

# Google Summer of Code 2017

## Proposal to Python Software Foundation

Project: Font Reshaping and Font Fallback Support

### Sub-organization information

Sub-organization applying to work with: **Kivy**

### Mentors

Akshay Arora

Jacob Kovac

Matthew Einhorn

### Student Information

**Name:** Parth Wazurkar

**Alternate names:** parth

**Email:** parthwazurkar@gmail.com

**Telephone:** +919552283766/+918668509601

**Time Zone:** Nagpur, India UTC+5:30

**IRC:** parthw1@irc.freenode.net

**Source Control Username:** <https://github.com/parthw1>

**Website:** <https://parthw1.github.io>

**Blogs:** <https://parthw1.wordpress.com>

## University Information

**University:** Indian Institute of Information Technology, Nagpur

**Major:** Computer Science and Engineering

**Current Year:** 1<sup>st</sup> Year

**Expected Graduation date:** In June 2020

**Degree:** B-Tech

## Code contributions

I am an active contributor to Kivy and have been contributing to it since January 2017. I am also active on IRC discussing various features and aspects of Kivy.

Besides supporting the codebase I helped Kivy by spreading word about Kivy on quora and other social media platforms.

## Project Proposal Information

**Proposal Title:** Font Reshaping and Font fallback support

**Proposal Abstract:** Currently Kivy does not support reshaping for alphabets such as Arabic, Persian, Thai, or Devanagari and it also does not support font fallback. This project will focus on solving the shaping problem using the harfbuzz shaper and providing font fallback to Kivy using fontconfig.

This project will focus on providing a completely new text provider to Kivy utilizing harfbuzz shaper similar to the current text provider sdl, and font fallback support using fontconfig.

## Project Description:

### What is text layout?

Text Layout is how the characters we type show up on the screen. But this is just a 10000 ft view. Text layout and rendering are much more complex. According to me, Text layout and rendering are crucial components for any visible interface. It is one of the most visible part of an application UI. When text doesn't look good, the UI degrades.

Kivy currently uses sdl as a text provider which does not provide complex text layout capabilities, and creates a problem for the following reasons:

1. No uniformity in rendered text across platforms. Differences introduced due to different implementations of the rendering stack.
2. It is difficult to maintain so many implementations. High code complexity and higher chances of bugs creeping in. If there is a bug in the layout code, it has to be solved by someone who knows the layout code specific to that platform.
3. Rendering is also affected by OS specific bugs.

My project is aimed at solving all these problems by unifying the layout code by creating a new layout engine which uses Harfbuzz for Open Type glyph shaping.

After creating a new layout engine with Harfbuzz, It won't help us with bidirectionality. If we want to lay out text with mixed Hebrew and English (for example), we will need to ensure that the buffer provided to Harfbuzz has those characters in the correct layout order. This will be different from the logical order in which the Unicode text is stored.

In other words, the user will hit the keys in the following sequence:

- **A B C [space] א ב ג [space] D E F**, but will expect to see in the output: **ABC אבג DEF**
- This reordering is called *bidi processing* ("bidi" is short for bidirectional), and there's an algorithm as an annex to the Unicode Standard which tells you how to reorder a string from logical order into presentation order. Before sending any string to Harfbuzz, we

may need to apply the bidi algorithm to it. Libraries such as ICU and fribidi can do this for us.

- Here in this project we will be using fribidi for bidirectionality.

Font fallback support is required to be independent of text providers. So with the use of fontconfig this problem will be solved.

## Timeline and List of Deliverables

### Community Bonding Period

- Familiarize myself with widgets, layouts, and properties in Kivy which I have not used before.
- Will ensure that Harfbuzz can be compiled on every platform, and integrate it as a core text provider.
- Get acquainted with mentor and community. Establish clear times for meetings, code reviews, and spam sessions!
- Fill all the gaps in my understanding of Harfbuzz, freeebidi and Fontconfig.

### May 30-June 29,2017

#### (solving text shaping problem phase)

- Setup Harfbuzz free type font face.
- Create a buffer and put text in it.
- Guess the script, language and direction of the buffer.
- Create a face and a font, using Free Type for now.
- Shape!
- Get the glyph and position information.
- Iterate over each glyph.
- Tidy up.
- The above steps will take a UTF8 string, shape it, and give us the information required to lay it out correctly on a single horizontal (or vertical) line using the font provided.
- That is the extent of Harfbuzz's responsibility.
- Integrate the new text provider with Kivy.
- Remove pep8 errors, put proper comments and get the code ready.

- Write a blog report about my progress.
- Get the work done ready for first evaluations.

### **July 1 - July 29, 2017**

#### **(Solving *bidi* problem and making sure that the new text layout engine works on all the platforms )**

- Implementing the bidi algorithm.
- Use friBidi to extract the v2l and dir tables (get\_bidi\_types).
- friBiDi pass
- Using the l2v list identify the direction of every character.
- Split the original text into monodirectional runs.
- For each run, we save the lowest visual position (LVP) and the lowest logical position (LLP).
- Shape every segment separately (setting the appropriate buffer direction).
- Sort the segments based on their LVP.
- Add the LLP to the cluster of every glyph.
- Integrating friBidi with Harfbuzz shaper.
- Write a blog report about my progress.
- Remove pep8 errors, put proper comments and docs.
- Get the work done ready for second evaluations.

### **July 28-August 29,2017**

#### **(Providing font fallback support to Kivy )**

- Providing font fallback support using fontconfig.
- Similar to the one used in browsers like Firefox and chrome.
- I would discuss with mentors regarding use of fontconfig for font-fallback otherwise I am planning to use some other library like 'raqm' to provide font-fallback.
- Integrating this with Kivy.
- Write a blog report about my progress.
- Get the work done ready for final evaluations.

## Motivation for GSoC:

I've always felt the need for a multi-platform system which would make it easier to develop for all platforms, while maintaining uniformity in the application. Kivy meets those expectations brilliantly. I specifically chose to work on the Font reshaping part because I always used to wonder, how we are able to see so many different fonts on screen, how are they created.. I especially like Kivy because it is open source, written in Python (a language I am most comfortable with) and also because it can be used to deploy to multiple platforms using Kivy's capabilities. I hope to continue contributing to Kivy after GSoC.

## Other Commitments

- Have you applied with any other organizations? **No**
- Do you plan to have any other jobs or internships during this period? **No**
- Do you have any other short term commitments during this period?**No**
- Do you have exams or classes that overlap with this period?  
My current semester ends on 06<sup>th</sup> May and next semester starts from 27<sup>th</sup> July. From 06<sup>th</sup> May to 27<sup>th</sup> July, I am completely free and after college re-opens, I won't be able to work full time but I can manage to work for at least 6 - 8 hours per day.