Georgia Tech Psychology Head Up Display Designer For MiniSim

Georgia Tech School of Psychology Tech Report GT-PSYC-TR-2015-03

Thomas M. Gable, Bruce N. Walker, Bhargav Rajendra, & Fang He

Thomas.Gable@gatech.edu, Bruce.Walker@psych.gatech.edu,

bhargav.r@gatech.edu, fhe36@gatech.edu

August 27, 2015

Sonification Lab
School of Psychology
Georgia Institute of Technology
654 Cherry Street
Atlanta, GA, USA
http://sonify.psych.gatech.edu
Phone: +1-404-894-8265

0. Abstract

As in-vehicle systems use head up displays (HUDs) more often, researchers need to be able to investigate the correct way of displaying information on these interfaces. In order to do this, researchers need easily modifiable head up display interfaces to test against each other, something most easily done digitally in a simulated environment. In an effort to allow for this we have developed a HUD designer that allows data to put input to the system and then used to display any created HUD onto either MiniSim visuals directly or a secondary HUD display. This document was written as a guide for those who are interested in using the Georgia Tech HUD Designer software to inform them of what it requires, inform users of what the system is doing, and then linking them to the software to use to make the connections discussed. Please refer to the included software license. If you use the system for studies please simply cite this tech report.

1. Overview of the Designer

The GT Head Up Display (HUD) Designer is an independent and third-party module developed at Georgia Tech, which can serve both as a HUD module to run on top of NADS MiniSim and also a HUD element design/prototyping tool. It provides an intuitive approach to iteratively modifying a head-up display's design and rapidly testing them in the NADS MiniSim. The system was initially built to use data from the Georgia Tech STING module [1] but HUD Designer can also use any other data from other sources to drive the displays.

2. Use of the HUD Designer

In an effort to give other researchers this tool we are making the software open source for those who a using it for research and non-profit purposes. We undertook the development of this stand-alone software with the assistance of those at National Advanced Driving Simulator but this is not part of their software. Please also see the license that accompanies this software, especially with respect to the as-is nature of given source code. We recommend having this software installed on a separate machine, or at least in a secondary (copied) folder directory with a copy of the MiniSim and its required files.

3. Requirements

Note: HUD Designer can be used with any data source, but this tech report focuses on how the HUD Designer can work with the STING software [1].

To use HUD Designer, make sure to install the following software dependencies.

