Jordan University of Science and Technology
Faculty of Computer and Information Technology
Department of Computer Engineering

CPE 592: Graduation Project Final Report

[R&D of new Arabic Keyboard layout using AI]

Yahya Moh'd Khier Khaleel Al-Mubaideen 83755

Supervised By
Dr. Muhannad Quwaider

Date: December 2018

Table of Contents

Abstract

Today, handheld devices especially smartphones and Tablets are extensively used to write emails, notes, messages and long texts. However, the arrangement of keys in the current soft keyboards is not optimized to facilitate rapid and ergonomic typing, rather it is just a “reduced” version of the industrial standard keyboard layout used in Typewriters and full-size keyboards, which has been developed with the premise of using four fingers in each hand to type letters & numbers and one of the thumbs to the space bar, raising the fact that it is impossible to utilize handheld devices’ soft keyboard in the same manner. We set out to take a closer look and try to find a better design.

The focus is on Research, Analysis, and Implementation of a software system to achieve an optimized Arabic keyboard experience for touchscreen with an emphasis on quality of result. Regarding the scope of Computer Engineering, our project falls under Human-computer interaction (HCI) field of study.

Using Artificial Intelligence, the system will guess the answer to the question “to achieve such a design, what is the best letter placement in a layout?”, say, modelling and simulating an optimization problem, from the category of the facilities location problem.

I. Introduction

Many computer companies have developed more than 20 variants for the Arabic Keyboard layout, some of them in the late 1980s are the layouts of Microsoft Arabic word, Apple Mac, Sakher, AMEER, ALIS and Nafitha therefore to solve this chaos the Arab Standardization and Metrology Organization (ASMO) developed a standard for the Arabic Keyboard layout called (ASMO 1987), which haven't replaced the good old Typewriter layout (shown in Fig. 1) and its digital counterpart (shown in Fig. 2) for that same reason, it is not worthwhile to invest in changing over because of the low speed gain [1].

Looking at the four figures below one can easily deduce the common pattern (e.g. middle row always contains ك م ن ت ا ل ب ي س ش), for a first glance anyone would think "Since it has always been that way, why change it now? Wouldn't that be a burden on those who are already familiar and trained to type on the infamous layout".

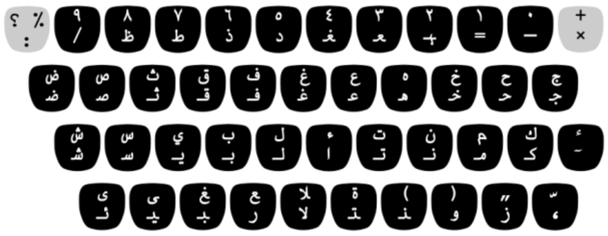


Fig.1: The Arabic Typewriter layout

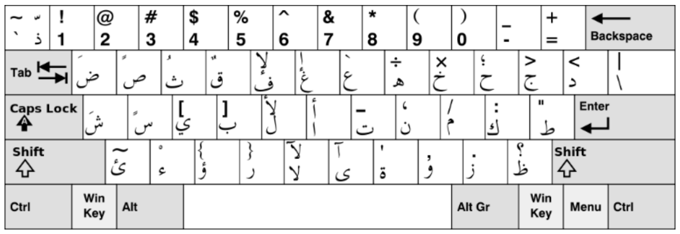


Fig. 2: The IBM PC Arabic Keyboard (most common PC layout)

