

# Google Code-In 2015 sample tasks

## Coding

### Add 'include me in replies' option in email

In the Ubuntu email client 'dekko' the default when replying to all mail is to include the sender in replies. This task is to add a setting (which defaults on) to allow users to disable this feature. This involves some user interface work in the settings screen in QML and some backend work to store the setting and check it when creating email replies.

The result of this work should be committed to the dekko email client repository for review by the maintainer.

### Inline images should be scaled to fit

In the Ubuntu email client 'dekko' the default is to show all embedded images in emails at the original size. This task is to resize inline images so they fit the width of the screen.

The result of this work should be committed to the dekko email client repository for review by the maintainer.

### Add 'move to' option to email view

This task is to add an option to the Ubuntu phone dekko email client to allow a user to move a mail from within the mail view. This requires some QML and C++ work to implement the minor change to the user interface, adding a move option to the menu, and the backend work to implement the move mail.

The result of this work should be committed to the dekko email client repository for review by the maintainer.

### Enable fast scroll in music app

The Ubuntu music app can list a large number of individual tracks. The Toolkit provides a 'fast scroll' to jump to tracks starting with a particular letter. This task is to implement fast scroll in the Music app. This should fit with the design and toolkit items used in other core applications such as the contacts app.

The result should be committed to the Music App repository.

## Port pastebinit to snappy using snapcraft

Snappy is a new packaging framework used in IoT devices. We need you to port pastebinit from the archive using snapcraft as an example application to demonstrate how to use snapcraft and package a snappy application.

## Develop a Juju charm for phpbb3

Juju is an orchestration service for the cloud. It allows any person to deploy 'charmed' services into different clouds, such as Google Compute Engine. In order for a service to be available, someone has to have written a charm for it - a set of scripts that will install, manage and configure the service on an instance.

We need you to develop a charm for phpbb3. You can write it in any language that you want and feel comfortable with. python3 and bash are some recommended options, but any language can be used. Ideally, this charm would be able to relate with other charms, and be scalable.

You can find more information on how to write Juju charms in the Charm Authors section of our Juju documentation: <https://jujucharms.com/docs/stable/authors-intro>

You can find more information about charms, charming and see examples of current charms at <http://jujucharms.com/>

You will not have to worry about paying for Cloud instances hours - you will be provided with a key in order to use instances for Charming completely for free.

## Documentation / Training

### Edit the 'Social' Wiki page

Ubuntu has a wiki site, in which there is a Social page. However, this page is outdated. We need you to collect the links for all Ubuntu official social media pages, and put them nicely into this wiki page. We also need you to make sure it has a nice design, that's attractive for people.

The most important thing here is to collect all relevant social information, so people can follow Ubuntu easily around the web. For the design part, we intend you to use headers, tables and maybe some text formatting. You can check how to use the moinmoin language here: <https://wiki.ubuntu.com/HelpOnMoinWikiSyntax>

## Do the announcement for the Ubuntu Global Jam

During each Ubuntu cycle, the Local Communities Council sets a date for the Ubuntu Global Jam. This is an event where people from all around the world get together and help Ubuntu get better.

We need you to write an email that is going to be distributed to mailing lists, and posted in the LoCo Council blog. Bear in mind, this needs to be presented in a semi-formal style, with correct spelling, grammar and punctuation. Should be between 4 and 5 paragraphs long.

## Wiki clean-up

The Ubuntu wiki has a lot of outdated documentation. This task is to identify pages which are no longer relevant and remove or clean up the content. This is most useful when getting rid of tutorials for releases of Ubuntu which are no longer supported, or guides which give advice which is no longer valid. For example a guide which details how to install outdated or deprecated software.

The result of this task should be a list of wiki pages which are candidates for deletion, and a list of pages which have been changed (where deletion wasn't appropriate).

## Outreach / Research

### Create a Getting Started video

Create a 5 minute video detailing how to get started developing on Ubuntu using the Ubuntu SDK. The video should cover the basics of getting the SDK installed on Ubuntu and creating and running a first 'hello world' application.

Video should be provided in clear English and made on a familiar stock Ubuntu installation. Text and other details should be clearly visible when played back.

### Create an QML Next Steps video

Create a 5 minute video showing how to use the Ubuntu SDK to create a simple application or game. The video should assume 'Getting Started' has already been watched as a prerequisite and should follow on from there. Introduce the basics of QML and using some of the Ubuntu SDK User Interface components in a simple application.

Video should be provided in clear English and made on a familiar stock Ubuntu installation. Text and other details should be clearly visible when played back.

## Create a Testing Apps video

Create a 5 minute video showing how to test applications developed with the Ubuntu SDK on the desktop, in an emulator and on a physical Ubuntu phone device. The video should assume the Getting started and QML Next Steps videos have been watched. The video should mention - but not go into too much detail - on the app build process, but focus more on the 'write once, run everywhere' nature of QML.

Video should be provided in clear English and made on a familiar stock Ubuntu installation. Text and other details should be clearly visible when played back.

## Create a simple Bacon2D game tutorial

Bacon2D is a component used in some Ubuntu Phone games which is useful when making 2D games. This task is to create a simple QML game using Bacon2D, and to write up the game as a tutorial for publishing on the Ubuntu developer website.

The game should be quite simple, and easy to understand. The tutorial should be well documented with code snippets and screenshots.

## Review new contributor documentation for Docs Team

The documentation team maintains a wiki which contains all the information needed to contribute. These instructions need to be reviewed and edited for clarity.

