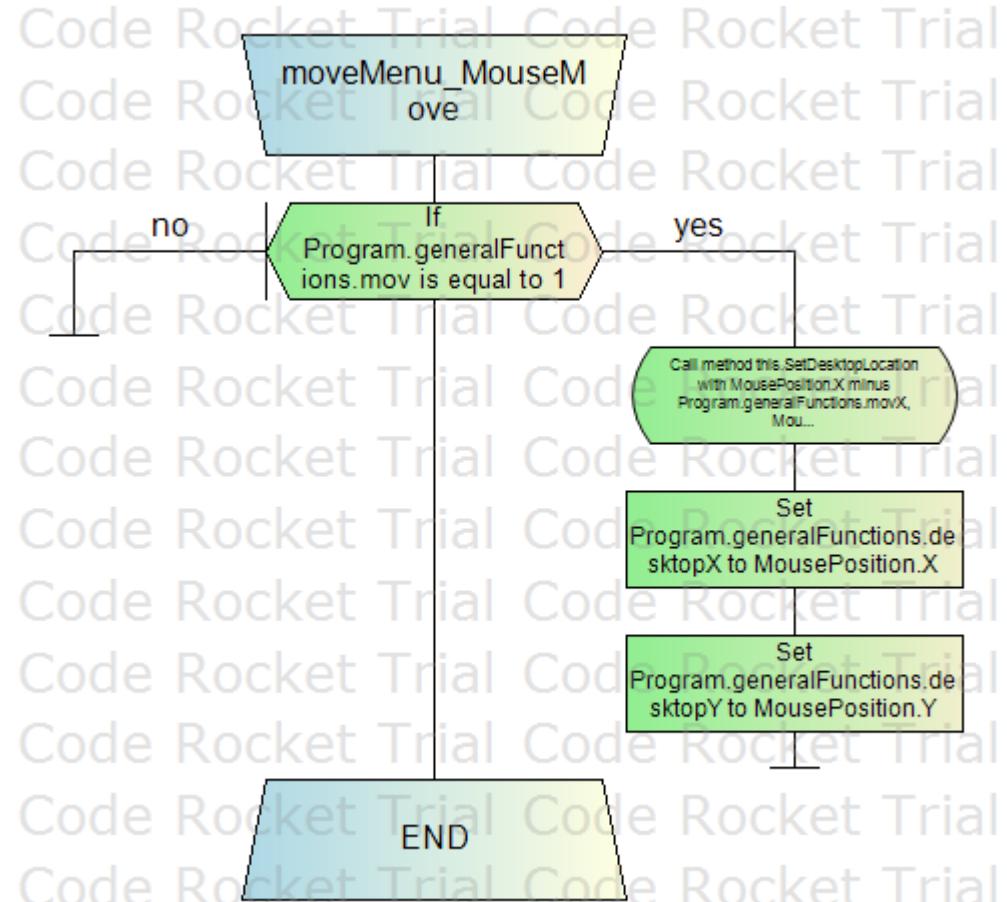
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



*Procedure: moveMenu\_MouseUp*

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

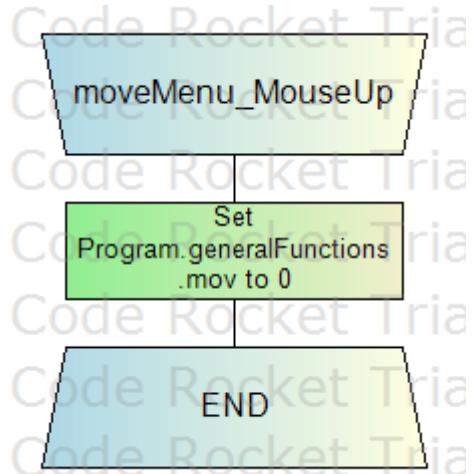
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Code File:** exitMenus.Designer.cs

**Namespace:** Sectrics\_V2

namespace Sectrics\_V2

**Class:** exitMenus

partial class exitMenus

**Attribute:** components

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure:** Dispose

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

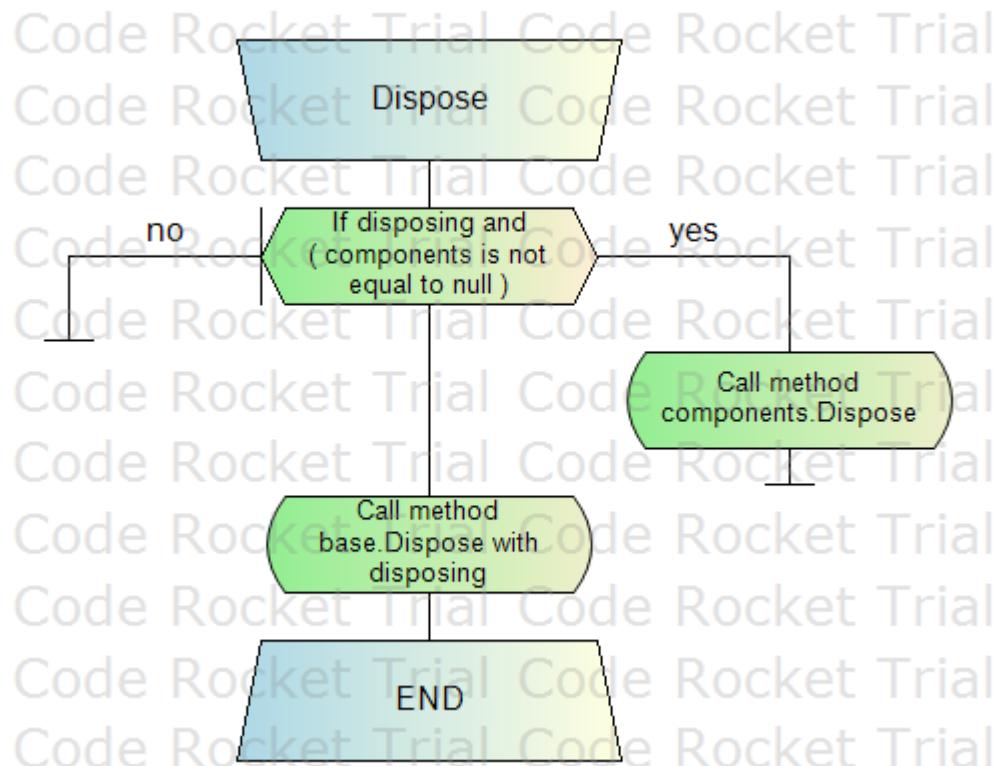
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

Flowchart



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Call method this.SuspendLayout
minimize
Set this.minimize.BackColor to System.Drawing.Color.Transparent
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.TabStop to false
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackColor to System.Drawing.Color.Transparent
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.TabStop to false
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
label1
Set this.label1.AutoSize to true
Set this.label1.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 12
Set this.label1.Text to "ARE YOU SURE YOU WANT TO QUIT?"
exitConfirm
Set this.exitConfirm.BackColor to System.Drawing.Color.Transparent
Set this.exitConfirm.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.exitConfirm.FlatAppearance.BorderSize to 2
Set this.exitConfirm.FlatAppearance.MouseOverBackColor to

```

```

System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.exitConfirm.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.exitConfirm.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.exitConfirm.Name to "exitConfirm"
Create new System.Drawing.Size
Set this.exitConfirm.TabIndex to 13
Set this.exitConfirm.TabStop to false
Set this.exitConfirm.Text to "YES"
Set this.exitConfirm.UseVisualStyleBackColor to false
This.exitConfirm.Click += new System.EventHandler(this.exitConfirm_Click);
exitDecline
Set this.exitDecline.BackColor to System.Drawing.Color.Transparent
Set this.exitDecline.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.exitDecline.FlatAppearance.BorderSize to 2
Set this.exitDecline.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.exitDecline.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.exitDecline.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.exitDecline.Name to "exitDecline"
Create new System.Drawing.Size
Set this.exitDecline.TabIndex to 14
Set this.exitDecline.TabStop to false
Set this.exitDecline.Text to "NO"
Set this.exitDecline.UseVisualStyleBackColor to false
This.exitDecline.Click += new System.EventHandler(this.exitDecline_Click);
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 40
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
exitMenus
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.exitDecline
Call method this.Controls.Add with this.exitConfirm
Call method this.Controls.Add with this.label1
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.DoubleBuffered to true
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "exitMenus"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
Call method this.ResumeLayout with false
Call method this.PerformLayout

```

## Flowchart



```
#endregion
```

*Attribute: minimize*

```
private System.Windows.Forms.Button minimize;
```

*Attribute: exitApplication*

```
private System.Windows.Forms.Button exitApplication;
```

*Attribute: label1*

```
private System.Windows.Forms.Label label1;
```

*Attribute: exitConfirm*

```
private System.Windows.Forms.Button exitConfirm;
```

*Attribute: exitDecline*

```
private System.Windows.Forms.Button exitDecline;
```

*Attribute: moveMenu*

```
private System.Windows.Forms.Panel moveMenu;
```

**Code File:** generalFunctions.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

**Namespace:** Sectrics\_V2

```
namespace Sectrics_V2
```

**Class:** generalFunctions

```
class generalFunctions
```

*Attribute: mov*

```
public int mov;
```

*Attribute: movX*

```
public int movX;
```

*Attribute: movY*

```
public int movY;
```

*Attribute: desktopX*

```
public int desktopX;
```

**Attribute: desktopY**

```
public int desktopY;
```

**Procedure: saveFileCSV**

```
public void saveFileCSV()
```

Returns Void

**Pseudocode**

**Try**

```
    Create new SaveFileDialog
    Set saveBridge.DefaultExt to "csv"
    Set saveBridge.Filter to "CSV|*.csv"
    If saveBridge.ShowDialog is equal to DialogResult.OK
        If File.Exists with saveBridge.FileName
            Call method File.Delete with saveBridge.FileName
        EndIf
        using (Stream s = File.Open(saveBridge.FileName, FileMode.CreateNew))
            using (StreamWriter sw = new StreamWriter(s))
                Set of
                For i is 0, i is less than Program.bridgeData.nodes.Count, i
increments by 1
                    Set i of nodes to Program.bridgeData.nodes
                    Set i of nodes to Program.bridgeData.nodes
                EndFor
                Set of
                For i is 0, i is less than
Program.bridgeData.degreesOfFreedom.Count, i increments by 1
                    Set i of degreesOfFreedom to
Program.bridgeData.degreesOfFreedom
                    Set i of degreesOfFreedom to
Program.bridgeData.degreesOfFreedom
                EndFor
                Set of
                For i is 0, i is less than
Program.bridgeData.memberConnection.Count, i increments by 1
                    Set i of memberConnections to
Program.bridgeData.memberConnection
                    Set i of memberConnections to
Program.bridgeData.memberConnection
                EndFor
                Set of
                For i is 0, i is less than Program.bridgeData.forces.Count,
i increments by 1
                    Set i of forces to Program.bridgeData.forces
                    Set i of forces to Program.bridgeData.forces
                EndFor
                Create new int
                For i is 0, i is less than
Program.bridgeData.supportNode.Count, i increments by 1
                    Set i of supportNode to Program.bridgeData.supportNode
                EndFor
                Create new string
                For i is 0, i is less than
Program.bridgeData.supportType.Count, i increments by 1
                    Set i of supportType to Program.bridgeData.supportType
                EndFor
                Initialise restrainedDegreesOfFreedom to
Program.bridgeData.restrainedDegreesOfFreedom.ToArray
                Initialise stiffness to Program.bridgeData.stiffness.ToArray
                Initialise areas to Program.bridgeData.areas.ToArray
                Initialise ndof to Program.bridgeData.ndof
                Initialise nodeIndex to Program.bridgeData.nodesIndex
                Initialise memberIndex to Program.bridgeData.memberIndex
```

```

        Initialise materialPropertyIndex to
Program.bridgeData.materialPropertiesIndex
            Finds the maximum length array
            If nodes.GetLength with 0 is greater than maxLength
                Set maxLength to nodes.GetLength with 0
            EndIf
            If degreesOfFreedom.GetLength with 0 is greater than
maxLength
                Set maxLength to degreesOfFreedom.GetLength with 0
            EndIf
            If memberConnections.GetLength with 0 is greater than
maxLength
                Set maxLength to memberConnections.GetLength with 0
            EndIf
            If forces.GetLength with 0 is greater than maxLength
                Set maxLength to forces.GetLength with 0
            EndIf
            If restrainedDegreesOfFreedom.Length is greater than
maxLength
                Set maxLength to restrainedDegreesOfFreedom.Length
            EndIf
            If stiffness.Length is greater than maxLength
                Set maxLength to stiffness.Length
            EndIf
            If areas.Length is greater than maxLength
                Set maxLength to areas.Length
            EndIf
            If supportNode.Length is greater than maxLength
                Set maxLength to supportNode.Length
            EndIf
            If supportType.Length is greater than maxLength
                Set maxLength to supportType.Length
            EndIf
            Writes Into CSV
            For i is 0, i is less than maxLength, i increments by 1
                If i
                    Call method sw.Write with nodes plus "," plus
nodes plus ","
                Else
                    Call method sw.Write with ","
                EndIf
                If i
                    Call method sw.Write with degreesOfFreedom plus
"," plus degreesOfFreedom plus ","
                Else
                    Call method sw.Write with ","
                EndIf
                If i
                    Call method sw.Write with memberConnections plus
"," plus memberConnections plus ","
                Else
                    Call method sw.Write with ","
                EndIf
                If i
                    Call method sw.Write with forces plus "," plus
forces plus ","
                Else
                    Call method sw.Write with ","
                EndIf
                If i
                    Call method sw.Write with
restrainedDegreesOfFreedom plus ","
                Else
                    Call method sw.Write with ","

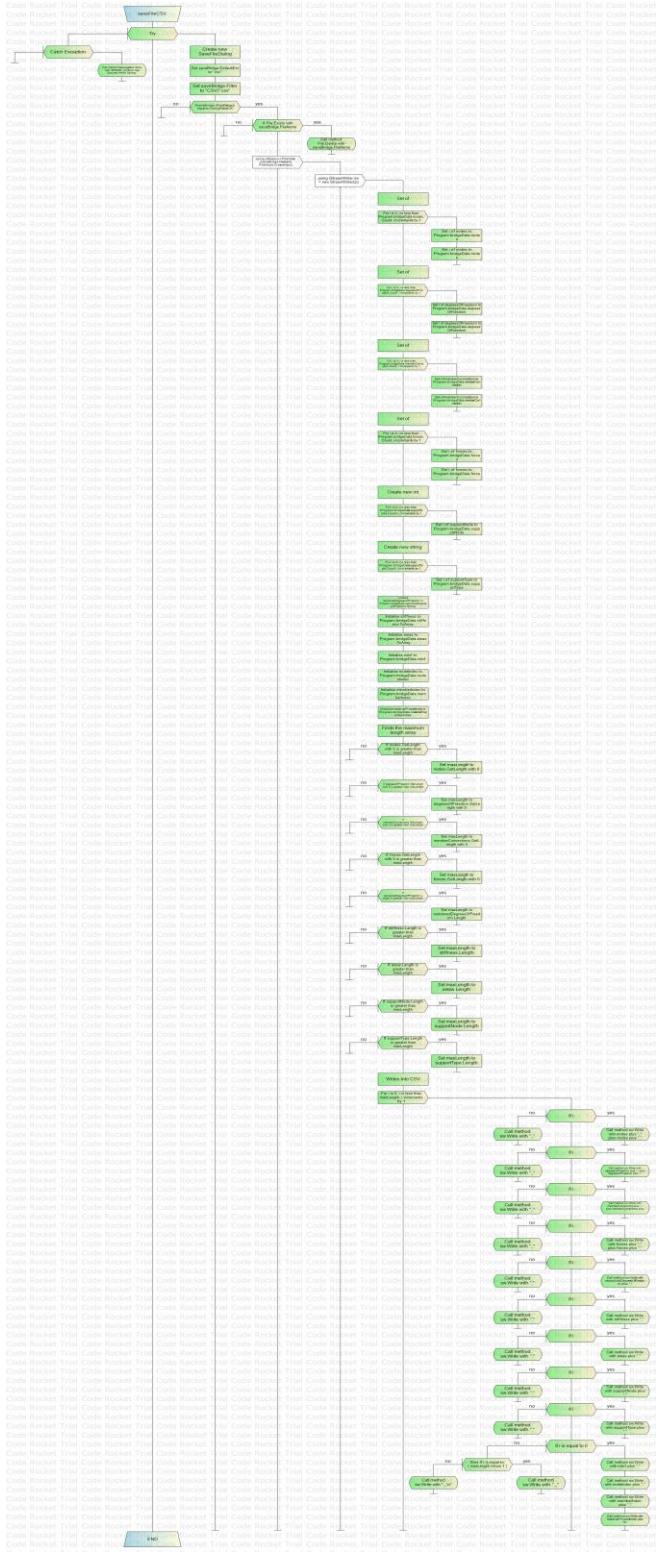
```

```

EndIf
If i
    Call method sw.Write with stiffness plus ","
Else
    Call method sw.Write with ","
EndIf
If i
    Call method sw.Write with areas plus ","
Else
    Call method sw.Write with ","
EndIf
If i
    Call method sw.Write with supportNode plus ","
Else
    Call method sw.Write with ","
EndIf
If i
    Call method sw.Write with supportType plus ","
Else
    Call method sw.Write with ","
EndIf
If i is equal to 0
    Call method sw.Write with ndof plus ","
    Call method sw.Write with nodeIndex plus ","
    Call method sw.Write with memberIndex plus ","
    Call method sw.Write with materialPropertyIndex
plus "\n"
Else if i is equal to ( maxLength minus 1 )
    Call method sw.Write with ",,,"
Else
    Call method sw.Write with ",,, \n"
EndIf
EndFor
EndStatement
EndStatement
EndIf
Catch Exception
    Call method MessageBox.Show with "ERROR: An Error Has Occured Whilst Saving "
EndTry

```

Flowchart



**Procedure: LoadFileCSV**

```
public void LoadFileCSV()
```

Returns Void

**Pseudocode**

**Try**

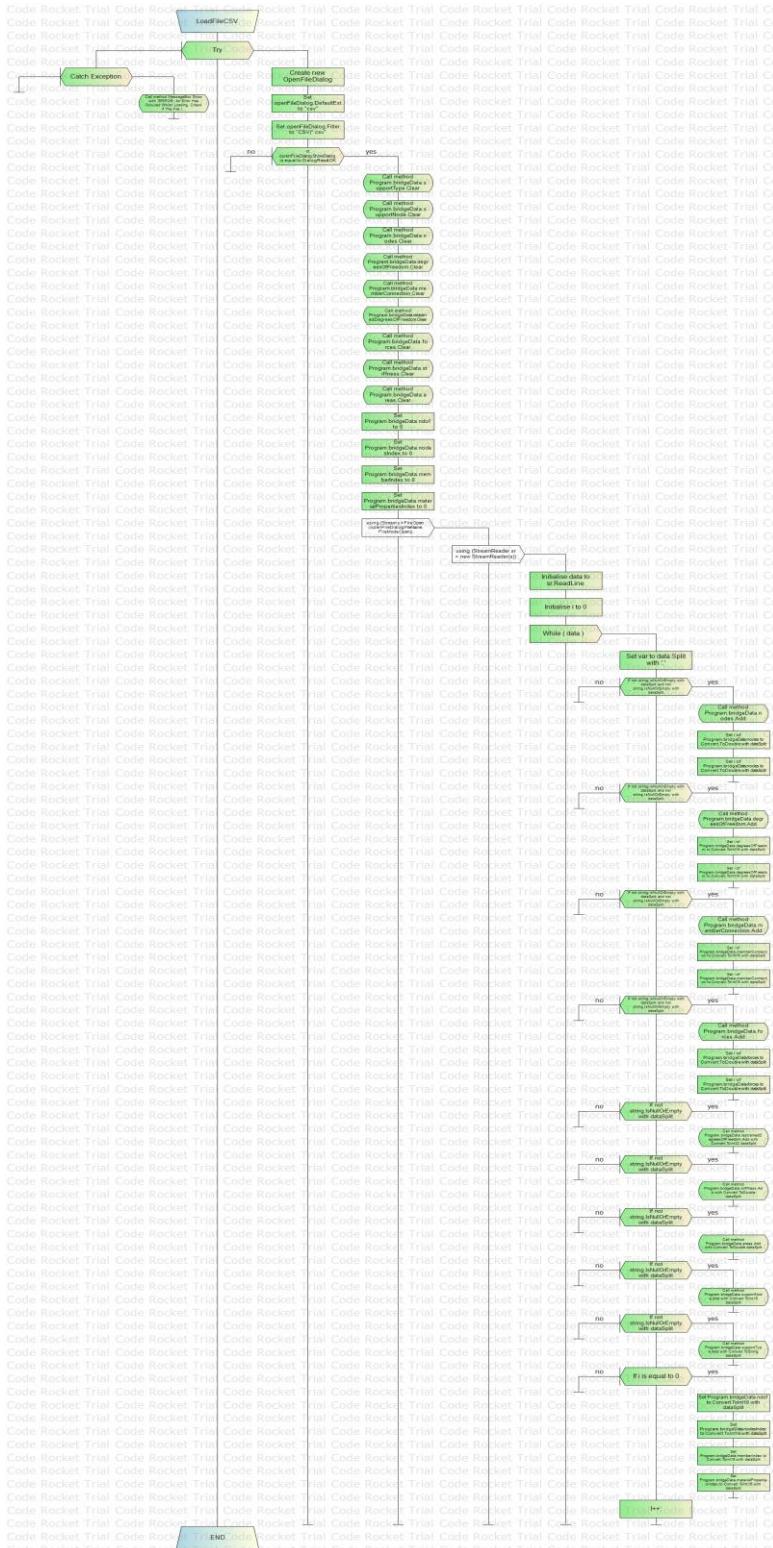
```
    Create new OpenFileDialog
    Set openFileDialog.DefaultExt to "csv"
    Set openFileDialog.Filter to "CSV|*.csv"
    If openFileDialog.ShowDialog is equal to DialogResult.OK
        Call method Program.bridgeData.supportType.Clear
        Call method Program.bridgeData.supportNode.Clear
        Call method Program.bridgeData.nodes.Clear
        Call method Program.bridgeData.degreesOfFreedom.Clear
        Call method Program.bridgeData.memberConnection.Clear
        Call method Program.bridgeData.restrainedDegreesOfFreedom.Clear
        Call method Program.bridgeData.forces.Clear
        Call method Program.bridgeData.stiffness.Clear
        Call method Program.bridgeData.areas.Clear
        Set Program.bridgeData.ndof to 0
        Set Program.bridgeData.nodesIndex to 0
        Set Program.bridgeData.memberIndex to 0
        Set Program.bridgeData.materialPropertiesIndex to 0
        using (Stream s = File.Open(openFileDialog.FileName, FileMode.Open))
            using (StreamReader sr = new StreamReader(s))
                Initialise data to sr.ReadLine
                Initialise i to 0
                While ( data )
                    Set var to data.Split with ','
                    If not string.IsNullOrEmpty with dataSplit and not
string.IsNullOrEmpty with dataSplit
                        Call method Program.bridgeData.nodes.Add
                        Set i of Program.bridgeData.nodes to
Convert.ToDouble with dataSplit
                        Set i of Program.bridgeData.nodes to
Convert.ToDouble with dataSplit
                    EndIf
                    If not string.IsNullOrEmpty with dataSplit and not
string.IsNullOrEmpty with dataSplit
                        Call method
Program.bridgeData.degreesOfFreedom.Add
                        Set i of Program.bridgeData.degreesOfFreedom to
Convert.ToInt16 with dataSplit
                        Set i of Program.bridgeData.degreesOfFreedom to
Convert.ToInt16 with dataSplit
                    EndIf
                    If not string.IsNullOrEmpty with dataSplit and not
string.IsNullOrEmpty with dataSplit
                        Call method
Program.bridgeData.memberConnection.Add
                        Set i of Program.bridgeData.memberConnection to
Convert.ToInt16 with dataSplit
                        Set i of Program.bridgeData.memberConnection to
Convert.ToInt16 with dataSplit
                    EndIf
                    If not string.IsNullOrEmpty with dataSplit and not
string.IsNullOrEmpty with dataSplit
                        Call method Program.bridgeData.forces.Add
                        Set i of Program.bridgeData.forces to
Convert.ToDouble with dataSplit
                        Set i of Program.bridgeData.forces to
Convert.ToDouble with dataSplit
                    EndIf
                    If not string.IsNullOrEmpty with dataSplit
```

```

        Call method
Program.bridgeData.restrainedDegreesOfFreedom.Add with Convert.ToInt32 dataSplit
        EndIf
        If not string.IsNullOrEmpty with dataSplit
            Call method Program.bridgeData.stiffness.Add
with Convert.ToDouble dataSplit
        EndIf
        If not string.IsNullOrEmpty with dataSplit
            Call method Program.bridgeData.areas.Add with
Convert.ToDouble dataSplit
        EndIf
        If not string.IsNullOrEmpty with dataSplit
            Call method Program.bridgeData.supportNode.Add
with Convert.ToInt16 dataSplit
        EndIf
        If not string.IsNullOrEmpty with dataSplit
            Call method Program.bridgeData.supportType.Add
with Convert.ToString dataSplit
        EndIf
        If i is equal to 0
            Set Program.bridgeData.ndof to Convert.ToInt16
with dataSplit
            Set Program.bridgeData.nodesIndex to
Convert.ToInt16 with dataSplit
            Set Program.bridgeData.memberIndex to
Convert.ToInt16 with dataSplit
            Set Program.bridgeData.materialPropertiesIndex
to Convert.ToInt16 with dataSplit
        EndIf
        I++;
    EndWhile
EndStatement
EndStatement
EndIf
Catch Exception
    Call method MessageBox.Show with "ERROR: An Error Has Occured Whilst Loading.
Check If The File Is Already In Use Or If The Data Is Valid & Try Again."
EndTry

```

## Flowchart



**Code File: loadsMenus.cs**

```
using Sectrics_V2.Properties;
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: loadsMenus**

```
public partial class loadsMenus : Form
```

**Attribute: zoom**

```
float zoom = 1f;
```

**Attribute: xMouseOffset**

```
double xMouseOffset;
```

**Attribute: yMouseOffset**

```
double yMouseOffset;
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

**Procedure: Constructor**

```
public loadsMenus()
```

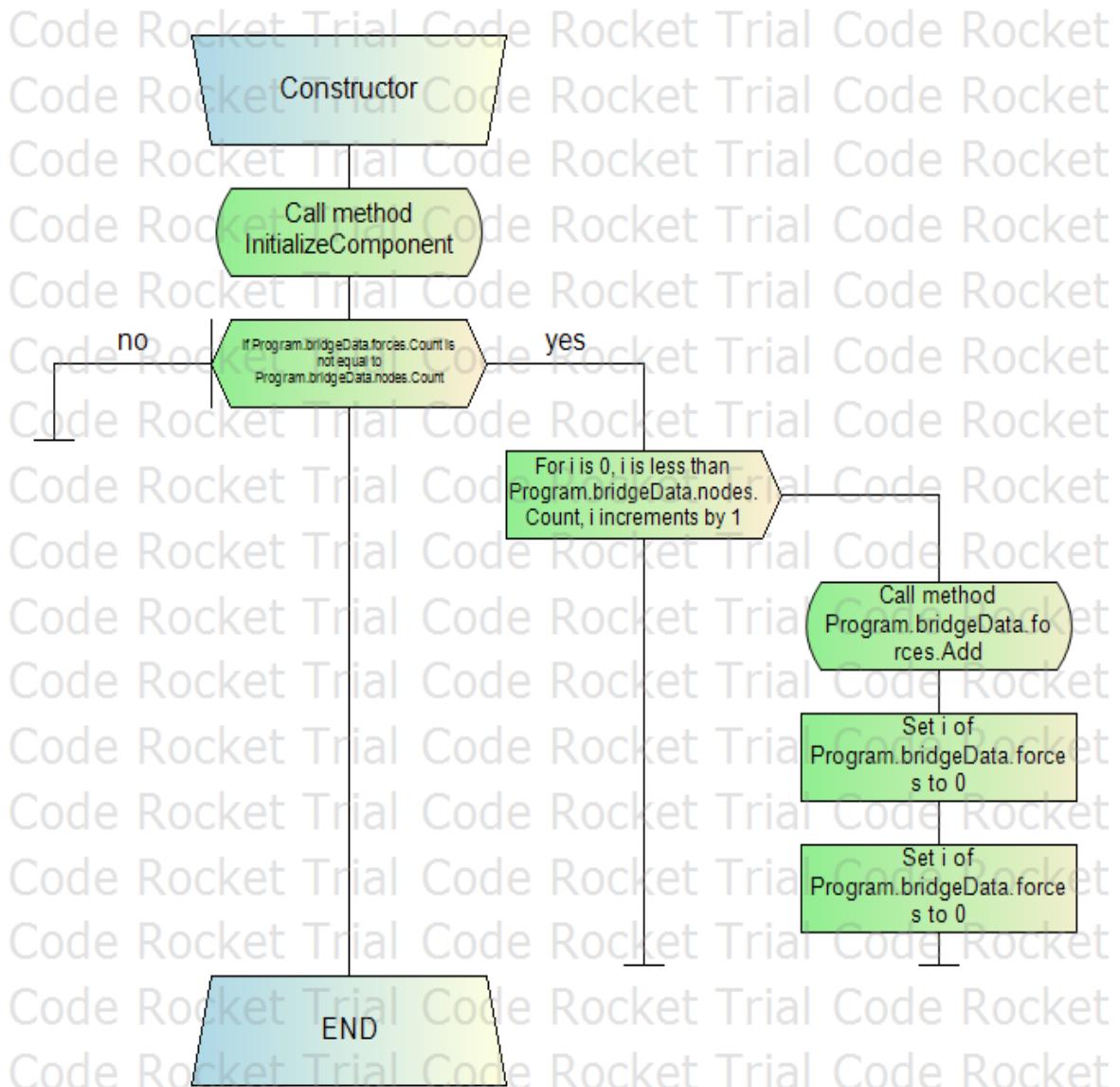
Returns loadsMenus

**Pseudocode**

```
Call method InitializeComponent
```

```
If Program.bridgeData.forces.Count is not equal to Program.bridgeData.nodes.Count
    For i is 0, i is less than Program.bridgeData.nodes.Count, i increments by 1
        Call method Program.bridgeData.forces.Add
        Set i of Program.bridgeData.forces to 0
        Set i of Program.bridgeData.forces to 0
    EndFor
EndIf
```

**Flowchart**



**Procedure: WndProc**

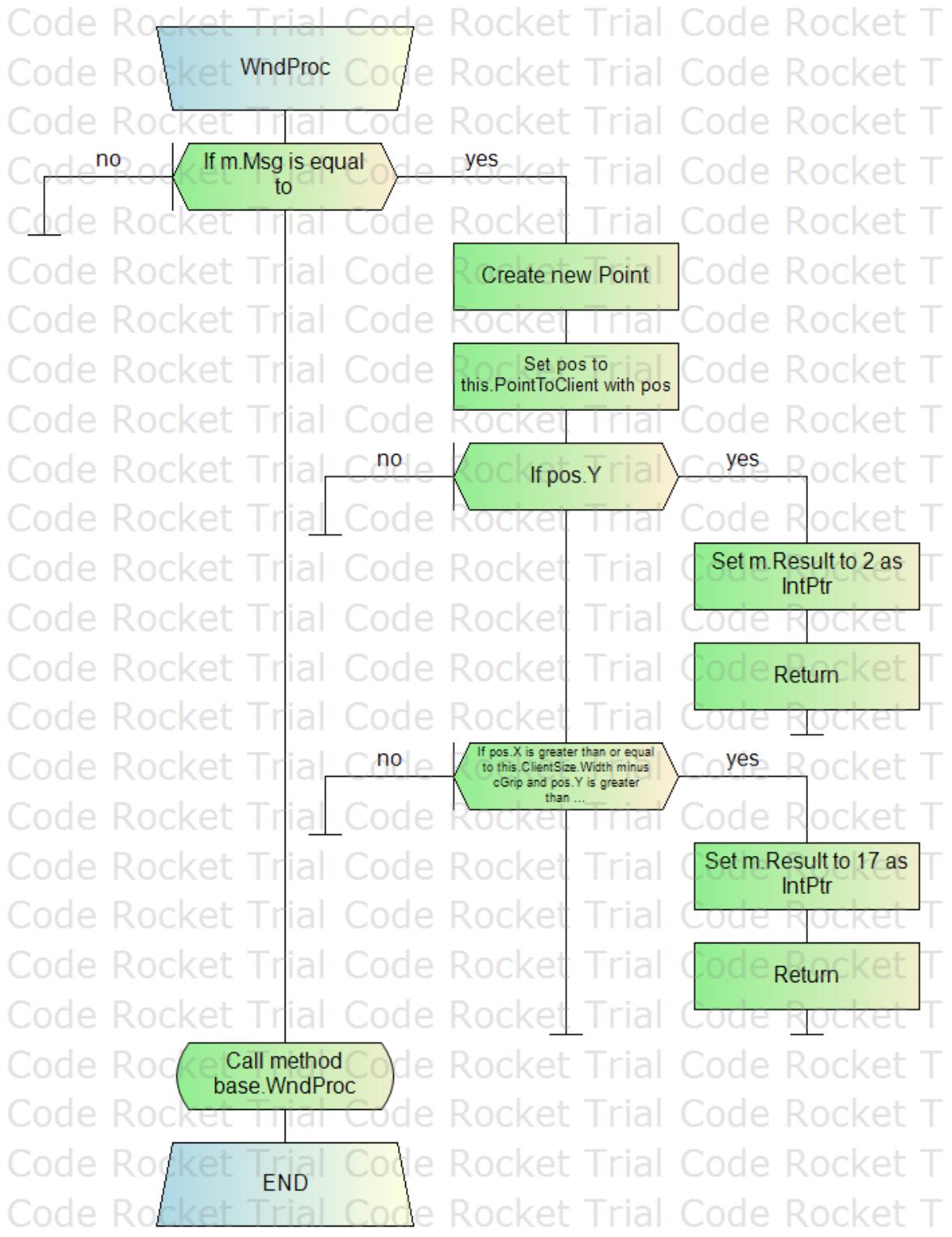
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

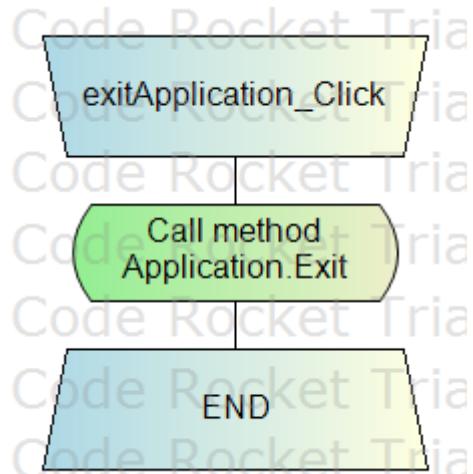
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

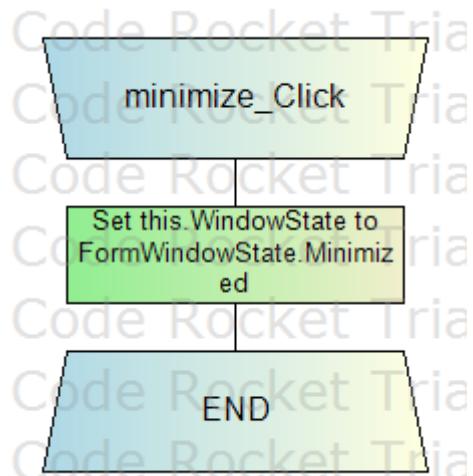
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

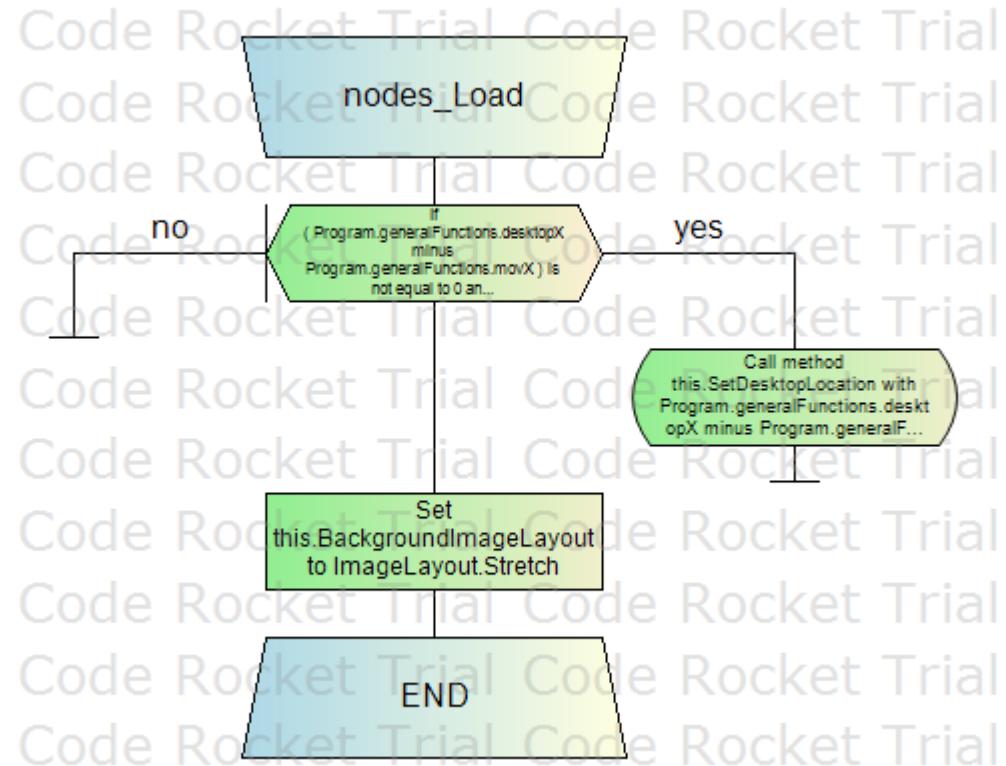
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: yForceTextbox\_TextChanged**

```
private void yForceTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

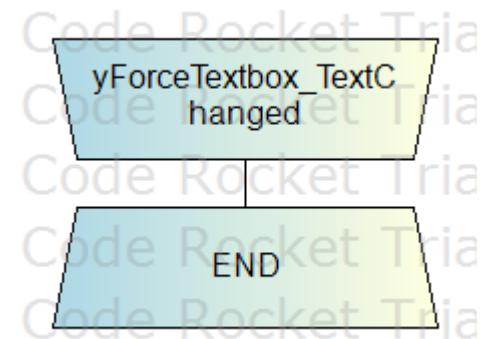
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: xForceTextbox\_TextChanged**

```
private void xForceTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

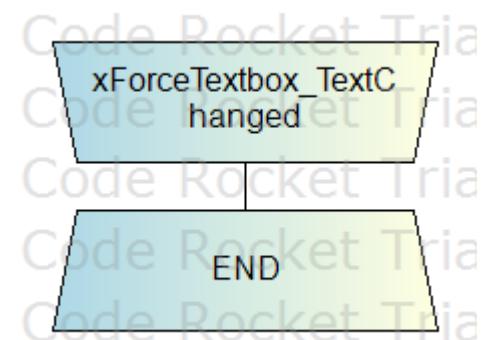
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: nodeTextbox\_TextChanged**

```
private void nodeTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

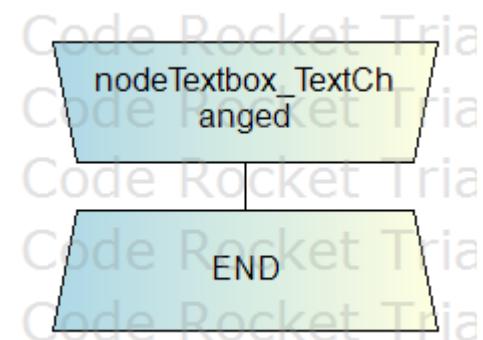
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: addLoadButton\_Click**

```
private void addLoadButton_Click(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

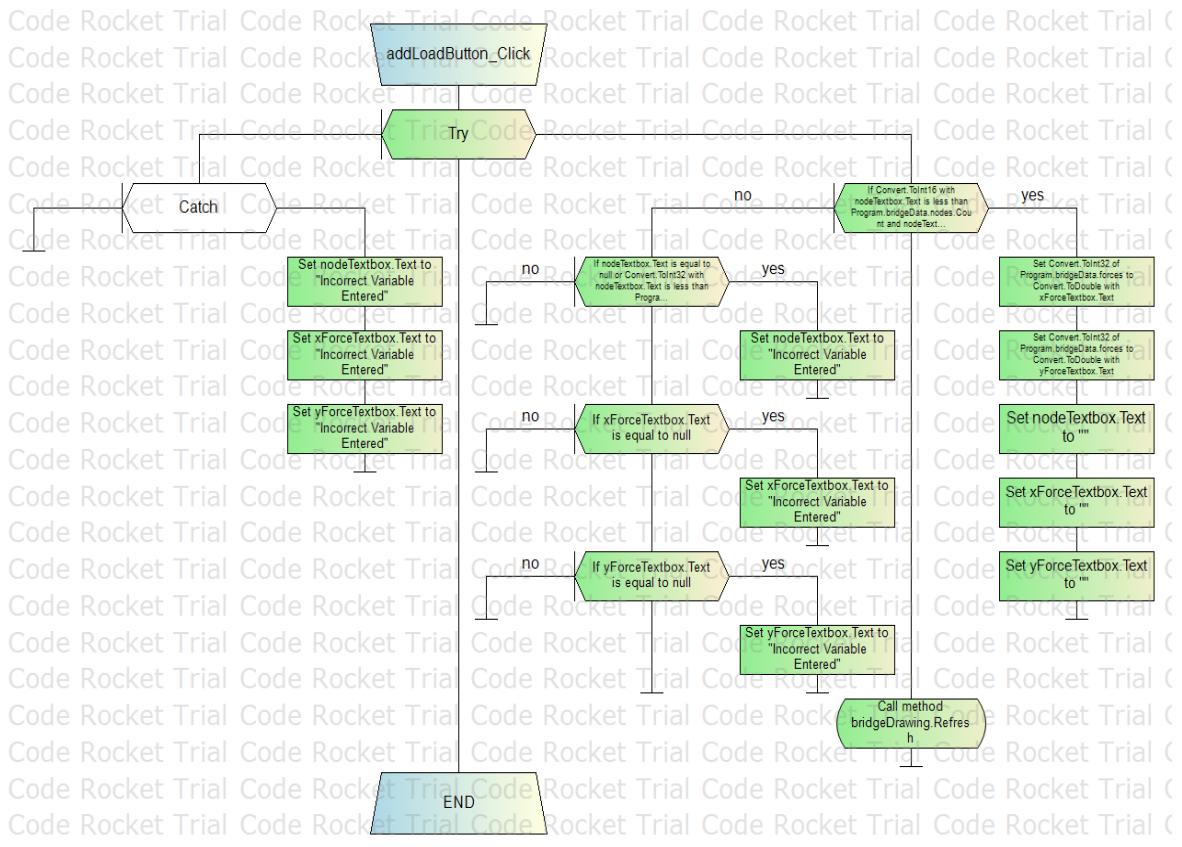
Returns Void

**Pseudocode**

**Try**

```
    If Convert.ToInt16 with nodeTextbox.Text is less than
    Program.bridgeData.nodes.Count and nodeTextbox.Text is not equal to null and
    xForceTextbox.Text is not equal to null and yForceTextbox.Text is not equal to null
        Set Convert.ToInt32 of Program.bridgeData.forces to Convert.ToDouble
    with xForceTextbox.Text
        Set Convert.ToInt32 of Program.bridgeData.forces to Convert.ToDouble
    with yForceTextbox.Text
        Set nodeTextbox.Text to ""
        Set xForceTextbox.Text to ""
        Set yForceTextbox.Text to ""
    Else
        If nodeTextbox.Text is equal to null or Convert.ToInt32 with
    nodeTextbox.Text is less than Program.bridgeData.nodes.Count
            Set nodeTextbox.Text to "Incorrect Variable Entered"
        EndIf
        If xForceTextbox.Text is equal to null
            Set xForceTextbox.Text to "Incorrect Variable Entered"
        EndIf
        If yForceTextbox.Text is equal to null
            Set yForceTextbox.Text to "Incorrect Variable Entered"
        EndIf
    EndIf
    Call method bridgeDrawing.Refresh
Catch
    Set nodeTextbox.Text to "Incorrect Variable Entered"
    Set xForceTextbox.Text to "Incorrect Variable Entered"
    Set yForceTextbox.Text to "Incorrect Variable Entered"
EndTry
```

**Flowchart**



*Procedure: nodeListView\_SelectedIndexChanged*

```
private void nodeListView_SelectedIndexChanged(object sender,  
EventArgs e)
```

Parameters

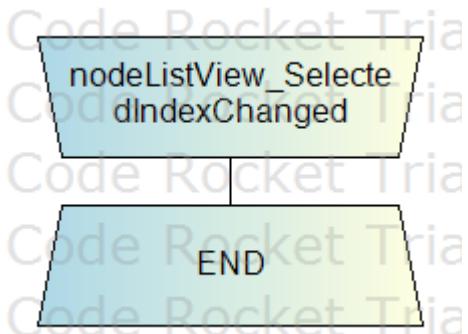
- sender - Object
- e - EventArgs

Returns Void

Pseudocode

N/A

Flowchart



*Procedure: loadsListView\_SelectedIndexChanged*

```
private void loadsListView_SelectedIndexChanged(object sender,  
EventArgs e)
```

Parameters

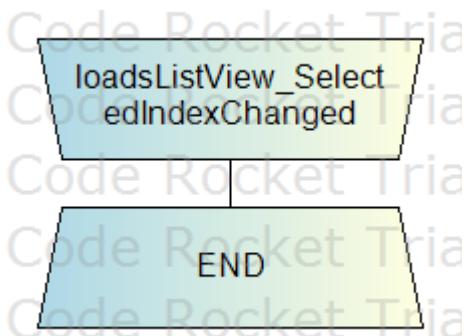
- sender - Object
- e - EventArgs

Returns Void

Pseudocode

N/A

Flowchart



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

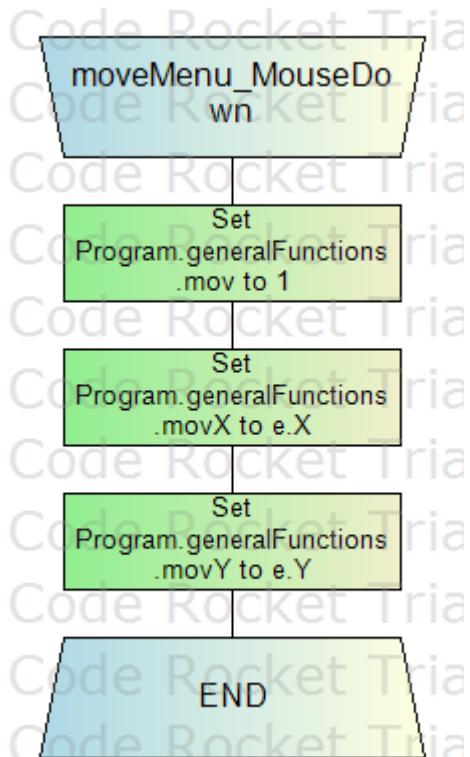
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

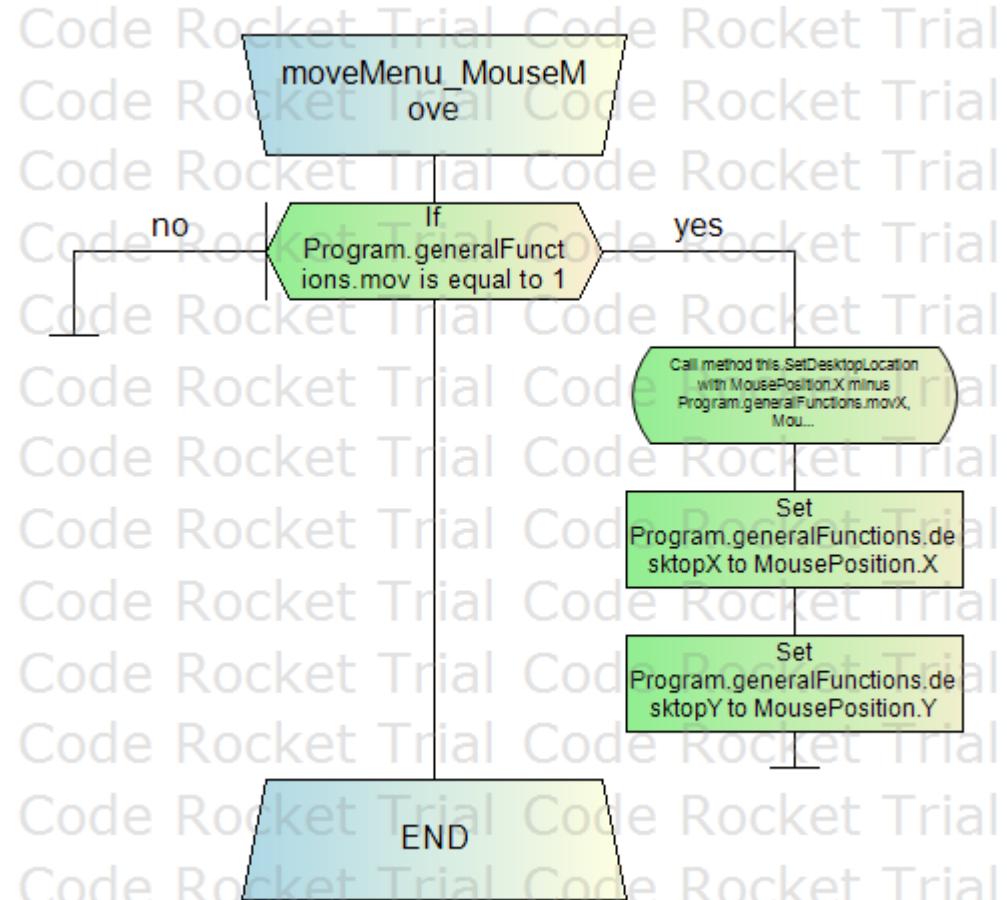
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseUp**

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

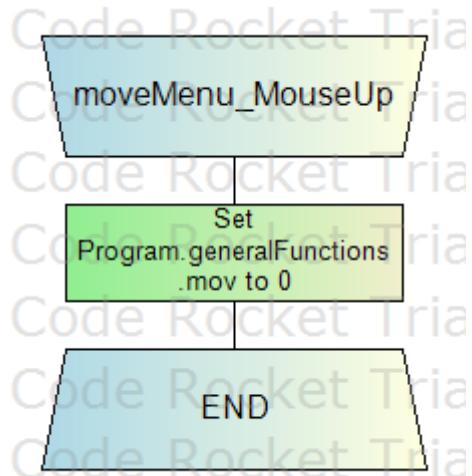
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

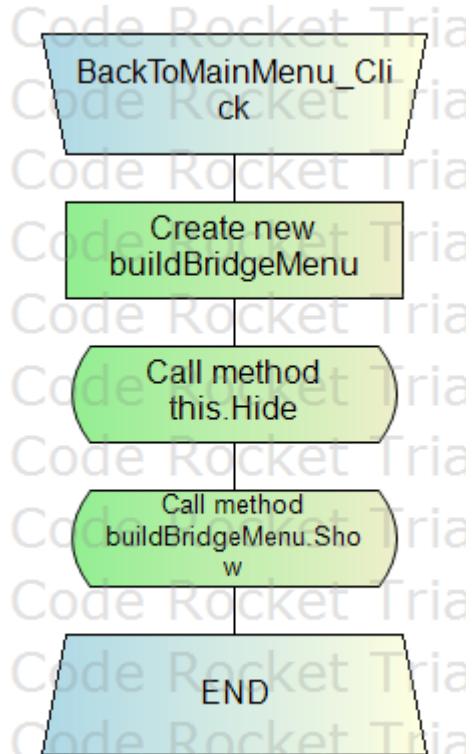
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

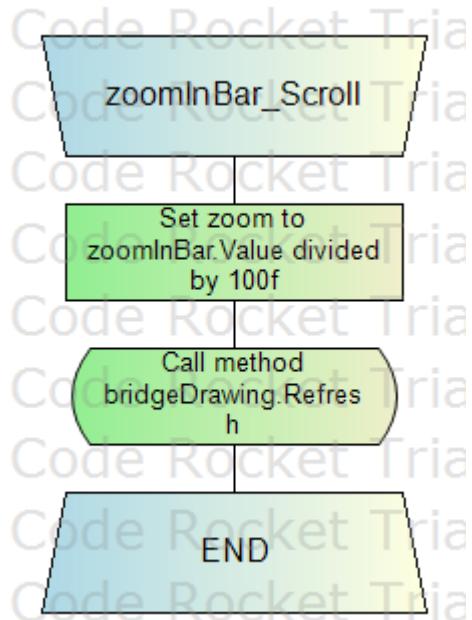
Returns Void

**Pseudocode**

Set zoom to zoomInBar.Value divided by 100f

Call method bridgeDrawing.Refresh

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

**Try**

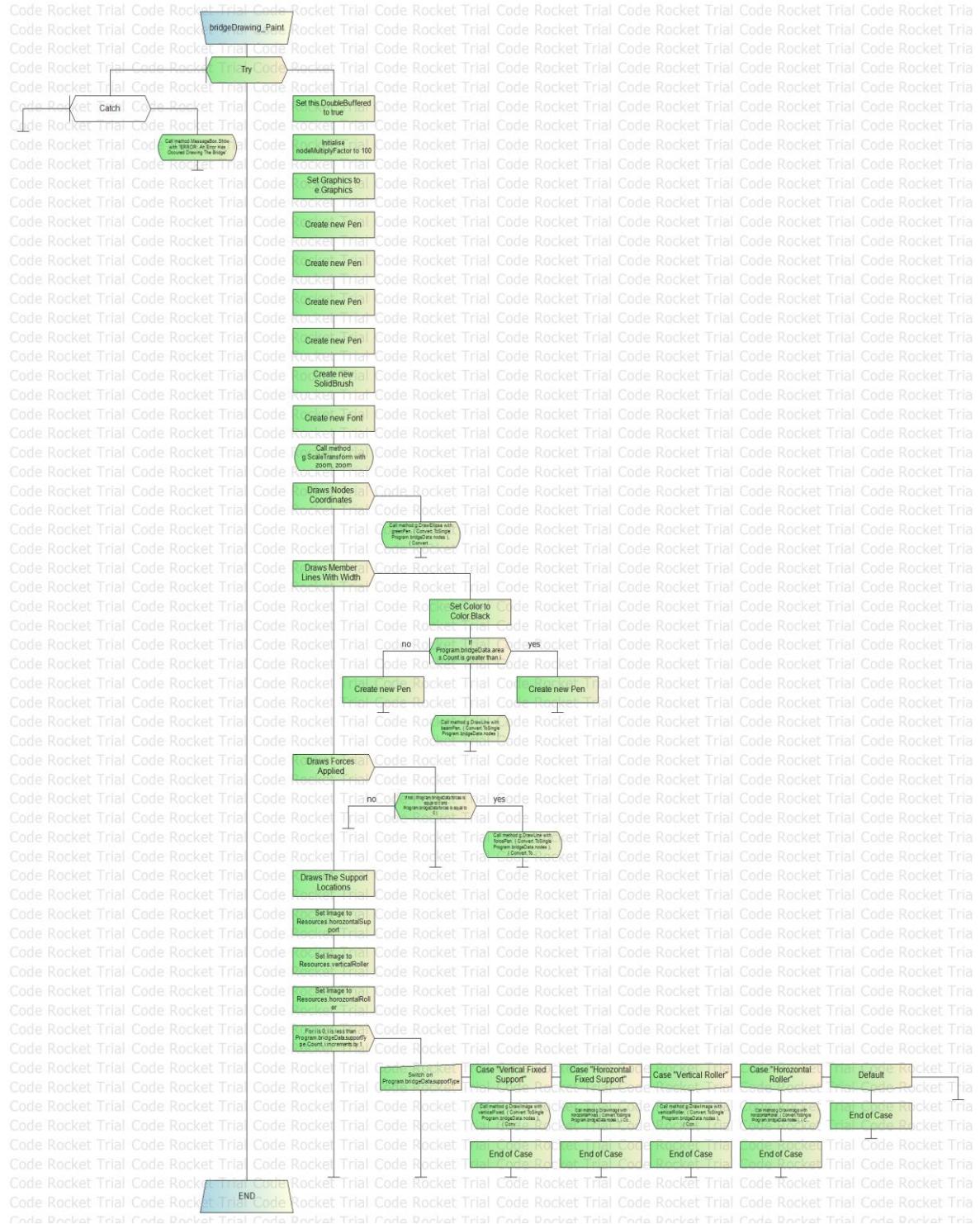
```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```

        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry

```

### Flowchart



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

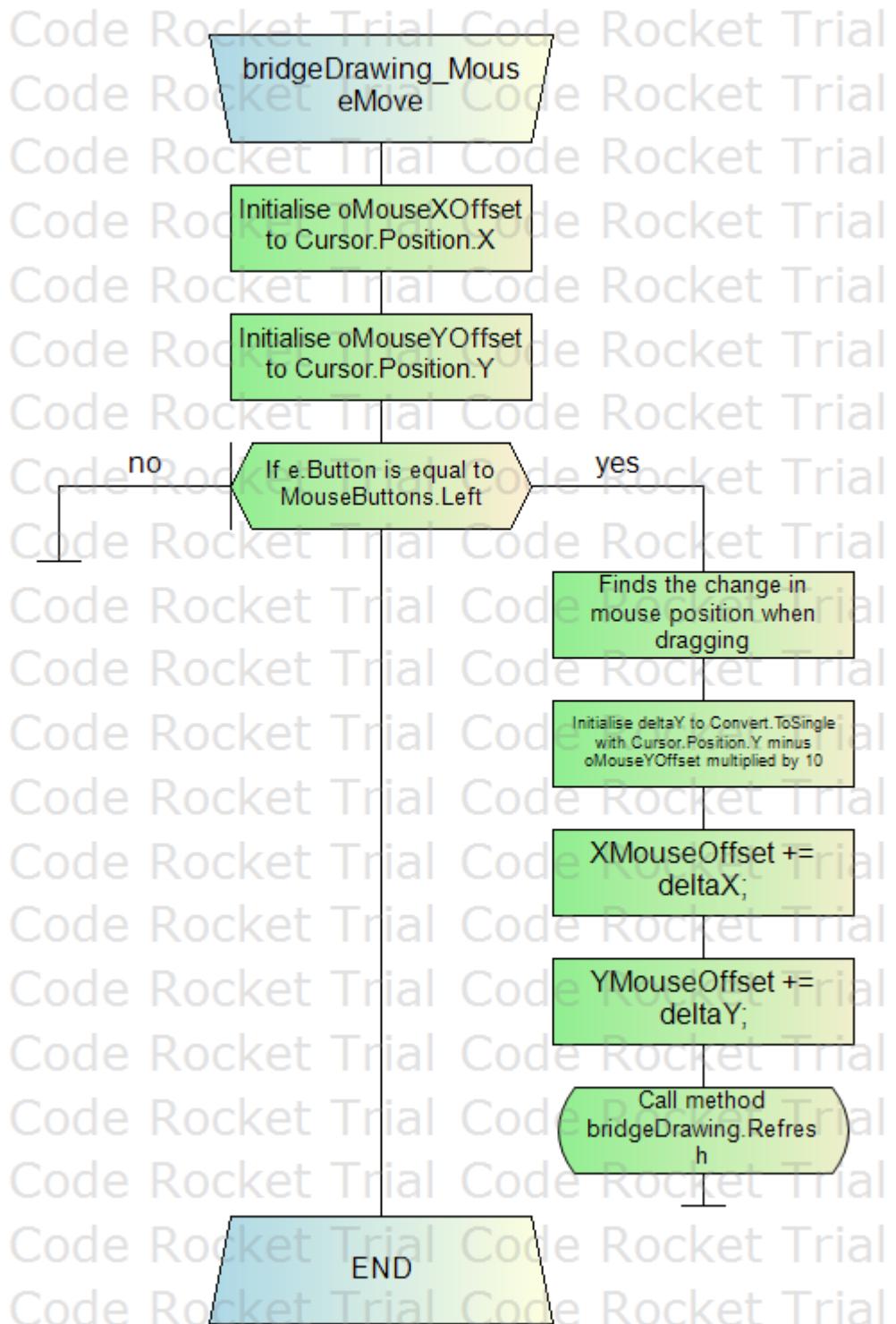
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



*Procedure: label3\_Click*

```
private void label3_Click(object sender, EventArgs e)
```

**Parameters**

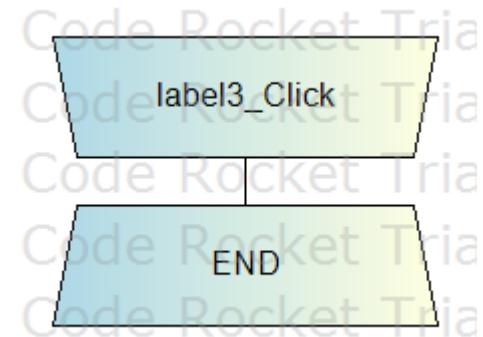
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: label2\_Click**

```
private void label2_Click(object sender, EventArgs e)
```

**Parameters**

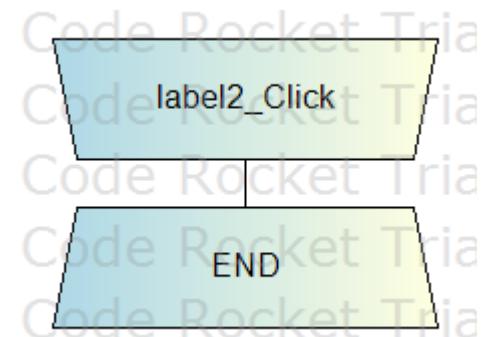
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Procedure: label1\_Click*

```
private void label1_Click(object sender, EventArgs e)
```

**Parameters**

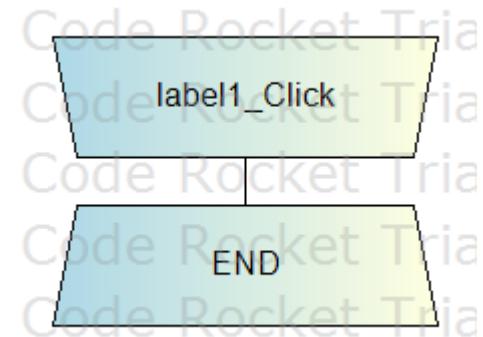
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: loadMenusTable\_Click**

```
private void loadMenusTable_Click(object sender, EventArgs e)
```

**Parameters**

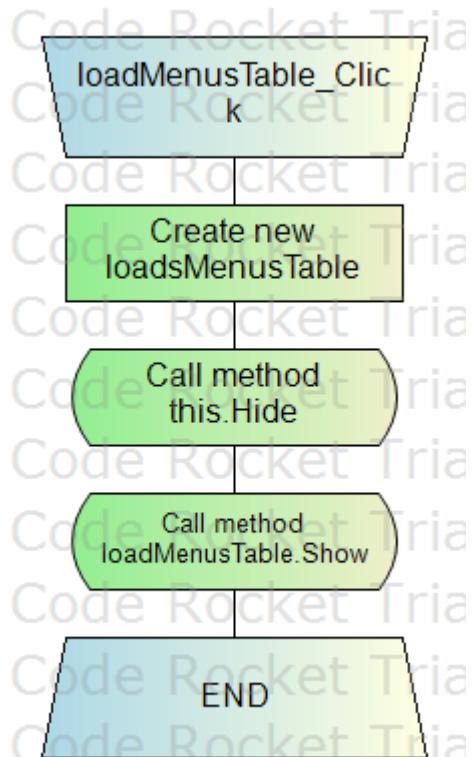
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new loadsMenusTable  
Call method this.Hide  
Call method loadMenusTable.Show
```

**Flowchart**



*Code File: loadsMenus.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

**Class: loadsMenus**

partial class loadsMenus

**Attribute: components**

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure: Dispose**

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

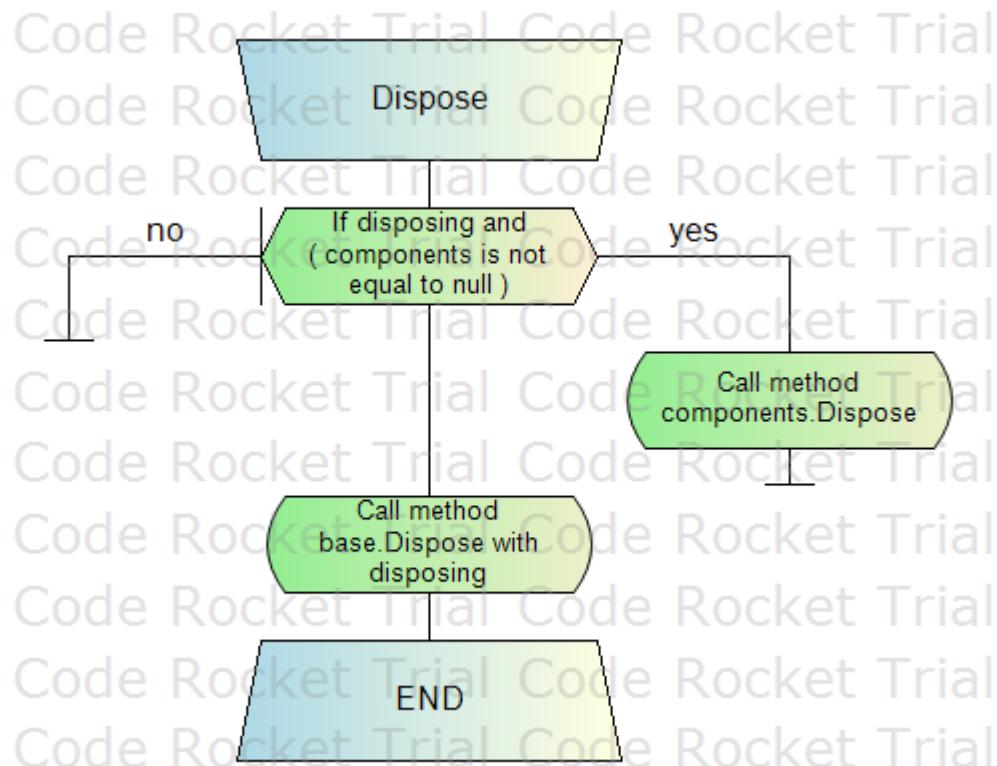
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

**Flowchart**



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.Button
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackColor to System.Drawing.Color.Transparent
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.TabStop to false
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackColor to System.Drawing.Color.Transparent
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.TabStop to false
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
addLoadButton
Set this.addLoadButton.BackColor to System.Drawing.Color.Transparent
Set this.addLoadButton.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.addLoadButton.FlatAppearance.BorderSize to 2
Set this.addLoadButton.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( int ), ( int ) )

```

```

Set this.addLoadButton.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.addLoadButton.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.addLoadButton.Name to "addLoadButton"
Create new System.Drawing.Size
Set this.addLoadButton.TabIndex to 31
Set this.addLoadButton.TabStop to false
Set this.addLoadButton.Text to "ADD LOADS"
Set this.addLoadButton.UseVisualStyleBackColor to false
This.addLoadButton.Click += new System.EventHandler(this.addLoadButton_Click);
xForceTextbox
Create new System.Drawing.Font
Create new System.Drawing.Point
Set this.xForceTextbox.Name to "xForceTextbox"
Create new System.Drawing.Size
Set this.xForceTextbox.TabIndex to 1
Set this.xForceTextbox.Text to "Enter the X axis forces here"
This.xForceTextbox.TextChanged += new
System.EventHandler(this.xForceTextbox_TextChanged);
label2
Set this.label2.AutoSize to true
Set this.label2.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label2.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label2.Name to "label2"
Create new System.Drawing.Size
Set this.label2.TabIndex to 29
Set this.label2.Text to "X AXIS FORCES:"
This.label2.Click += new System.EventHandler(this.label2_Click);
nodeTextbox
Create new System.Drawing.Font
Create new System.Drawing.Point
Set this.nodeTextbox.Name to "nodeTextbox"
Create new System.Drawing.Size
Set this.nodeTextbox.TabIndex to 0
Set this.nodeTextbox.Text to "Enter the node here"
This.nodeTextbox.TextChanged += new
System.EventHandler(this.nodeTextbox_TextChanged);
label1
Set this.label1.AutoSize to true
Set this.label1.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 27
Set this.label1.Text to "NODE:"
This.label1.Click += new System.EventHandler(this.label1_Click);
yForceTextbox
Create new System.Drawing.Font
Create new System.Drawing.Point
Set this.yForceTextbox.Name to "yForceTextbox"
Create new System.Drawing.Size
Set this.yForceTextbox.TabIndex to 2
Set this.yForceTextbox.Text to "Enter the Y axis forces here"
This.yForceTextbox.TextChanged += new
System.EventHandler(this.yForceTextbox_TextChanged);
label3
Set this.label3.AutoSize to true
Set this.label3.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font

```

```

Set this.label3.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label3.Name to "label3"
Create new System.Drawing.Size
Set this.label3.TabIndex to 32
Set this.label3.Text to "Y AXIS FORCES:"
This.label3.Click += new System.EventHandler(this.label3_Click);
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 37
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent
Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 38
Set this.BackToMainMenu.TabStop to false
Set this.BackToMainMenu.Text to "BACK"
Set this.BackToMainMenu.UseVisualStyleBackColor to false
This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 40
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 39
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
loadMenusTable
Set this.loadMenusTable.BackColor to System.Drawing.Color.Transparent
Set this.loadMenusTable.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.loadMenusTable.FlatAppearance.BorderSize to 2
Set this.loadMenusTable.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )

```

```

Set this.loadMenusTable.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.loadMenusTable.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.loadMenusTable.Name to "loadMenusTable"
Create new System.Drawing.Size
Set this.loadMenusTable.TabIndex to 41
Set this.loadMenusTable.TabStop to false
Set this.loadMenusTable.Text to "TABLES"
Set this.loadMenusTable.UseVisualStyleBackColor to false
This.loadMenusTable.Click += new System.EventHandler(this.loadMenusTable_Click);
loadsMenus
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.loadMenusTable
Call method this.Controls.Add with this.yForceTextbox
Call method this.Controls.Add with this.labelX3
Call method this.Controls.Add with this.xForceTextbox
Call method this.Controls.Add with this.labelX2
Call method this.Controls.Add with this.nodeTextbox
Call method this.Controls.Add with this.labelX1
Call method this.Controls.Add with this.bridgeDrawing
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.BackToMainMenu
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.addLoadButton
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.DoubleBuffered to true
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "loadsMenus"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitialize.ISupportInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout

```

### Flowchart



```

#endif

private System.Windows.Forms.Button minimize;
Attribute: minimize

private System.Windows.Forms.Button exitApplication;
Attribute: exitApplication

private System.Windows.Forms.Button addLoadButton;
Attribute: addLoadButton

private System.Windows.Forms.Button addLoadButton;
Attribute: xForceTextbox

private System.Windows.Forms.TextBox xForceTextbox;
Attribute: label2

private System.Windows.Forms.Label label2;
Attribute: nodeTextbox

private System.Windows.Forms.TextBox nodeTextbox;
Attribute: label1

private System.Windows.Forms.Label label1;
Attribute: yForceTextbox

private System.Windows.Forms.TextBox yForceTextbox;
Attribute: label3

private System.Windows.Forms.Label label3;
Attribute: moveMenu

private System.Windows.Forms.Panel moveMenu;
Attribute: BackToMainMenu

private System.Windows.Forms.Button BackToMainMenu;
Attribute: bridgeDrawing

private System.Windows.Forms.Panel bridgeDrawing;
Attribute: zoomInBar

private System.Windows.Forms.TrackBar zoomInBar;
Attribute: loadMenusTable

private System.Windows.Forms.Button loadMenusTable;
Code File: loadsMenusTable.cs

using Sectrics_V2.Properties;
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: loadsMenusTable**

```
public partial class loadsMenusTable : Form
```

**Attribute: zoom**

```
float zoom = 1f;
```

**Attribute: xMouseOffset**

```
double xMouseOffset;
```

**Attribute: yMouseOffset**

```
double yMouseOffset;
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

**Procedure: Constructor**

