



NOTE: This tool is in alpha testing. Please report problems (and successes!) to shaileen@media.mit.edu

How To use the App Inventor Merger for Development as a Team

The App Inventor Merger tool can be very useful when developing an app within a team. The tool allows for multiple developers to work on different screens of the app and then merge them together. This document outlines the process of using the App Inventor Merger to develop an app in a team environment. It will use the example of a simple two screen app developed by two different developers to demonstrate this process.

[Overview](#)

[Dividing Work](#)

[Developer 1 Work In App Inventor](#)

[Design View](#)

[Blocks Editor](#)

[Download Source Code](#)

[Developer 2 Work In App Inventor](#)

[Design View](#)

[Blocks Editor](#)

[Download Source Code](#)

[Merging into one Project](#)

[Launch the App Inventor Merger](#)

[Find and Load Both Projects](#)

[Merge the Projects](#)

[Upload Final Project to App Inventor](#)

[Extras](#)

[Universal Assets and Databases](#)

[Assets](#)

[Databases](#)

[Merging More Than Two Projects](#)

[Appendix](#)

[Appendix A: Complete Blocks for Screen1 of CountdownScreen1](#)

[Appendix B: Complete Blocks for SetTime Screen](#)

Overview

Each developer will work on their own separate project file. These project files can be either under the same username or different usernames. There can only be one "Screen1" per project



and “Screen1” can not be renamed; therefore only the developer designing the first screen to appear on the app should populate “Screen1.” All other developers should leave “Screen1” blank and only develop additional screens.

Developers can write code to call a screen they are not developing but they must know the name that has been assigned to that screen by its developer. Also, two different developers/screens can use the same database or asset but they must be named the same (for more details on this see the [Universal Databases and Assets](#) section below). Finally, no two different screens or assets can have the same name. For these reasons it is important to decide beforehand the name of each screen and a naming convention for assets that will insure no unwanted duplicates.

Once separate projects are complete they can be merged together using the App Inventor Merger.

Dividing Work

Work should be divided by screens. Each screen should be assigned to a developer and there should be only one version of each screen to be merged into the final app. This document will follow an example for which there are two screens and two developers working on an app named “CountDown.” The CountDown app will work like a timer. The first screen shows the time counting down and allows the user to start, stop or reset the timer. From the first screen the user can also switch to a second screen to set the amount of time to count down.

Developer1 will work on the first screen and call it “Screen1”. Developer2 will work on the second screen and call it “SetTime”. The naming convention for assets will be the screen name followed by the asset name. There are no universal assets or databases. Screen1 will open the SetTime screen and when the SetTime screen is closed it will pass a number, representing the total number of seconds to countdown, to Screen1.

Developer 1 Work In App Inventor

Developer1 will log into App Inventor using their account and create a new project called “CountDownScreen1.”

Design View

The Design view for CountDownScreen1’s Screen1 is shown below in Figure 1. Note that the project name, the screen name and assets follow the predetermined naming convention.

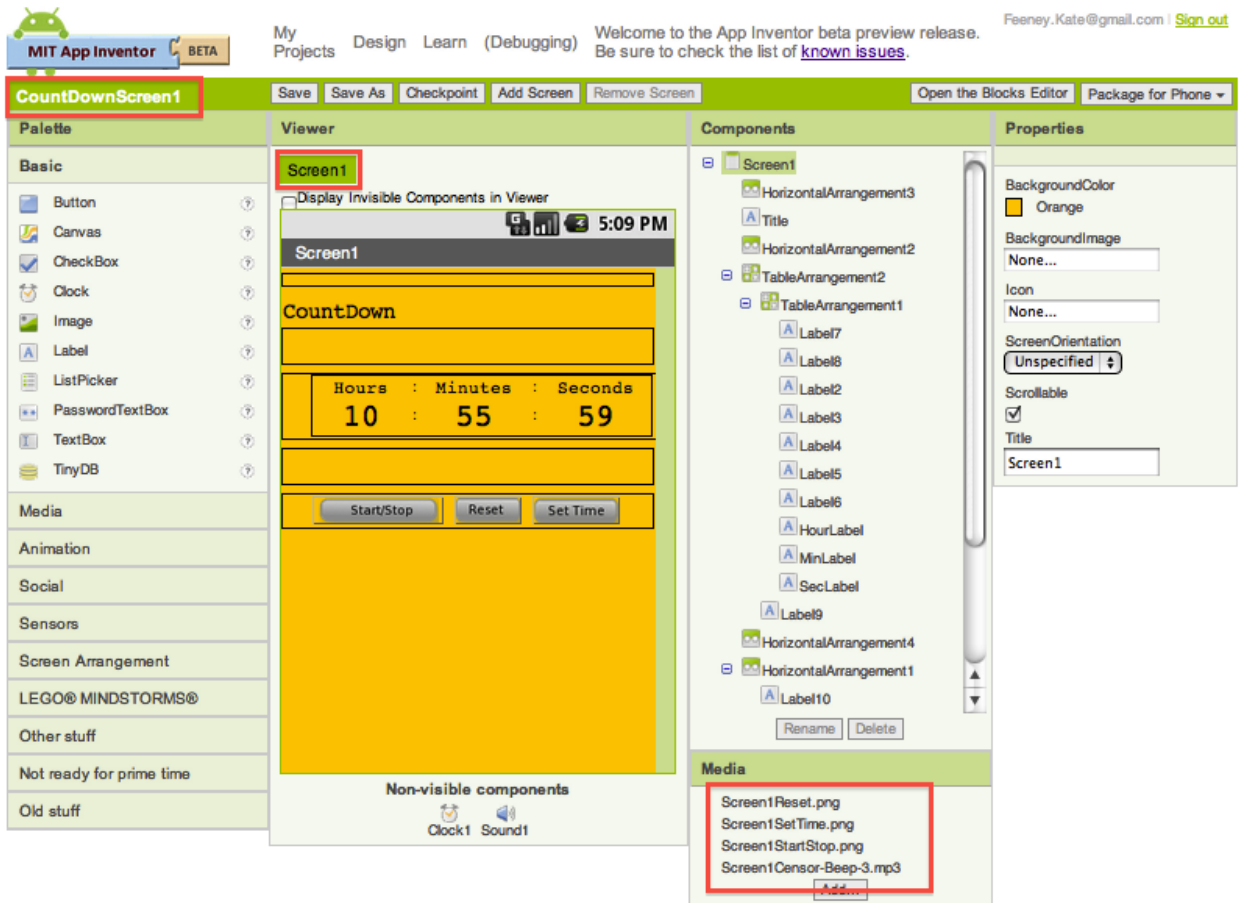


Figure 1: CountdownScreen1's Screen1 Design View

Blocks Editor

Developer1 then creates the blocks needed for Screen1. These blocks included a block to open the SetTime screen when the "Set Time" button is clicked and a block to handle when the SetTime screen is closed. Views of the SetTime.Click block and the Screen1.OtherScreenClosed block are shown in Figure 2 and 3 respectively. The complete set of blocks for Screen1 are shown in Appendix A.