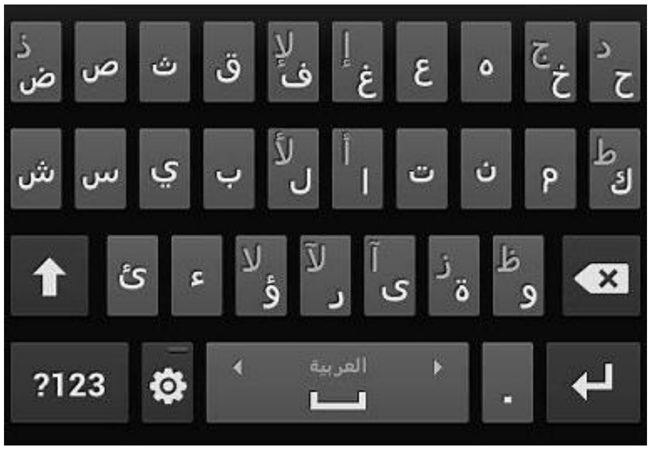
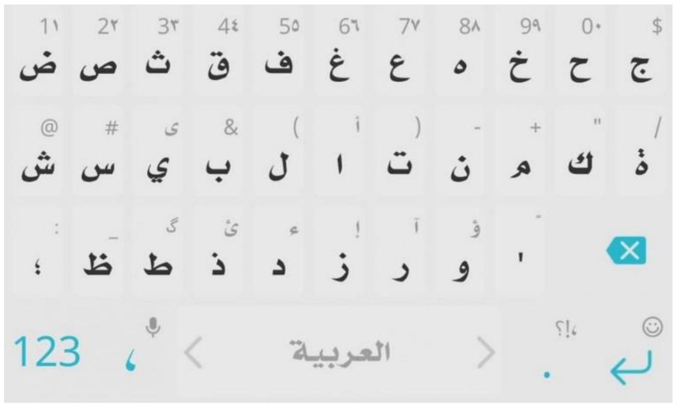



Fig. 3: The Android Arabic keyboard layout

Fig. 4: The iOS Arabic keyboard layout

let me interrupt here and say, it doesn't need Sherlock to figure that out, but what about the limitations of it on touchscreen devices, for once, we are not proposing a complete overhaul of the old layout for all systems, rather, it is only an evolution necessary by the shift imposed by the technology itself.

Listing few of those changes:

- The keyboard is just below the content, on the same screen, and the ergonomics of human hand, and finger movement, raises new layouts where the upper middle section has higher potential, and to be used more often.
- There is no home row to "land on" with your non-thumb fingers while resting the thumbs on the space bar (Touch typing style), due to glass-like surface and absence of tactile feedback.
- Most "if not all" touchscreen typing happens with up to two fingers at the same time, changing the reachability for keys.

Most of the users have difficulties using the current soft keyboard in small screens which is more noticeable for the keys that are directly placed near the lower edges of the screen, this is perhaps in direct connection with the physical attributes and motion of the hand that is holding the device and the fingers that are used to type, as it was for the older generation of handheld devices, namely cell phones and PDAs, the angle of the finger at the joints is greatly proportional to the performance of the user [2].

And although most keyboard designs were sought after for the expert entry rate also known as potential, it is equally important for mobile devices soft keyboards to have a highly successful beginner experience.

The Arabic language is unlike English, because the number of letters is different, lack of Letter case and the text is written from right to left, which affects the placement of letters on the keyboard. To optimize the Arabic keyboard layout, we must take into consideration that hand-held devices are typically small, and hence the keyboard grid will be small as well. Therefore, not all characters can be assigned to different keys. Special characters will have to share the same location and accessed through a popup. The placement of letters on the keyboard must consider the frequency of appearance of single letters, and of pairs of letters as well. In addition, the movement direction of the finger from right to left must be considered in the layout design.

Day in, day out, the number of electronic devices in the touchscreen arena are increasing, for which people use them to achieve their daily tasks via multiple common application such as Facebook, Twitter, WhatsApp, Google and many more, so the need to design a new layout for the keyboard (the users main input interface) in which comfortableness, simplicity, attractiveness, in addition to speeding up the typing process seems to be the logical step forward.

To have an empirical comparison of the different keyboards, in addition to the learning curve for each design, here we define the users' performance in terms of the well-known pair: Typing speed in words per minute (WPM) and Accuracy in percentage of correct entries.

Our main goal is an application that will operate on common platforms: Android and Windows, the former is an interface with special functionality, acting as an input device, the soft keyboard, while the latter is a specialized simulation and optimization problem solver.