```
public loadsMenusTable()
```

Returns loadsMenusTable

**Pseudocode**

Call method InitializeComponent

Adds The Array Coordinates Into ListBox

**For** i is 0, i is less than Program.bridgeData.nodes.Count, i increments by 1

    Call method nodeListView.Items.Add with "Node " plus i plus " " plus "X

Coordinate: " plus Program.bridgeData.nodes plus " | Y Coordinate: " plus

Program.bridgeData.nodes

**EndFor**

Adds The Array Coordinates Into ListBox

**For** i is 0, i is less than Program.bridgeData.forces.Count, i increments by 1

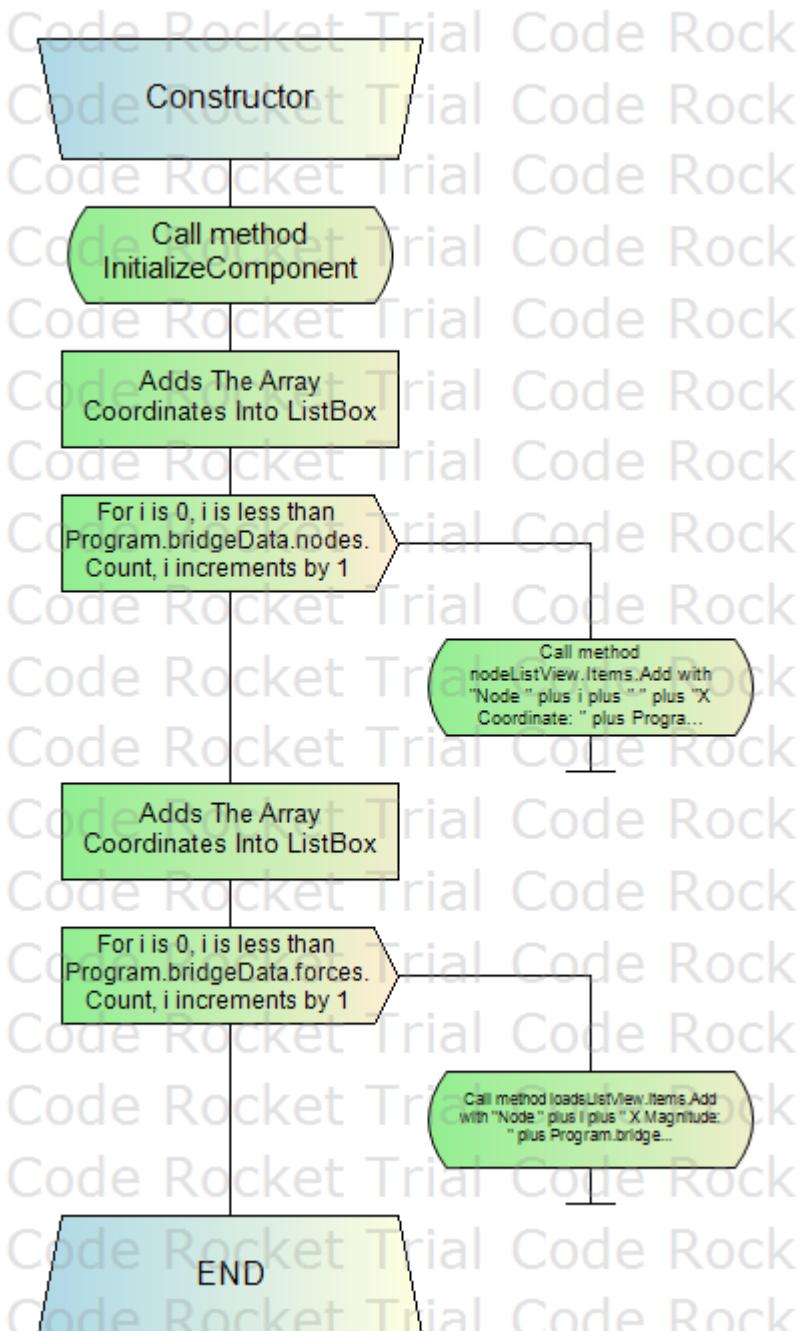
    Call method loadsListView.Items.Add with "Node " plus i plus " X Magnitude: "

plus Program.bridgeData.forces plus " | Y Magnitude: " plus

Program.bridgeData.forces

**EndFor**

**Flowchart**



**Procedure: WndProc**

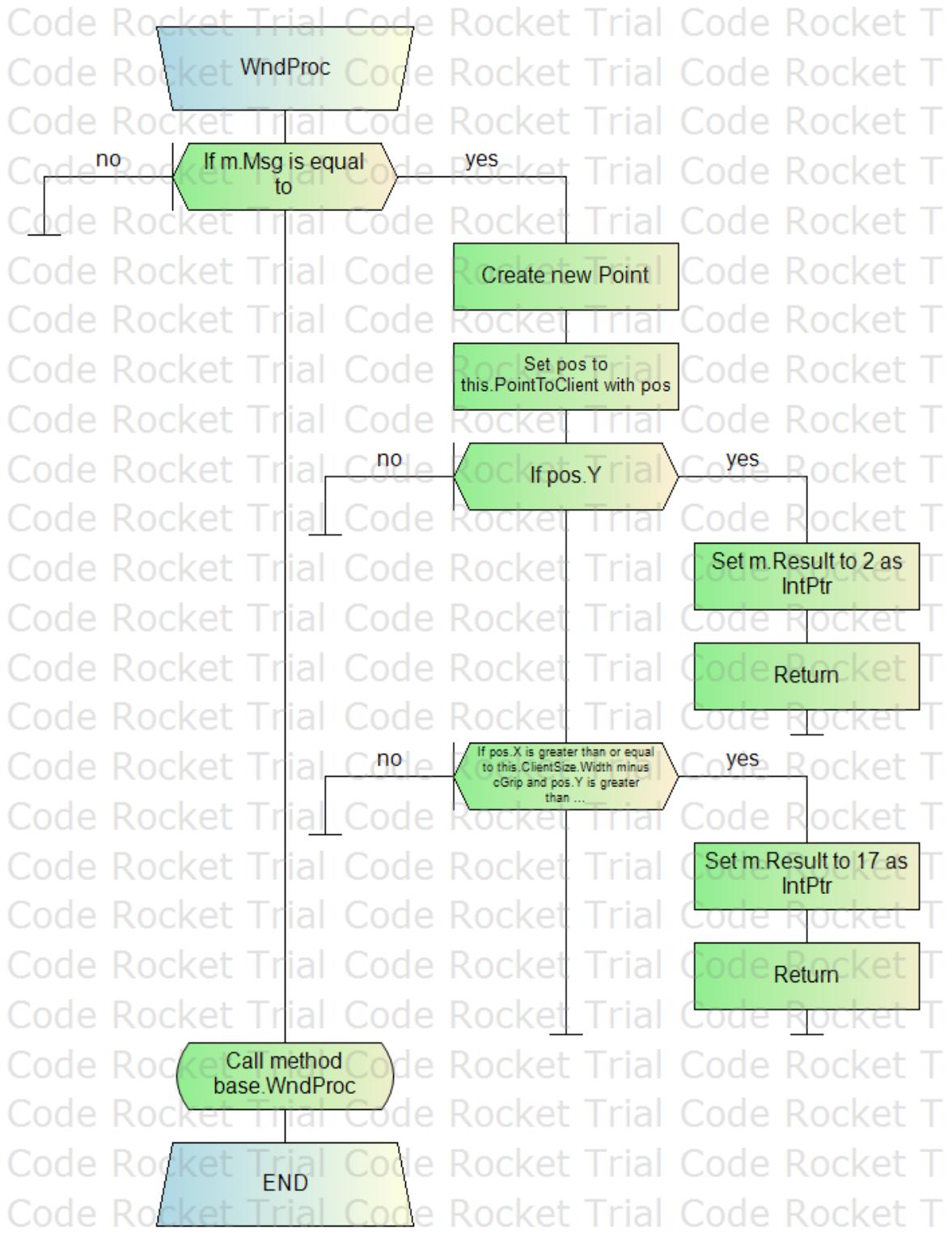
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

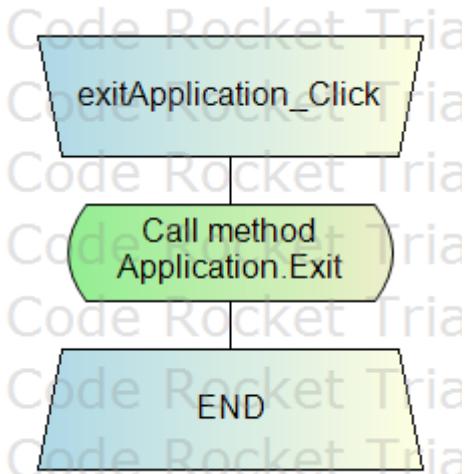
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



**Procedure: minimize\_Click**

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

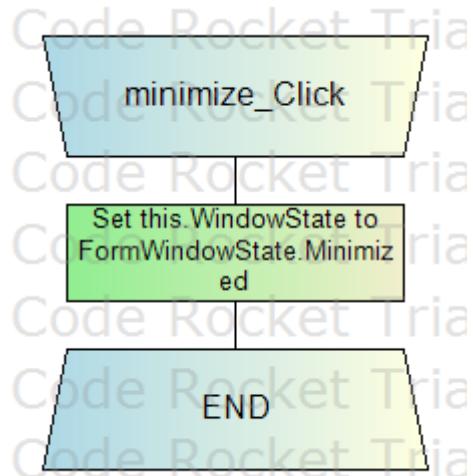
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

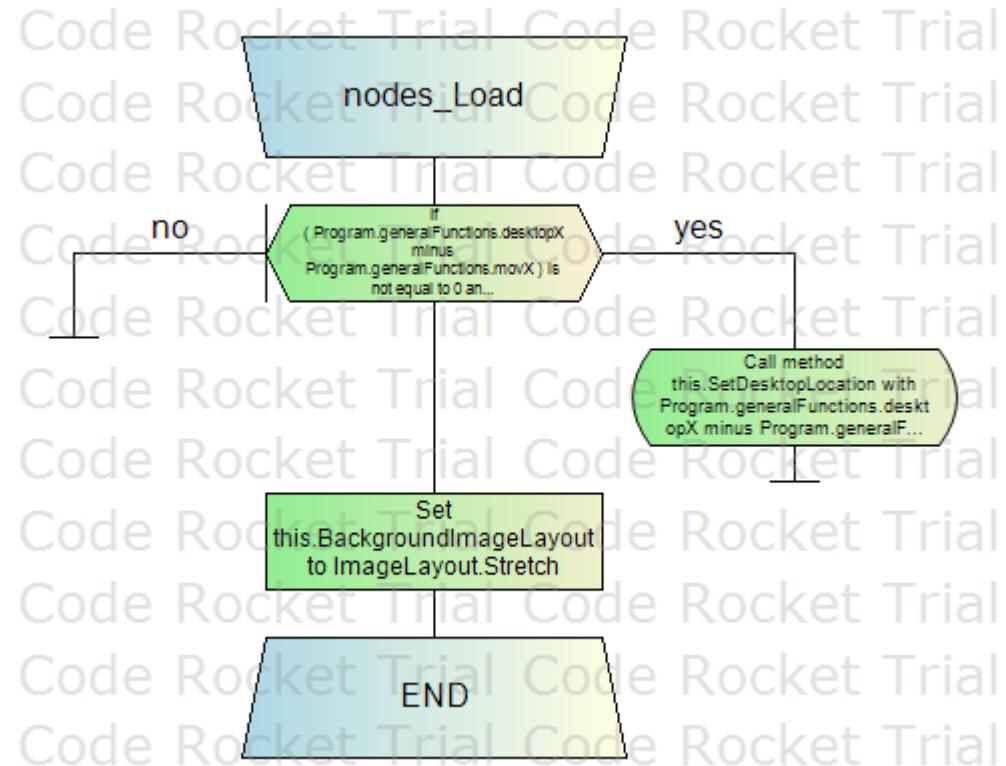
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

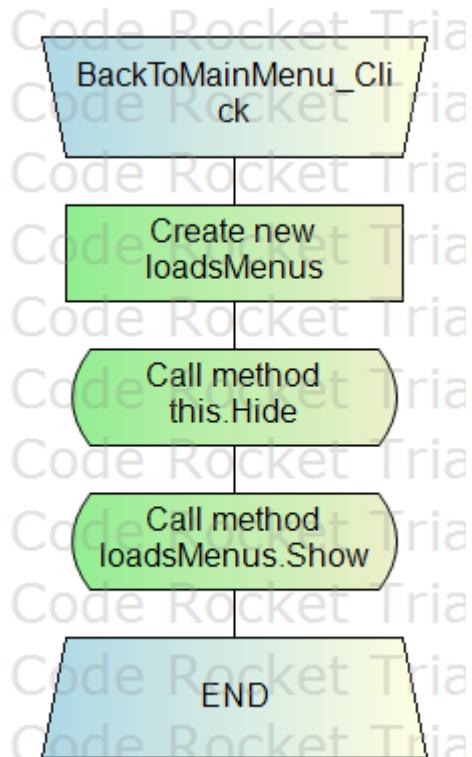
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new loadsMenus  
Call method this.Hide  
Call method loadsMenus.Show
```

**Flowchart**



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

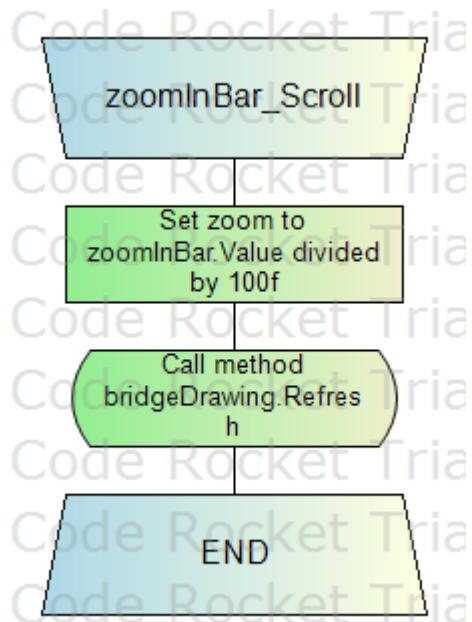
Returns Void

**Pseudocode**

Set zoom to zoomInBar.Value divided by 100f

Call method bridgeDrawing.Refresh

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

**Try**

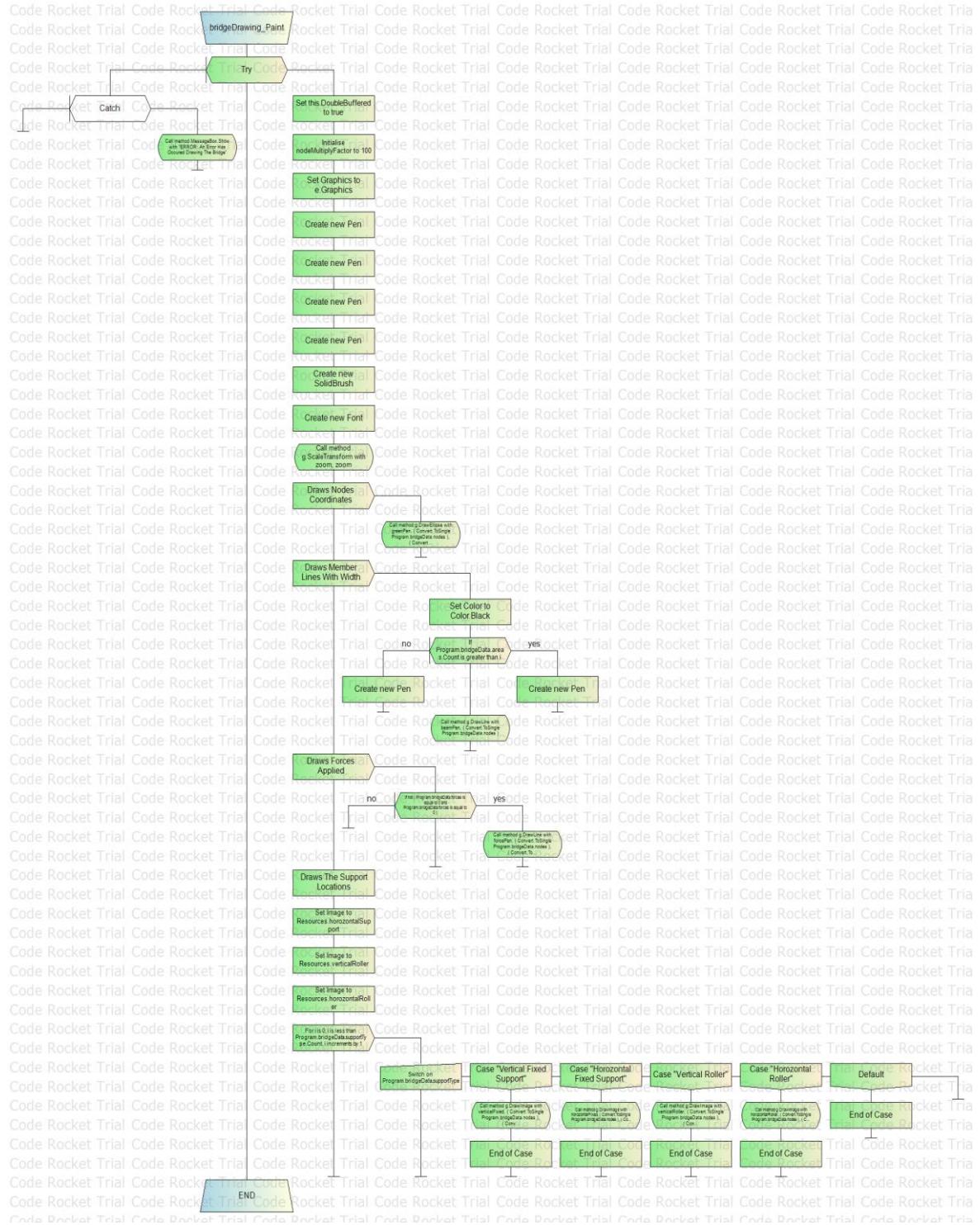
```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```

        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry

```

### Flowchart



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

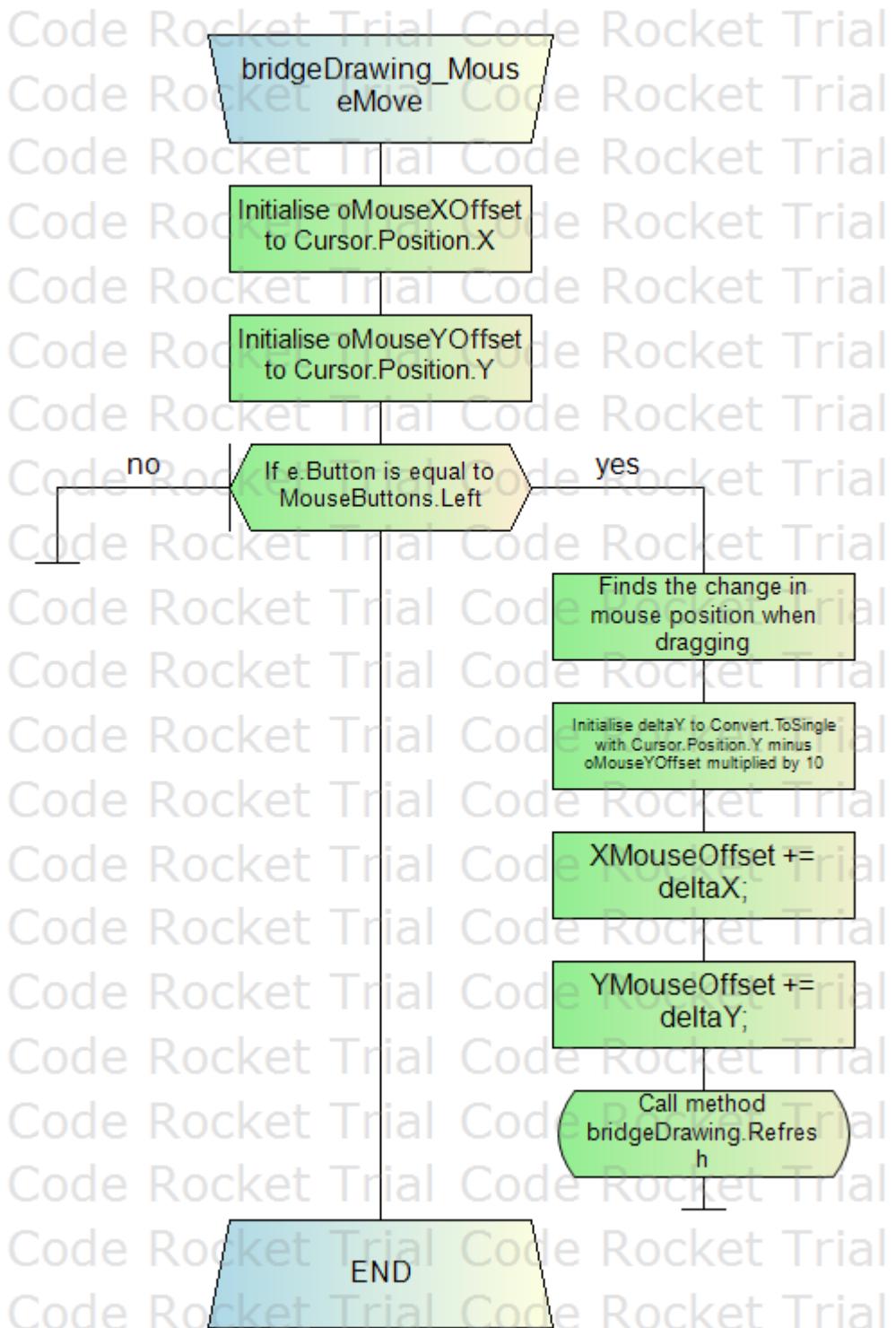
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



**Procedure: removeLoad\_Click**

```
private void removeLoad_Click(object sender, EventArgs e)
```

**Parameters**

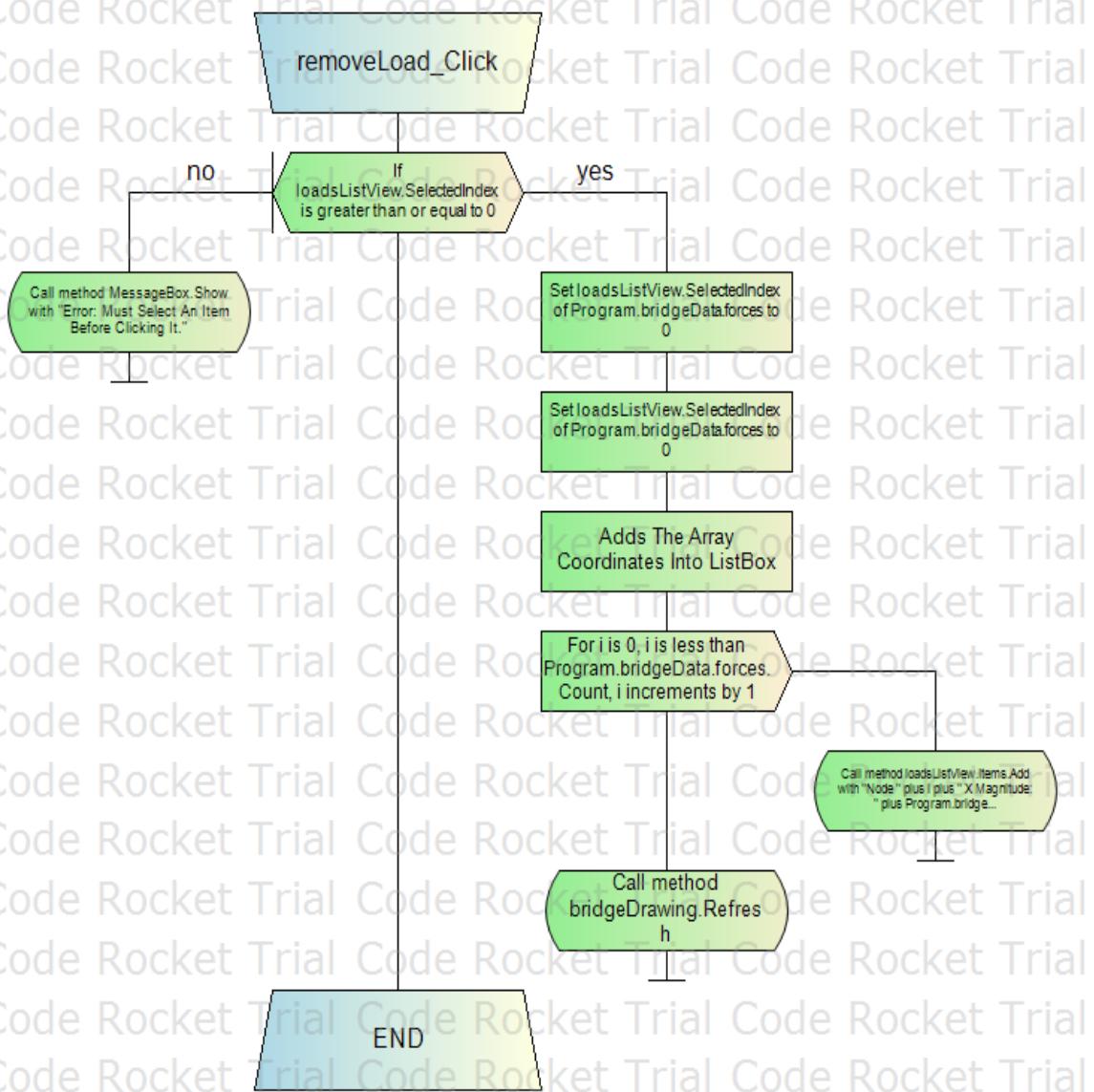
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If loadsListView.SelectedIndex is greater than or equal to 0
    Set loadsListView.SelectedIndex of Program.bridgeData.forces to 0
    Set loadsListView.SelectedIndex of Program.bridgeData.forces to 0
    Adds The Array Coordinates Into ListBox
    For i is 0, i is less than Program.bridgeData.forces.Count, i increments by 1
        Call method loadsListView.Items.Add with "Node " plus i plus " X
Magnitude: " plus Program.bridgeData.forces plus " | Y Magnitude: " plus
Program.bridgeData.forces
    EndFor
    Call method bridgeDrawing.Refresh
Else
    Call method MessageBox.Show with "Error: Must Select An Item Before Clicking
It."
EndIf
```

**Flowchart**



**Procedure: removeAll\_Click**

```
private void removeAll_Click(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
For i is 0, i is less than Program.bridgeData.forces.Count, i increments by 1
    Set i of Program.bridgeData.forces to 0
    Set i of Program.bridgeData.forces to 0
```

**EndFor**

Adds The Array Coordinates Into ListBox

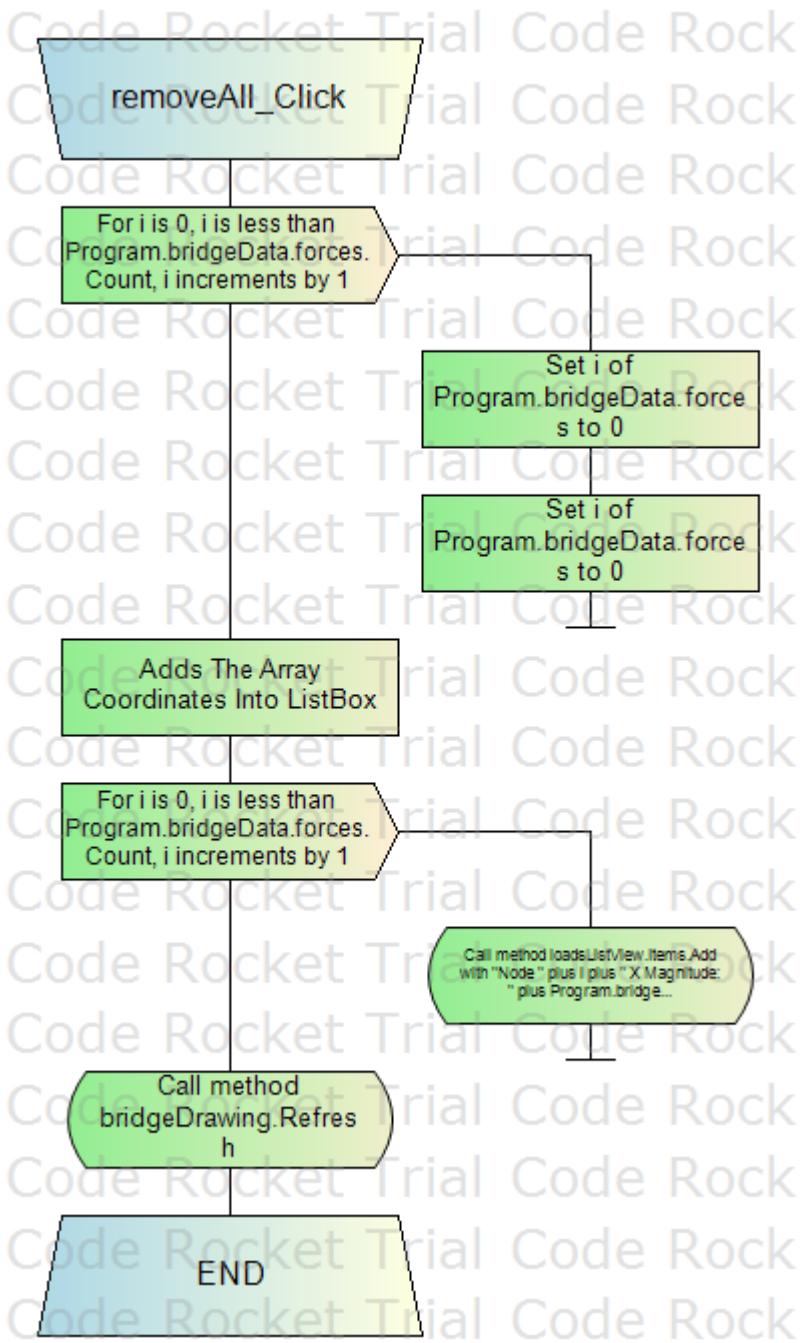
```
For i is 0, i is less than Program.bridgeData.forces.Count, i increments by 1
    Call method loadsListView.Items.Add with "Node " plus i plus " X Magnitude: "
    plus Program.bridgeData.forces plus " | Y Magnitude: " plus
```

Program.bridgeData.forces

**EndFor**

Call method bridgeDrawing.Refresh

**Flowchart**



**Code File:** loadsMenusTable.Designer.cs

**Namespace:** Sectrics\_V2

```
namespace Sectrics_V2
```

**Class:** loadsMenusTable

```
partial class loadsMenusTable
```

**Attribute:** components

```
private System.ComponentModel.IContainer components = null;
```

Required designer variable.

**Procedure:** Dispose

```
protected override void Dispose (bool disposing)
```

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

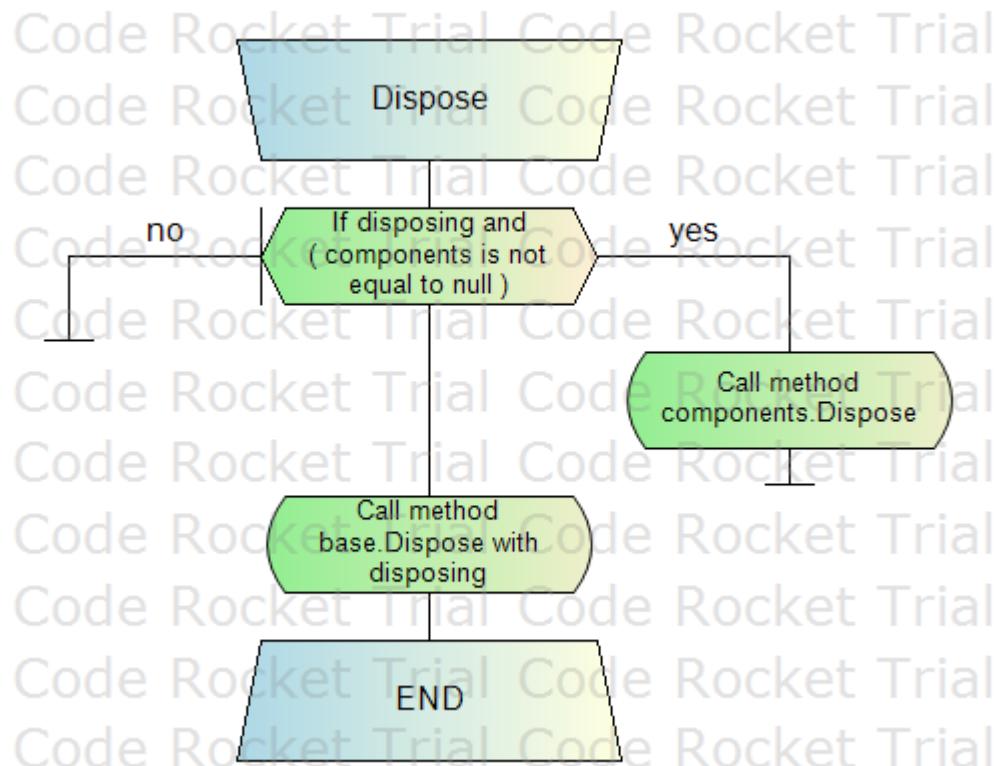
**Pseudocode**

```
If disposing and ( components is not equal to null )  
    Call method components.Dispose
```

EndIf

```
Call method base.Dispose with disposing
```

Flowchart



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.ListBox
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.ListBox
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent
Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( int ), ( int ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 39
Set this.BackToMainMenu.TabStop to false
Set this.BackToMainMenu.Text to "BACK"

```

```

Set this.BackToMainMenu.UseVisualStyleBackColor to false
This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
nodeListView
Set this.nodeTypeView.FormattingEnabled to true
Set this.nodeTypeView.ItemHeight to 25
Create new System.Drawing.Point
Set this.nodeTypeView.Name to "nodeListView"
Create new System.Drawing.Size
Set this.nodeTypeView.TabIndex to 40
removeAll
Set this.removeAll.BackColor to System.Drawing.Color.Transparent
Set this.removeAll.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.removeAll.FlatAppearance.BorderSize to 2
Set this.removeAll.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.removeAll.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.removeAll.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.removeAll.Name to "removeAll"
Create new System.Drawing.Size
Set this.removeAll.TabIndex to 43
Set this.removeAll.TabStop to false
Set this.removeAll.Text to "CLEAR ALL"
Set this.removeAll.UseVisualStyleBackColor to false
This.removeAll.Click += new System.EventHandler(this.removeAll_Click);
removeLoad
Set this.removeLoad.BackColor to System.Drawing.Color.Transparent
Set this.removeLoad.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.removeLoad.FlatAppearance.BorderSize to 2
Set this.removeLoad.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.removeLoad.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.removeLoad.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.removeLoad.Name to "removeLoad"
Create new System.Drawing.Size
Set this.removeLoad.TabIndex to 42
Set this.removeLoad.TabStop to false
Set this.removeLoad.Text to "REMOVE LOAD"
Set this.removeLoad.UseVisualStyleBackColor to false
This.removeLoad.Click += new System.EventHandler(this.removeLoad_Click);
loadsListView
Set this.loadsListView.FormattingEnabled to true
Set this.loadsListView.ItemHeight to 25
Create new System.Drawing.Point
Set this.loadsListView.Name to "loadsListView"
Create new System.Drawing.Size
Set this.loadsListView.TabIndex to 41
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 45
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point

```

```

Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 44
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
loadsMenusTable
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.bridgeDrawing
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.nodeTypeView
Call method this.Controls.Add with this.removeAll
Call method this.Controls.Add with this.removeLoad
Call method this.Controls.Add with this.loadsListView
Call method this.Controls.Add with this.BackToMainMenu
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "loadsMenusTable"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitializeInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout

```

### Flowchart



```

#endif

Attribute: minimize

private System.Windows.Forms.Button minimize;
Attribute: exitApplication

private System.Windows.Forms.Button exitApplication;
Attribute: BackToMainMenu

private System.Windows.Forms.Button BackToMainMenu;
Attribute: nodeListView

private System.Windows.Forms.ListBox nodeListView;
Attribute: removeAll

private System.Windows.Forms.Button removeAll;
Attribute: removeLoad

private System.Windows.Forms.Button removeLoad;
Attribute: loadsListView

private System.Windows.Forms.ListBox loadsListView;
Attribute: bridgeDrawing

private System.Windows.Forms.Panel bridgeDrawing;
Attribute: zoomInBar

private System.Windows.Forms.TrackBar zoomInBar;
Code File: materialProperties.cs

using Sectrics_V2.Properties;
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: Sectrics_V2

namespace Sectrics_V2
Class: materialProperties

public partial class materialProperties : Form

```

*Attribute: zoom*

```
float zoom = 1f;
```

*Attribute: xMouseOffset*

```
double xMouseOffset;
```

*Attribute: yMouseOffset*

```
double yMouseOffset;
```

*Attribute: cGrip*

```
private const int cGrip = 16;
```

*Attribute: cCaption*

```
private const int cCaption = 32;
```

*Procedure: Constructor*

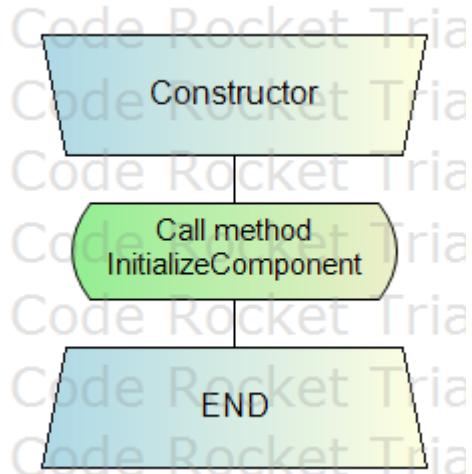
```
public materialProperties()
```

Returns materialProperties

**Pseudocode**

```
Call method InitializeComponent
```

Flowchart



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

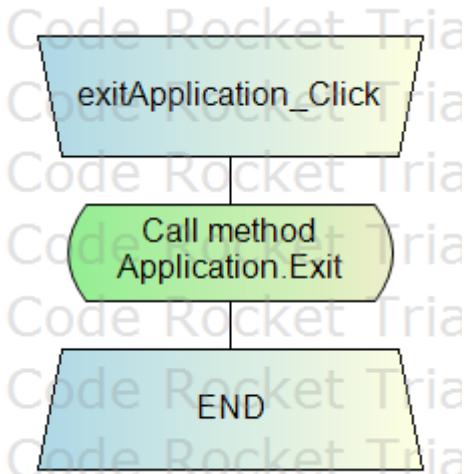
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



**Procedure: minimize\_Click**

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

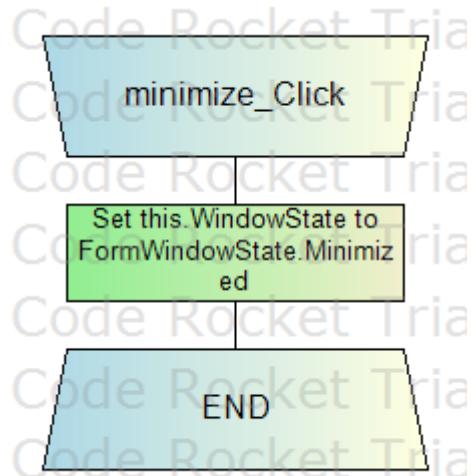
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

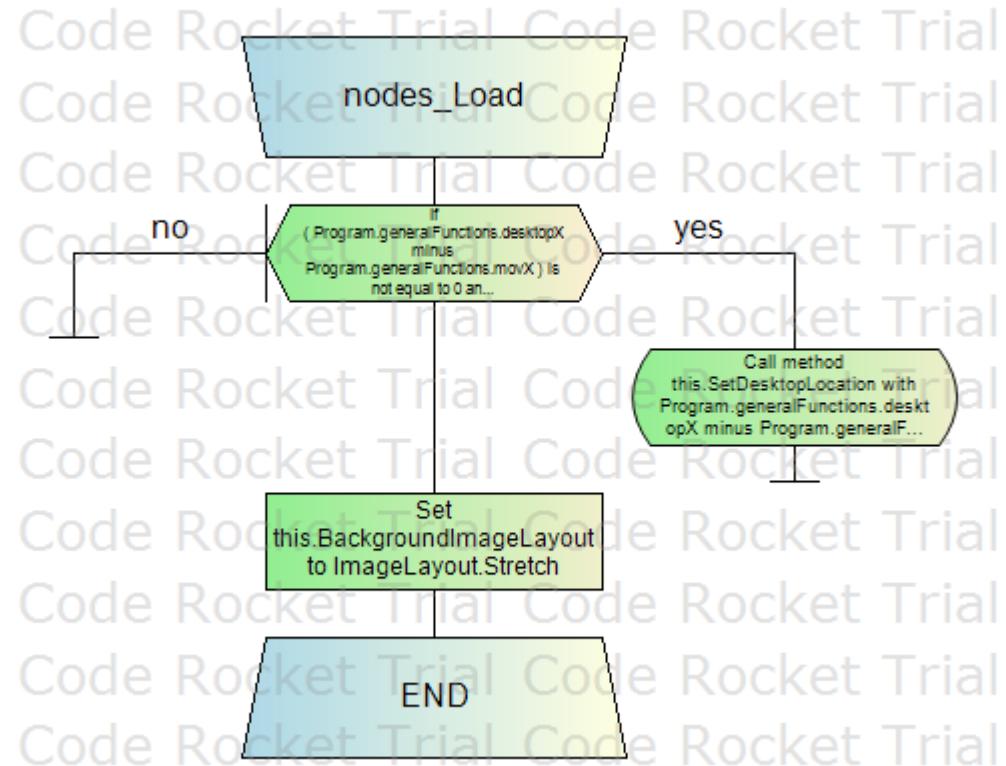
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



### Procedure: addMaterialPropertiesButton\_Click

```
private void addMaterialPropertiesButton_Click(object sender,  
EventArgs e)
```

#### Parameters

- sender - Object
- e - EventArgs

Returns Void

#### Pseudocode

##### Try

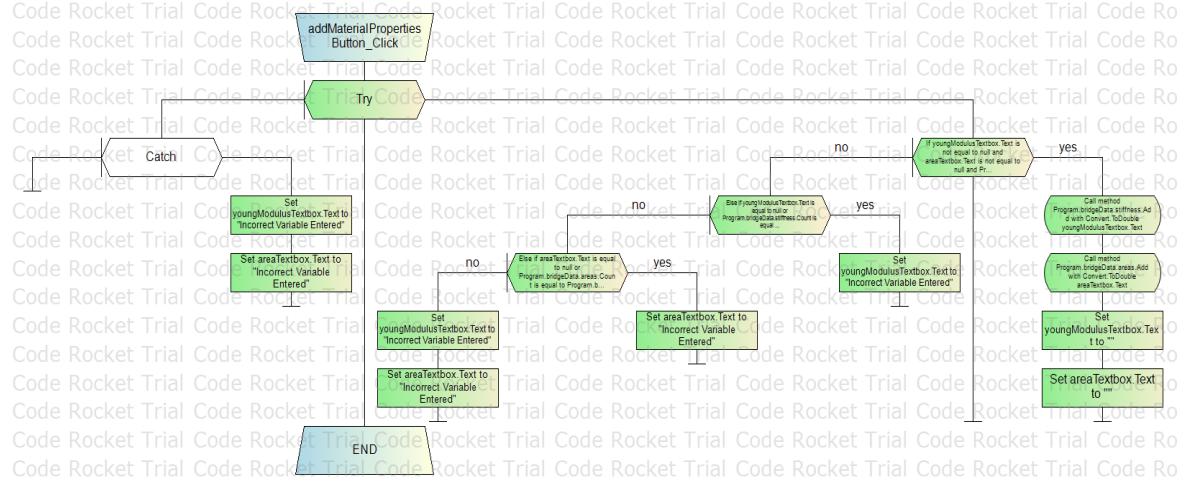
```
If youngModulusTextbox.Text is not equal to null and areaTextbox.Text is not  
equal to null and Program.bridgeData.stiffness.Count is not equal to  
Program.bridgeData.memberConnection.Count and Program.bridgeData.areas.Count is not  
equal to Program.bridgeData.memberConnection.Count  
    Call method Program.bridgeData.stiffness.Add with Convert.ToDouble  
youngModulusTextbox.Text  
    Call method Program.bridgeData.areas.Add with Convert.ToDouble  
areaTextbox.Text  
    Set youngModulusTextbox.Text to ""  
    Set areaTextbox.Text to ""  
Else if youngModulusTextbox.Text is equal to null or  
Program.bridgeData.stiffness.Count is equal to  
Program.bridgeData.memberConnection.Count  
    Set youngModulusTextbox.Text to "Incorrect Variable Entered"  
Else if areaTextbox.Text is equal to null or Program.bridgeData.areas.Count is  
equal to Program.bridgeData.memberConnection.Count  
    Set areaTextbox.Text to "Incorrect Variable Entered"  
Else  
    Set youngModulusTextbox.Text to "Incorrect Variable Entered"  
    Set areaTextbox.Text to "Incorrect Variable Entered"  
EndIf
```

##### Catch

```
Set youngModulusTextbox.Text to "Incorrect Variable Entered"  
Set areaTextbox.Text to "Incorrect Variable Entered"
```

##### EndTry

#### Flowchart



**Procedure: areaTextbox\_TextChanged**

```
private void areaTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

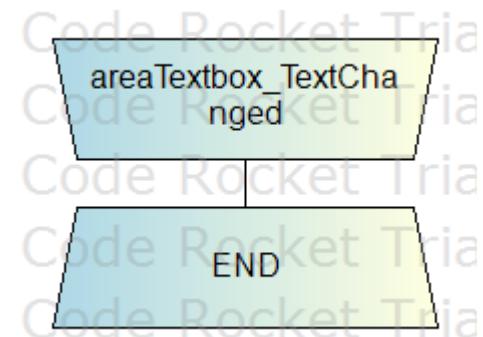
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: youngModulusTextbox\_TextChanged**

```
private void youngModulusTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

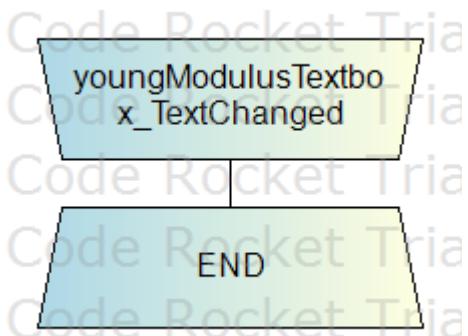
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Procedure: membersListView\_SelectedIndexChanged*

```
private void membersListView_SelectedIndexChanged(object sender,  
EventArgs e)
```

**Parameters**

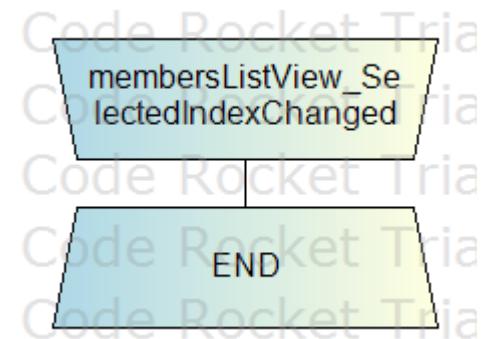
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Procedure: materialsListView\_SelectedIndexChanged*

```
private void materialsListView_SelectedIndexChanged(object sender,  
EventArgs e)
```

**Parameters**

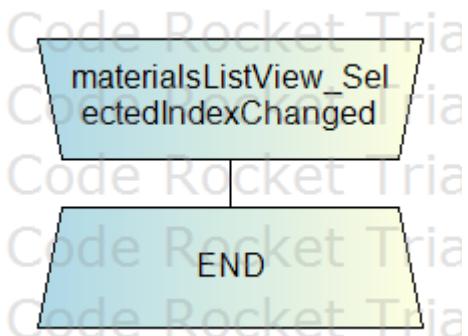
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

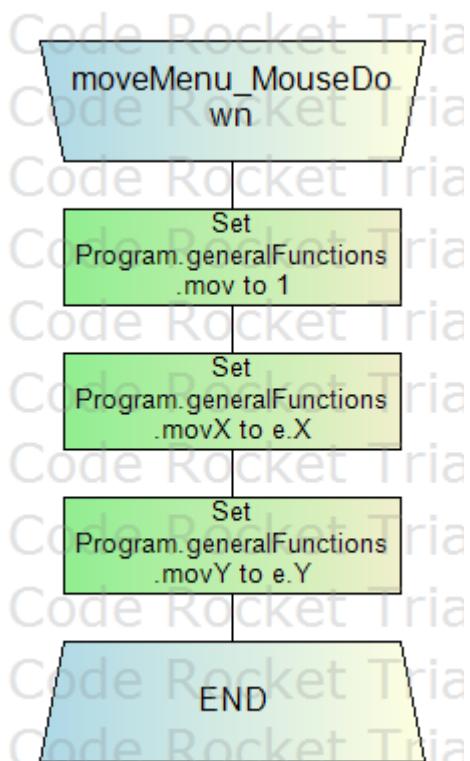
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

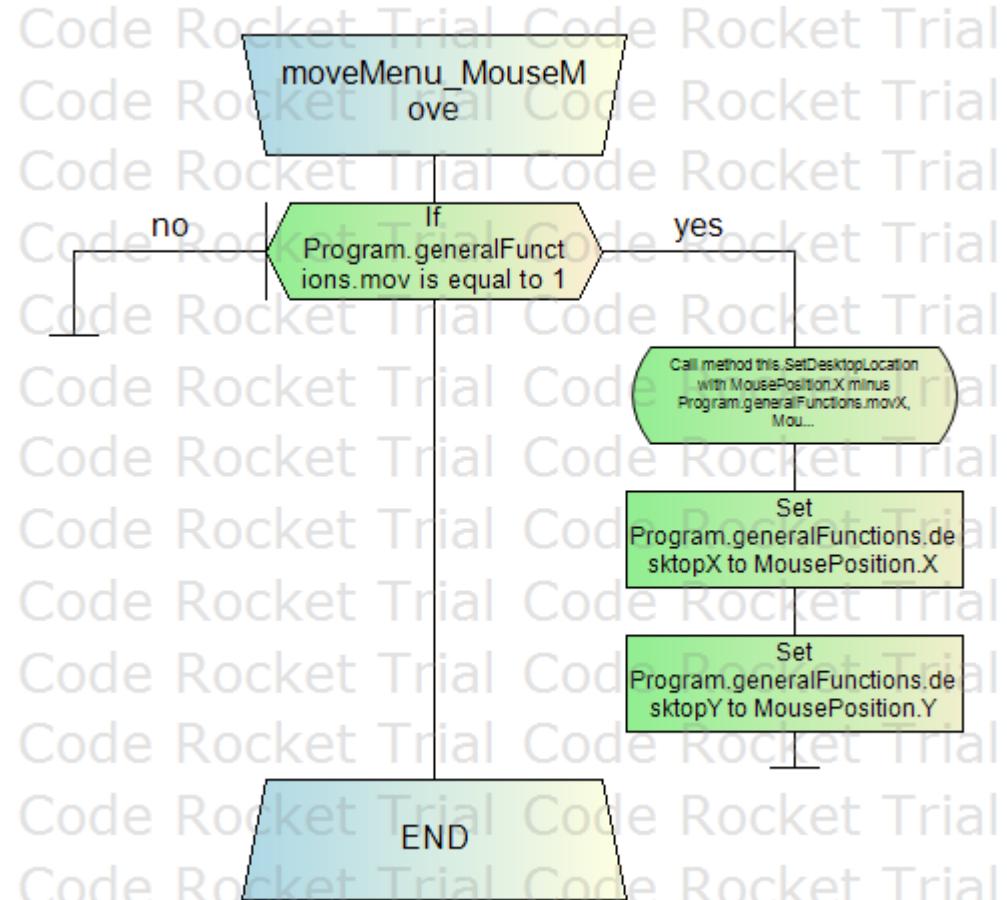
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseUp**

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

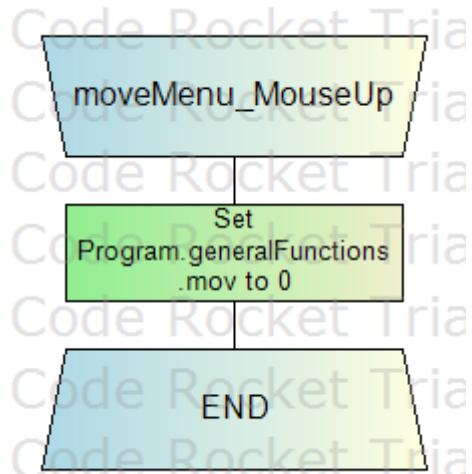
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

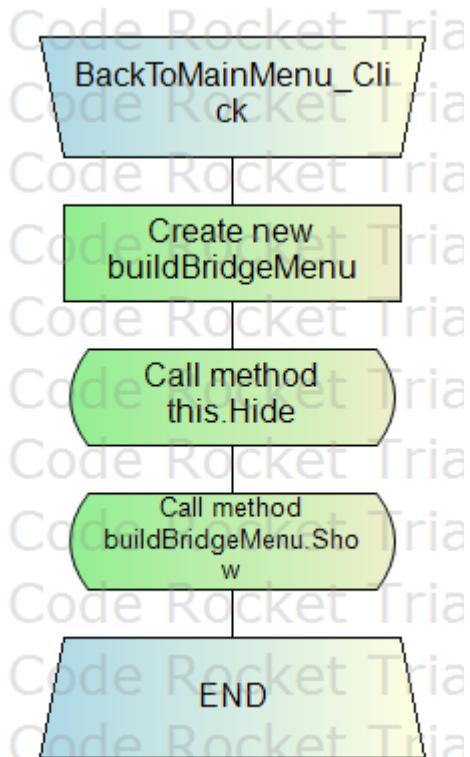
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

**Try**

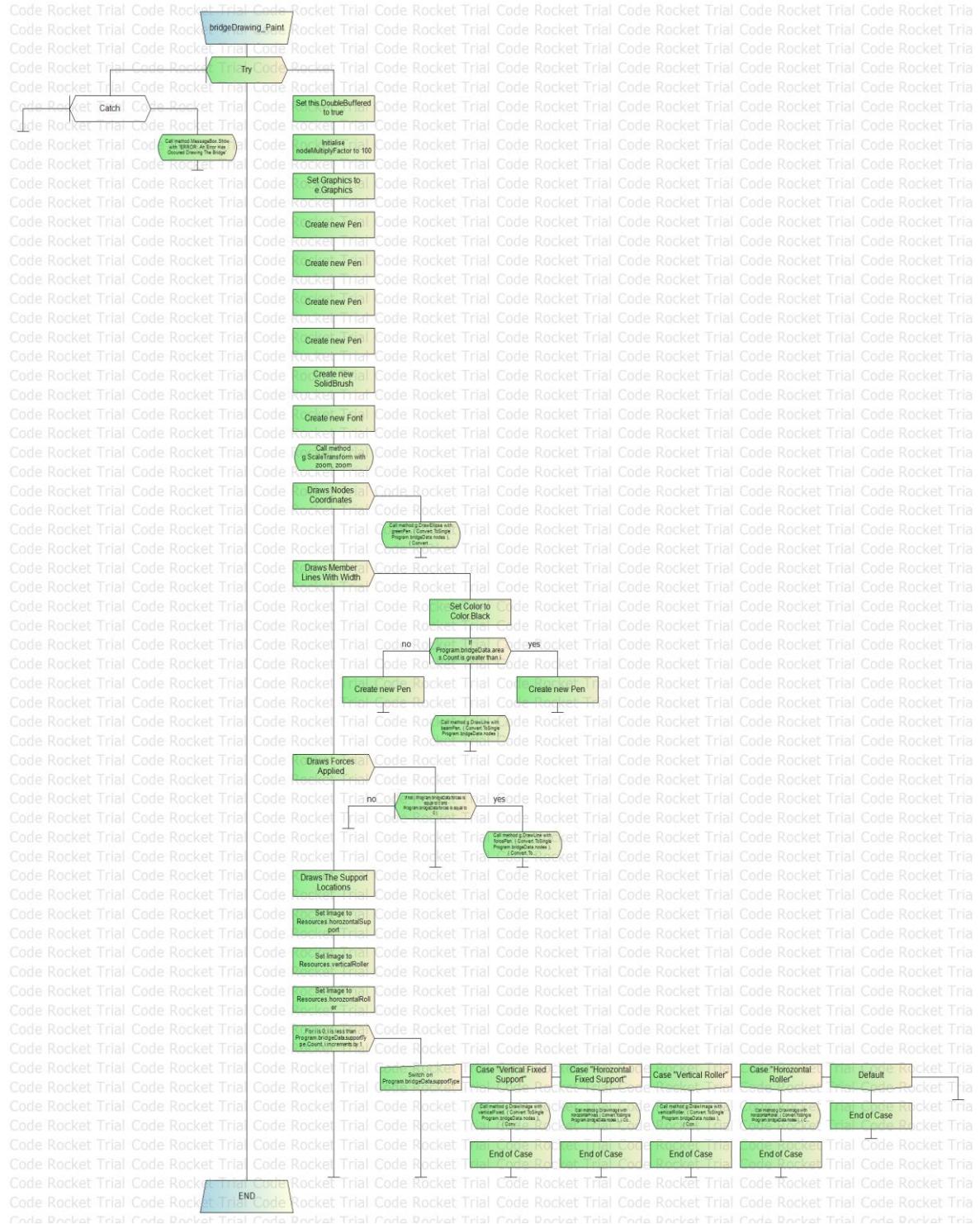
```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```

        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry

```

### Flowchart



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

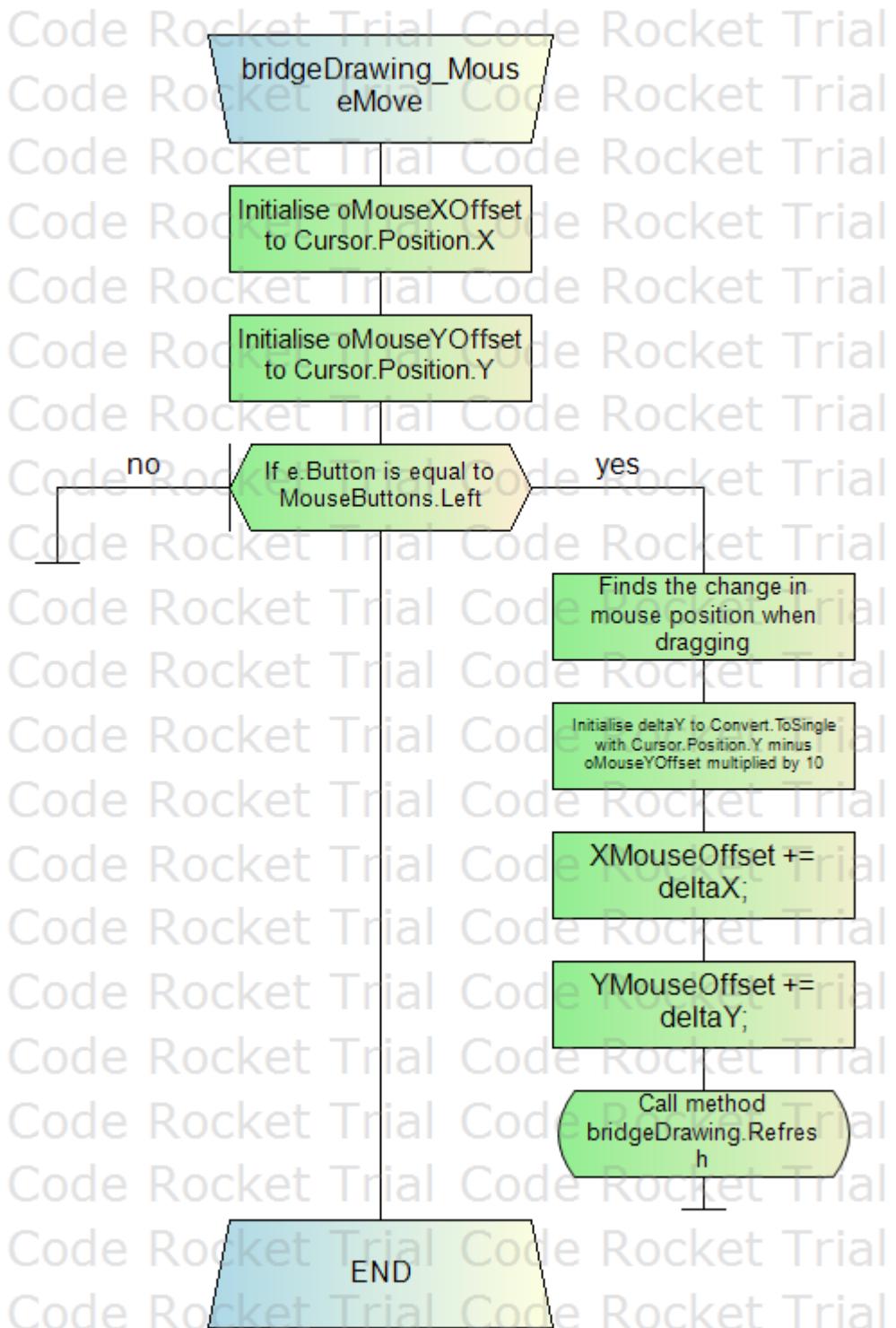
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

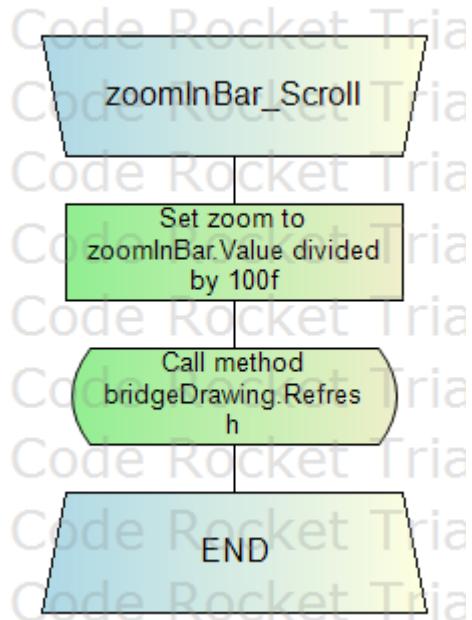
Returns Void

**Pseudocode**

```
Set zoom to zoomInBar.Value divided by 100f
```

```
Call method bridgeDrawing.Refresh
```

**Flowchart**



*Procedure: membersTable\_Click*

```
private void membersTable_Click(object sender, EventArgs e)
```

**Parameters**

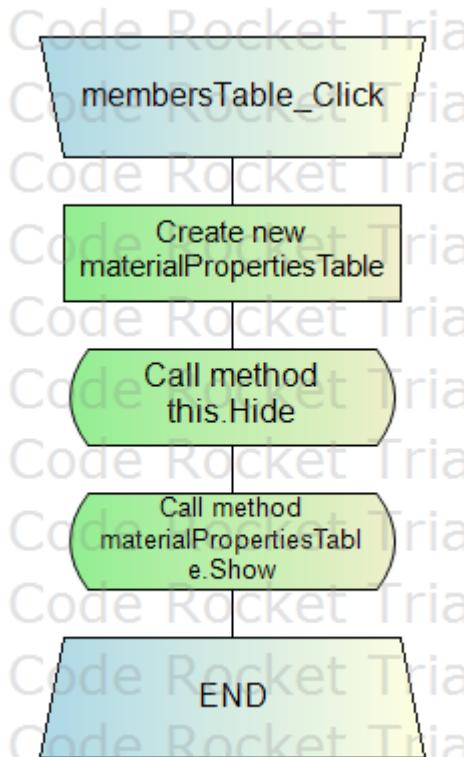
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new materialPropertiesTable  
Call method this.Hide  
Call method materialPropertiesTable.Show
```

**Flowchart**



*Code File: materialProperties.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

**Class: materialProperties**

partial class materialProperties

**Attribute: components**

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure: Dispose**

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

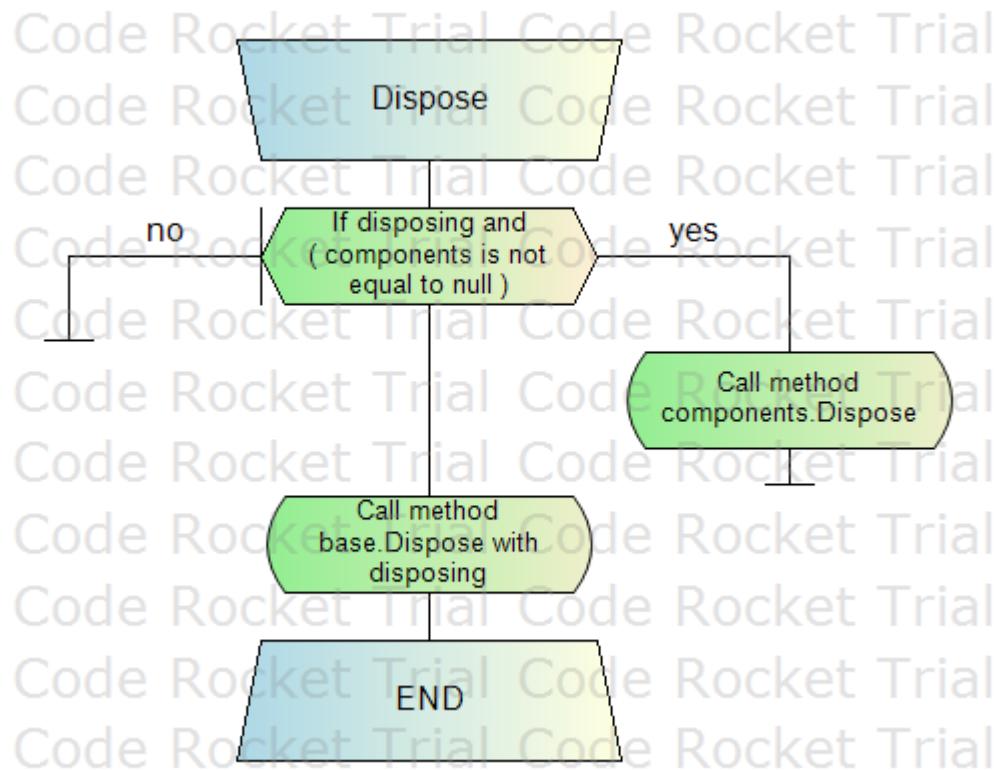
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

Flowchart



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.Button
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackColor to System.Drawing.Color.Transparent
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.TabStop to false
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackColor to System.Drawing.Color.Transparent
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.TabStop to false
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
addMaterialPropertiesButton
Set this.addMaterialPropertiesButton.BackColor to System.Drawing.Color.Transparent
Set this.addMaterialPropertiesButton.FlatAppearance.BorderColor to
System.Drawing.Color.White
Set this.addMaterialPropertiesButton.FlatAppearance.BorderSize to 2
Set this.addMaterialPropertiesButton.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.addMaterialPropertiesButton.FlatStyle to

```

```

System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.addMaterialPropertiesButton.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.addMaterialPropertiesButton.Name to "addMaterialPropertiesButton"
Create new System.Drawing.Size
Set this.addMaterialPropertiesButton.TabIndex to 26
Set this.addMaterialPropertiesButton.TabStop to false
Set this.addMaterialPropertiesButton.Text to "ADD MATERIAL PROPERTIES"
Set this.addMaterialPropertiesButton.UseVisualStyleBackColor to false
This.addMaterialPropertiesButton.Click += new
System.EventHandler(this.addMaterialPropertiesButton_Click);
areaTextbox
Create new System.Drawing.Font
Create new System.Drawing.Point
Set this.areaTextbox.Name to "areaTextbox"
Create new System.Drawing.Size
Set this.areaTextbox.TabIndex to 1
Set this.areaTextbox.Text to "Enter the cross-sectional area here"
This.areaTextbox.TextChanged += new
System.EventHandler(this.areaTextbox_TextChanged);
label2
Set this.label2.AutoSize to true
Set this.label2.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label2.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label2.Name to "label2"
Create new System.Drawing.Size
Set this.label2.TabIndex to 24
Set this.label2.Text to "CROSS-SECTIONAL AREA:"
youngModulusTextbox
Create new System.Drawing.Font
Create new System.Drawing.Point
Set this.youngModulusTextbox.Name to "youngModulusTextbox"
Create new System.Drawing.Size
Set this.youngModulusTextbox.TabIndex to 0
Set this.youngModulusTextbox.Text to "Enter the young modulus here"
This.youngModulusTextbox.TextChanged += new
System.EventHandler(this.youngModulusTextbox_TextChanged);
label1
Set this.label1.AutoSize to true
Set this.label1.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 22
Set this.label1.Text to "YOUNG MODULUS:"
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 30
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent

```

```

Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ), ( ( int ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 31
Set this.BackToMainMenu.TabStop to false
Set this.BackToMainMenu.Text to "BACK"
Set this.BackToMainMenu.UseVisualStyleBackColor to false
This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 33
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 32
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
membersTable
Set this.membersTable.BackColor to System.Drawing.Color.Transparent
Set this.membersTable.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.membersTable.FlatAppearance.BorderSize to 2
Set this.membersTable.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ), ( ( int ) )
Set this.membersTable.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.membersTable.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.membersTable.Name to "membersTable"
Create new System.Drawing.Size
Set this.membersTable.TabIndex to 34
Set this.membersTable.TabStop to false
Set this.membersTable.Text to "TABLE"
Set this.membersTable.UseVisualStyleBackColor to false
This.membersTable.Click += new System.EventHandler(this.membersTable_Click);
materialProperties
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.membersTable
Call method this.Controls.Add with this.bridgeDrawing
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.addMaterialPropertiesButton

```

```
Call method this.Controls.Add with this.areaTextbox
Call method this.Controls.Add with this.label2
Call method this.Controls.Add with this.youngModulusTextbox
Call method this.Controls.Add with this.label1
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Call method this.Controls.Add with this.BackToMainMenu
Set this.DoubleBuffered to true
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "materialProperties"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout
```

### Flowchart



```

#endif
Attribute: minimize

private System.Windows.Forms.Button minimize;
Attribute: exitApplication

private System.Windows.Forms.Button exitApplication;
Attribute: addMaterialPropertiesButton

private System.Windows.Forms.Button addMaterialPropertiesButton;
Attribute: areaTextbox

private System.Windows.Forms.TextBox areaTextbox;
Attribute: label2

private System.Windows.Forms.Label label2;
Attribute: youngModulusTextbox

private System.Windows.Forms.TextBox youngModulusTextbox;
Attribute: label1

private System.Windows.Forms.Label label1;
Attribute: moveMenu

private System.Windows.Forms.Panel moveMenu;
Attribute: BackToMainMenu

private System.Windows.Forms.Button BackToMainMenu;
Attribute: bridgeDrawing

private System.Windows.Forms.Panel bridgeDrawing;
Attribute: zoomInBar

private System.Windows.Forms.TrackBar zoomInBar;
Attribute: membersTable

private System.Windows.Forms.Button membersTable;
Code File: materialPropertiesTable.cs

using Sectrics_V2.Properties;
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: materialPropertiesTable**

```
public partial class materialPropertiesTable : Form
```

**Attribute: zoom**

```
float zoom = 1f;
```

**Attribute: xMouseOffset**

```
double xMouseOffset;
```

**Attribute: yMouseOffset**

```
double yMouseOffset;
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

**Procedure: Constructor**

```
public materialPropertiesTable()
```

Returns materialPropertiesTable

**Pseudocode**

Call method InitializeComponent

Adds The Array Coordinates Into ListBox

**For** i is 0, i is less than Program.bridgeData.memberConnection.Count, i increments by 1

    Call method membersListView.Items.Add with "To Member: " plus Program.bridgeData.memberConnection plus " | From Member: " plus Program.bridgeData.memberConnection

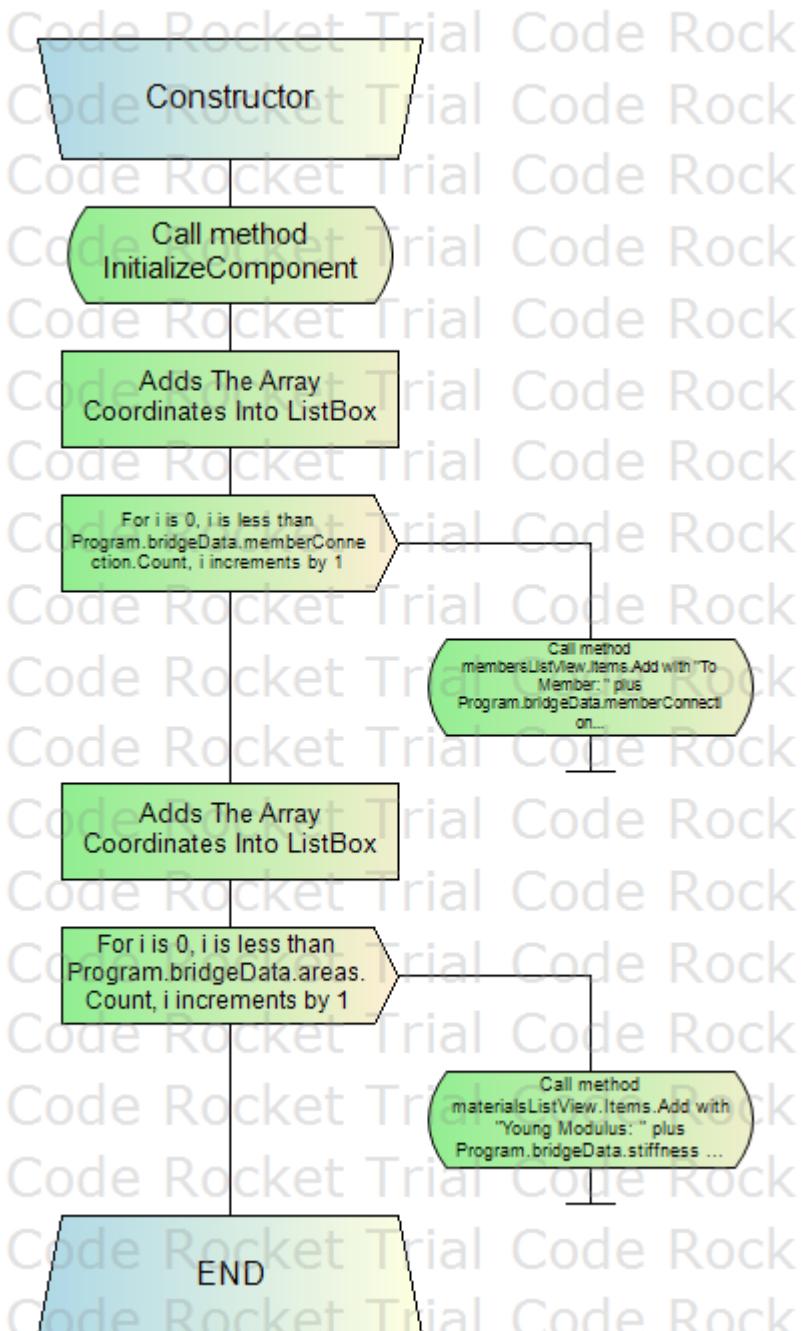
**EndFor**

Adds The Array Coordinates Into ListBox

**For** i is 0, i is less than Program.bridgeData.areas.Count, i increments by 1  
    Call method materialsListView.Items.Add with "Young Modulus: " plus Program.bridgeData.stiffness plus " | Cross-Sectional Area: " plus Program.bridgeData.areas

