

*Presentation
starts at 1:35 pm*

IN104 : Projet informatique

Build a navigation app for your smartphone

Stephen CREFF - stephen.creff@gmail.com

1st project session

April 10, 2020

Outline

1. IN104 project specificities – a reminder
 - Evaluation, milestones, ...
2. The project: Build a navigation app for your smartphone
3. About organizing your work
 - A word on software development and project management
4. Building a smartphone app
 - Cross-platform overview
 - The kivy platform
5. First project session – choose, install, make tutorials...
6. Q&A

1] IN104 project specificities – a reminder

- IN104 must help you learn manage your autonomy and independence
 - Essential today for an engineer
 - 1 ECTS = 25h of work (all included) -> Expect Min. 2h weekly of extra work at home required
 - Work in teams (pairs), coordinate with Git
- What are normal grades from past years?
 - No Exam -> Everything between 10 -12 and 20.
 - Up to you to get beyond average
 - A self-learning project
 - Internet is to be used
- **Report: 1 single PDF containing link to repository:**
 - Quality over Quantity
 - Results (plots, insights, conclusions and learnings): more important than lengthy reports explaining the problem

1] Reminder: Evaluation

1. 25% **Source code**: features, tests, code documentation, etc. Source code must be documented, the report including a link to the repository must be in a .zip file. The repository will contain the sources, as well as a plain text file README.md which indicates the actual operational features and limitations. Python code should compile, the teaching assistants are not supposed to make significant corrections for it to compile: a code with few features, but that compiles and does not crash will be preferred to a more complete code but which is not directly operational!
2. 25% **Defense** (10 minutes presentation, 5 min. questions): The formal quality of the presentation will be an important element. The defense needs to include a demonstration on the basis of the source code, an analysis of the difficulties encountered and implemented solutions. It will not include a presentation of the problem or the method of resolution that the teaching assistant obviously knows already well. The defense is open to everyone (subject to the acceptance of the pair that will present). The chronological order of defense will be given by the list of each group.
3. 20% **Practice Analysis**: You will return a critical (max. 5 pages) report before the defense in which you analyze and criticize the progress of your project and its success and failure factors. This evaluation component also includes the oral treatment of this question during the defense.
4. 30% **Continuous Progress Evaluation** of the practical work (mid-term evaluation) during the practical lab sessions (based on git commits).

1] The project milestones

Work on Project weeks

- 10/04/2020 13:30 - 15:30
- 24/04/2020 13:30 - 15:30
- 15/05/2020 13:30 - 15:30 -> intermediate report
- 18/05/2020 08:30 - 11:45

Last Session Project Defense Day (10/15 min of soutenance per team including questions)

- 22/05/2020 13:30 - 15:30

Deadline to send report (containing link to project repository) to your Teaching Assistant:
2 days before defense day

1] Autonomous work does not mean working on your own

- Assistance is provided
 - Through the forum: <https://ecampus.paris-saclay.fr/mod/hsuforum/discuss.php?d=450>
 - Via discord (during sessions): <https://discord.gg/taanwsv>
 - By mail to stephen.creff@gmail.com
- Mid-term report hand in:
 - Show to your TA, at least, the first 50% of your project report in the equator of the course
 - makes sure you are on track
 - The report can include (short):
 - Project Requirements
 - Software Architecture
 - Classes, Tests, Plots, Versions, Documentation...
 - Schedule & progression

1] To overcome the current covid context..