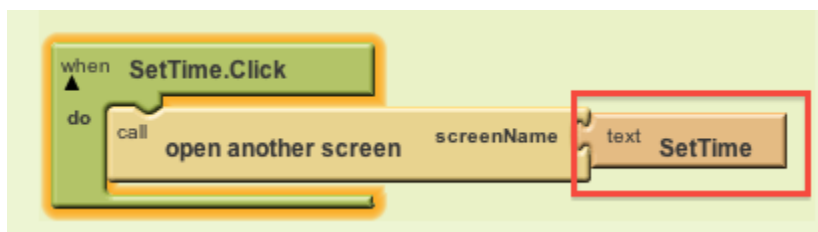


Figure 2: CountdownScreen1's Screen1 SetTime.Click block

In the SetTime.Click block, Screen1 opens the SetTime screen. Note that the string assigned to screenName must be exactly what was decided in advance as the name of the second screen.

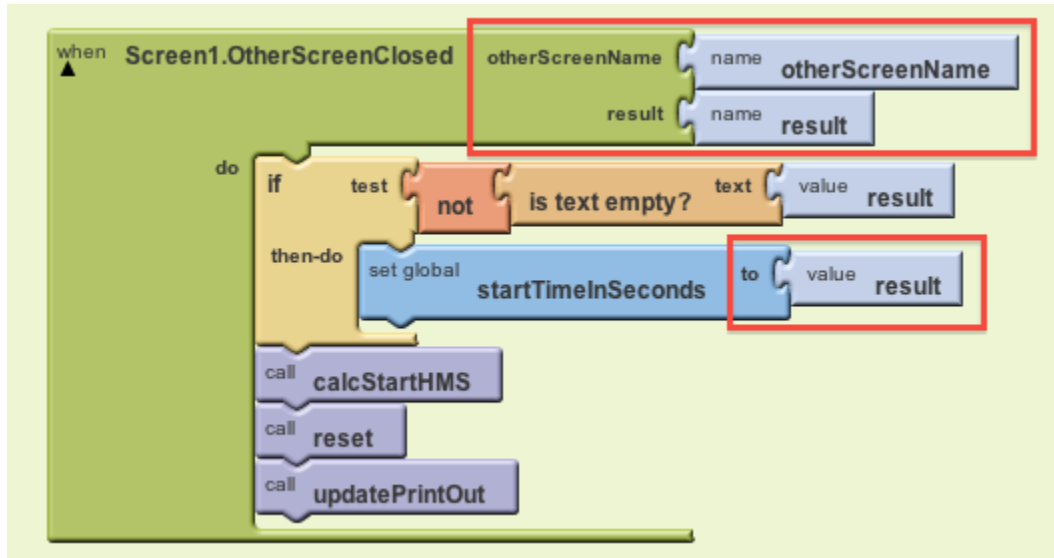
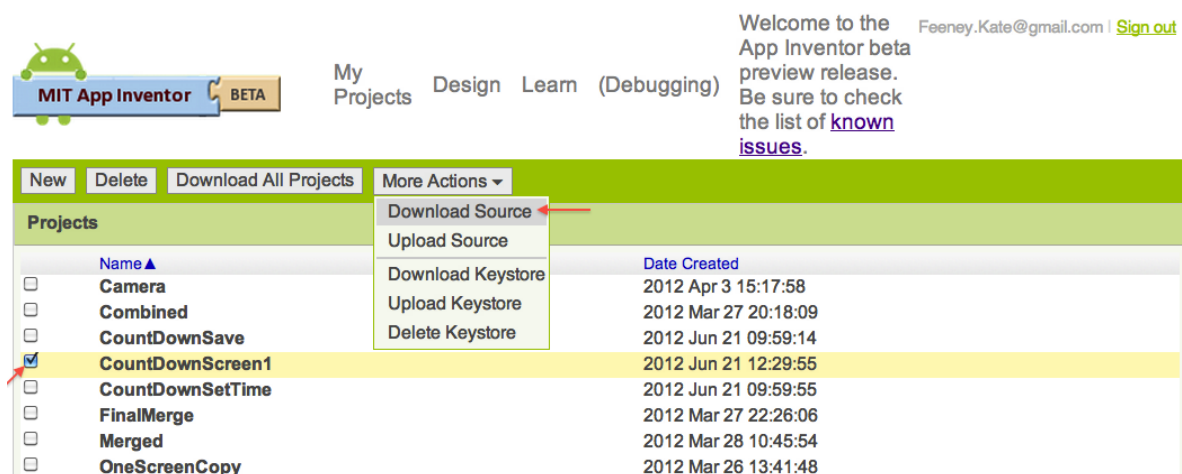


Figure 3: CountdownScreen1's Screen1 OtherScreenClosed block

The Screen1.OtherScreenClosed receives the number of seconds to countdown from the SetTime screen when the screen is closed and sets the startTimeInSeconds variable to it.

Download Source Code

Once Developer1 completes Screen1 they download the source code. This is done by going to the My Projects view and checking the checkbox next to the CountdownScreen1 project. Then clicking on the "More Actions" dropdown and selecting "Download Source". In Figure 4 red arrows show how to select a project and where to click to download the project's source.



Welcome to the App Inventor beta preview release. Be sure to check the list of [known issues](#).

Name ▲	Date Created
<input type="checkbox"/> Camera	2012 Apr 3 15:17:58
<input type="checkbox"/> Combined	2012 Mar 27 20:18:09
<input type="checkbox"/> CountdownSave	2012 Jun 21 09:59:14
<input checked="" type="checkbox"/> CountdownScreen1	2012 Jun 21 12:29:55
<input type="checkbox"/> CountdownSetTime	2012 Jun 21 09:59:55
<input type="checkbox"/> FinalMerge	2012 Mar 27 22:26:06
<input type="checkbox"/> Merged	2012 Mar 28 10:45:54
<input type="checkbox"/> OneScreenCopy	2012 Mar 26 13:41:48

Figure 4: Steps to download CountdownScreen1 source

Developer 2 Work In App Inventor

Developer2 will log into App Inventor using their account and create a new project called “CountDownSetTime.”

Design View

The Design view for CountDownSetTime’s Screen1 is shown below in Figure 5. The Screen1 is empty except for a button that takes you to the SetTime screen. This button is only for Developer2 to get to the SetTime screen during testing and debugging. This Screen1 will not be merged into the final app.