**EndFor**

Flowchart



**Procedure: WndProc**

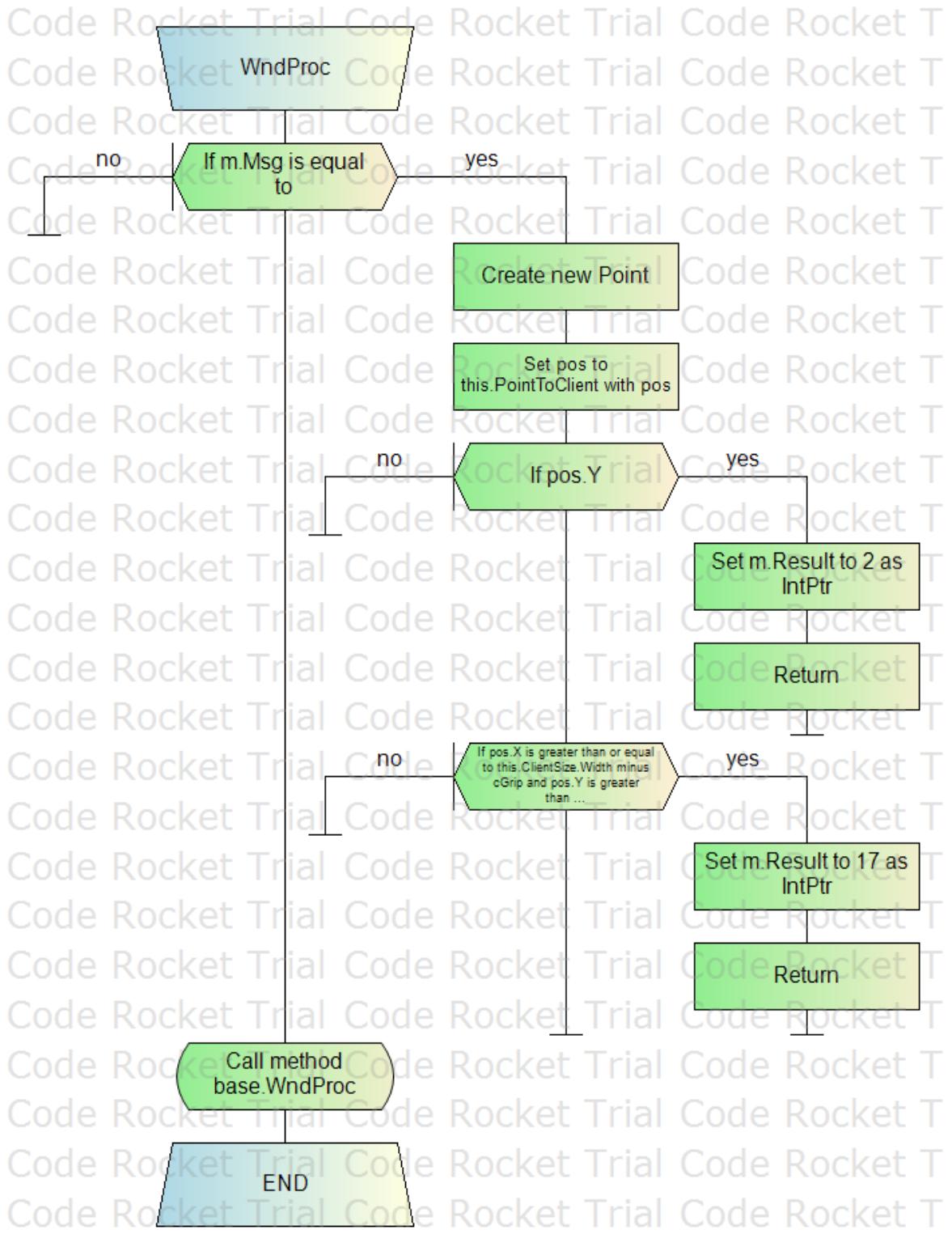
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

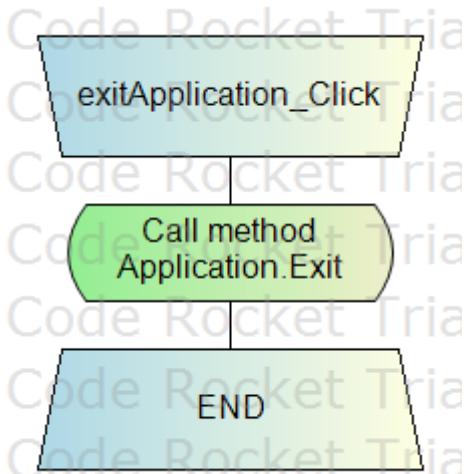
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

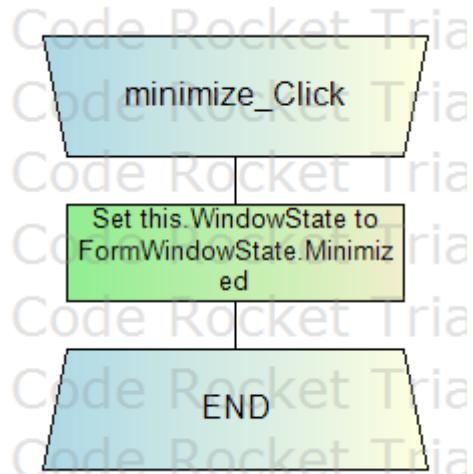
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

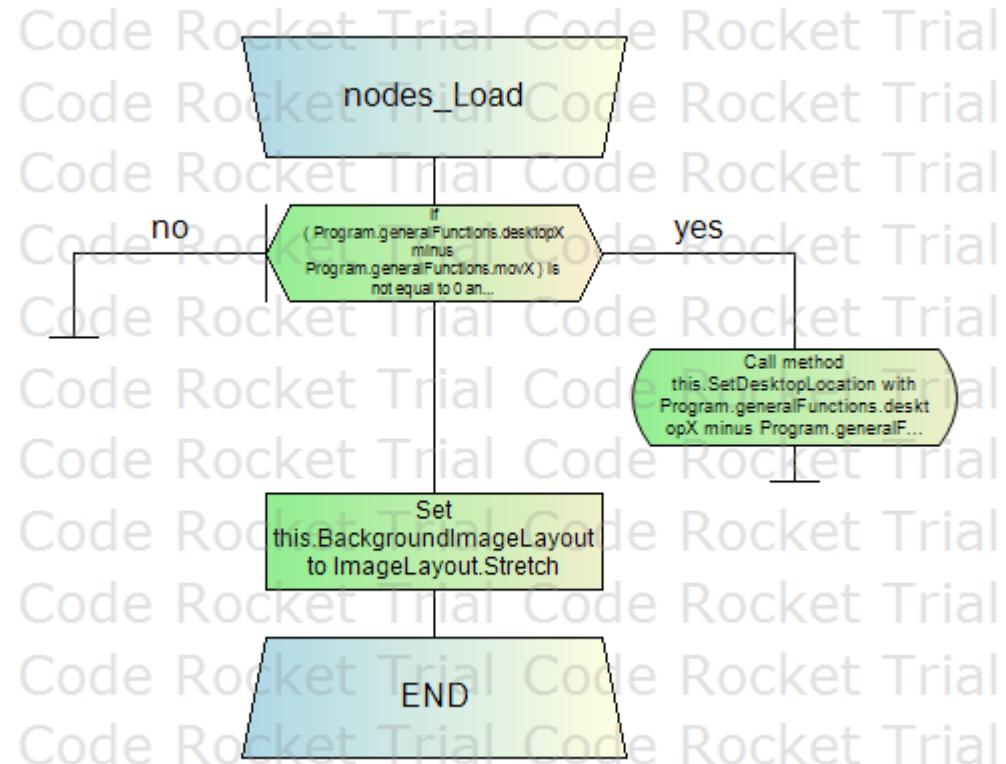
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

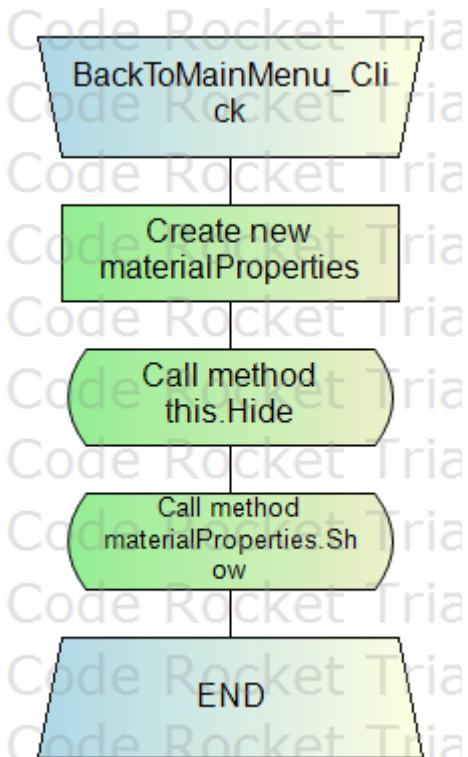
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new materialProperties  
Call method this.Hide  
Call method materialProperties.Show
```

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

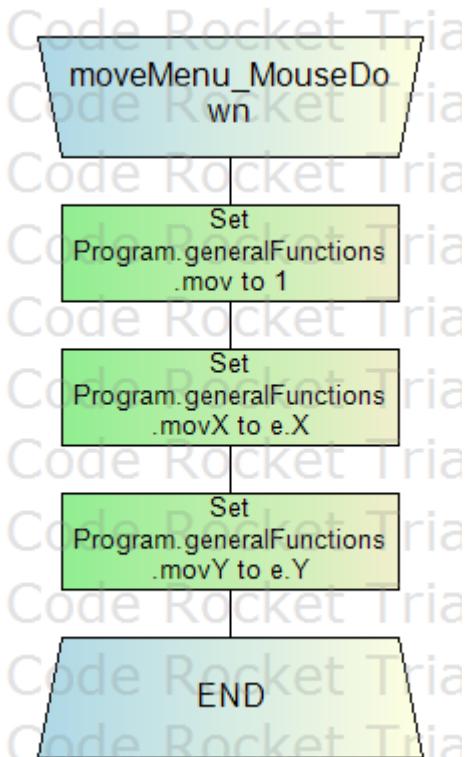
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

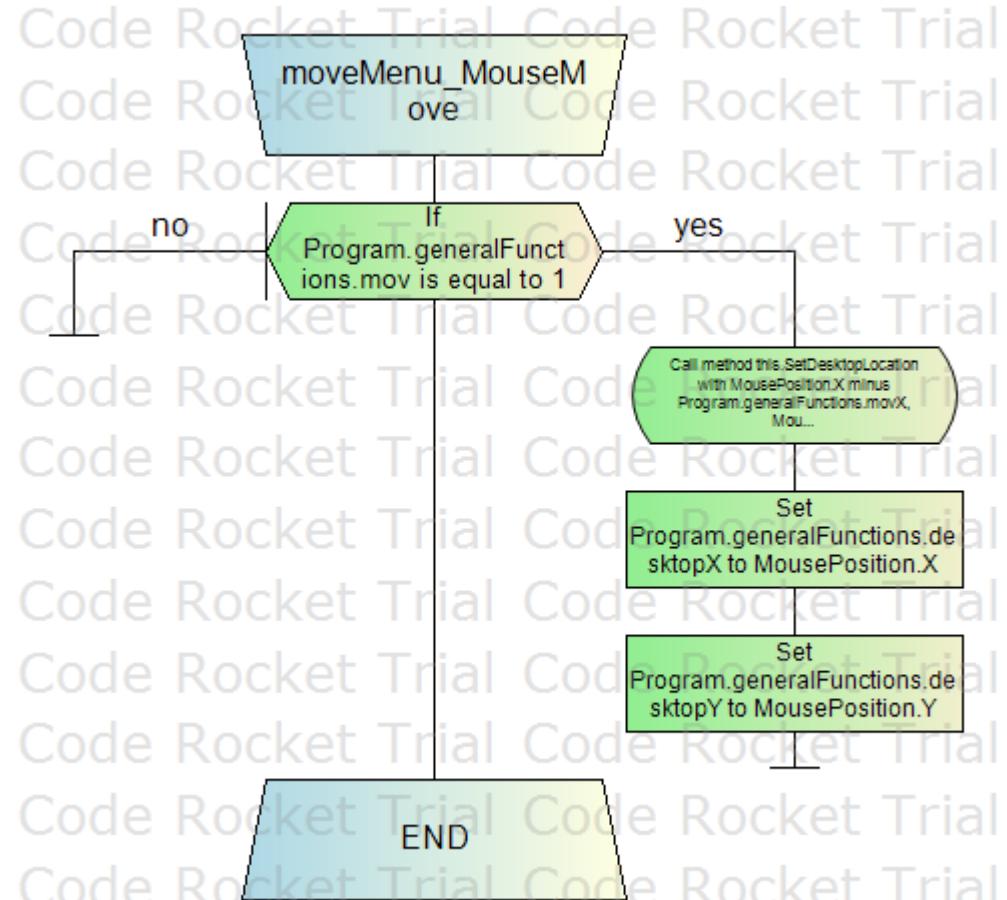
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseUp**

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

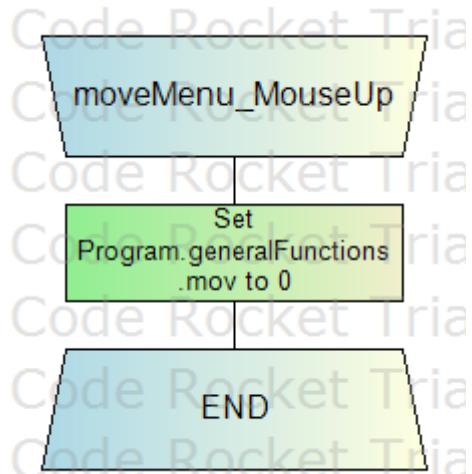
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: clearAll\_Click**

```
private void clearAll_Click(object sender, EventArgs e)
```

**Parameters**

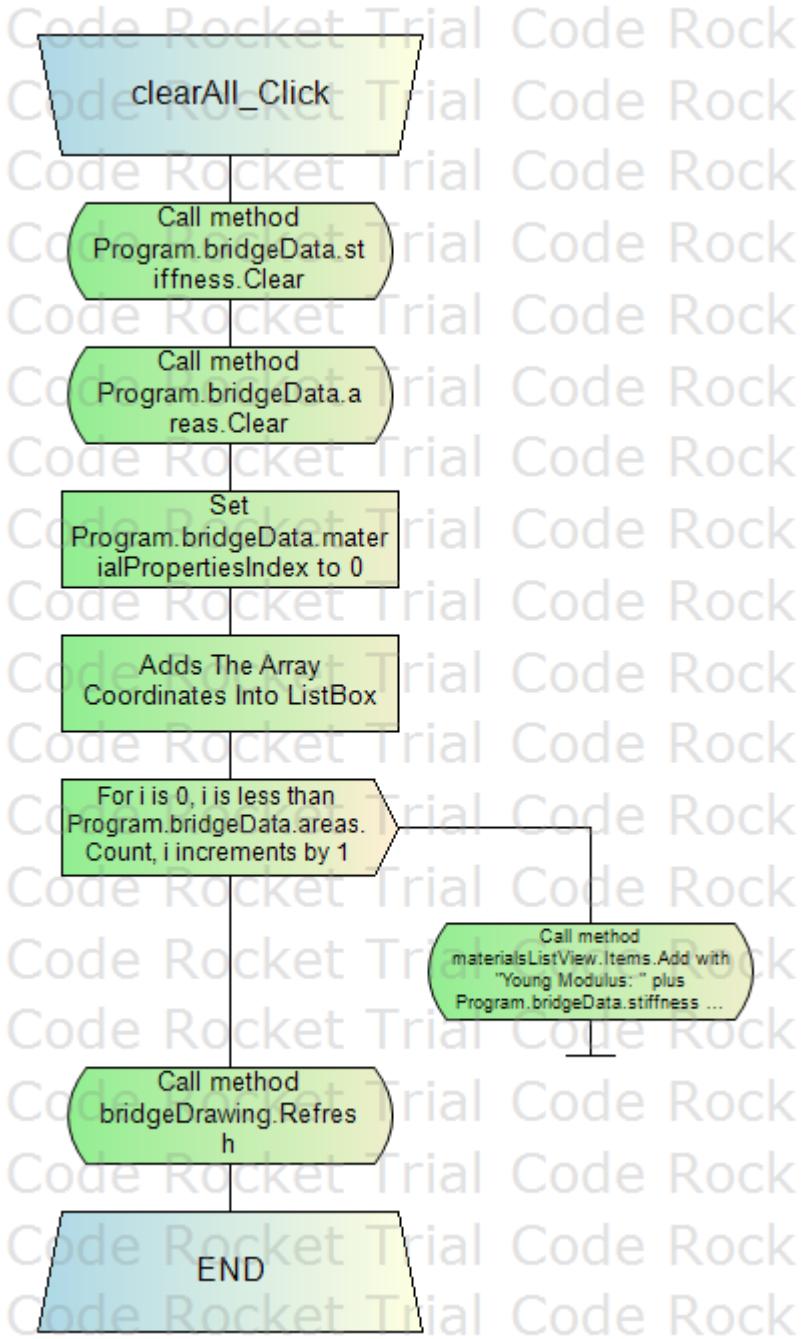
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Program.bridgeData.stiffness.Clear  
Call method Program.bridgeData.areas.Clear  
Set Program.bridgeData.materialPropertiesIndex to 0  
Adds The Array Coordinates Into ListBox  
For i is 0, i is less than Program.bridgeData.areas.Count, i increments by 1  
    Call method materialsListView.Items.Add with "Young Modulus: " plus  
    Program.bridgeData.stiffness plus " | Cross-Sectional Area: " plus  
    Program.bridgeData.areas  
EndFor  
Call method bridgeDrawing.Refresh
```

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

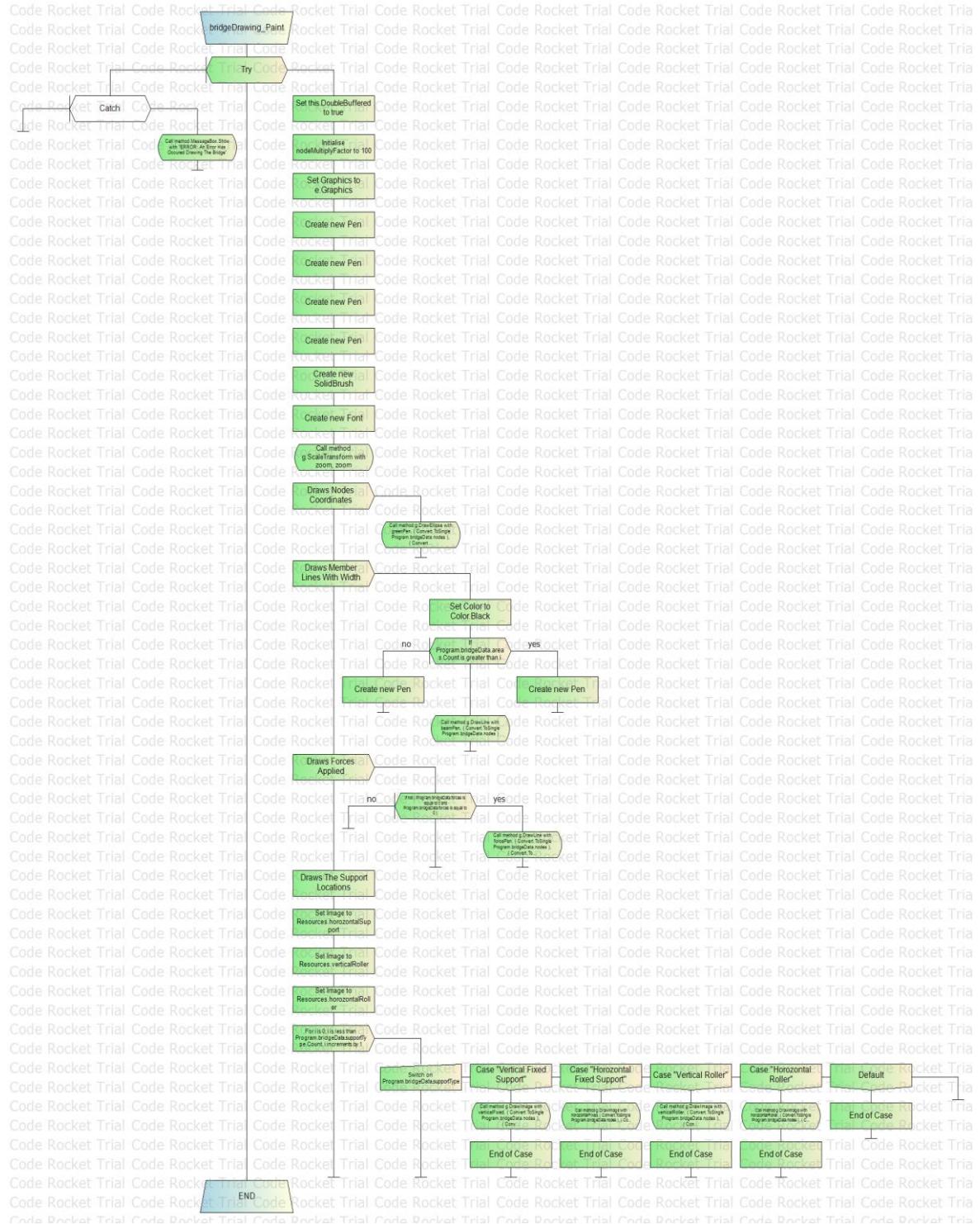
**Pseudocode**

**Try**

```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```
        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry
```

## Flowchart



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

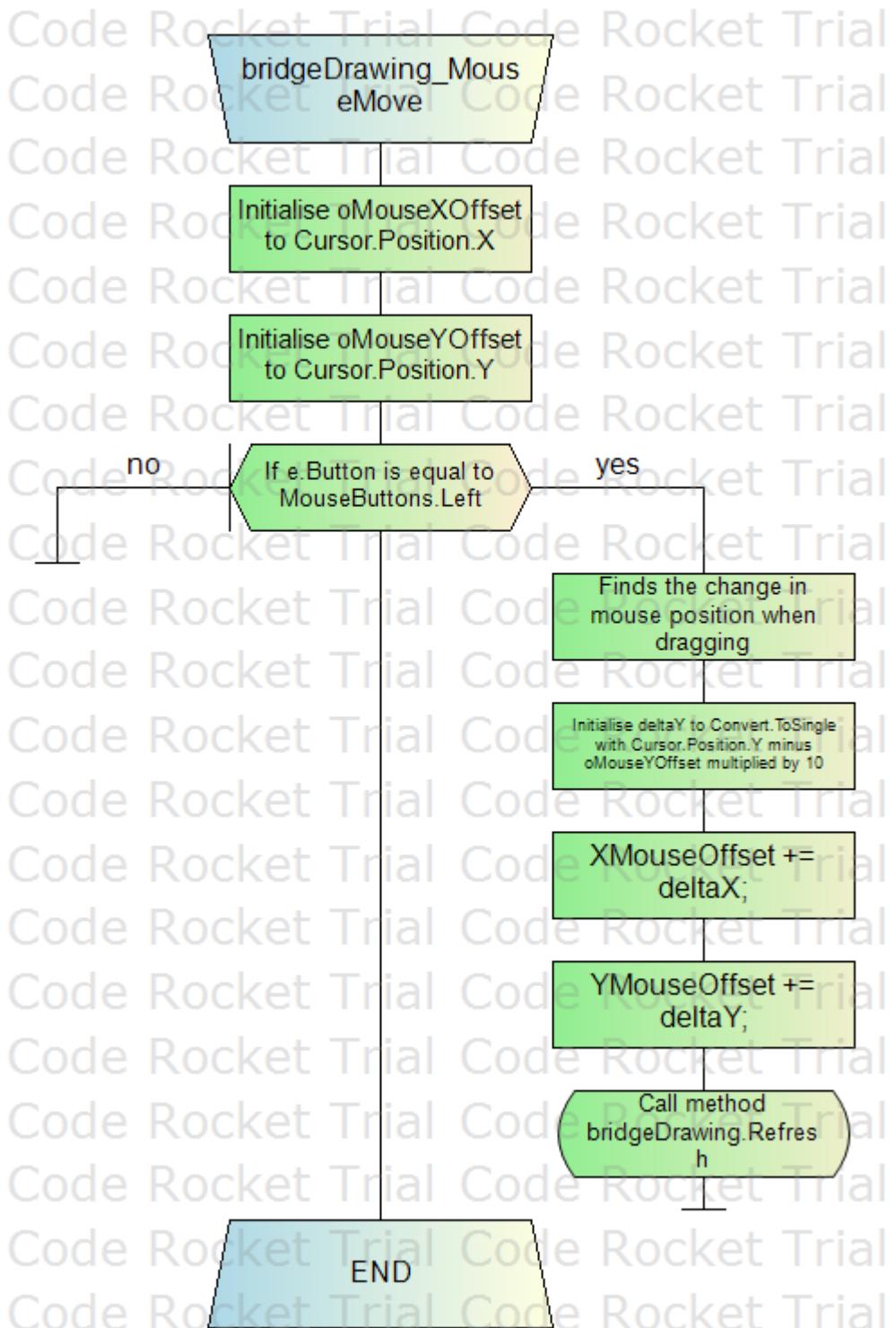
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

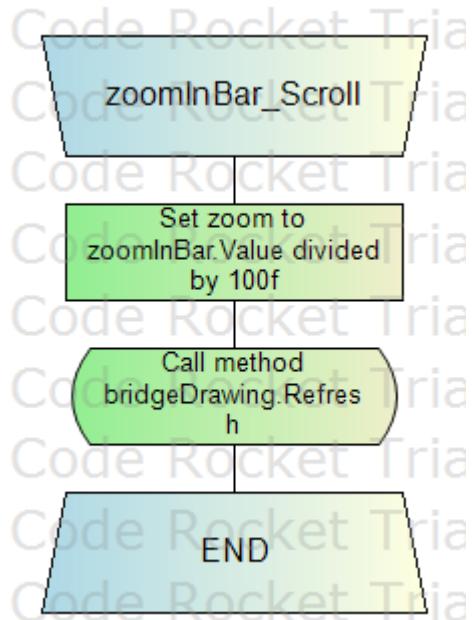
Returns Void

**Pseudocode**

```
Set zoom to zoomInBar.Value divided by 100f
```

```
Call method bridgeDrawing.Refresh
```

**Flowchart**



*Procedure: materialsListView\_SelectedIndexChanged*

```
private void materialsListView_SelectedIndexChanged(object sender,  
EventArgs e)
```

**Parameters**

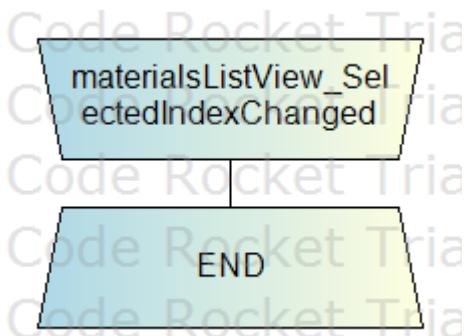
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Code File: materialPropertiesTable.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

*Class: materialPropertiesTable*

```
partial class materialPropertiesTable
```

*Attribute: components*

```
private System.ComponentModel.IContainer components = null;
```

Required designer variable.

*Procedure: Dispose*

```
protected override void Dispose (bool disposing)
```

Clean up any resources being used.

*Parameters*

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

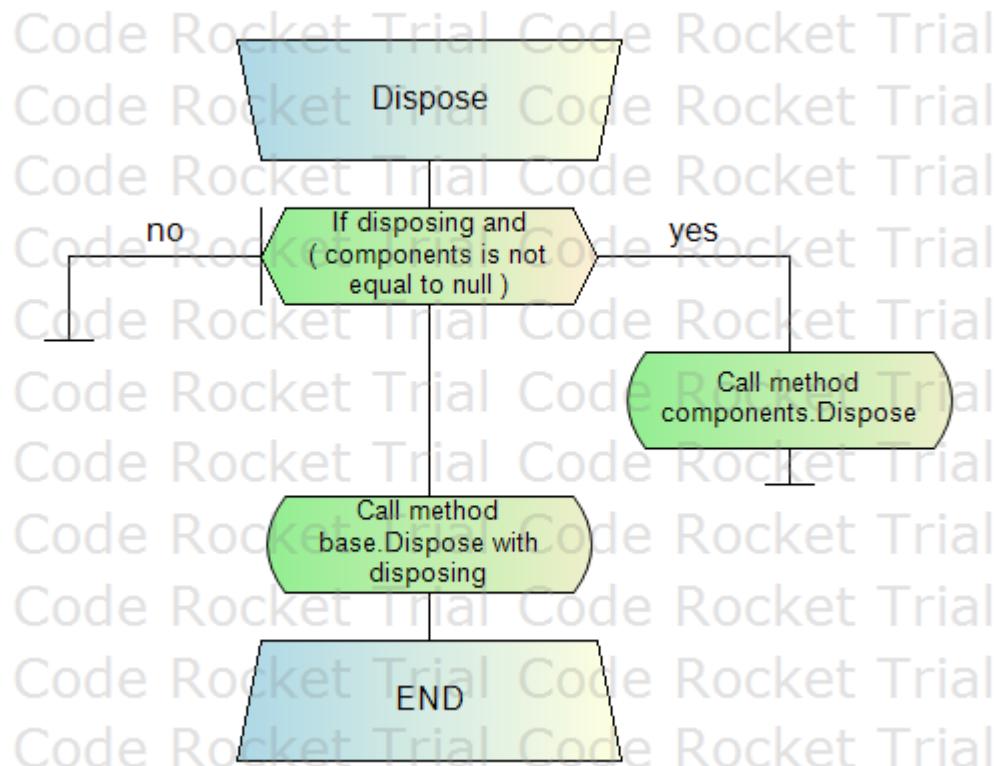
*Pseudocode*

```
If disposing and ( components is not equal to null )  
    Call method components.Dispose
```

*EndIf*

```
Call method base.Dispose with disposing
```

Flowchart



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.ListBox
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.ListBox
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent
Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( int ), ( int ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 25
Set this.BackToMainMenu.TabStop to false
Set this.BackToMainMenu.Text to "BACK"

```

```

Set this.BackToMainMenu.UseVisualStyleBackColor to false
This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 35
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 34
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 36
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
membersListView
Set this.membersListView.FormattingEnabled to true
Set this.membersListView.ItemHeight to 25
Create new System.Drawing.Point
Set this.membersListView.Name to "membersListView"
Create new System.Drawing.Size
Set this.membersListView.TabIndex to 37
clearAll
Set this.clearAll.BackColor to System.Drawing.Color.Transparent
Set this.clearAll.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.clearAll.FlatAppearance.BorderSize to 2
Set this.clearAll.FlatAppearance.MouseOverBackColor to System.Drawing.Color.FromArgb
with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.clearAll.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.clearAll.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.clearAll.Name to "clearAll"
Create new System.Drawing.Size
Set this.clearAll.TabIndex to 39
Set this.clearAll.TabStop to false
Set this.clearAll.Text to "CLEAR ALL"
Set this.clearAll.UseVisualStyleBackColor to false
This.clearAll.Click += new System.EventHandler(this.clearAll_Click);
materialsListView
Set this.materialsListView.FormattingEnabled to true
Set this.materialsListView.ItemHeight to 25
Create new System.Drawing.Point

```

```

Set this.materialsListView.Name to "materialsListView"
Create new System.Drawing.Size
Set this.materialsListView.TabIndex to 38
This.materialsListView.SelectedIndexChanged += new
System.EventHandler(this.materialsListView_SelectedIndexChanged);
materialPropertiesTable
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.membersListView
Call method this.Controls.Add with this.clearAll
Call method this.Controls.Add with this.materialsListView
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.bridgeDrawing
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.BackToMainMenu
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "materialPropertiesTable"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout

```

## Flowchart



```
#endregion

Attribute: minimize

private System.Windows.Forms.Button minimize;
Attribute: exitApplication

private System.Windows.Forms.Button exitApplication;
Attribute: BackToMainMenu

private System.Windows.Forms.Button BackToMainMenu;
Attribute: bridgeDrawing

private System.Windows.Forms.Panel bridgeDrawing;
Attribute: zoomInBar

private System.Windows.Forms.TrackBar zoomInBar;
Attribute: moveMenu

private System.Windows.Forms.Panel moveMenu;
Attribute: membersListView

private System.Windows.Forms.ListBox membersListView;
Attribute: clearAll

private System.Windows.Forms.Button clearAll;
Attribute: materialsListView

private System.Windows.Forms.ListBox materialsListView;
Code File: Maths.cs

using System;
Namespace: Sectrics_V2

namespace Sectrics_V2
Class: Maths

class Maths
Normal Maths Formulas
Procedure: distanceFormula
public double distanceFormula(double x1, double y1, double x2, double y2)
Parameters

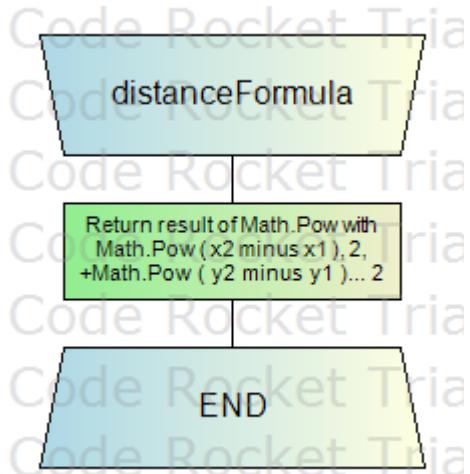
- x1 - Double
- y1 - Double
- x2 - Double
- y2 - Double

Returns Double
```

### Pseudocode

```
Return result of Math.Pow with Math.Pow ( x2 minus x1 ), 2, +Math.Pow ( y2 minus y1 )... 2
```

### Flowchart



## Matrix Operations

### Procedure: vectorMatrixDotProduct

```
public double[] vectorMatrixDotProduct(double[,] mainMatrix, double[]
mainVector)
```

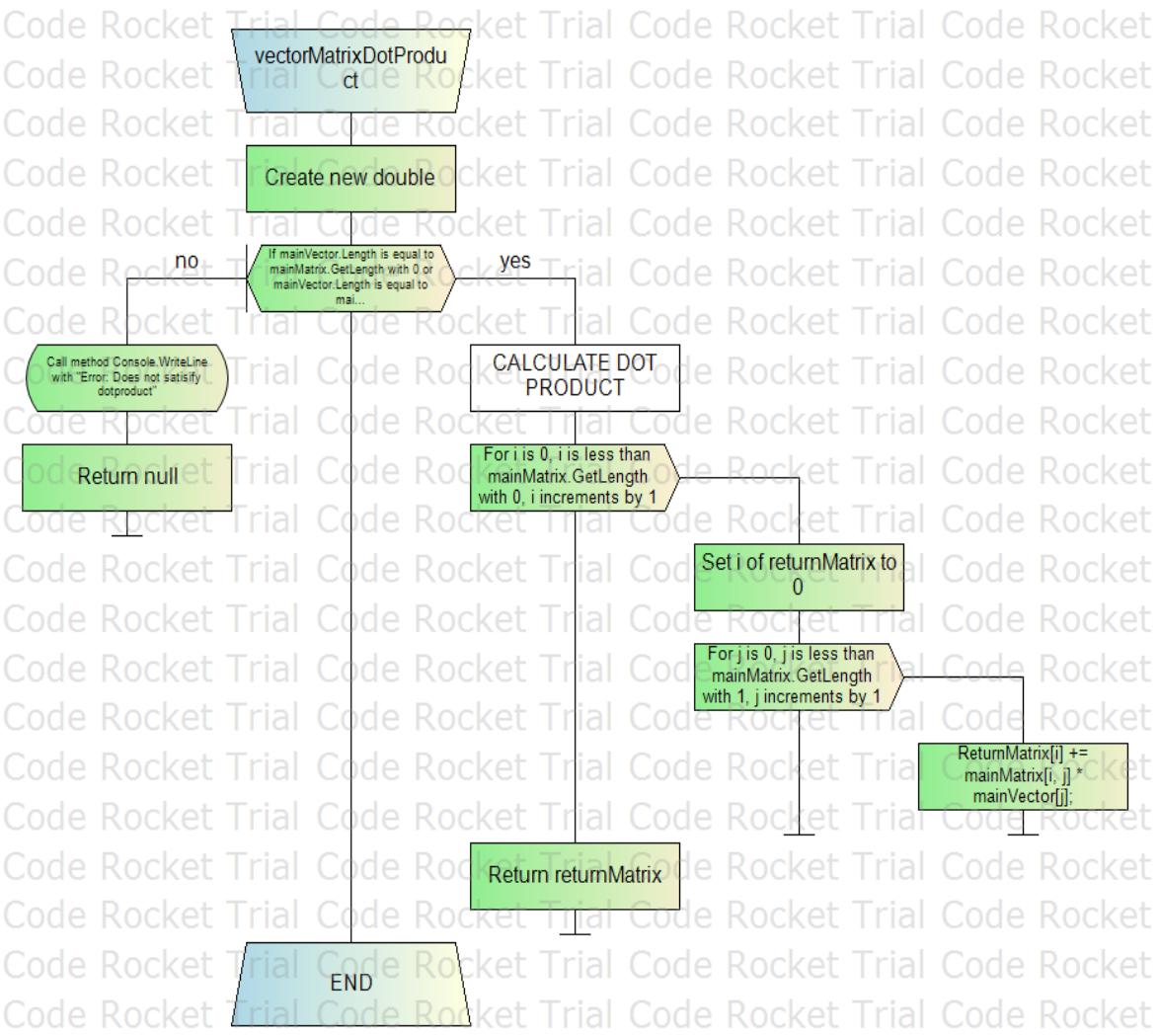
#### Parameters

- mainMatrix - No type
- mainVector - No type

#### Pseudocode

```
Create new double
If mainVector.Length is equal to mainMatrix.GetLength with 0 or mainVector.Length is
equal to mainMatrix.GetLength with 1
    CALCULATE DOT PRODUCT
    For i is 0, i is less than mainMatrix.GetLength with 0, i increments by 1
        Set i of returnMatrix to 0
        For j is 0, j is less than mainMatrix.GetLength with 1, j increments by
1
            ReturnMatrix[i] += mainMatrix[i, j] * mainVector[j];
        EndFor
    EndFor
    Return returnMatrix
Else
    Call method Console.WriteLine with "Error: Does not satisfy dotproduct"
    Return null
EndIf
```

#### Flowchart



### Procedure: vectorVectorRemove

```
public double[] vectorVectorRemove(double[] mainVector, int[] row)
```

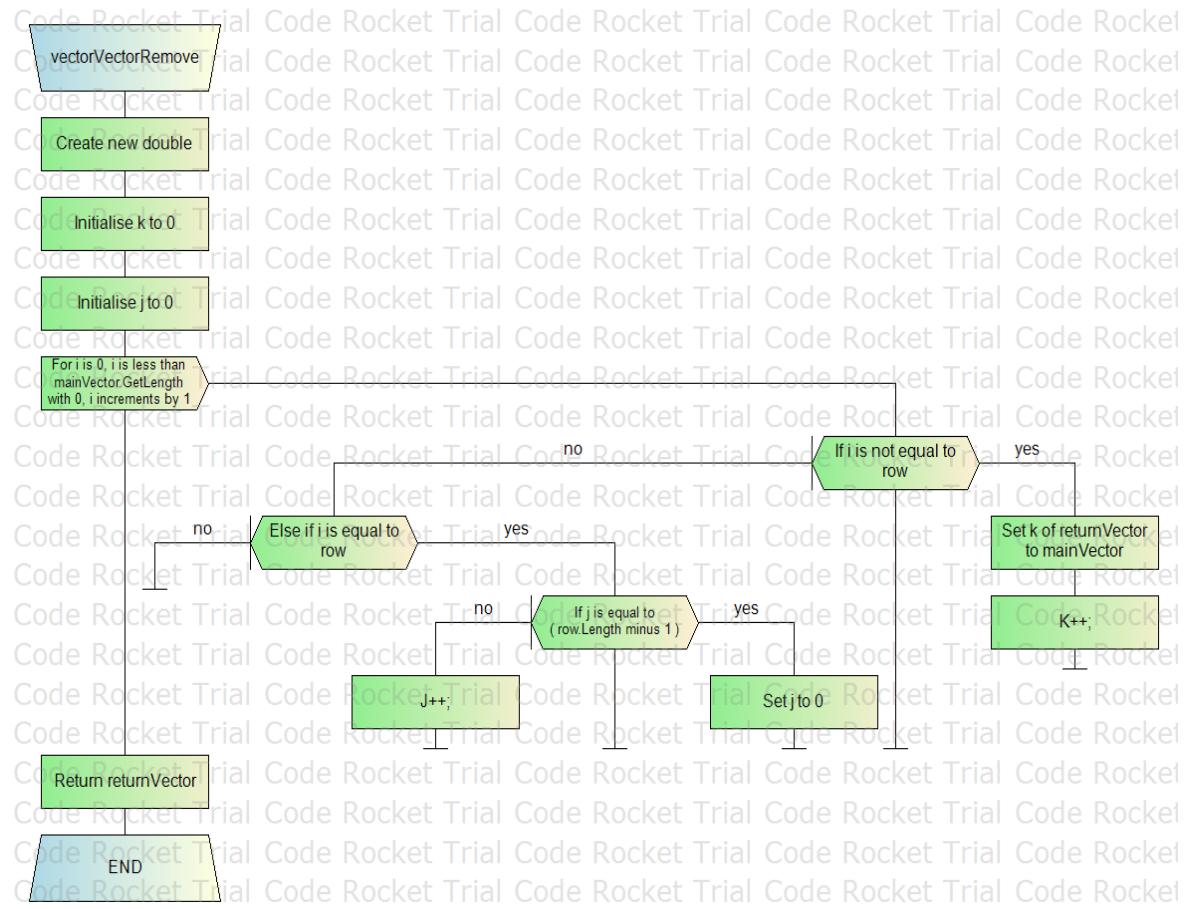
Parameters

- mainVector - No type
- row - No type

Pseudocode

```
Create new double
Initialise k to 0
Initialise j to 0
For i is 0, i is less than mainVector.GetLength with 0, i increments by 1
    If i is not equal to row
        Set k of returnVector to mainVector
        K++;
    Else if i is equal to row
        If j is equal to ( row.Length minus 1 )
            Set j to 0
        Else
            J++;
        EndIf
    EndIf
EndFor
Return returnVector
```

### Flowchart



**Procedure: vectorMatrixRemove**

```
public double[,] vectorMatrixRemove(double[,] mainMatrix, int[] rows,  
int[] columns)
```

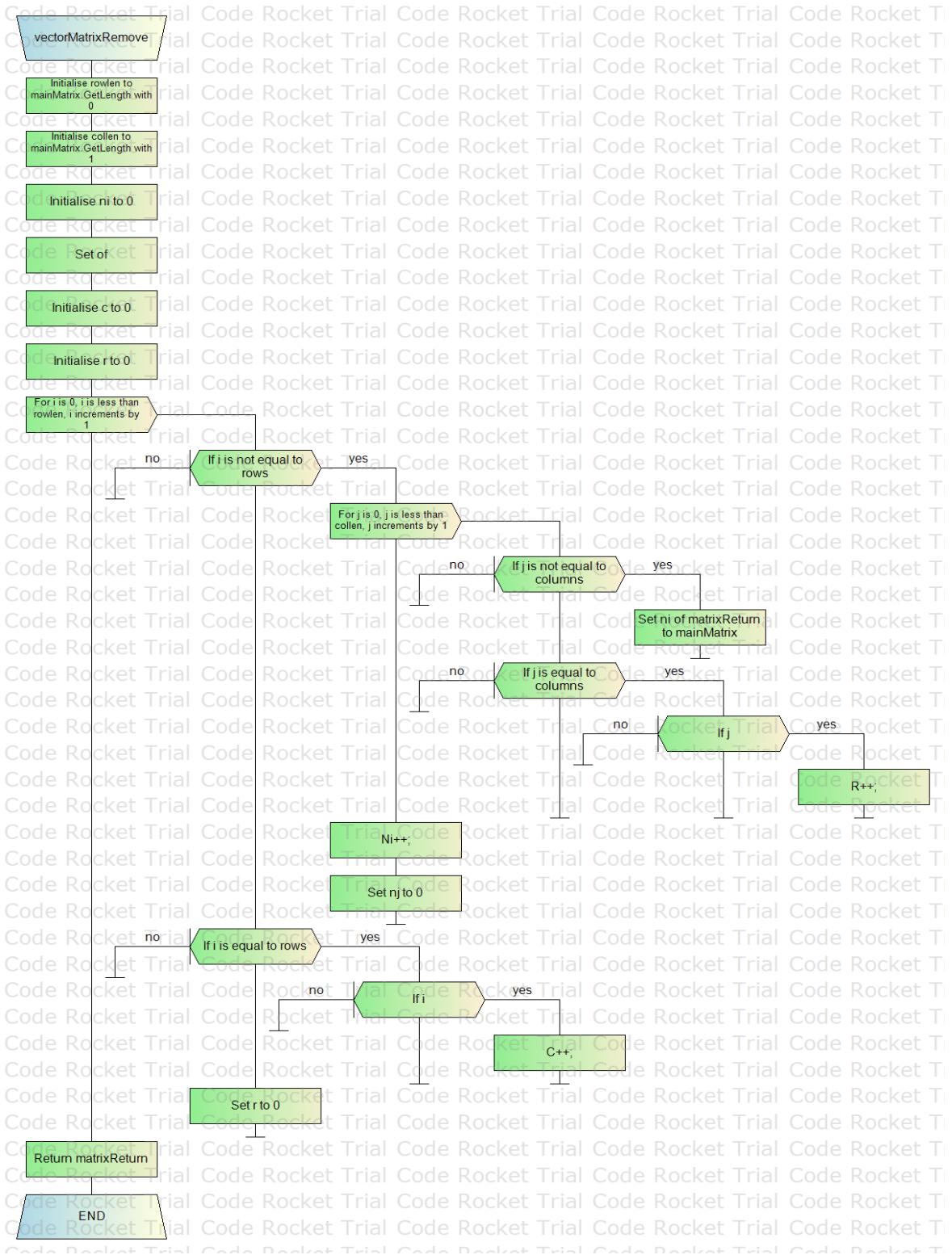
**Parameters**

- mainMatrix - No type
- rows - No type
- columns - No type

**Pseudocode**

```
Initialise rowlen to mainMatrix.GetLength with 0  
Initialise collen to mainMatrix.GetLength with 1  
Initialise ni to 0  
Set of  
Initialise c to 0  
Initialise r to 0  
For i is 0, i is less than rowlen, i increments by 1  
    If i is not equal to rows  
        For j is 0, j is less than collen, j increments by 1  
            If j is not equal to columns  
                Set ni of matrixReturn to mainMatrix  
            EndIf  
            If j is equal to columns  
                If j  
                    R++;  
                EndIf  
            EndIf  
        EndFor  
        Ni++;  
        Set nj to 0  
    EndIf  
    If i is equal to rows  
        If i  
            C++;  
        EndIf  
    EndIf  
    Set r to 0  
EndFor  
Return matrixReturn
```

**Flowchart**



**Procedure: vectorAdditionByOne**

```
public int[] vectorAdditionByOne(int value, int[] vector)
```

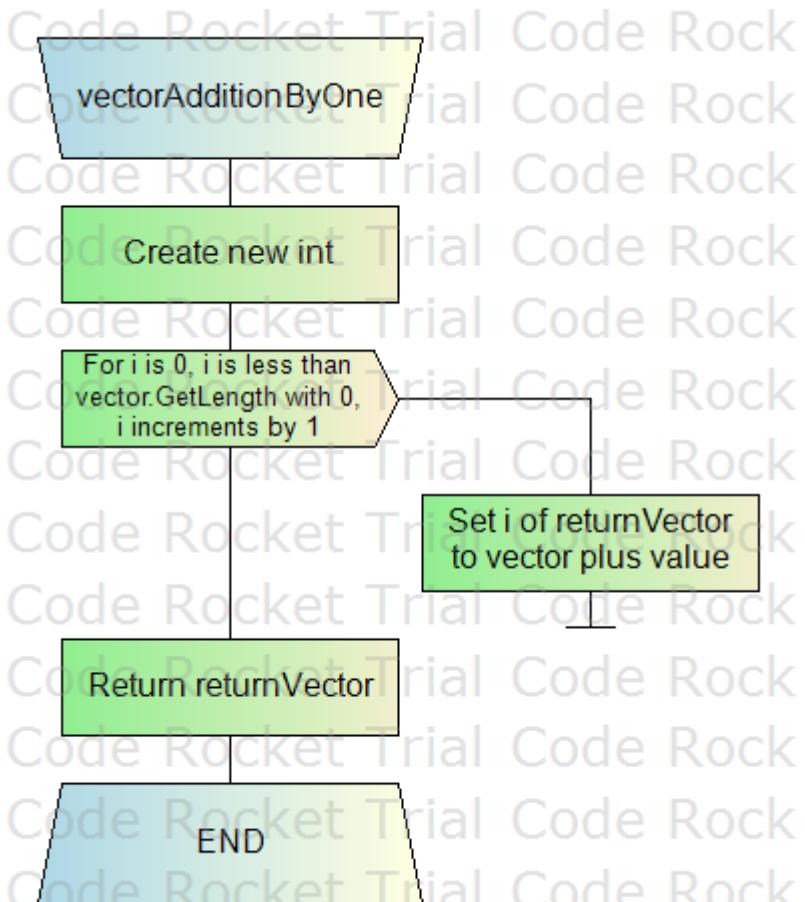
**Parameters**

- - Integer

**Pseudocode**

```
Create new int  
For i is 0, i is less than vector.GetLength with 0, i increments by 1  
    Set i of returnVector to vector plus value  
EndFor  
Return returnVector
```

**Flowchart**



**Procedure: vectorDisplay**

```
public void vectorDisplay(double[] vector)
```

**Parameters**

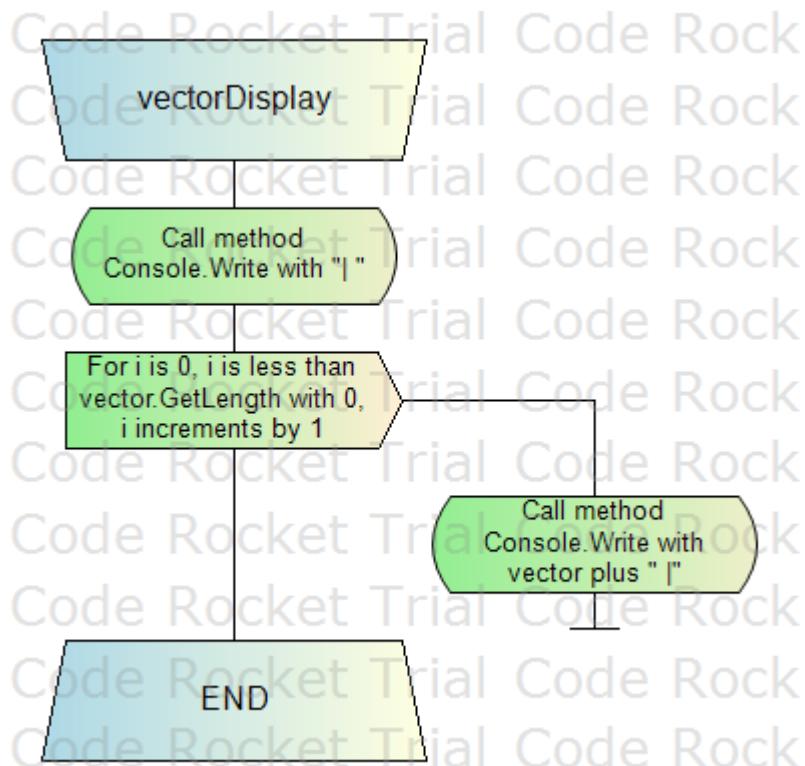
- vector - No type

Returns Void

**Pseudocode**

```
Call method Console.Write with "| "
For i is 0, i is less than vector.GetLength with 0, i increments by 1
    Call method Console.Write with vector plus " |"
EndFor
```

**Flowchart**



**Procedure: matrixAddition**

```
public double[,] matrixAddition(double[,] MatrixOne, double[,]  
MatrixTwo)
```

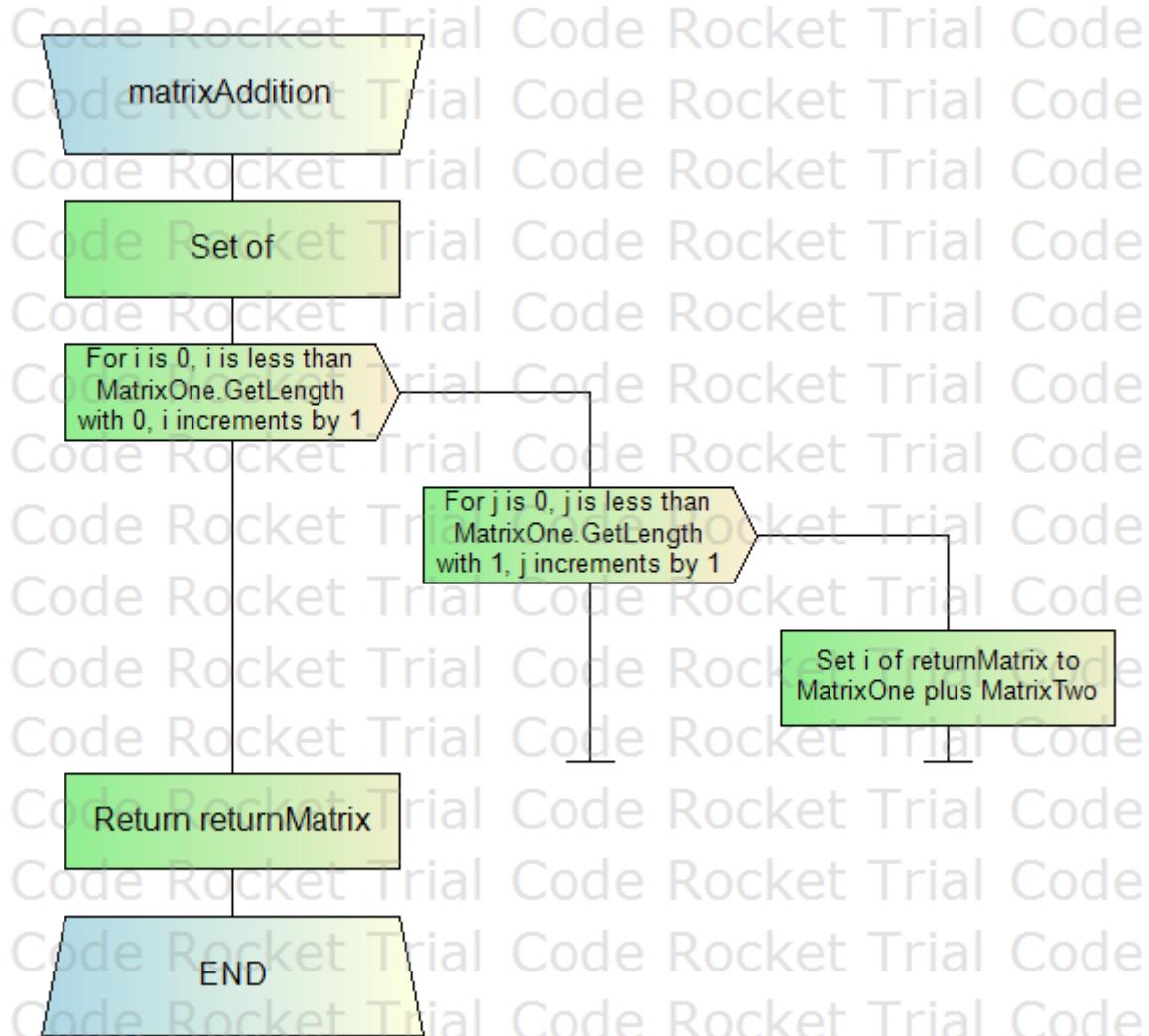
**Parameters**

- MatrixOne - No type
- MatrixTwo - No type

**Pseudocode**

```
Set of  
For i is 0, i is less than MatrixOne.GetLength with 0, i increments by 1  
    For j is 0, j is less than MatrixOne.GetLength with 1, j increments by 1  
        Set i of returnMatrix to MatrixOne plus MatrixTwo  
    EndFor  
EndFor  
Return returnMatrix
```

**Flowchart**



**Procedure: dotProduct**

```
public double dotProduct(double[] vec1, double[] vec2)
```

**Parameters**

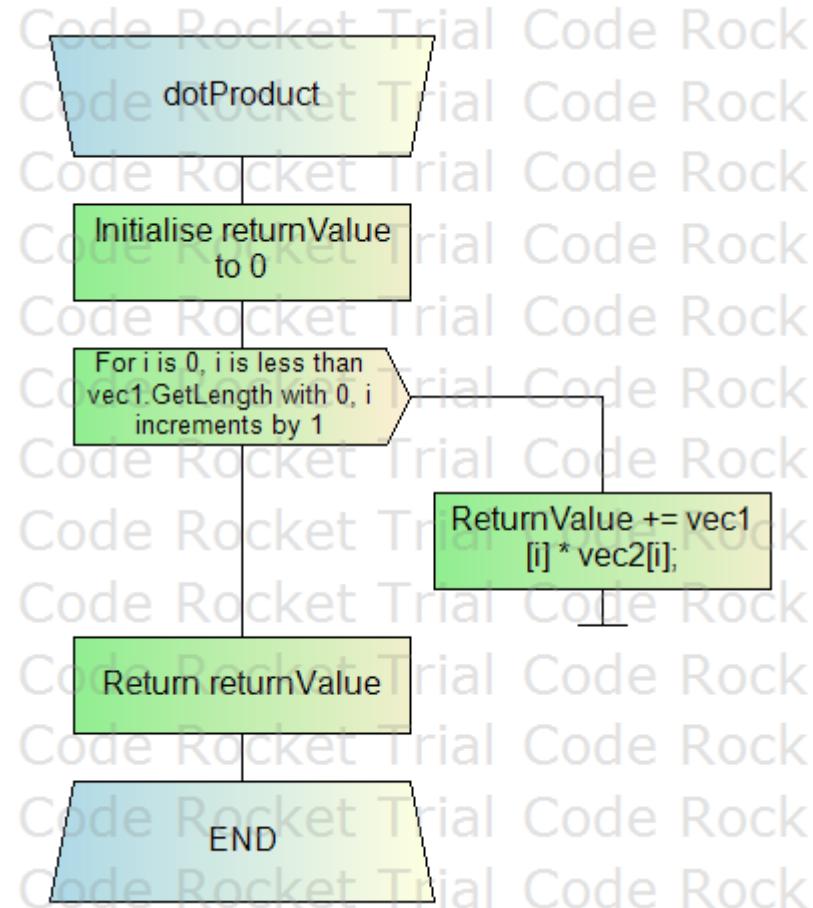
- vec1 - No type
- vec2 - No type

Returns Double

**Pseudocode**

```
Initialise returnValue to 0
For i is 0, i is less than vec1.GetLength with 0, i increments by 1
    ReturnValue += vec1[i] * vec2[i];
EndFor
Return returnValue
```

**Flowchart**



**Procedure: zeroMatrix**

```
public double[,] zeroMatrix(int columns, int width)
```

**Parameters**

- columns - Integer
- width - Integer

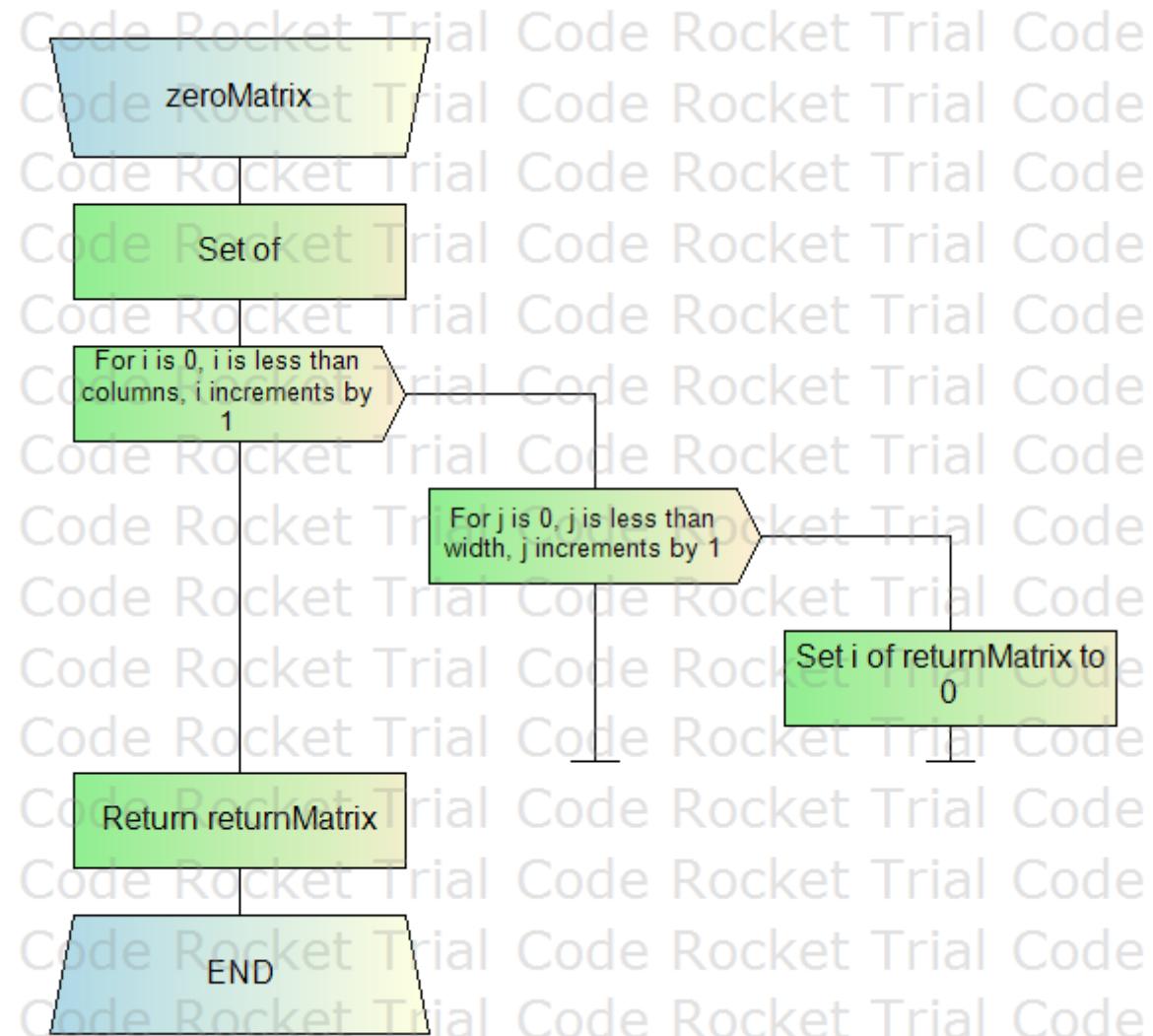
**Pseudocode**

Set of

```
For i is 0, i is less than columns, i increments by 1
    For j is 0, j is less than width, j increments by 1
        Set i of returnMatrix to 0
    EndFor
EndFor
```

Return returnMatrix

**Flowchart**



**Procedure: oneValueMultiplyMatrix**

```
public double[,] oneValueMultiplyMatrix(double oneValue, double[,]  
mainMatrix)
```

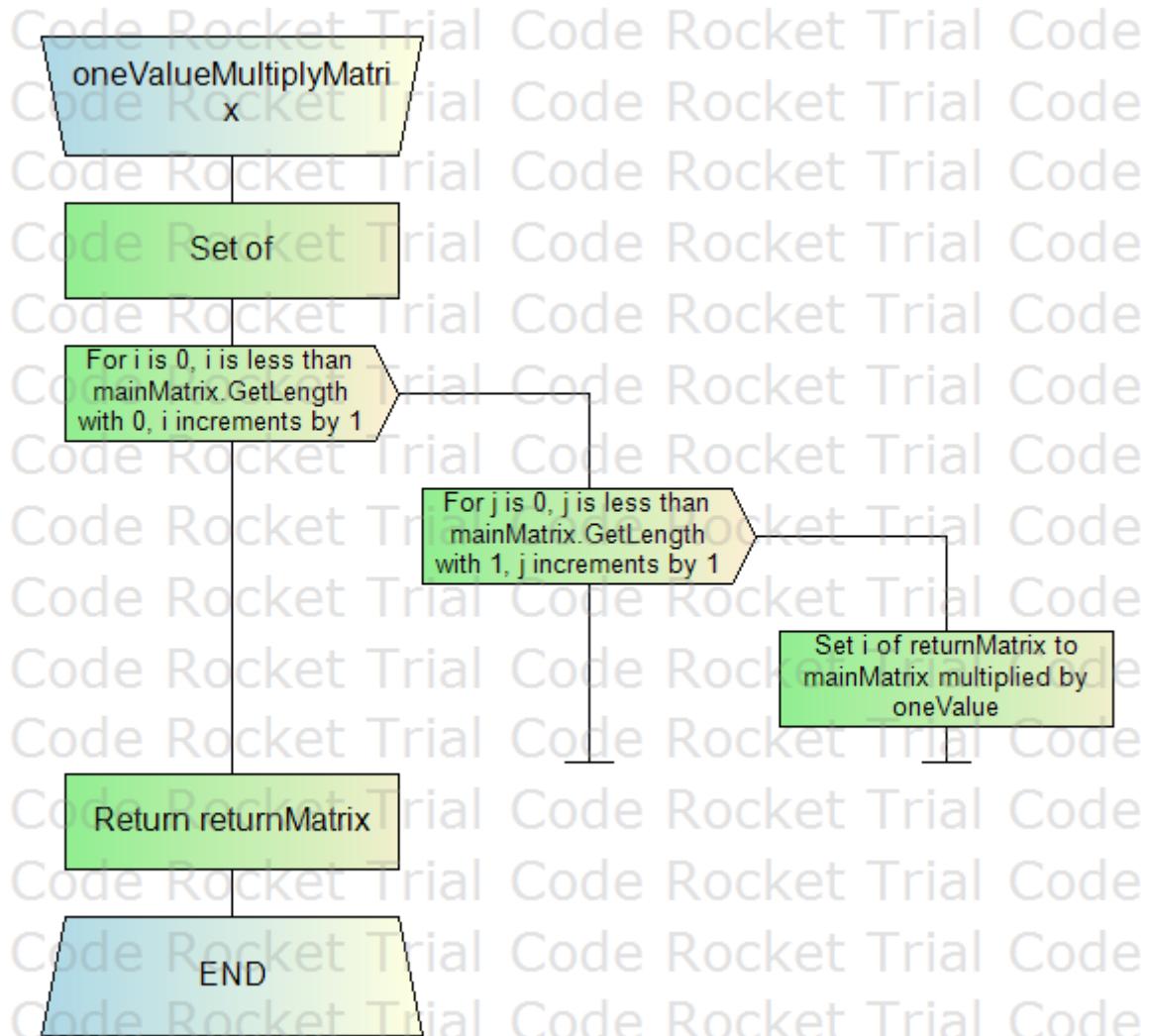
**Parameters**

- oneValue - Double
- mainMatrix - No type

**Pseudocode**

```
Set of  
For i is 0, i is less than mainMatrix.GetLength with 0, i increments by 1  
    For j is 0, j is less than mainMatrix.GetLength with 1, j increments by 1  
        Set i of returnMatrix to mainMatrix multiplied by oneValue  
    EndFor  
EndFor  
Return returnMatrix
```

**Flowchart**



**Procedure: matrixDisplay**

```
public void matrixDisplay(double[,] mainMatrix)
```

**Parameters**

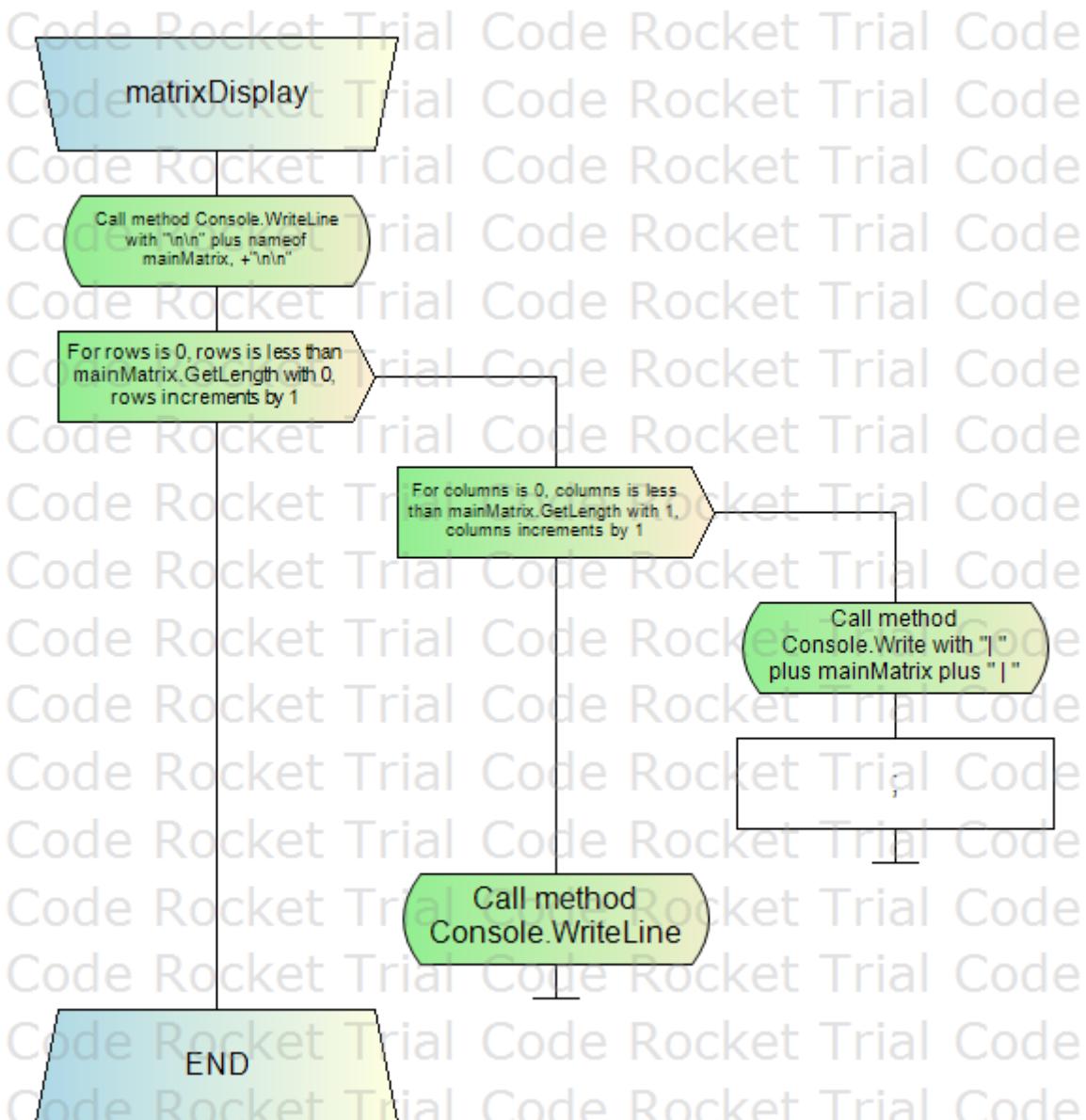
- mainMatrix - No type

Returns Void

**Pseudocode**

```
Call method Console.WriteLine with "\n\n" plus nameof mainMatrix, +"\n\n"
For rows is 0, rows is less than mainMatrix.GetLength with 0, rows increments by 1
    For columns is 0, columns is less than mainMatrix.GetLength with 1, columns
    increments by 1
        Call method Console.Write with "|" plus mainMatrix plus "|"
        ;
    EndFor
    Call method Console.WriteLine
EndFor
```

**Flowchart**



**Procedure: inverseMatrix**

```
public double[,] inverseMatrix(double[,] mainMatrix)
```

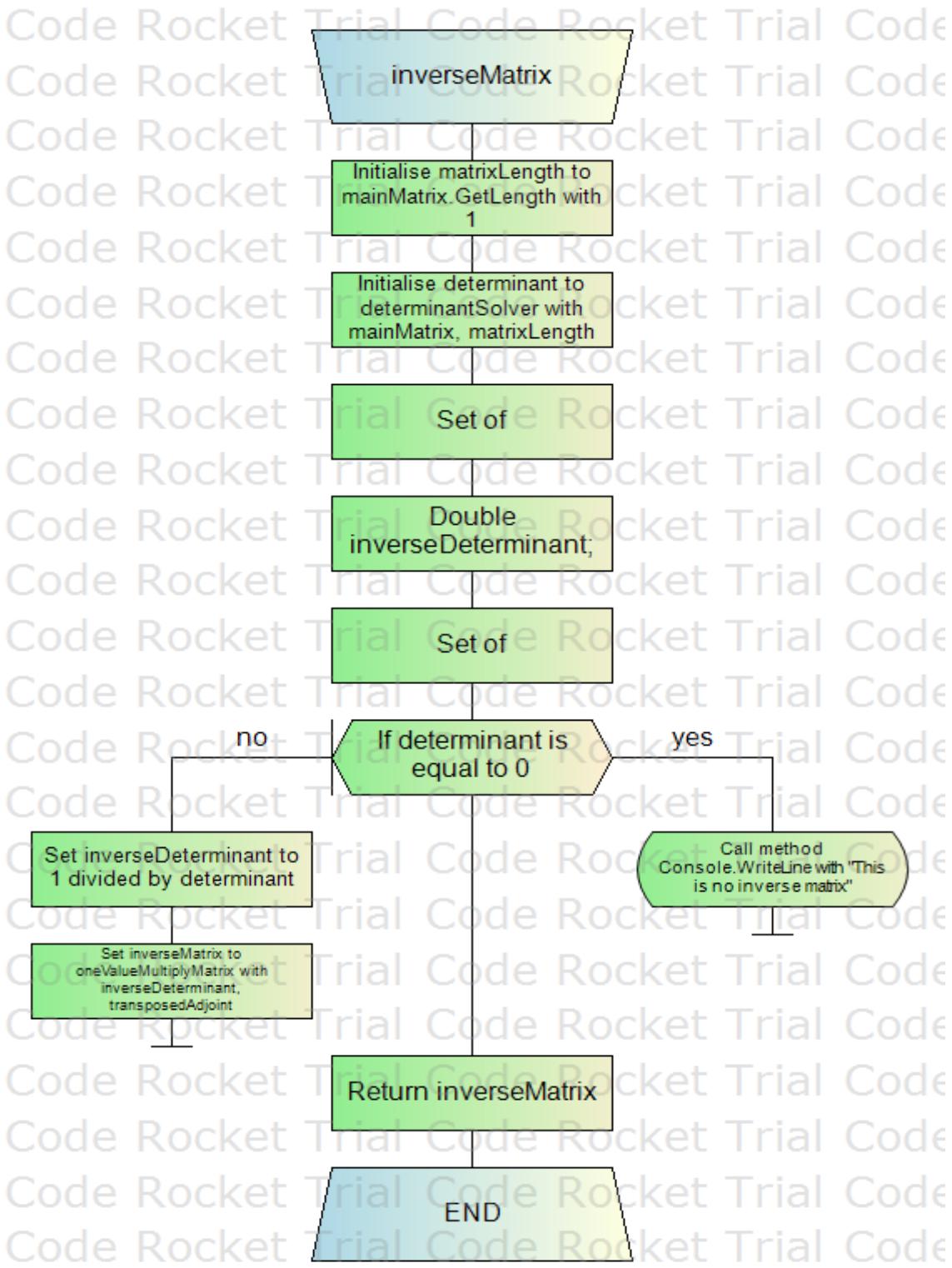
**Parameters**

- mainMatrix - No type

**Pseudocode**

