



PureWeb® STK 4.0

Quick Start Guide: Android
Client

The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Calgary Scientific Inc.

Copyright © 2013 Calgary Scientific Inc. All rights reserved.

About Calgary Scientific

Calgary Scientific Inc. is dedicated to providing advanced visualization, web enablement, and mobility enhancement solutions to industries looking for secure access and use of their data or graphics intensive applications, while using their existing systems. Visit www.calgaryscientific.com for more information.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Calgary Scientific Inc. cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Calgary Scientific products.

Trademarks

© 2013 Calgary Scientific Inc., ResolutionMD, PureWeb and the Calgary Scientific logo are trademarks and/or registered trademarks of Calgary Scientific Inc. or its subsidiaries. Any third-party company names and products are for identification purposes only and may be trademarks of their respective owners.

Released by

Calgary Scientific Inc. www.calgaryscientific.com.

Document Version: PW4.0_AndroidClient_QSG_07-2013_v1.000.00

Table of Contents

Preface	5
Chapter 1	Introduction.....	6
Chapter 2	Installing PureWeb	8
	Prerequisites.....	8
	Setting the Environment Variables	9
	JAVA_HOME.....	9
	ANT_HOME	9
	Path	10
	Installing the STK	10
Chapter 3	Building the Applications	11
	Creating an Android Virtual Device	11
	Building with Ant.....	12
	Building with Eclipse.....	13
	Creating a Classpath Variable	13
	Importing the Sample Application	14
	Creating a Run Target.....	14
	Configuring the Application in the Device.....	15
	Launching the Application	15
Chapter 4	Sample Applications Overview	16
	Scribble.....	16
	Drawing Lines	16
	Erasing Lines	17
	Changing Pen Color	17
	Sharing the Application.....	18
	Closing Scribble.....	19
	Asteroids.....	19
	Single-Player Mode	19
	Two-Player Mode.....	20

Chapter 5	Exploring the Code.....	21
	General Code Pointers	21
	Common Library Code Description.....	22
	Scribble Code Pointers.....	22
	Asteroids Code Pointers.....	23
	Android Classes	24
	Debugging the Code.....	25

Preface

Welcome to the *PureWeb STK 4.0 Quick Start Guide: Android Client*, a document that provides instructions to install the PureWeb® STK, and to build, run and troubleshoot sample PureWeb Android client applications.

Intended Audience

This document is primarily intended for software developers who plan to install and use the STK to PureWeb enable an Android client. It has been written with the assumption that you have:

- an intermediate knowledge of Java,
- a working knowledge of Eclipse,
- a degree of competency in Android applications and their deployment.

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to techpubs@calgaryscientific.com.

Contacting Calgary Scientific Support

Use one of the methods in the table below to contact Calgary Scientific support.

Web Site	E-Mail
support.getpureweb.com	support@getpureweb.com

1 Introduction

PureWeb® is a platform that enables the rapid transformation of enterprise software into cloud-ready, web and mobile applications.

The PureWeb STK (software transformation kit) is a set of tools that provides developers everything they need to implement the PureWeb solution in their own applications, in particular service and client APIs, and sample applications in each supported programming language.

The sample applications illustrate key concepts of PureWeb enablement; they can be used by developers as a starting point to create their own PureWeb-enabled applications.

The PureWeb *Quick Start Guide* series is intended to get developers up and running as quickly as possible, using these sample applications as models.

As its name indicates, the current document, *PureWeb STK Quick Start Guide: Android Client*, focuses on the Android client. The guide provides instructions to install the PureWeb STK, and to build, run and troubleshoot a sample client.

Two Android sample clients are available:

- Scribble, which allows users to draw lines on a canvas, change line color, and erase lines.
- Asteroids, a PureWeb-enabled implementation of the classic video game with a twist -- it supports both single-player and two-player modes; each player independently controls his or her own ship.

Before working with the sample Android client, you will need to build the sample service; instructions for doing in so can be found in the *Quick Start Guide* for each service development platform.

Recommended Reading

This *Quick Start Guide* is intended as an introduction and does not provide in-depth information; it should be used in conjunction with the other guides of the PureWeb STK documentation suite, in particular:

- *Installation Guide*
- *Developer's Guide*
- *Server Administration Guide*
- Android client API reference material

To access these references in a Windows operating system, press the `Windows` key on the keyboard to display the **Start** menu (**Start** screen in Windows 8), then type "pureweb" as the search term.