First, the system will be built using common programming languages such as C# and Java, using Integrated Development Environments, Visual Studio and Android Studio. We will find the optimized layout using an Evolutionary algorithm in Artificial Intelligence (AI) which is the Genetic algorithm, that is based on taking some populations from the different possibilities in our project the keyboard layouts, whose total number is of the magnitude of factorial of 30 or more, the fitness function we use it to evaluate each generation, and those with highest score are further enhanced while those with low score of fitness will be dropped. Another thing we need in our project is statistics for Arabic characters' occurrence, so we used results from other researchers finding: "Figures 4: Frequency of the Arabic Letters" and" Table 2: Frequency of Arabic Letter Pairs (First letter on the column, second letter on the row)". Tareq M. Malas, et al., Toward Optimal Arabic Keyboard Layout Using Genetic Algorithm [1].

There are two modes relating keyboard layout, which are tow finger and single finger. Our main optimization criteria are typing speed which is different in these modes, for example to increase the typing speed in the multiple finger case, the most frequent pairs of letters should place far from each other, so that when the first letter is being typed another finger is ready to type the next letter. Contrary to this, the frequent pairs of letters in the single finger keyboard must be placed adjacent to each other, so that the finger can move from one letter to another; our design is based on both modes.

Using two fingers or mostly the two thumbs, in this project we will try to create an algorithm that finds the layout that increases typing speed and be flexible for thumb based on the finger transition, distance and size of button and frequency of letter pairs.

Further explanation: a genetic algorithm is a guided random search technique inspired by evolutionary Biology (Goldberg 1989) [3]. It involves a population of candidate solutions to a problem, called individuals, which live, reproduce, and die based on their fitness relative to the rest of the population. The genetic algorithm consists of five phases: initialization, evaluation, selection, reproduction, and competition. A typical genetic algorithm requires two definitions:

1. Genetic representation of the solution domain
2. Fitness function to evaluate the solution domain

To evaluate a solution (Keyboard Layout), five measures are taken into consideration in our object and we will build the evaluation function on them:

1. Finger transition.
2. Row transition.
3. Distance.
4. Size of the button.
5. Probability of characters.

The optimization performed by a genetic algorithm can search a large space for a solution which is close to the optimum with relatively short time. The search space (possible keyboard layouts) is factorial of the count of movable keys (the Arabic character keys) which is: 31! = $8.2\text{x}10^{33}$, which is between (2^112) and (2^113). Evaluating all these layouts is infeasible as it takes very long time.

In order to define a keyboard model, three objects have to be defined: The geometric layout, the character and symbol sets. Concerning the geometric layout, the fact that keyboards might have a different number of rows, columns and keys has to be taken into account. Most commercial keyboards have a layout, which corresponds to a rectangle having rows and columns. Even if those conventional keyboards present offset rows – in a presently unjustified ergonomic way – this is due to ancient mechanical considerations. Each column is not generally assigned to a finger and a so-called home row is defined where the fingers stay in a relaxed position.

Therefore, an abstract keyboard layout may be resumed

in a key matrix, each key being defined by its geometric position, i.e.:

1. Hand (left, right).
2. Column.
3. Row.

The attributes that have been considered are: finger transition, row weight and finger weight.

Keys that are adjacent in a row or column have distance one. The Euclidean distance between keys will be cost for traveling between them. Using a modification of the Fitts' laws that used to model the act of *pointing*, either by physically touching an object with a hand or finger, or virtually, by pointing to an object on a smart phone monitor using a pointing device.

$$t = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{P_{i \cdot j}}{IP} Log_2 \left(\frac{D_{i.j}}{w_{i.j}} + 1 \right)$$

t: the average time cost per key press.

P: the probability matrix of going from one key to another.

D: the matrix of distances between keys.

W: an array of key sizes.

IP: The Fits performance index in bits per second.

N: the size of our alphabet.