```
Initialise matrixLength to mainMatrix.GetLength with 1
Initialise determinant to determinantSolver with mainMatrix, matrixLength
Set of
Double inverseDeterminant;
Set of
If determinant is equal to 0
    Call method Console.WriteLine with "This is no inverse matrix"
Else
    Set inverseDeterminant to 1 divided by determinant
    Set inverseMatrix to oneValueMultiplyMatrix with inverseDeterminant,
transposedAdjoint
EndIf
Return inverseMatrix
```

**Flowchart**



**Procedure: matrixMultiplication**

```
public double[,] matrixMultiplication(double[,] matrix1, double[,] matrix2)
```

**Parameters**

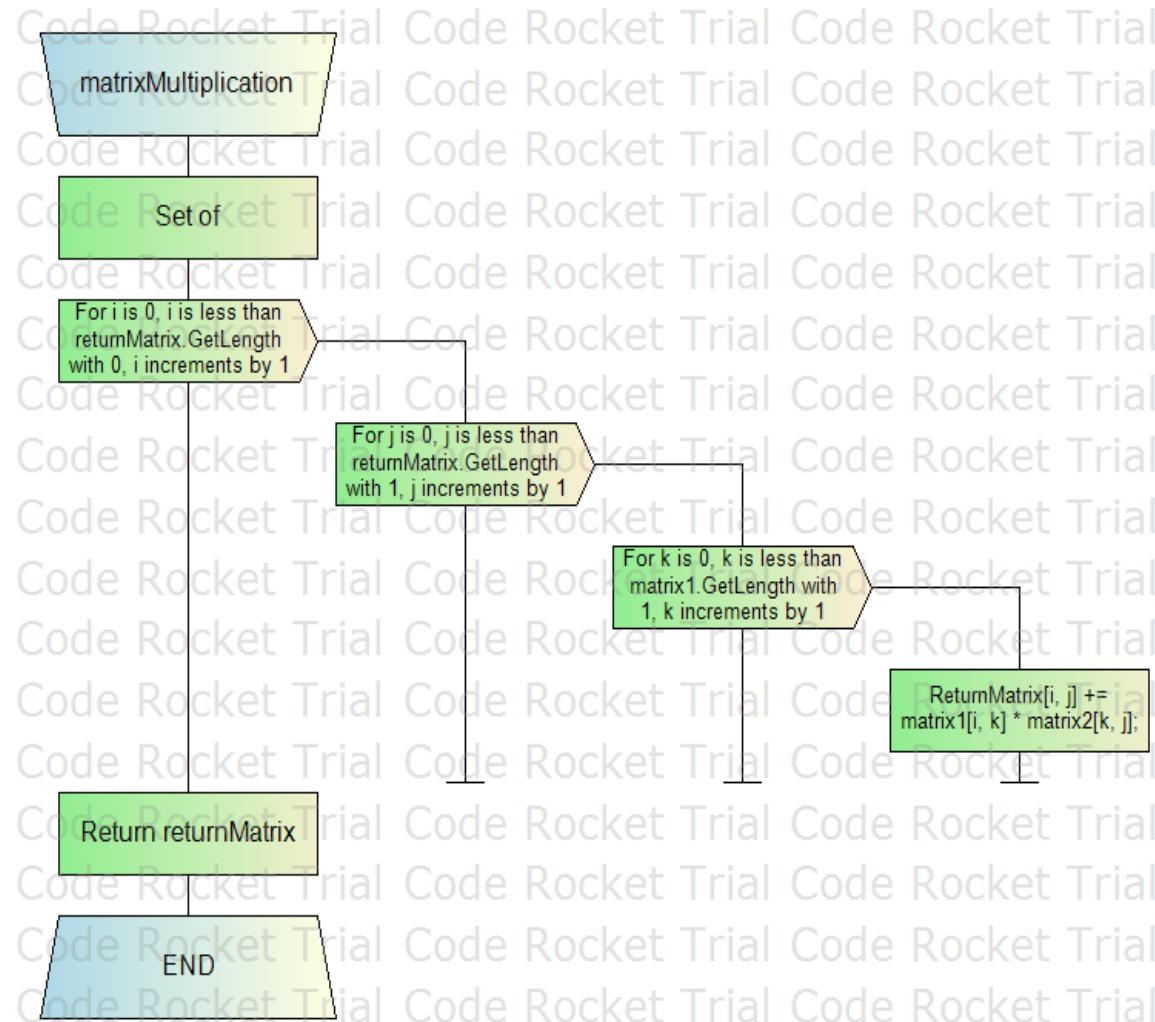
- matrix1 - No type
- matrix2 - No type

**Pseudocode**

Set of

```
For i is 0, i is less than returnMatrix.GetLength with 0, i increments by 1
    For j is 0, j is less than returnMatrix.GetLength with 1, j increments by 1
        For k is 0, k is less than matrix1.GetLength with 1, k increments by 1
            ReturnMatrix[i, j] += matrix1[i, k] * matrix2[k, j];
        EndFor
    EndFor
EndFor
Return returnMatrix
```

**Flowchart**



**Procedure: transposeSolver**

```
public double[,] transposeSolver(double[,] mainMatrix)
```

**Parameters**

- mainMatrix - No type

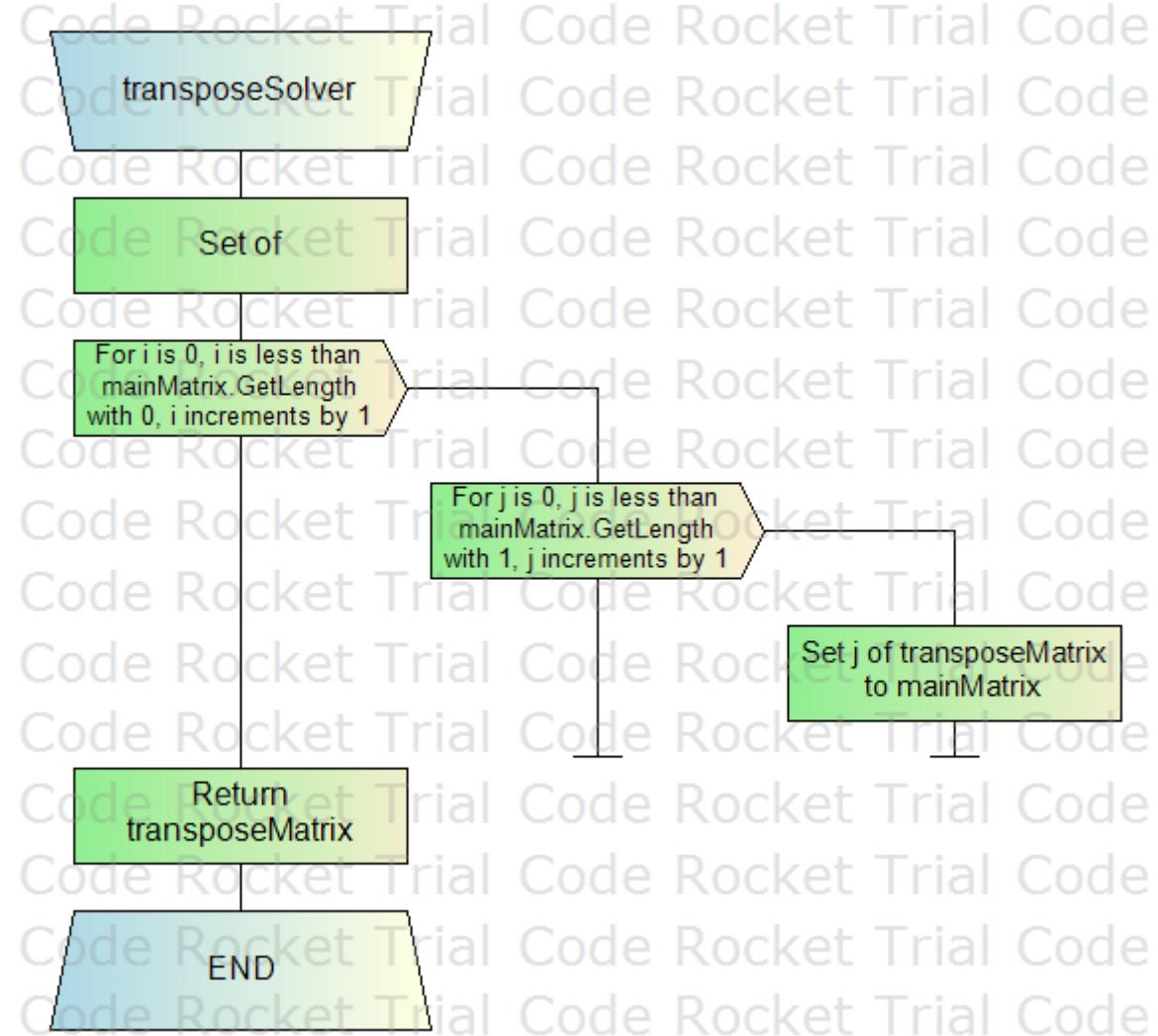
**Pseudocode**

Set of

```
For i is 0, i is less than mainMatrix.GetLength with 0, i increments by 1
    For j is 0, j is less than mainMatrix.GetLength with 1, j increments by 1
        Set j of transposeMatrix to mainMatrix
    EndFor
EndFor
```

Return transposeMatrix

**Flowchart**



**Procedure: adjointSolver**

```
public double[,] adjointSolver(double[,] mainMatrix)
```

**Parameters**

- mainMatrix - No type

**Pseudocode**

Matrix of coefficents

Set of

Set of

For rows is 0, rows is less than lengthOfMatrix, rows increments by 1

    For columns is 0, columns is less than lengthOfMatrix, columns increments by 1

        Set miniMatrix to rowColumnRemover with mainMatrix, rows, columns

        Set rows of returnMatrix to determinantSolver with miniMatrix,

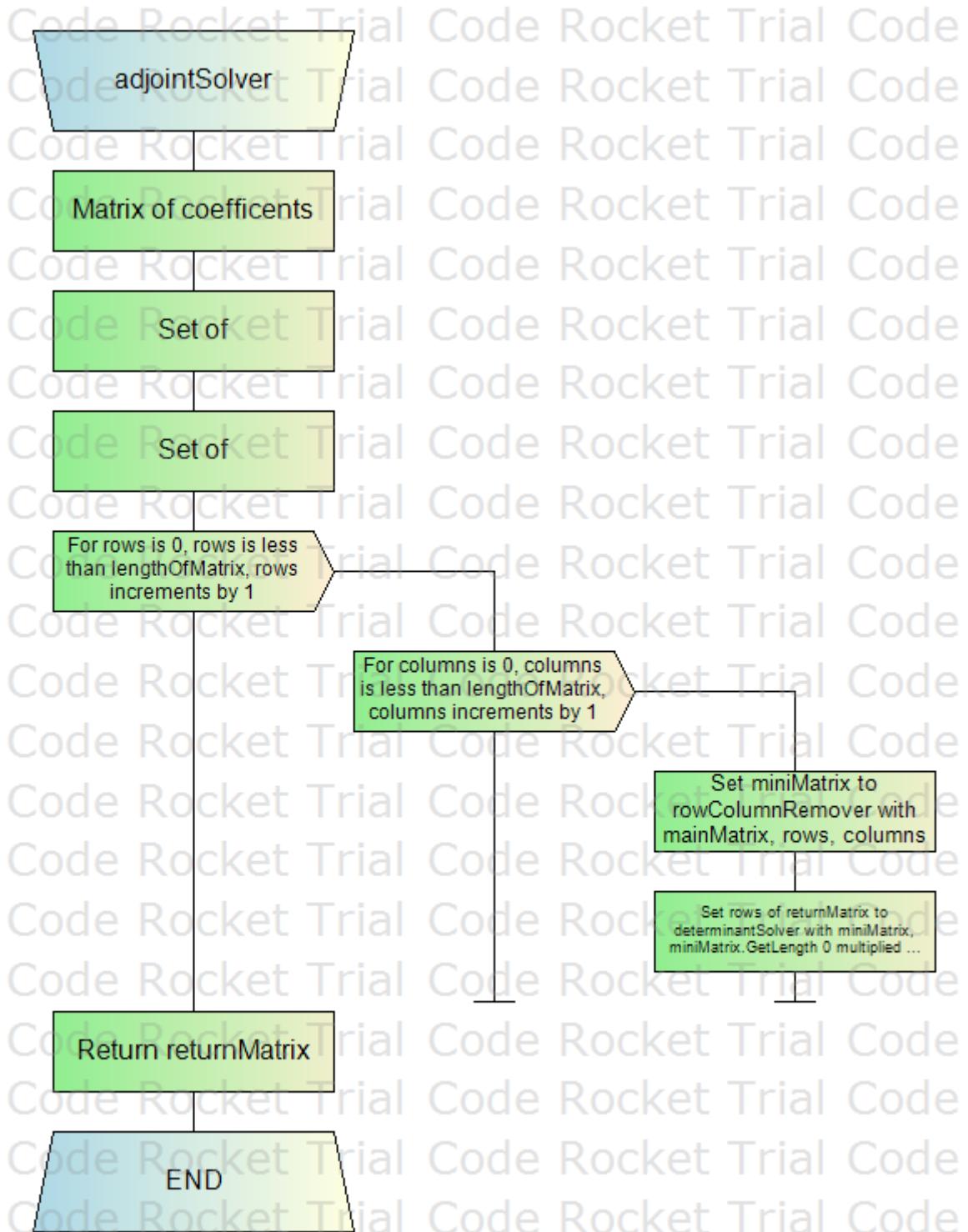
miniMatrix.GetLength 0 multiplied by ( Math.Pow with ( -1 ), ( rows plus 1 plus  
columns plus 1 ) )

    EndFor

EndFor

Return returnMatrix

**Flowchart**



**Procedure: rowColumnRemover**

```
public double[,] rowColumnRemover(double[,] mainMatrix, int row, int column)
```

**Parameters**

- mainMatrix - No type
- row - Integer
- column - Integer

**Pseudocode**

Set of

Initialise k to 0

Initialise l to 0

Initialise m to 0

Initialise n to 0

**For** i is 0, i is less than mainMatrix.GetLength with 1, i increments by 1

**For** j is 0, j is less than mainMatrix.GetLength with 1, j increments by 1

**If** i is not equal to row and j is not equal to column

            Set m of matrixReturn to mainMatrix

            M++;

**If** m is equal to ( mainMatrix.GetLength with 0, -1 )

                Set m to 0

                N++;

**EndIf**

**EndIf**

**If** k

            K++;

**Else**

            Set k to 0

            L++;

**EndIf**

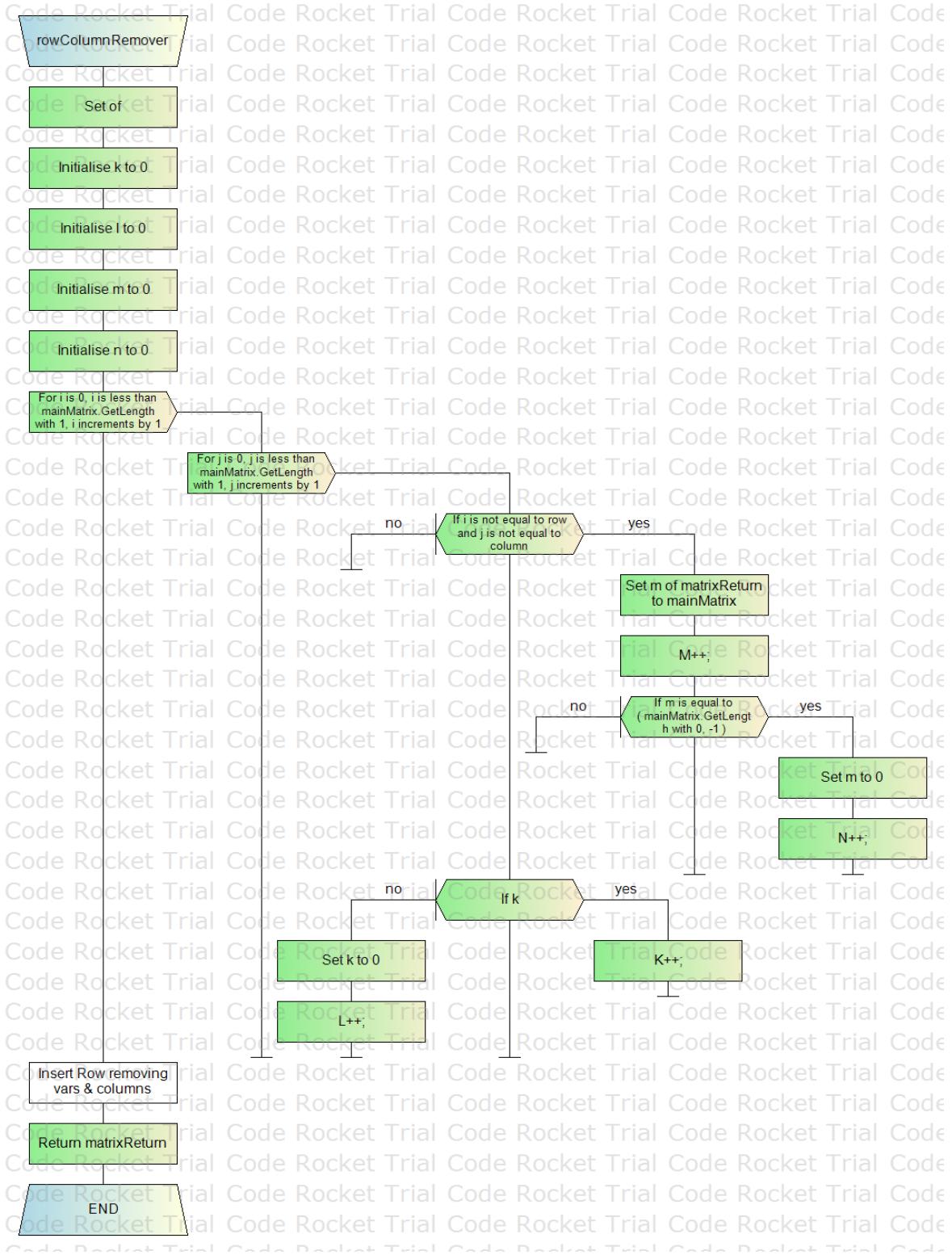
**EndFor**

**EndFor**

Insert Row removing vars & columns

Return matrixReturn

**Flowchart**



**Procedure: determinantSolver**

```
public double determinantSolver(double[,] mainMatrix, int  
lengthOfMatricies)
```

**Parameters**

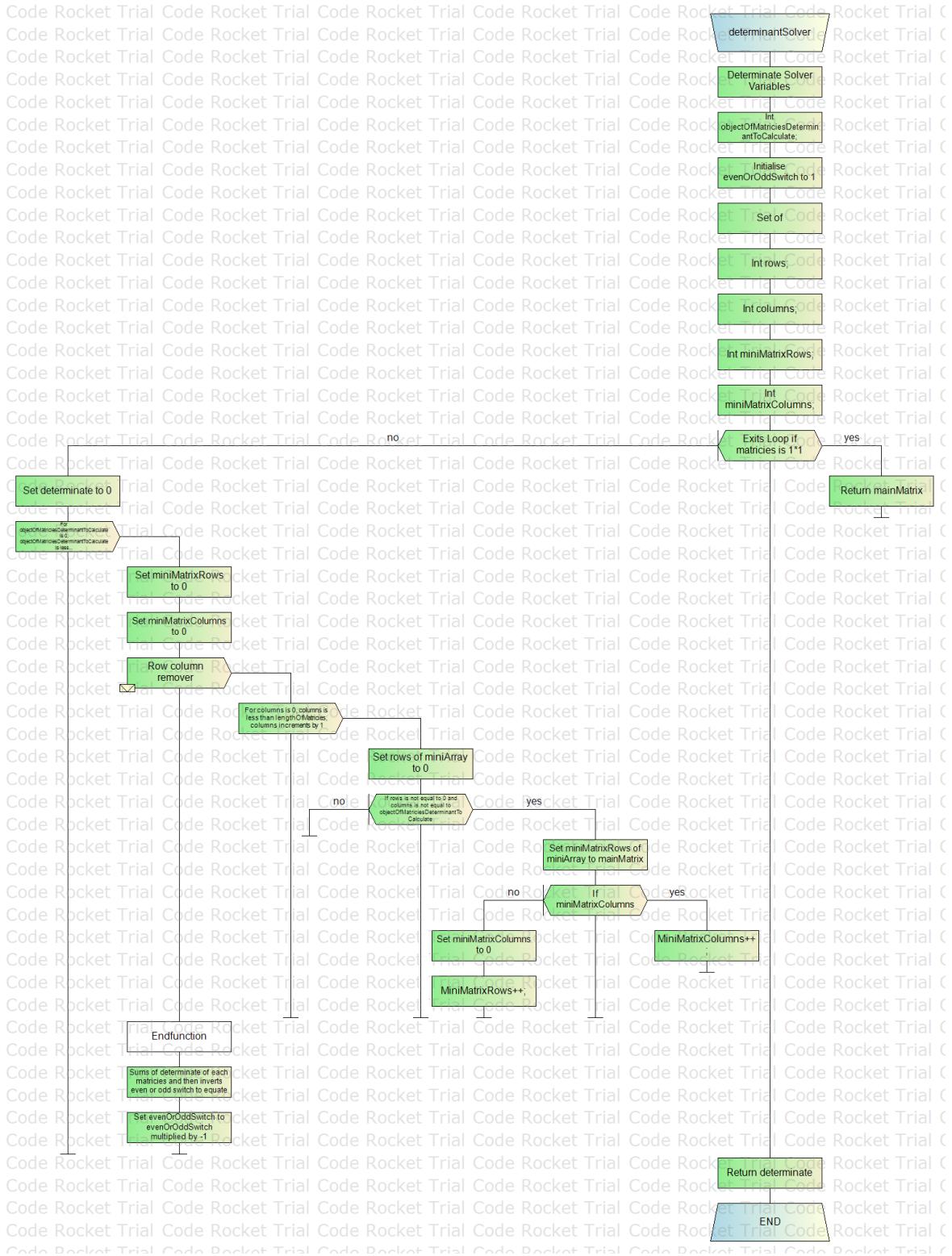
- mainMatrix - No type
- lengthOfMatricies - Integer

Returns Double

**Pseudocode**

```
Determinate Solver Variables  
Int objectOfMatriciesDeterminantToCalculate;  
Initialise evenOrOddSwitch to 1  
Set of  
Int rows;  
Int columns;  
Int miniMatrixRows;  
Int miniMatrixColumns;  
Exits Loop if matricies is 1*1  
    Return mainMatrix  
Else  
    Set determinate to 0  
    For objectOfMatriciesDeterminantToCalculate is 0,  
objectOfMatriciesDeterminantToCalculate is less than lengthOfMatricies,  
objectOfMatriciesDeterminantToCalculate increments by 1  
        Set miniMatrixRows to 0  
        Set miniMatrixColumns to 0  
        Row column remover  
            For columns is 0, columns is less than lengthOfMatricies, columns  
increments by 1  
                Set rows of miniArray to 0  
                If rows is not equal to 0 and columns is not equal to  
objectOfMatriciesDeterminantToCalculate  
                    Set miniMatrixRows of miniArray to mainMatrix  
                    If miniMatrixColumns  
                        MiniMatrixColumns++;  
                    Else  
                        Set miniMatrixColumns to 0  
                        MiniMatrixRows++;  
                    EndIf  
                EndIf  
            EndFor  
        EndFor  
    Endfunction  
    Sums of determinate of each matricies and then inverts even or odd  
switch to equate  
    Set evenOrOddSwitch to evenOrOddSwitch multiplied by -1  
 EndFor  
EndIf  
Return determinate
```

Flowchart



**Code File: members.cs**

```
using Sectrics_V2.Properties;
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: members**

```
public partial class members : Form
```

**Attribute: zoom**

```
float zoom = 1f;
```

**Attribute: xMouseOffset**

```
double xMouseOffset;
```

**Attribute: yMouseOffset**

```
double yMouseOffset;
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

**Procedure: Constructor**

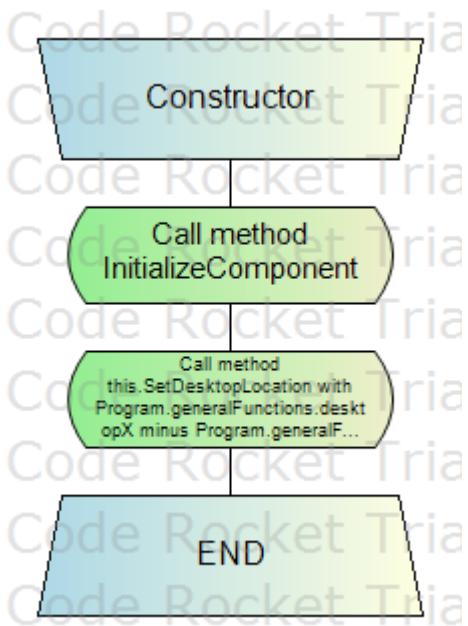
```
public members()
```

Returns members

**Pseudocode**

```
Call method InitializeComponent
Call method this.SetDesktopLocation with Program.generalFunctions.desktopX minus
Program.generalFunctions.movX, Program.generalFunctions.desktopY minus
Program.generalFunctions.movY
```

**Flowchart**



**Procedure: WndProc**

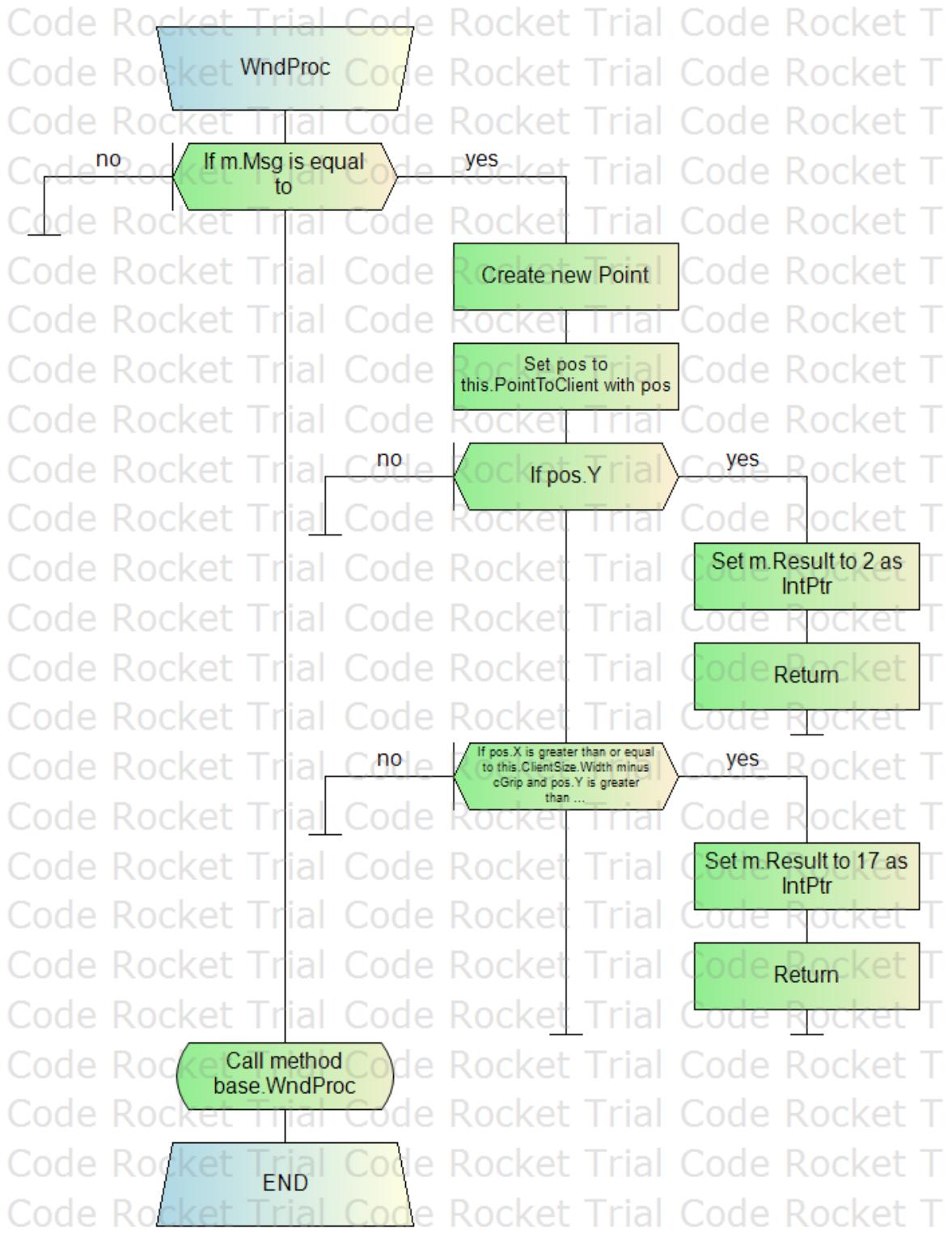
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

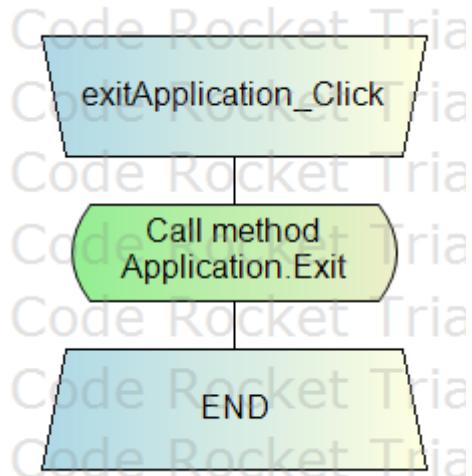
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



**Procedure: minimize\_Click**

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

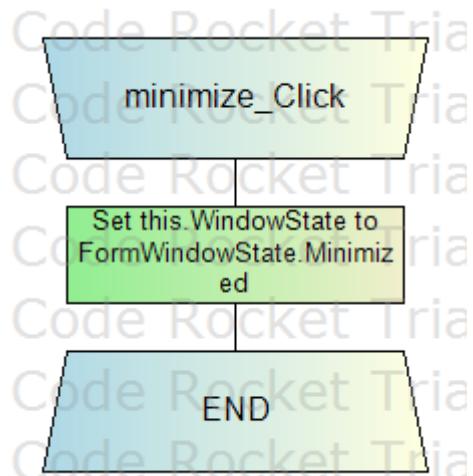
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

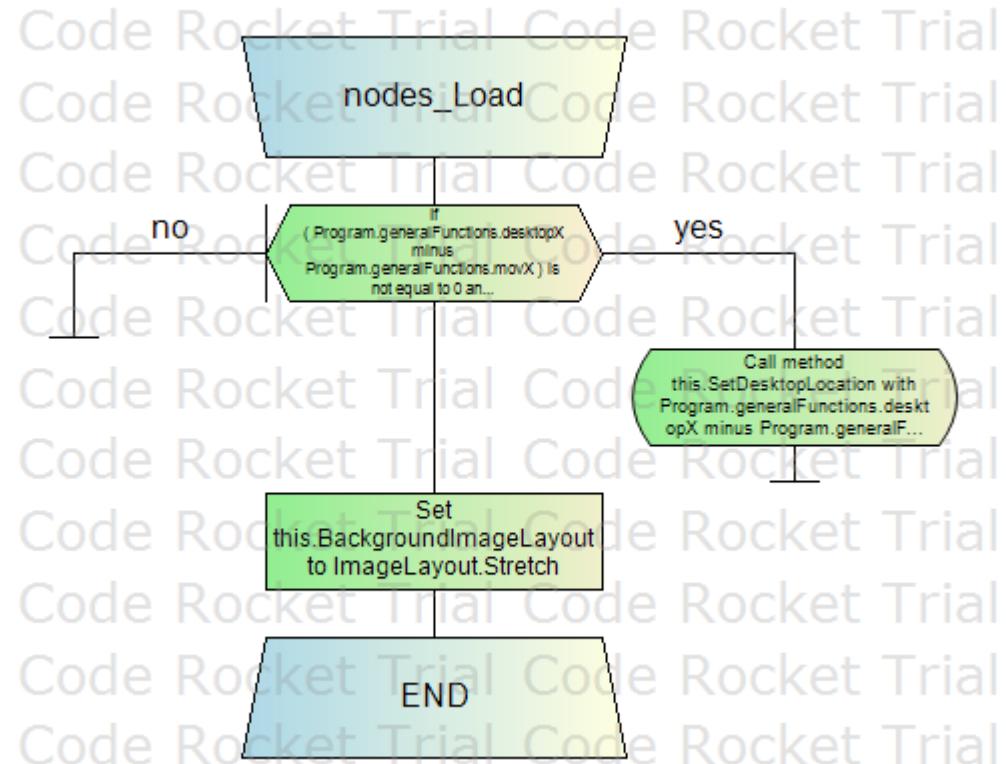
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: addMemberButton\_Click**

```
private void addMemberButton_Click(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

**Try**

```
    Initialise repeatedMember to false
```

```
    For i is 0, i is less than Program.bridgeData.memberConnection.Count, i  
    increments by 1  
        If Program.bridgeData.memberConnection is equal to Convert.ToInt16 with  
        toMemberTextbox.Text and Program.bridgeData.memberConnection is equal to  
        Convert.ToInt16 with fromMemberTextbox.Text  
            Set repeatedMember to true
```

```
        EndIf
```

```
    EndFor
```

```
    If toMemberTextbox.Text is not equal to null and fromMemberTextbox.Text is not  
    equal to null and Convert.ToInt16 with toMemberTextbox.Text is less than  
    Program.bridgeData.nodes.Count and Convert.ToInt16 with fromMemberTextbox.Text is  
    less than Program.bridgeData.nodes.Count and repeatedMember is equal to false
```

```
        Call method Program.bridgeData.memberConnection.Add
```

```
        Set Program.bridgeData.memberIndex of
```

```
        Program.bridgeData.memberConnection to Convert.ToInt16 with fromMemberTextbox.Text  
        Set Program.bridgeData.memberIndex of
```

```
        Program.bridgeData.memberConnection to Convert.ToInt16 with toMemberTextbox.Text  
        Program.bridgeData.memberIndex++;
```

```
        Set fromMemberTextbox.Text to ""
```

```
        Set toMemberTextbox.Text to ""
```

```
    Else if toMemberTextbox.Text is equal to null
```

```
        Set toMemberTextbox.Text to "Incorrect Variable Entered"
```

```
    Else if fromMemberTextbox.Text is equal to null
```

```
        Set fromMemberTextbox.Text to "Incorrect Variable Entered"
```

```
    Else
```

```
        Set toMemberTextbox.Text to "Incorrect Variable Entered"
```

```
        Set fromMemberTextbox.Text to "Incorrect Variable Entered"
```

```
    EndIf
```

```
    Call method bridgeDrawing.Refresh
```

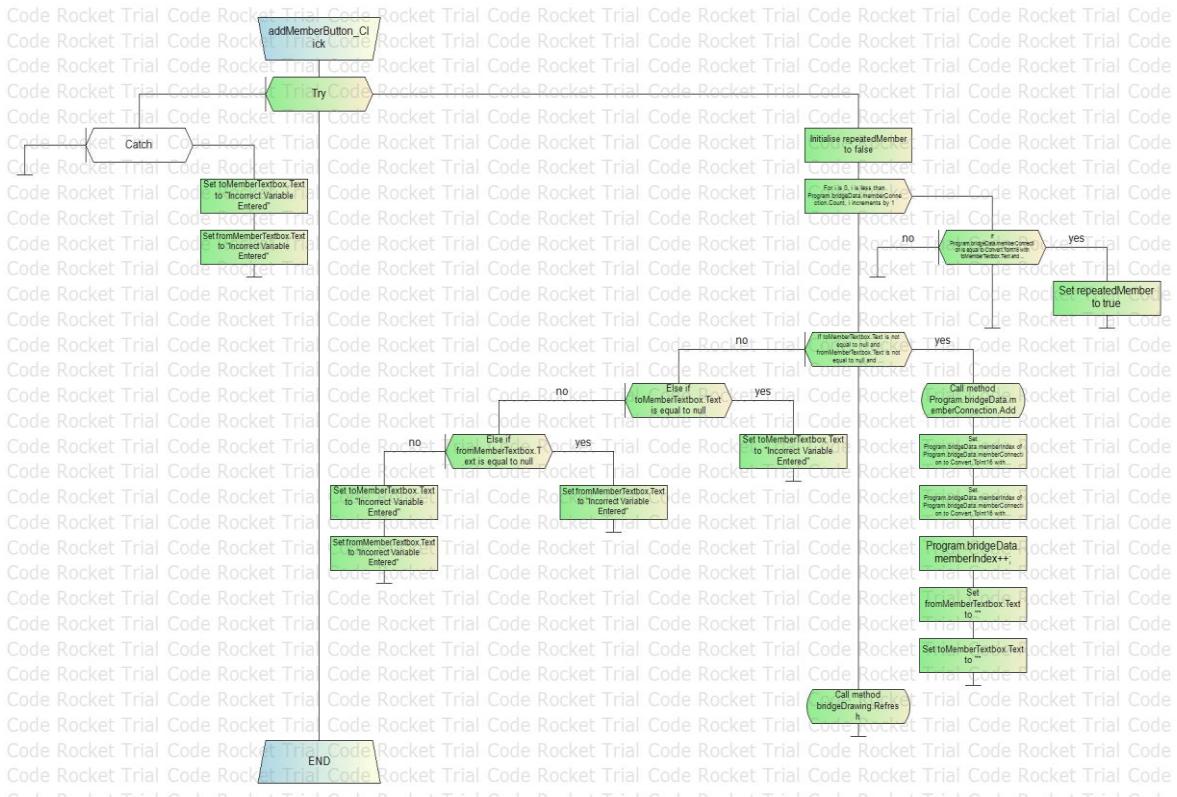
**Catch**

```
    Set toMemberTextbox.Text to "Incorrect Variable Entered"
```

```
    Set fromMemberTextbox.Text to "Incorrect Variable Entered"
```

**EndTry**

Flowchart



**Procedure: fromMemberTextbox\_TextChanged**

```
private void fromMemberTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

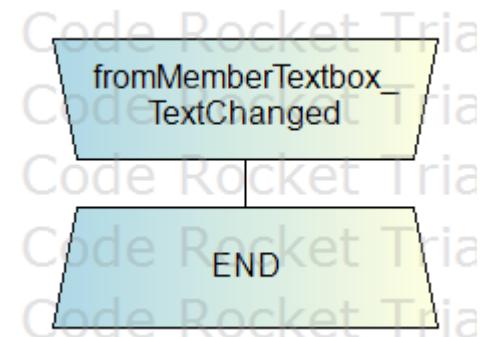
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Procedure: toMemberTextbox\_TextChanged*

```
private void toMemberTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

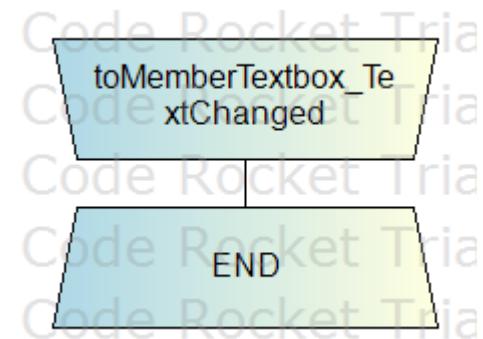
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Procedure: membersListView\_SelectedIndexChanged*

```
private void membersListView_SelectedIndexChanged(object sender,  
EventArgs e)
```

**Parameters**

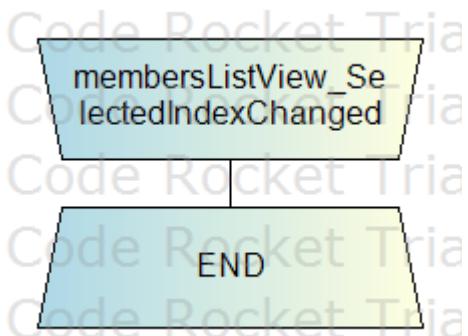
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Procedure: nodeListView\_SelectedIndexChanged*

```
private void nodeListView_SelectedIndexChanged(object sender,  
EventArgs e)
```

**Parameters**

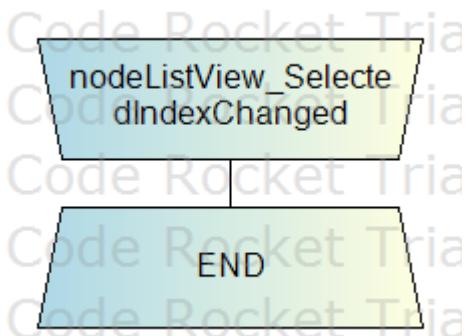
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

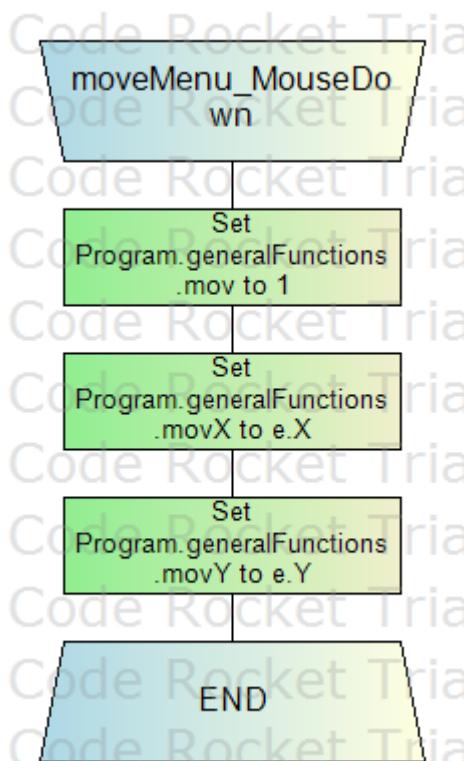
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

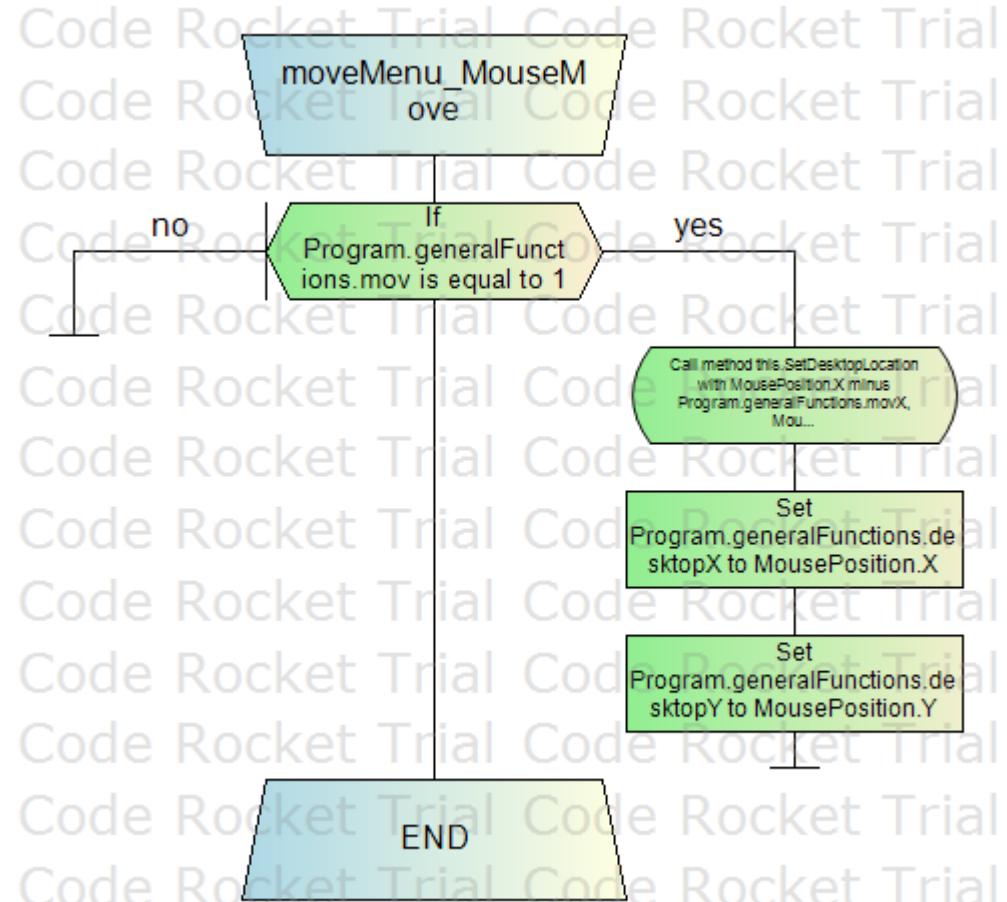
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseUp**

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

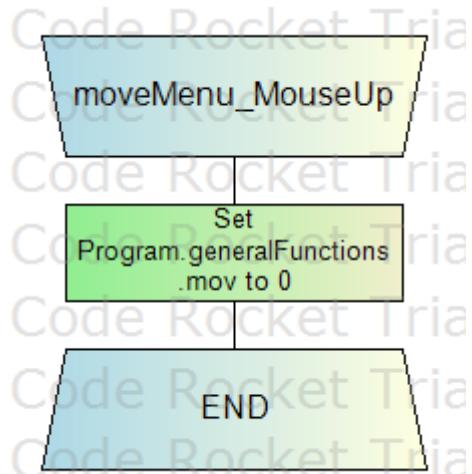
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

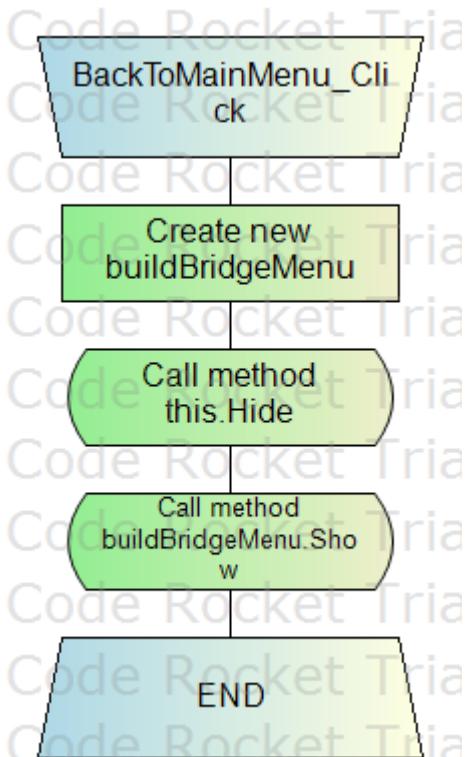
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

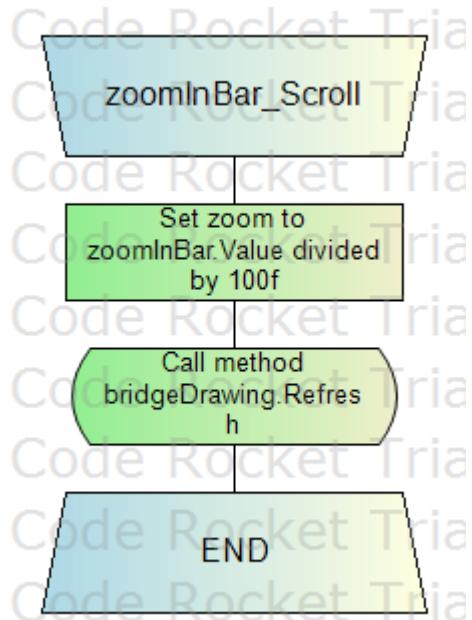
Returns Void

**Pseudocode**

```
Set zoom to zoomInBar.Value divided by 100f
```

```
Call method bridgeDrawing.Refresh
```

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

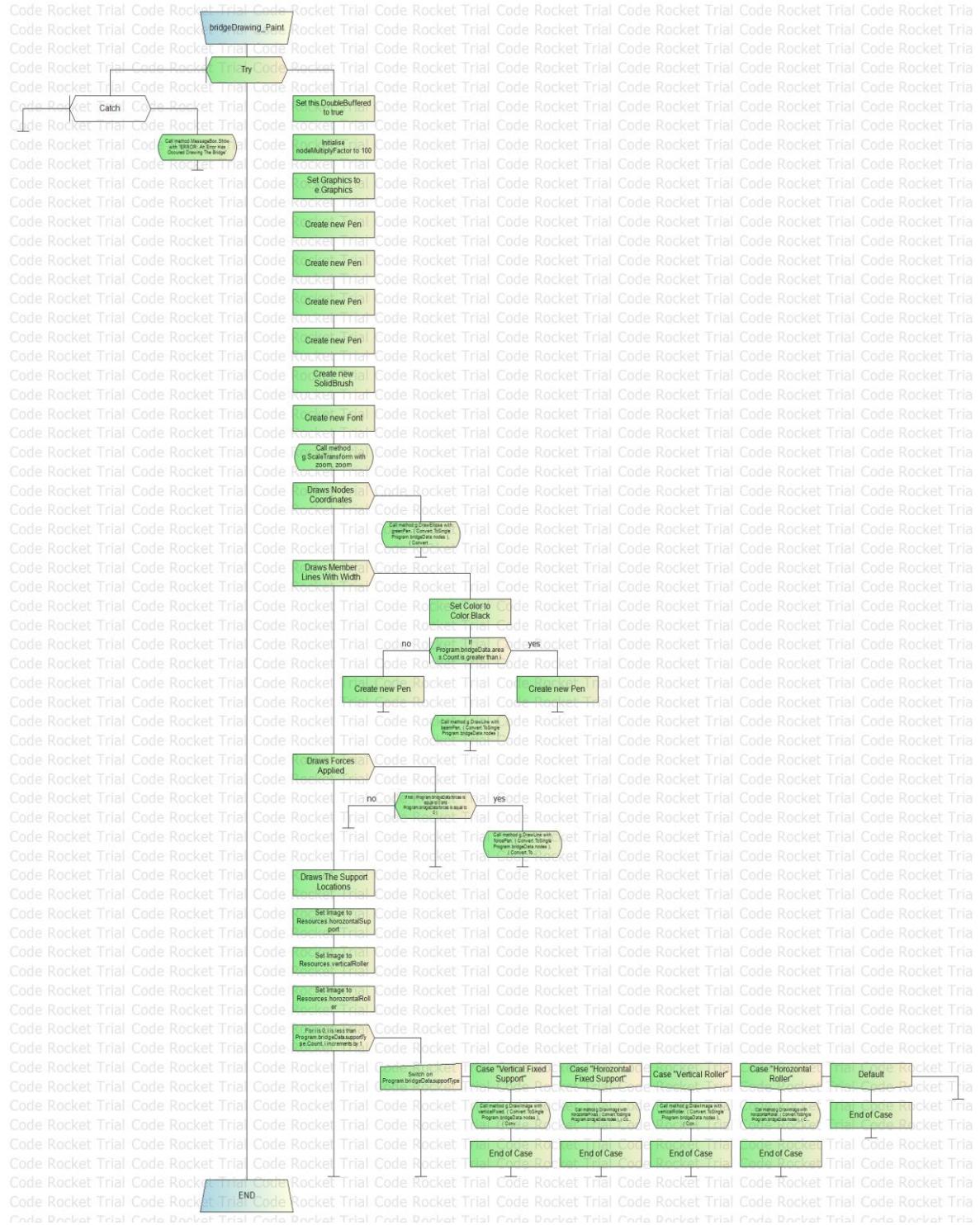
**Pseudocode**

**Try**

```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```
        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry
```

## Flowchart



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

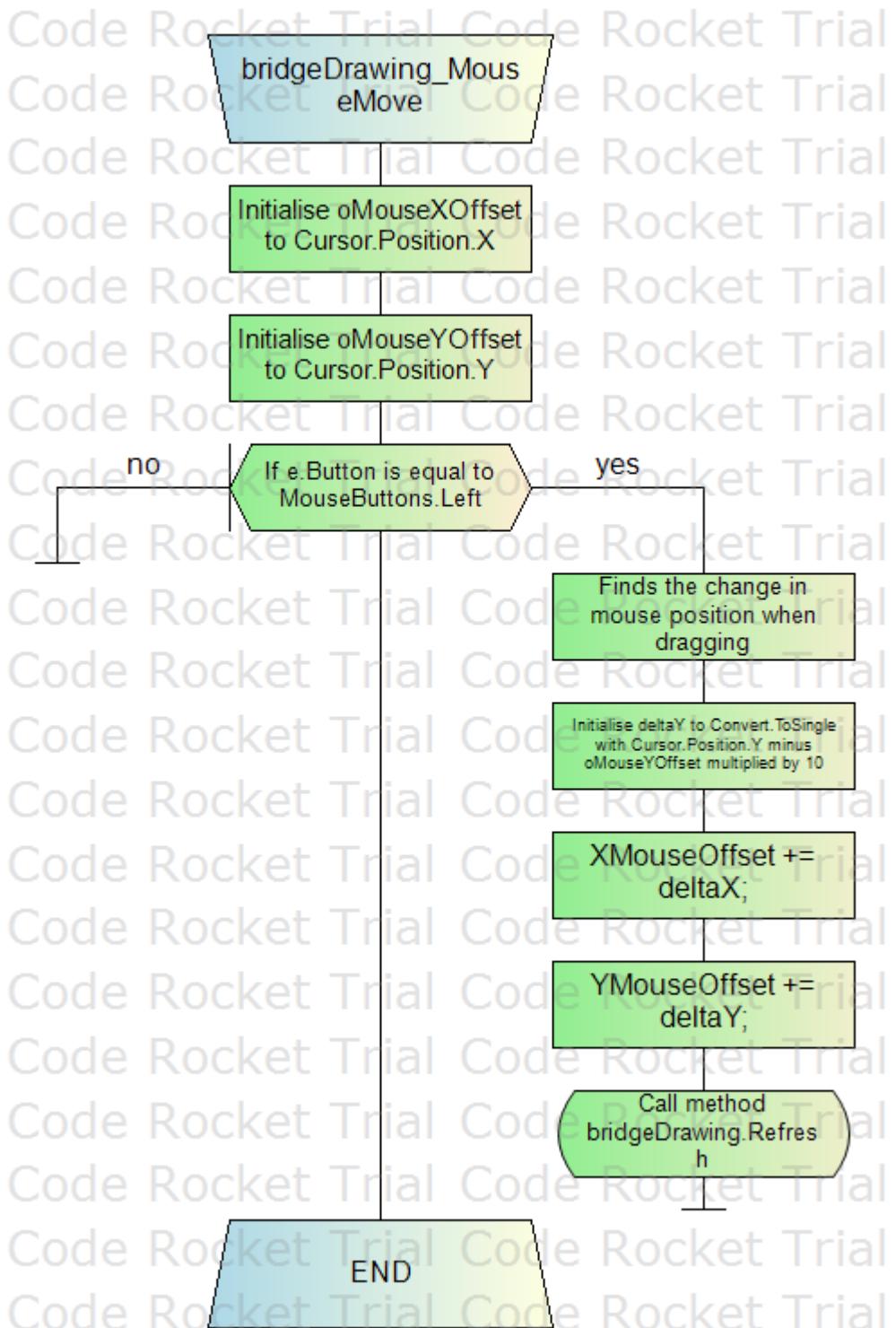
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



*Procedure: membersTable\_Click*

```
private void membersTable_Click(object sender, EventArgs e)
```

**Parameters**

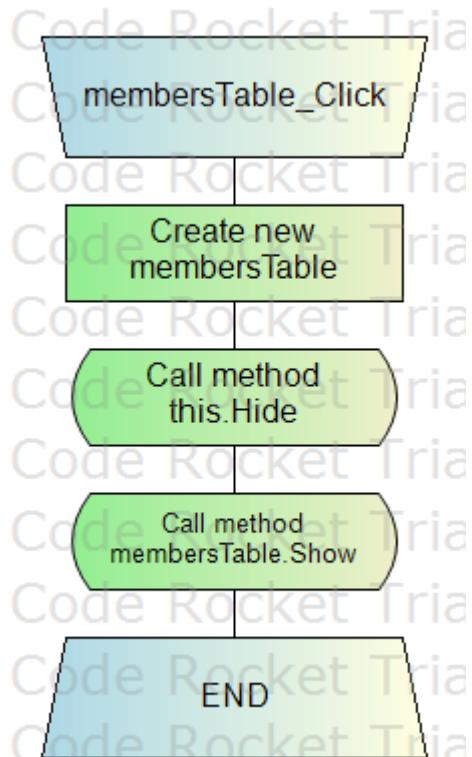
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new membersTable  
Call method this.Hide  
Call method membersTable.Show
```

**Flowchart**



*Code File: members.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

**Class: members**

partial class members

**Attribute: components**

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure: Dispose**

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

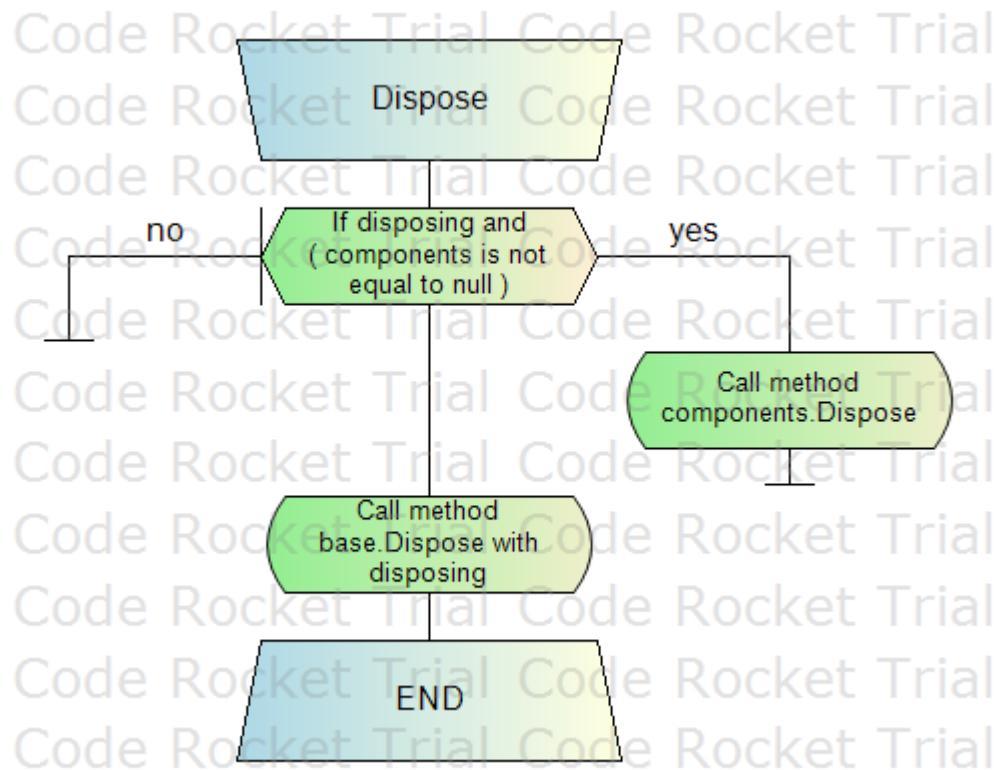
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

**Flowchart**



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.Button
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackColor to System.Drawing.Color.Transparent
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.TabStop to false
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackColor to System.Drawing.Color.Transparent
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.TabStop to false
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
addMemberButton
Set this.addMemberButton.BackColor to System.Drawing.Color.Transparent
Set this.addMemberButton.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.addMemberButton.FlatAppearance.BorderSize to 2
Set this.addMemberButton.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ) ), ( ( int ) ) )
Set this.addMemberButton.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font

```

```

Set this.addMemberButton.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.addMemberButton.Name to "addMemberButton"
Create new System.Drawing.Size
Set this.addMemberButton.TabIndex to 21
Set this.addMemberButton.TabStop to false
Set this.addMemberButton.Text to "ADD MEMBER"
Set this.addMemberButton.UseVisualStyleBackColor to false
This.addMemberButton.Click += new System.EventHandler(this.addMemberButton_Click);
fromMemberTextbox
Create new System.Drawing.Font
Create new System.Drawing.Point
Set this.fromMemberTextbox.Name to "fromMemberTextbox"
Create new System.Drawing.Size
Set this.fromMemberTextbox.TabIndex to 0
Set this.fromMemberTextbox.Text to "Enter the from member here"
This.fromMemberTextbox.TextChanged += new
System.EventHandler(this.fromMemberTextbox_TextChanged);
label2
Set this.label2.AutoSize to true
Set this.label2.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label2.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label2.Name to "label2"
Create new System.Drawing.Size
Set this.label2.TabIndex to 19
Set this.label2.Text to "FROM MEMBER:"
toMemberTextbox
Create new System.Drawing.Font
Create new System.Drawing.Point
Set this.toMemberTextbox.Name to "toMemberTextbox"
Create new System.Drawing.Size
Set this.toMemberTextbox.TabIndex to 1
Set this.toMemberTextbox.Text to "Enter the to member here"
This.toMemberTextbox.TextChanged += new
System.EventHandler(this.toMemberTextbox_TextChanged);
label1
Set this.label1.AutoSize to true
Set this.label1.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 17
Set this.label1.Text to "TO MEMBER:"
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 26
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent
Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to

```

```

System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 27
Set this.BackToMainMenu.TabStop to false
Set this.BackToMainMenu.Text to "BACK"
Set this.BackToMainMenu.UseVisualStyleBackColor to false
This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 29
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 28
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
membersTable
Set this.membersTable.BackColor to System.Drawing.Color.Transparent
Set this.membersTable.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.membersTable.FlatAppearance.BorderSize to 2
Set this.membersTable.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.membersTable.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.membersTable.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.membersTable.Name to "membersTable"
Create new System.Drawing.Size
Set this.membersTable.TabIndex to 30
Set this.membersTable.TabStop to false
Set this.membersTable.Text to "TABLE"
Set this.membersTable.UseVisualStyleBackColor to false
This.membersTable.Click += new System.EventHandler(this.membersTable_Click);
members
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.membersTable
Call method this.Controls.Add with this.bridgeDrawing
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.addMemberButton
Call method this.Controls.Add with this.fromMemberTextbox
Call method this.Controls.Add with this.label2
Call method this.Controls.Add with this.toMemberTextbox

```

```
Call method this.Controls.Add with this.label1
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Call method this.Controls.Add with this.BackToMainMenu
Set this.DoubleBuffered to true
Set this.ForeColor to System.Drawing.SystemColors.ControlText
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "members"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout
```

### Flowchart



```

#endifregion
Attribute: minimize

private System.Windows.Forms.Button minimize;
Attribute: exitApplication

private System.Windows.Forms.Button exitApplication;
Attribute: addMemberButton

private System.Windows.Forms.Button addMemberButton;
Attribute: fromMemberTextbox

private System.Windows.Forms.TextBox fromMemberTextbox;
Attribute: label2

private System.Windows.Forms.Label label2;
Attribute: toMemberTextbox

private System.Windows.Forms.TextBox toMemberTextbox;
Attribute: label1

private System.Windows.Forms.Label label1;
Attribute: moveMenu

private System.Windows.Forms.Panel moveMenu;
Attribute: BackToMainMenu

private System.Windows.Forms.Button BackToMainMenu;
Attribute: bridgeDrawing

private System.Windows.Forms.Panel bridgeDrawing;
Attribute: zoomInBar

private System.Windows.Forms.TrackBar zoomInBar;
Attribute: membersTable

private System.Windows.Forms.Button membersTable;
Code File: membersTable.cs

using Sectrics_V2.Properties;
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: membersTable**

```
public partial class membersTable : Form
```

**Attribute: zoom**

```
float zoom = 1f;
```

**Attribute: xMouseOffset**

```
double xMouseOffset;
```

**Attribute: yMouseOffset**

```
double yMouseOffset;
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

**Procedure: Constructor**

```
public membersTable()
```

Returns membersTable

**Pseudocode**

Call method InitializeComponent

Call method this.SetDesktopLocation with Program.generalFunctions.desktopX minus  
Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY

Adds The Array Coordinates Into ListBox

**For** i is 0, i is less than Program.bridgeData.nodes.Count, i increments by 1  
    Call method nodeListView.Items.Add with "Node " plus i plus " " plus "X

Coordinate: " plus Program.bridgeData.nodes plus " | Y Coordinate: " plus  
Program.bridgeData.nodes

**EndFor**

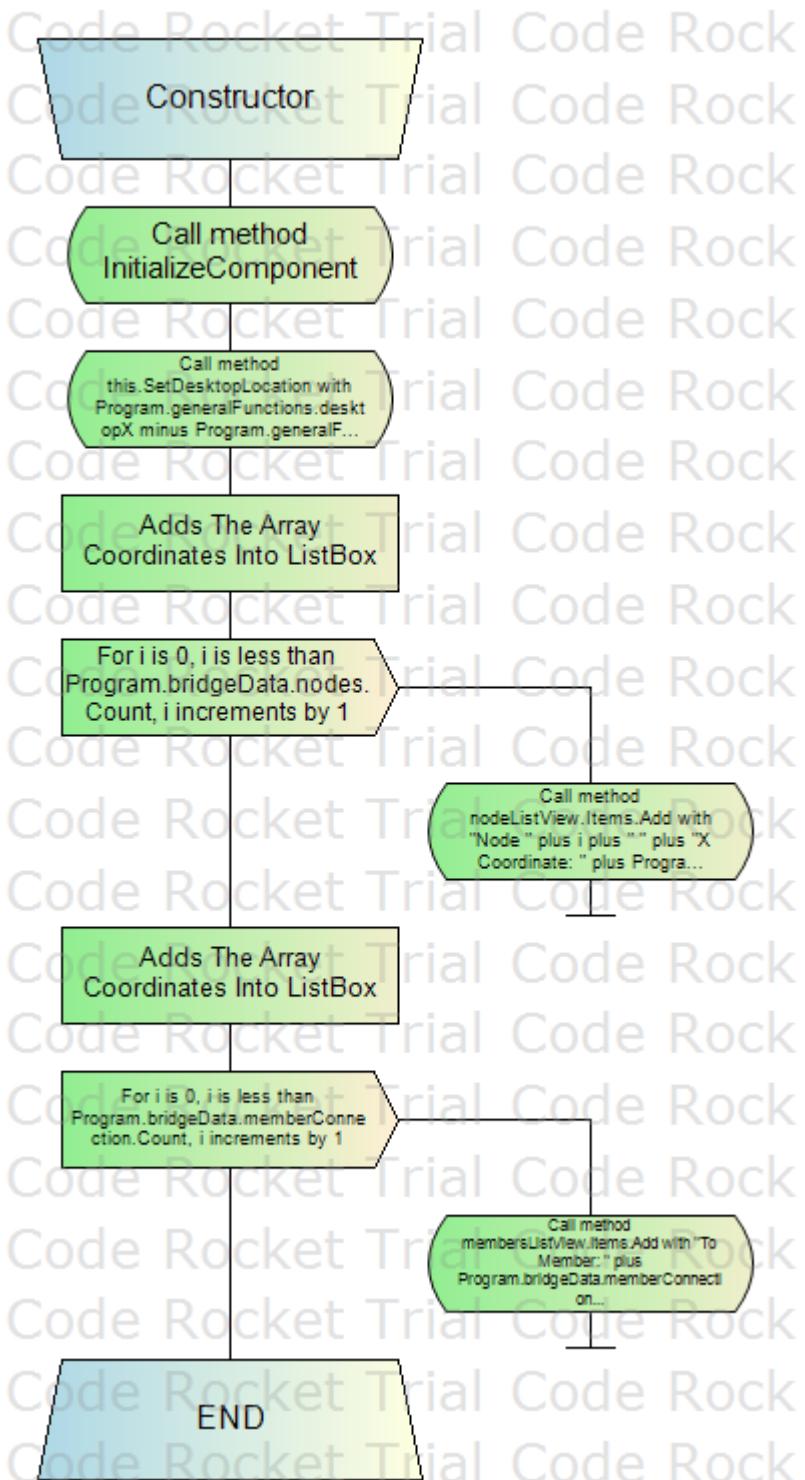
Adds The Array Coordinates Into ListBox

**For** i is 0, i is less than Program.bridgeData.memberConnection.Count, i increments  
by 1

    Call method membersListView.Items.Add with "To Member: " plus  
Program.bridgeData.memberConnection plus " | From Member: " plus  
Program.bridgeData.memberConnection

**EndFor**

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

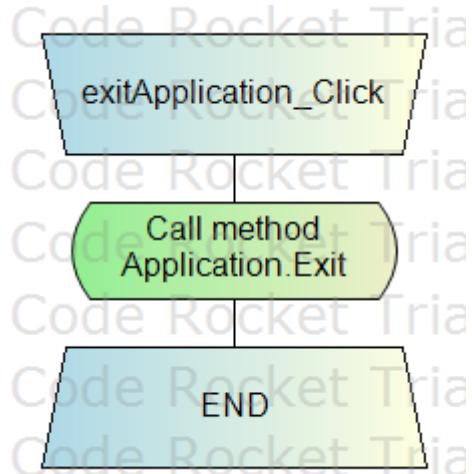
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

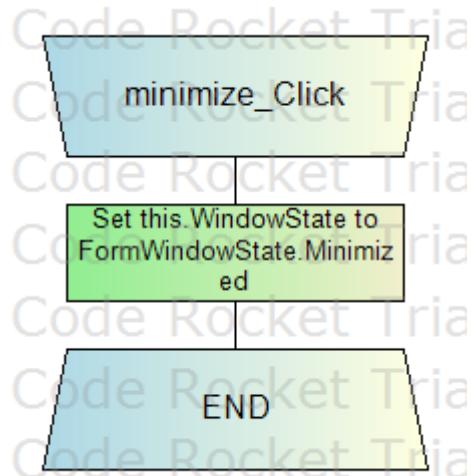
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

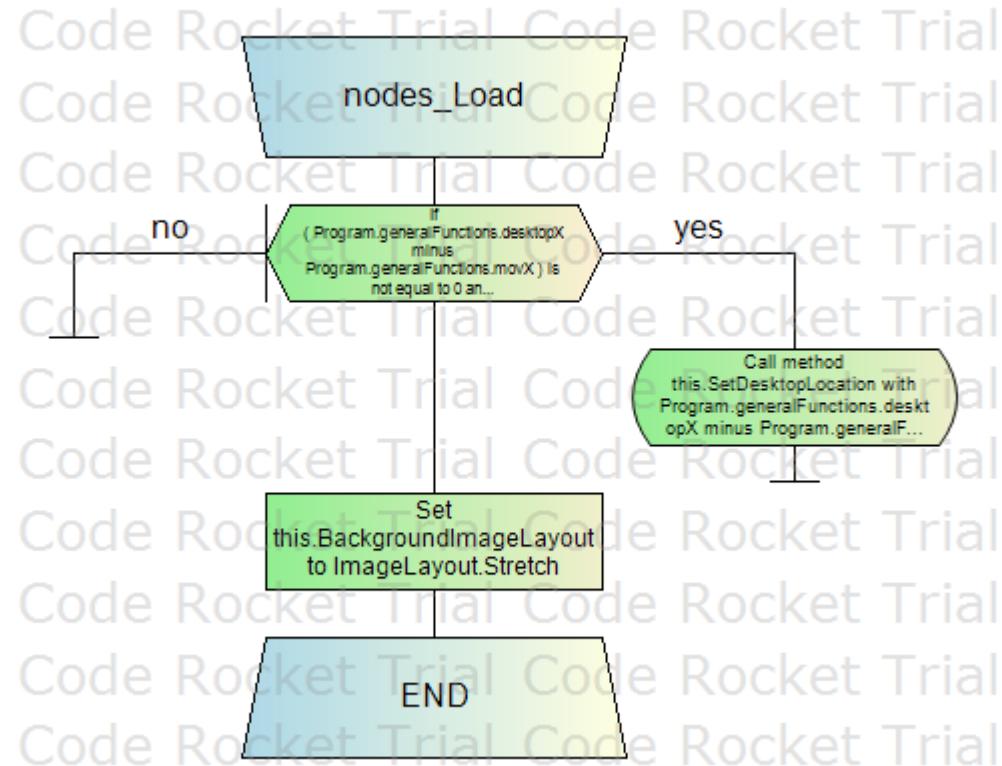
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