2 Installing PureWeb

In this chapter, you will find prerequisite information as well as instructions to install the PureWeb STK, and set up the environment variables.

For more detailed instructions, such as how to handle a side-by-side installation of the STK, please refer to the complete *PureWeb STK Installation Guide*.

Prerequisites

To install the PureWeb STK, you will need:

- Java JDK 1.7
- C++ Runtime, either of: 2008 SP1 x86/x64 or 2010 SP1 ATL x86/x64; the 2010 version can be downloaded from the following location:
www.microsoft.com/en-us/download/details.aspx?id=5555
- Windows 64-bit operating system, either of: Windows 7, Windows 8, Windows Server 2008 R2, or Windows Server 2012

Note: Although only 64-bit platforms are supported for application development, you can target the applications developed using the PureWeb STK to deploy on either a 32-bit or a 64-bit platform.

In addition, to build and run the Android sample client application, you will need:

- Android 4.0 (for your own client applications, you can use a more recent version)
- Apache Ant 1.8.1
- a Java IDE; we recommend Eclipse Juno 4.2 with:
 - ADT (Android Developer Tools) plugin
 - Android SDK Level 14, which includes the Android Debug Bridge tool (for your own client applications, you can use a more recent version)

We also recommend that you read and follow the instructions provided by Google for getting started with Android development at

<http://developer.android.com/sdk/index.html>

Setting the Environment Variables

The PureWeb server requires the `JAVA_HOME` environment variable to be pointing to the correct instance of the JDK. If building with Ant, you will also need to set the `ANT_HOME` variable. In addition, the `Path` variable must include the path to JDK 1.7, Ant 1.8.1, as well as the path to your Android tools and Android platform tools directories.

On a Windows operating system, to access the Environment Variables dialog:

1. Press the `Windows` key on your keyboard to display the **Start** menu (or **Start** screen in Windows 8).
2. Type “environment variables” to search that string.
3. Select **Edit the System Environment Variables** from the search results. This will display the System Properties dialog box, with the Advanced tab selected.
4. Click the **Environment Variables** button.

JAVA_HOME

Check that you have a `JAVA_HOME` entry under System variables:

- If the entry is there and the path points to version 1.7 of the JDK, skip this section and check your `ANT_HOME` variable, as described in next section.
- If the entry is there, but the path points to the wrong version of the JDK: click the **Edit** button, change the path in the **Variable value** field and click **OK**.
- If the entry is missing:
 - a. Click on the **New** button.
 - b. Type `JAVA_HOME` in the **Variable name** field.
 - c. Enter the path to JDK 1.7 in the **Variable value** field (for example, the default for update 6 is `C:\Program Files\Java\jdk1.7.0_6`)
 - d. Click the **OK** button to save your changes.

ANT_HOME

Check that you have a `ANT_HOME` entry under System variables:

- If the entry is there and the path points to your instance of Ant 1.8.1, skip this section and check your `Path` variable, as described in next section.
 - If the entry is there, but the path points to the wrong version of Ant: click the **Edit** button, change the path in the **Variable value** field and click **OK**.
 - If the entry is missing:
 - a. Click on the **New** button.
 - b. Type `ANT_HOME` in the **Variable name** field.
 - c. Enter the path to Ant 1.8.1 in the **Variable value** field (default is `C:\Program Files (x86)\apache-ant-1.8.1`).
5. Click the **OK** button to save your changes

Path

If your `Path` system variable is missing the path to the JDK, Ant, Android tools, or Android platform tools:

1. In the list of system variables, select the `Path` variable.
2. Click on the **Edit...** button.
3. In the **Variable value** string, add the following paths if they are missing:
 - Path to `JAVA_HOME` (`%JAVA_HOME%\bin;`)
 - Path to `ANT_HOME` (`%ANT_HOME%\bin;`)
 - Path to your Android tools, for example:
`C:\sdk\tools;`
 - Path to your Android platform tools, for example:
`C:\sdk\platform-tools;`
4. Click the **OK** button to save your changes

Note: Every entry in the `Path` system variable must be separated by a semi-colon.

