Real Time Memory Editor Development

Copyright PureIso 2013

Table of Contents

[1. Getting Started 2](#_Toc347232181)

[2. Using Peek Poker Interface 2](#_Toc347232182)

[2.1 API 2](#_Toc347232183)

[2.1.1 PeekPoker.Interface.AbstractPlugin 2](#_Toc347232184)

[2.2 Form Set-up 2](#_Toc347232185)

[2.3 Coding 3](#_Toc347232186)

[3. Links 5](#_Toc347232187)

[4. Thanks 5](#_Toc347232188)

# Getting Started

A real time editor is an application that allows you to edit memory in real time.

We will be using Fifa 13 for the following example.

# Using Peek Poker Interface

## API

The Application Programming interface contains details of the object classes available to use in PeekPoker Interface library.

### PeekPoker.Interface.AbstractPlugin

(Coming Soon)

## Form Set-up

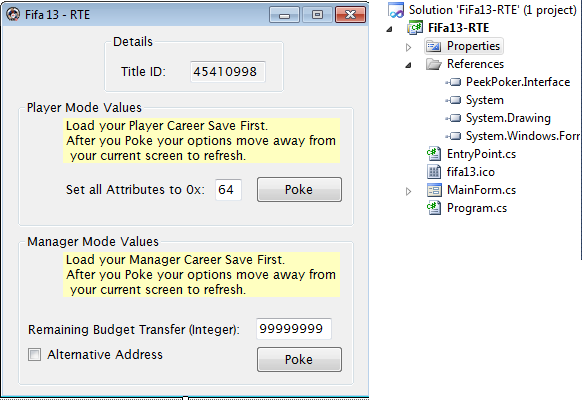
* Open Visual Studios and Start a new Windows Form Application Project

(As a convention)

* Rename your Form1 to MainForm
* Change MainForm Text -> \*\*\* - RTE (\*\*\* = Name of the game)
* FormBorderStyle -> FixedDialog
* MaximizeBox -> False
* Icon -> Browse for your Icon
* Font -> Lucida Sans, 9pt
* Add Reference -> PeekPoker.Interface
* Properties -> Target Framework -> .Net Framework 2.0 (Better compatibility)
* Properties -> Icon (Browse for your Icon)

(Checkout: http://www.coolutils.com/online/image-converter/)

* Add New Class
* Rename Class1 -> EntryPoint



## Coding – EntryPoint

Below is the full EntryPoint code with comments:

using PeekPoker.Interface;

using System.Windows.Forms;

//===============================================

//Author: PureIso

//Description: FIFA 13 RTE EntryPoint

//===============================================

namespace FiFa13\_RTE

{

//Entry Point inherits from PeekPoker.Interface.AbstractPlugin

public class EntryPoint : AbstractPlugin

{

/// <summary>

/// A Thread Safe way to show messagebox using the codes in

/// PeekPoker

/// </summary>

internal static ShowMessageBoxHandler ShowMessageBox;

/// <summary>

/// A Thread Safe way to update the progressbar using the codes in

/// PeekPoker

/// </summary>

internal static UpdateProgressBarHandler UpdateProgressBar;

/// <summary>

/// A Thread Safe way to set text in a textbox using the codes in

/// PeekPoker

/// </summary>

internal static SetTextBoxTextDelegateHandler SetText;

/// <summary>

/// A Thread Safe way to get text from a textbox using the codes in

/// PeekPoker

/// </summary>

internal static GetTextBoxTextHandler GetText;

/// <summary>

/// A Thread Safe way to enable or disable controls using the codes in

/// PeekPoker

/// </summary>

internal static EnableControlHandler EnableControl;

/// <summary>

/// The real time memory accessor

/// </summary>

internal static RealTimeMemory Rtm;

/// <summary>

/// Constructor

/// </summary>

public EntryPoint()

{

//Application Information

//If you don't fill out any of the information AbstractPlugin will use

//the default which is "Unavailable"

base.ApplicationName = "Fifa 13";

base.Author = "PureIso";

base.Description = "Fifa 13 RealTime Editor";

base.Version = "1.0.0.0";

//Get the proper Icon

MainForm form = new MainForm();

base.Icon = form.Icon;

//Clean up

form.Dispose();

form.Close();

}

/// <summary>

/// Overrides the abstract Display method - if not called an empty form

/// will be loaded

/// </summary>

/// <param name="parent">The PeekPoker will be the parent</param>

public override void Display(Form parent)

{

//Handlers

ShowMessageBox = base.APShowMessageBox;

UpdateProgressBar = base.APUpdateProgressBar;

GetText = base.APGetTextBoxText;

SetText = base.APSetTextBoxText;

EnableControl = base.APEnableControl;

//You can get the RTM being using in PeekPoker or

//you can initialize you own

Rtm = base.APRtm;

MainForm form = base.IsMdiChild

? new MainForm { MdiParent = parent }

: new MainForm();

form.Show();

}

}

}

## Coding – MainForm

Below is the full Main Form code with comments:

Note: I am not using any pointer.

using System;

using System.Windows.Forms;

//===============================================

//Author: PureIso

//Description: FIFA 13 RTE MainForm

//===============================================

namespace FiFa13\_RTE

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

//When you click the attribute Poke button

private void attributePokeButton\_Click(object sender, EventArgs e)

{

try

{

//Initialise the hexString

string hexString = "";

//Get the byte String

string byteHex = EntryPoint.GetText(attributeValueTextBox);

//We need 34 bytes of the same value as Hex String

for (int i = 0; i < 34; i++)

{

//Concat each byte string to the Hex String

//It's faster than Getting the textbox text

hexString += byteHex;

}

//Poke that Address C53D7729 with the hexString Value - New Save

//This address is only useful after loading your save - Attributes

EntryPoint.Rtm.Poke("C53D7729", hexString);

//This address is for Skill Moves Star Rating

EntryPoint.Rtm.Poke("C53D77A8", "04");

EntryPoint.ShowMessageBox("Poke Successful!", "FIFA 13 - RTE", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

EntryPoint.ShowMessageBox(ex.Message,"FIFA 13 - RTE",MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

//When you click the Budget Allocation Poke button

private void budgetPokeButton\_Click(object sender, EventArgs e)

{

try

{

string remaianingBudgetTransfer = uint.Parse(EntryPoint.GetText(remainingBudgetTextBox)).ToString("X");

if ((remaianingBudgetTransfer.Length/2)%2 != 0)

remaianingBudgetTransfer = "0" + remaianingBudgetTransfer;

//Poke that Address CD296230 with the hexString Value - New Save

EntryPoint.Rtm.Poke("CD296230", remaianingBudgetTransfer);

//Alternative - In case the first doesn't work

if (alternateCheckBox.Checked)

{

EntryPoint.Rtm.Poke("CD1C8F90", remaianingBudgetTransfer);

}

EntryPoint.ShowMessageBox("Poke Successful!", "FIFA 13 - RTE", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

EntryPoint.ShowMessageBox(ex.Message, "FIFA 13 - RTE", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

# Links

# Thanks