Now after set W = IP = 1 to input a corpus probability matrix and a keyboard distance matrix because all keys have same size, and all keys need same time to execute (they trigger same amount of computer instructions).

$$t = \sum_{i=1}^{n} \sum_{j=1}^{n} P_{i \cdot j} Log_2 (D_{i \cdot j} + 1)$$

Now to find the evaluation function we can model the keyboard layout problem in terms of the Famous Quadratic Assignment Problem (QAP), which is a well-known combinational optimization problem and it is used to model the facility layout problem where N facilities must be located in N locations. Through the previous paragraph there are almost 8.223*10^33 layouts.

The used fitness function is the typing time using the keyboard layout. This fitness function takes into consideration attributes that give short typing time such as hand alternation and using the home row. It also takes into consideration attributes that give long typing time such as typing two consecutive letters with the same finger. Smaller fitness values are better. The used fitness is the aggregate of the products of all letter pair frequencies f and the corresponding letter pair typing times t formally:

$$Fitness = \sum f * t$$

The pair typing times are found using a linear regression model (Hiraga et al. 1980) [5].

But, before users can take advantage of the optimized movement efficiency, they have to go through a period of learning during which most of their time is spent on visual search of the target keys (Zhai et al., 2000) [7], rather than stylus movement to select the keys.

For another example, Smith and Zhai (2001) [8], studied the effect of “alphabetical tuning” of an optimized stylus keyboard. They introduced a keyboard layout with an alphabetical bias from upper left to lower right corner of the keyboard while maintaining an optimal layout for the stylus movement efficiency. They found that novice users could perform 10% faster on a stylus keyboard with alphabetical tuning than without.

Although typing can be performed without involving explicit declarative knowledge once it is well-learned, declarative knowledge plays a significant role during the early stages of skill acquisition. Early in the learning process, conscious cognitive processes are heavily involved to understand the nature of the task and how it should be performed. With much rehearsal, the task procedure becomes more automated and requires less attention. According to this well accepted theory, a novice user of a stylus keyboard may type with a disproportionate amount cognitive effort compared to an expert user who may type autonomously. Therefore, at the initial learning stage, users may benefit from better cognitive strategies. Visual search is a good candidate for applying cognitive strategies during an initial learning stage of a new stylus keyboard. If novice users spend a great portion of their time searching for the target keys, then better search and memory strategies for the layout may reduce search time and therefore overall typing speed [6].

Three top down learning methods - alphabetical grouping, word based-grouping, and word generation - were designed to provide users with constructive visual search strategies and improve their memory of the keyboard layout during learning sessions.

With such groupings, the alphabetical bias of the keyboard was highlighted to users and they were given a visual search strategy to reduce their search space during typing. They were told to determine if the target letter is roughly near the beginning, middle, or end of the alphabet, and search sub-areas on the keyboard (e.g. upper-left corner, lower-right, etc.) that corresponded to those divisions to reduce search time.

To improve users’ memory of the keyboard layout, a word-based spatial grouping strategy was used. It aimed to exploit both visuo-spatial and linguistic aspects of typing. Linguistically, the letters were learned by forming words, which are more meaningful than random set of letters. In general, it is easier to remember information without much inherent meanings by associating it with other known objects. One such technique is the mnemonic that converts a group of letters into more meaningful words (Atkinson and Raugh, 1975) [9]. Past research suggests that typing is faster when the text is composed of words rather than random letters, and identification accuracy for a single letter is higher when the letter is presented in a word than when it is presented singly (Reicher, 1969) [10].

Fun or enjoyable experience is very important to product design; but an efficient method may well be ultimately more fun and more enjoyable than a less efficient one. Nevertheless, it is important to keep in mind that physical efficiency should be treated as an indicator of interaction quality not the ultimate design goal. If better user experience can be found in a design that at odds with efficiency, the former should be given greater consideration. Although often ignored, there should be important cultural and linguistic consideration in user interface design [11]. And not neglecting the important cultural and linguistic consideration in user interface design (del Galdo & Nielsen, 1996) [12] (Shneiderman, 2000) [13].

