

ALBANIAN LANGUAGE APPLICATION

ANDIA TAFA

BACHELORS DEGREE

EPOKA UNIVERSITY

June 2014

ALBANIAN LANGUAGE APPLICATION

By

ANDIA TAFA

Thesis submitted to the Faculty of Architecture and Engineering, Epoka University, in Fulfilment of The Requirement for the Degree of Bachelor of Science

June 2014

DEDICATION

To my teachers and professors who have taught me the importance of education.

ABSTRACT

Faculty of Architecture and Engineering

Advisor: Igli Hakrama

As Albanian language is considered challenging to be learnt, this application provides some very basic information of the language in a fun and interactive way, in order to make it easier for the users to benefit in their knowledge.

The purpose of the application is to help everyone who wants to learn a bit of Albanian language, improve their grammar and deepen the knowledge for further studies of this area. It is designed in a way of a screen manager where users can switch between screens very easily. Each screen contains information for a different area of the language, starting with the abstract, then with the alphabet, continuing with simple words and phrases and finally with some verb conjugation in three different tenses; past – present and future. Even though the ‘lectures’ contained in this application are quite basic, they are very useful for someone who is a complete beginner and wants to start learning.

I have tried to present everything in a nice interactive way using photos and colorful text with two different fonts. Also the letters ë and ç are available with necessary explanations.

ABSTRAKT

Fakulteti i Arkitektures dhe Inxhinierisë

Udhëheqës: Igli Hakrama

Teksa gjuha shqipe konsiderohet e vështirë për tu mësuar, ky aplikacion ka në brëndësi disa informacione dhe mësimë bazë të gjuhës që paraqiten në një mënyrë interactive për tu mësuar.

Qëllimi i këtij aplikacioni është të ndihmojë këdo që do të mësojë sadopak shqip, të përmiresojë gramatikën dhe të thellojë njohuritë e tij për mësimë të mëtejshme të gjuhës.

Është dizenuar në një mënyrë të tillë, ku përdoruesi mund të lëvizë nëpërmjet ekraneve, ku cdo ekran përfaqëson një kategori të caktuar të gjuhës shqipe, si përshembull; alfabeti, fjalët më të përdorura, shprehje dhe në fund gramatikë me zgjedhimin e foljeve në tre kohë të ndryshme; e shkuar – e tashme dhe e ardhme. Edhe pse mësimet e këtij aplikacioni janë tejet të thjeshta, janë mëse të dobishme për dikë që nuk ka asnjë njohuri të gjuhës shqipe dhe do të mësojë.

Jam përpjekur të paraqes cdo gjë në një mënyrë të këndshme, të sqaruar me figurat përkatëse dhe tekst me font dhe ngjyra të ndryshme. Gjithashtu germet Ë dhe Ç janë paraqitur me shpjegimet përkatëse.

ACKNOWLEDGEMENTS

I would have never been able to prepare this thesis without the great work of all my professors during these three years at Epoka, my supervisor mr.Igli Hakrama for his work and dedication, and all my friends – the extraordinary people I've met. A special thanks goes my friend Nertil Zhuri for his extraordinary support during my studies and encouraging me to always keep going! Last but not least, I am thankful to my family for giving me so much and helped me become the person I am today.

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Epoka University or other institutions.

Andia Tafa
19 June 2014

Table of Contents

DEDICATION	ii
ABSTRACT	iii
ABSTRAKT	iv
ACKNOWLEDGEMENTS	v
DECLARATION	vi
TABLE OF CONTENTS	vii
LIST OF ABBREVIATIONS	viii
CHAPTER 1	9
Introduction.....	9
CHAPTER 2.....	12
Literature Review.....	12
CHAPTER 3.....	16
SOFTWARE ANALYSIS AND DESIGN	16
3.1 Brief problem statement.....	16
3.2 Functional Requirement.....	18
3.3 Non Functional Requirement	18
3.4 Functions.....	19
3.5 Software Requirement.....	19
3.5 Hardware Requirements.....	19
CHAPTER 4	20
System modeling and Design.....	21
Simmilar Work.....	25
CHAPTER 5.....	26
Conclusion and Future Work	26
CHAPTER 6.....	27
References.....	27

LIST OF ABBREVIATIONS

.kv – File extension where all the style properties are stored

.py – Python file, extension of the main file.