After changing environment variables, if you had a console window open, you will need to restart it for these changes to take effect.

Installing the STK

If you already have PureWeb installed, uninstall it first.

To install the PureWeb STK:

1. Double-click on its executable file, `PureWeb_Win_Setup-[version].exe`, and follow the instructions in the wizard. The installation directory path cannot contain any spaces.
2. Install your PureWeb license, a `.lic` file which you will have received from Calgary Scientific separately from the STK:
 - a. Navigate to `[Installed_location]\Server\conf\`.
 - b. Place the `.lic` license file into that folder.
 - c. Double-click on the **Start PureWeb** desktop icon to start the server.
3. Check that the following environment variables are set correctly:
 - `PUREWEB_HOME:`
`[Installed_location]\Server`
 - `PUREWEB_LIBS:`
`[Installed_location]\SDK\Redistributable\Libs`

3 Building the Applications

This chapter describes how to build and run the sample clients for Android.

Note: Before completing the procedures in this section, you must first build a sample service, as described in the *Quick Start Guide* for each service development platform.

You can choose to build the application using either the Apache Ant command-line interface or the Eclipse IDE.

Note: If you switch between development using Eclipse and Apache Ant, you may need to remove the application from the device or emulator using the built-in Application Management Tool.

For more information about development outside of Eclipse, see <http://developer.android.com/tools/projects/projects-cmdline.html>

Creating an Android Virtual Device

If you do not have an Android device and plan to work with an Android Virtual Device (AVD) instead, you must create that emulator first before completing the other procedures in this chapter.

1. Open Eclipse.
2. Select **Window | Android Virtual Device Manager** from the main menu.
3. Click the **New** button.
4. Enter a device name in the Name field.
5. Select any device of your choice from the **Device** drop-down list.
6. Select **Android 4.0 API Level 14** from the **Target** drop-down list.
7. If you have several options in the **CPU/ABI** field, select one.
8. Click the **OK** button to create the AVD.
9. Close AVD Manager.

Building with Ant

This is a two-fold process which involves first performing common steps for both projects (Scribble and Asteroids), then building the selected application.

Common Procedure

1. Open a console (command prompt) window.
2. Change directory to the following location:
`[Installed_location]\SDK\Samples\Android`
3. Check which targets you have available by using the following command:
`android.bat list targets`
4. Set up the path to point to the Android SDK tools directory, replacing the [number] placeholder with the target id associated with Android-14 (in your own application, you can use any SDK version 14 and above, but for the sample application, we recommend using 14); make a note of this id, as you will need it again in the next procedure, when you actually build the application:
`android update project --target [number] --path ./pureweb`
5. Copy the following .jar files from:
`[Installed_location]\SDK\Redistributable\Libs\Android`
to:
`[Installed_location]\SDK\Samples\Android\pureweb\libs:`
 - `CSI.PureWeb.Android.Client.jar`
 - `CSI.PureWeb.Client.jar`
 - `jdom.jar`
 - `slf4j-android-1.5.8.jar`

Application Build Procedure

1. Launch the PureWeb server using the **Start PureWeb** desktop icon.
2. Open the server interface in a browser (`http://localhost:8080`) and log in using `admin` in both the name and password fields.
3. Ensure that your command prompt is still open and points to the following directory:
`[Installed_location]\SDK\Samples\Android`
4. Set up the application project path to point to the Android SDK tools directory, replacing the [number] placeholder with the target id associated with Android-14:
`android update project --target [number] --path ./scribble`
or
`android update project --target [number] --path ./asteroids`

5. Ensure that either a device is connected or an emulator is running. If not, to run the emulator you created in “Creating an Android Virtual Device” on [page 11](#):
 - a. Open the Android AVD Manager in Eclipse (**Window | Android Virtual Device Manager**).
 - b. Select the name of the device.
 - c. Click the **Start** button.
 - d. Check the option **Scale display to real size** (optional).
 - e. Click the **Launch** button.
6. Change the directory to the sample application you are working with:
`[Installed_location]\SDK\Samples\Android\Scribble`
or
`[Installed_location]\SDK\Samples\Android\Asteroids`
7. Type the following command to create and install a debug build:
`ant debug install`

Next Step: configure the sample application in the emulator, as described on [page 15](#).