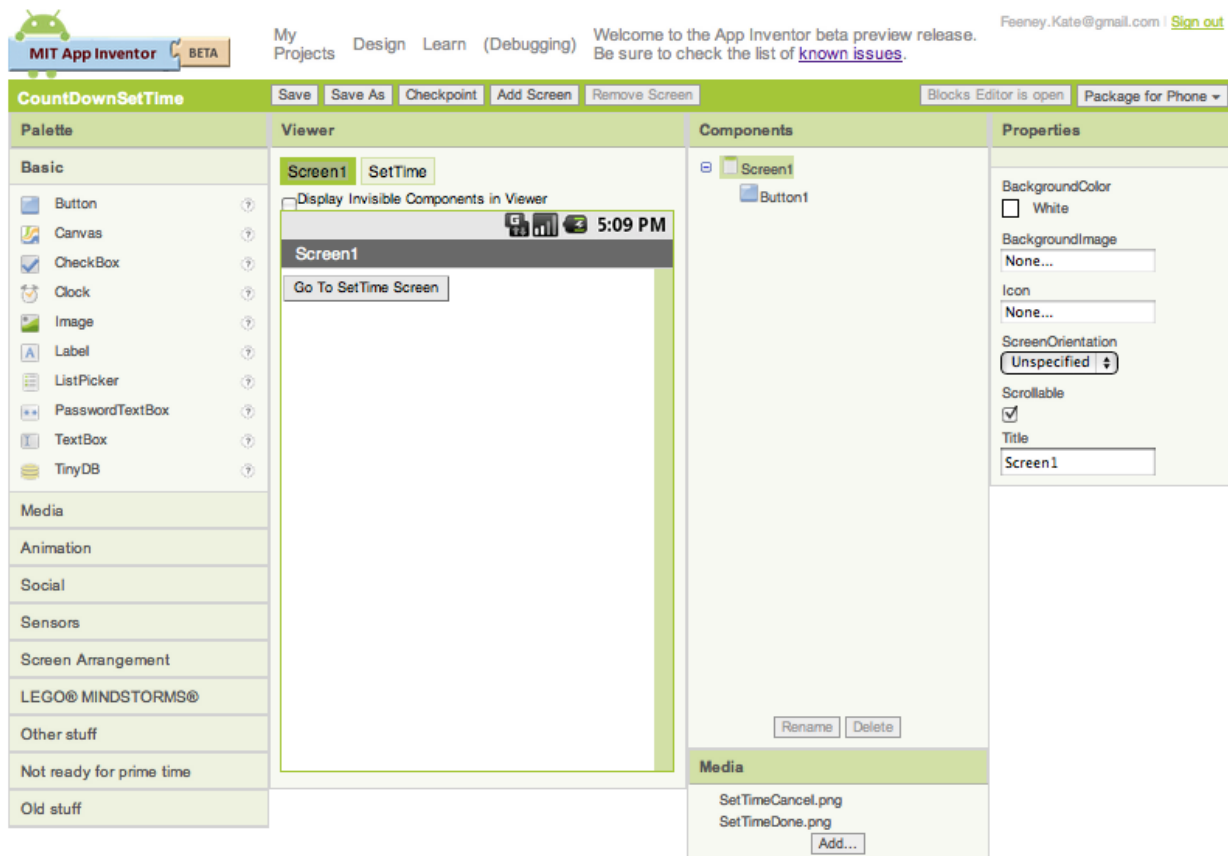


Figure 5: CountDownSetTime’s Screen1 Design View

The Design view for CountDownSetTime’s SetTime screen is shown below in Figure 6. Note that the project name, the screen name and assets follow the predetermined naming convention.

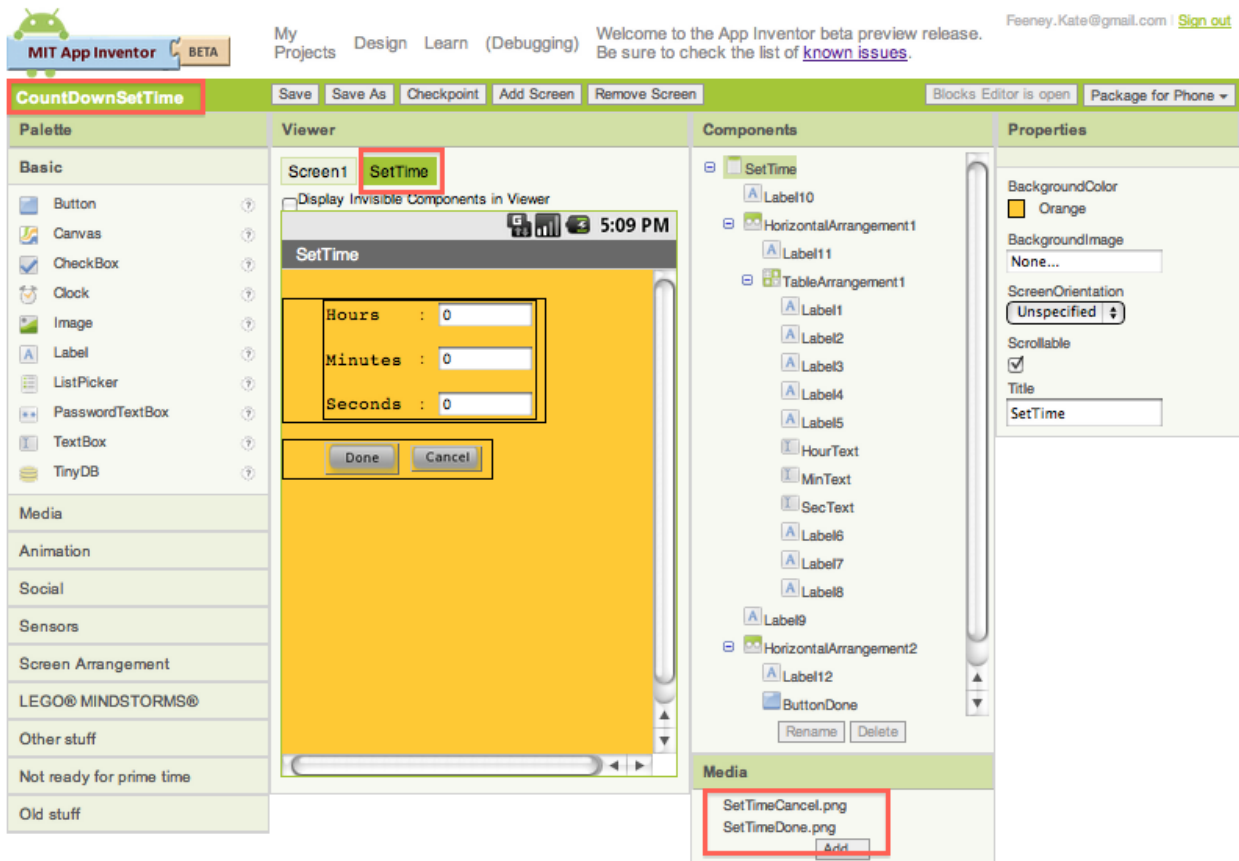


Figure 6: CountdownSetTime's SetTime screen Design View

Blocks Editor

Developer2 then creates the blocks needed for the SetTime screen. These blocks include a block to send the number of seconds to countdown to Screen1 when the SetTime screen is closed. This block is the ButtonDone.Click block shown in Figure 7. A complete set of blocks for the SetTime screen are shown in Appendix B.