App – application. Used to call in the main class

Cols - columns

Pos - position

CHAPTER 1

INTRODUCTION

“Albanian is an Indo-European language spoken by approximately 7.4 million people, primarily in Albania, Kosovo, the Republic of Macedonia and Greece but also in other areas of the Balkans in which there is an Albanian population, including Montenegro and Serbia (Presevo Valley). Centuries-old communities speaking Albanian-based dialects can be found scattered in Greece, southern Italy, Sicily, and Ukraine.

For this and other reasons, there are several people who are willing to learn even a bit of this language.

With the development of Desktop and Smartphone applications, almost every language – especially the most popular ones – has a specific app to help the users gain language skills.

This application provides its users with basics knowledge of this language, words and grammar.

I have designed a kivy app for this reason. It is build in a form of a screen manager, a window with a few buttons below, where users can click to switch on different screens (or windows).

There are five different buttons below. First one is titled “Pershendetje”. It is the main screen of the project which serves as well as the ‘about, section. It has the title and a few lines of information displayed. Then, user can proceed in learning the alphabet. Each letter is represented by a button for better understanding. This option is made especially for users whose native language doesn’t have Latin letters. As well as to distinguish a feature of Albanian alphabet of “double letters”, this option is made for users not to confuse them and learn those letters better.

The other page displays some of the most commonly used Albanian words with English translation. There are also some pictures displayed for each category of words. They are aligned in different positions in order to have a better visualization. Same goes with the next screen – however there are displayed some of the most common expressions (in sentences). Also the images are arranged to fit the explained sentences with English translation. The last screen is dedicated to grammar. More specifically, simple verbs and personal pronouns explanations with given examples.

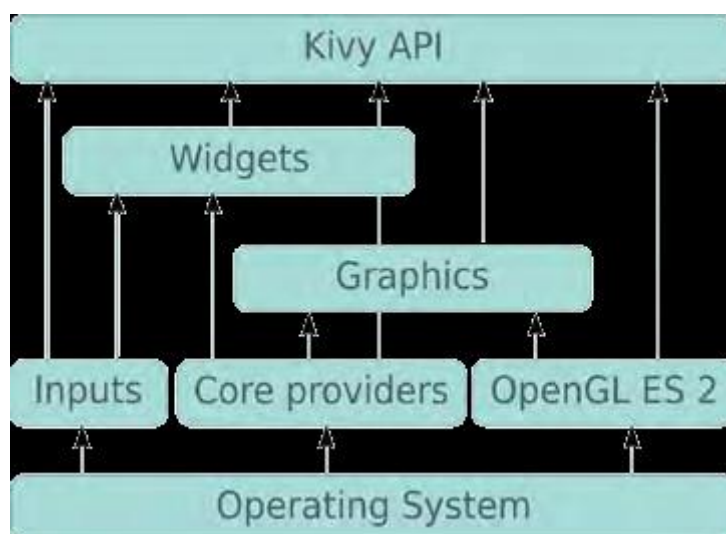
CHAPTER 2

LITERATURE REVIEW

This thesis application is built using Kivy Technology: a Python (<http://python.org>) / OpenGL ES 2 framework, adapted to create Natural User I interfaces, and run on Windows, Mac OSX, Linux and Android. Kivy has been released under the LGPL: you can freely use it for open-source or commercial product.

It's under the license of MIT and free to use, with a stable framework and well developed API. The graphics engine is built over OpenGL ES 2, using a modern and fast graphics pipeline. The toolkit comes with more than 20 widgets, all highly extensible.

Kivy is an evolution of the PyMT (Hansen et al., 2009) project: the graphics engine and Widgets library have been completely rewritten in order to provide much better rendering performance and to introduce property bindings that connect object properties to graphics primitives. The API has been normalized; inputs, core providers and events model have not changed, with the exception of adding new ones for Android support.



The Kivy framework is written in Python and Cython (<http://cython.org/>). Kivy uses Cython to accelerate all the components related to graphics rendering, matrix calculation, and event dispatching and properties operation.

By using Kivy API, your application will be neither OS nor device dependent, and will be able to run on any platform or devices. You can use the high level widgets, or do your own graphics using low-level abstraction.

The graphics engine is written completely in Cython. Python is great for rapid development, but Cython is used better to have the maximum performance with the same flexibility.

Kivy have been accepted inside the Google Summer of Code (GSoC) under the Nuigroup (<http://nuigroup.com>) umbrella.

Kivy goes beyond a framework for creating simple apps for your phone or desktop. It can interface with the phone's hardware and do advanced rendering. It has access to rendering engines. It can do so much more than a simple web app could do.