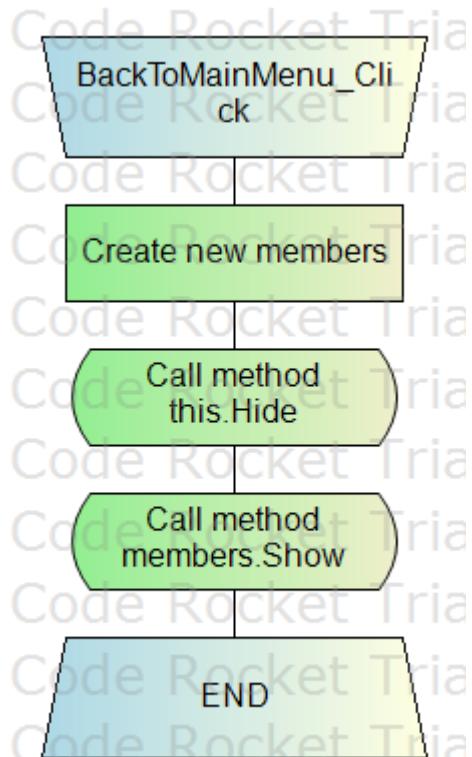
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new members  
Call method this.Hide  
Call method members.Show
```

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

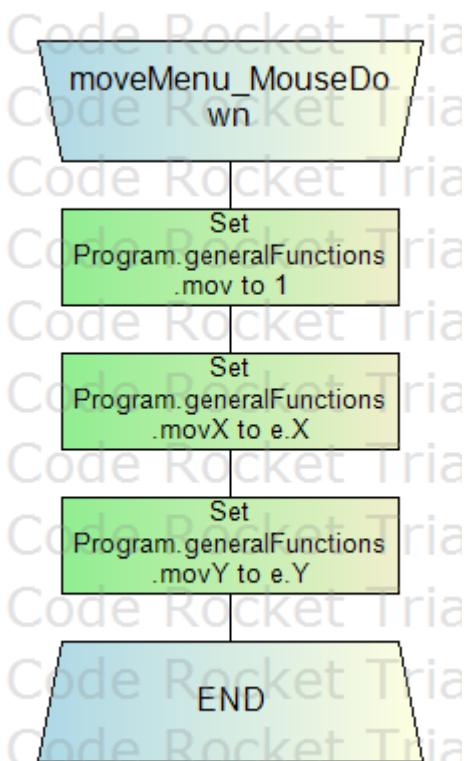
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

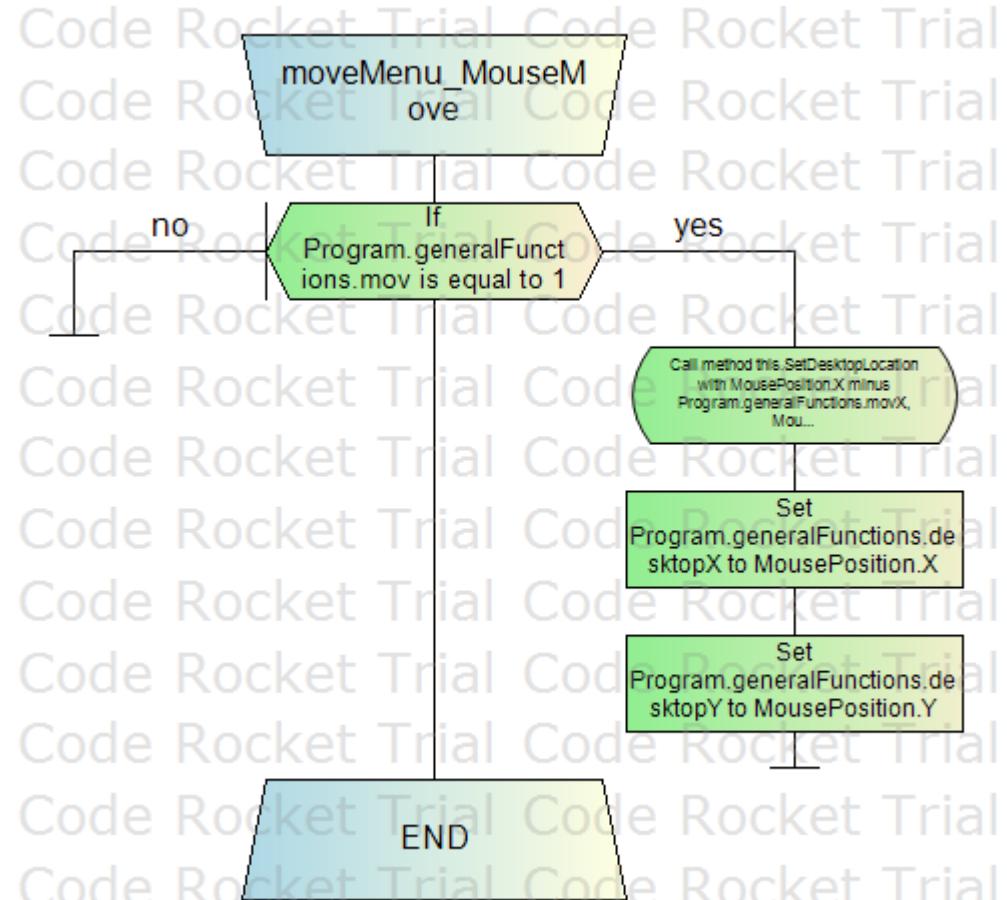
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseUp**

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

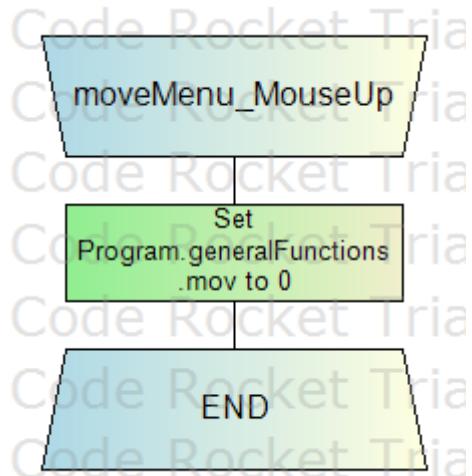
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

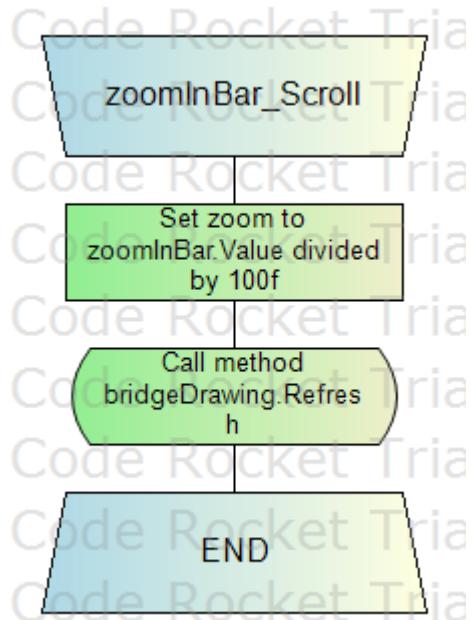
Returns Void

**Pseudocode**

```
Set zoom to zoomInBar.Value divided by 100f
```

```
Call method bridgeDrawing.Refresh
```

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

**Try**

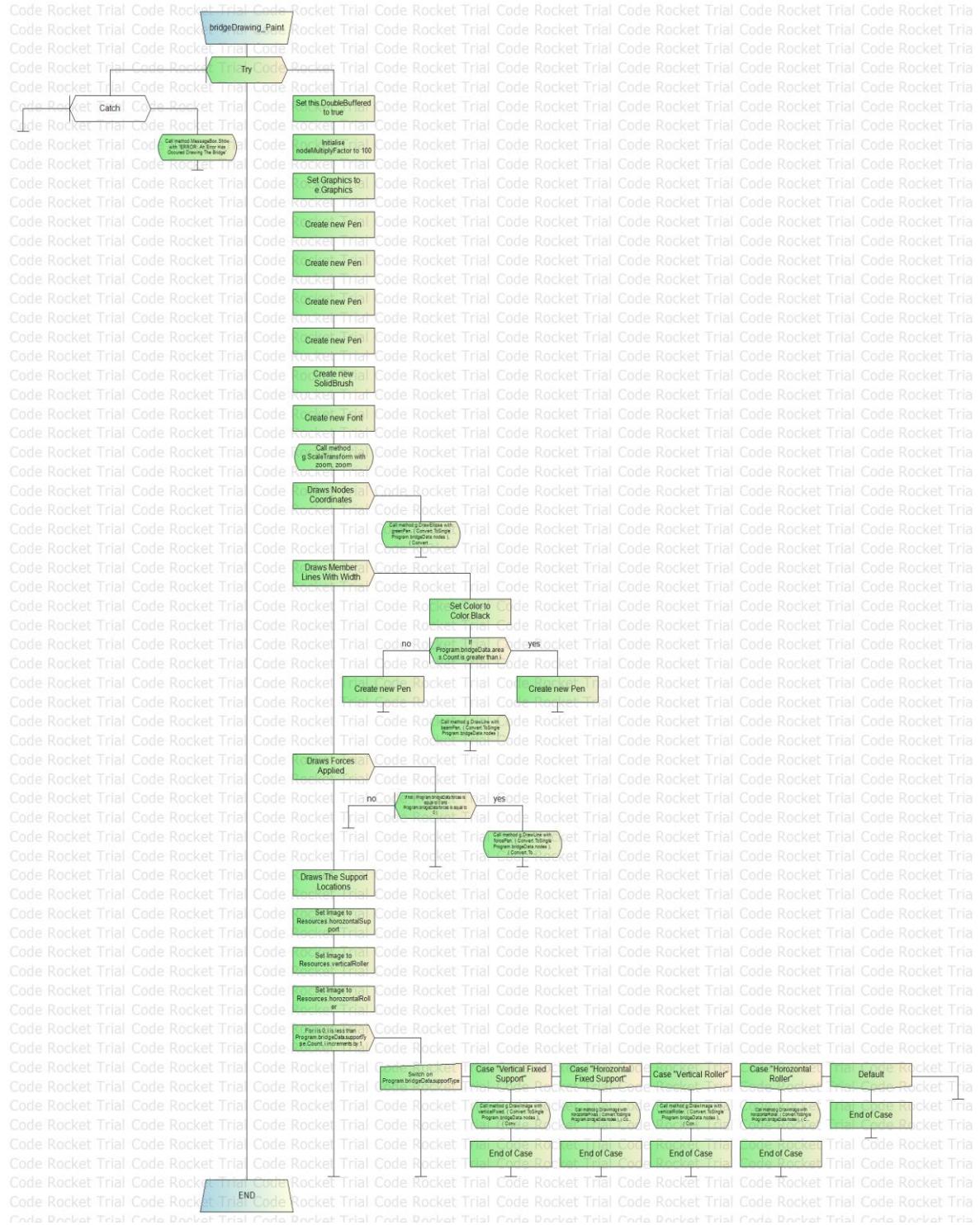
```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```

        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry

```

### Flowchart



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

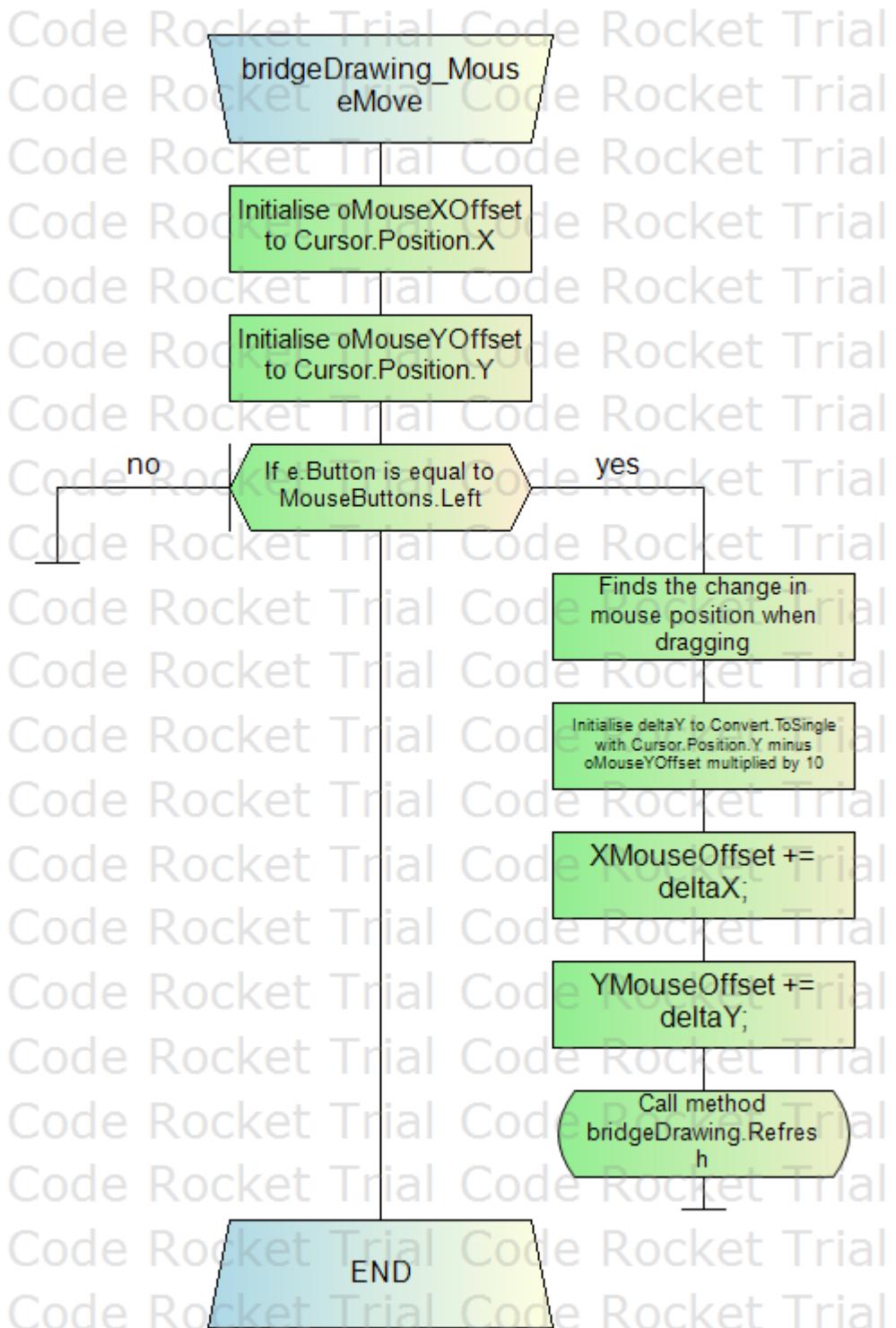
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



**Procedure: removeMember\_Click**

```
private void removeMember_Click(object sender, EventArgs e)
```

**Parameters**

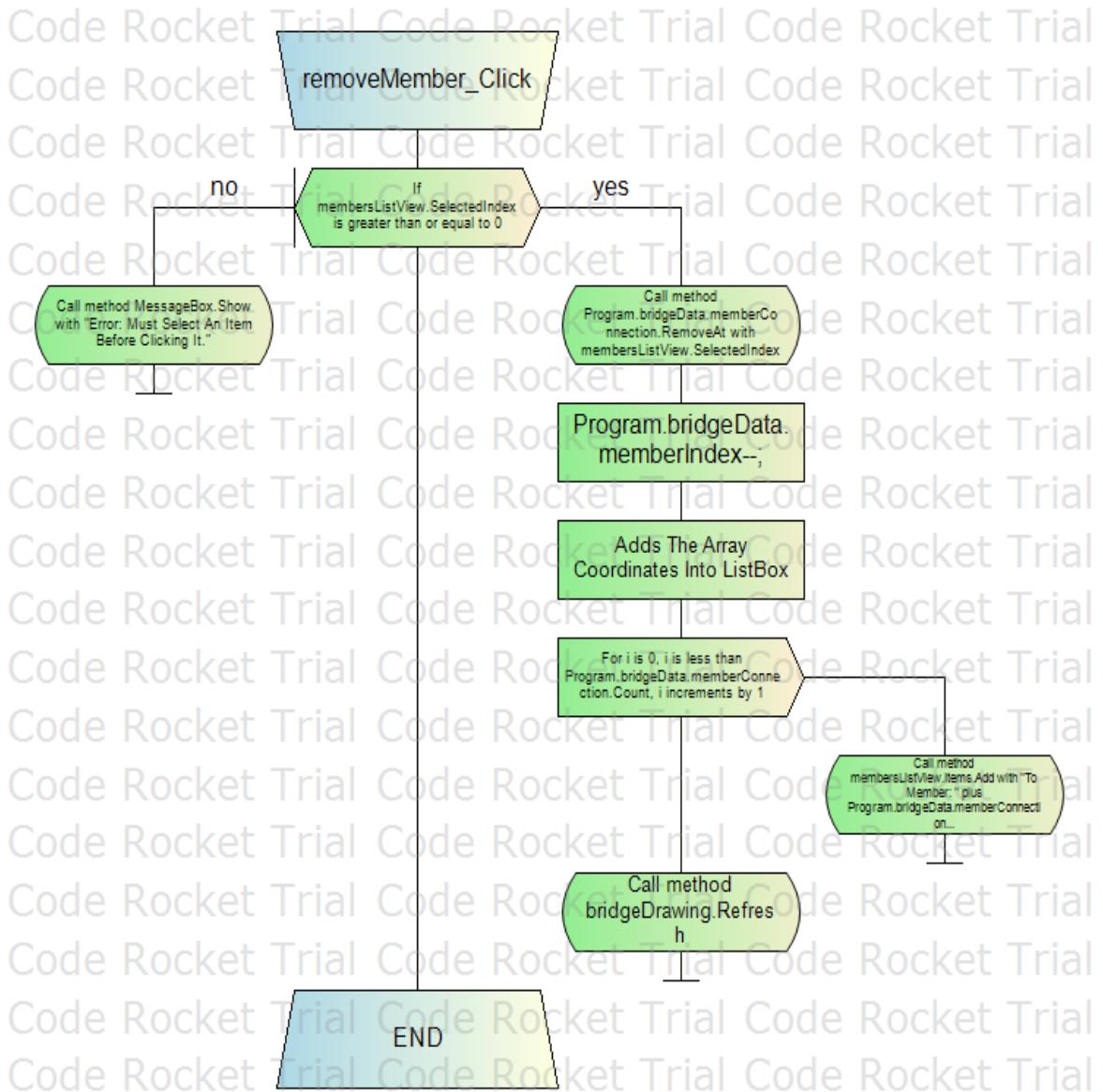
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If membersListView.SelectedIndex is greater than or equal to 0
    Call method Program.bridgeData.memberConnection.RemoveAt with
membersListView.SelectedIndex
    Program.bridgeData.memberIndex--;
    Adds The Array Coordinates Into ListBox
    For i is 0, i is less than Program.bridgeData.memberConnection.Count, i
increments by 1
    Call method membersListView.Items.Add with "To Member: " plus
Program.bridgeData.memberConnection plus " | From Member: " plus
Program.bridgeData.memberConnection
    EndFor
    Call method bridgeDrawing.Refresh
Else
    Call method MessageBox.Show with "Error: Must Select An Item Before Clicking
It."
EndIf
```

**Flowchart**



**Procedure: clearAll\_Click**

```
private void clearAll_Click(object sender, EventArgs e)
```

**Parameters**

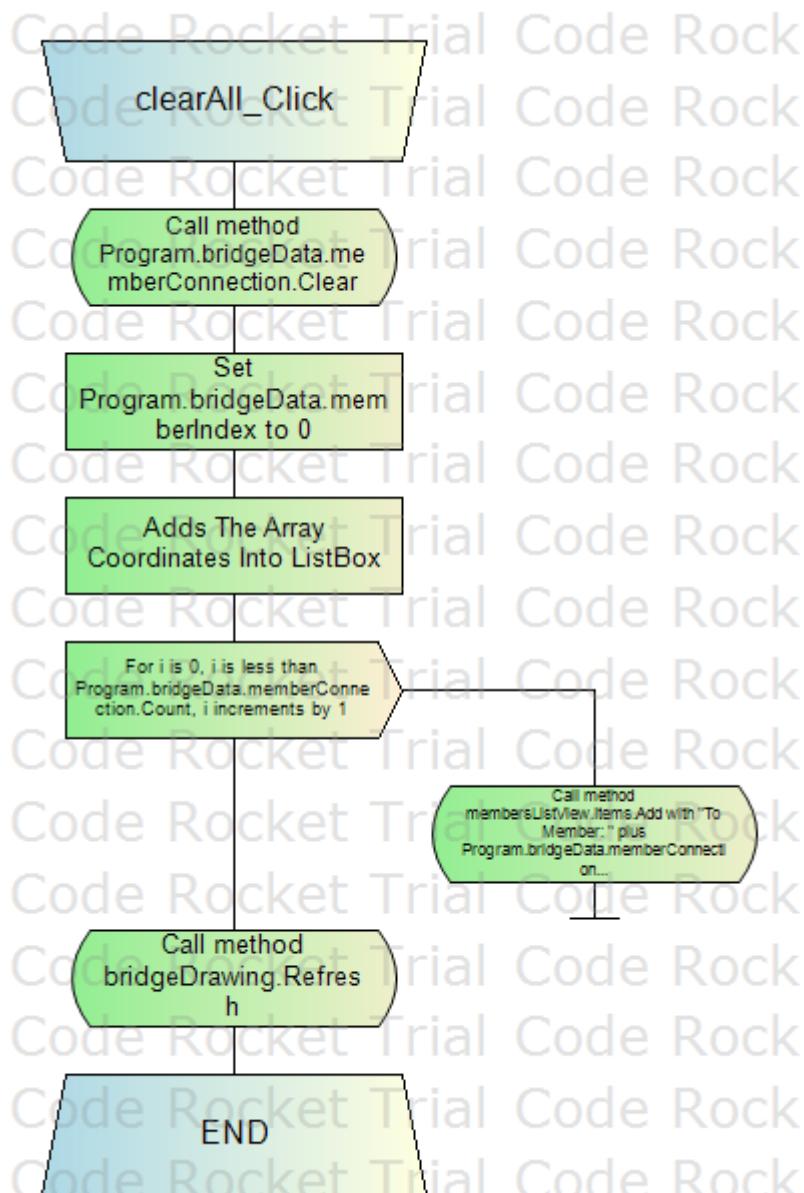
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Program.bridgeData.memberConnection.Clear  
Set Program.bridgeData.memberIndex to 0  
Adds The Array Coordinates Into ListBox  
For i is 0, i is less than Program.bridgeData.memberConnection.Count, i increments by 1  
    Call method membersListView.Items.Add with "To Member: " plus  
    Program.bridgeData.memberConnection plus " | From Member: " plus  
    Program.bridgeData.memberConnection  
EndFor  
Call method bridgeDrawing.Refresh
```

**Flowchart**



*Code File: membersTable.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

**Class: membersTable**

partial class membersTable

**Attribute: components**

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure: Dispose**

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

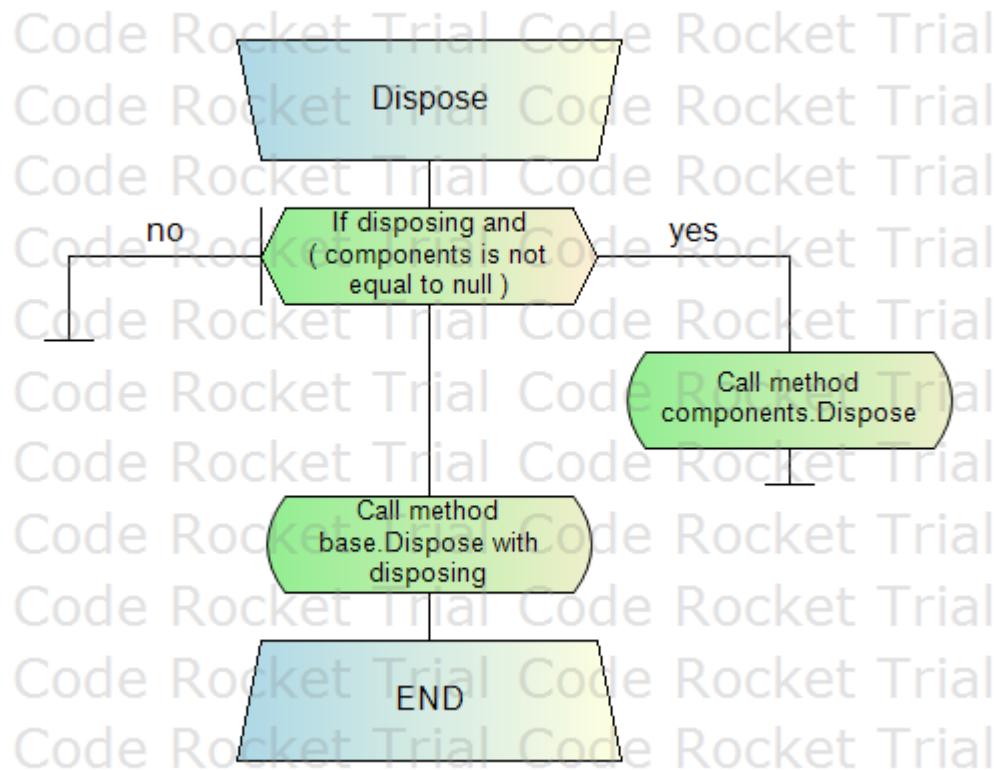
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

**Flowchart**



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.ListBox
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.ListBox
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent
Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( int ), ( int ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 24
Set this.BackToMainMenu.TabStop to false

```

```

Set this.BackToMainMenu.Text to "BACK"
Set this.BackToMainMenu.UseVisualStyleBackColor to false
This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 27
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 28
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 29
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
nodeListView
Set this.nodeTypeView.FormattingEnabled to true
Set this.nodeTypeView.ItemHeight to 25
Create new System.Drawing.Point
Set this.nodeTypeView.Name to "nodeListView"
Create new System.Drawing.Size
Set this.nodeTypeView.TabIndex to 33
clearAll
Set this.clearAll.BackColor to System.Drawing.Color.Transparent
Set this.clearAll.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.clearAll.FlatAppearance.BorderSize to 2
Set this.clearAll.FlatAppearance.MouseOverBackColor to System.Drawing.Color.FromArgb
with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.clearAll.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.clearAll.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.clearAll.Name to "clearAll"
Create new System.Drawing.Size
Set this.clearAll.TabIndex to 32
Set this.clearAll.TabStop to false
Set this.clearAll.Text to "CLEAR ALL"
Set this.clearAll.UseVisualStyleBackColor to false
This.clearAll.Click += new System.EventHandler(this.clearAll_Click);
removeMember
Set this.removeMember.BackColor to System.Drawing.Color.Transparent
Set this.removeMember.FlatAppearance.BorderColor to System.Drawing.Color.White

```

```

Set this.removeMember.FlatAppearance.BorderSize to 2
Set this.removeMember.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.removeMember.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.removeMember.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.removeMember.Name to "removeMember"
Create new System.Drawing.Size
Set this.removeMember.TabIndex to 31
Set this.removeMember.TabStop to false
Set this.removeMember.Text to "REMOVE MEMBER"
Set this.removeMember.UseVisualStyleBackColor to false
This.removeMember.Click += new System.EventHandler(this.removeMember_Click);
membersListView
Set this.membersListView.FormattingEnabled to true
Set this.membersListView.ItemHeight to 25
Create new System.Drawing.Point
Set this.membersListView.Name to "membersListView"
Create new System.Drawing.Size
Set this.membersListView.TabIndex to 30
membersTable
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.nodesListView
Call method this.Controls.Add with this.clearAll
Call method this.Controls.Add with this.removeMember
Call method this.Controls.Add with this.membersListView
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.bridgeDrawing
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.BackToMainMenu
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "membersTable"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout

```

## Flowchart



```

#endif

Attribute: minimize

private System.Windows.Forms.Button minimize;
Attribute: exitApplication

private System.Windows.Forms.Button exitApplication;
Attribute: BackToMainMenu

private System.Windows.Forms.Button BackToMainMenu;
Attribute: moveMenu

private System.Windows.Forms.Panel moveMenu;
Attribute: zoomInBar

private System.Windows.Forms.TrackBar zoomInBar;
Attribute: bridgeDrawing

private System.Windows.Forms.Panel bridgeDrawing;
Attribute: nodeListView

private System.Windows.Forms.ListBox nodeListView;
Attribute: clearAll

private System.Windows.Forms.Button clearAll;
Attribute: removeMember

private System.Windows.Forms.Button removeMember;
Attribute: membersListView

private System.Windows.Forms.ListBox membersListView;
Code File: nodes.cs

using Sectrics_V2.Properties;
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: Sectrics_V2

namespace Sectrics_V2

```

**Class: nodes**

```
public partial class nodes : Form
```

**Attribute: zoom**

```
float zoom = 1f;
```

**Attribute: xMouseOffset**

```
double xMouseOffset;
```

**Attribute: yMouseOffset**

```
double yMouseOffset;
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

**Procedure: Constructor**

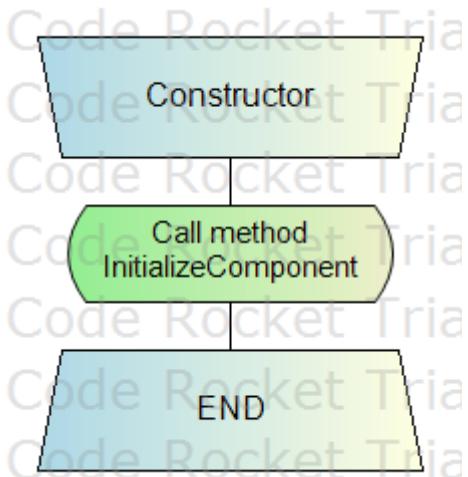
```
public nodes()
```

Returns nodes

**Pseudocode**

```
Call method InitializeComponent
```

Flowchart



**Procedure: WndProc**

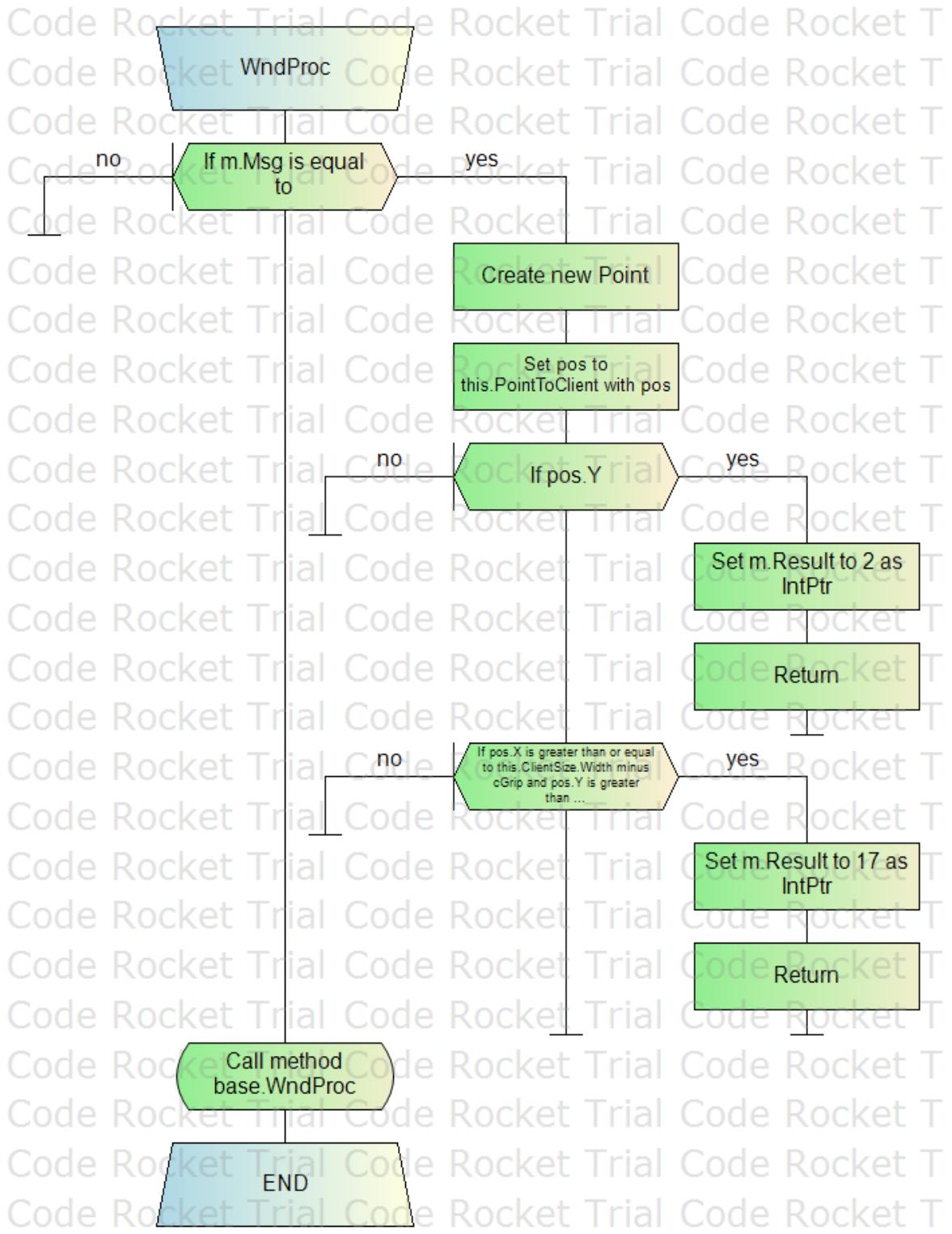
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

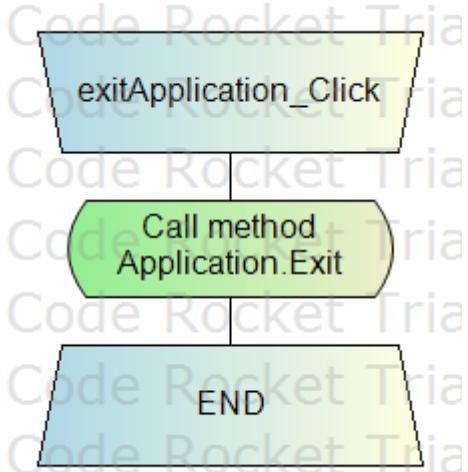
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

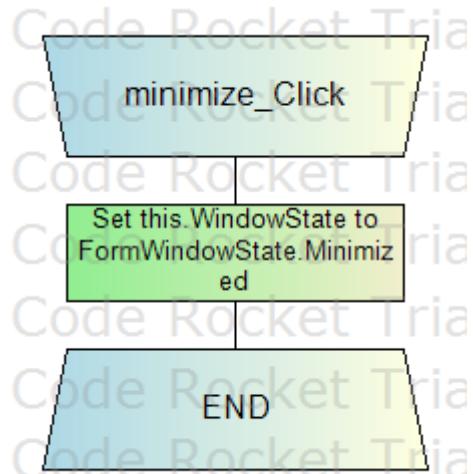
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

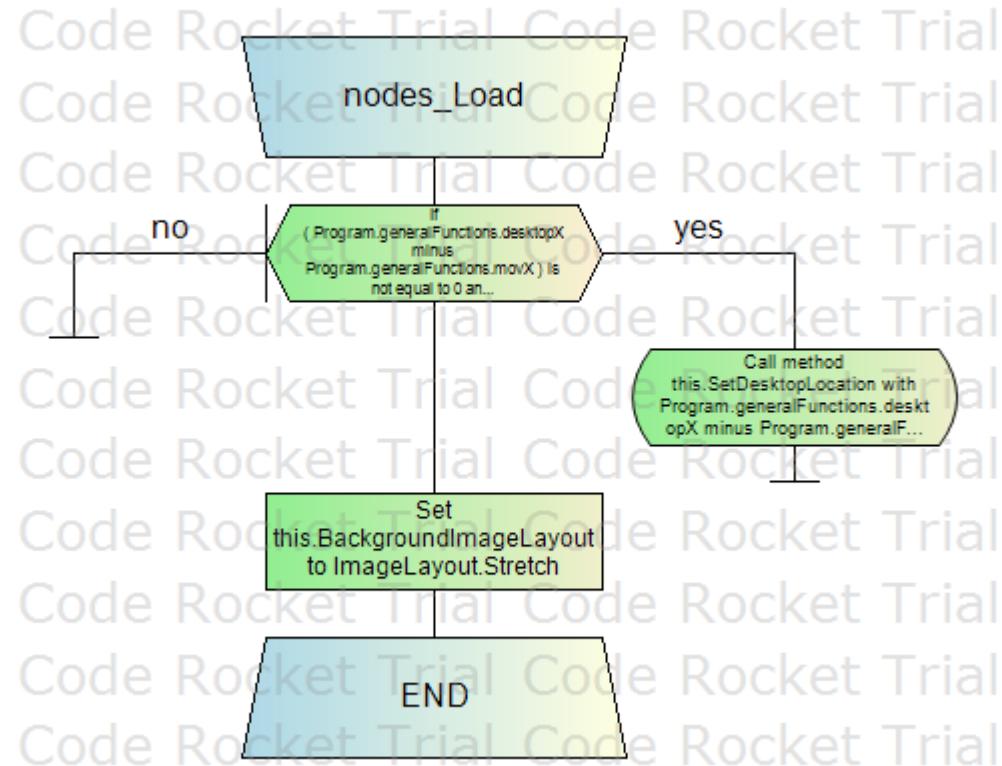
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: addNodeButton\_Click**

```
private void addNodeButton_Click(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

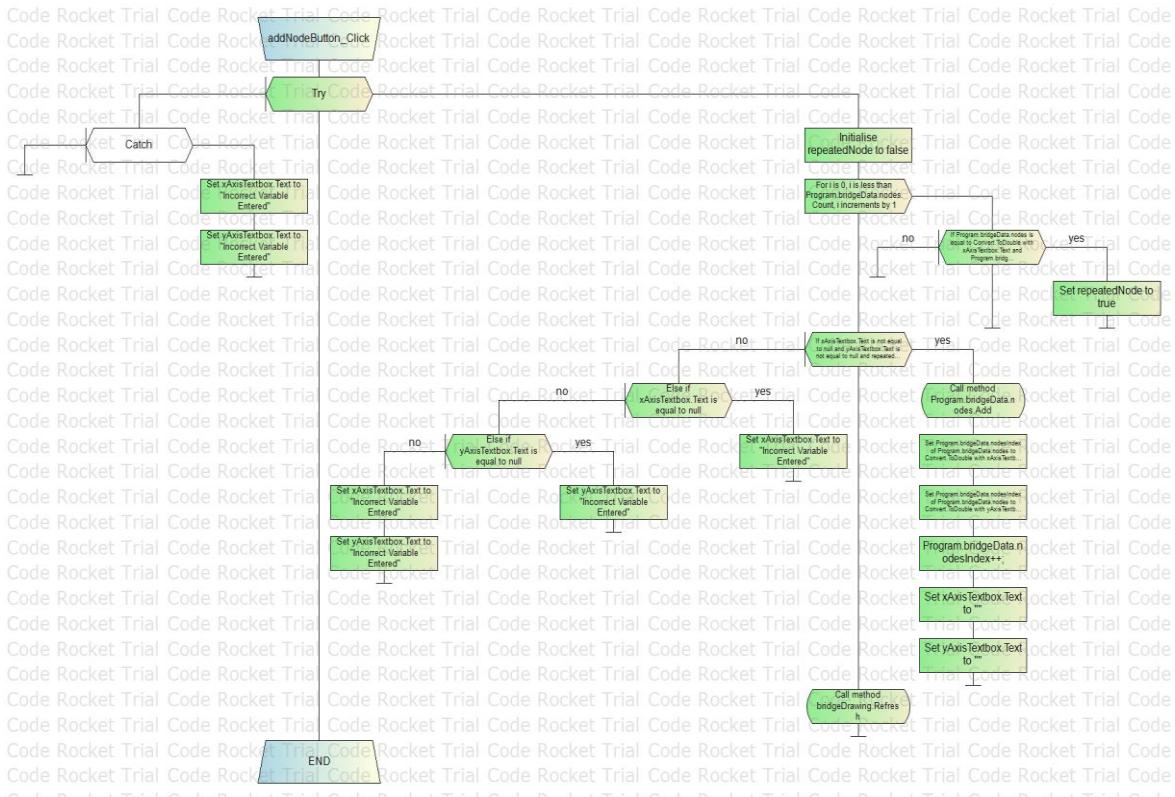
Returns Void

**Pseudocode**

**Try**

```
    Initialise repeatedNode to false
    For i is 0, i is less than Program.bridgeData.nodes.Count, i increments by 1
        If Program.bridgeData.nodes is equal to Convert.ToDouble with
            xAxisTextbox.Text and Program.bridgeData.nodes is equal to Convert.ToDouble with
            yAxisTextbox.Text
            Set repeatedNode to true
        EndIf
    EndFor
    If xAxisTextbox.Text is not equal to null and yAxisTextbox.Text is not equal
        to null and repeatedNode is equal to false
        Call method Program.bridgeData.nodes.Add
        Set Program.bridgeData.nodesIndex of Program.bridgeData.nodes to
        Convert.ToDouble with xAxisTextbox.Text
        Set Program.bridgeData.nodesIndex of Program.bridgeData.nodes to
        Convert.ToDouble with yAxisTextbox.Text
        Program.bridgeData.nodesIndex++;
        Set xAxisTextbox.Text to ""
        Set yAxisTextbox.Text to ""
    Else if xAxisTextbox.Text is equal to null
        Set xAxisTextbox.Text to "Incorrect Variable Entered"
    Else if yAxisTextbox.Text is equal to null
        Set yAxisTextbox.Text to "Incorrect Variable Entered"
    Else
        Set xAxisTextbox.Text to "Incorrect Variable Entered"
        Set yAxisTextbox.Text to "Incorrect Variable Entered"
    EndIf
    Call method bridgeDrawing.Refresh
Catch
    Set xAxisTextbox.Text to "Incorrect Variable Entered"
    Set yAxisTextbox.Text to "Incorrect Variable Entered"
EndTry
```

Flowchart



**Procedure: yAxisTextbox\_TextChanged**

```
private void yAxisTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

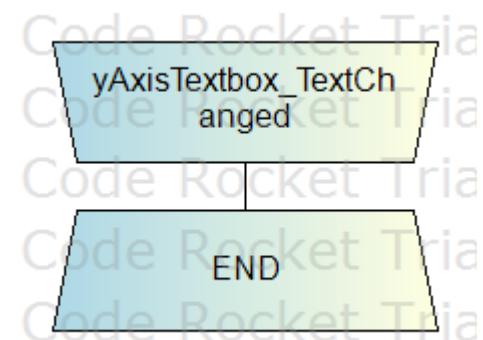
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: *xAxisTextbox\_TextChanged***

```
private void xAxisTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

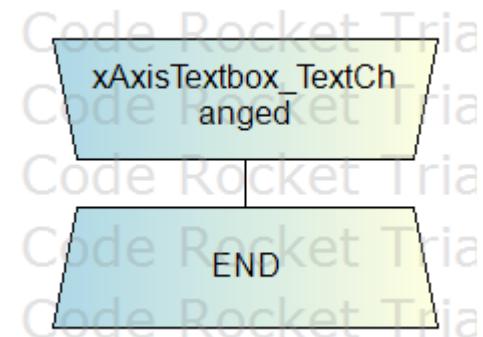
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Procedure: nodeListView\_SelectedIndexChanged*

```
private void nodeListView_SelectedIndexChanged(object sender,  
EventArgs e)
```

**Parameters**

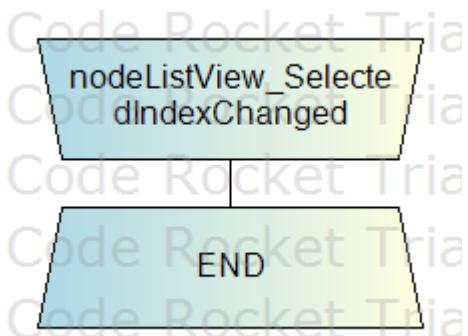
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Procedure: moveMenu\_Paint*

```
private void moveMenu_Paint(object sender, PaintEventArgs e)
```

**Parameters**

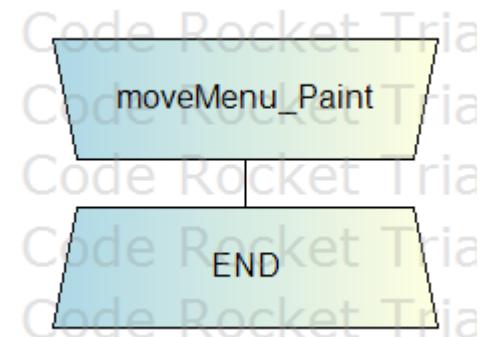
- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

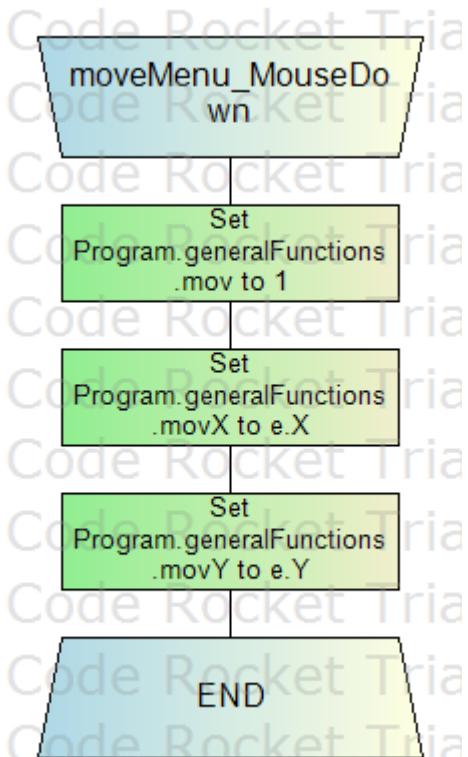
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

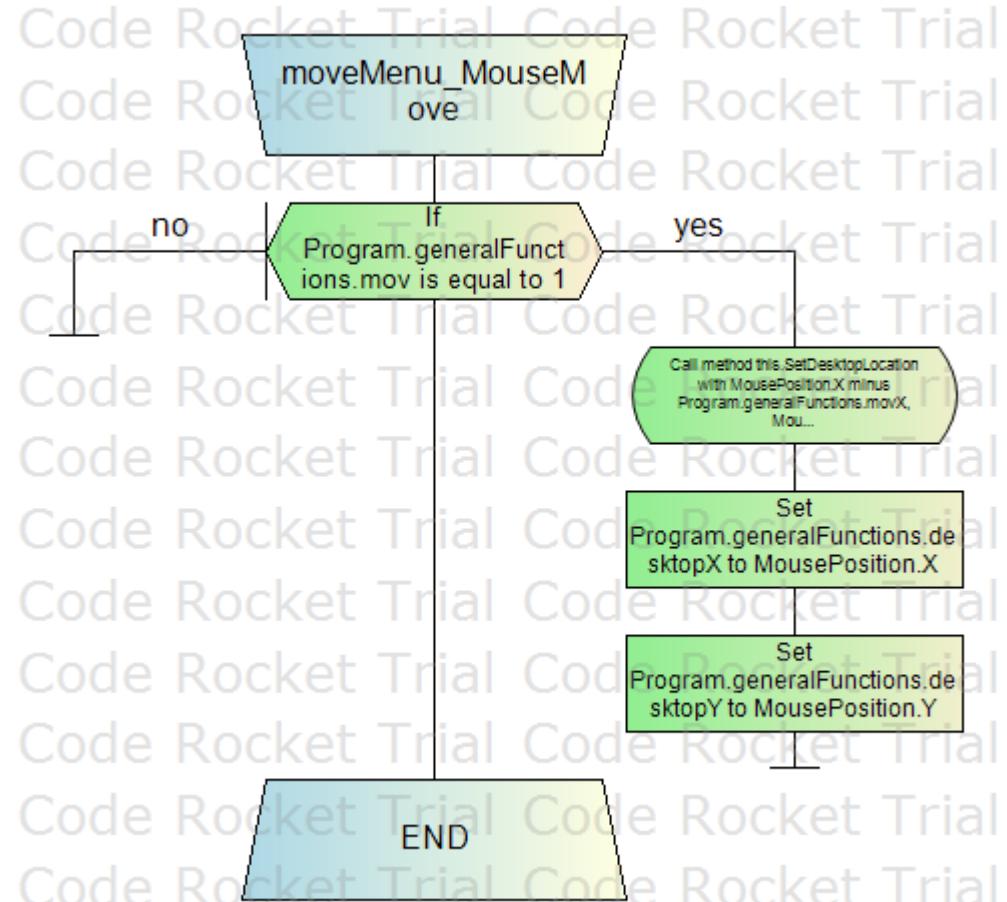
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseUp**

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

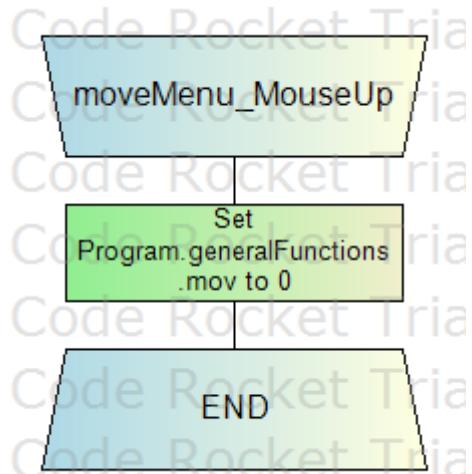
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

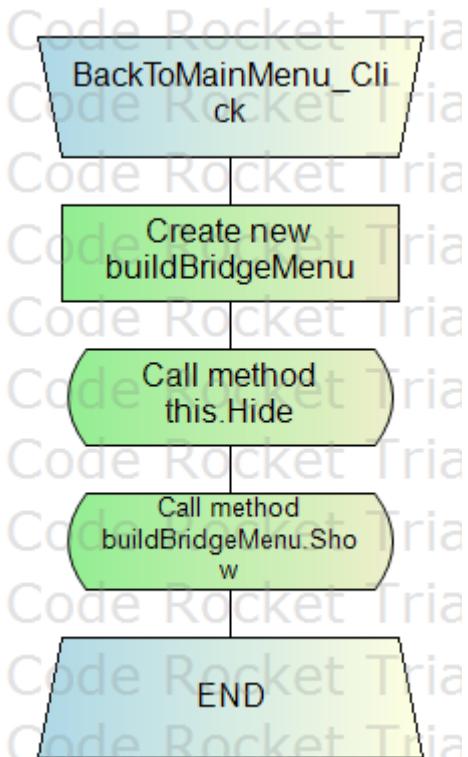
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

**Try**

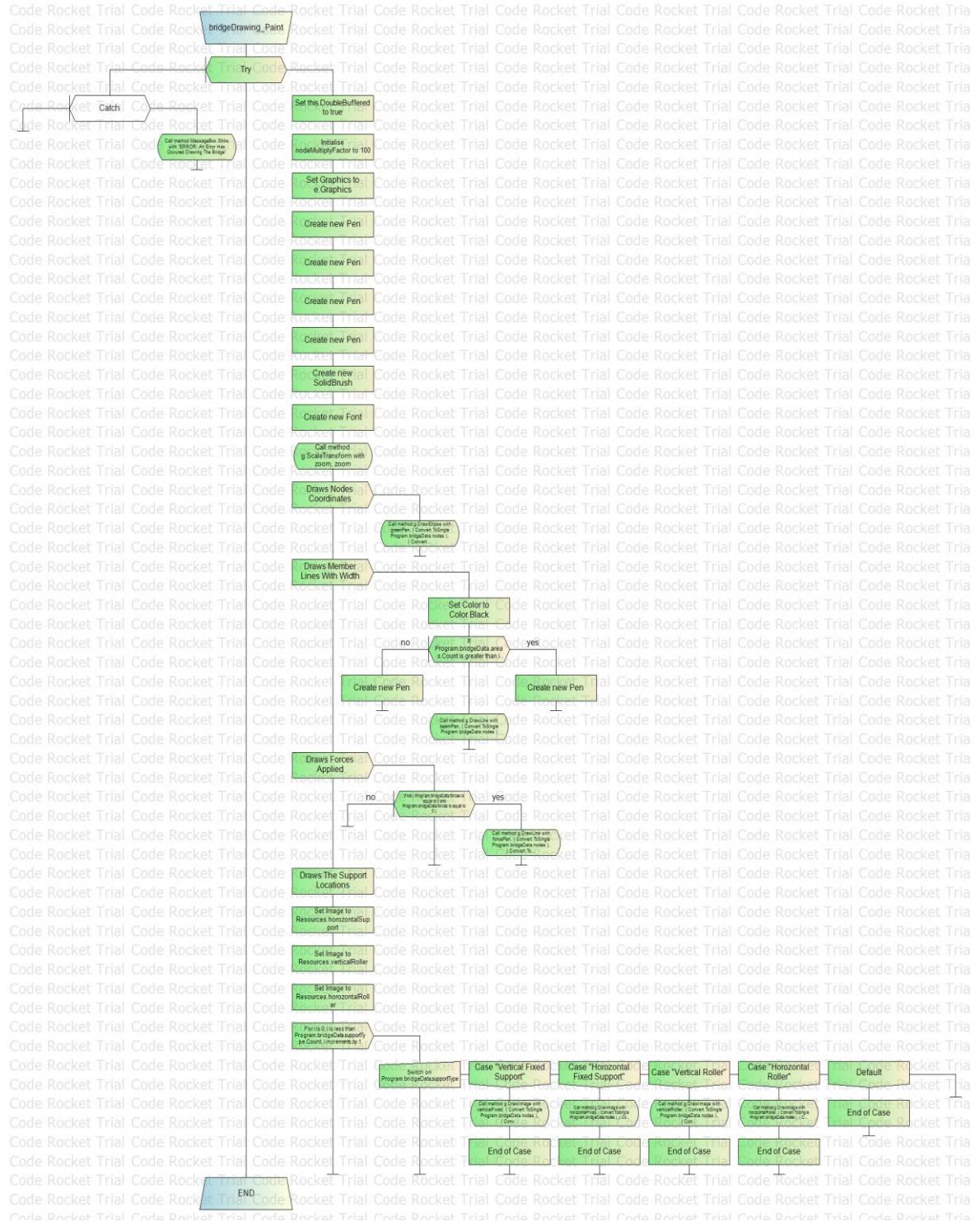
```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```

        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry

```

## Flowchart



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

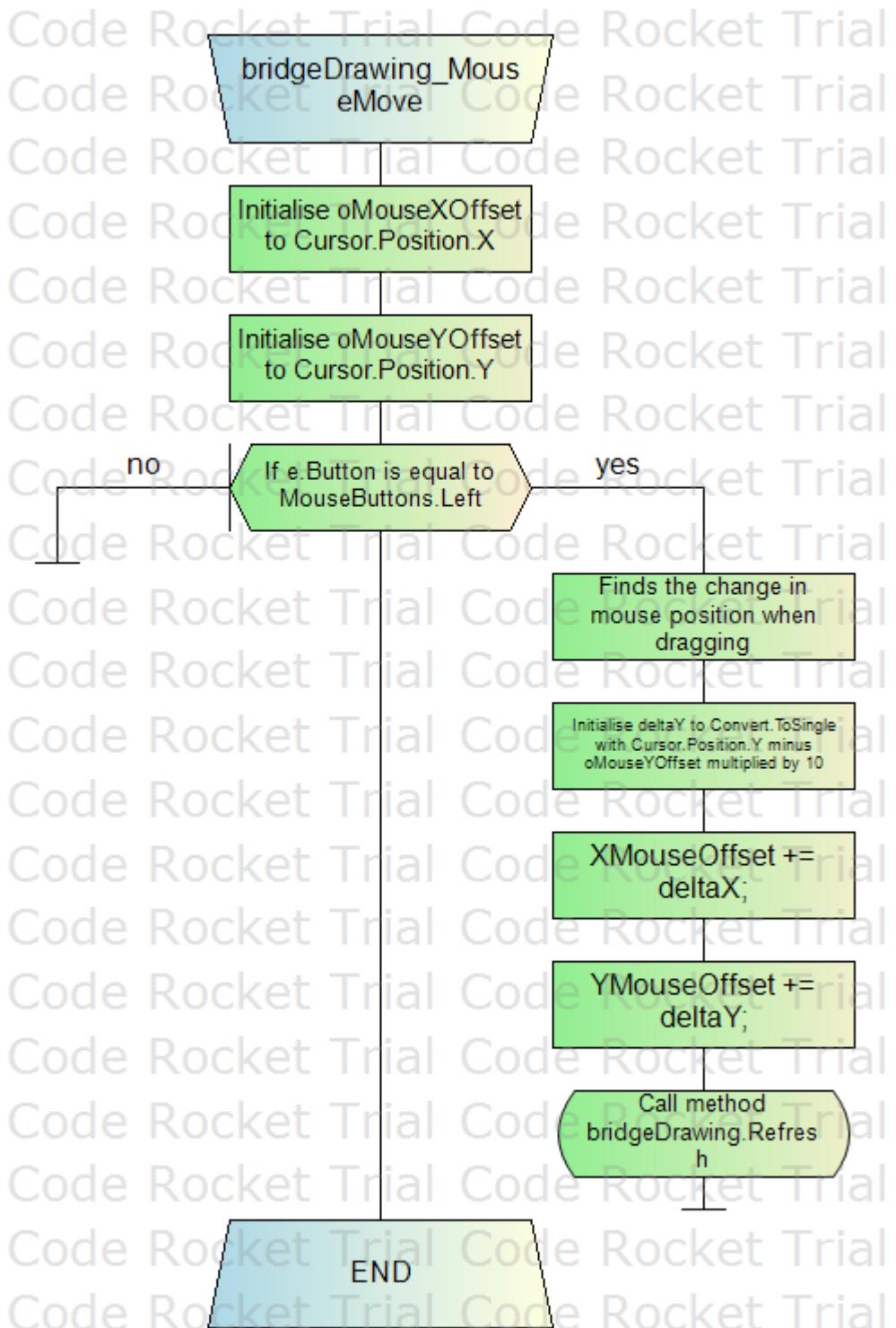
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

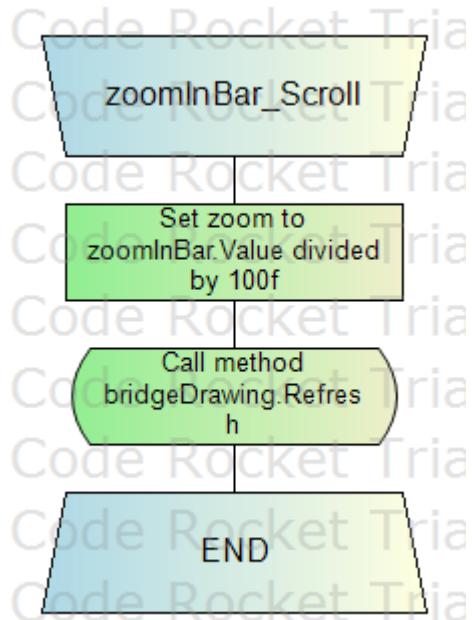
Returns Void

**Pseudocode**

```
Set zoom to zoomInBar.Value divided by 100f
```

```
Call method bridgeDrawing.Refresh
```

**Flowchart**



**Procedure: nodesTable\_Click**

```
private void nodesTable_Click(object sender, EventArgs e)
```

**Parameters**

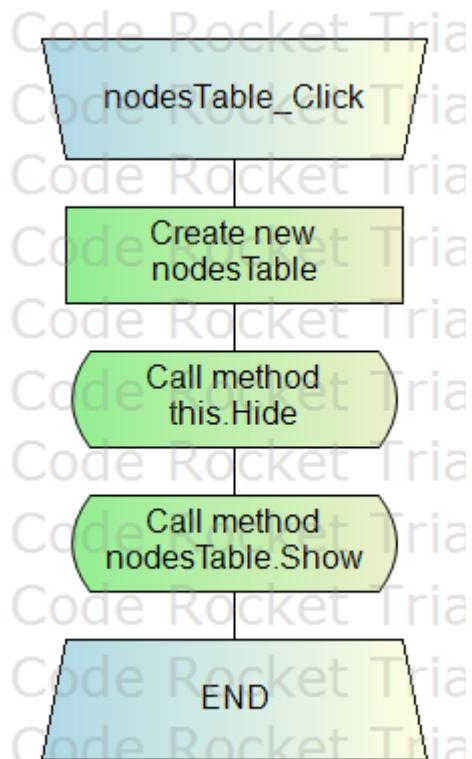
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new nodesTable  
Call method this.Hide  
Call method nodesTable.Show
```

**Flowchart**



*Code File: nodes.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

**Class: nodes**

partial class nodes

**Attribute: components**

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure: Dispose**

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

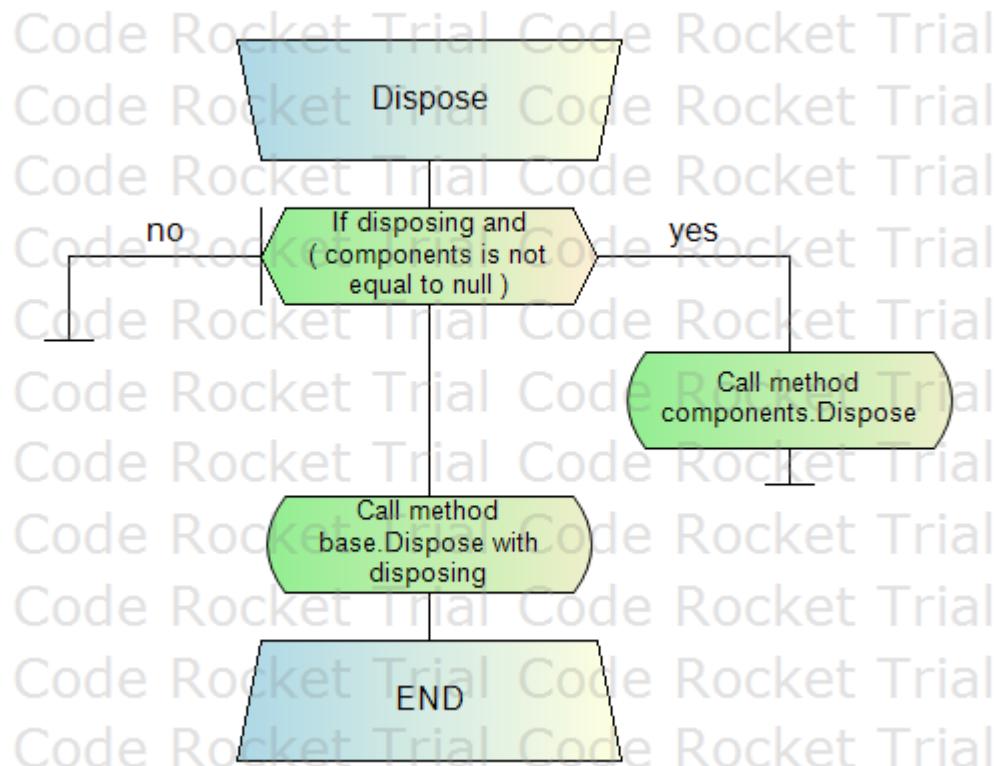
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

Flowchart



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
label1
Set this.label1.AutoSize to true
Set this.label1.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 12
Set this.label1.Text to "X AXIS COORDINATES:"
xAxisTextbox
Create new System.Drawing.Font
Create new System.Drawing.Point
Set this.xAxisTextbox.Name to "xAxisTextbox"
Create new System.Drawing.Size
Set this.xAxisTextbox.TabIndex to 0
Set this.xAxisTextbox.Text to "Enter the x-axis coordinates here."
This.xAxisTextbox.TextChanged += new
System.EventHandler(this.xAxisTextbox_TextChanged);
yAxisTextbox
Create new System.Drawing.Font
Create new System.Drawing.Point
Set this.yAxisTextbox.Name to "yAxisTextbox"
Create new System.Drawing.Size
Set this.yAxisTextbox.TabIndex to 1
Set this.yAxisTextbox.Text to "Enter the y-axis coordinates here."
This.yAxisTextbox.TextChanged += new
System.EventHandler(this.yAxisTextbox_TextChanged);
label2
Set this.label2.AutoSize to true
Set this.label2.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label2.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label2.Name to "label2"
Create new System.Drawing.Size
Set this.label2.TabIndex to 14
Set this.label2.Text to "Y AXIS COORDINATES:"
addNodeButton
Set this.addNodeButton.BackColor to System.Drawing.Color.Transparent

```

```

Set this.addNodeButton.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.addNodeButton.FlatAppearance.BorderSize to 2
Set this.addNodeButton.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ), ( ( int ) )
Set this.addNodeButton.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.addNodeButton.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.addNodeButton.Name to "addNodeButton"
Create new System.Drawing.Size
Set this.addNodeButton.TabIndex to 16
Set this.addNodeButton.TabStop to false
Set this.addNodeButton.Text to "ADD NODE"
Set this.addNodeButton.UseVisualStyleBackColor to false
This.addNodeButton.Click += new System.EventHandler(this.addNodeButton_Click);
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Set this.moveMenu.ForeColor to System.Drawing.Color.Transparent
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 22
This.moveMenu.Paint += new
System.Windows.Forms.PaintEventHandler(this.moveMenu_Paint);
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent
Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ), ( ( int ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 23
Set this.BackToMainMenu.TabStop to false
Set this.BackToMainMenu.Text to "BACK"
Set this.BackToMainMenu.UseVisualStyleBackColor to false
This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 27
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"

```

```

Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 26
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
minimize
Set this.minimize.BackColor to System.Drawing.Color.Transparent
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.TabStop to false
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackColor to System.Drawing.Color.Transparent
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.TabStop to false
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
nodesTable
Set this.nodesTable.BackColor to System.Drawing.Color.Transparent
Set this.nodesTable.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.nodesTable.FlatAppearance.BorderSize to 2
Set this.nodesTable.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ) ), ( ( int ) ) )
Set this.nodesTable.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.nodesTable.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.nodesTable.Name to "nodesTable"
Create new System.Drawing.Size
Set this.nodesTable.TabIndex to 28
Set this.nodesTable.TabStop to false
Set this.nodesTable.Text to "TABLE"
Set this.nodesTable.UseVisualStyleBackColor to false
This.nodesTable.Click += new System.EventHandler(this.nodesTable_Click);
nodes
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.nodesTable
Call method this.Controls.Add with this.addNodeButton
Call method this.Controls.Add with this.yAxisTextbox
Call method this.Controls.Add with this.label2
Call method this.Controls.Add with this.xAxisTextbox

```

```
Call method this.Controls.Add with this.label1
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.BackToMainMenu
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Call method this.Controls.Add with this.bridgeDrawing
Set this.DoubleBuffered to true
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "nodes"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout
```

### Flowchart



```

#endifregion
Attribute: minimize

private System.Windows.Forms.Button minimize;
Attribute: exitApplication

private System.Windows.Forms.Button exitApplication;
Attribute: label1

private System.Windows.Forms.Label label1;
Attribute: xAxisTextbox

private System.Windows.Forms.TextBox xAxisTextbox;
Attribute: yAxisTextbox

private System.Windows.Forms.TextBox yAxisTextbox;
Attribute: label2

private System.Windows.Forms.Label label2;
Attribute: addNodeButton

private System.Windows.Forms.Button addNodeButton;
Attribute: moveMenu

private System.Windows.Forms.Panel moveMenu;
Attribute: BackToMainMenu

private System.Windows.Forms.Button BackToMainMenu;
Attribute: bridgeDrawing

private System.Windows.Forms.Panel bridgeDrawing;
Attribute: zoomInBar

private System.Windows.Forms.TrackBar zoomInBar;
Attribute: nodesTable

private System.Windows.Forms.Button nodesTable;
Code File: nodesTable.cs

using Sectrics_V2.Properties;
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: nodesTable**

```
public partial class nodesTable : Form
```

**Attribute: zoom**

```
float zoom = 1f;
```

**Attribute: xMouseOffset**

```
double xMouseOffset;
```

**Attribute: yMouseOffset**

```
double yMouseOffset;
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

**Procedure: Constructor**

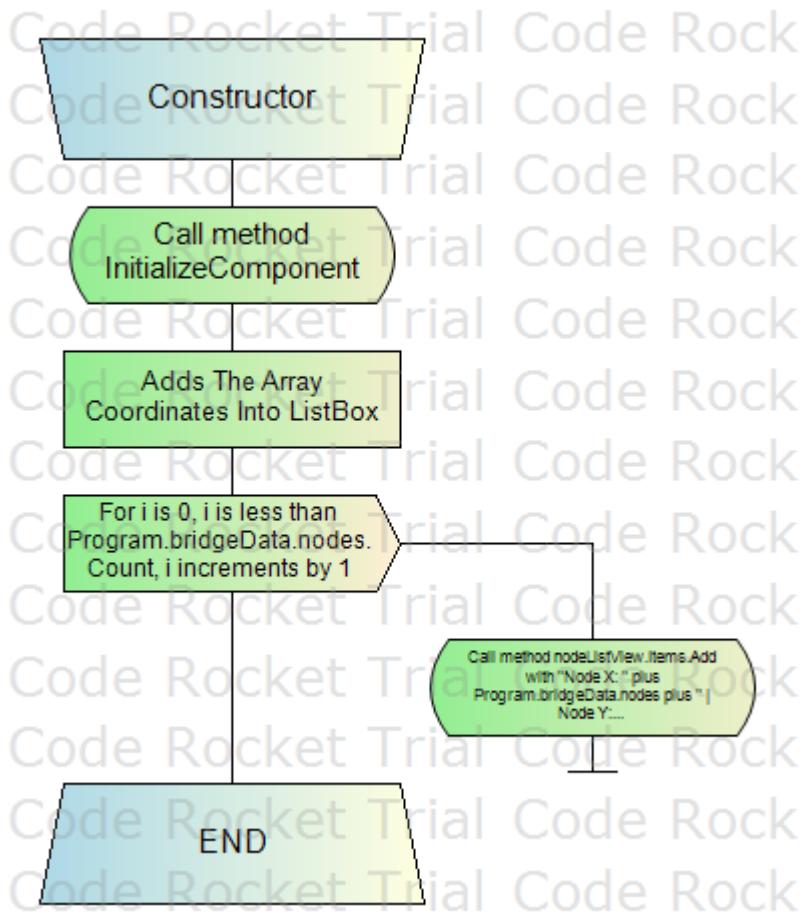
```
public nodesTable()
```

Returns nodesTable

**Pseudocode**

```
Call method InitializeComponent
Adds The Array Coordinates Into ListBox
For i is 0, i is less than Program.bridgeData.nodes.Count, i increments by 1
    Call method nodeListView.Items.Add with "Node X: " plus
    Program.bridgeData.nodes plus " | Node Y: " plus Program.bridgeData.nodes
EndFor
```

Flowchart



**Procedure: WndProc**

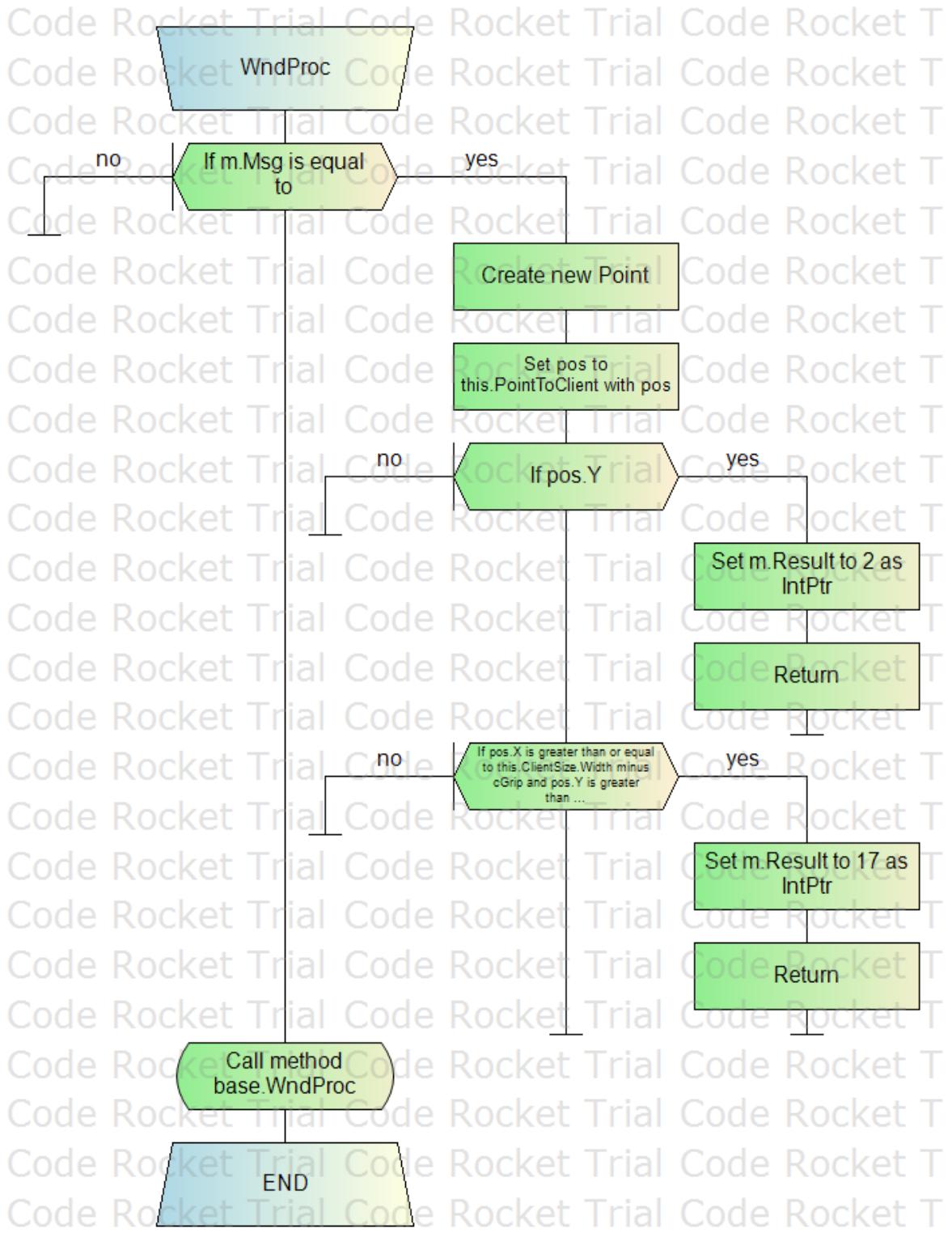
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

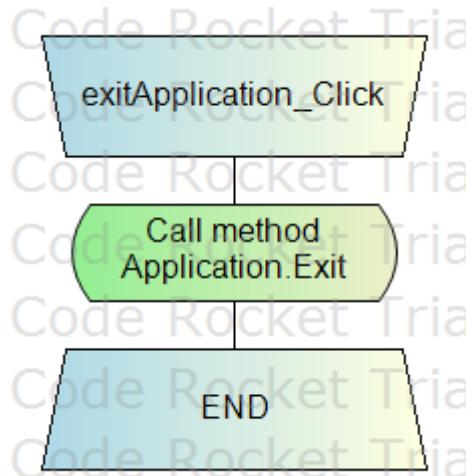
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

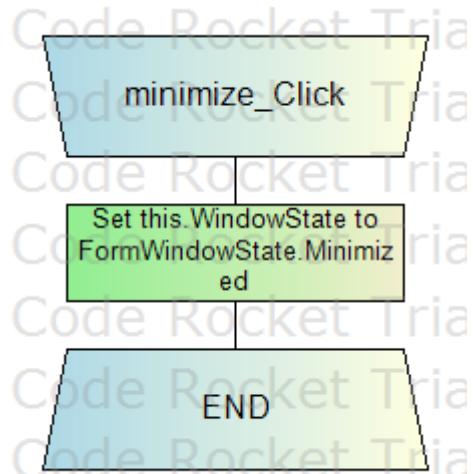
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