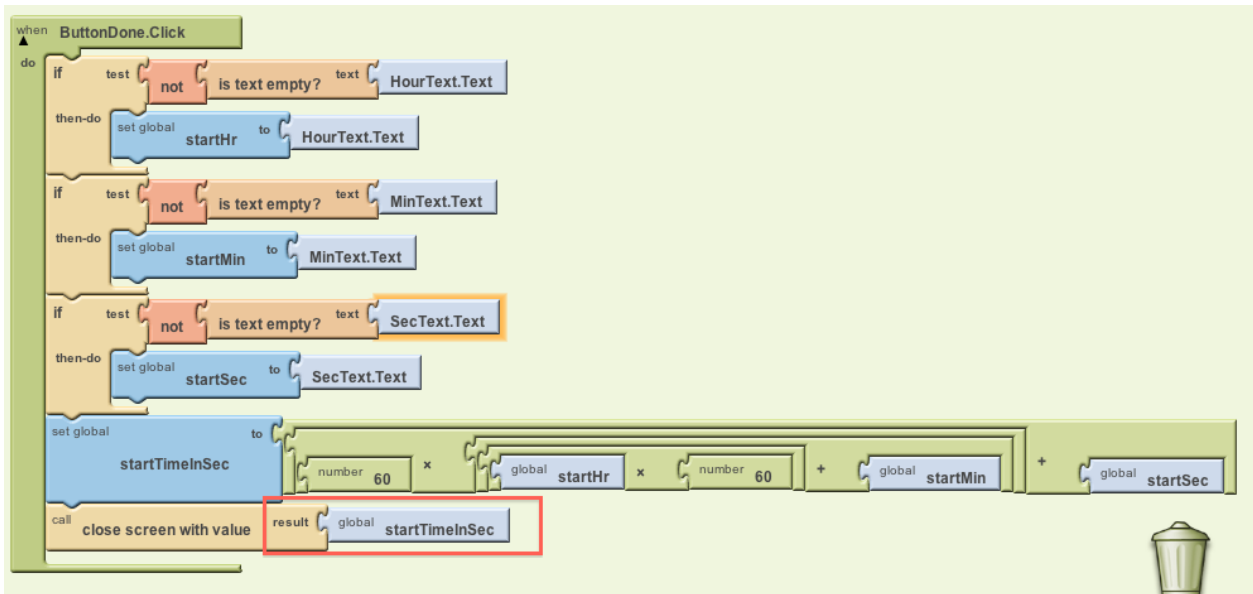


Figure 7: ButtonDone.Click block for CountdownSetTime's SetTime screen

Download Source Code

Developer2 downloads the source for CountdownSetTime following the same steps Developer1 followed to download the source for CountdownScreen1.

Merging into one Project

Once both developers have downloaded the source code for their respective projects, the two projects can be merged into the final app using the following steps.

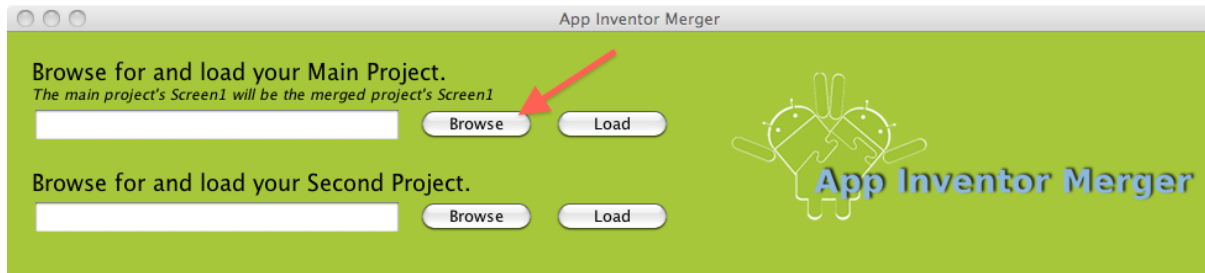
Launch the App Inventor Merger

If not already done, download the latest version of the App Inventor Merger. It can be found [TO BE DETERMINED]. App Inventor Merger downloads as a jar file which can be saved wherever is convenient. To launch the application simply double click on the jar file.

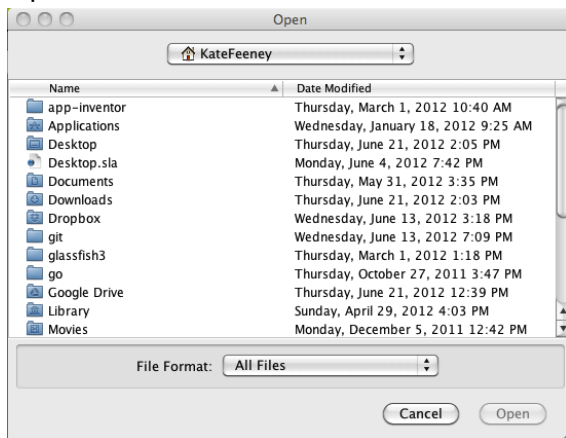
Find and Load Both Projects

The main project will be CountdownScreen1 since Screen1 from this project will be the Screen1 for the final app and the second project will be CountdownSetTime.

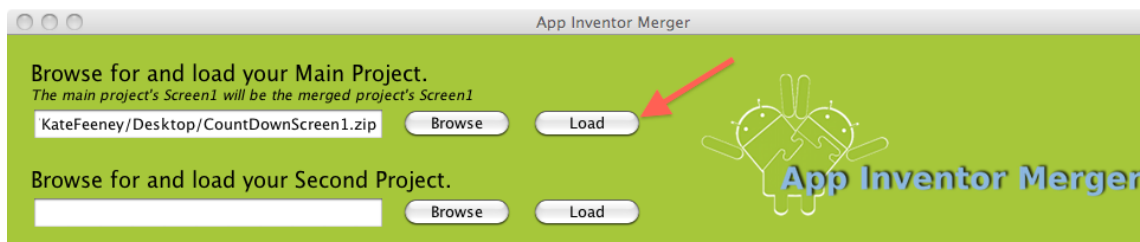
Select the browse button for the main project.



A file browser window will appear. Find and select the CountdownScreen1.zip file, then click Open.