Make sure they are in a simple language that anyone can understand, and contain references to other pages that can help the recent contributor to get used to the interface, language used, styling, and more.

## Help organize the Ubuntu on Air! channel in playlists

The Ubuntu on Air! Youtube channel ([youtube.com/ubuntuonair](https://youtube.com/ubuntuonair)) is used for various live streams, including Q&As, events, and more. We need you to help us organize the latest videos into the already-created playlists, so people can easily find the videos they are the most interested on. You will be given access to the channel administrative interface in order to organize the videos.

## Quality Assurance

Add a Camera app test to share a photo to the automated phone testing suite

The QA team maintains an automated testsuite for testing the Ubuntu Phone OS. The testsuite needs additional tests for the Camera app, including sharing a photo from the photo roll. We need you to utilize the testing API helpers to add this test.

Create manual testcase for calendar application

Manual testcases are an important part of closing the coverage gap for what cannot be automated. We need you to add a testcase for the new calendar application that will become part of the default desktop image in the upcoming (16.04) release.

Create a post-installation boot test for desktop images

Everyday new images of Ubuntu are generated from the latest version of packages available in the archive. These images are tested to insure they install, but not are currently not tested to make sure they boot up afterwards. We need you to add a post-installation test that will verify the installation boots to a desktop session.

Add remote test loading to Pilot application

The pilot application allows for manual testcases to be run by end users of Ubuntu Phone devices. Currently, we need to bundle the application with the tests, and thus we have to send lots of application updates in order to change tests. We need you to add a feature to the Pilot application to allow us to load the tests from a URL instead during the initial load of the application.

Implement the new design for qatracker

qatracker is a tool we utilize to record manual test results. It's a custom drupal module. There is a new community design which makes the site easier to use, done by other community members. However, we need someone to implement it. You will work with the designer to implement the design, and then have it deployed to production.

Create a test case for empty state in Music app

The Ubuntu Phone Music app displays a message when there is no music files found on the device. This task is to create an Autopilot test case in python to detect that the Music app is presenting the message correctly. The test should pass correctly and fit with the style and coding standards of the existing tests in the application.

The result should be committed to the Music app repository.

## Review merge proposals for Juju Charms

Juju is an orchestration service for the cloud. It allows any person to deploy 'charmed' services into different clouds, such as Google Compute Engine. In order for a service to be available, someone has to have written a charm for it - a set of scripts that will install, manage and configure the service on an instance. Charms, as any piece of code, need to be maintained to ensure that it works, any bugs are fixed, and any software updates are applied.

There are several merge proposals in the Charm Review Queue, located at [review.juju.solutions](https://review.juju.solutions). We need you to choose three of them, make sure that they comply with the Charm Review Criteria found at <https://jujucharms.com/docs/stable/reference-reviewers>, and that they work as expected (everything runs and the service starts). The task will be submitted as comments on these merge proposals, one on the respective Launchpad page for each proposal, containing a +1 or -1, with comments on how the Charm can be fixed, or if everything runs as expected.

Please remember that a Juju Charmer will base its review on your comments, and depending on that you may help the author to fix it before it lands, or stall the process! You're a vital part here.

You will not have to worry about paying for Cloud instances hours - you will be provided with a key in order to use instances for Charming completely for free.

## Review an incoming Juju Charm for the Charm Store

Juju is an orchestration service for the cloud. It allows any person to deploy 'charmed' services into different clouds, such as Google Compute Engine. In order for a service to be available, someone has to have written a charm for it - a set of scripts that will install, manage and configure the service on an instance.

The Charm Review Queue located at [review.juju.solutions](https://review.juju.solutions) contains a list of all open merge proposals and Charm submissions for the Charm Store. We need you to pick one of the Charms (not merge proposals) and make sure it follows all Charm review guidelines (<https://jujucharms.com/docs/stable/reference-reviewers>). Once you verify that everything is in place and works as expected, you can leave a review comment on the bug letting Charmers know that the charm can now be promulgated. If you find discrepancies or something is not running correctly, state that in your comment. Please, make sure to be as precise as possible. -1 comments which state all the errors/discrepancies found and maybe even tips on how to fix them are greatly appreciated, since the author will have a shot to review the Charm before it gets to the hands of another Juju charmer. If you place a -1 comment, please mark the bug as Incomplete as well.

You will not have to worry about paying for Cloud instances hours - you will be provided with a key in order to use instances for Charming completely for free.

## Triage Weather app bugs for relevance

The Ubuntu Weather app was recently rewritten mostly from scratch. Many bugs exist on the bug tracker which may no longer apply. This task is to review all the existing bugs to see if they do or not.

The bugs should each have their status set appropriately, with additional comments left if the bug can or cannot be reproduced anymore.

## User Interface

### Create a desktop interface for the Weather app

The Ubuntu Phone Weather application was primarily designed for use on a mobile device. When run on a desktop it occupies only a small window. This task involves creating mock-ups for a converged desktop view of weather in multiple locations.

The results should follow the Ubuntu design guidelines and be consistent with other core applications.

The materials should be provided in source (psd / svg) format and exported as PNG.

### Create a tabbed user interface for the Terminal app

The Ubuntu Phone Terminal application already has a tabbed interface, but it's not consistent with current designs of other applications on Ubuntu phone. This task involves creating mock-ups for a converged tabbed user interface which works across various form factors (mobile, tablet, desktop) and is consistent with Ubuntu design guidelines and other applications such as the Webbrowser app.

The materials should be provided in source (psd / svg) format and exported as PNG.