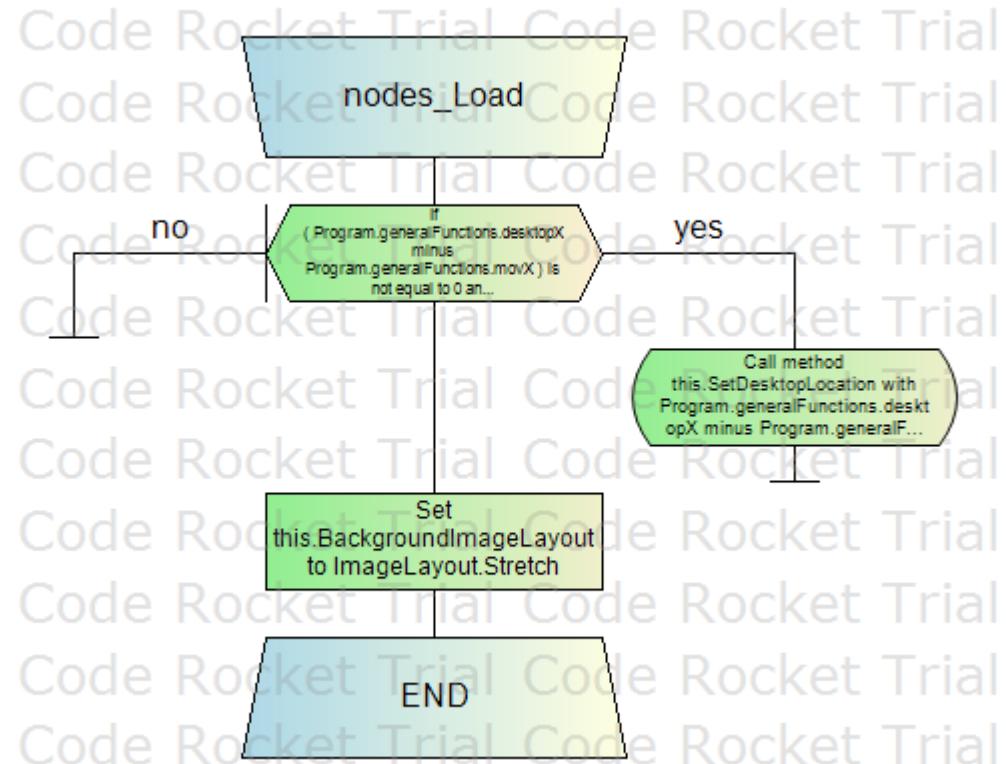
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: clearAll\_Click**

```
private void clearAll_Click(object sender, EventArgs e)
```

**Parameters**

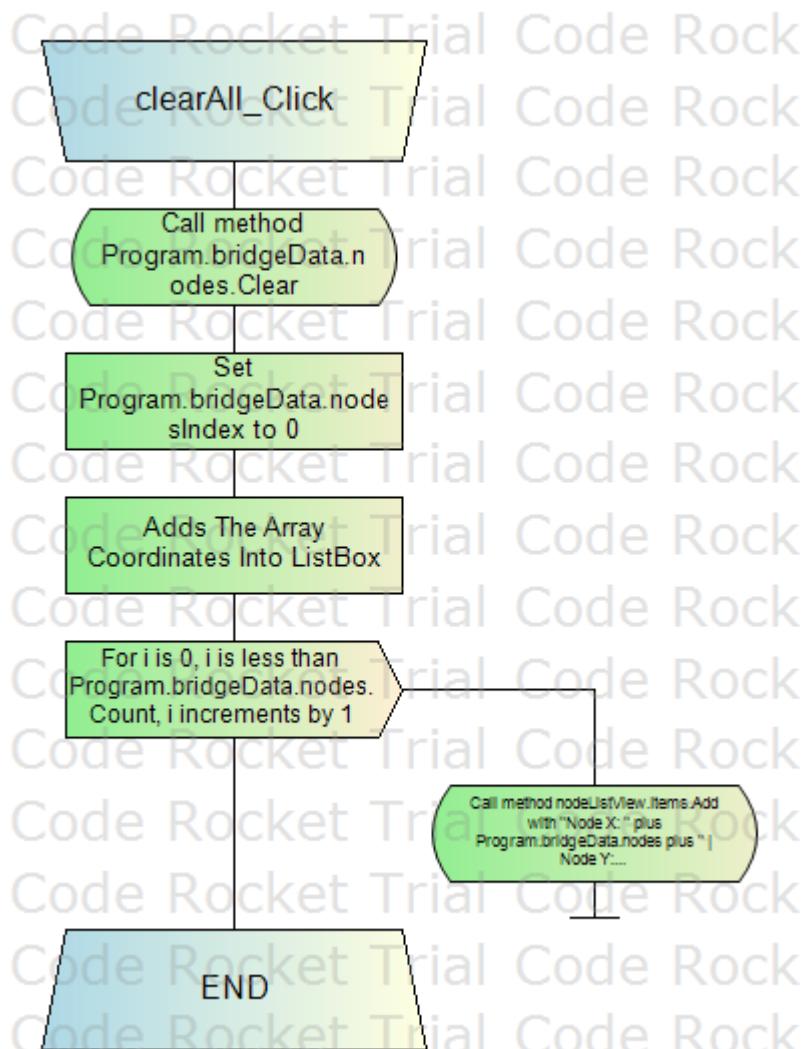
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Program.bridgeData.nodes.Clear  
Set Program.bridgeData.nodesIndex to 0  
Adds The Array Coordinates Into ListBox  
For i is 0, i is less than Program.bridgeData.nodes.Count, i increments by 1  
    Call method nodeListView.Items.Add with "Node X: " plus  
    Program.bridgeData.nodes plus " | Node Y: " plus Program.bridgeData.nodes  
EndFor
```

**Flowchart**



**Procedure: removeNode\_Click**

```
private void removeNode_Click(object sender, EventArgs e)
```

**Parameters**

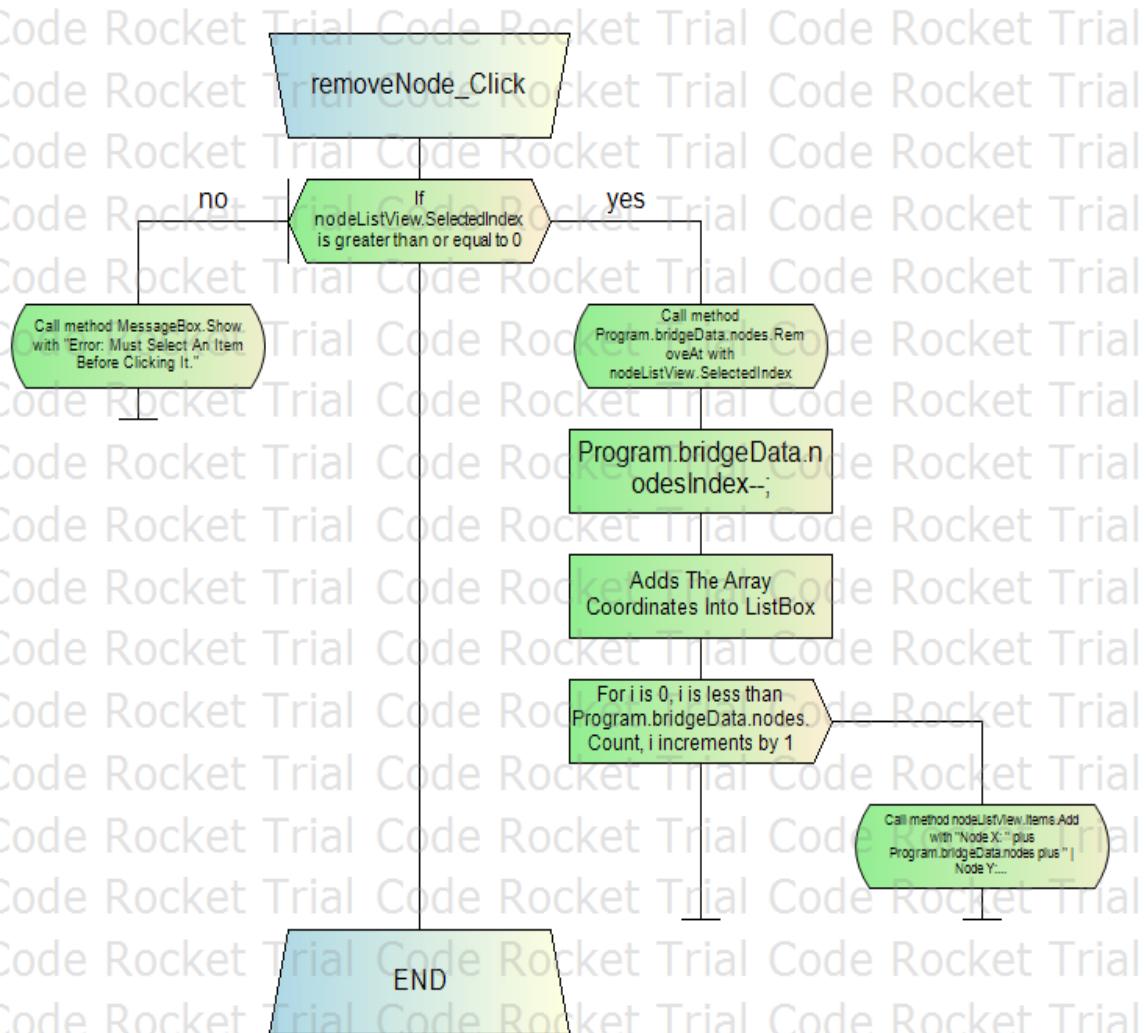
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If nodeListView.SelectedIndex is greater than or equal to 0
    Call method Program.bridgeData.nodes.RemoveAt with nodeListView.SelectedIndex
    Program.bridgeData.nodesIndex--;
    Adds The Array Coordinates Into ListBox
    For i is 0, i is less than Program.bridgeData.nodes.Count, i increments by 1
        Call method nodeListView.Items.Add with "Node X: " plus
Program.bridgeData.nodes plus " | Node Y: " plus Program.bridgeData.nodes
    EndFor
Else
    Call method MessageBox.Show with "Error: Must Select An Item Before Clicking
It."
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

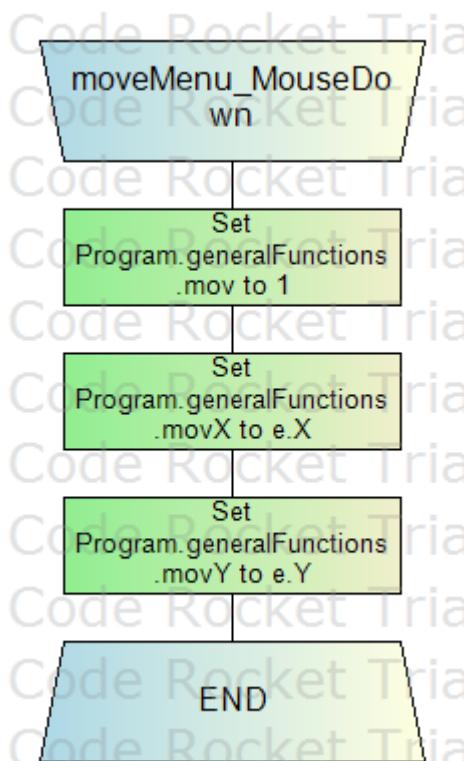
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

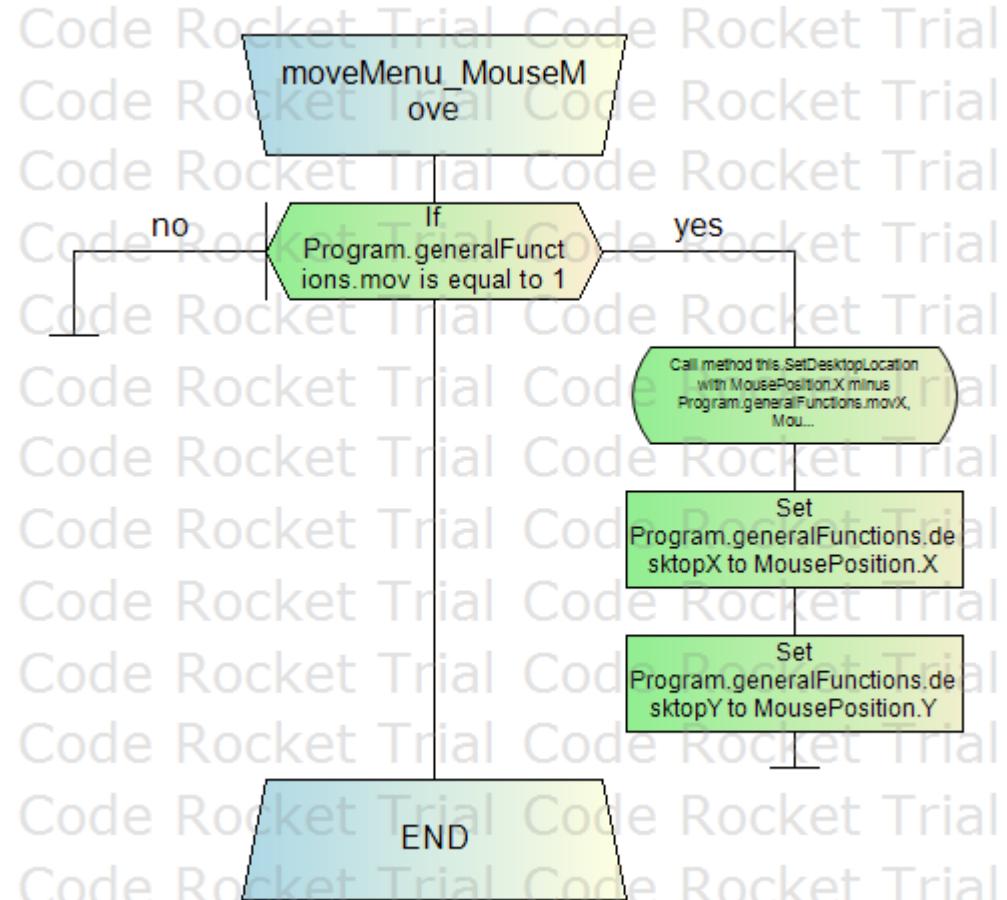
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseUp**

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

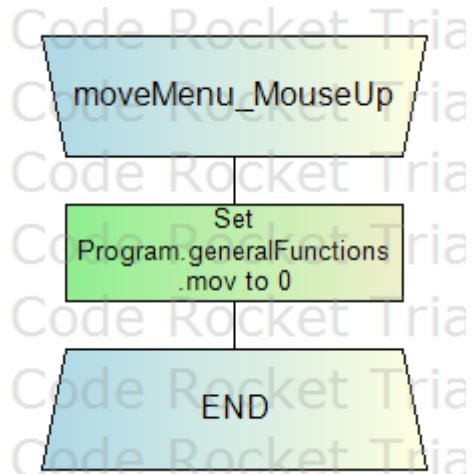
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: backButton\_Click**

```
private void backButton_Click(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

Returns Void

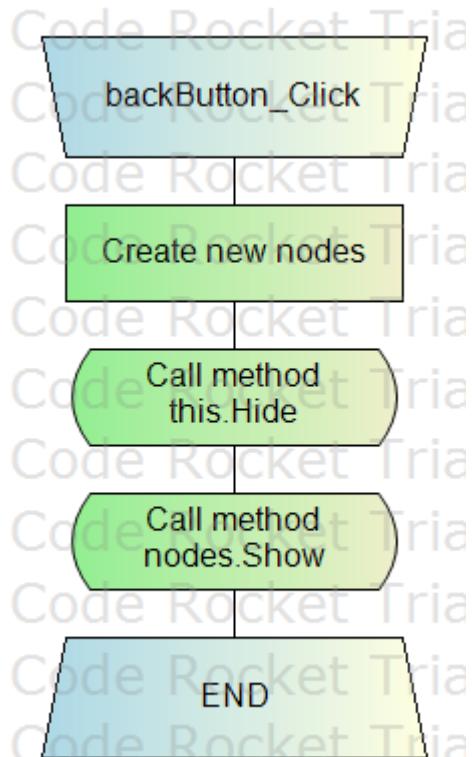
**Pseudocode**

Create new nodes

Call method this.Hide

Call method nodes.Show

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

**Try**

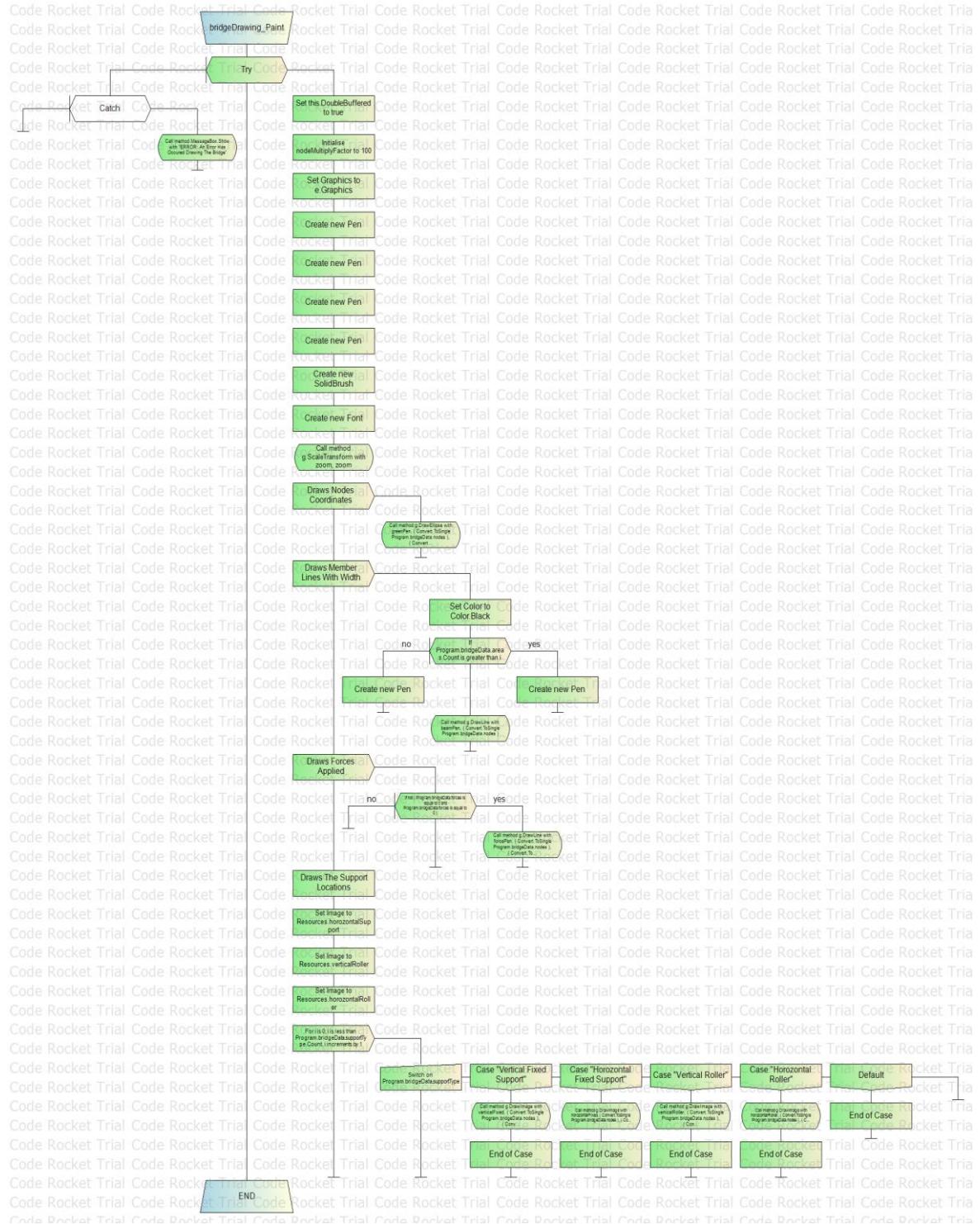
```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```

        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry

```

### Flowchart



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

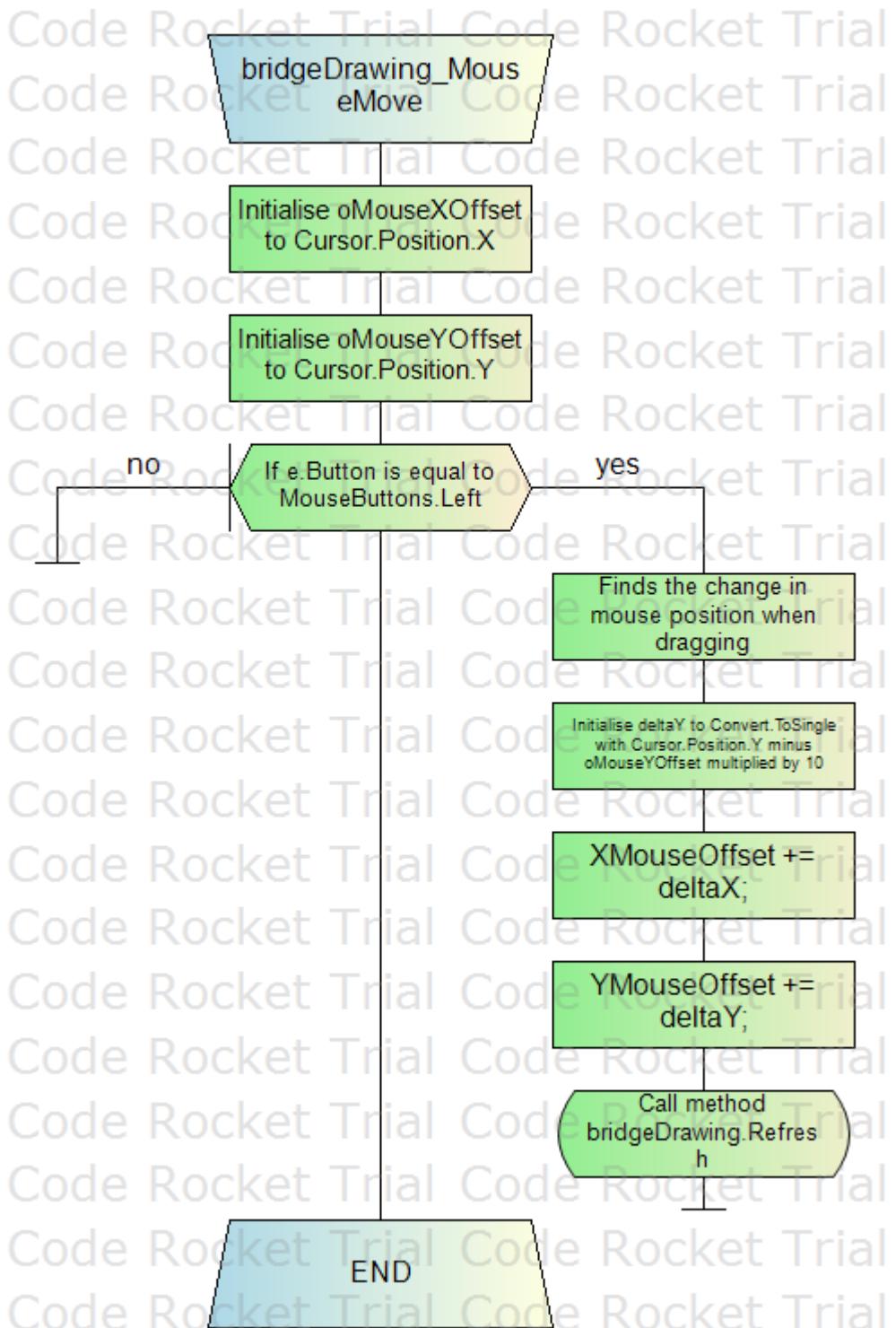
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

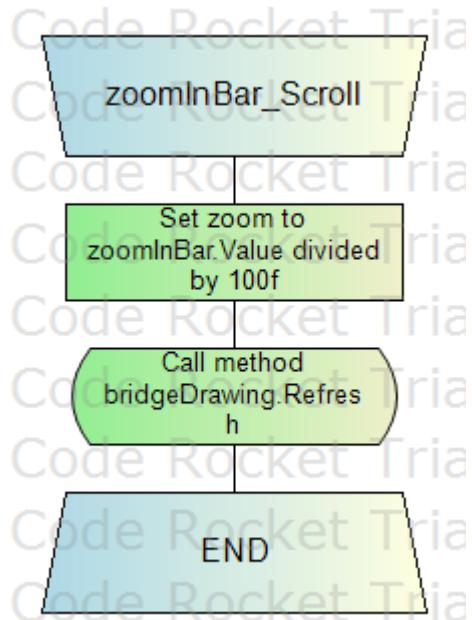
Returns Void

**Pseudocode**

```
Set zoom to zoomInBar.Value divided by 100f
```

```
Call method bridgeDrawing.Refresh
```

**Flowchart**



**Code File:** nodesTable.Designer.cs

**Namespace:** Sectrics\_V2

namespace Sectrics\_V2

**Class:** nodesTable

partial class nodesTable

**Attribute:** components

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure:** Dispose

protected override void Dispose(bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

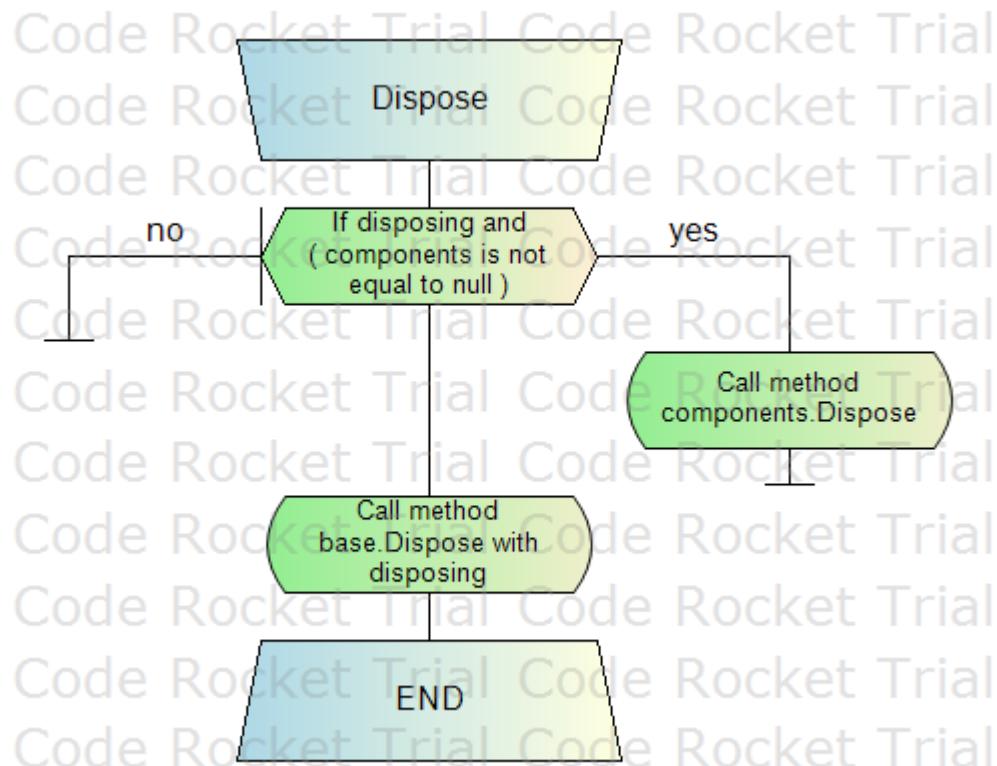
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

Flowchart



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.ListBox
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.Panel
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
backButton
Set this.backButton.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.backButton.FlatAppearance.BorderSize to 2
Set this.backButton.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( ( int ) ), ( ( int ) ) )
Set this.backButton.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.backButton.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.backButton.Name to "backButton"
Create new System.Drawing.Size
Set this.backButton.TabIndex to 12
Set this.backButton.Text to "BACK"
Set this.backButton.UseVisualStyleBackColor to false
This.backButton.Click += new System.EventHandler(this.backButton_Click);

```

```

clearAll
Set this.clearAll.BackColor to System.Drawing.Color.Transparent
Set this.clearAll.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.clearAll.FlatAppearance.BorderSize to 2
Set this.clearAll.FlatAppearance.MouseOverBackColor to System.Drawing.Color.FromArgb
with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.clearAll.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.clearAll.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.clearAll.Name to "clearAll"
Create new System.Drawing.Size
Set this.clearAll.TabIndex to 22
Set this.clearAll.TabStop to false
Set this.clearAll.Text to "CLEAR ALL"
Set this.clearAll.UseVisualStyleBackColor to false
This.clearAll.Click += new System.EventHandler(this.clearAll_Click);
removeNode
Set this.removeNode.BackColor to System.Drawing.Color.Transparent
Set this.removeNode.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.removeNode.FlatAppearance.BorderSize to 2
Set this.removeNode.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.removeNode.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.removeNode.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.removeNode.Name to "removeNode"
Create new System.Drawing.Size
Set this.removeNode.TabIndex to 21
Set this.removeNode.TabStop to false
Set this.removeNode.Text to "REMOVE NODE"
Set this.removeNode.UseVisualStyleBackColor to false
This.removeNode.Click += new System.EventHandler(this.removeNode_Click);
nodeListView
Set this.nodeListView.FormattingEnabled to true
Set this.nodeListView.ItemHeight to 25
Create new System.Drawing.Point
Set this.nodeListView.Name to "nodeListView"
Create new System.Drawing.Size
Set this.nodeListView.TabIndex to 20
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Set this.moveMenu.ForeColor to System.Drawing.Color.Transparent
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 23
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 28
Set this.zoomInBar.TabStop to false

```

```

Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 29
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
nodesTable
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.bridgeDrawing
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.clearAll
Call method this.Controls.Add with this.removeNode
Call method this.Controls.Add with this.nodeListView
Call method this.Controls.Add with this.backButton
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "nodesTable"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitialize.ISupportInitializeInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout

```

## Flowchart



```

#endif

Attribute: minimize

private System.Windows.Forms.Button minimize;
Attribute: exitApplication

private System.Windows.Forms.Button exitApplication;
Attribute: backButton

private System.Windows.Forms.Button backButton;
Attribute: clearAll

private System.Windows.Forms.Button clearAll;
Attribute: removeNode

private System.Windows.Forms.Button removeNode;
Attribute: nodeListView

private System.Windows.Forms.ListBox nodeListView;
Attribute: moveMenu

private System.Windows.Forms.Panel moveMenu;
Attribute: zoomInBar

private System.Windows.Forms.TrackBar zoomInBar;
Attribute: bridgeDrawing

private System.Windows.Forms.Panel bridgeDrawing;
Code File: solveMenus.cs

using Sectrics_V2.Properties;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Drawing.Imaging;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Diagnostics;
using System.Text.RegularExpressions;

```

**Namespace: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: solveMenus**

```
public partial class solveMenus : Form
```

**Attribute: stressForPanel**

```
double[] stressForPanel = new  
double[Program.bridgeData.memberConnection.Count];
```

**Attribute: reactionForceForPanel**

```
double[] reactionForceForPanel = new  
double[Program.bridgeData.supportNode.Count];
```

**Attribute: zoom**

```
float zoom = 1f;
```

**Attribute: xMouseOffset**

```
double xMouseOffset;
```

**Attribute: yMouseOffset**

```
double yMouseOffset;
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

**Procedure: Constructor**

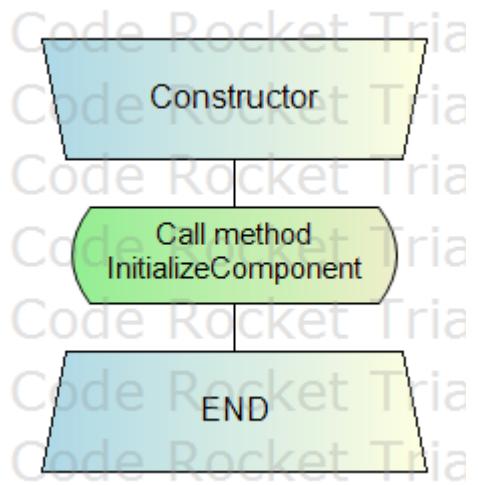
```
public solveMenus()
```

Returns solveMenus

**Pseudocode**

```
Call method InitializeComponent
```

Flowchart



**Procedure: WndProc**

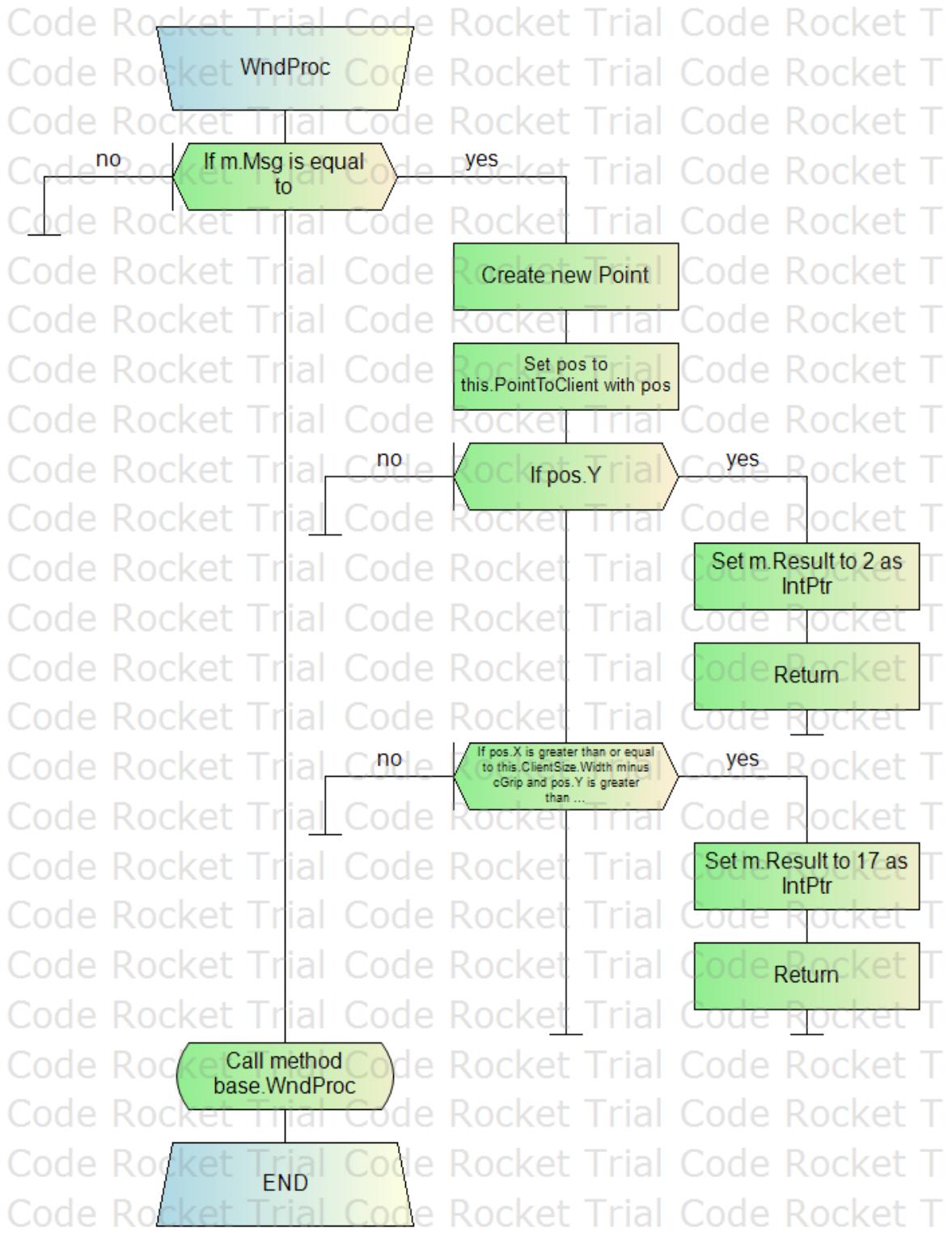
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

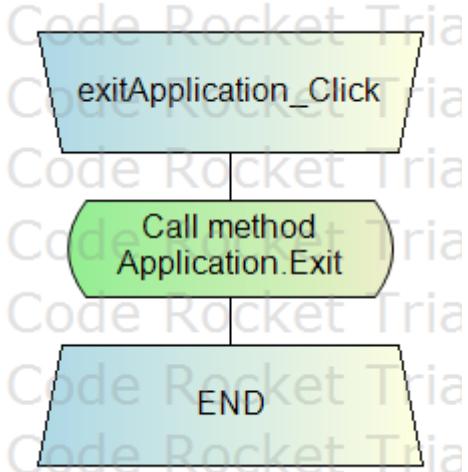
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

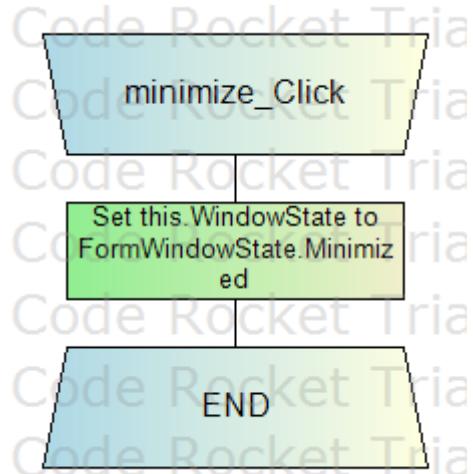
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

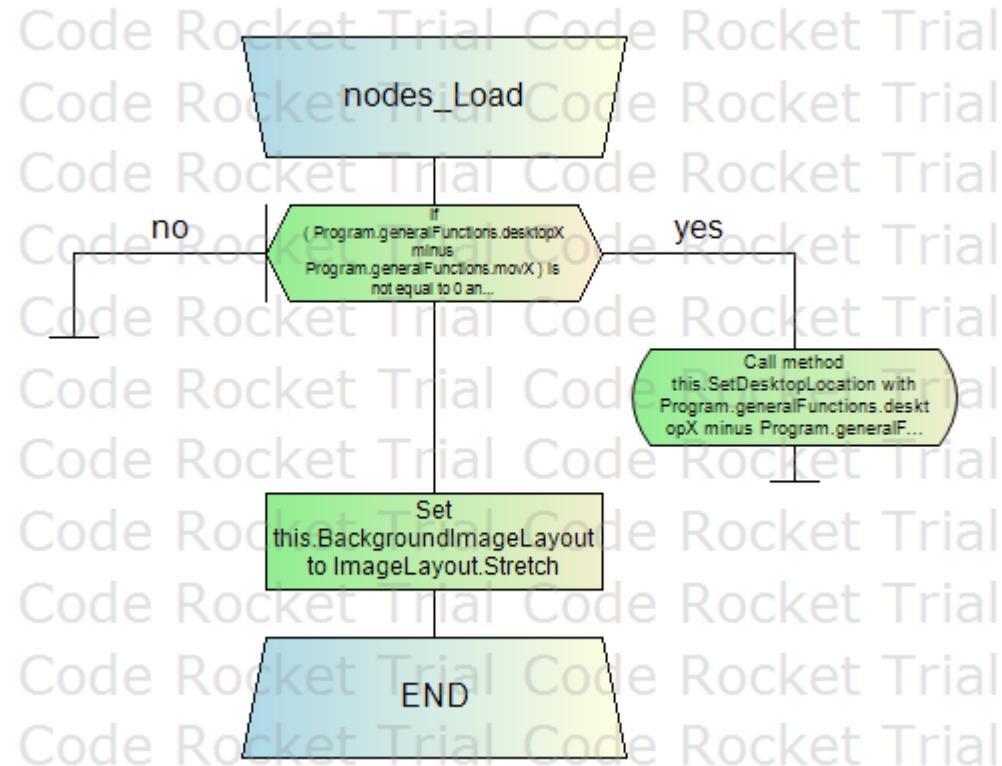
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



*Procedure: barStressesTextbox\_TextChanged*

```
private void barStressesTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

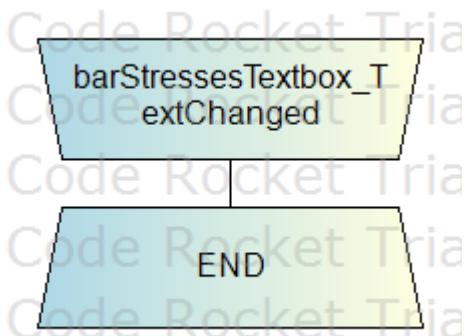
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: solveForForces\_Click**

```
private void solveForForces_Click(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

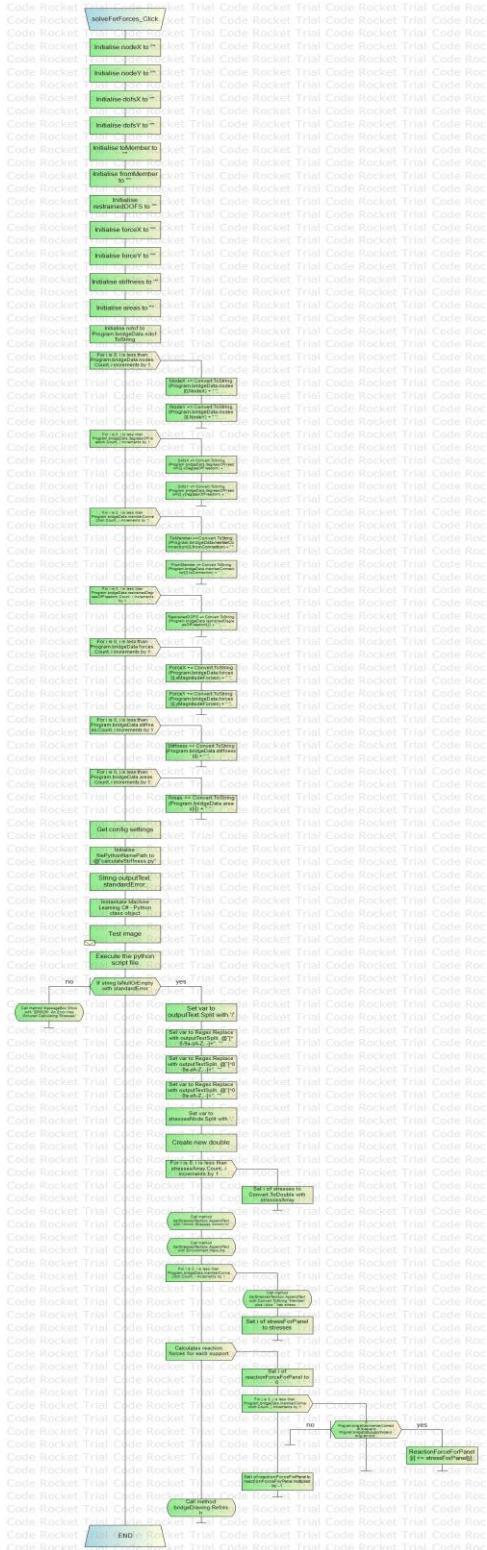
```
Initialise nodeX to ""
Initialise nodeY to ""
Initialise dofsX to ""
Initialise dofsY to ""
Initialise toMember to ""
Initialise fromMember to ""
Initialise restrainedDOFS to ""
Initialise forceX to ""
Initialise forceY to ""
Initialise stiffness to ""
Initialise areas to ""
Initialise ndof to Program.bridgeData.ndof.ToString
For i is 0, i is less than Program.bridgeData.nodes.Count, i increments by 1
    NodeX += Convert.ToString(Program.bridgeData.nodes[i].NodeX) + " ";
    NodeY += Convert.ToString(Program.bridgeData.nodes[i].NodeY) + " ";
EndFor
For i is 0, i is less than Program.bridgeData.degreesOfFreedom.Count, i increments by 1
    DofsX +=
Convert.ToString(Program.bridgeData.degreesOfFreedom[i].xDegreeOfFreedom) + " ";
    DofsY +=
Convert.ToString(Program.bridgeData.degreesOfFreedom[i].yDegreesOfFreedom) + " ";
EndFor
For i is 0, i is less than Program.bridgeData.memberConnection.Count, i increments by 1
    ToMember +=
Convert.ToString(Program.bridgeData.memberConnection[i].fromConnection) + " ";
    FromMember +=
Convert.ToString(Program.bridgeData.memberConnection[i].toConnection) + " ";
EndFor
For i is 0, i is less than Program.bridgeData.restrainedDegreesOfFreedom.Count, i increments by 1
    RestrainedDOFS +=
Convert.ToString(Program.bridgeData.restrainedDegreesOfFreedom[i]) + " ";
EndFor
For i is 0, i is less than Program.bridgeData.forces.Count, i increments by 1
    ForceX += Convert.ToString(Program.bridgeData.forces[i].xMagnitudeForces) + "
";
    ForceY += Convert.ToString(Program.bridgeData.forces[i].yMagnitudeForces) + "
";
EndFor
For i is 0, i is less than Program.bridgeData.stiffness.Count, i increments by 1
    Stiffness += Convert.ToString(Program.bridgeData.stiffness[i]) + " ";
EndFor
For i is 0, i is less than Program.bridgeData.areas.Count, i increments by 1
    Areas += Convert.ToString(Program.bridgeData.areas[i]) + " ";
EndFor
Get config settings
Initialise filePythonNamePath to @"calculateStiffness.py"
String outputText, standardError;
Instantiate Machine Learning C# - Python class object
Test image
Execute the python script file
If string.IsNullOrEmpty with standardError
    Set var to outputText.Split with '/'
    Set var to Regex.Replace with outputTextSplit, @"[^ 0-9a-zA-Z,.]+", ""
```

```

Set var to Regex.Replace with outputTextSplit, @"[^0-9a-zA-Z,.]+", ""
Set var to Regex.Replace with outputTextSplit, @"[^0-9a-zA-Z,.]+", ""
Set var to stressesNode.Split with ','
Create new double
For i is 0, i is less than stressesArray.Count, i increments by 1
    Set i of stresses to Convert.ToDouble with stressesArray
EndFor
Call method barStressesTextbox.AppendText with "===== Stresses =====\r\n"
Call method barStressesTextbox.AppendText with Environment.NewLine
For i is 0, i is less than Program.bridgeData.memberConnection.Count, i
increments by 1
    Call method barStressesTextbox.AppendText with Convert.ToString "Member"
plus i plus " has stresses of" plus stresses, +" units" plus "\r\n"
    Set i of stressForPanel to stresses
EndFor
Calculates reaction forces for each support
Set i of reactionForceForPanel to 0
For j is 0, j is less than Program.bridgeData.memberConnection.Count, j
increments by 1
    If Program.bridgeData.memberConnection is equal to
Program.bridgeData.supportNode or Program.bridgeData.memberConnection is equal to
Program.bridgeData.supportNode
        ReactionForceForPanel[i] += stressForPanel[j];
    EndIf
EndFor
Set i of reactionForceForPanel to reactionForceForPanel multiplied by -1
EndFor
Call method bridgeDrawing.Refresh
Else
    Call method MessageBox.Show with "ERROR: An Error Has Occured Calculating
Stresses"
EndIf

```

### Flowchart



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

**Try**

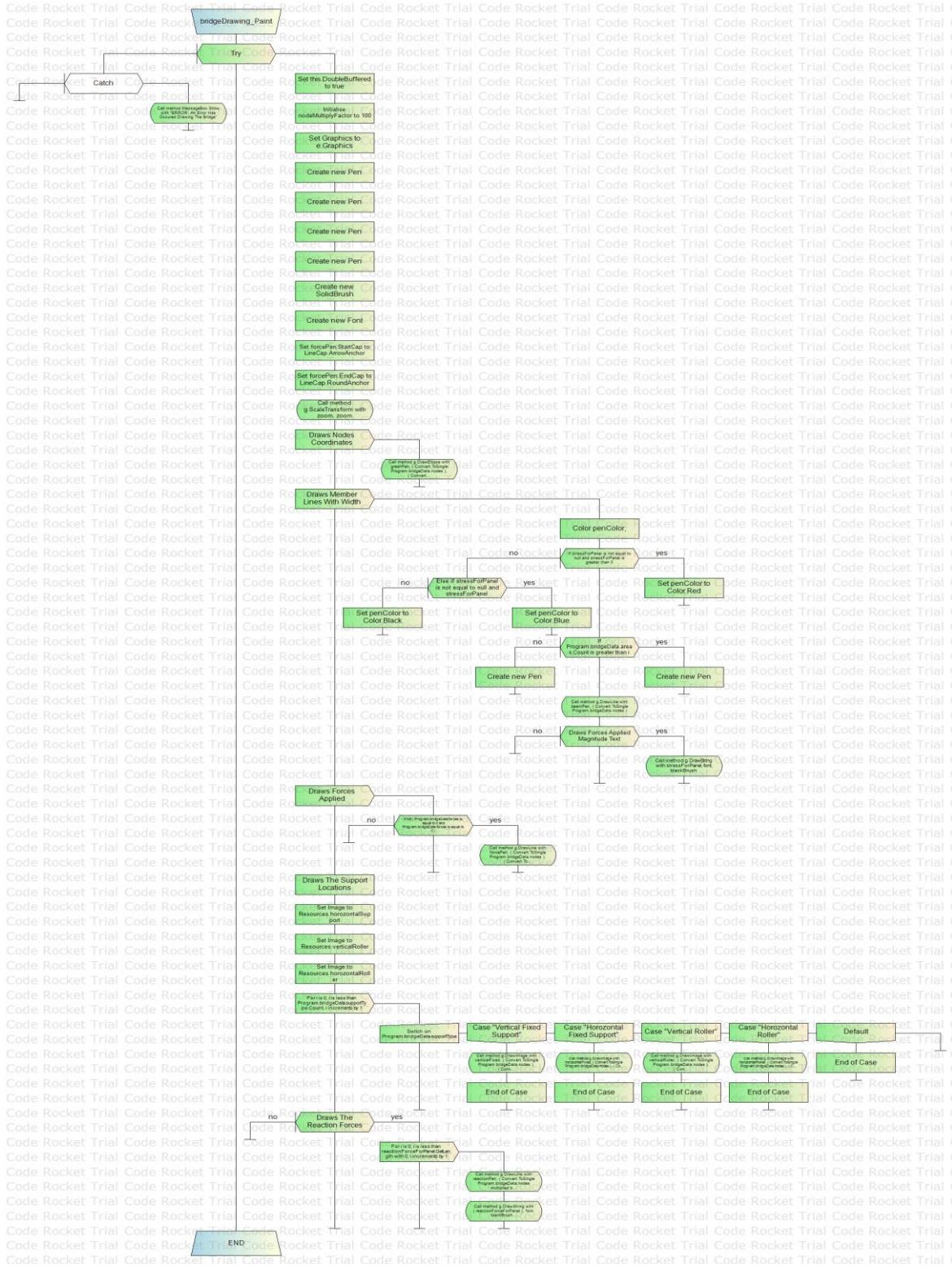
```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Set forcePen.StartCap to LineCap.ArrowAnchor
    Set forcePen.EndCap to LineCap.RoundAnchor
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble
Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Color penColor;
        If stressForPanel is not equal to null and stressForPanel is greater
than 0
            Set penColor to Color.Red
        Else if stressForPanel is not equal to null and stressForPanel
            Set penColor to Color.Blue
        Else
            Set penColor to Color.Black
        EndIf
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble
Program.bridgeData.nodes )
        Draws Forces Applied Magnitude Text
            Call method g.DrawString with stressForPanel, font, blackBrush
        EndIf
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble
Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble
Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
```

```

    Case "Horizontal Fixed Support"
        Call method g.DrawImage with horizontalFixed, ( Convert.ToDouble
Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Case "Vertical Roller"
        Call method g.DrawImage with verticalRoller, ( Convert.ToDouble
Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Case "Horizontal Roller"
        Call method g.DrawImage with horizontalRoller, ( Convert.ToDouble
Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
        End of Case
EndSwitch
EndFor
Draws The Reaction Forces
For i is 0, i is less than reactionForceForPanel.GetLength with 0, i
increments by 1
    Call method g.DrawLine with reactionPen, ( Convert.ToDouble
Program.bridgeData.nodes multiplied by nodeMultiplyFactor plus Convert.ToDouble )
        Call method g.DrawString with ( reactionForceForPanel ), font,
blackBrush...
    EndFor
EndIf
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry

```

### Flowchart



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

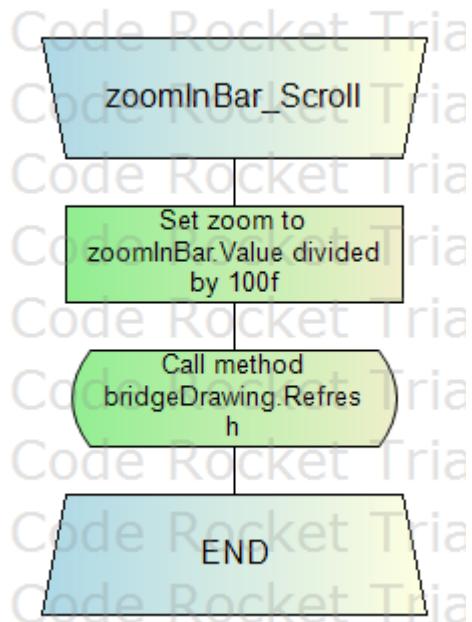
Returns Void

**Pseudocode**

```
Set zoom to zoomInBar.Value divided by 100f
```

```
Call method bridgeDrawing.Refresh
```

**Flowchart**



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

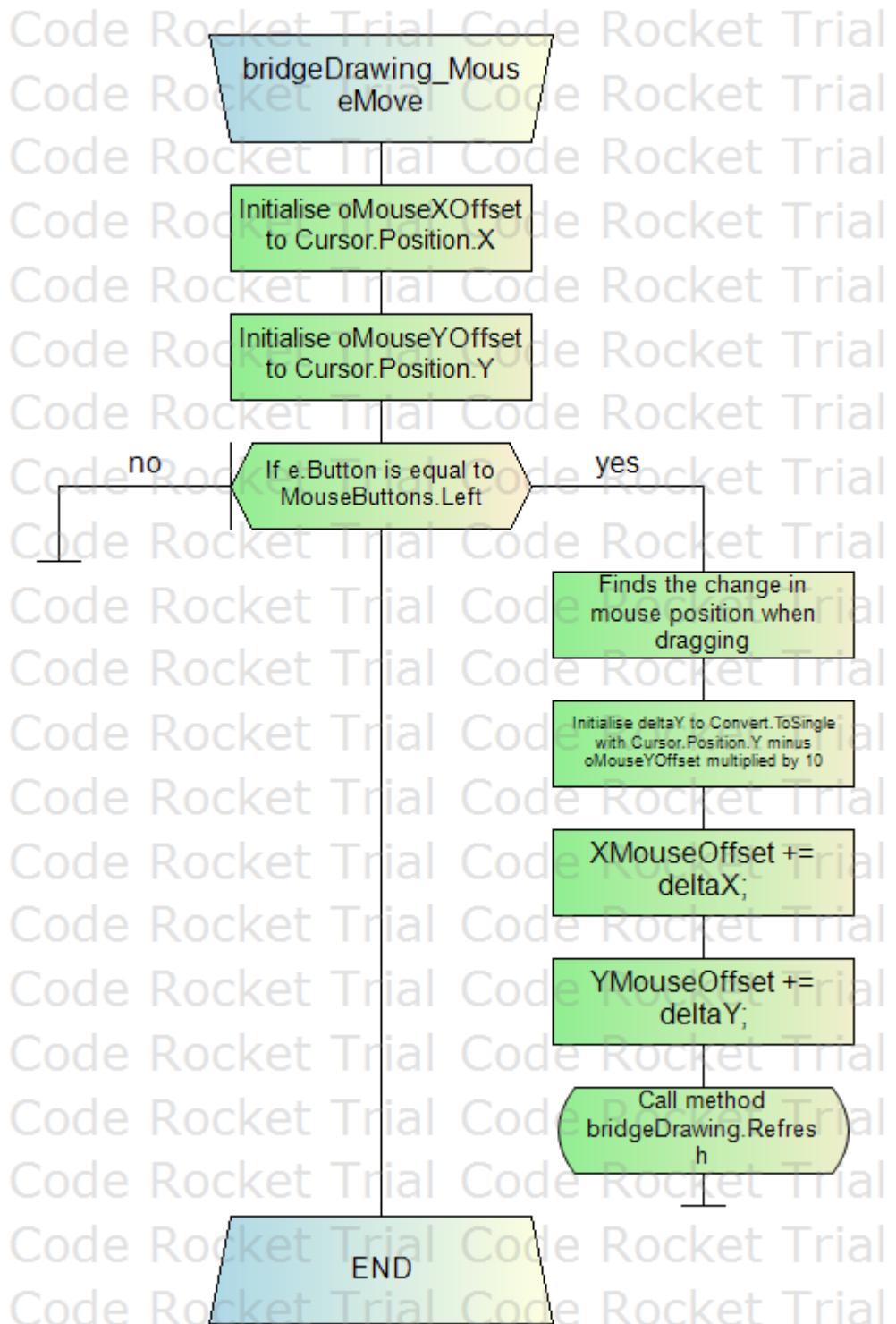
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

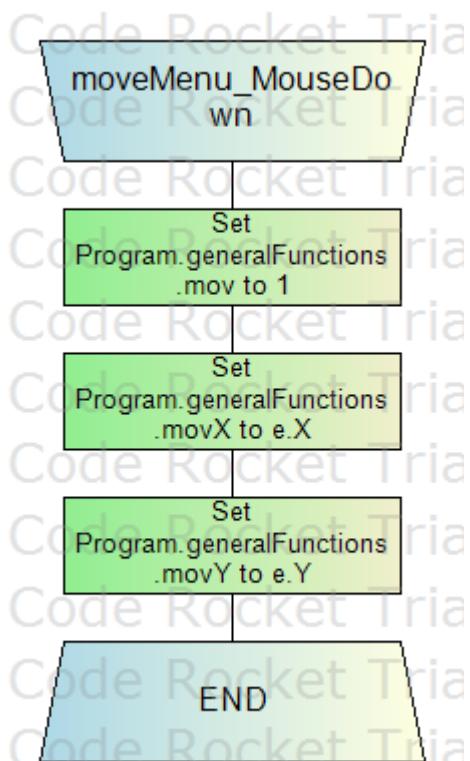
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

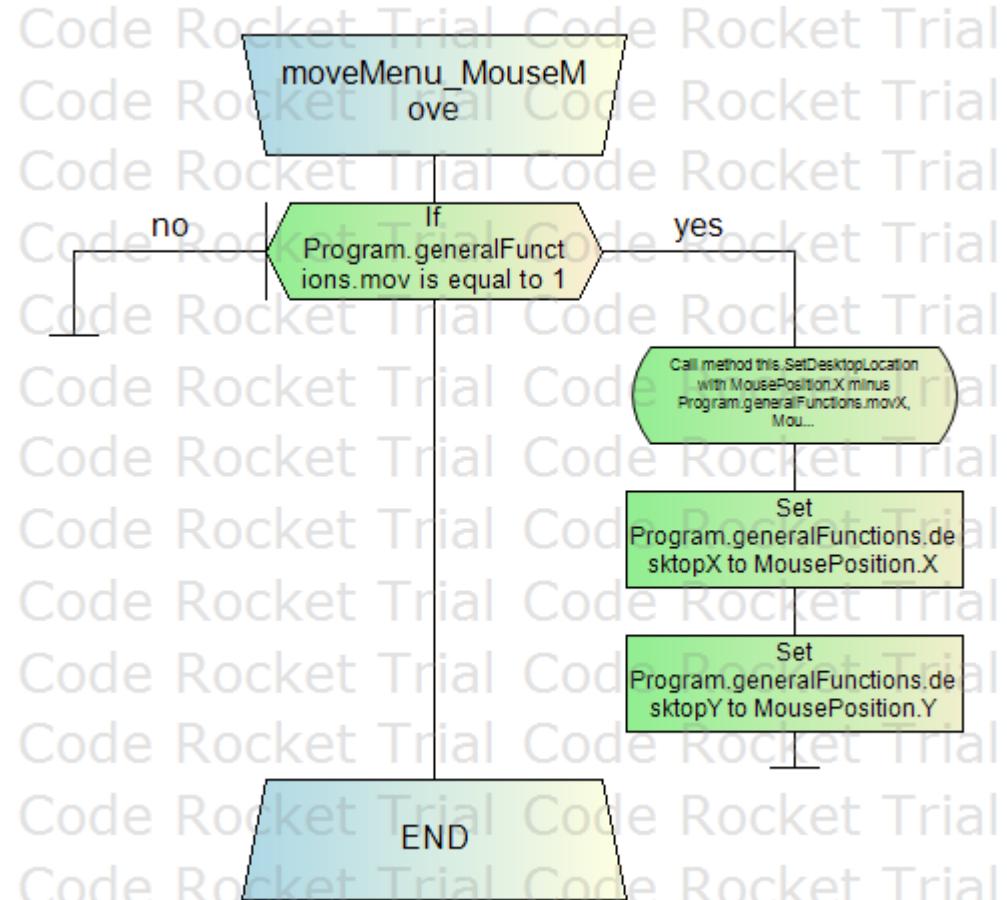
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



*Procedure: moveMenu\_MouseUp*

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

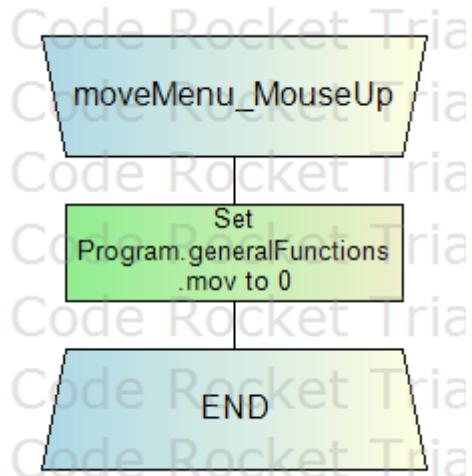
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

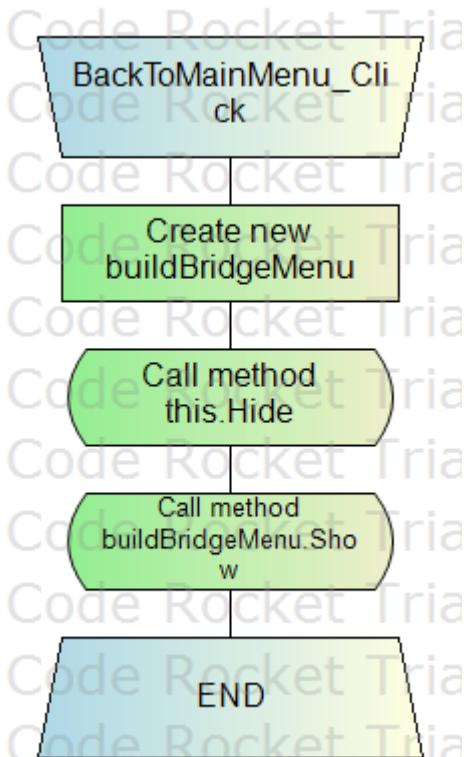
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



**Interface: IMLSharpPython**

```
public interface IMLSharpPython
```

**Procedure: ExecutePythonScript**

```
string ExecutePythonScript(string filePythonScript,
```

**Parameters**

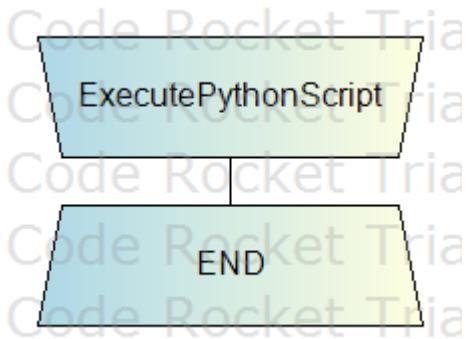
- filePythonScript - String

Returns String

**Pseudocode**

N/A

**Flowchart**



```
out string standardError);
```

**Class: MLSharpPython**

```
public class MLSharpPython : IMLSharpPython
```

**Attribute: filePythonExePath**

```
public readonly string filePythonExePath;
```

**Procedure: Constructor**

```
public MLSharpPython(string exePythonPath)
```

**Parameters**

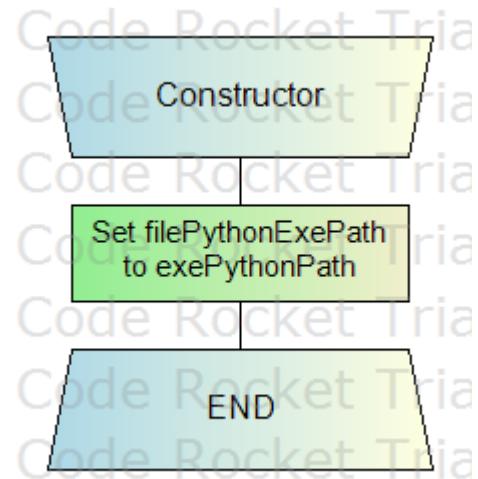
- exePythonPath - String

Returns MLSharpPython

**Pseudocode**

```
Set filePythonExePath to exePythonPath
```

**Flowchart**



**Procedure: ExecutePythonScript**

```
public string ExecutePythonScript(string filePythonScript, out string  
standardError)
```

**Parameters**

- filePythonScript - String

Returns String

**Pseudocode**

```
Initialise outputText to string.Empty  
Set standardError to string.Empty
```

**Try**

```
    using (Process process = new Process())  
        Create new ProcessStartInfo  
        Call method process.Start  
        Set outputText to process.StandardOutput.ReadToEnd  
        Set outputText to outputText.Replace with Environment.NewLine,  
string.Empty  
        Set standardError to process.StandardError.ReadToEnd  
        Call method process.WaitForExit
```

**EndStatement**

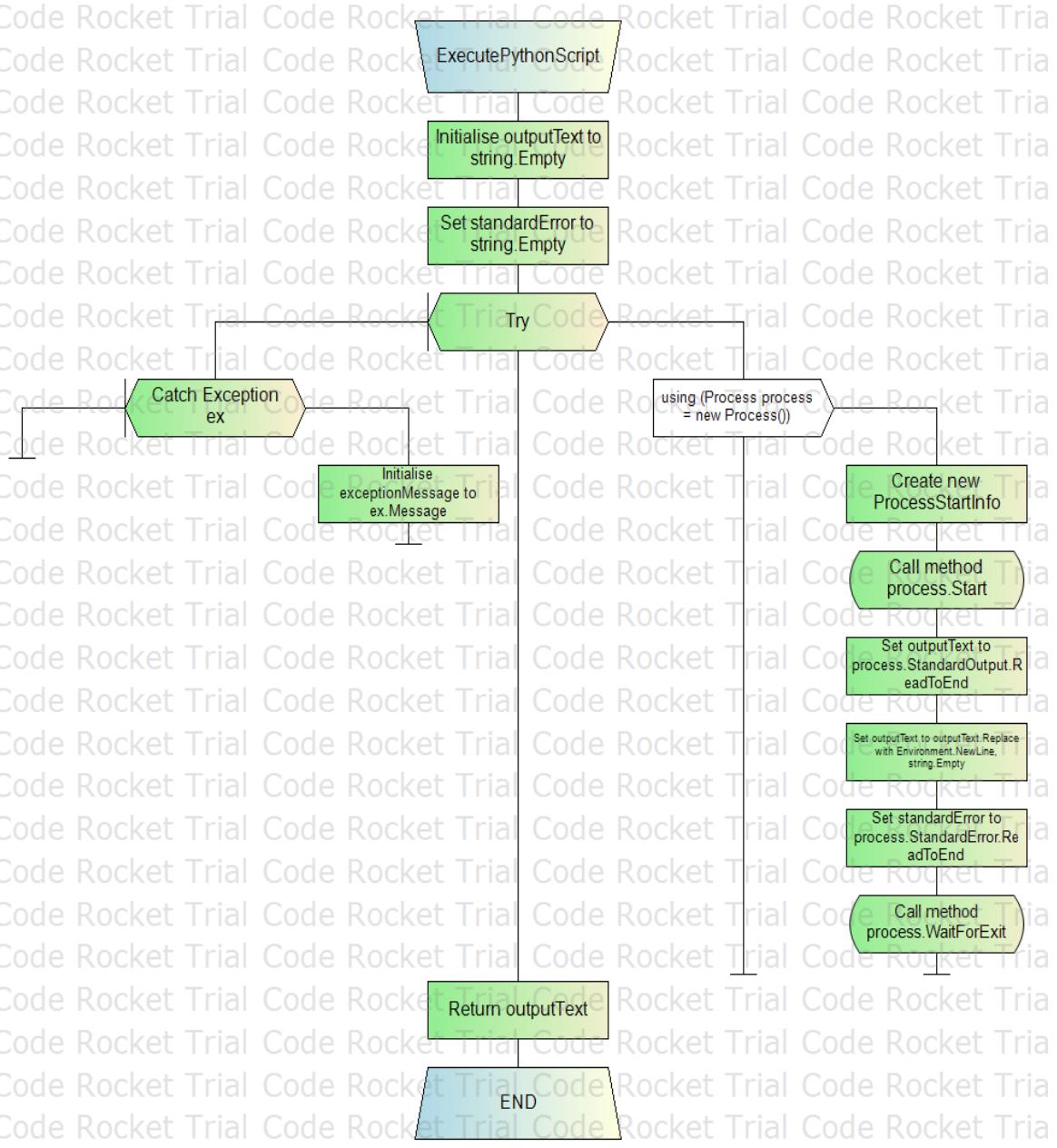
**Catch** Exception ex

```
    Initialise exceptionMessage to ex.Message
```

**EndTry**

```
Return outputText
```

Flowchart



*Code File: solveMenus.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

**Class: solveMenus**

partial class solveMenus

**Attribute: components**

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure: Dispose**

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

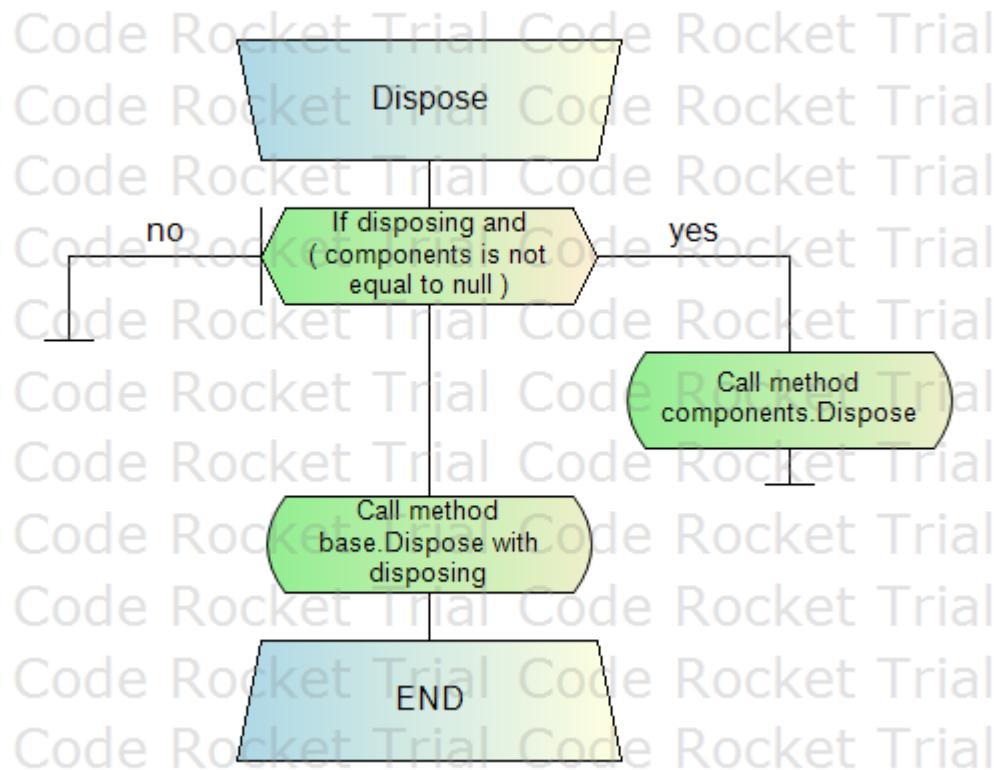
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

**Flowchart**



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.Button
with ( System.ComponentModel.ISupportInitializeInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackColor to System.Drawing.Color.Transparent
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.TabStop to false
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackColor to System.Drawing.Color.Transparent
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.TabStop to false
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
barStressesTextbox
Create new System.Drawing.Point
Set this.barStressesTextbox.Name to "barStressesTextbox"
Set this.barStressesTextbox.ReadOnly to true
Create new System.Drawing.Size
Set this.barStressesTextbox.TabIndex to 12
Set this.barStressesTextbox.TabStop to false
This.barStressesTextbox.TextChanged += new
System.EventHandler(this.barStressesTextbox_TextChanged);
solveForForces
Set this.solveForForces.BackColor to System.Drawing.Color.Transparent
Set this.solveForForces.FlatAppearance.BorderColor to System.Drawing.Color.White

```

```

Set this.solveForForces.FlatAppearance.BorderSize to 2
Set this.solveForForces.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.solveForForces.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.solveForForces.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.solveForForces.Name to "solveForForces"
Create new System.Drawing.Size
Set this.solveForForces.TabIndex to 13
Set this.solveForForces.TabStop to false
Set this.solveForForces.Text to "SOLVE FOR STRESS"
Set this.solveForForces.UseVisualStyleBackColor to false
This.solveForForces.Click += new System.EventHandler(this.solveForForces_Click);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 14
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 15
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 38
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent
Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 39
Set this.BackToMainMenu.TabStop to false
Set this.BackToMainMenu.Text to "BACK"
Set this.BackToMainMenu.UseVisualStyleBackColor to false

```

```

This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
solveMenus
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.BackToMainMenu
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.solveForForces
Call method this.Controls.Add with this.barStressesTextbox
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Call method this.Controls.Add with this.bridgeDrawing
Set this.DoubleBuffered to true
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "solveMenus"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitializeInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout

```

### Flowchart



```
#endregion  
Attribute: minimize  
  
private System.Windows.Forms.Button minimize;  
Attribute: exitApplication  
  
private System.Windows.Forms.Button exitApplication;  
Attribute: barStressesTextbox  
  
private System.Windows.Forms.TextBox barStressesTextbox;  
Attribute: solveForForces  
  
private System.Windows.Forms.Button solveForForces;  
Attribute: zoomInBar  
  
private System.Windows.Forms.TrackBar zoomInBar;  
Attribute: bridgeDrawing  
  
private System.Windows.Forms.Panel bridgeDrawing;  
Attribute: moveMenu  
  
private System.Windows.Forms.Panel moveMenu;  
Attribute: BackToMainMenu  
  
private System.Windows.Forms.Button BackToMainMenu;
```

**Code File:** *splashScreen.cs*

```
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:** *Sectrics\_V2*

```
namespace Sectrics_V2  
Class: splashScreen
```

```
public partial class splashScreen : Form
```

**Procedure: Constructor**

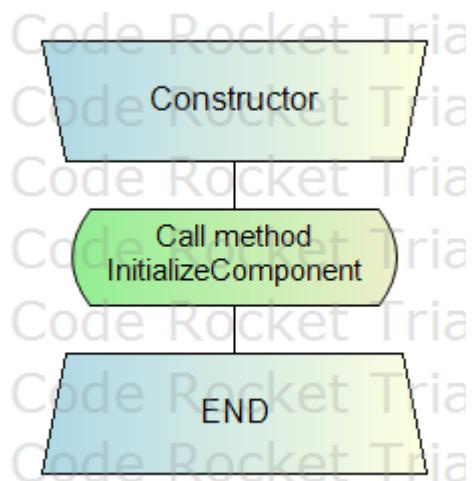
```
public splashScreen()
```

Returns *splashScreen*

**Pseudocode**

Call method InitializeComponent

Flowchart



**Procedure: splashScreen\_Load**

```
private void splashScreen_Load(object sender, EventArgs e)
```

**Parameters**

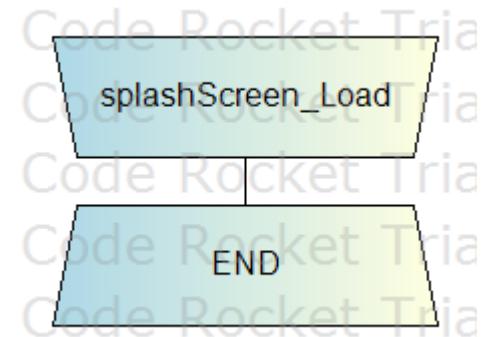
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Code File: splashScreen.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

*Class: splashScreen*

partial class splashScreen

*Attribute: components*

private System.ComponentModel.IContainer components = null;

Required designer variable.