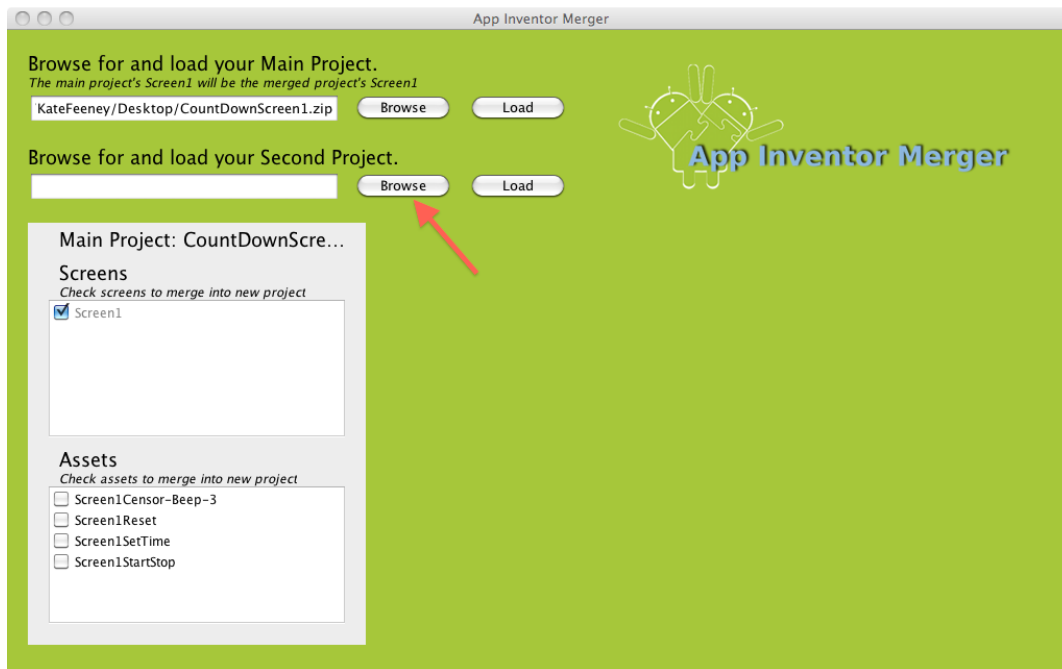


The path for the project file will appear in the main project text box. Click the main project's Load button to load the project into the App Inventor Merger.

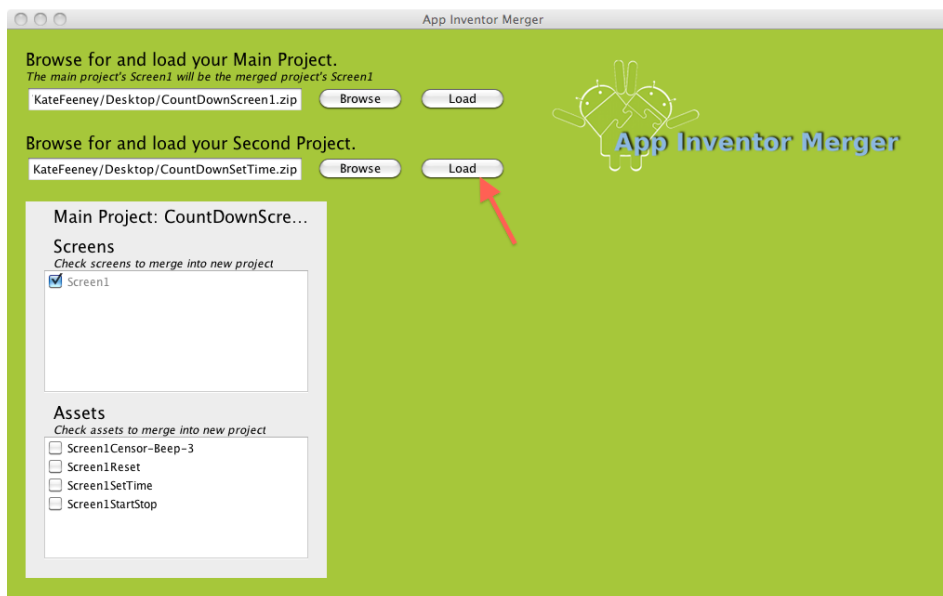


A list of the main project's assets and screens will appear in the lower left hand corner of the screen and this means that the main project has been loaded into the App Inventor Merger.

Click the browse button for the second project.



A file browser window will appear. Find and select the CountDownSetTime.zip file, then click Open. The path for the project file will appear in the second project text box. Click the second project's Load button to load the project into the App Inventor Merger.

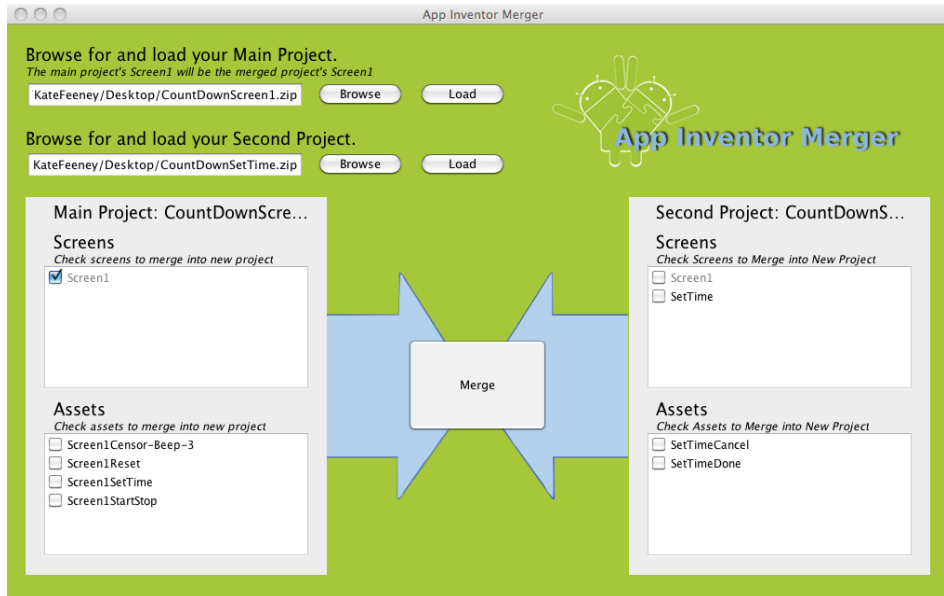


A list of the second project's assets and screens will appear in the lower right hand corner of the screen and this means that the second project has been loaded into the App Inventor Merger.



Merge the Projects

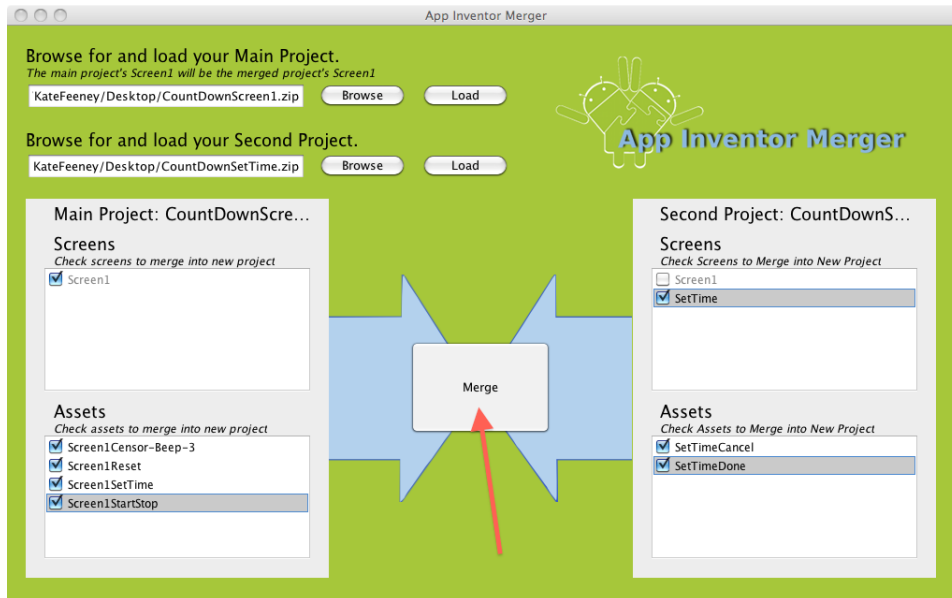
Once two projects have been loaded a Merge button will appear between them.