- **Camera:** There are built in adapters for smartphone cameras. This is one of the first requirements that will have me jumping over to creating a native app, rather than creating a responsive website. Though there are, in theory, some HTML5 inputs for cameras, the coverage for this in browsers is spotty at best.
- **Video:** Kivy includes modules for reading and playing videos. True, this can be done through a web app, but the experience can sometimes be shaky. Older phones might try to play the video without taking up the whole screen, or the memory usage might be higher than it should be.

- **Multitouch and gestures:** Kivy has a robust module for input, allowing for multitouch and gestures.
- **Visual effects and OpenGL:** Because Kivy has access to OpenGL, it can perform a large array of visual effects and do quite a bit of rendering. Kivy also uses PyGame, so creating games becomes much easier (though you do have to rewrite existing PyGame code for Kivy). PyGame comes with modules for drawing shapes, rendering, dealing with colors, and playing music (as well as blowing things up, naturally).
- **Asynchronous network requests:** One important feature of any native app, at least for people like me who travel in and out of network areas, is the ability to finish requests when a network is available. I like being able to work on my phone while in a tunnel, then have it sync my work to the cloud when I hit a station.
- **UI features:** Kivy comes with a number of widgets and controls that are beautifully designed. This can be a real benefit to your project if you're trying to prototype quickly, and don't yet have access to a designer.

What You Need to Know to Successfully Develop Kivy Apps

You should have a pretty solid grasp on Python. Learning Python while trying to learn Kivy might end up confusing the beginner. You don't need to be an expert Python programmer, but you should be comfortable delving into docs and using APIs.

You should also be familiar with MVC patterns. You can write a Kivy program without them, but it might drive you slowly mad as the program grows.

Layout in Kivy

Kivy comes with a special language for defining layout. This allows you to keep your logic and presentation separate. Kv (the layout language) isn't complex, but you should review the documentation briefly before diving in.

A Kv file looks something like this:

<ClassName>:

LayoutType:

WidgetType:

pos: self.center_x -5, 0

size: 10, self.height

LayoutType2:

font_size: 70

center_x: root.width /4

top: root.top -50

text : "0"

Where You Can Learn More

Obviously, the best place to start is kivy.org! There, they have documentation, example projects, and tutorials.

CHAPTER 3

SOFTWARE ANALYSIS AND DESIGN

3.1 Brief problem statement

Learning a new foreign language nowadays is not just a necessity – but somehow also becoming a trend. Albanian language is one of the earliest languages of Europe, special in its kind, and challenging to be learnt. For native people – as in every country – it's more than necessary to read speak and write properly the mother tongue, as much as it is for foreigners to learn in order to interact. Everyone who, for different reasons is willing to learn Albanian – or beginners in this language, made me think of developing something that would come to their help. A simple desktop/mobile app with very basics of this language, words and grammar is provided for these people to make their learning fun and interesting and moreover – effective!

Numerous books and literature has been written in decades – however, in a simple app, the very main things can be not only learnt, but also never forgotten.

As stated previously, the idea of this application is to 'teach the users the Albanian language.

User profile and characteristics

There are no “restrictions” when it comes to users. Absolutely anyone who can use a computer and a mobile can ‘navigate, surf and explore the app. Everyone can use and benefit even a little thing from it. A will and desire or even a necessity to learn Albanian, is more than enough for anyone to use the program.

The application – especially the intermediate version – is more likely to be used from students, giving suggestions for their spell check in writing or grammar correction. This version will be compatible with what is being taught to school, so it will serve to students to complete and fulfill their written work in the very best way – with no logical or syntax errors in writing.

Obstacles may occur when not finding what they really need. As the application is being developed in the first step – the information provided is simple, basic and somehow limited. However this doesn’t lack anything from its usefulness.

Normal User:

- Whole application sections and subsections
- Learning area

Users will not have the need to have a personal profile. The application is for learning purposes. There will not be a need for scores, saving data or anything similar – just the platform with the provided lessons and information and a recommended note-taking book

3.2 Functional Requirements

- user functionalities

- a) Install kivy to run the app
- b) Get the app
- c) Surf through the app
- d) Learn Albanian from the app
- e) Quit
- f) Uninstall at anytime

3.3 Non – Functional Requirements

- Product Requirements:

- Reliability
 - The application should be fast enough when being used.
- Usability
 - User friendly interface
 - Easy to use the menu
 - Clarity of the lessons provided
 - Security and Performance

3.4 Functions

The main functions of the application: Provide the user with certain knowledge.

