Hardware Shopper Android Network Application

Thomas Yu| ECE 4564 – Fall 2013 | Virginia Tech | thomaswy@vt.edu

Overview

This application was created in eclipse using the Android Development Bundle. As the name suggests, the purpose of this application is to facilitate shopping for computer hardware components. Currently, there are 2 sites that are searched: newegg.com and frys.com. The application will search these sites for a specified hardware component and return the first 8 search results from each site.

Note that search parameters must be relatively precise due to the varying layouts of each website. For example, a search parameter of "graphics card" may fail to return any search results for one or both websites. However, a more specific search of "gtx 660" (for the Geforce GTX 660 graphics card) will return valid results.

Purpose

This application uses HTTP GET requests to query the newegg.com and frys.com servers for information. It does these requests asynchronously on different threads. GET parameters are used to relay information to the web server. There is no specified order in which data is requested or must finish first. However, replies must be received from BOTH web servers before any data is processed. Once the HTML replies are received, the raw data can be processed to yield the first 8 search results from each website. The replies can also be discarded without processing by simply searching again. During each part of the process, messages are displayed that inform the user of the current state of the application.

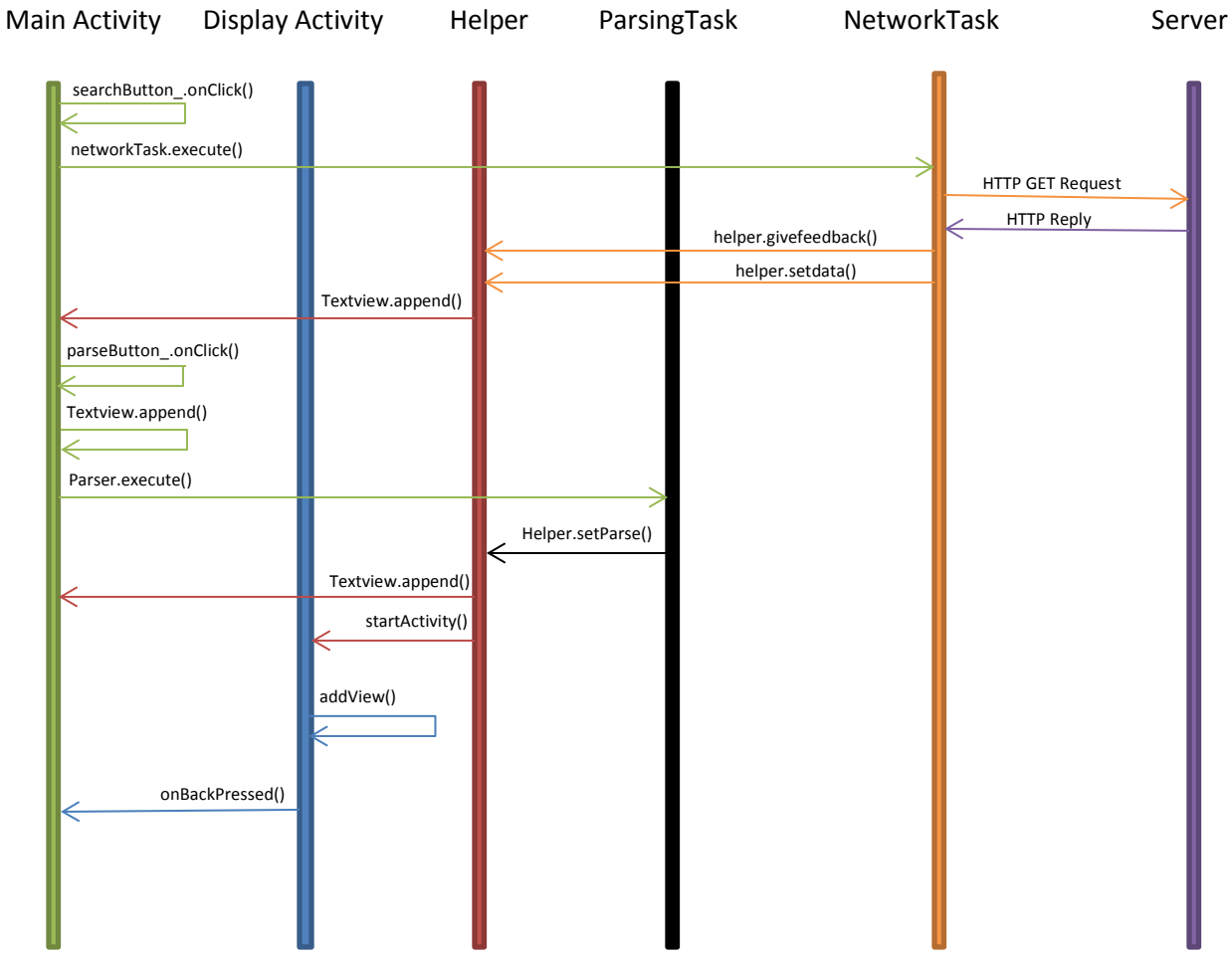
High Level Source Design

This application is comprised of 5 different classes, two of which are Activities. Each class represents a specific task which must be done: Networking, Parsing, Main GUI, Display GUI, and a helper to facilitate threading. Main GUI and Display GUI are activities.

The Main activity serves as the startup screen and the primary source of user interaction. The Main activity takes user input and invokes two asynchronous network tasks used for querying the web servers. The network tasks are run and pass replies to the helper task which will keep track of received information as well as the state of the application. Upon finishing any required network traffic, the networking tasks exit and can be invoked again if necessary.

Once the command is given to process data, the Main activity will invoke two asynchronous instances of the parsing task. The Main activity retrieves the raw data from the helper task and passes it to the parsing tasks. Upon completion, the parsing tasks give the processed information to the helper. The helper then invokes a new instance of the Display Activity and sends processed data to it through the usage of an intent. The Display activity simply displays the processed information using TextViews within a TableLayout.

Sequence Diagram



**There are actually 2 separate instances of both NetworkTask and ParsingTask that run asynchronously

Class Diagram