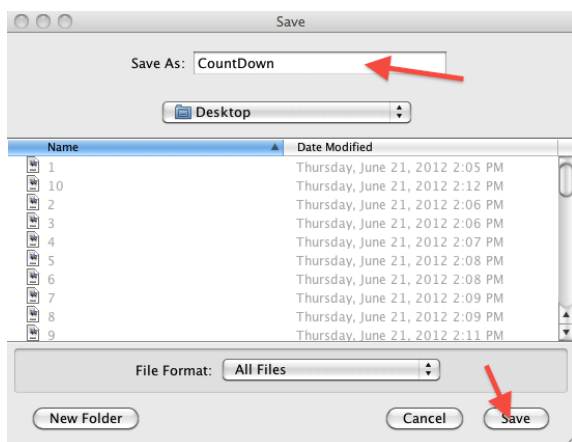
Check the boxes next to all of the screens and assets you wish to merge into the final app.

In this case only one screen is being selected from each project but in other cases multiple screens can be merged from the same project file. For this example all assets and screens are selected except Screen1 from the second project. Screen1 from the second project is grayed out since Screen1 from the main project is required and two screens with the same name can be merged.

Click Merge.



A save dialog window will appear. Browse to where you would like to save the project, enter the project's name and then click Save. The project will be saved as a zip file. In this example the project's name will be Countdown and the file Countdown.zip will be saved on the desktop since that is the directory selected (although this file can be saved in any directory).

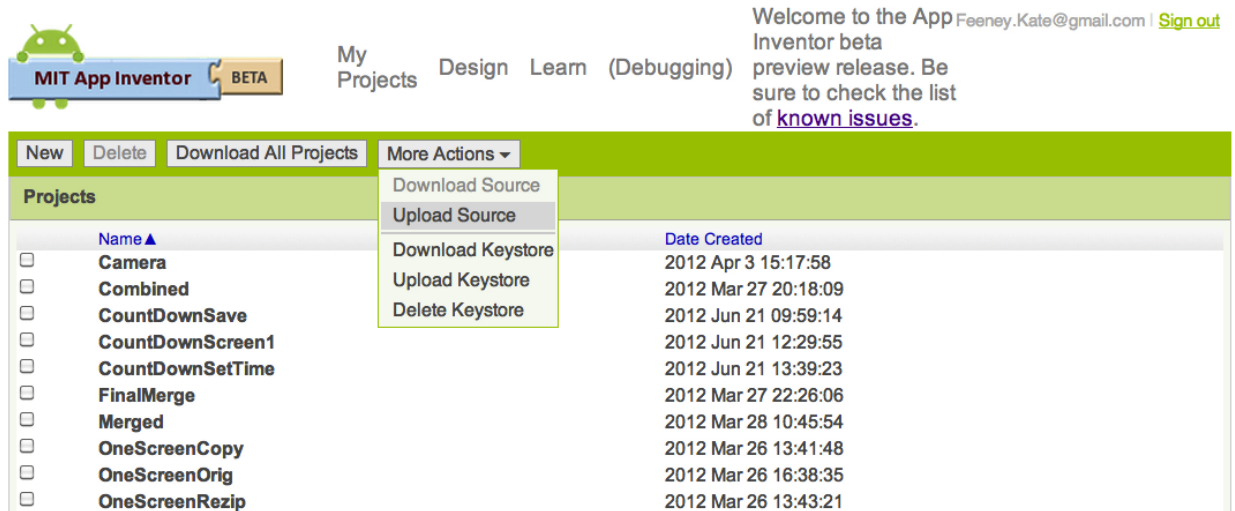


A dialog box will appear letting you know that your projects have been successfully merged. There will now be a zip file on the desktop named Countdown.zip. The dialog box will also ask if you would like to merge another project. Since this example only has two developers working on two different project files, there is no need to merge another project. Click No and confirm that you want to close the merger. For more information about merging more than two project see the [Merging More Than Two Projects](#) section below.

Upload Final Project to App Inventor

The zip file created by the App Inventor Merger is your new project file. This project file can be uploaded to App Inventor so that you have a complete project in App Inventor.

To upload the file, launch App Inventor and go to the My Projects page. Click on the More Actions dropdown and select Upload Source.



Welcome to the App Inventor beta
Feeney.Kate@gmail.com | [Sign out](#)
preview release. Be sure to check the list of [known issues](#).

MIT App Inventor BETA

My Projects Design Learn (Debugging)

New Delete Download All Projects More Actions ▾

Projects

Name ▲	Date Created
<input type="checkbox"/> Camera	2012 Apr 3 15:17:58
<input type="checkbox"/> Combined	2012 Mar 27 20:18:09
<input type="checkbox"/> CountdownSave	2012 Jun 21 09:59:14
<input type="checkbox"/> CountdownScreen1	2012 Jun 21 12:29:55
<input type="checkbox"/> CountdownSetTime	2012 Jun 21 13:39:23
<input type="checkbox"/> FinalMerge	2012 Mar 27 22:26:06
<input type="checkbox"/> Merged	2012 Mar 28 10:45:54
<input type="checkbox"/> OneScreenCopy	2012 Mar 26 13:41:48
<input type="checkbox"/> OneScreenOrig	2012 Mar 26 16:38:35
<input type="checkbox"/> OneScreenRezip	2012 Mar 26 13:43:21

Download Source
Upload Source
Download Keystore
Upload Keystore
Delete Keystore

The following dialog box will appear. Select Choose File. A file browsing window will appear. Find and select the zip file that was just created by the App Inventor Merger. For this example select the Countdown.zip file located on the desktop and then click Open followed by OK.



Your new project now appears in your list of projects and opens in the App Inventor window.

Extras

Universal Assets and Databases



Assets

If your app uses the same asset on multiple screens, then you can make that asset universal. This is done by giving it a name that does not follow the normal naming convention but instead is repeated by all screens. When the projects are merged the universal asset will only need to be selected from one of the projects.

For example, imagine you have a logo, which is an image, that should appear at the top of every screen. Instead of having every developer name the logo something different (Screen1Logo, Screen2Logo ...), as the naming convention would require, every developer can simply name it Logo. When the projects are merged only check the Logo asset listed under the main project so that the asset is only loaded into the final app once but all the screens will be able to access it.