1) Microsoft .NET Framework v4
2) NADS MiniSim v2

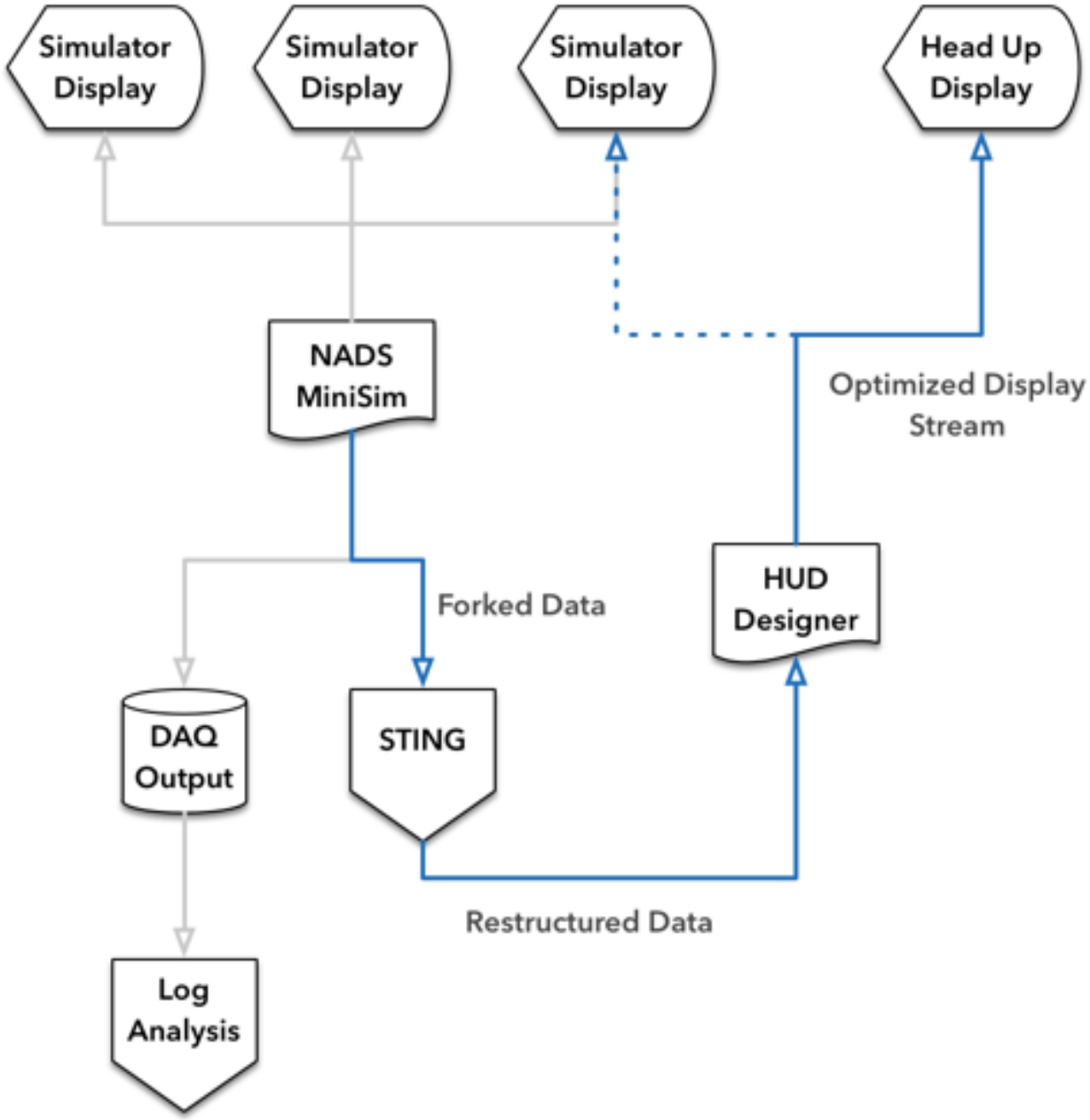
4. Dependencies

Modifying the HUD Designer requires the following dependencies to be installed.

1) Microsoft Visual Studio 2010 or above.

2) Microsoft Blend.

5. Architecture Diagram



6. Installing and Running the HUD Designer

To install the HUD Designer software one would do the following:

1) Check the code out from the repository or download and unzip the package.
2) The HUD Designer requires Microsoft .NET Framework v4. Make sure to install that before running the executable.
3) The HUD Designer has no dependencies on MiniSim, and thus can reside in an independent folder (or even on a separate machine). Open *HUDDesigner.exe* to run the application.

7. Modifying the HUD Designer

1) Check out code from the HUDDesigner repository or download the package.
2) Open the existing *csproj* on Visual Studio 2010 or above.
3) This project relies on the Windows Presentation Framework that follows an MVC approach to handling code. The *XAML* file for this project manages the user interface and the view, while the *XAML.CS* file manages the core operations.
4) To modify or add a HUD element, prepare a vector image of the element either on Adobe Illustrator or a similar vector generation tool.
5) Export the vector design as an *SVG* file, which we will use as an intermediate step.
6) Open the *SVG* file in Microsoft Blend, and convert it to a *XAML* design.
7) This conversion process automatically provides the code structure for Visual Studio.
8) Create a new *XAML* file under project resources and paste the content of exported *XAML*.
9) Modify the *XAML.CS* file to handle any interactions you require from the HUD element.
10) To utilize data from any data source, configuration parameters are provided as variables in the *Main* methods code block of the HUD Designer.

11) Update the ReceiveMessage() method of the HUD Designer to update data you receive from the data source in a local variable. The example below shows how one can read the speed variable from the STING software. Use the speed variable in the code block for the newly created HUD element to provide conditional interactions.

```
private void ReceiveMessage()
    {
        while (true)
        {
            IPEndPoint remoteIPEndPoint = new IPEndPoint(IPAddress.Any, port);
            byte[] content = udpClient.Receive(ref remoteIPEndPoint);

            if (content.Length > 0)
            {
                string speed = Encoding.ASCII.GetString(content);
                this.Dispatcher.Invoke(myDelegate, new object[] { speed });
            }
        }
    }
```

8. Questions and Contact

If there are questions or concerns about this system please contact the authors or Professor Bruce N. Walker at the Sonification Lab in the School of Psychology at the Georgia Institute of Technology. Other tech reports from the GT Sonification Lab also exist in the Georgia Tech SmartTech repository (https://smartech.gatech.edu/) regarding the MiniSim. This includes the Simulator Telemetry INteGration (STING) module, which we developed as a sub-module of the NADS MiniSim, to send out data from the simulator (GT Psychology Tech Report GT-PSYC-TR-2015-02). We also have a MiniSim training module (GT Psychology Tech Report GT-

PSYC-TR-2015-01), and a simulator sickness screening protocol (GT Psychology Tech Report GT-PSYC-TR-2013-01) if researchers are interested in any of those documents/resources.

9. References

[1] Gable, T. M, Walker, B. N., & Rajendra, B. (2015). Georgia Tech Simulator Telemetry Integration (STING) Module for NADS MiniSim. Georgia Tech School of Psychology Tech Report GT-PSYC-TR-2015-02.