Building with Eclipse

Follow the procedures below to build either Scribble or Asteroids.

Creating a Classpath Variable

1. Open Eclipse.
2. Select **Window | Preferences** from the main menu.
3. Navigate to **Java | Build Path | Classpath Variables** to display the Classpath Variables dialog box.
4. Click the **New** button to open the New Variable Entry dialog.
5. Enter `PUREWEB_SDK` in the **Name** field.
6. In the **Path** field, enter:
`[Installed_location]\SDK\Redistributable\Libs\Android`
7. Click **OK** to close the New Variable Entry dialog.
8. Click **OK** to close the Preferences dialog.

Importing the Sample Application

1. Select **File | Import** from the Eclipse main menu.
2. Choose **General | Existing Projects Into Workspace** and click **Next**.
3. Choose the **Select root directory** option.
4. Click the **Browse** button and navigate to:
`[Installed_location]\SDK\Samples\Android`
5. Click the **OK** button.
6. In the **Projects** section, check the Asteroids, PureWeb, and Scribble projects.
7. Click the **Finish** button to import the project into your workspace.

Eclipse will build the project automatically. You can always rebuild the project by selecting **Project | Build All** from the Eclipse main menu.

Note: If build errors occur, select **Project | Clean** from the main menu. If the errors persist, restart Eclipse, and select **Project | Clean** again.

Creating a Run Target

1. Launch the PureWeb server with the **Start PureWeb** desktop icon.
2. Select **Run | Run Configurations** from the Eclipse menu.
3. Select **Android Application** and click the **New** icon to display the Run Configuration dialog.
4. In the **Name** field of the Android tab, Enter:
 - For Scribble: `ScribbleApp`.
 - For Asteroids: `AsteroidsApp`.
5. Click the **Browse** button to select your project (Scribble or Asteroids), then click the **OK** button to close the Project Selection window.
6. Choose the **Launch Default Activity** option.
7. Go to the **Target** tab.
8. Choose the **Always prompt to pick device** option in the Deployment Target Selection Mode section. This allows you to switch between the simulator and the actual device.
9. Click the **Run** button.
10. In the Android Device Chooser dialog box, select **Choose a running Android device** if you have one, otherwise:
 - a. Select **Launch a New Android Virtual Device**.
 - b. Highlight the name of the device you created earlier.
 - c. Click the **Start** button to display the simulator's home screen.
 - d. Click the **Launch** button to display the simulator's home screen.

Next Step: configure the sample application in the emulator, as described below.

Configuring the Application in the Device

1. Launch the PureWeb server by double-clicking the **Start PureWeb** desktop icon. The sample application icon will appear on the menu page of the device.
2. Drag the sample application icon to the Home page of the device.



3. Click the sample application icon to display the application login page.
4. Click the **Menu** button on the left side of the page.
5. Select the Settings option to display the **Settings** page. From here you can:
 - ♦ set the `ServerUrl` for the service application,
 - ♦ set the username and password for the server,
 - ♦ ensure security while collaborating.
6. Select **Server URL** and change `hostname` to the IP of your server.
7. Click **OK** and press the Back button to return to the login page.

Launching the Application

At this point, you should have started the Android device or the emulator, and configured the application.

1. Click the PureWeb icon to launch the application to display the Login page.
2. The username and password field should already contain the login credentials; if they do not, enter `admin` in both the **Username** and **Password** fields.
3. Click **Connect** to launch the application.

For more information about running Android applications, see developer.android.com/guide/developing/device.html

4 Sample Applications Overview

This chapter briefly describes the functionality available in each of the two sample applications, Scribble and Asteroids.

In the procedures in this chapter, “Click” should be replaced by a finger touch if working directly on a device instead of a simulator.

