GSoC 2017 Proposal to Kivy (Python Software Foundation)

Project Title: Kivy: Plyer

Sub-Organization Information

Kivy

Mentor

Akshay Arora

Student Information

Name: Bhavesh Anand
Alternate names: bhaveshAn
Email: bhaveshanand7@gmail.com
Telephone: +91-9536128183
Time zone: Dehradun, India UTC+5:30
GSoC Blog RSS feed Url: https://bhaveshan.wordpress.com/

University Information

University Name: G.B. Pant University of Agriculture and Technology, Pantnagar
Major: Information Technology
Current year: 3rd year
Expected graduation date: In June 2018
Degree: Bachelor of Technology (B.Tech)

Other Contact Information

Alternate Email: bhaveshanand96@gmail.com
Website: https://bhaveshan.github.io/
IRC: bhaveshAn@irc.freenode.net
Github: http://www.github.com/bhaveshAn
Skype: bhaveshanand96@gmail.com (Bhavesh Anand)
Quora: https://www.quora.com/profile/Bhavesh-Anand-5
Instant Messaging: https://www.facebook.com/bhavesh.anand.3

Project Proposal Information

Proposal Title: Kivy: Plyer

Proposal Abstract:

Providing stable and platform independent Plyer API for accessing features on desktop and mobile devices

- Improvement in existing implemented features
- Implementation of existing features for other platforms
- Adding and implementing new features
- Documenting and evaluation

Project Description and Timeline:

Plyer is a platform-independent Python API to use features commonly found on the desktop and mobile platforms supported by Kivy. The idea is to provide a stable API to the user for accessing features of their desktop or mobile device.

- ➢ Requirements
  Access to Linux, Windows, OS X, iOS device, Android device.
  However, I have personal Linux, Windows, Android and iOS systems and planning to buy OS X system during the community bonding period.

- ➢ Moving some of the PyJNIus and PyOBJus code respectively from P4A and kivy-ios to Plyer.

- ➢ Familiarity with other modules and API references :
  - PyJNIus: A Python module to access Java classes as Python classes using JNIUS.
  - PyOBJus: A Python module to access Objective-C classes as Python classes using Objective-C runtime reflection.
  - P4A/python-for-android: It is a project to create your own Python distribution including the modules you want, create an apk including python, libs, and your application.

- Kivy-ios: It is designed to compile the necessary libraries for iOS to run the application and manage the creation of the Xcode project
- Android Documentation, iOS Developers Library, ctypes, Dbus.

Four stages declared in the abstract are described as follows.

Improvement in existing implemented features

Need

Currently the Plyer API is having lots of issues which are discussed here #issues.
Out of these, wifi feature for many platforms have external dependencies. For Linux wifi depends on a python wifi module. For windows also it is not working (#318, #272).
Notification feature for Linux depends on python – dbus module and notify-send #318.
There are many other issues which are raised due to dependencies.

Goals

For Linux the network accessing binaries and shell scripts like nmcli, ifconfig, ifquery etcwill be used to configure and query the wifi connectivity. For Notification issue of Linux, apart from notify-send I will try to find the utility on other Linux distros as well and will make to come to a specific solution.

Benefits

There are a lot of benefits. Cleaning code, removing unwanted dependencies for different features in each platform can eventually solves many of the issues raised for existing features discussed in this section.

Implementation of existing features for other platforms

Overview

As mentioned in the README there are many features that are not yet implemented for other platforms, mainly the desktop operating systems #important . These features includes Accelerometer (#33, #9) Camera taking pictures (#34, #30)Audio recording, Orientation, Email (#32) Flash, Light, Proximity, Barometer, Temperature, Gravity, GPS (#186,#53) Notification #23 etc. However, iOS is also missing many sensors.

Implementation

Camera taking pictures and Audio recording for Linux systems can be solved using gstreamer1.0 (a command line tool).Many pre-installed binaries and shell scripts can be used to add missing features in Linux.
For sensor implementation on OS X and iOS coremotion framework provides handler for motion events.
For Android, the implementation for Native file chooser and Wifi is needed. Implementation of Native file chooser will make use of DocumentProvider framework added in API 19 and this framework which supports searching, creating, opening, editing and deleting files. The persist permissions are also available. For wifi access implementation WifiP2pManager, WifiConfiguration and WifiInfo classes helps to add connect, disconnect and information gathering of wifi connection. Here the work in the pull request #155 will be useful.
The individual API references are described in the table below. APIs of Linux systems are not included as further research is needed by me.

Platforms	Android	iOS	Windows	OS X
Accelerometer			Sensor	
Audio recording		AVAudioRecorder	MediaCapture	AVAudioRecorder
Barometer		Core Motion	Sensor	
Camera(taking picture)			MediaCapture	
Compass			Compass	
GPS			Get Location	CoreLocation
Gravity		Gravity	Sensor	
Gyroscope			GyroMeter	
Light			LightSensor	
Magnetic Field	Magnetometer	CMDeviceMotion	Magnetometer	
Native file Chooser	DocumentProvider			
Notifications		userNotification		
Orientation		UIDevice		
Pedometer	StepDetector	CMPedometer		

Proximity		UIDevice		
Rotation Vector	Rotate		OrientationSensor	
Wifi	WifiP2pManager	NetworkExtension		

Advantages

From focusing on existing and implementing features for other platforms we will be much closer to the stable release of Plyer. This will help us to shift towards the newly introducing features as well.