*Procedure: Dispose*

protected override void Dispose (bool disposing)

Clean up any resources being used.

*Parameters*

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

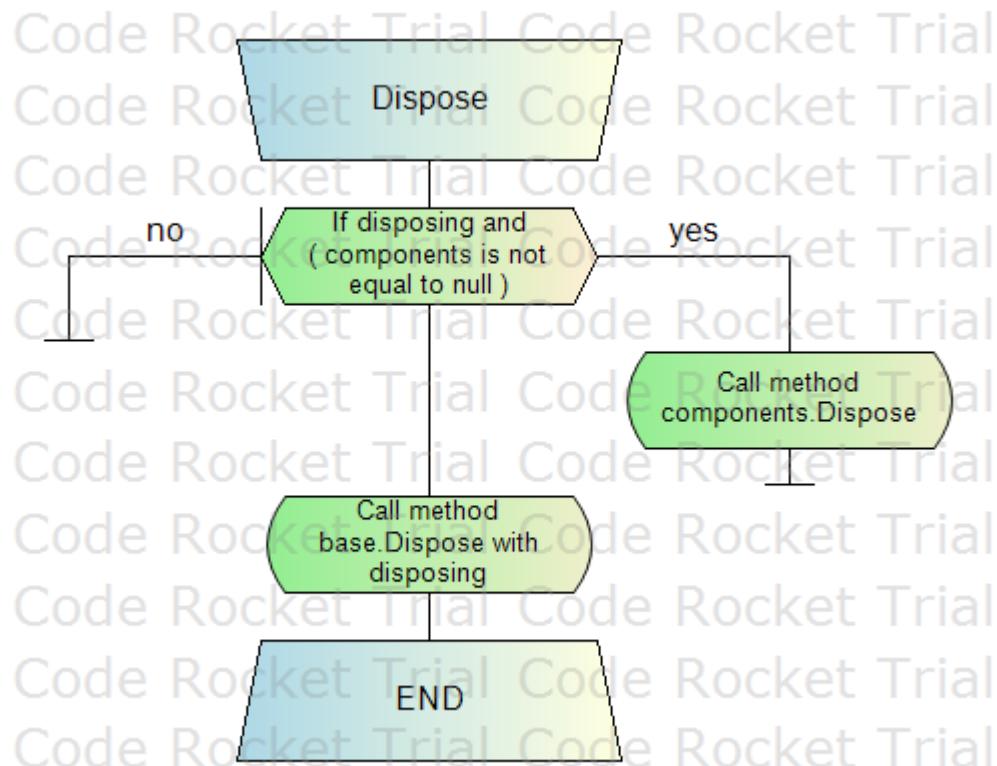
*Pseudocode*

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

*Flowchart*

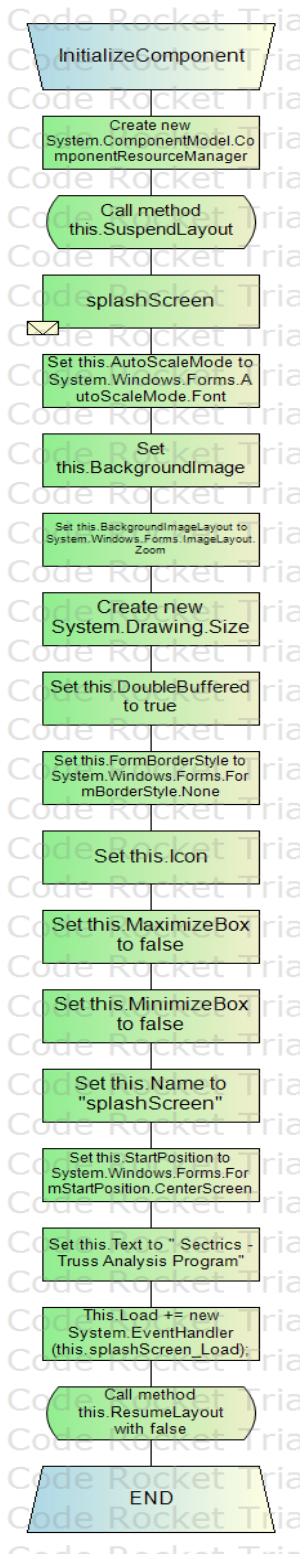


```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Call method this.SuspendLayout
splashScreen
Set this.AutoScaleMode to System.Windows.Forms.AutoScaleMode.Font
Set this.BackgroundImage
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.Zoom
Create new System.Drawing.Size
Set this.DoubleBuffered to true
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.MaximizeBox to false
Set this.MinimizeBox to false
Set this.Name to "splashScreen"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to " Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.splashScreen_Load);
Call method this.ResumeLayout with false

```

### Flowchart



```
#endregion
```

**Code File: startMenu.cs**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;
```

**Namespace: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: startMenu**

```
public partial class startMenu : Form
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

**Procedure: SplashScreen**

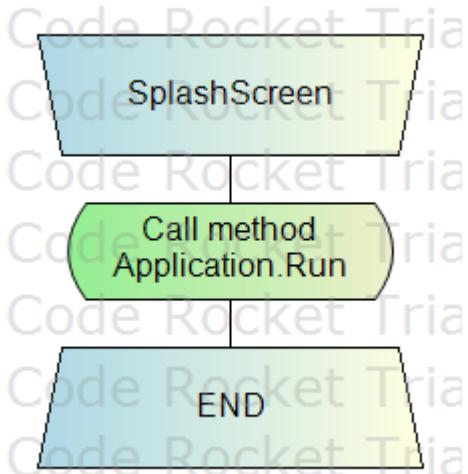
```
public void SplashScreen()
```

Returns Void

**Pseudocode**

```
Call method Application.Run
```

Flowchart



*Procedure: Constructor*

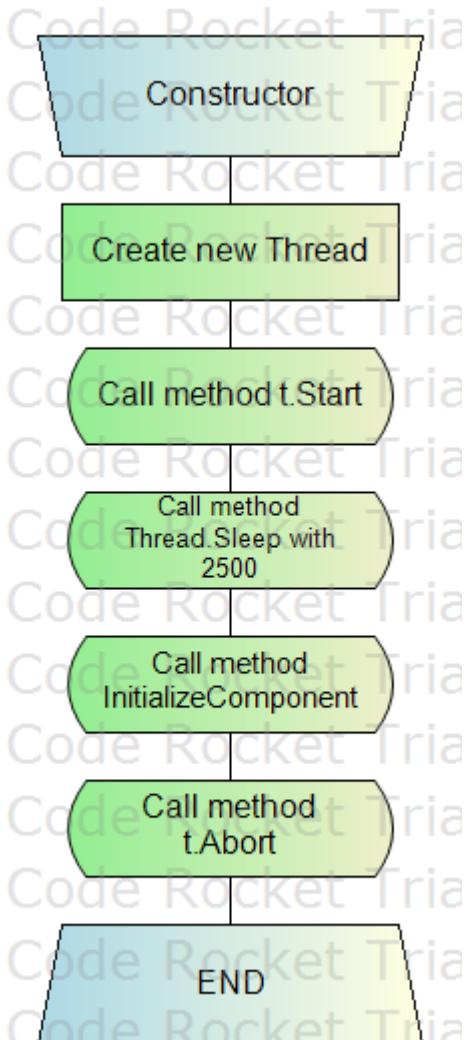
```
public startMenu()
```

Returns startMenu

**Pseudocode**

```
Create new Thread  
Call method t.Start  
Call method Thread.Sleep with 2500  
Call method InitializeComponent  
Call method t.Abort
```

Flowchart



**Procedure: WndProc**

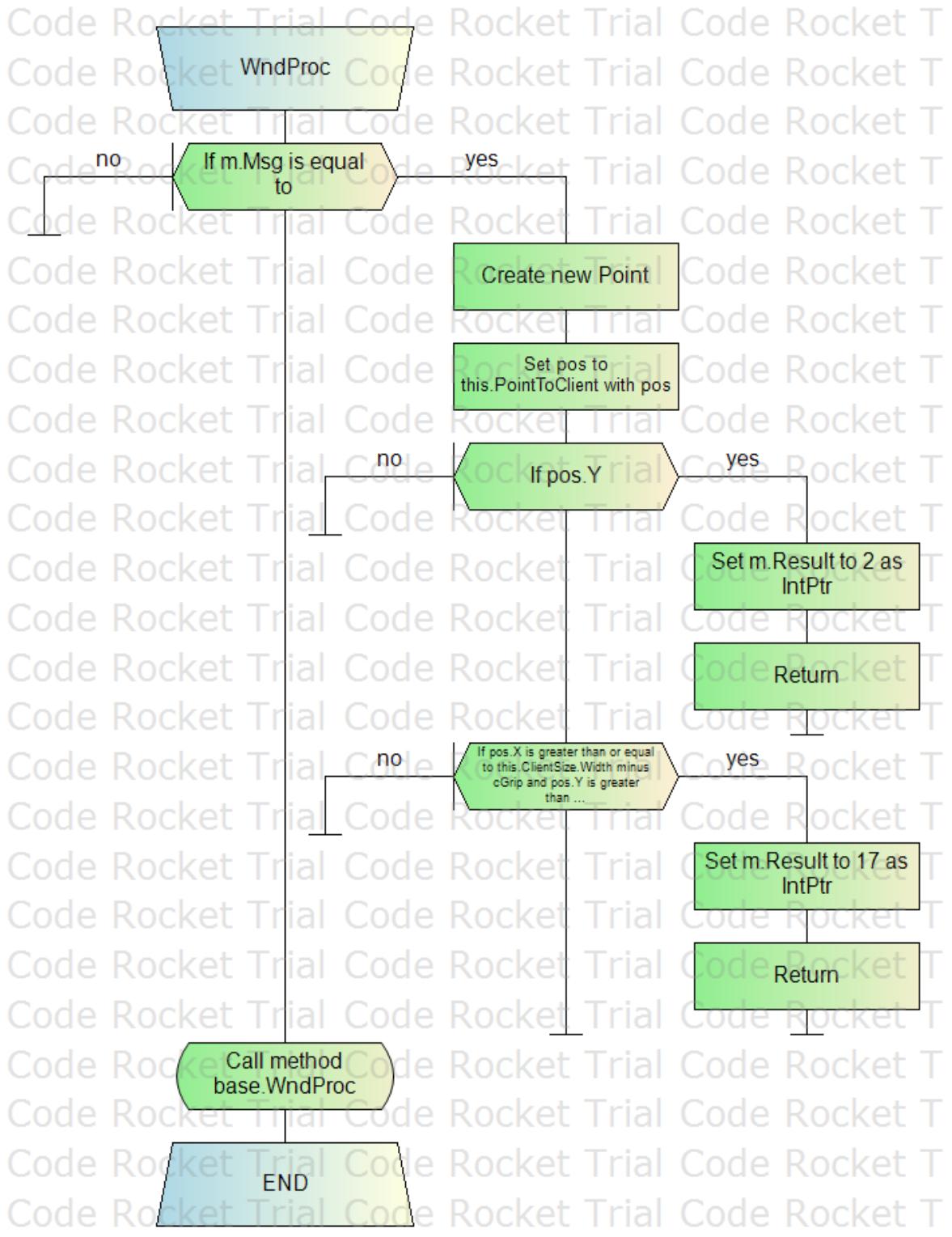
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

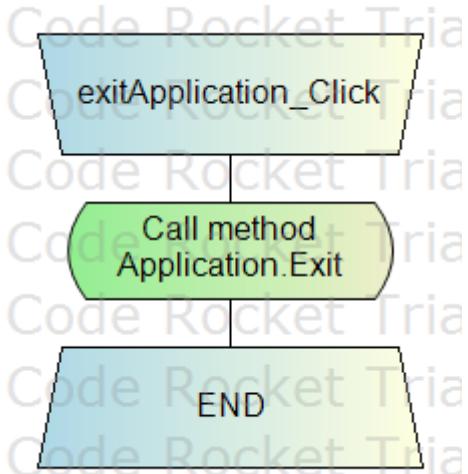
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

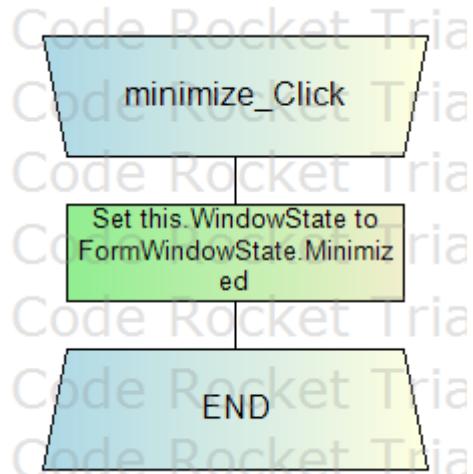
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



*Procedure: startApplication\_Click*

```
private void startApplication_Click(object sender, EventArgs e)
```

**Parameters**

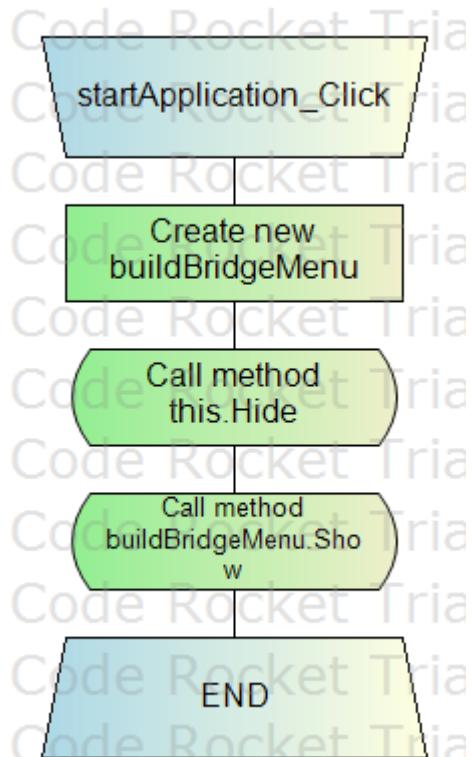
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

Flowchart



**Procedure: helpButton\_Click**

```
private void helpButton_Click(object sender, EventArgs e)
```

**Parameters**

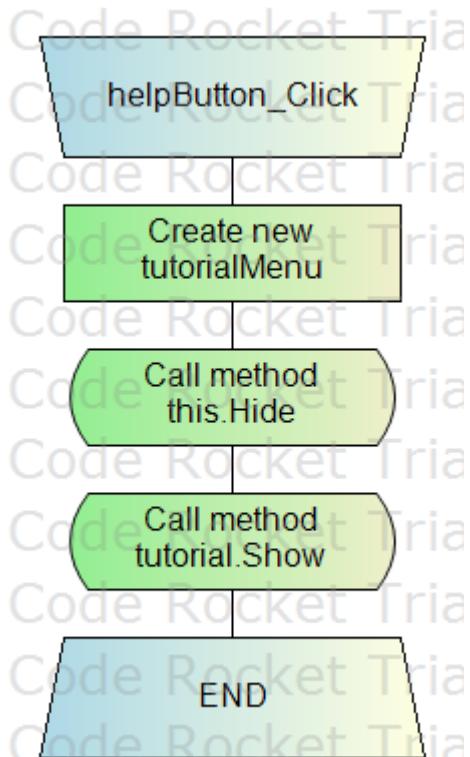
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new tutorialMenu  
Call method this.Hide  
Call method tutorial.Show
```

**Flowchart**



**Procedure: startMenu\_Load**

```
private void startMenu_Load(object sender, EventArgs e)
```

**Parameters**

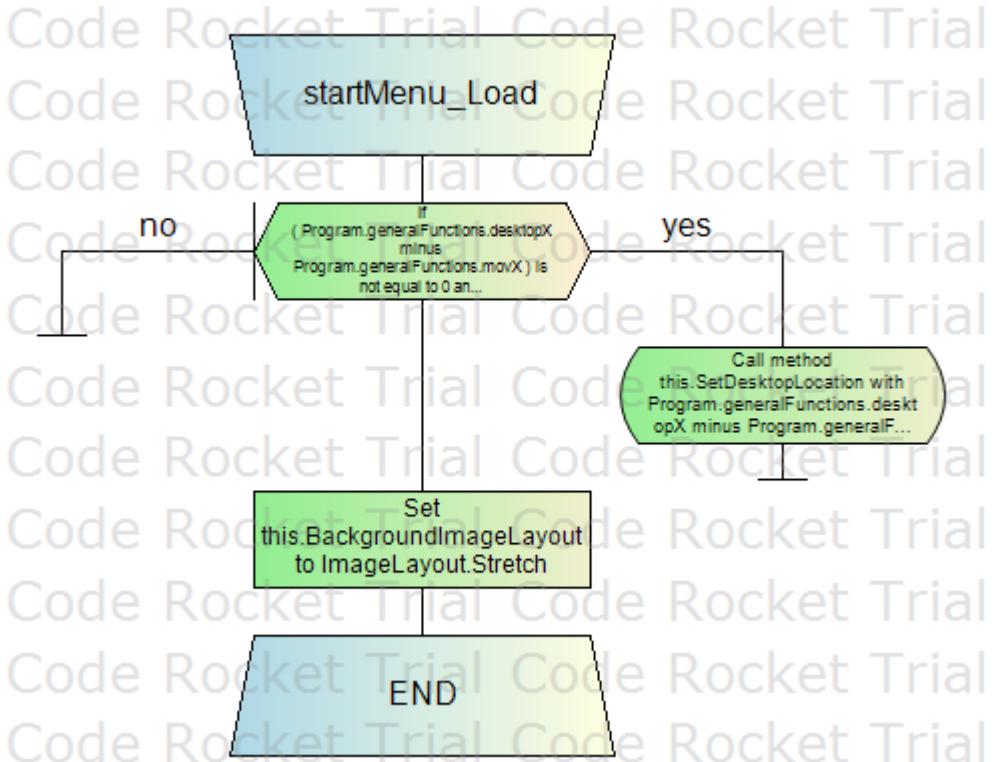
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

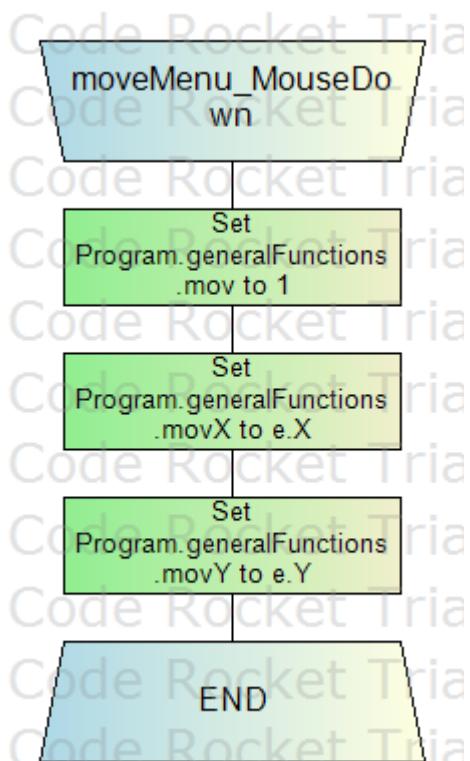
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

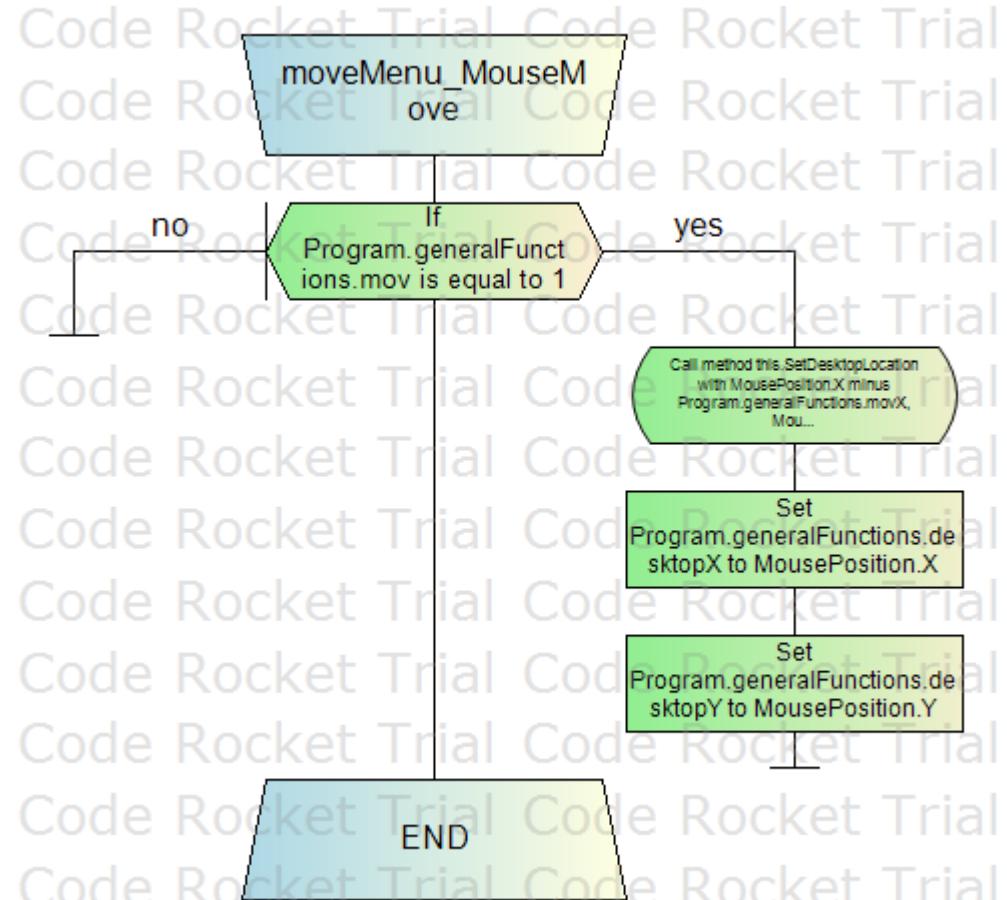
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseUp**

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

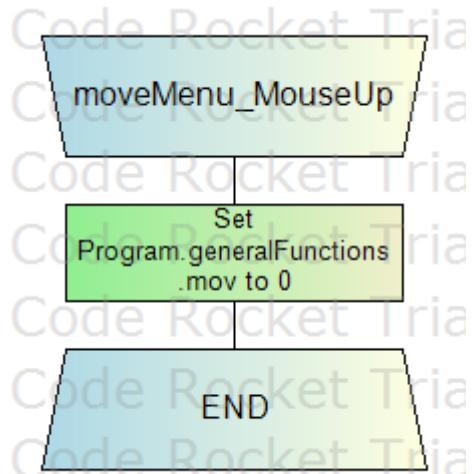
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Code File:** startMenu.Designer.cs

**Namespace:** Sectrics\_V2

namespace Sectrics\_V2

**Class:** startMenu

partial class startMenu

**Attribute:** components

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure:** Dispose

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

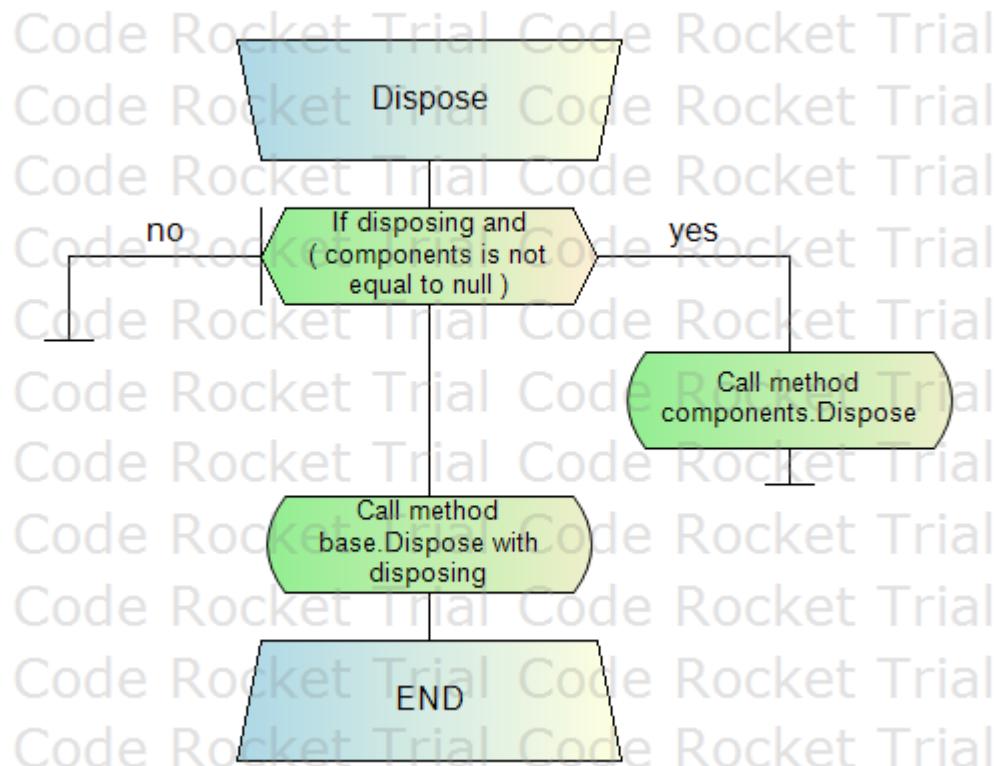
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

Flowchart



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.PictureBox
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
exitApplication
Set this.exitApplication.BackColor to System.Drawing.Color.Transparent
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 0
Set this.exitApplication.TabStop to false
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
minimize
Set this.minimize.BackColor to System.Drawing.Color.Transparent
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 1
Set this.minimize.TabStop to false
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
startApplication
Set this.startApplication.BackColor to System.Drawing.Color.Transparent
Set this.startApplication.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.startApplication.FlatAppearance.BorderSize to 2
Set this.startApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( int ), ( int ) )
Set this.startApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.startApplication.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.startApplication.Name to "startApplication"
Create new System.Drawing.Size
Set this.startApplication.TabIndex to 0

```

```

Set this.startApplication.TabStop to false
Set this.startApplication.Text to "START"
Set this.startApplication.UseVisualStyleBackColor to false
This.startApplication.Click += new System.EventHandler(this.startApplication_Click);
label1
Set this.label1.AutoSize to true
Set this.label1.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 3
Set this.label1.Text to "STUDENT EDITION"
helpButton
Set this.helpButton.BackColor to System.Drawing.Color.Transparent
Set this.helpButton.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.helpButton.FlatAppearance.BorderSize to 2
Set this.helpButton.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.helpButton.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.helpButton.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.helpButton.Name to "helpButton"
Create new System.Drawing.Size
Set this.helpButton.TabIndex to 1
Set this.helpButton.TabStop to false
Set this.helpButton.Text to "HELP"
Set this.helpButton.UseVisualStyleBackColor to false
This.helpButton.Click += new System.EventHandler(this.helpButton_Click);
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 42
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
pictureBox1
Set this.pictureBox1.ImageLayout to System.Windows.Forms.ImageLayout.Zoom
Create new System.Drawing.Point
Set this.pictureBox1.Name to "pictureBox1"
Create new System.Drawing.Size
Set this.pictureBox1.TabIndex to 43
Set this.pictureBox1.TabStop to false
startMenu
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.helpButton
Call method this.Controls.Add with this.label1
Call method this.Controls.Add with this.startApplication
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Call method this.Controls.Add with this.pictureBox1
Set this.DoubleBuffered to true
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None

```

```
Set this.Icon
Set this.MaximizeBox to false
Set this.MinimizeBox to false
Set this.Name to "startMenu"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.startMenu_Load);
with ( System.ComponentModel.ISupportInitialize.ISupportInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout
```

### Flowchart



```
#endregion  
Attribute: exitApplication  
  
private System.Windows.Forms.Button exitApplication;  
Attribute: minimize  
  
private System.Windows.Forms.Button minimize;  
Attribute: startApplication  
  
private System.Windows.Forms.Button startApplication;  
Attribute: label1  
  
private System.Windows.Forms.Label label1;  
Attribute: helpButton  
  
private System.Windows.Forms.Button helpButton;  
Attribute: moveMenu  
  
private System.Windows.Forms.Panel moveMenu;  
Attribute: pictureBox1  
  
private System.Windows.Forms.PictureBox pictureBox1;
```

**Code File: Program.cs**

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using System.Runtime.InteropServices;
```

**Namespace: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: Program**

```
static class Program
```

**Attribute: maths**

```
public static Maths maths = new Maths();
```

**Attribute: bridgeData**

```
public static BridgeData bridgeData = new BridgeData();
```

**Attribute: generalFunctions**

```
public static generalFunctions generalFunctions = new  
generalFunctions();
```

**Procedure: SetProcessDpiAwareness**

```
[DllImport("Shcore.dll")]
    static extern int SetProcessDpiAwareness(int
PROCESS_DPI_AWARENESS);
```

**Parameters**

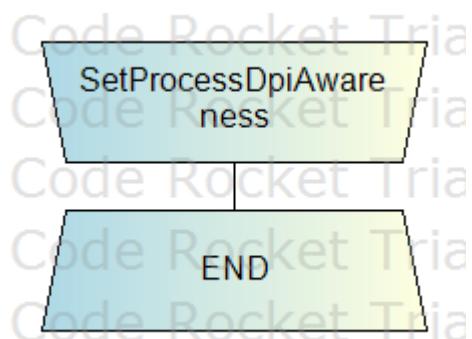
- PROCESS\_DPI\_AWARENESS - Integer

Returns Integer

**Pseudocode**

N/A

**Flowchart**



**Procedure: Main**

[STAThread]

```
    static void Main()
```

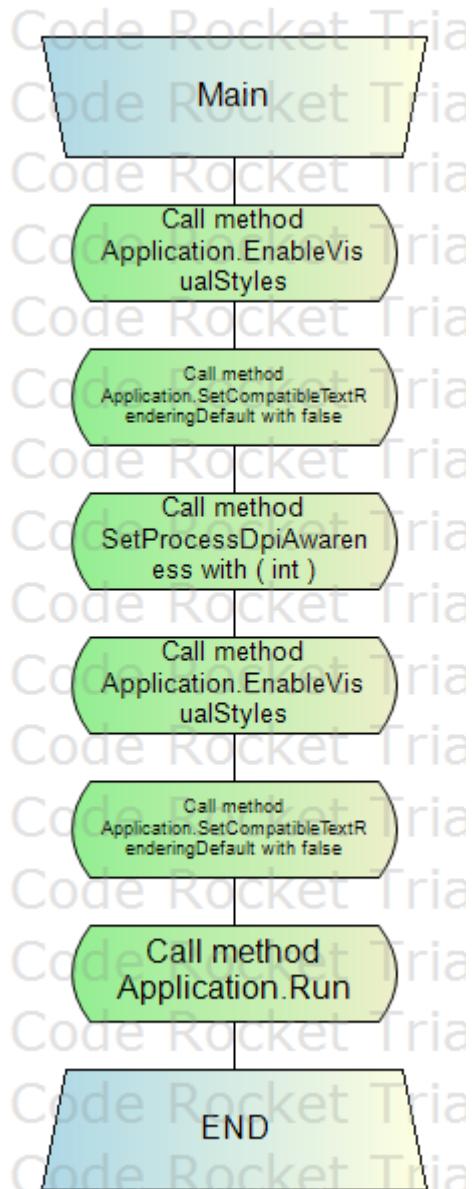
The main entry point for the application.

Returns Void

**Pseudocode**

```
Call method Application.EnableVisualStyles  
Call method Application.SetCompatibleTextRenderingDefault with false  
Call method SetProcessDpiAwareness with ( int )  
Call method Application.EnableVisualStyles  
Call method Application.SetCompatibleTextRenderingDefault with false  
Call method Application.Run
```

**Flowchart**



**Code File: AssemblyInfo.cs**

```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
[assembly: AssemblyTitle("Sectrics V2")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("Sectrics V2")]
[assembly: AssemblyCopyright("Copyright © 2018")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]
[assembly: ComVisible(false)]
[assembly: Guid("8b9e7887-289b-4d3e-8f74-08b8f5c37d54")]
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
```

**Code File: supportMenus.cs**

```
using Sectrics_V2.Properties;
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: supportMenus**

```
public partial class supportMenus : Form
```

**Attribute: zoom**

```
float zoom = 1f;
```

**Attribute: xMouseOffset**

```
double xMouseOffset;
```

**Attribute: yMouseOffset**

```
double yMouseOffset;
```

*Attribute: cGrip*

```
private const int cGrip = 16;
```

*Attribute: cCaption*

```
private const int cCaption = 32;
```

*Procedure: Constructor*

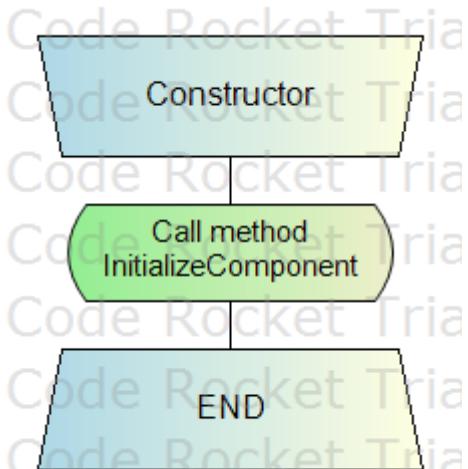
```
public supportMenus()
```

Returns supportMenus

**Pseudocode**

```
Call method InitializeComponent
```

Flowchart



**Procedure: WndProc**

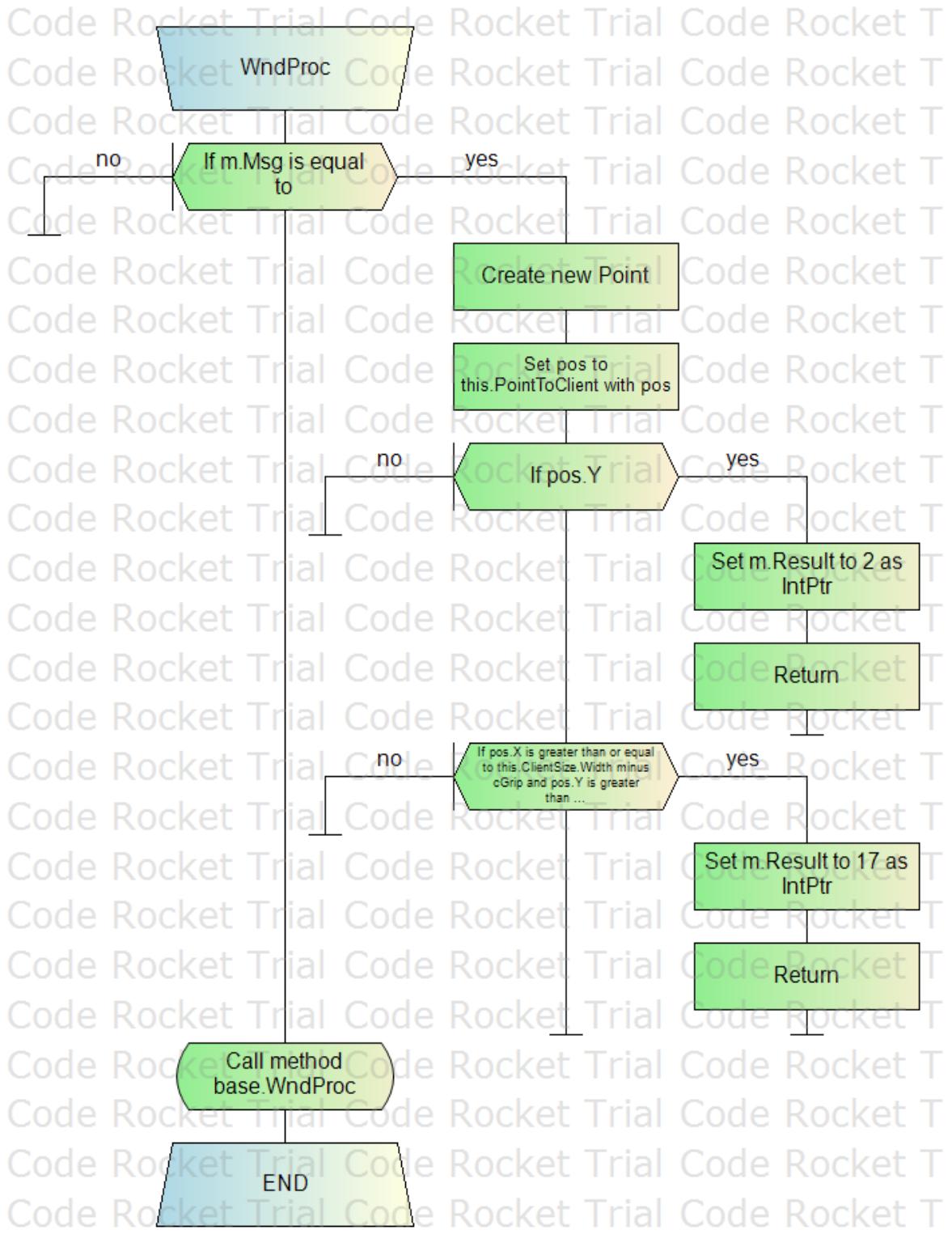
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

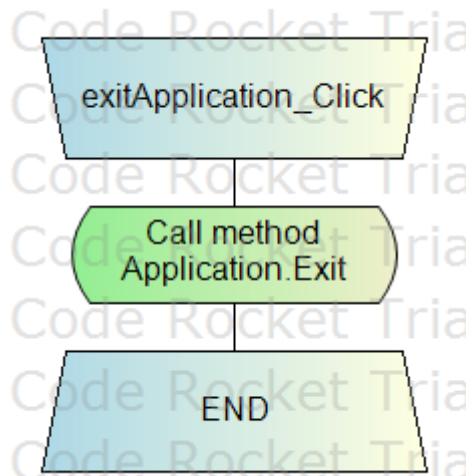
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

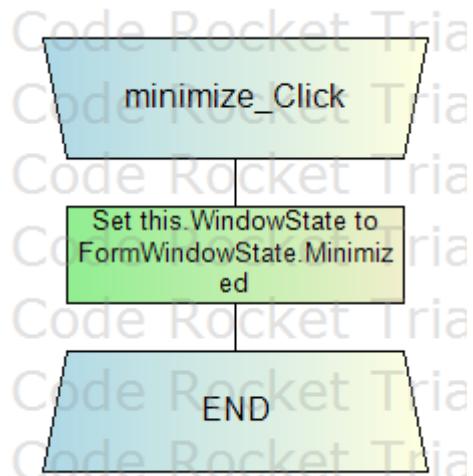
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

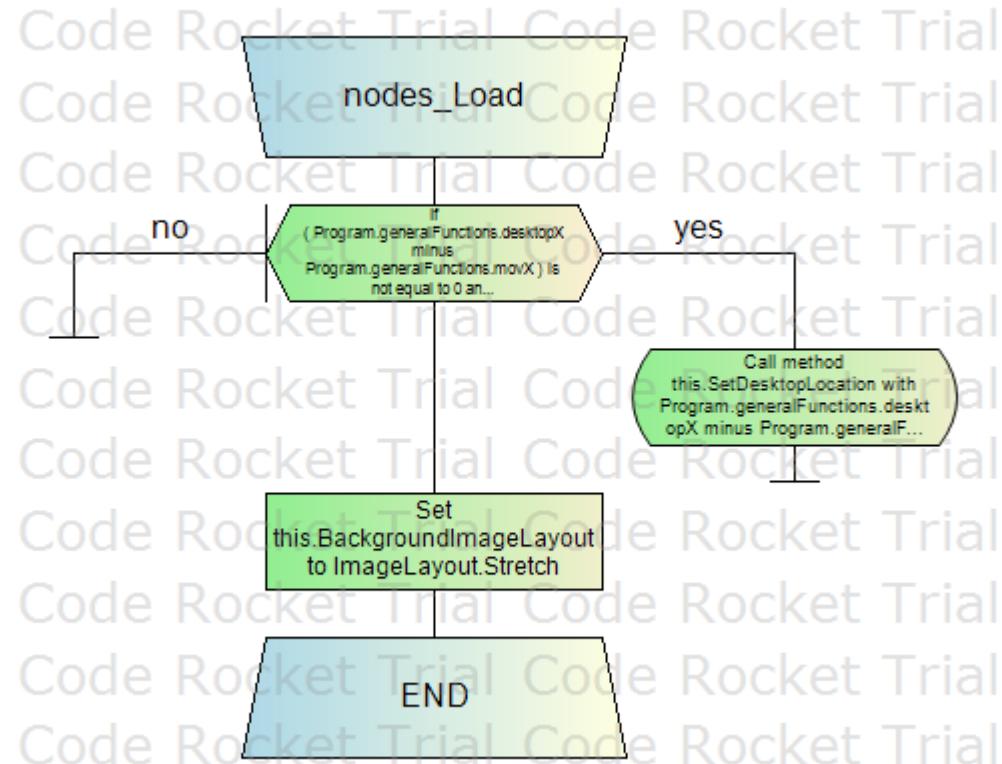
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: addSupportButton\_Click**

```
private void addSupportButton_Click(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

**Try**

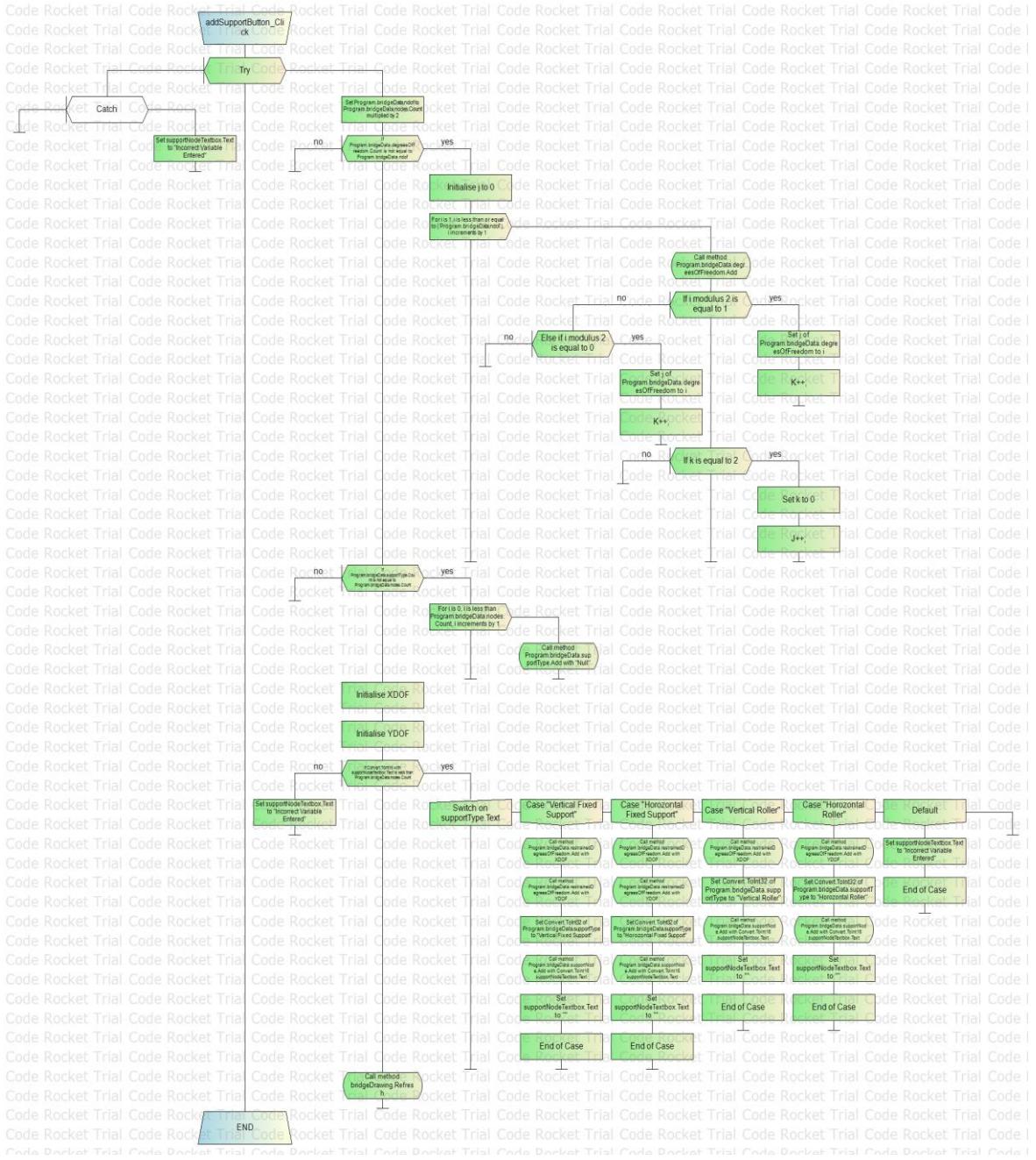
```
    Set Program.bridgeData.ndof to Program.bridgeData.nodes.Count multiplied by 2
    If Program.bridgeData.degreesOfFreedom.Count is not equal to
Program.bridgeData.ndof
        Initialise j to 0
        For i is 1, i is less than or equal to ( Program.bridgeData.ndof ), i
increments by 1
            Call method Program.bridgeData.degreesOfFreedom.Add
            If i modulus 2 is equal to 1
                Set j of Program.bridgeData.degreesOfFreedom to i
                K++;
            Else if i modulus 2 is equal to 0
                Set j of Program.bridgeData.degreesOfFreedom to i
                K++;
            EndIf
            If k is equal to 2
                Set k to 0
                J++;
            EndIf
        EndFor
    EndIf
    If Program.bridgeData.supportType.Count is not equal to
Program.bridgeData.nodes.Count
        For i is 0, i is less than Program.bridgeData.nodes.Count, i increments
by 1
            Call method Program.bridgeData.supportType.Add with "Null"
        EndFor
    EndIf
    Initialise XDOF
    Initialise YDOF
    If Convert.ToInt16 with supportNodeTextbox.Text is less than
Program.bridgeData.nodes.Count
        Switch on supportType.Text
        Case "Vertical Fixed Support"
            Call method Program.bridgeData.restrainedDegreesOfFreedom.Add with
XDOF
            Call method Program.bridgeData.restrainedDegreesOfFreedom.Add with
YDOF
            Set Convert.ToInt32 of Program.bridgeData.supportType to "Vertical
Fixed Support"
            Call method Program.bridgeData.supportNode.Add with
Convert.ToInt16 supportNodeTextbox.Text
            Set supportNodeTextbox.Text to ""
            End of Case
        Case "Horizontal Fixed Support"
            Call method Program.bridgeData.restrainedDegreesOfFreedom.Add with
XDOF
            Call method Program.bridgeData.restrainedDegreesOfFreedom.Add with
YDOF
            Set Convert.ToInt32 of Program.bridgeData.supportType to
"Horizontal Fixed Support"
            Call method Program.bridgeData.supportNode.Add with
Convert.ToInt16 supportNodeTextbox.Text
            Set supportNodeTextbox.Text to ""
            End of Case
        Case "Vertical Roller"
```

```

        Call method Program.bridgeData.restrainedDegreesOfFreedom.Add with
XDOF
        Set Convert.ToInt32 of Program.bridgeData.supportType to "Vertical
Roller"
        Call method Program.bridgeData.supportNode.Add with
Convert.ToInt16 supportNodeTextbox.Text
        Set supportNodeTextbox.Text to ""
        End of Case
    Case "Horozontal Roller"
        Call method Program.bridgeData.restrainedDegreesOfFreedom.Add with
YDOF
        Set Convert.ToInt32 of Program.bridgeData.supportType to
"Horozontal Roller"
        Call method Program.bridgeData.supportNode.Add with
Convert.ToInt16 supportNodeTextbox.Text
        Set supportNodeTextbox.Text to ""
        End of Case
    Default
        Set supportNodeTextbox.Text to "Incorrect Variable Entered"
        End of Case
EndSwitch
Else
    Set supportNodeTextbox.Text to "Incorrect Variable Entered"
EndIf
Call method bridgeDrawing.Refresh
Catch
    Set supportNodeTextbox.Text to "Incorrect Variable Entered"
EndTry

```

Flowchart



**Procedure: supportType\_SelectedIndexChanged**

```
private void supportType_SelectedIndexChanged(object sender, EventArgs e)
```

**Parameters**

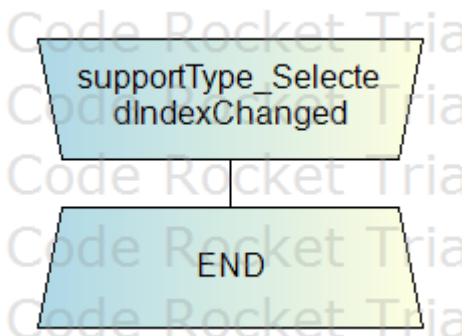
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Procedure: supportNodeTextbox\_TextChanged*

```
private void supportNodeTextbox_TextChanged(object sender, EventArgs e)
```

**Parameters**

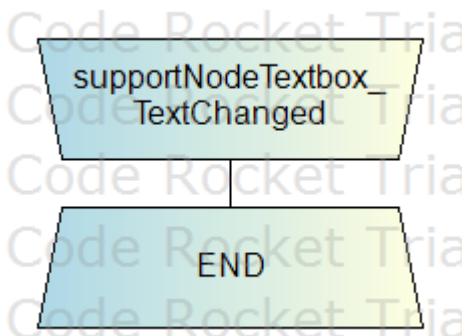
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



*Procedure: nodeListView\_SelectedIndexChanged*

```
private void nodeListView_SelectedIndexChanged(object sender,  
EventArgs e)
```

**Parameters**

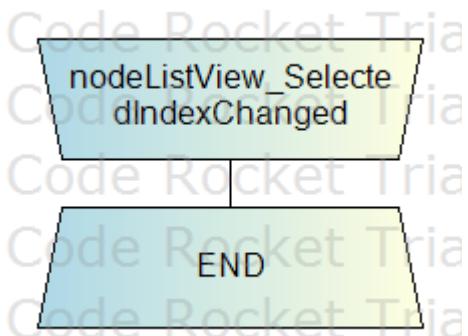
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: supportListView\_SelectedIndexChanged**

```
private void supportListView_SelectedIndexChanged(object sender,  
EventArgs e)
```

**Parameters**

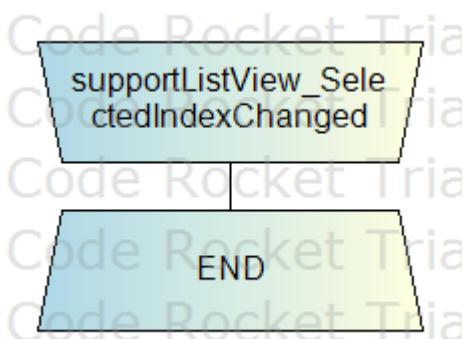
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

N/A

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

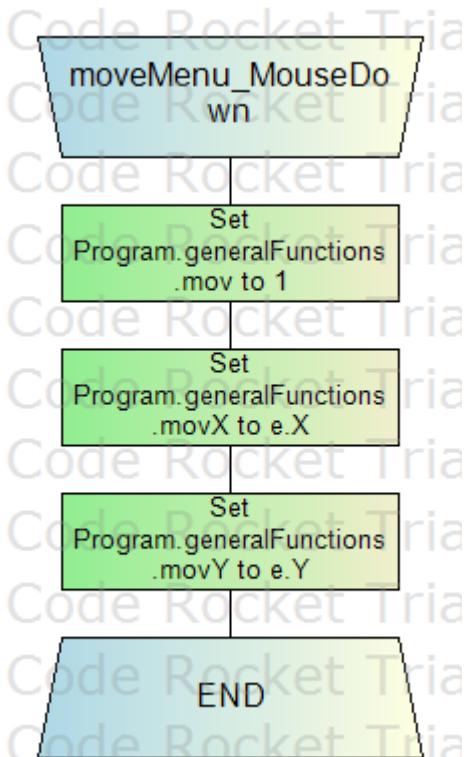
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

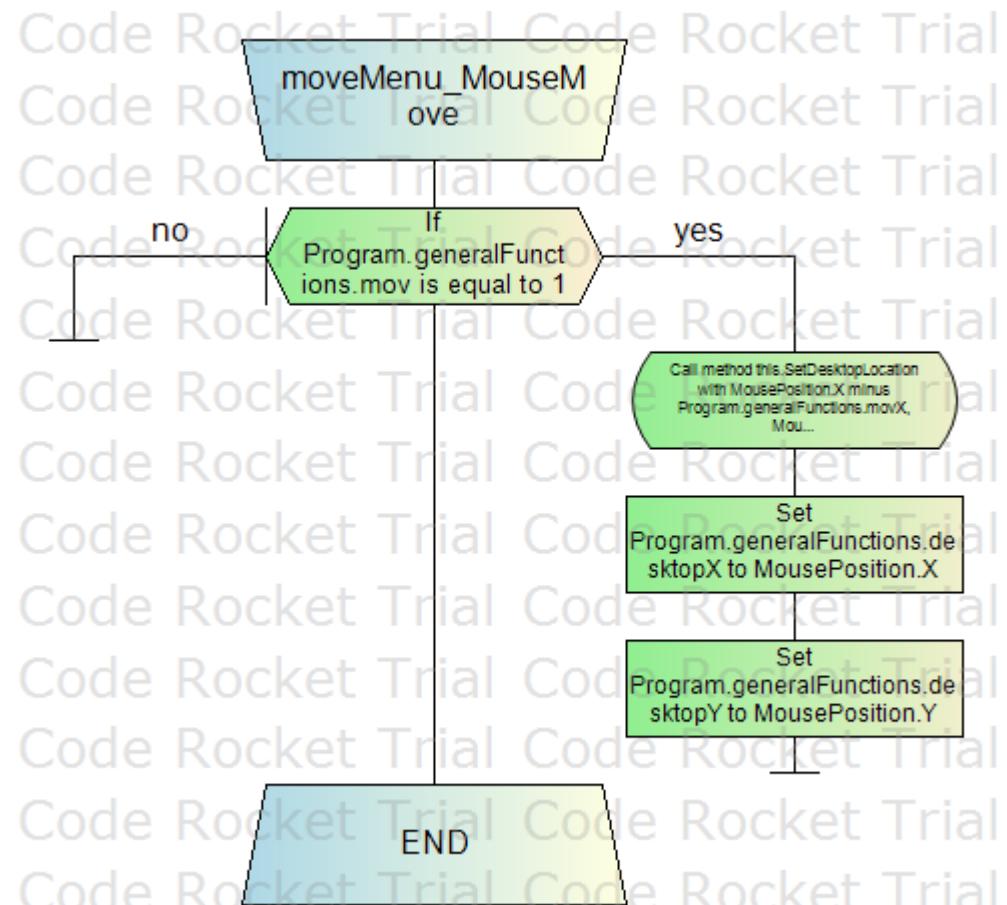
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



**Procedure: moveMenu\_MouseUp**

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

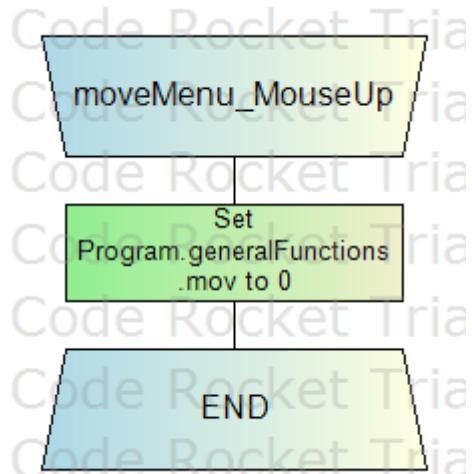
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

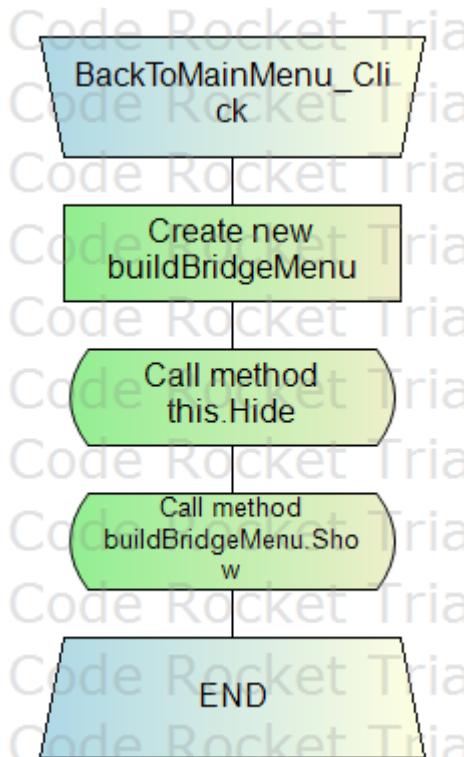
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

**Try**

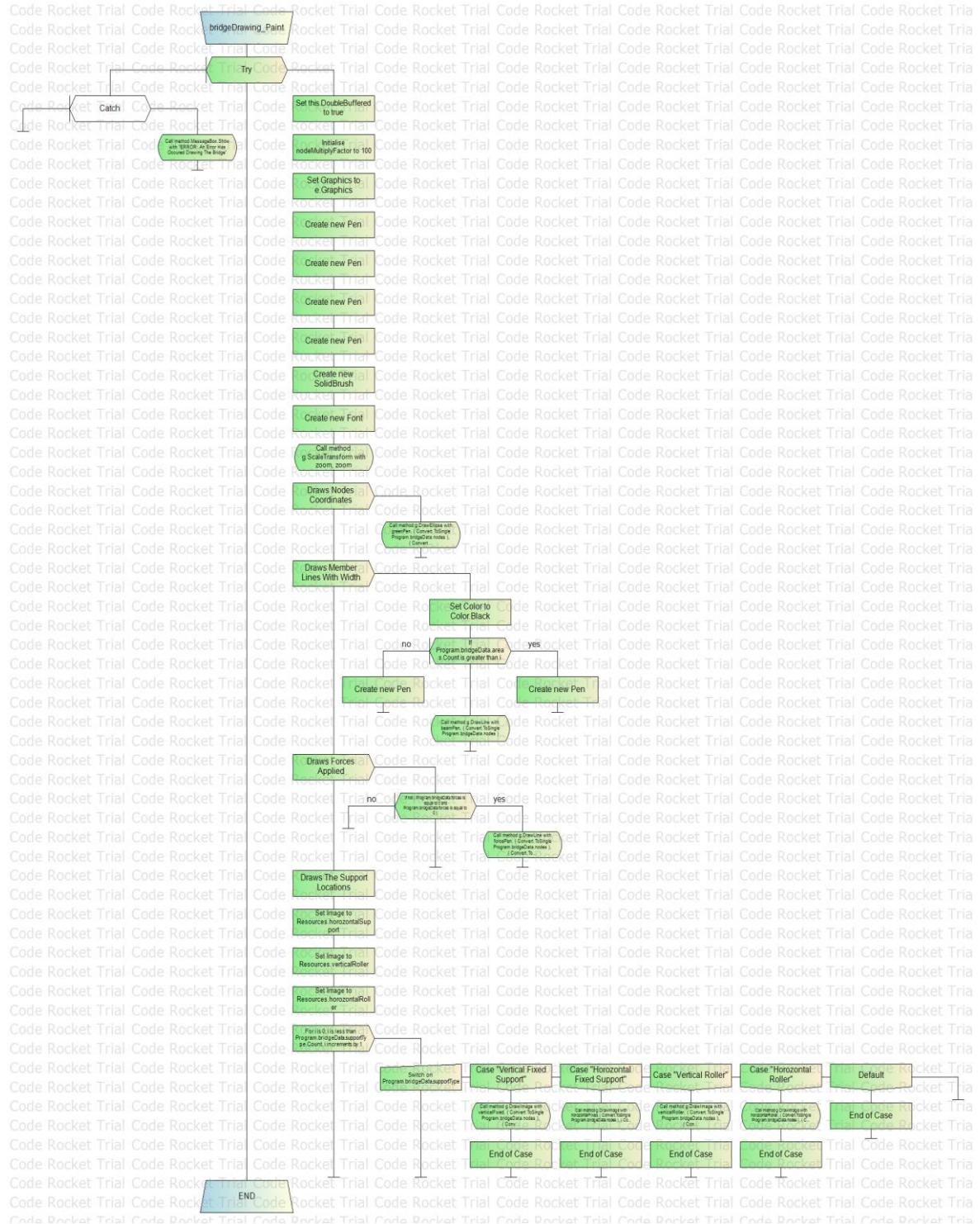
```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```

        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry

```

## Flowchart



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

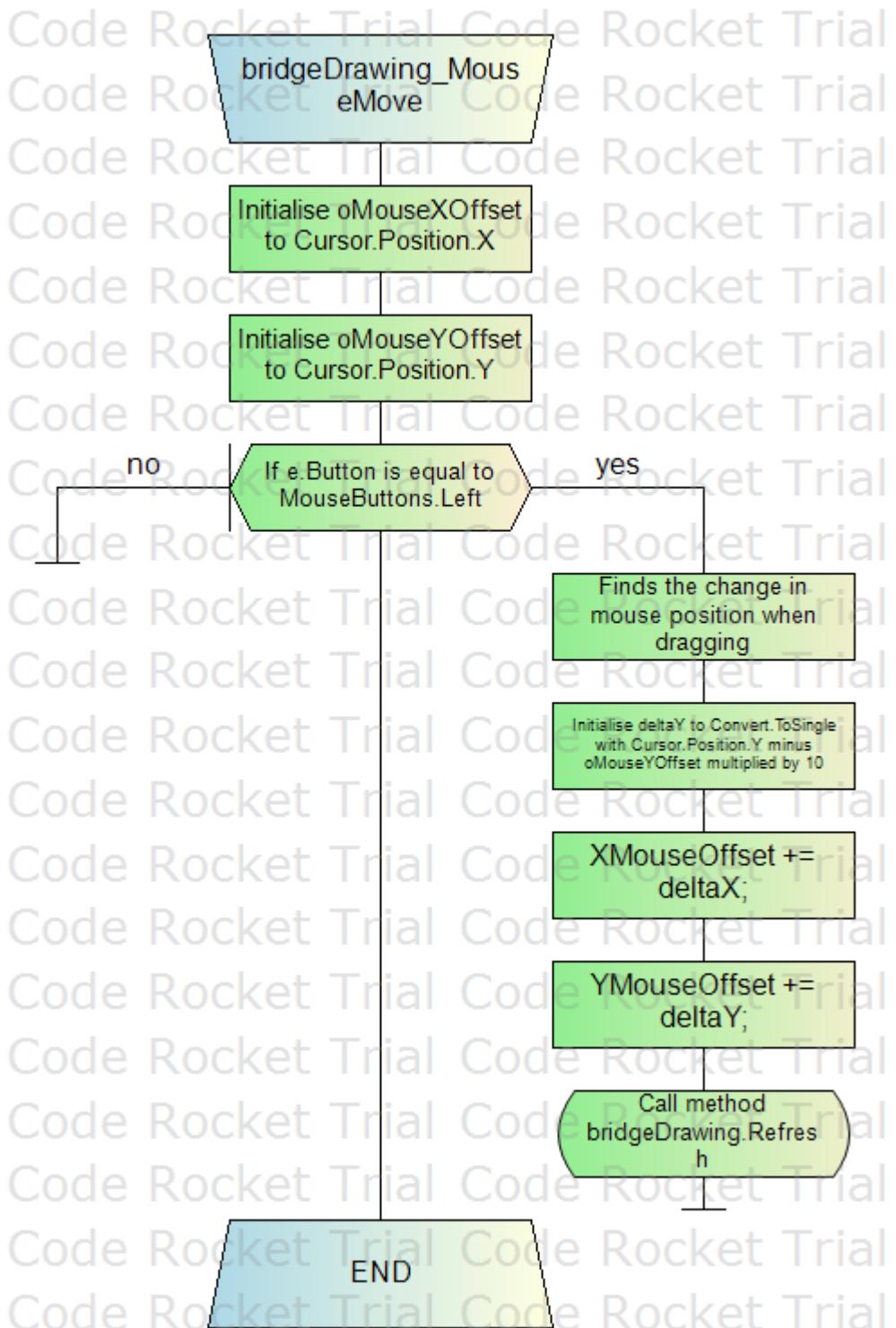
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

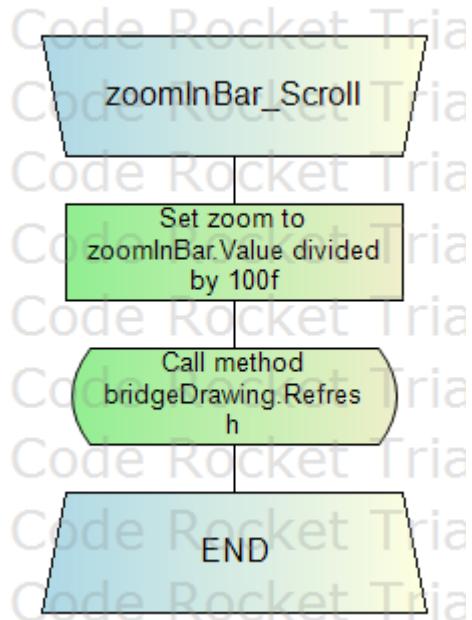
Returns Void

**Pseudocode**

Set zoom to zoomInBar.Value divided by 100f

Call method bridgeDrawing.Refresh

**Flowchart**



**Procedure: supportMenuTable\_Click**

```
private void supportMenuTable_Click(object sender, EventArgs e)
```

**Parameters**

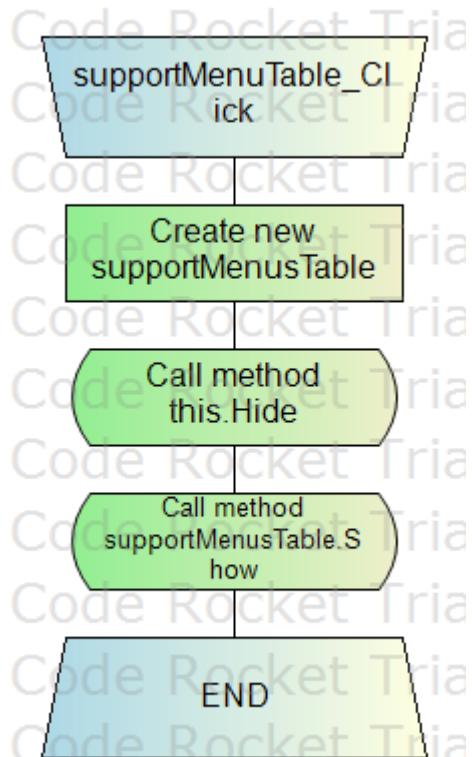
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new supportMenusTable  
Call method this.Hide  
Call method supportMenusTable.Show
```

**Flowchart**



**Code File:** supportMenus.Designer.cs

**Namespace:** Sectrics\_V2

namespace Sectrics\_V2

**Class:** supportMenus

partial class supportMenus

**Attribute:** components

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure:** Dispose

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

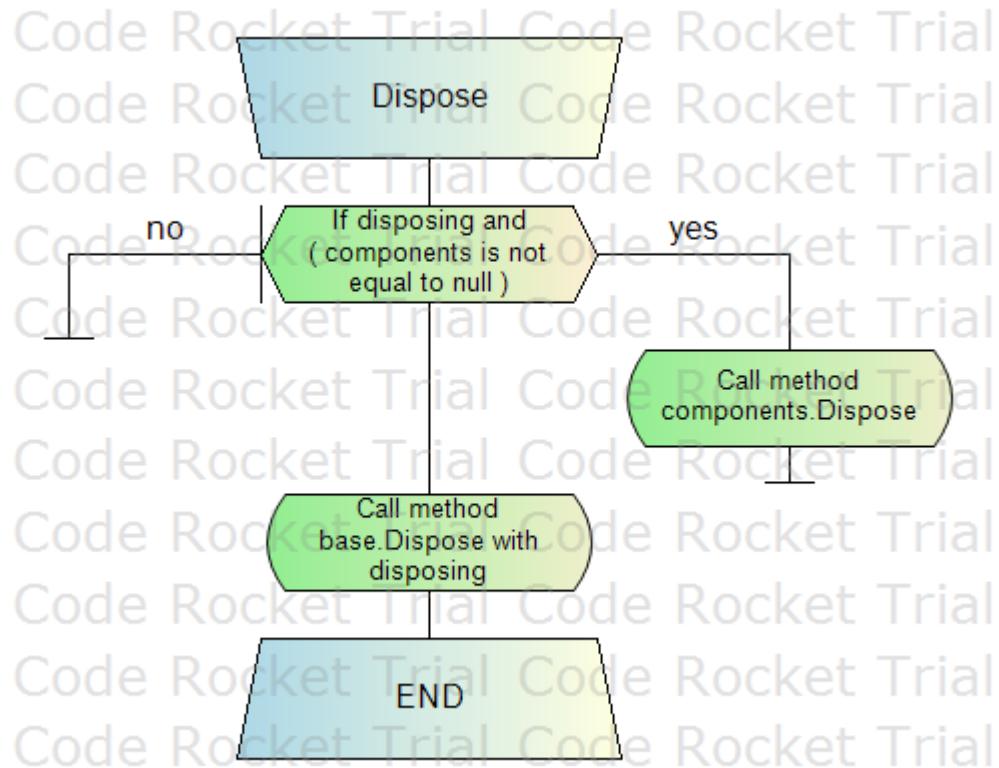
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

Flowchart



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.TextBox
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.ListBox
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.Button
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackColor to System.Drawing.Color.Transparent
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.TabStop to false
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackColor to System.Drawing.Color.Transparent
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.TabStop to false
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
addSupportButton
Set this.addSupportButton.BackColor to System.Drawing.Color.Transparent
Set this.addSupportButton.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.addSupportButton.FlatAppearance.BorderSize to 2
Set this.addSupportButton.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.addSupportButton.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font

```

```

Set this.addSupportButton.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.addSupportButton.Name to "addSupportButton"
Create new System.Drawing.Size
Set this.addSupportButton.TabIndex to 31
Set this.addSupportButton.TabStop to false
Set this.addSupportButton.Text to "ADD SUPPORT"
Set this.addSupportButton.UseVisualStyleBackColor to false
This.addSupportButton.Click += new System.EventHandler(this.addSupportButton_Click);
label2
Set this.label2.AutoSize to true
Set this.label2.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label2.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label2.Name to "label2"
Create new System.Drawing.Size
Set this.label2.TabIndex to 29
Set this.label2.Text to "SUPPORT TYPE:"
supportNodeTypebox
Create new System.Drawing.Font
Create new System.Drawing.Point
Set this.supportNodeTypebox.Name to "supportNodeTypebox"
Create new System.Drawing.Size
Set this.supportNodeTypebox.TabIndex to 0
Set this.supportNodeTypebox.Text to "Enter the support node here"
This.supportNodeTypebox.TextChanged += new
System.EventHandler(this.supportNodeTypebox_TextChanged);
label1
Set this.label1.AutoSize to true
Set this.label1.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 27
Set this.label1.Text to "SUPPORT NODE:"
supportType
Create new System.Drawing.Font
Set this.supportType.FormattingEnabled to true
Set this.supportType.ItemHeight to 33
Call method this.supportType.Items.AddRange
Create new System.Drawing.Point
Set this.supportType.Name to "supportType"
Create new System.Drawing.Size
Set this.supportType.TabIndex to 1
This.supportType.SelectedIndexChanged += new
System.EventHandler(this.supportType_SelectedIndexChanged);
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 36
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent
Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White

```