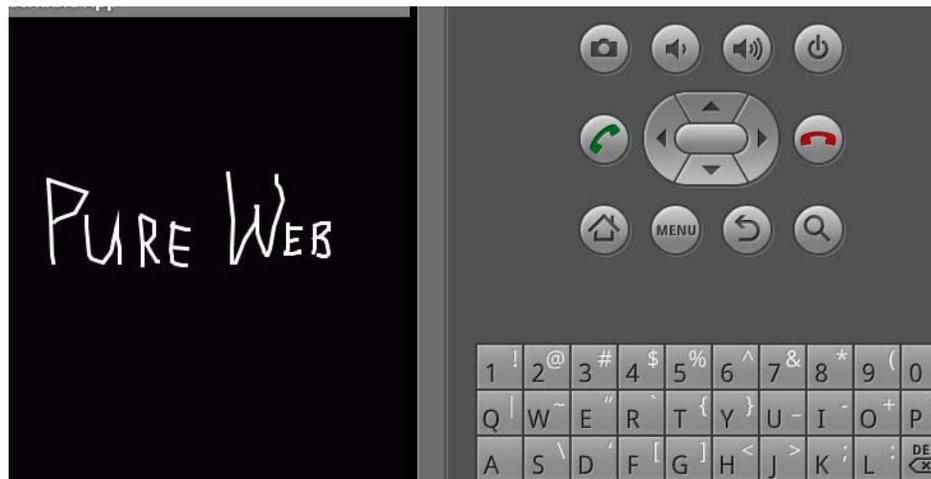
Scribble

The Scribble sample application is a simple canvas that users can use to draw lines, change line color and erase lines. You can also share the application with another user.

Drawing Lines

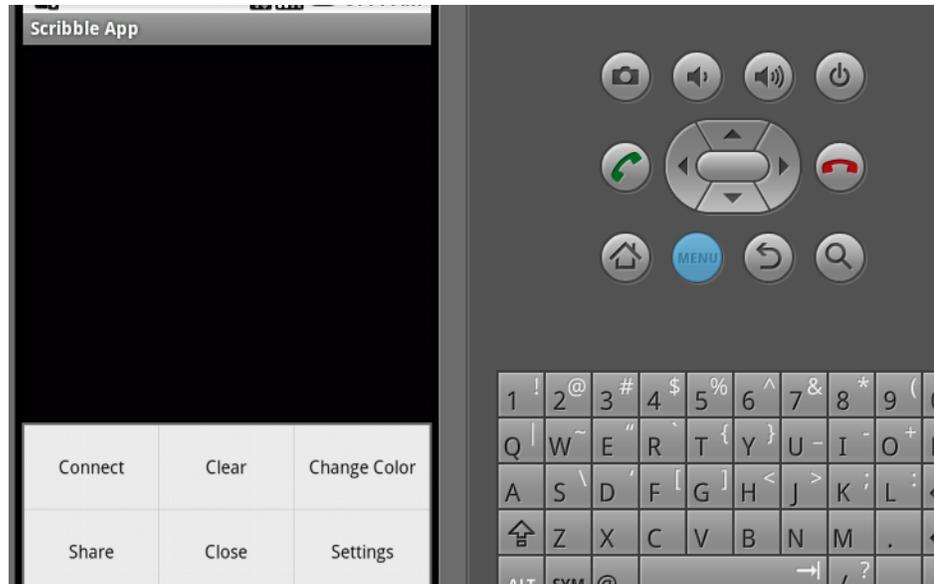
When you first launch Scribble, it displays a blank canvas.

You can draw lines on the canvas by left-clicking the mouse and dragging.



Erasing Lines

Display the menu by clicking on the **Menu** button, either through the emulator control panel or on the device itself.



Erase the lines on the canvas by clicking the **Clear** menu option.

Changing Pen Color

Click the **Change Color** menu option to display the Select a pen color dialog.



Click the drop down list arrow to display the color selection list.



Click the option button of your chosen color to return to the Select a pen color dialog.

Click the **Ok** menu option to commit your change or the **Cancel** menu option to keep your current color.

Sharing the Application

Go to the Android main menu and select the **Share** menu option to open an email application. The body of the email will contain a URL; send this URL to another user.

Second User:

Open the received message and click on the URL to display the PureWeb Collaboration Login page in your browser.

Select your preferred client from the drop down box, enter `Scientific` as your password and click the **Sign In** button to join the Scribble client session.

Closing Scribble

Close the Scribble client by clicking on the Close menu option.