3.5 Software Requirements

In order to run this application, the kivy platform should be installed first. It runs on Windows, Linux and Mac OS. For windows, a 'portable package' is provided on kivy official website and you don't have to install anything 'system' wide. Just unzip it and run. The installation method is very easy because it bundles python interpreter together with kivy environment and libraries. The users don't need to have any knowledge on Python programming language to run the application. After getting kivy, everything is compiled at any moment.

3.6 Hardware requirements:

For this kind of system the basic hardware requirement is the laptop/PC. Kivy is available for both 32 and 64 bit processors and they require about 72-80 Mb of free space

CHAPTER 4

SYSTEM IMPLEMENTATION

Kivy language is restricted in not using widgets together with other .uix library.

This means – restrictions in designing the application just the way I want – instead, it should be done in a way that it adapts kivy libraries and platform restrictions.

While working with my application, I managed to build and learnt some interesting features of kivy which were not mentioned in the official site of the documentation.

Some of them were 1.creating a new line inside of a label, 2.connecting more than one font, 3.size and reposition of the image using negative numbers, 4.display of alphanumeric characters such as letters e and c.

New Line: there are several ways to do it, however they were explained in using them inside the `print('newline\n');` function inside the main class. However, I had already written my text in Labels form in .kv file. Instead, I used multiple labels and changes `size_hint` property in order to give the idea of two or more lines.

Connecting Font: kivy allows font connections in labels using `font_name` attributes but many difficulties occur if one is not familiar with the language.

One of the steps, is to register the `font_name` in the `LabelBase` class like this:

```
from kivy.core.text import LabelBase

LabelBase.register(name="RobotoCondensed",

fn_regular="data/fonts/RobotoCondensed-Light.ttf",

fn_bold="data/fonts/RobotoCondensed-Regular.ttf",
```

```
fn_italic="data/fonts/RobotoCondensed-LightItalic.ttf",
fn_bolditalic="data/fonts/RobotoCondensed-Italic.ttf")
```

Then, proceed to create the settings part in the main.py file in order to ‘call, this font in the project.

I have to admit that adapting the application idea with the platform restriction, is the hardest part of developing and writing the code. Because you never know what difficulties you encounter while coding and what ‘surprises you might get while running the code that it’s supposed to work properly.

```
KIVY_FONTS = [

    {

        "name": "RobotoCondensed",

        "fn_regular": "data/fonts/RobotoCondensed-Light.ttf",

        "fn_bold": "data/fonts/RobotoCondensed-Regular.ttf",

        "fn_italic": "data/fonts/RobotoCondensed-LightItalic.ttf",

        "fn_bolditalic": "data/fonts/RobotoCondensed-Italic.ttf"

    },

    {

        "name": "FreeSerif",

        "fn_regular": "data/fonts/FreeSerif.ttf"

    }

]
```

And then, again in the main.py file – to register the upper font:

```
for font in settings.KIVY_FONTS:

    LabelBase.register(**font)
```

In my thesis, I modified the given example in order to use another font called “Hopeless_Place”. I have proceeded by declaring which font to use in the test.kv file, by declaring a ‘global label’ <Label>: font_name: 'Hopeless_Place'. By doing this, each text would take this font by default. However by using other font properties like [b] – for bold and [i] for italic, the text takes the other font (RobotoCondensed) property. In this way, I chose which font to use in here.

Position of the images: Kivy divides the screen of the ‘screen manager’ in a ‘trigonometric circle way’, where the (0,0) origin point is right in the very middle of the screen. Then to position any image, you have to use coordinate numbers together with their signs – for example, (-100,-100) displays an image below left. In the given documentations, image alignments is explained by using keywords i.e “center, left, right”, however I found this coordinate style positioning quite accurate.

Alphanumeric: kivy by default supports characters like E or C, however this doesn’t happen when writing in buttons or by using a connected font (i.e Hopeless_place).

The platform is itself is new and being tested all the time. So many bugs and lack of diversity to do different stuff can occur.