```

Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 37
Set this.BackToMainMenu.TabStop to false
Set this.BackToMainMenu.Text to "BACK"
Set this.BackToMainMenu.UseVisualStyleBackColor to false
This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 39
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 38
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
supportMenuTable
Set this.supportMenuTable.BackColor to System.Drawing.Color.Transparent
Set this.supportMenuTable.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.supportMenuTable.FlatAppearance.BorderSize to 2
Set this.supportMenuTable.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.supportMenuTable.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.supportMenuTable.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.supportMenuTable.Name to "supportMenuTable"
Create new System.Drawing.Size
Set this.supportMenuTable.TabIndex to 40
Set this.supportMenuTable.TabStop to false
Set this.supportMenuTable.Text to "TABLE"
Set this.supportMenuTable.UseVisualStyleBackColor to false
This.supportMenuTable.Click += new System.EventHandler(this.supportMenuTable_Click);
supportMenus
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.supportMenuTable
Call method this.Controls.Add with this.bridgeDrawing
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.BackToMainMenu
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.supportType

```

```
Call method this.Controls.Add with this.addSupportButton
Call method this.Controls.Add with this.label2
Call method this.Controls.Add with this.supportNodeTextbox
Call method this.Controls.Add with this.label1
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.DoubleBuffered to true
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "supportMenus"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout
```

### Flowchart



```

#endifregion
Attribute: minimize

private System.Windows.Forms.Button minimize;
Attribute: exitApplication

private System.Windows.Forms.Button exitApplication;
Attribute: addSupportButton

private System.Windows.Forms.Button addSupportButton;
Attribute: label2

private System.Windows.Forms.Label label2;
Attribute: supportNodeTextbox

private System.Windows.Forms.TextBox supportNodeTextbox;
Attribute: label1

private System.Windows.Forms.Label label1;
Attribute: supportType

private System.Windows.Forms.ListBox supportType;
Attribute: moveMenu

private System.Windows.Forms.Panel moveMenu;
Attribute: BackToMainMenu

private System.Windows.Forms.Button BackToMainMenu;
Attribute: bridgeDrawing

private System.Windows.Forms.Panel bridgeDrawing;
Attribute: zoomInBar

private System.Windows.Forms.TrackBar zoomInBar;
Attribute: supportMenuTable

private System.Windows.Forms.Button supportMenuTable;
Code File: supportMenusTable.cs

using Sectrics_V2.Properties;
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: Sectrics\_V2**

```
namespace Sectrics_V2
```

**Class: supportMenusTable**

```
public partial class supportMenusTable : Form
```

**Attribute: zoom**

```
float zoom = 1f;
```

**Attribute: xMouseOffset**

```
double xMouseOffset;
```

**Attribute: yMouseOffset**

```
double yMouseOffset;
```

**Attribute: cGrip**

```
private const int cGrip = 16;
```

**Attribute: cCaption**

```
private const int cCaption = 32;
```

**Procedure: Constructor**

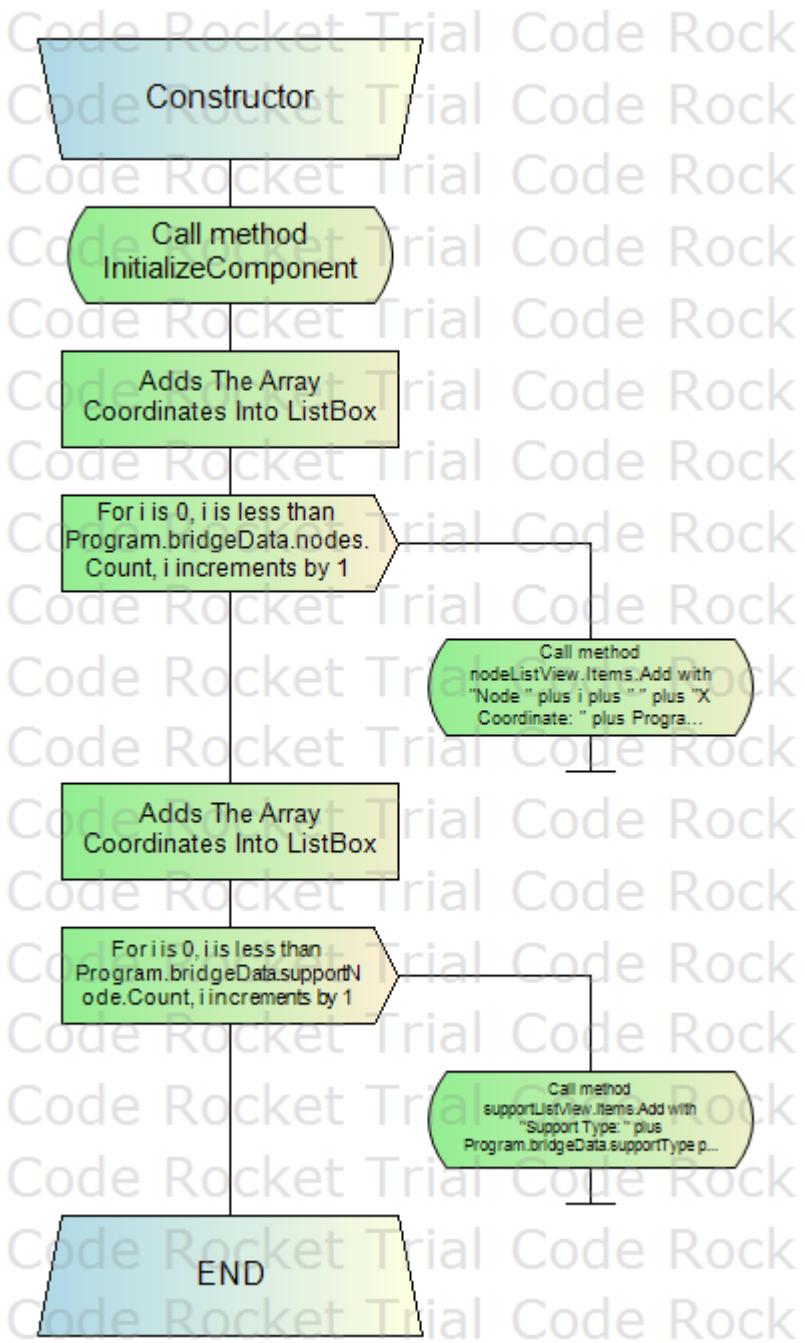
```
public supportMenusTable()
```

Returns supportMenusTable

**Pseudocode**

```
Call method InitializeComponent
Adds The Array Coordinates Into ListBox
For i is 0, i is less than Program.bridgeData.nodes.Count, i increments by 1
    Call method nodeListView.Items.Add with "Node " plus i plus " " plus "X
    Coordinate: " plus Program.bridgeData.nodes plus " | Y Coordinate: " plus
    Program.bridgeData.nodes
EndFor
Adds The Array Coordinates Into ListBox
For i is 0, i is less than Program.bridgeData.supportNode.Count, i increments by 1
    Call method supportListView.Items.Add with "Support Type: " plus
    Program.bridgeData.supportType plus " | At Node: " plus
    Program.bridgeData.supportNode
EndFor
```

Flowchart



**Procedure: WndProc**

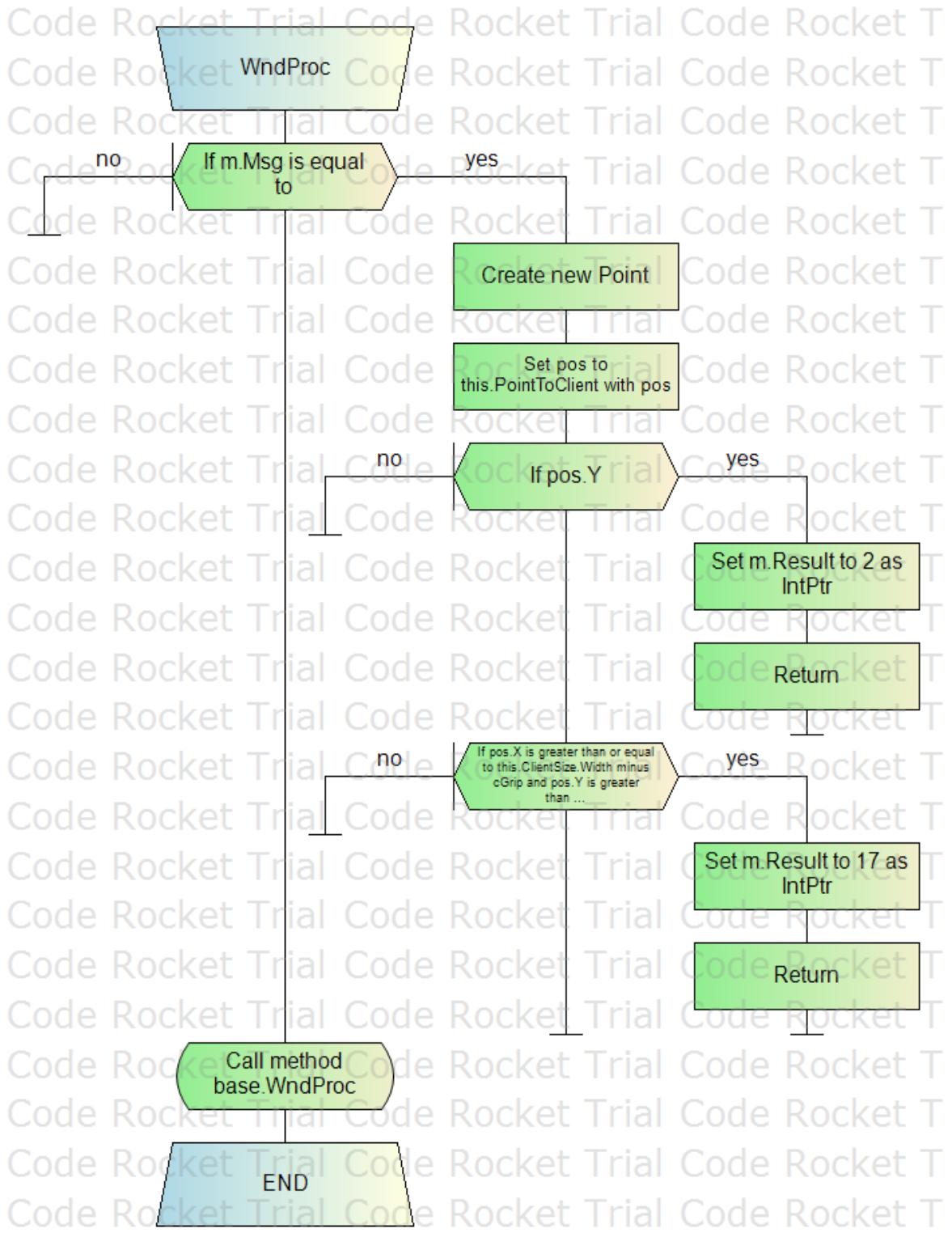
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

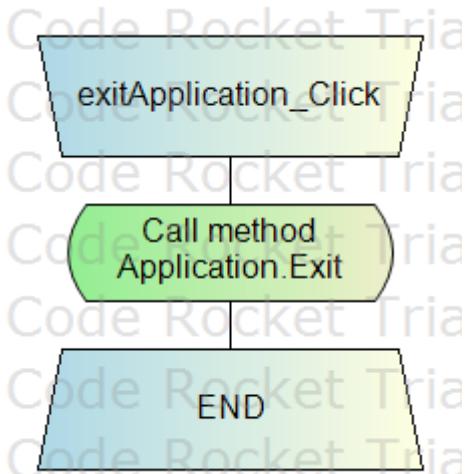
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

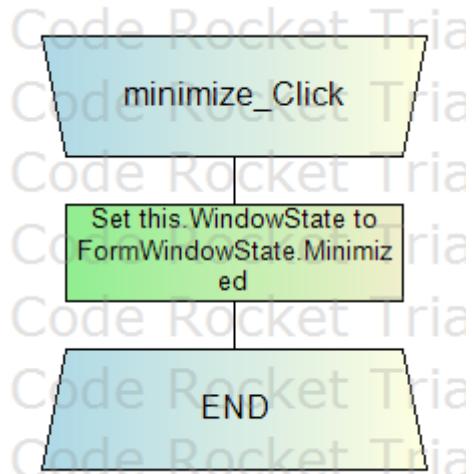
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

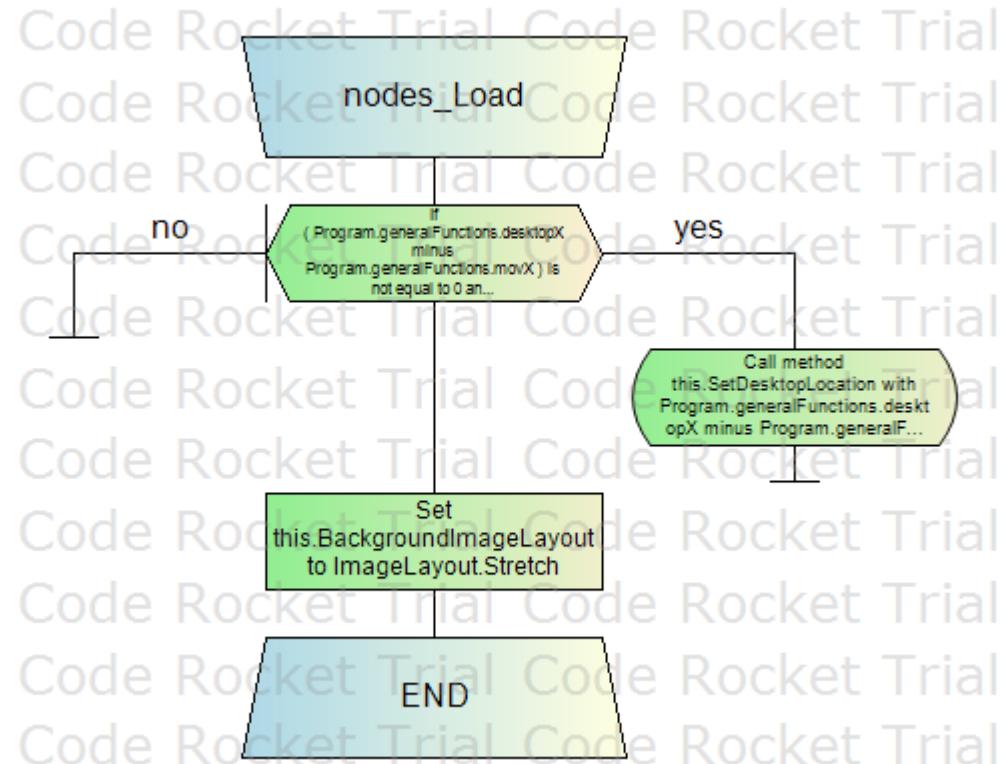
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: BackToMainMenu\_Click**

```
private void BackToMainMenu_Click(object sender, EventArgs e)
```

**Parameters**

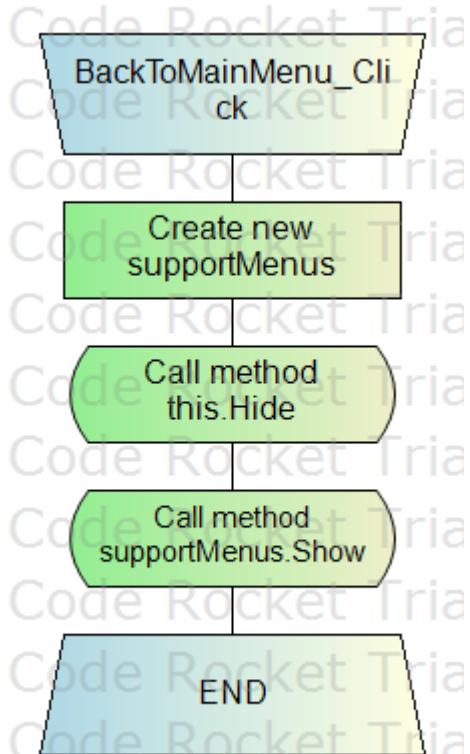
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new supportMenus  
Call method this.Hide  
Call method supportMenus.Show
```

**Flowchart**



**Procedure: clearAll\_Click**

```
private void clearAll_Click(object sender, EventArgs e)
```

**Parameters**

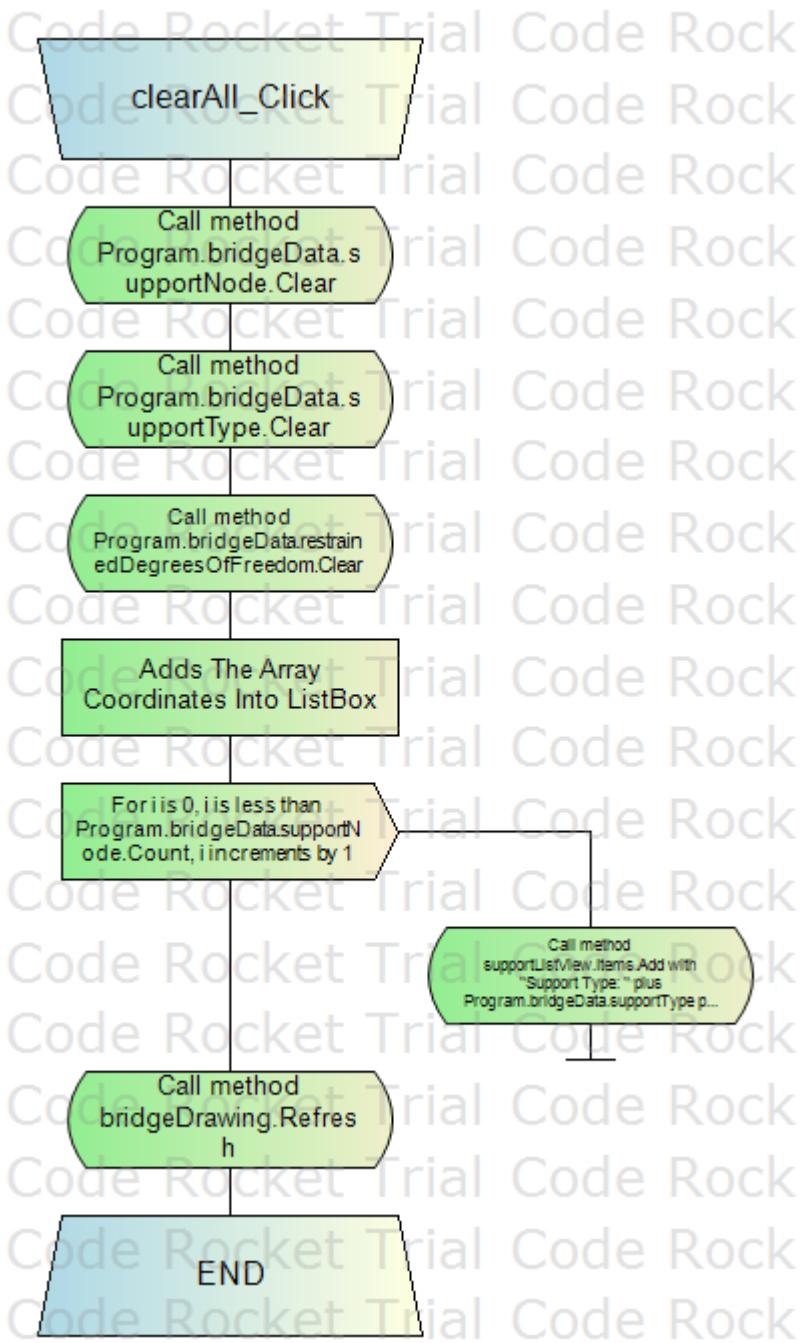
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Program.bridgeData.supportNode.Clear  
Call method Program.bridgeData.supportType.Clear  
Call method Program.bridgeData.restrainedDegreesOfFreedom.Clear  
Adds The Array Coordinates Into ListBox  
For i is 0, i is less than Program.bridgeData.supportNode.Count, i increments by 1  
    Call method supportListView.Items.Add with "Support Type: " plus  
    Program.bridgeData.supportType plus " | At Node: " plus  
    Program.bridgeData.supportNode  
EndFor  
Call method bridgeDrawing.Refresh
```

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

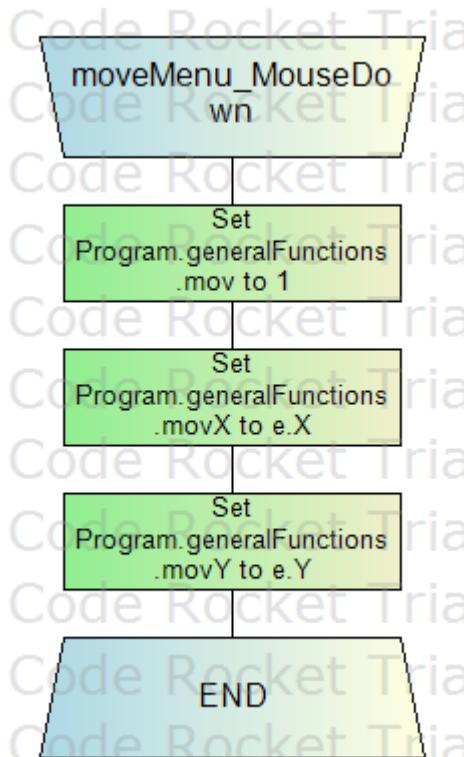
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

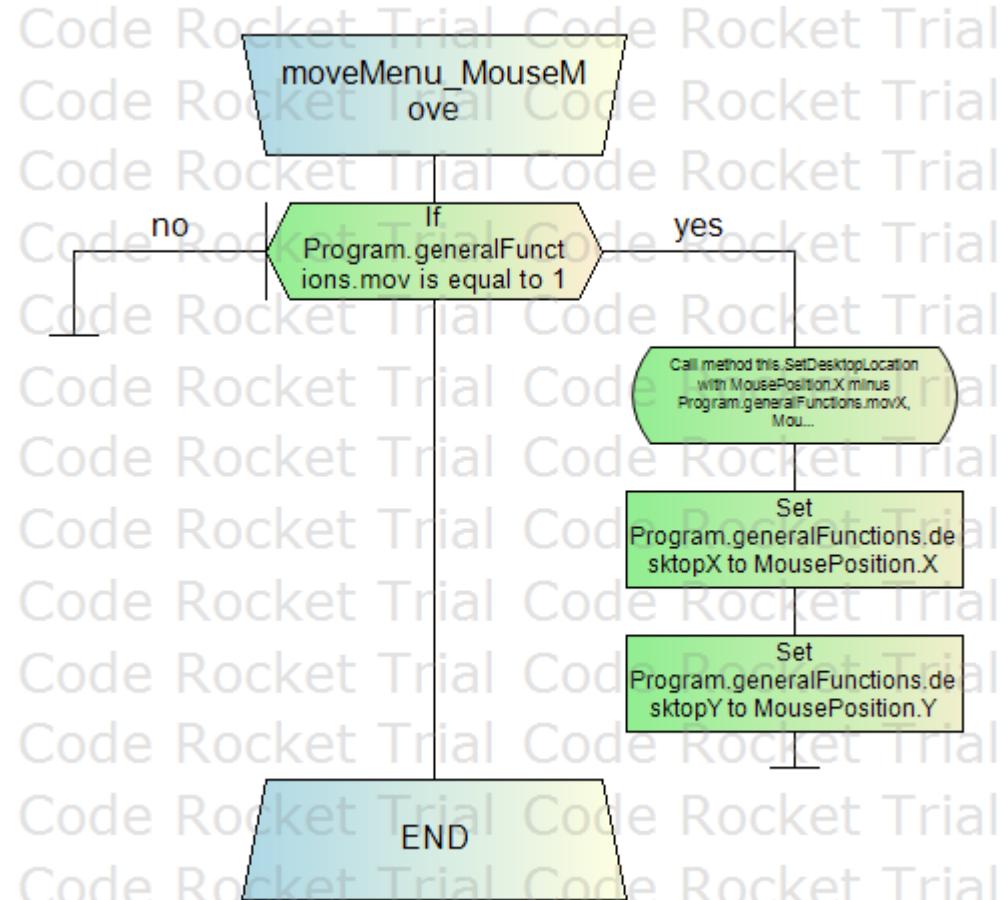
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



*Procedure: moveMenu\_MouseUp*

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

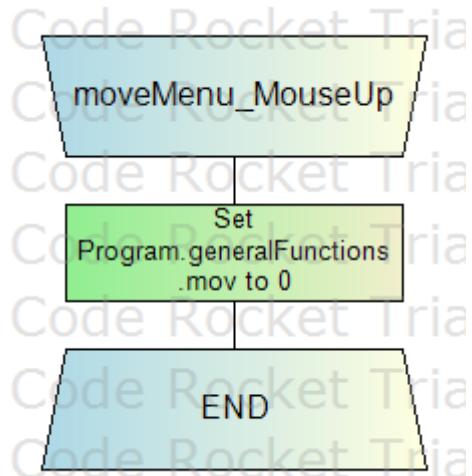
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



**Procedure: bridgeDrawing\_Paint**

```
private void bridgeDrawing_Paint(object sender, PaintEventArgs e)
```

**Parameters**

- sender - Object
- e - PaintEventArgs

Returns Void

**Pseudocode**

**Try**

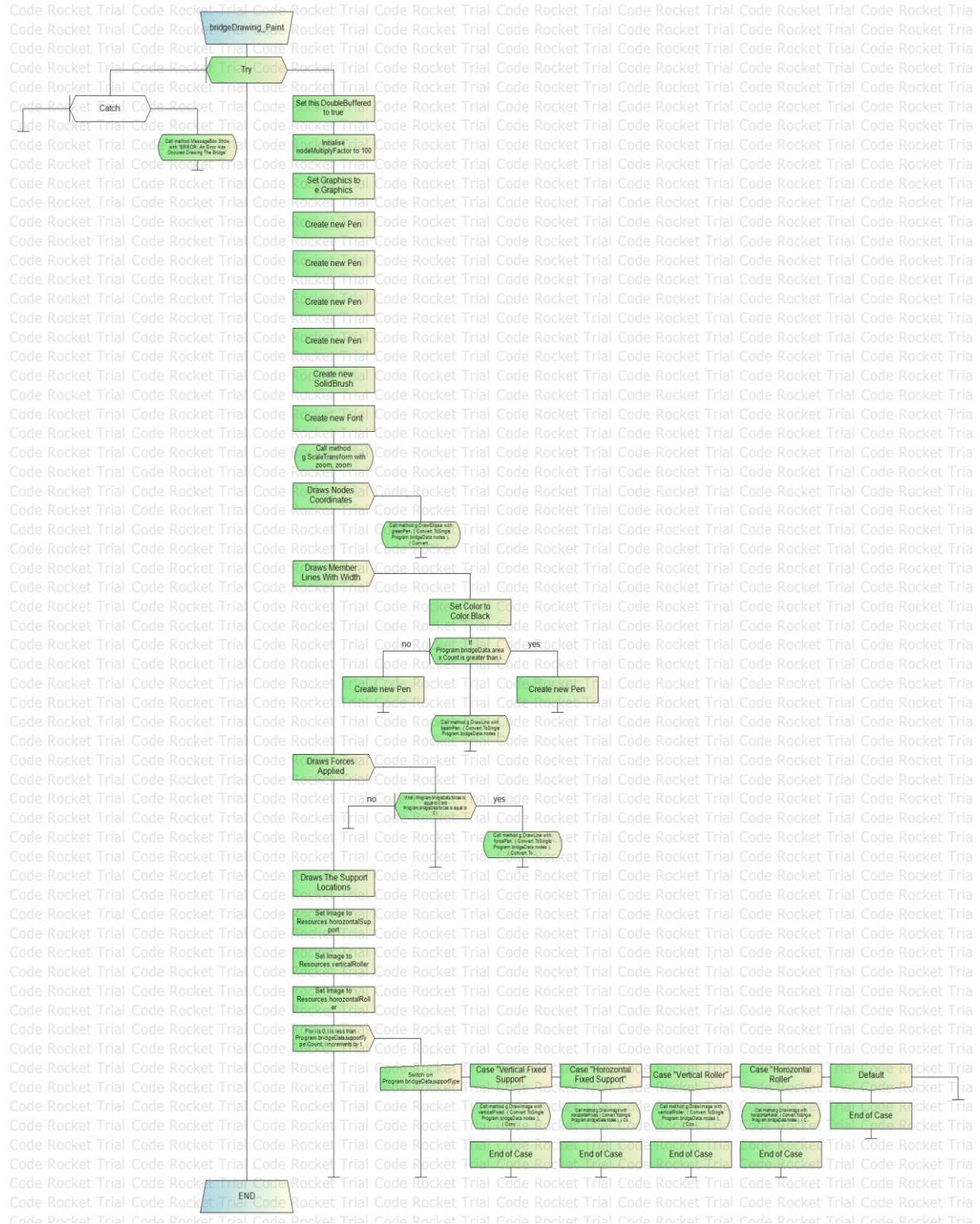
```
    Set this.DoubleBuffered to true
    Initialise nodeMultiplyFactor to 100
    Set Graphics to e.Graphics
    Create new Pen
    Create new Pen
    Create new Pen
    Create new Pen
    Create new SolidBrush
    Create new Font
    Call method g.ScaleTransform with zoom, zoom
    Draws Nodes Coordinates
        Call method g.DrawEllipse with greenPen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
    EndFor
    Draws Member Lines With Width
        Set Color to Color.Black
        If Program.bridgeData.areas.Count is greater than i
            Create new Pen
        Else
            Create new Pen
        EndIf
        Call method g.DrawLine with beamPen, ( Convert.ToDouble Program.bridgeData.nodes )
    EndFor
    Draws Forces Applied
        If not ( Program.bridgeData.forces is equal to 0 and
Program.bridgeData.forces is equal to 0 )
            Call method g.DrawLine with forcePen, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )...
        EndIf
    EndFor
    Draws The Support Locations
    Set Image to Resources.horozntalSupport
    Set Image to Resources.verticalRoller
    Set Image to Resources.horozntalRoller
    For i is 0, i is less than Program.bridgeData.supportType.Count, i increments
by 1
        Switch on Program.bridgeData.supportType
        Case "Vertical Fixed Support"
            Call method g.DrawImage with verticalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Fixed Support"
            Call method g.DrawImage with horozntalFixed, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Vertical Roller"
            Call method g.DrawImage with verticalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
        Case "Horozntal Roller"
            Call method g.DrawImage with horozntalRoller, ( Convert.ToDouble Program.bridgeData.nodes ), ( Convert.ToDouble Program.bridgeData.nodes )
        End of Case
    Default
```

```

        End of Case
    EndSwitch
EndFor
Catch
    Call method MessageBox.Show with "ERROR: An Error Has Occured Drawing The
Bridge"
EndTry

```

## Flowchart



**Procedure: bridgeDrawing\_MouseMove**

```
private void bridgeDrawing_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

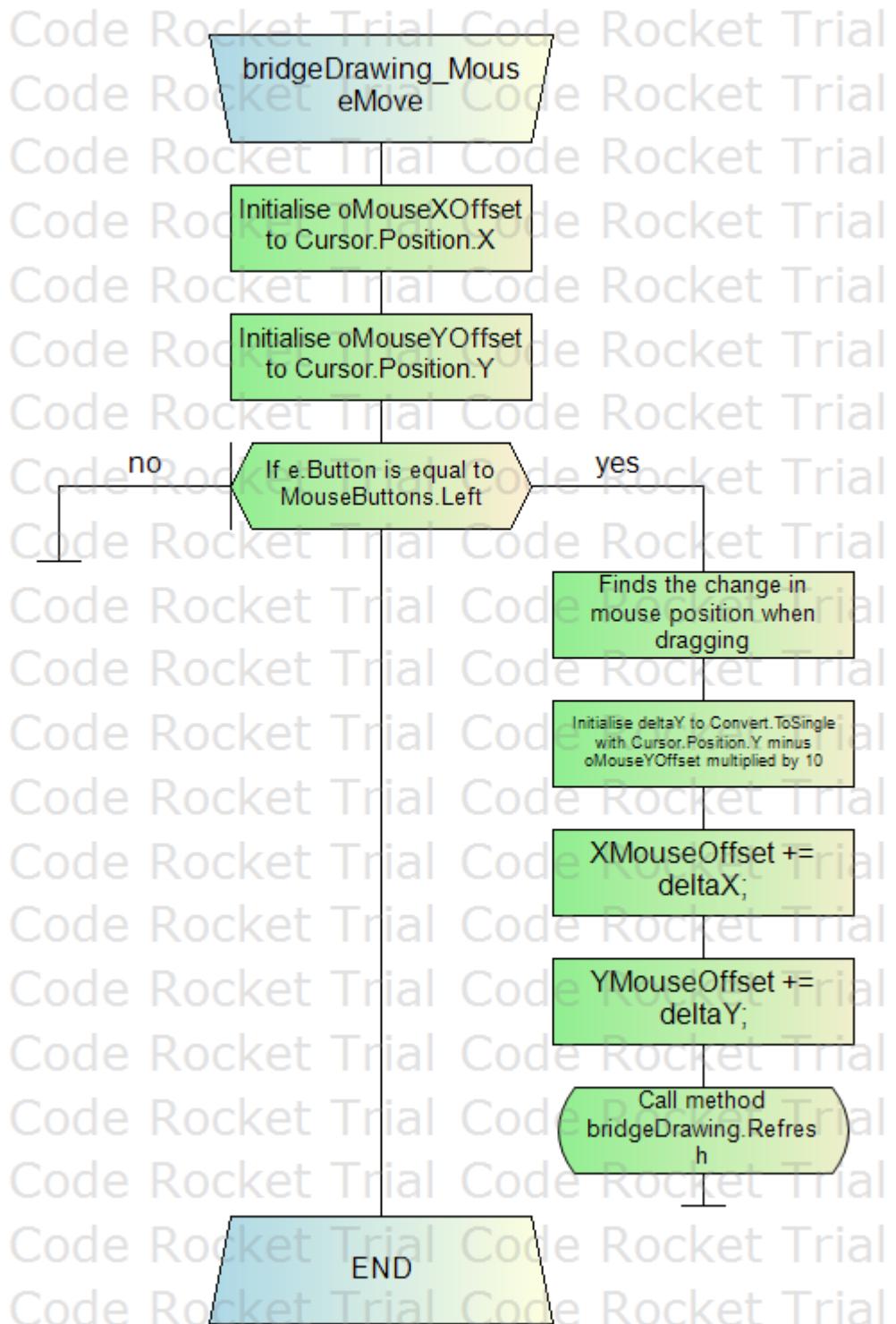
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Initialise oMouseXOffset to Cursor.Position.X  
Initialise oMouseYOffset to Cursor.Position.Y  
If e.Button is equal to MouseButtons.Left  
    Finds the change in mouse position when dragging  
    Initialise deltaY to Convert.ToSingle with Cursor.Position.Y minus  
oMouseYOffset multiplied by 10  
    XMouseOffset += deltaX;  
    YMouseOffset += deltaY;  
    Call method bridgeDrawing.Refresh  
EndIf
```

**Flowchart**



**Procedure: zoomInBar\_Scroll**

```
private void zoomInBar_Scroll(object sender, EventArgs e)
```

**Parameters**

- sender - Object
- e - EventArgs

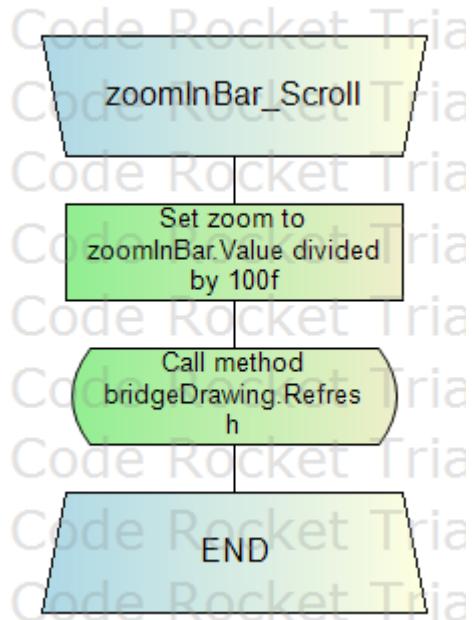
Returns Void

**Pseudocode**

```
Set zoom to zoomInBar.Value divided by 100f
```

```
Call method bridgeDrawing.Refresh
```

**Flowchart**



*Code File: supportMenusTable.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

**Class: supportMenusTable**

partial class supportMenusTable

**Attribute: components**

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure: Dispose**

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

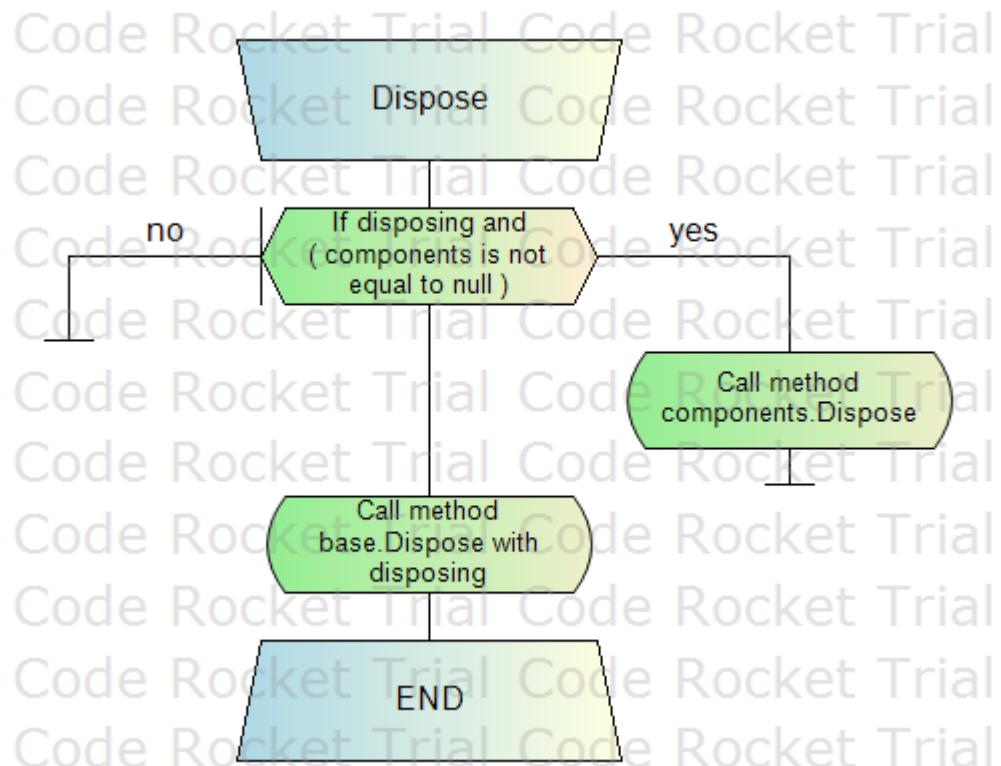
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

**Flowchart**



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Panel
Create new System.Windows.Forms.TrackBar
Create new System.Windows.Forms.ListBox
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.ListBox
Create new System.Windows.Forms.Panel
with ( System.ComponentModel.ISupportInitialize )
Call method this.SuspendLayout
minimize
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
BackToMainMenu
Set this.BackToMainMenu.BackColor to System.Drawing.Color.Transparent
Set this.BackToMainMenu.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.BackToMainMenu.FlatAppearance.BorderSize to 2
Set this.BackToMainMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ), ( int ), ( int ) )
Set this.BackToMainMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.BackToMainMenu.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.BackToMainMenu.Name to "BackToMainMenu"
Create new System.Drawing.Size
Set this.BackToMainMenu.TabIndex to 38
Set this.BackToMainMenu.TabStop to false
Set this.BackToMainMenu.Text to "BACK"

```

```

Set this.BackToMainMenu.UseVisualStyleBackColor to false
This.BackToMainMenu.Click += new System.EventHandler(this.BackToMainMenu_Click);
bridgeDrawing
Set this.bridgeDrawing.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.bridgeDrawing.Name to "bridgeDrawing"
Create new System.Drawing.Size
Set this.bridgeDrawing.TabIndex to 41
This.bridgeDrawing.Paint += new
System.Windows.Forms.PaintEventHandler(this.bridgeDrawing_Paint);
This.bridgeDrawing.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.bridgeDrawing_MouseMove);
zoomInBar
Set this.zoomInBar.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ),
( ( int ) ), ( ( int ) )
Create new System.Drawing.Point
Set this.zoomInBar.Maximum to 100
Set this.zoomInBar.Minimum to 1
Set this.zoomInBar.Name to "zoomInBar"
Create new System.Drawing.Size
Set this.zoomInBar.TabIndex to 40
Set this.zoomInBar.TabStop to false
Set this.zoomInBar.Value to 100
This.zoomInBar.Scroll += new System.EventHandler(this.zoomInBar_Scroll);
nodeListView
Set this.nodeTypeView.FormattingEnabled to true
Set this.nodeTypeView.ItemHeight to 25
Create new System.Drawing.Point
Set this.nodeTypeView.Name to "nodeTypeView"
Create new System.Drawing.Size
Set this.nodeTypeView.TabIndex to 42
clearAll
Set this.clearAll.BackColor to System.Drawing.Color.Transparent
Set this.clearAll.FlatAppearance.BorderColor to System.Drawing.Color.White
Set this.clearAll.FlatAppearance.BorderSize to 2
Set this.clearAll.FlatAppearance.MouseOverBackColor to System.Drawing.Color.FromArgb
with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.clearAll.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Font
Set this.clearAll.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.clearAll.Name to "clearAll"
Create new System.Drawing.Size
Set this.clearAll.TabIndex to 44
Set this.clearAll.TabStop to false
Set this.clearAll.Text to "CLEAR ALL"
Set this.clearAll.UseVisualStyleBackColor to false
This.clearAll.Click += new System.EventHandler(this.clearAll_Click);
supportListView
Set this.supportListView.FormattingEnabled to true
Set this.supportListView.ItemHeight to 25
Create new System.Drawing.Point
Set this.supportListView.Name to "supportListView"
Create new System.Drawing.Size
Set this.supportListView.TabIndex to 43
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 45
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new

```

```

System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
supportMenusTable
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.nodeListView
Call method this.Controls.Add with this.clearAll
Call method this.Controls.Add with this.supportListView
Call method this.Controls.Add with this.bridgeDrawing
Call method this.Controls.Add with this.zoomInBar
Call method this.Controls.Add with this.BackToMainMenu
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "supportMenusTable"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
with ( System.ComponentModel.ISupportInitialize.ISupportInitialize )
Call method this.ResumeLayout with false
Call method this.PerformLayout

```

### Flowchart



```

#endifregion
Attribute: minimize

private System.Windows.Forms.Button minimize;
Attribute: exitApplication

private System.Windows.Forms.Button exitApplication;
Attribute: BackToMainMenu

private System.Windows.Forms.Button BackToMainMenu;
Attribute: bridgeDrawing

private System.Windows.Forms.Panel bridgeDrawing;
Attribute: zoomInBar

private System.Windows.Forms.TrackBar zoomInBar;
Attribute: nodeListView

private System.Windows.Forms.ListBox nodeListView;
Attribute: clearAll

private System.Windows.Forms.Button clearAll;
Attribute: supportListView

private System.Windows.Forms.ListBox supportListView;
Attribute: moveMenu

private System.Windows.Forms.Panel moveMenu;
Code File: tutorialMenu.cs

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: Sectrics_V2

namespace Sectrics_V2
Class: tutorialMenu

public partial class tutorialMenu : Form

```

*Attribute: cGrip*

```
private const int cGrip = 16;
```

*Attribute: cCaption*

```
private const int cCaption = 32;
```

*Procedure: Constructor*

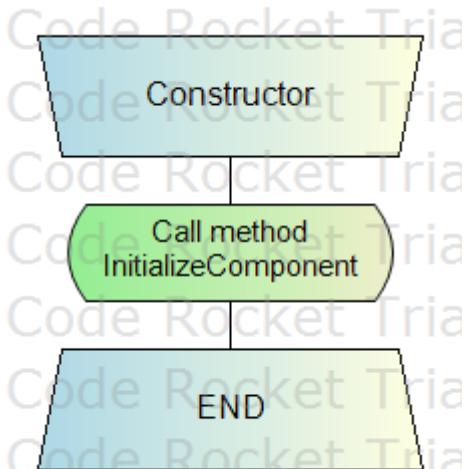
```
public tutorialMenu()
```

Returns tutorialMenu

**Pseudocode**

```
Call method InitializeComponent
```

Flowchart



**Procedure: WndProc**

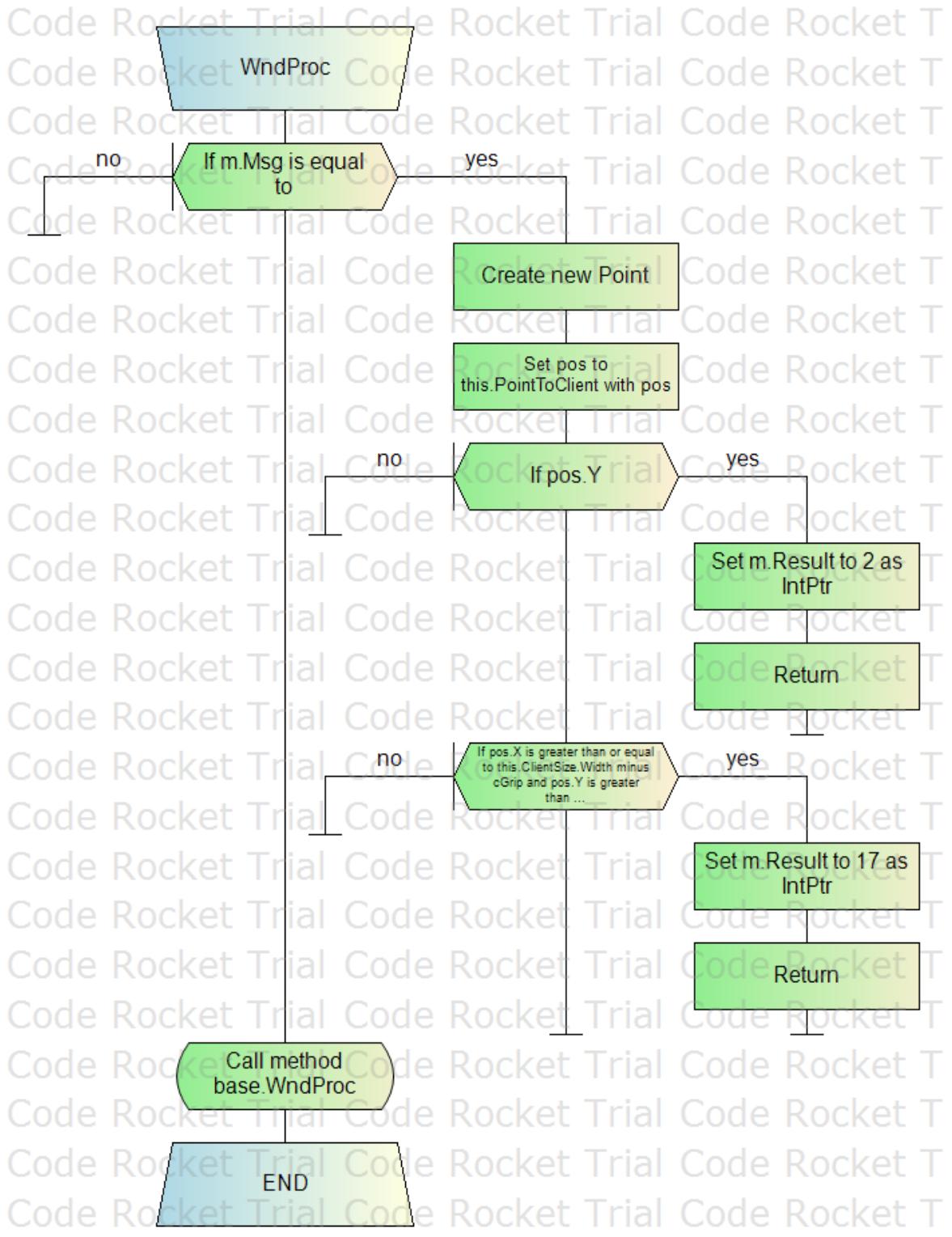
```
protected override void WndProc(ref Message m)
```

Returns Void

**Pseudocode**

```
If m.Msg is equal to
    Create new Point
    Set pos to this.PointToClient with pos
    If pos.Y
        Set m.Result to 2 as IntPtr
        Return
    EndIf
    If pos.X is greater than or equal to this.ClientSize.Width minus cGrip and
    pos.Y is greater than or equal to this.ClientSize.Height minus cGrip
        Set m.Result to 17 as IntPtr
        Return
    EndIf
EndIf
Call method base.WndProc
```

**Flowchart**



**Procedure: exitApplication\_Click**

```
private void exitApplication_Click(object sender, EventArgs e)
```

**Parameters**

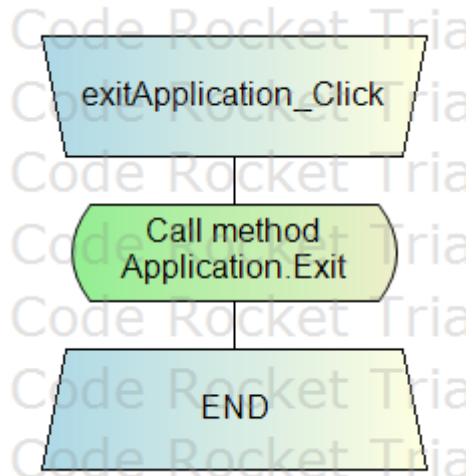
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Call method Application.Exit
```

**Flowchart**



*Procedure: minimize\_Click*

```
private void minimize_Click(object sender, EventArgs e)
```

**Parameters**

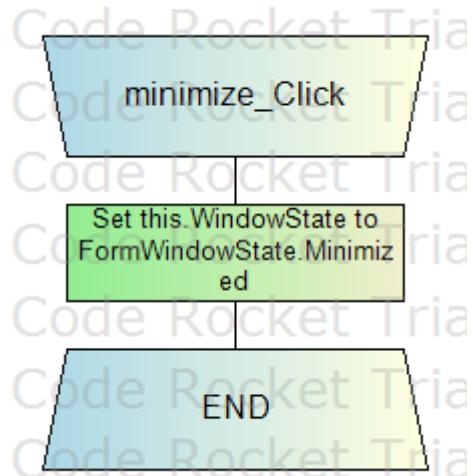
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Set this.WindowState to FormWindowState.Minimized
```

**Flowchart**



**Procedure: nodesMenu\_Click**

```
private void nodesMenu_Click(object sender, EventArgs e)
```

**Parameters**

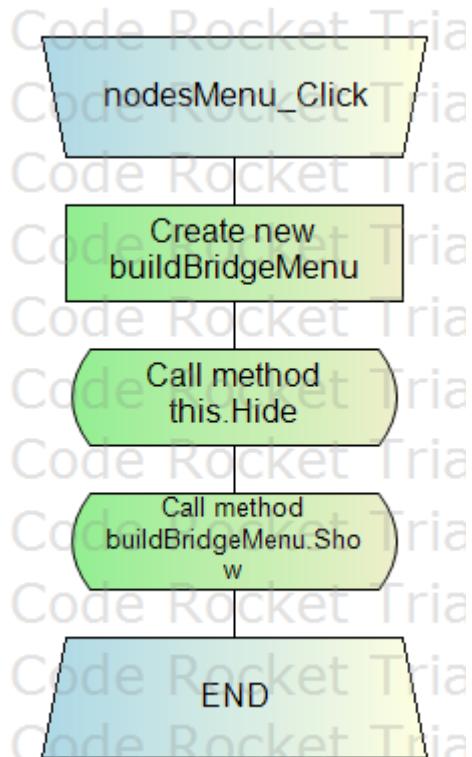
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
Create new buildBridgeMenu  
Call method this.Hide  
Call method buildBridgeMenu.Show
```

**Flowchart**



**Procedure: nodes\_Load**

```
private void nodes_Load(object sender, EventArgs e)
```

**Parameters**

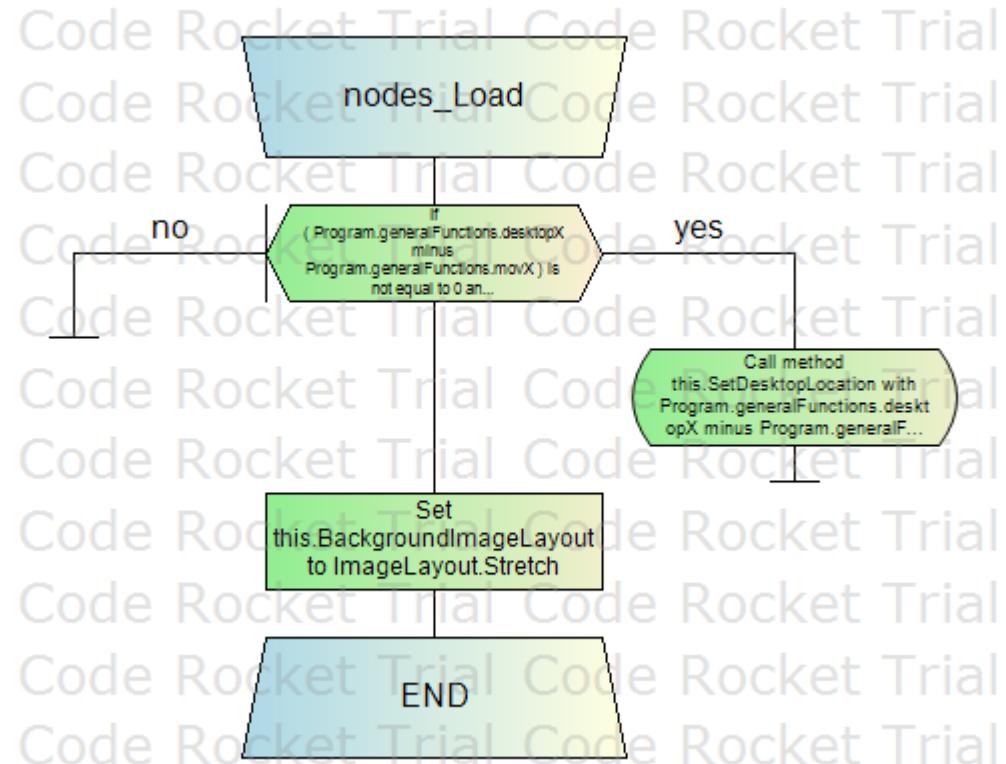
- sender - Object
- e - EventArgs

Returns Void

**Pseudocode**

```
If ( Program.generalFunctions.desktopX minus Program.generalFunctions.movX ) is not  
equal to 0 and ( Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY ) is not equal to null  
    Call method this.SetDesktopLocation with Program.generalFunctions.desktopX  
minus Program.generalFunctions.movX, Program.generalFunctions.desktopY minus  
Program.generalFunctions.movY  
EndIf  
Set this.BackgroundImageLayout to ImageLayout.Stretch
```

**Flowchart**



**Procedure: moveMenu\_MouseDown**

```
private void moveMenu_MouseDown(object sender, MouseEventArgs e)
```

**Parameters**

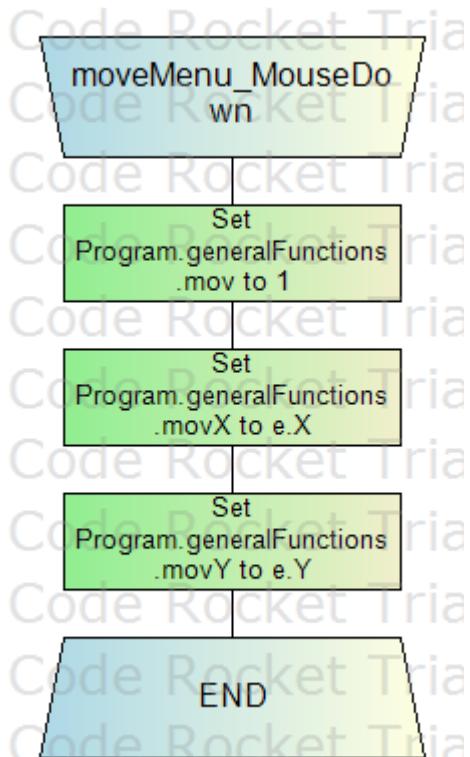
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 1  
Set Program.generalFunctions.movX to e.X  
Set Program.generalFunctions.movY to e.Y
```

**Flowchart**



**Procedure: moveMenu\_MouseMove**

```
private void moveMenu_MouseMove(object sender, MouseEventArgs e)
```

**Parameters**

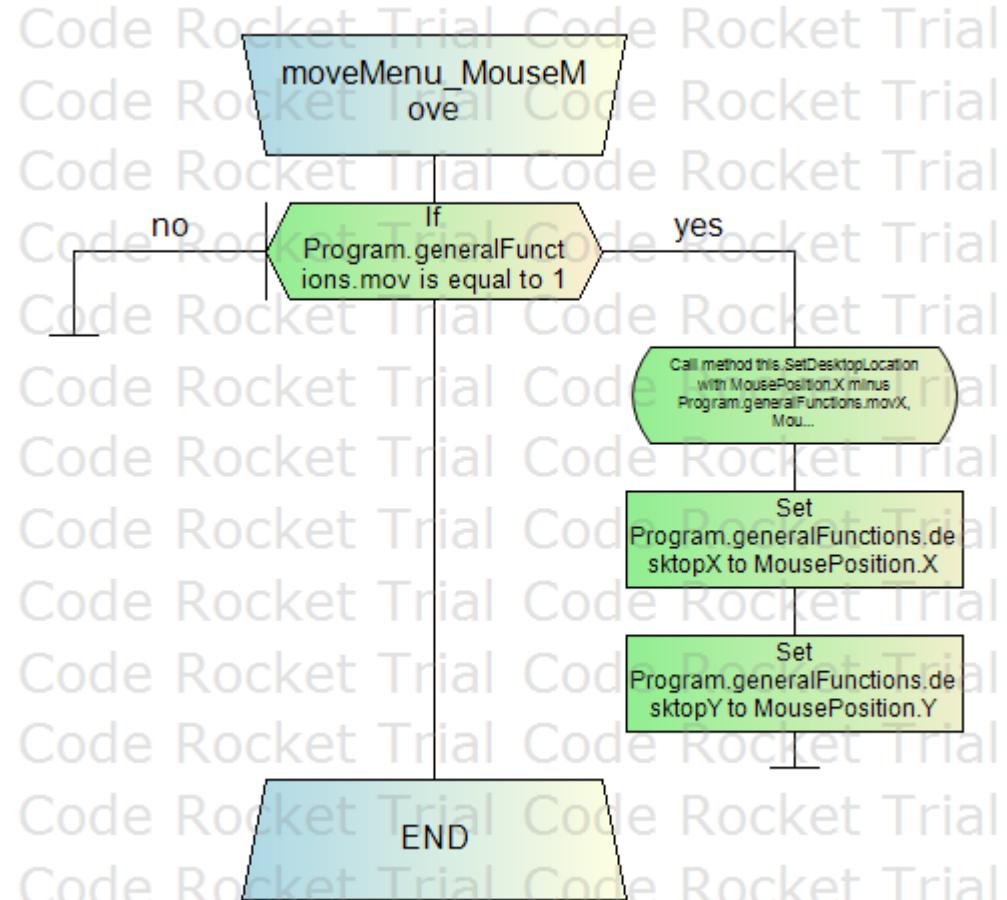
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
If Program.generalFunctions.mov is equal to 1
    Call method this.SetDesktopLocation with MousePosition.X minus
    Program.generalFunctions.movX, MousePosition.Y minus Program.generalFunctions.movY
        Set Program.generalFunctions.desktopX to MousePosition.X
        Set Program.generalFunctions.desktopY to MousePosition.Y
EndIf
```

**Flowchart**



*Procedure: moveMenu\_MouseUp*

```
private void moveMenu_MouseUp(object sender, MouseEventArgs e)
```

**Parameters**

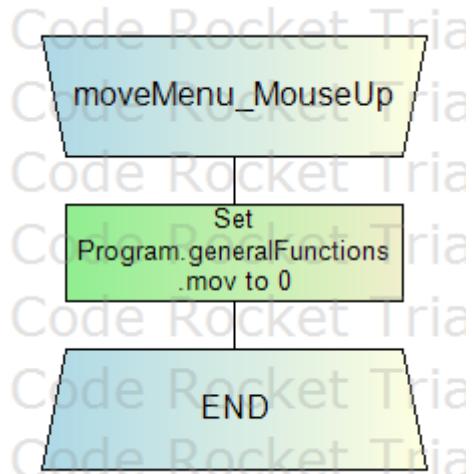
- sender - Object
- e - MouseEventArgs

Returns Void

**Pseudocode**

```
Set Program.generalFunctions.mov to 0
```

**Flowchart**



*Code File: tutorialMenu.Designer.cs*

*Namespace: Sectrics\_V2*

namespace Sectrics\_V2

**Class: tutorialMenu**

partial class tutorialMenu

**Attribute: components**

private System.ComponentModel.IContainer components = null;

Required designer variable.

**Procedure: Dispose**

protected override void Dispose (bool disposing)

Clean up any resources being used.

**Parameters**

- disposing - true if managed resources should be disposed; otherwise, false.

Returns Void

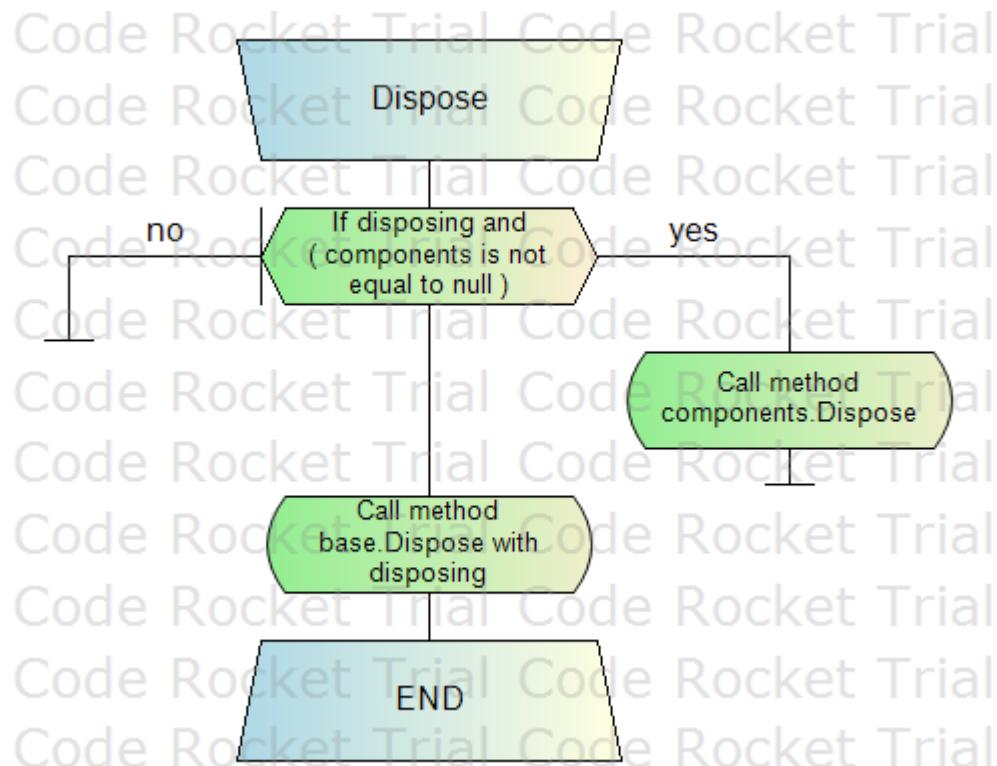
**Pseudocode**

If disposing and ( components is not equal to null )  
    Call method components.Dispose

EndIf

Call method base.Dispose with disposing

**Flowchart**



```

#region Windows Form Designer generated code
Procedure: InitializeComponent
private void InitializeComponent()
Required method for Designer support - do not modify
the contents of this method with the code editor.
Returns Void
Pseudocode
Create new System.ComponentModel.ComponentResourceManager
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Button
Create new System.Windows.Forms.Label
Create new System.Windows.Forms.Panel
Call method this.SuspendLayout
minimize
Set this.minimize.BackColor to System.Drawing.Color.Transparent
Set this.minimize.BackgroundImage
Set this.minimize.FlatAppearance.BorderSize to 0
Set this.minimize.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.minimize.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.minimize.Name to "minimize"
Create new System.Drawing.Size
Set this.minimize.TabIndex to 3
Set this.minimize.TabStop to false
Set this.minimize.UseVisualStyleBackColor to false
This.minimize.Click += new System.EventHandler(this.minimize_Click);
exitApplication
Set this.exitApplication.BackColor to System.Drawing.Color.Transparent
Set this.exitApplication.BackgroundImage
Set this.exitApplication.FlatAppearance.BorderSize to 0
Set this.exitApplication.FlatAppearance.MouseDownBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.Transparent
Set this.exitApplication.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Create new System.Drawing.Point
Set this.exitApplication.Name to "exitApplication"
Create new System.Drawing.Size
Set this.exitApplication.TabIndex to 2
Set this.exitApplication.TabStop to false
Set this.exitApplication.UseVisualStyleBackColor to false
This.exitApplication.Click += new System.EventHandler(this.exitApplication_Click);
nodesMenu
Set this.nodesMenu.BackColor to System.Drawing.Color.Transparent
Set this.nodesMenu.FlatAppearance.BorderSize to 0
Set this.nodesMenu.FlatAppearance.MouseOverBackColor to
System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ), ( ( int ) )
Set this.nodesMenu.FlatStyle to System.Windows.Forms.FlatStyle.Flat
Set this.nodesMenu.Image
Create new System.Drawing.Point

```

```

Set this.nodesMenu.Name to "nodesMenu"
Create new System.Drawing.Size
Set this.nodesMenu.TabIndex to 4
Set this.nodesMenu.TabStop to false
Set this.nodesMenu.UseVisualStyleBackColor to false
This.nodesMenu.Click += new System.EventHandler(this.nodesMenu_Click);
label1
Set this.label1.AutoSize to true
Set this.label1.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label1.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label1.Name to "label1"
Create new System.Drawing.Size
Set this.label1.TabIndex to 5
Set this.label1.Text to "Welcome To Sectrics, A Truss Analysis Program Made By Shaan
Khan"
label2
Set this.label2.AutoSize to true
Set this.label2.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label2.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label2.Name to "label2"
Create new System.Drawing.Size
Set this.label2.TabIndex to 6
Set this.label2.Text to "This program is designed to find the tensile/compressive
forces in each beam give" plus
label3
Set this.label3.AutoSize to true
Set this.label3.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label3.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label3.Name to "label3"
Create new System.Drawing.Size
Set this.label3.TabIndex to 7
Set this.label3.Text to "To use this program follow these steps:"
label4
Set this.label4.AutoSize to true
Set this.label4.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label4.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label4.Name to "label4"
Create new System.Drawing.Size
Set this.label4.TabIndex to 8
Set this.label4.Text to "1. Click on nodes and enter the nodes, in nodes enter the X
& Y locations of each" plus
label5
Set this.label5.AutoSize to true
Set this.label5.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label5.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label5.Name to "label5"
Create new System.Drawing.Size
Set this.label5.TabIndex to 9
Set this.label5.Text to resources.GetString with "label5.Text"
label6
Set this.label6.AutoSize to true
Set this.label6.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label6.ForeColor to System.Drawing.Color.White

```

```

Create new System.Drawing.Point
Set this.label6.Name to "label6"
Create new System.Drawing.Size
Set this.label6.TabIndex to 10
Set this.label6.Text to resources.GetString with "label6.Text"
label7
Set this.label7.AutoSize to true
Set this.label7.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label7.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label7.Name to "label7"
Create new System.Drawing.Size
Set this.label7.TabIndex to 11
Set this.label7.Text to resources.GetString with "label7.Text"
label8
Set this.label8.AutoSize to true
Set this.label8.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label8.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label8.Name to "label8"
Create new System.Drawing.Size
Set this.label8.TabIndex to 12
Set this.label8.Text to resources.GetString with "label8.Text"
label9
Set this.label9.AutoSize to true
Set this.label9.BackColor to System.Drawing.Color.Transparent
Create new System.Drawing.Font
Set this.label9.ForeColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.label9.Name to "label9"
Create new System.Drawing.Size
Set this.label9.TabIndex to 13
Set this.label9.Text to "6. Now you're done. Just press solve to view the
tensile/compressive forces in ea" plus
moveMenu
Set this.moveMenu.BackColor to System.Drawing.Color.White
Create new System.Drawing.Point
Set this.moveMenu.Name to "moveMenu"
Create new System.Drawing.Size
Set this.moveMenu.TabIndex to 41
This.moveMenu.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseDown);
This.moveMenu.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseMove);
This.moveMenu.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.moveMenu_MouseUp);
tutorialMenu
Set this.BackColor to System.Drawing.Color.FromArgb with ( ( int ) ), ( ( int ) ),
( ( int ) )
Set this.BackgroundImageLayout to System.Windows.Forms.ImageLayout.None
Create new System.Drawing.Size
Call method this.Controls.Add with this.moveMenu
Call method this.Controls.Add with this.label9
Call method this.Controls.Add with this.label8
Call method this.Controls.Add with this.label7
Call method this.Controls.Add with this.label6
Call method this.Controls.Add with this.label5
Call method this.Controls.Add with this.label4
Call method this.Controls.Add with this.label3
Call method this.Controls.Add with this.label2
Call method this.Controls.Add with this.label1
Call method this.Controls.Add with this.nodesMenu

```

```
Call method this.Controls.Add with this.minimize
Call method this.Controls.Add with this.exitApplication
Set this.DoubleBuffered to true
Set this.FormBorderStyle to System.Windows.Forms.FormBorderStyle.None
Set this.Icon
Set this.Name to "tutorialMenu"
Set this.StartPosition to System.Windows.Forms.FormStartPosition.CenterScreen
Set this.Text to "Sectrics - Truss Analysis Program"
This.Load += new System.EventHandler(this.nodes_Load);
Call method this.ResumeLayout with false
Call method this.PerformLayout
```

### Flowchart



```
#endregion  
Attribute: minimize  
  
private System.Windows.Forms.Button minimize;  
Attribute: exitApplication  
  
private System.Windows.Forms.Button exitApplication;  
Attribute: nodesMenu  
  
private System.Windows.Forms.Button nodesMenu;  
Attribute: label1  
  
private System.Windows.Forms.Label label1;  
Attribute: label2  
  
private System.Windows.Forms.Label label2;  
Attribute: label3  
  
private System.Windows.Forms.Label label3;  
Attribute: label4  
  
private System.Windows.Forms.Label label4;  
Attribute: label5  
  
private System.Windows.Forms.Label label5;  
Attribute: label6  
  
private System.Windows.Forms.Label label6;  
Attribute: label7  
  
private System.Windows.Forms.Label label7;  
Attribute: label8  
  
private System.Windows.Forms.Label label8;  
Attribute: label9  
  
private System.Windows.Forms.Label label9;  
Attribute: moveMenu  
  
private System.Windows.Forms.Panel moveMenu;
```

**Code File: Resources.Designer.cs**

---

**Namespace: Sectrics\_V2.Properties**

```
namespace Sectrics_V2.Properties
```

## **Class: Resources**

A strongly-typed resource class, for looking up localized strings, etc.

```
private static global::System.Resources.ResourceManager
```

**Attribute:** resourceMan

```
resourceMan;
```

```
private static global::System.Globalization.CultureInfo
```

**Attribute:** resourceCulture

```
resourceCulture;
```

**Procedure:** Constructor

```
[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance", "CA1811:AvoidUncalledPrivateCode")]
```

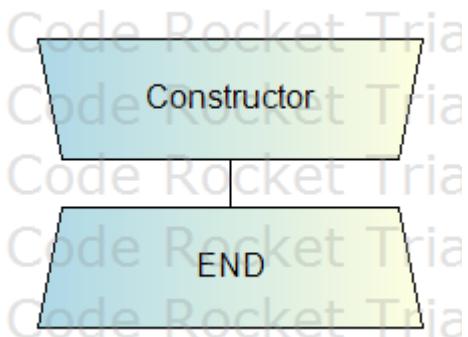
```
    internal Resources()
```

Returns Resources

**Pseudocode**

N/A

Flowchart



```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
```

```
    internal static global::System.Resources.ResourceManager
```

Returns the cached ResourceManager instance used by this class.

**Property: ResourceManager**

```
ResourceManager {  
    get {  
        if (object.ReferenceEquals(resourceMan, null)) {  
            global::System.Resources.ResourceManager temp =  
new  
global::System.Resources.ResourceManager("Sectrics_V2.Properties.Resou  
rces", typeof(Resources).Assembly);  
            resourceMan = temp;  
        }  
        return resourceMan;  
    }  
}
```

```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
```

```
    internal static global::System.Globalization.CultureInfo
```

Overrides the current thread's CurrentUICulture property for all  
resource lookups using this strongly typed resource class.

**Property: Culture**

```
Culture {  
    get {  
        return resourceCulture;  
    }  
    set {  
        resourceCulture = value;  
    }  
}
```

**Property: horizontalRoller**

```
internal static System.Drawing.Bitmap horizontalRoller {  
    get {  
        object obj =  
ResourceManager.GetObject("horizontalRoller", resourceCulture);  
        return ((System.Drawing.Bitmap)(obj));  
    }  
}
```

Looks up a localized resource of type System.Drawing.Bitmap.

**Property: horizontalSupport**

```

internal static System.Drawing.Bitmap horozontalSupport {
    get {
        object obj =
ResourceManager.GetObject("horozontalSupport", resourceCulture);
        return ((System.Drawing.Bitmap) (obj));
    }
}

```

Looks up a localized resource of type System.Drawing.Bitmap.

*Property: verticalRoller*

```

internal static System.Drawing.Bitmap verticalRoller {
    get {
        object obj =
ResourceManager.GetObject("verticalRoller", resourceCulture);
        return ((System.Drawing.Bitmap) (obj));
    }
}

```

Looks up a localized resource of type System.Drawing.Bitmap.

*Property: verticalSupport*

```

internal static System.Drawing.Bitmap verticalSupport {
    get {
        object obj =
ResourceManager.GetObject("verticalSupport", resourceCulture);
        return ((System.Drawing.Bitmap) (obj));
    }
}

```

Looks up a localized resource of type System.Drawing.Bitmap.

*Code File: Settings.Designer.cs*

*Namespace: Sectrics\_V2.Properties*

namespace Sectrics\_V2.Properties

*Class: Settings*

```

[global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.Vis
ualStudioEditors.SettingsDesigner.SettingsSingleFileGenerator",
"11.0.0.0")]

    internal sealed partial class Settings :
global::System.Configuration.ApplicationSettingsBase

```

*Attribute: defaultInstance*

```
private static Settings defaultInstance =
((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new Settings())));

```

**Property: Default**

```
public static Settings Default
{
    get
    {
        return defaultInstance;
    }
}
```