To properly disconnect the client from the server, always close your application using the Home or Back buttons, or the menu close option. Closing the application by any other means will leave an orphan process running on the server.

The process will eventually terminate when the lack of client activity is detected and timeout is exceeded (timeout value is configurable).

Asteroids

The Asteroids sample application is a PureWeb-enabled implementation of the classic video game with a twist -- it supports both single-player and two-player modes; each player independently controls his or her own ship.



Single-Player Mode

- Press any key to start the game.
- Use the up/down arrow keys (or buttons on an actual device) to move forward/reverse.
- Use the left/right arrow keys (or buttons on an actual device) to change your heading.
- Press the red button icon to fire torpedoes.
- Activate shields by pressing the shield icon. Each player gets 5 seconds of shield time per game. Note that in two-player mode, shields are also enabled.
- Collide with an asteroid with your shields up to destroy the asteroid and collect the points.
- Display the client menu by clicking on the **Menu** button either through the emulator control panel, or on the device itself. From there, you can:
 - Share the application by clicking on the **Share** menu option (see Sharing the Application, on [page 18](#)).
 - Close the application by clicking on the **Close** menu option.

To properly disconnect the client from the server, always close your application using the Home or Back buttons, or the menu Close option. Otherwise an orphan process will be left running on the server.

The process will eventually terminate when the lack of client activity is detected and timeout is exceeded (timeout value is configurable).

Two-Player Mode

Instructions on how to share the application with another user are provided in the section [Sharing the Application](#), on [page 18](#).

Once the application is shared, additional two-player mode functionality includes:

- Collide with an unshielded opponent with your shields up to destroy their ship and collect all of their points.
- Collide with a shielded opponent with your shields up to create an elastic collision.
- Shoot and kill an unshielded opponent to collect all his or her points.

5 Exploring the Code

One of the main purposes of including sample applications with the STK is for developers who are new to PureWeb to use these samples as a starting point for coding and to experiment with them.

Therefore, don't be shy about editing the code and trying new things. That's the best way to learn, and you'll see how quickly you'll become proficient with PureWeb.

This chapter gives some pointers about the code in the sample Android client; however, it assumes you are familiar PureWeb concepts such as views, state, commands and event handlers, which are described in the *Developer's Guide*.

For an overview of the service-side code, please refer to the Quick Start Guide for the Java service.

General Code Pointers

To inspect the application code described in this section, navigate to the following directory:

```
[Installed_location]\SDK\Samples\Android
```

There is one subfolder for Scribble, one for Asteroids, and a `pureweb` subfolder for the code common to both applications.

Refer to the diagram “Android Classes” on [page 24](#) for a summary of the Android client classes.

There are only two public classes specific to the Android platform:

- `pureweb.android.client.AndroidUiDispatcher`—provides the underlying PureWeb APIs with a way to dispatch commands and application state changes to the user interface (UI) thread.
- `pureweb.android.client.ui.View`—is the only UI widget provided by the PureWeb STK. `View` extends from an `android.view.View` to provide the main rendering interface.

Common Library Code Description

The `PureWebActivity` base class provides functionality common to both the Scribble and Asteroids samples, including the ability to display a login view or a redirect URL, connect/disconnect from a PureWeb service application, and display common error dialogs.

The `onCreate()` method is called by the `ScribbleAppActivity.onCreate()` method. It creates the Android UI dispatcher, and PureWeb Framework instance. It also sets up a stall changed handler to display a connection stalled warning dialog.

The `ScribbleAppActivity.onCreate()` method also calls the `PureWebActivity.connect()` method. The method will determine where a share URL or a login view should be displayed.

The `onConnected()` method connects to the service app, and when the connection is successfully established, it returns to the `ScribbleAppActivity.onConnected()` method, so that it can display its main content view.

When you chose the Share menu option in the Android client application, the `ShareRequestCompleted` class creates the email form that contains the share URL.

The `disconnect()` method is called by the `ScribbleAppActivity.onDestroy()` method. It disconnects the client from the service application.