Adding and implementing new features

Overview

There are new sensors which are not implemented yet for any platform such as Pedometer, Magnetometer, Humidity, etc. The features like Spell Checker, Barcode and QR Scanner, Bluetooth low energy, Contacts (access and editing), Finger Print Scanner, Google Play Integration, Geolocation, In-App billing, In-App browser, Internationalization, Network Information, NFC, Sharing (images, media and text), Speech to text, Speech Recognition, Status Bar, etc are also required to be proposed.

Implementation

The features discussed in this section will be a bit tricky. For Android and iOS these will be implemented using APIs for respective mobile platform. Geolocation for android will make use of Location and LocationManager classes and LocationListener interface. For NFC, the NfcManager, NfcEvent and NfcAdapter classes will be used. Implementation of other features on each platform will be done making use of API reference of corresponding platform and any pre-installed binaries found if any. However, references to individual APIs are given below. APIs of Linux systems are not included as further research is needed by me.

Platforms	Android	iOS	Windows	OS X
Bluetooth	Bluetooth	CoreBluetooth	Bluetooth	CoreBluetooth
Camera(Capture Video)	capture-video	AVFoundation	MediaCapture	
Fingerprint Scanner	fingerprint			

Geolocation	LocationManager	CLLocManager		
In-app Billing	Billing			
Internationalization	Localization			
Magnetic Field	Magnetometer	CMDeviceMotion	Magnetometer	
Network Information	NetworkInfo			
NFC	NfcAdapter			
Pedometer	StepDetector	CMPedometer		
Rotation Vector	Rotate		OrientationSensor	
ScreenShot		UIView		
Sharing			Share-data	
Speech Recognition	Recognizer	Speech	Recognizer	NsSpeech
Spell Checker	SpellCheck	UITextChecker		NsSpellChecker
Storage Path	Storage	FileManager		FileManager
System Information		UIDevice	DeviceInfo	

Advantages

From working on the features mentioned in this section, we can allow the users to introduce newly developed features in the modern era for their respective applications. More importantly, applications which will be made using the new release of Plyer API by different users across the globe will cover most of the currently demanding features which will make to believe on real and most reliable power of Plyer.

Documenting and Evaluation

After going through the above mentioned stages, the project will require documenting every feature which is added during the coding period. This requires thorough review of each and every feature for all platforms and made changes where-ever required.

Timeline:

Community Bonding period:

I will be reading about the implementation for these features, gain more knowledge of PyJNIus, PyOBJus, kivy-ios, p4a and will go through required documentations (mentioned in References). I will be in touch with my mentors and take suggestions.

Coding Period:

Here I have divided my time and workflow according to the 4 stages which are described in the project description.

(Week 1)

- **Improvement in existing implemented features** will be carried on.
- Modification of examples and facades will be done during this period simultaneously.
- If time permits, will move to the next stage as discussed earlier.

(Week 2 to Week 6)

- **Implementation of existing features for other platforms** will be focussed on.
- Moving the code from p4a and kivy-ios to plyer.
- The implementations which are made for Linux will also be checked on its different distributions as well.
- Being in touch with the mentor and will tell about every feature implementation made.
- Continuous contributions of these implementations on GitHub so that all the developers can have their view on code.
- Writing a blog about the progress of the project each week by the weekend.

First Evaluation (Week 4 and Week 5)

- Make preparation for the first Evaluation
- Regular guidance and feedback from the mentor.
- Submitting the Evaluation, sharp.

(Week 7 to Week 11)

- **Adding and Implementing new features** as mentioned in the abstract.
- Discussion with the mentor about other new features as well.
- Coding to implement new features known by the discussion with the mentor till that time.
- Continuous contributions of implementation of newly proposed features on GitHub so that all the developers can have their eyes on the code.

- Blog posts on the addition of new features with the illustrations at the end of each week.

Second Evaluation (Week 8 and Week 9)

- Make preparation for the second Evaluation
- Regular guidance and feedback from the mentor.
- Submitting the Evaluation on time,sharp.

(Week 12)

- **Documenting and Evaluation** of the project.
- Continue to implement features, if left any.
- Completing any missing documentation or the example.
- Complete evaluations and code submission sharply before 29th August.

Code Sample:

Link to a patch/code sample, preferably one you have submitted to your sub-org (*):

- Contributions in Plyer:

1. Linux Audio feature. (Waiting for approval)
2. Linux Camera feature. (Waiting for approval)
3. Android Relative Humidity. (Waiting for approval)
4. Android Magnetometer feature. (Waiting for approval)
5. Android Pedometer feature. (Waiting for approval)
6. Android Rotation Vector feature. (Waiting for approval)
7. Android Linear Acceleration feature. (Waiting for approval)
8. Linux Orientation feature. (Waiting for approval)
9. Linux ScreenShot feature. (Waiting for approval)
10. Linux Bluetooth feature. (Work in progress)
11. OS X ScreenShot feature. (Work in progress)
12. OS X Audio feature. (Work in progress)
13. OS X Camera feature. (Work in progress)
14. Wifi Facade correction. (Merged)
15. UniqueId Facade documentation correction. (Merged)

- Contributions in Kivy project:
    1. Example correction. (Merged)

Other Commitments:

- Do you have any other commitments during the main GSoC time period?
  No.
- Do you have exams or classes that overlap with this period?
  No, I do not have any kind of exams between 1st June and 15th July. However, my next semester will start from mid-July, so I will not be able to work full time except for the weekends. As it will be my final year, so work load from academic side will be less.
- 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
- Have you applied with any other organizations? If so, do you have a preferred project/org?
  No