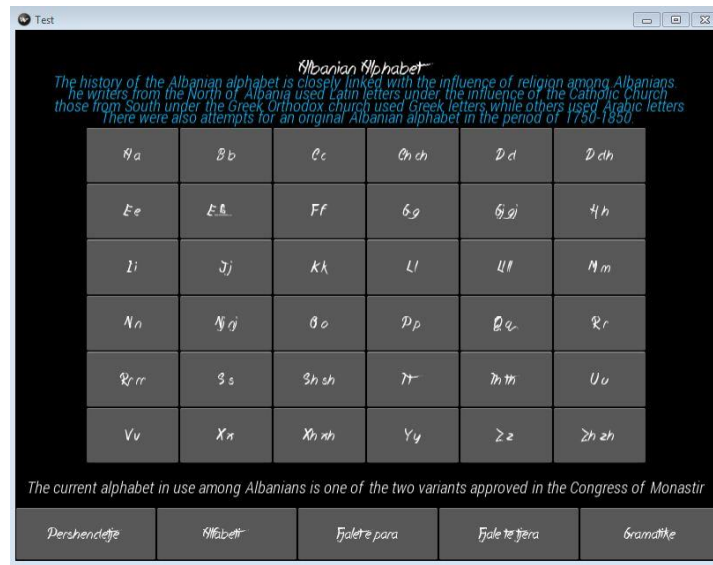
While implementing this application, I've taken some screenshots of five screens provided. Starting with the About (pershendetje), Alphabet(alfabeti), Fjalet e para (first words), Fjale te tjera (other words) and Grammar (gramatike).



Figure 1. Main Page

The alphabet letters are built using buttons. The size, number or columns is declared in the very beginning of the screen design, right after the screen_id.

Padding is also set, in any case of window resize.

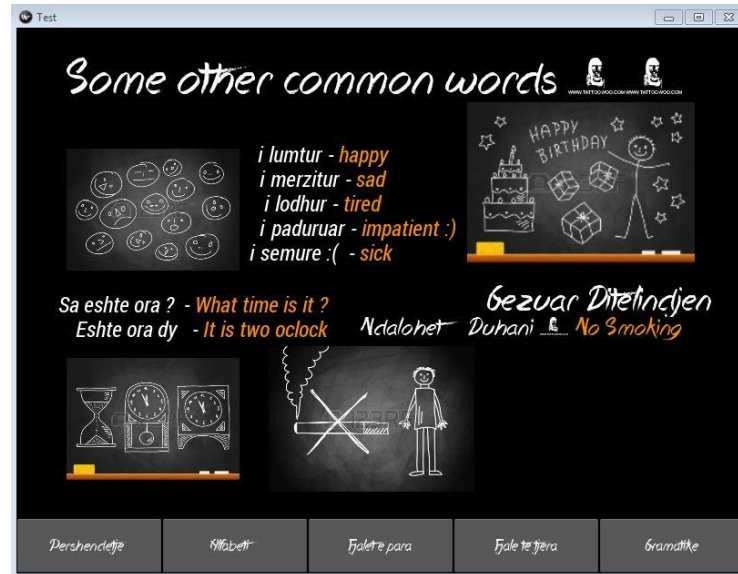
Figure 2. the alphabet page

This screen was captured while I was building it. I was still experimenting with image position, and that was the moment where I figured out how the positioning using coordinates works on kivy.

Figure 3. First words

In this section, there are not just words, but most common expressions that are used in daily life. Figures are used for common demonstration. In the final version, there are used also some smaller sized images and text.

Figure 4. Some other common words



This last page, was captured while being under construction. The images below were aligned by using the same y coordinate and just changing the x position by adding 150 pixels to the x coordinate.

Figure 5. Grammar



Similar work

So far, there have already been developed a few mobile application related to Albanian language. However, none of them is related directly to language and grammar learning.

One of these apps in Albanian language is the Virtual Teacher – educative for kids aged 3 to 12 years old. It's an interactive application that enables them through games and playing, the learning of basic words of different categories. This application, at the moment is available only in iOS platform.

Another related Albanian language application is the iThenie – available in iOS platform which includes quotes and famous sayings of well known authors and personalities in Albanian language.

Developers have also made an Alphabet app – currently being improved to add the sound feature.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This was all about what I have implemented so far as a thesis. However, I would totally continue my work to contribute more for the cause of Albanian language.

There is so much to be done, from the teaching of the very basics, up to professional use of the language in written media or schools for teaching purposes.

Different apps can be made, especially to help students of State Matura in the Literature and language exam after high school. Therefore the interest is quite high.

CHAPTER 6

REFERENCES

[1] - Sommerville Ian, *Software engineering, 9th edition*. Pearson Education 2011.

[2] - Albanian Language – Wikipedia
http://en.wikipedia.org/wiki/Albanian_language

[3] - Kivy platform
<http://kivy.org/#home>

[4] - Virtual Teacher – application
https://play.google.com/store/apps/details?id=com.gmail.besnikrama.virtual_teacher_android

[5] on kivy: <http://www.informit.com/articles/article.aspx?p=2170160>