Scribble Code Pointers

The code described in this section can be found in the `ScribbleAppActivity.java` file.

onCreate

The `onCreate` method of the `ScribbleAppActivity` class calls the `onCreate()` method of the `PureWebActivity` base class. It also calls `PureWebActivity.connect()`, which determines if the redirect URL or the login view should be displayed. Finally, it registers a value changed handler to listen for color changes from a collaborator.

onOptionsItemSelected

This method contains a switch statement with six possible cases:

- `case R.id.clear`—erases the canvas and refreshes the view.
- `case R.id.close`—closes the application.
- `case R.id.change_color`—displays the color change dialog. The color change dialog creates a drop down box containing color choices, an Ok menu option, and a Cancel menu option. The Ok menu option sets the pen to the selected color.
- `case R.id.settings`—displays the setting dialog. The setting dialog allows you to enable security, set the server URL, the username, and the password.
- `case R.id.connect`—calls the connect method.
- `case R.id.share`—creates an email containing the share URL.

We recommend subclassing the View class to best translate touch and gesture events into commands for your back-end process. By default, we provide a mapping of touches to mouse events, where a single finger drag is mapped to left-mouse button + move.

Asteroids Code Pointers

The code described in this section can be found in the `AsteroidsAppActivity.java` file.

onCreate

The `onCreate` method of the `AsteroidsAppActivity` class calls the `onCreate()` method of the `PureWebActivity` base class. It also calls `PureWebActivity.connect()` which determines if the redirect URL or the login view should be displayed.

onOptionsItemSelected

This method contains an if/else statement with four possible results:

- `case R.id.close`—closes the application.
- `case R.id.settings`—displays the setting dialog. The setting dialog allows you to enable security, set the `serverUrl`, the User Name, and the Password.
- `case R.id.connect`—calls the connect method.
- `case R.id.share`—creates an email containing the share URL.

onTouch

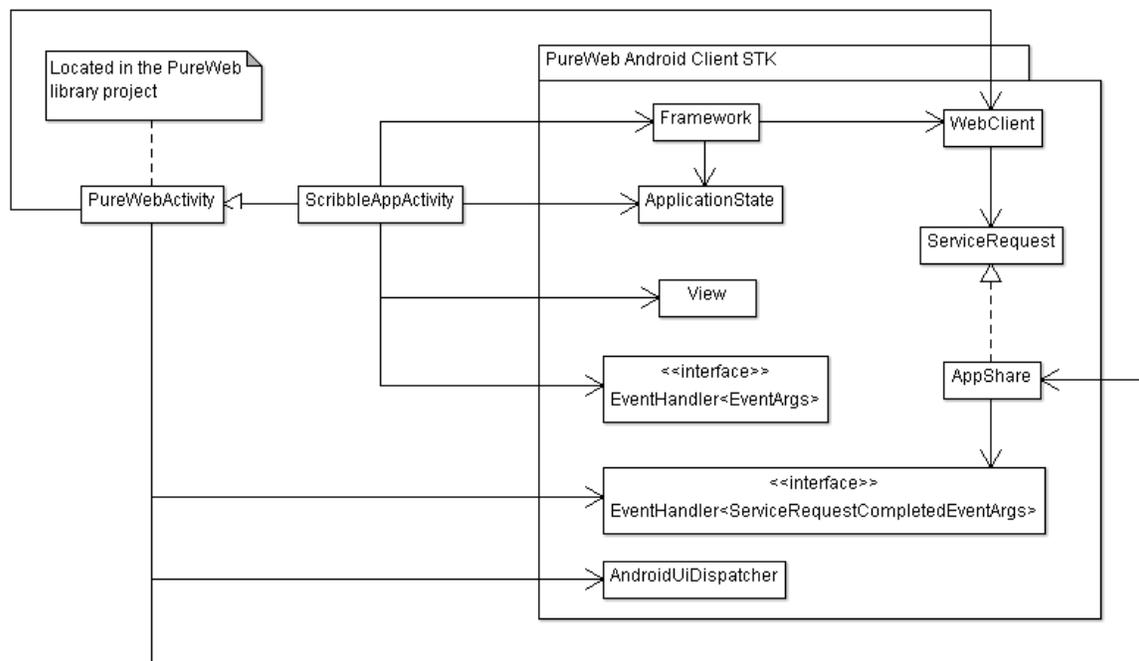
This method handles game button touches. It translates the button presses/releases into keyboard events. This method calls the `queueKeyboardEvent` which is described below.

queueKeyboardEvent

The method creates a PureWeb input command for the keyboard events that correspond to the game buttons. It uses a `switch` statement with a case for each of the game buttons (fire, shields, forward, reverse, left, and right). The information is put into a queue to be sent to the Asteroids service application.