Fill-up the google sheet, for me to have all the contexts so I can provide specific content:

- <https://docs.google.com/spreadsheets/d/1m9cbjcSYFWNlvqheDi7zgtVsbgH3TxXOTPL3X78AN3c/edit?usp=sharing>

When formulating your questions:

- Be as specific as you can
 - Installation issues, precise the os, the framework...
- Push all your code on git so I can be up to date
- Don't hesitate to take and send screenshots

2] Build a navigation app for your smartphone*

- Use native or cross-platform app development framework to build an iOS or Android app,
- Choose your language (Python, Java, ...) and find your development framework (tech stack),
- Understand the framework and the APIs (widgets, ...),



- Make your System Analysis and Requirements, make it incremental!
 - Add features and sensors solicitation step by step, from trivial to rich location-based
 - Define map (simplified, real, feature-rich), set geo-markers, calculate and draw route, ...
- Define a storyboard and mock-up the Interface design
- Define and build functions using smartphone sensors
 - Touch screen, Geolocation service, ...
- Package & deploy your app on the targeted OS

*the smartphone is not provided

2] Choose your features, enhance some

There is no “best” result for this project, choose to enhance some features

- Mock-up the Interface design
 - From trivial,
 - To integration of frameworks
- Map
 - Build a simplified map
 - Integrate existing map services, e.g open street map,
 - ...
- Navigation
 - Location service
 - Routing service
 - e.g. optimal routing problems, Dijkstra’s algorithm, and pathfinding. We ultimately present the general concepts involved in the process of finding the shortest path between two locations (in a graph)



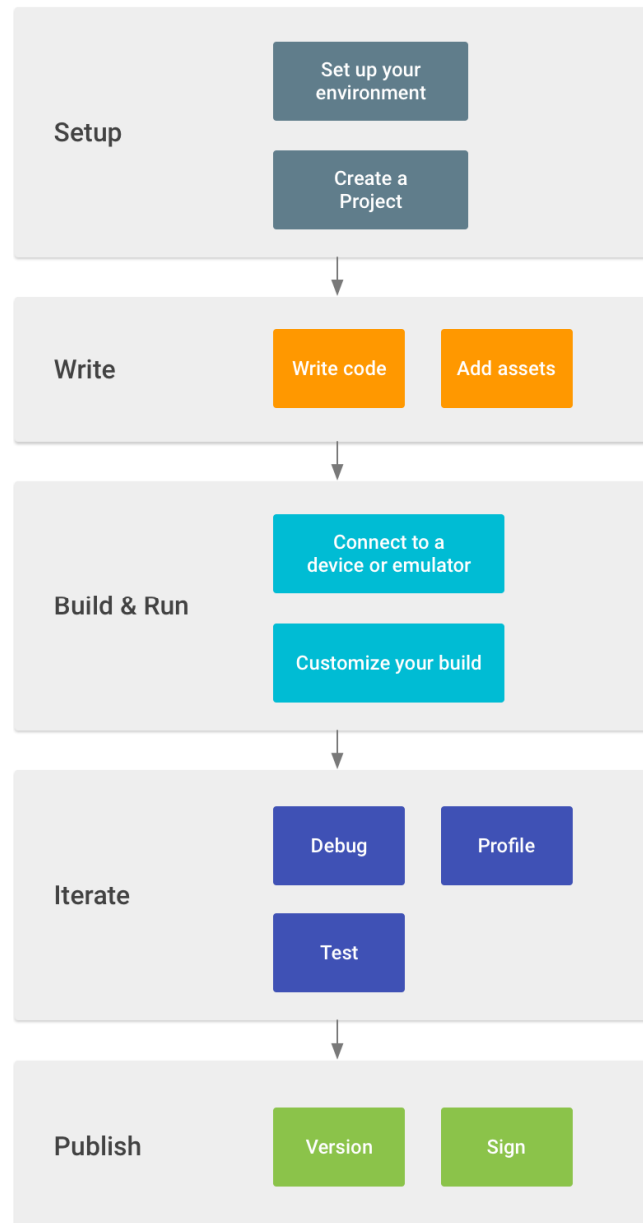
3] Software Development Activities

- Software development process
 - Project management plan (The lifecycle of the software development project is composed of), e.g. :
 - Software specification,
 - Software detailed design,
 - Software coding and unit tests,
 - Software integration
 - Software verification tests
 - Make a Planning
 - Gantt or equivalent
 - Define the rôles and responsibilities
 - List of roles and responsibilities for each of development activities
 - Allocation of roles to person in the team