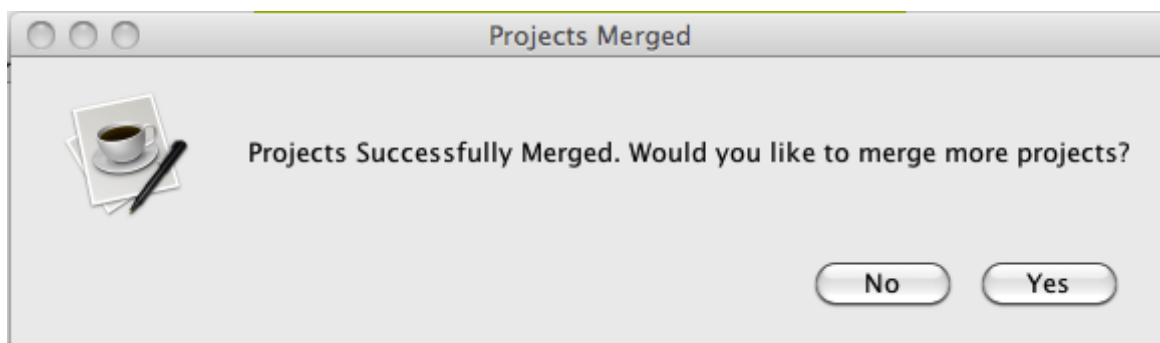
Databases

Apps that use databases can be merged using the App Inventor Merger very easily. Different developers can even work on different screens, that use the same database, separately and then merge them together at the end.

The only requirement for merging screens that share a database is that the name assigned to the database for each screen is the same. Once the projects are merged the same database will work for all the screens.

Merging More Than Two Projects

The App Inventor Merger is still a very useful tool even if there are more than two developers working on more than two project files. The process is exactly the same as the two developer process, described above, until the dialog box letting you know your project has been successfully merged pops up (shown below).



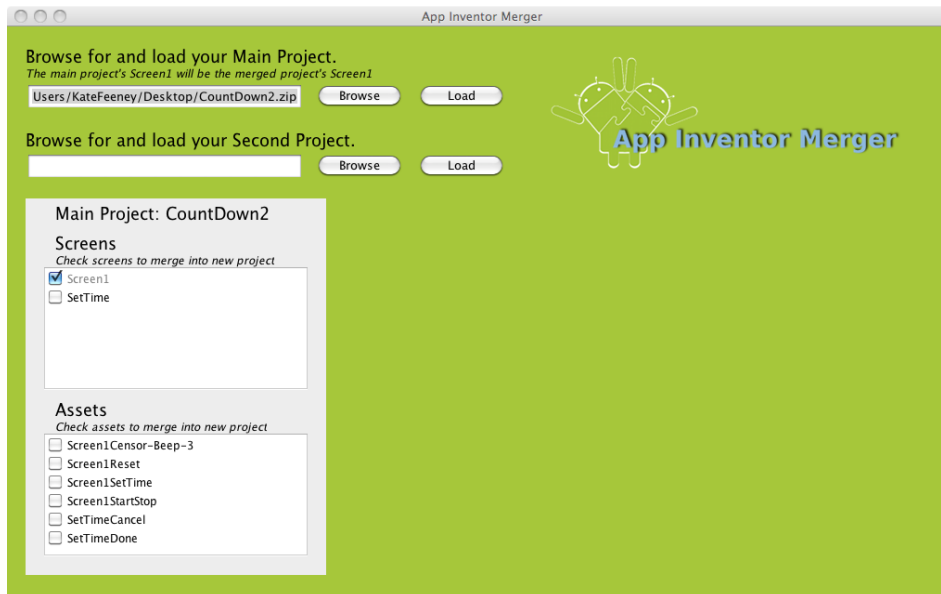


When there were only two developers we selected no but for the case of more than two developers select Yes since there are more projects to be merged. A new dialog box will appear asking if you would like to use the project you just created as the new main project.



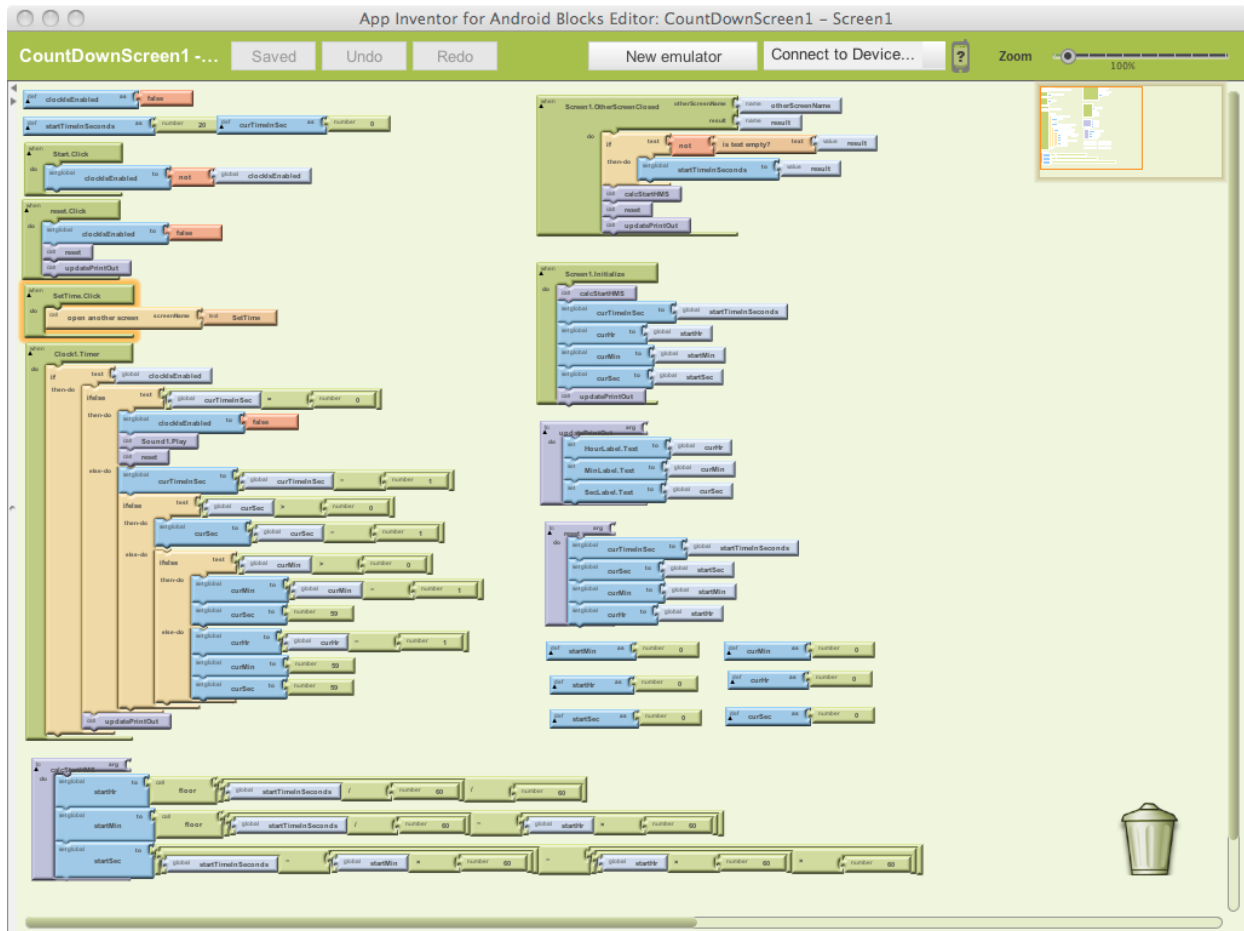
If you select No, the App Inventor Merger will restart with no projects loaded, the same as if you just launched it. This would be used to merge two new projects together.