Android Classes

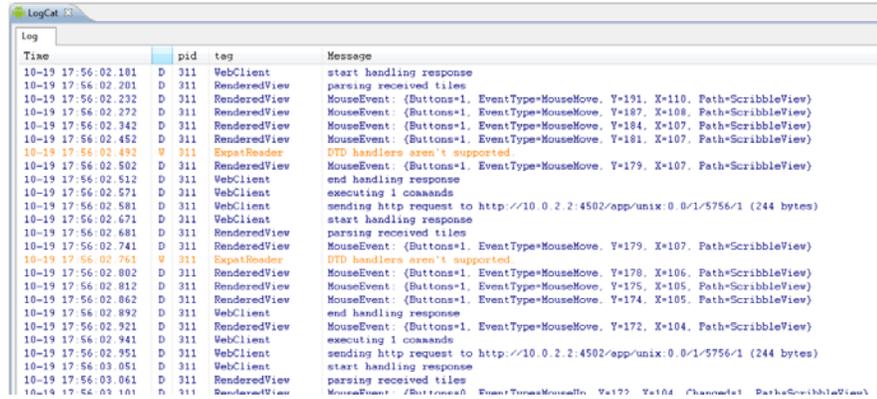
The diagram below shows the classes comprising the Android Scribble client. For Asteroids, simply replace `ScribbleClient` with `AsteroidsClient`.



Debugging the Code

To debug a PureWeb-enabled Android client, we recommend that you add the Dalvik Debug Monitor Server (DDMS) plugin to Eclipse. With it, you can view the messages produced by the emulator or device.

The sample application is released as debuggable in the manifest. To debug, simply review the request, response, and event information to the log file.



Time	pid	tag	Message
10-19 17:56:02.181	D 311	WebClient	start handling response
10-19 17:56:02.201	D 311	RenderedView	parsing received tiles
10-19 17:56:02.232	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=191, X=110, Path=ScribbleView)
10-19 17:56:02.272	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=187, X=108, Path=ScribbleView)
10-19 17:56:02.342	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=184, X=107, Path=ScribbleView)
10-19 17:56:02.452	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=181, X=107, Path=ScribbleView)
10-19 17:56:02.492	V 311	ExpatriReader	DTD handlers aren't supported
10-19 17:56:02.502	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=179, X=107, Path=ScribbleView)
10-19 17:56:02.512	D 311	WebClient	end handling response
10-19 17:56:02.571	D 311	WebClient	executing 1 commands
10-19 17:56:02.581	D 311	WebClient	sending http request to http://10.0.2.2:4502/app/unix:0.0/1/5756/1 (244 bytes)
10-19 17:56:02.671	D 311	WebClient	start handling response
10-19 17:56:02.681	D 311	RenderedView	parsing received tiles
10-19 17:56:02.741	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=179, X=107, Path=ScribbleView)
10-19 17:56:02.761	V 311	ExpatriReader	DTD handlers aren't supported
10-19 17:56:02.802	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=178, X=106, Path=ScribbleView)
10-19 17:56:02.812	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=175, X=105, Path=ScribbleView)
10-19 17:56:02.862	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=174, X=105, Path=ScribbleView)
10-19 17:56:02.892	D 311	WebClient	end handling response
10-19 17:56:02.921	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=172, X=104, Path=ScribbleView)
10-19 17:56:02.941	D 311	WebClient	executing 1 commands
10-19 17:56:02.951	D 311	WebClient	sending http request to http://10.0.2.2:4502/app/unix:0.0/1/5756/1 (244 bytes)
10-19 17:56:03.051	D 311	WebClient	start handling response
10-19 17:56:03.061	D 311	RenderedView	parsing received tiles
10-19 17:56:03.101	D 311	RenderedView	MouseEvent: (Buttons=1, EventType=MouseMove, Y=172, X=104, Path=ScribbleView)

For more information about Android debugging, see <http://developer.android.com/tools/debugging/index.html>