In Documentation

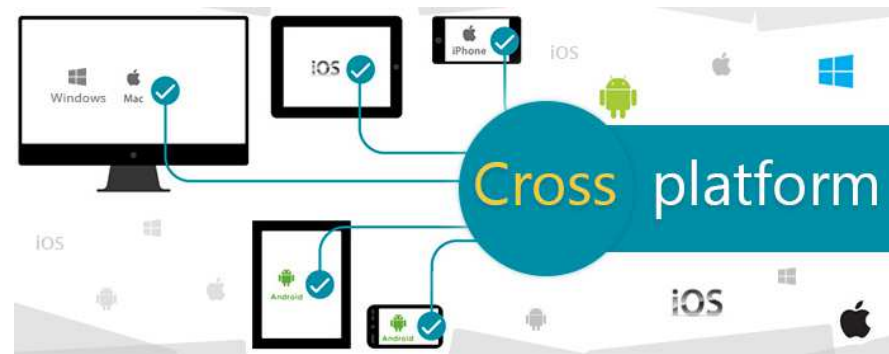
Don't loose the schedule
10/04/2020 13:30 - 15:30
24/04/2020 13:30 - 15:30
15/05/2020 13:30 - 15:30 -> intermediate
report
18/05/2020 08:30 - 11:45
22/05/2020 13:30 - 15:30 -> soutenance

3] Follow a full system development workflow



4] About cross-platforms

- To be concise, cross-platform is a type of software that has the ability to run on multiple computing platforms i.e., Android, iOS, Windows, Blackberry, etc.



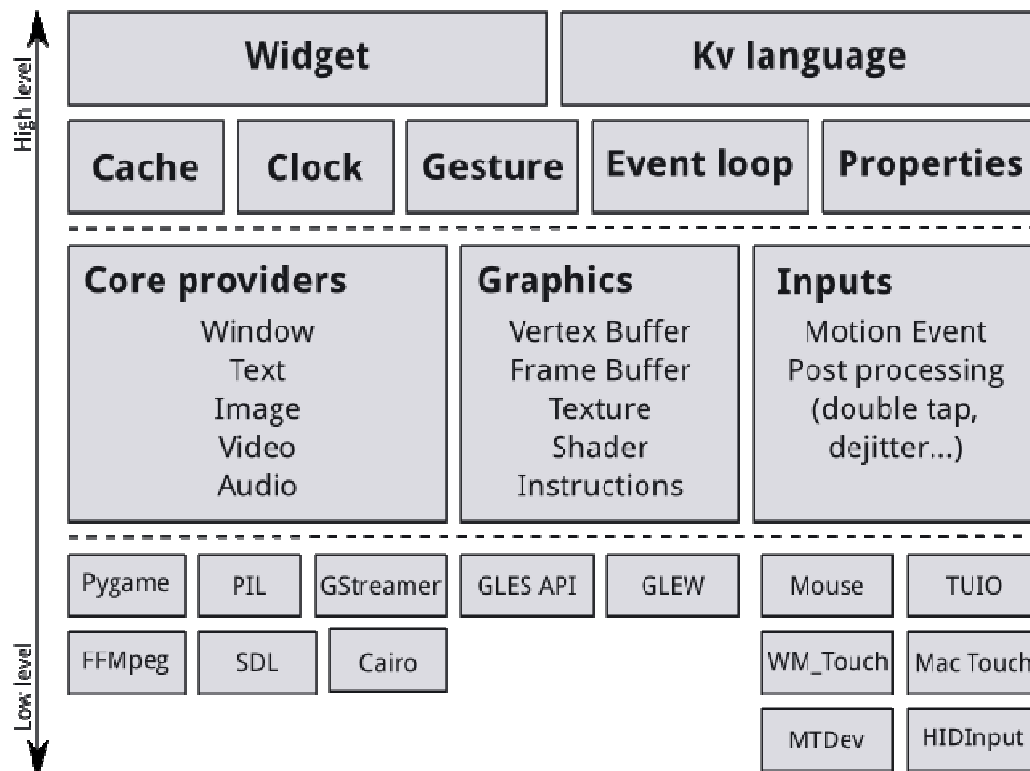
- <https://instabug.com/blog/cross-platform-development/>