Most of the seven well-known universal design principles are relevant to soft keyboards [4], enlisting three that we will try our best so that the final design has them:

- Principle 3: Simple and Intuitive Use
  Use of the design is easy to understand, regardless of the user's experience, knowledge, language skills, or current concentration level.
- Principle 6: Low Physical Effort
  The design can be used efficiently and comfortably and with a minimum of fatigue.
- Principle 7: Size and Space for Approach and Use

  Appropriate size and space is provided for approach, reach, manipulation, and use.

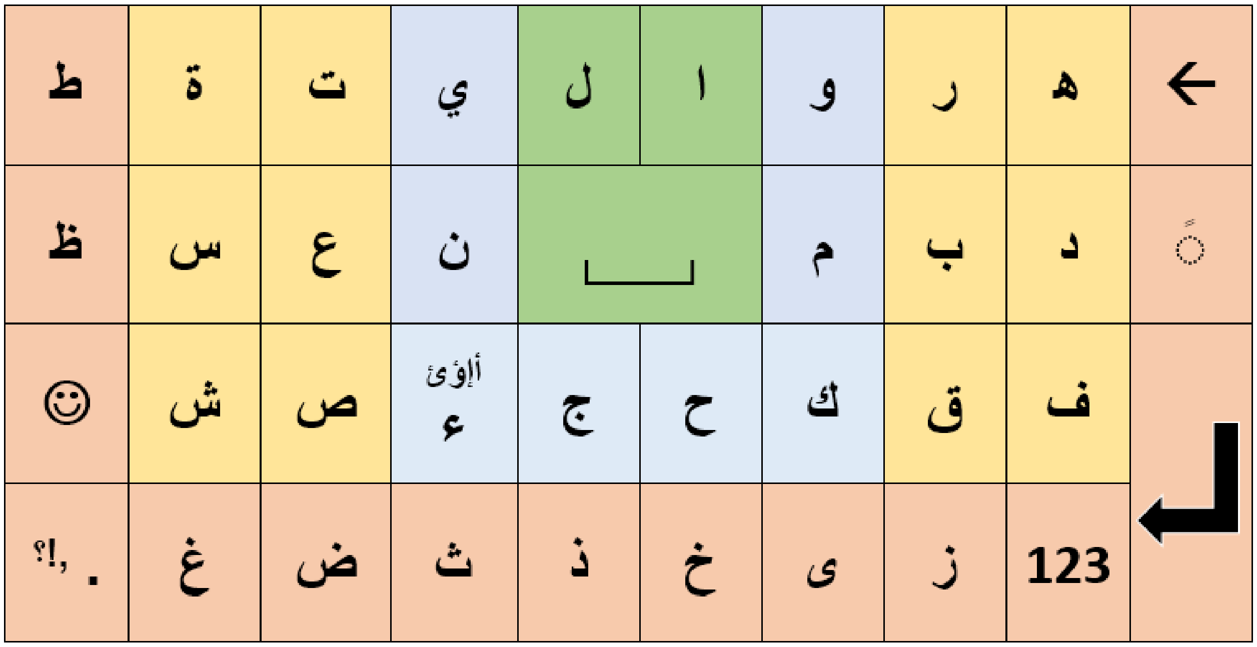
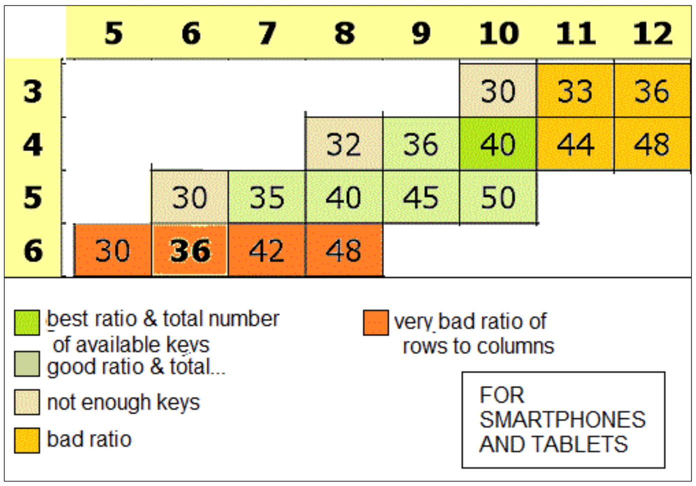


Fig. 5: The initial Arabic keyboard layout (Seed)

This layout is handpicked using:

1- The list and table provided by paper: (TOWARD OPTIMAL ARABIC KEYBOARD LAYOUT USING GENETIC ALGORITHM) [1].
2- Times-table to know how many rows and columns.