If you select Yes, the App Inventor Merger will restart with the project you just created loaded as the main project and the second project is blank (as shown in the picture below). This option can be used if you would like to merge a third project with the first two projects you previously merged.



Appendix

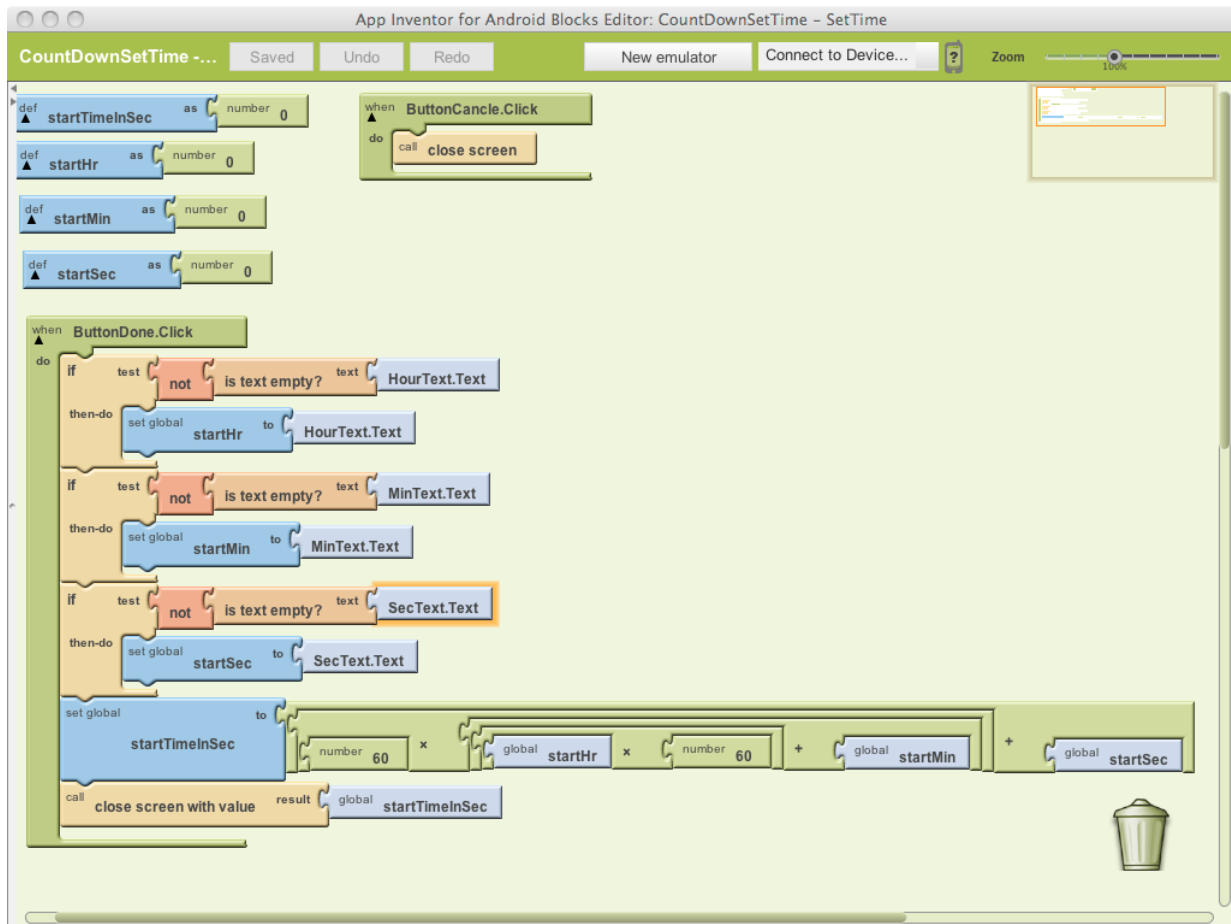
Appendix A: Complete Blocks for Screen1 of CountdownScreen1



The screenshot displays the App Inventor Blocks Editor interface for "CountDownScreen1 - Screen1". The workspace contains the following logic blocks:

- Screen Initialization:**
 - `Screen1.OnScreenClosed`: Checks if the screen is empty; if not, it starts the timer and updates the display.
 - `Screen1.Initialize`: Sets initial values for `startSeconds`, `curTimeSec`, `curMin`, `curSec`, `startMin`, `startSec`, and `curMin`.
 - `updatePrintOut`: Updates the labels for hours, minutes, and seconds.
- Button Clicks:**
 - `start.Click`: Starts the timer by setting `clockEnabled` to true.
 - `next.Click`: Increments the current time by one second.
 - `setTime.Click`: Updates the start time when the user clicks "set time".
- Countdown Timer:**
 - `clock.Timer`: A loop that runs while the timer is enabled. It checks if the current time is zero. If not, it updates the display and decrements the time. If zero, it stops the timer.
- Calculations:**
 - `calcHours`, `calcMin`, and `calcSec`: Functions that calculate the hours, minutes, and seconds from the total time.

Appendix B: Complete Blocks for SetTime Screen



The screenshot displays the App Inventor Blocks Editor for the 'SetTime' screen. The workspace contains the following blocks:

- Four variable definition blocks: `def startTimeInSec as number 0`, `def startHr as number 0`, `def startMin as number 0`, and `def startSec as number 0`.
- A `when ButtonCancel.Click` block with a `do` sub-block containing a `call close screen` block.
- A `when ButtonDone.Click` block with a `do` sub-block containing:
 - Three `if` blocks, each testing `not is text empty?` for `HourText.Text`, `MinText.Text`, and `SecText.Text`. Each `then-do` block sets the corresponding global variable: `set global startHr to HourText.Text`, `set global startMin to MinText.Text`, and `set global startSec to SecText.Text`.
 - A `set global startTimeInSec to` block followed by a mathematical expression: $(\text{number } 60 \times \text{global startHr} + \text{number } 60 + \text{global startMin} + \text{global startSec})$.
 - A `call close screen with value result` block where the `result` is `global startTimeInSec`.