4] Frameworks to build a Mobile Applications

- A python framework : <https://kivy.org/#home>
- A java framework : <https://www.codenameone.com/>
- Other framework, with combination of many languages i.e., HTML5, JavaScript, and CSS and Cordova wrapper :
<https://ionicframework.com/>

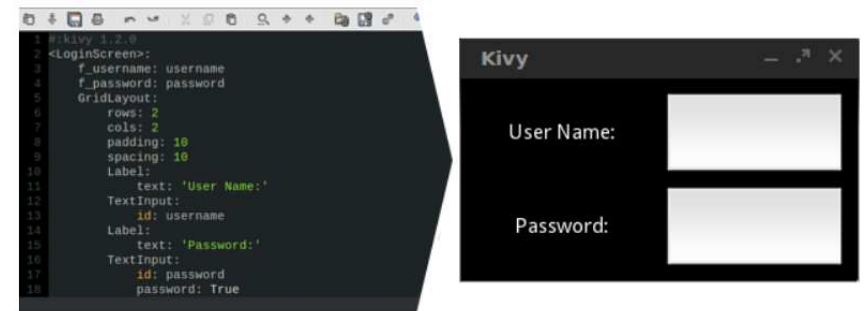
5] The Kivy cross-platform

Kivy Architecture

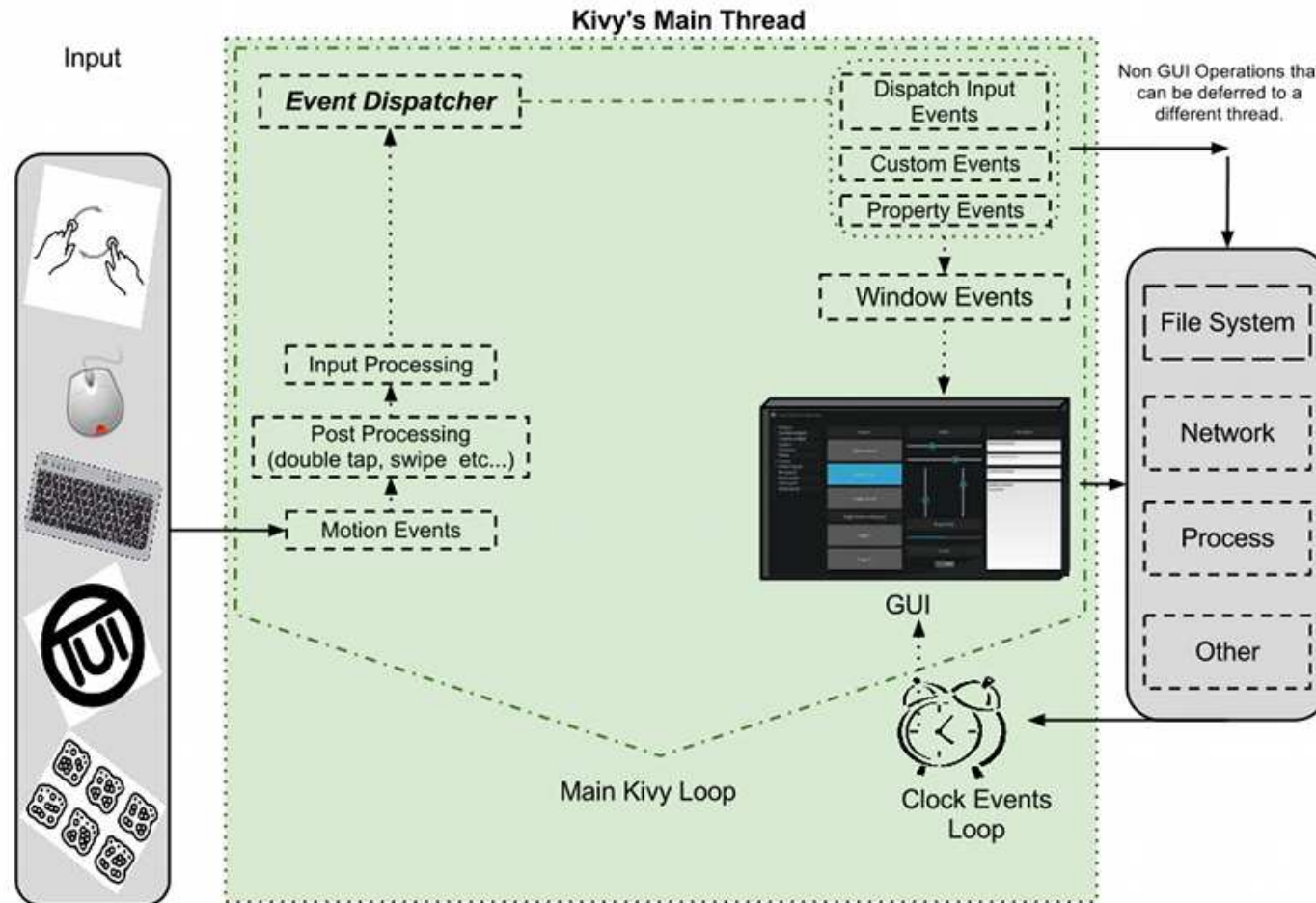


<https://kivy.org/doc/stable/guide/architecture.html>

- Events (next slide)
- Widgets (next slide)
- Kv language
 - <https://kivy.org/doc/stable/guide/lan g.html>