3- General knowledge on the problem (from other papers).
4- Common layout (current used layout by smartphones).
5- The coloring is used like a heat map for easy to press (green) to hard (pink).
6- Place the least common used letters at the bottom row.

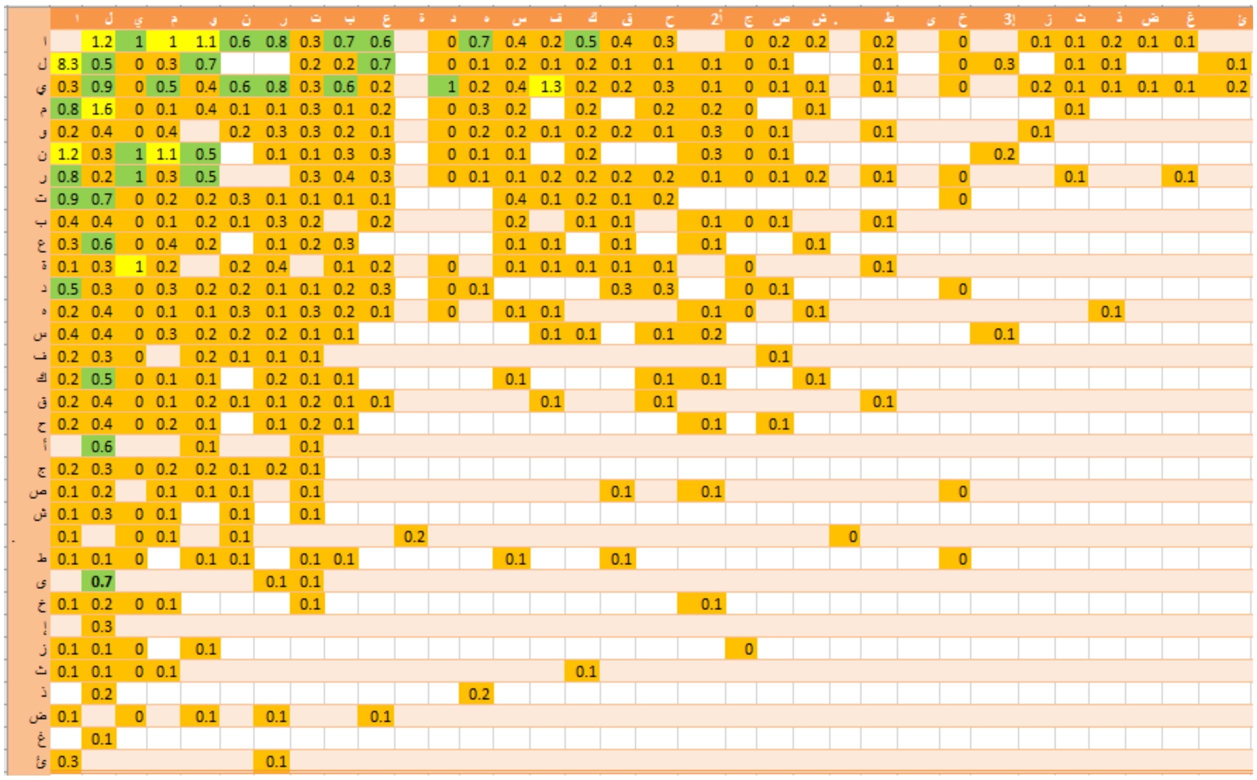


	5	6	7	8	9	10	11	12
3						30	33	36
4				32	36	40	44	48
5		30	35	40	45	50		
6	30	**36**	42	48				