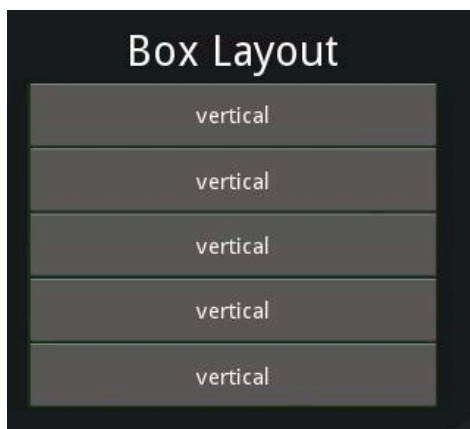
5] Kivy - Towards events



- <https://kivy.org/doc/stable/guide/events.html>

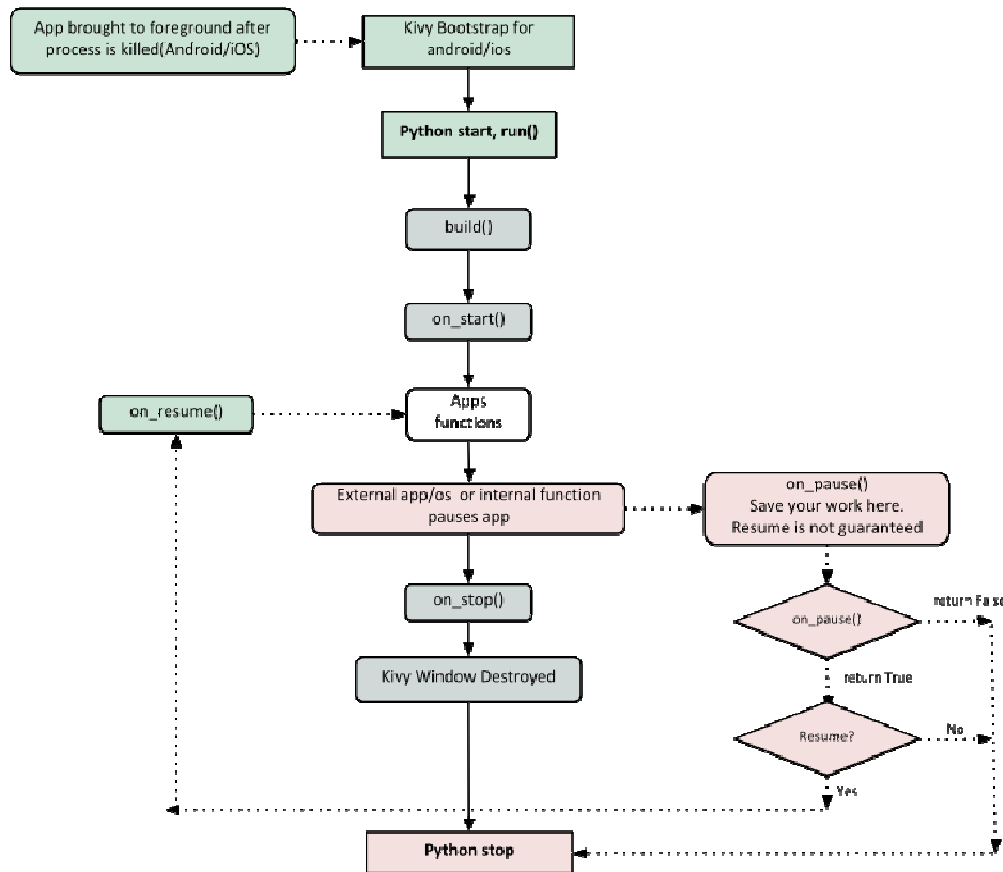
5] Kivy – Widgets & Layouts

- Widgets
 - Widgets are **user interface elements** that you add to your program to provide some kind of functionality. They may or may not be visible.
 - Examples would be a file browser, buttons, sliders, lists and so on.
 - Widgets receive MotionEvent.
- Layouts
 - You use **layouts** to **arrange** widgets.
 - It is of course possible to calculate your widgets' positions yourself, but often it is more convenient to use one of our ready made layouts.
 - Examples would be Grid Layouts or Box Layouts. You can also nest layouts.



<https://kivy.org/doc/stable/guide/widgets.html>

5] Kivy App Life Cycle



Helloworld example

```
import kivy
kivy.require('1.11.1')

from kivy.app import App
from kivy.uix.label import Label

class MyApp(App):

    def build(self):
        return Label(text='Hello world')

if __name__ == '__main__':
    MyApp().run()
```

<https://kivy.org/doc/stable/guide/basic.html>

6] First session

- Step 1: Choose your language (Python, Java, ...) and
- Step 2: Find your development framework (tech stack),
- Step 3: Install your development environment
- Step 4: Start discovering the environment
 - Understand the framework and the APIs (widgets, ...),
 - Make a tutorial
 - Make a tiny app,
 - ...

Useful links and references (to be completed)

Kivy related

- Installation <https://kivy.org/#download>
 - For windows, a preference for Conda
- Api: <https://kivy.org/doc/stable/api-kivy.html#>
- Kivy specific for android : <https://kivy.org/doc/stable/guide/android.html>
- **Create a package for Android**
- <https://kivy.org/doc/stable/guide/packaging-android.html>
- **Create a package for IOS**
- <https://kivy.org/doc/stable/guide/packaging-ios.html>

Useful links and references (to be completed)

Python related

- Pyjnius is a Python library for accessing Java classes
 - <https://pyjnius.readthedocs.io/en/stable/android.html#android>

Java related

- Pure Android development
 - Android studio (Java) <https://developer.android.com/studio>

Navigation app related

- <https://docs.mapbox.com/help/tutorials/android-navigation-sdk/>
- <https://github.com/tito/kivy-gmaps>
- <https://github.com/kivy-garden/garden.mapview>
- <https://www.mindk.com/blog/build-a-geolocation-app/>

Q&A