🟩 best ratio & total number of available keys
▨ good ratio & total...
▨ not enough keys
🟨 bad ratio
🟧 very bad ratio of rows to columns

FOR SMARTPHONES AND TABLETS

Fig. 6: Times-table (rows and columns on screen)



	ا	ل	ي	م	و	ن	ر	ت	ب	ع	ة	د	ه	س	ف	ك	ق	ح	2أ	ج	ص	ش	.	ط	ى	خ	3إ	ز	ث	ذ	ض	غ	ئ
ا		1.2	1	1	1.1	0.6	0.8	0.3	0.7	0.6		0	0.7	0.4	0.2	0.5	0.4	0.3		0	0.2	0.2		0.2		0		0.1	0.1	0.2	0.1	0.1	
ل	8.3	0.5	0	0.3	0.7			0.2	0.2	0.7		0	0.1	0.2	0.1	0.2	0.1	0.1	0.1	0	0.1			0.1		0	0.3		0.1	0.1			0.1
ي	0.3	0.9	0	0.5	0.4	0.6	0.8	0.3	0.6	0.2		1	0.2	0.4	1.3	0.2	0.2	0.3	0.1	0	0.1	0.1		0.1		0		0.2	0.1	0.1	0.1	0.1	0.2
م	0.8	1.6	0	0.1	0.4	0.1	0.1	0.3	0.1	0.2		0	0.3	0.2		0.2		0.2	0.2	0		0.1							0.1				
و	0.2	0.4	0	0.4		0.2	0.3	0.3	0.2	0.1		0	0.2	0.2	0.1	0.2	0.2	0.1	0.3	0	0.1			0.1				0.1					
ن	1.2	0.3	1	1.1	0.5		0.1	0.1	0.3	0.3		0	0.1	0.1		0.2			0.3	0	0.1						0.2						
ر	0.8	0.2	1	0.3	0.5			0.3	0.4	0.3		0	0.1	0.1	0.2	0.2	0.2	0.2	0.1	0	0.1	0.2		0.1		0			0.1			0.1	
ت	0.9	0.7	0	0.2	0.2	0.3	0.1	0.1	0.1	0.1				0.4	0.1	0.2	0.1	0.2								0							
ب	0.4	0.4	0	0.1	0.2	0.1	0.3	0.2		0.2				0.2		0.1	0.1		0.1	0	0.1			0.1									
ع	0.3	0.6	0	0.4	0.2		0.1	0.2	0.3					0.1	0.1		0.1		0.1			0.1											
ة	0.1	0.3	1	0.2		0.2	0.4		0.1	0.2		0		0.1	0.1	0.1	0.1	0.1		0				0.1									
د	0.5	0.3	0	0.3	0.2	0.2	0.1	0.1	0.2	0.3		0	0.1				0.3	0.3		0	0.1					0							
ه	0.2	0.4	0	0.1	0.1	0.3	0.1	0.3	0.2	0.1		0		0.1	0.1				0.1	0		0.1								0.1			
س	0.4	0.4	0	0.3	0.2	0.2	0.2	0.1	0.1						0.1	0.1		0.1	0.2								0.1						
ف	0.2	0.3	0		0.2	0.1	0.1	0.1													0.1												
ك	0.2	0.5	0	0.1	0.1		0.2	0.1	0.1					0.1				0.1	0.1			0.1											
ق	0.2	0.4	0	0.1	0.2	0.1	0.1	0.2	0.1	0.1					0.1			0.1						0.1									
ح	0.2	0.4	0	0.2	0.1		0.1	0.2	0.1										0.1		0.1												
أ		0.6			0.1			0.1																									
ج	0.2	0.3	0	0.2	0.2	0.1	0.2	0.1																									
ص	0.1	0.2		0.1	0.1	0.1		0.1									0.1		0.1							0							
ش	0.1	0.3	0	0.1		0.1		0.1																									
.	0.1		0	0.1		0.1					0.2												0										
ط	0.1	0.1	0		0.1	0.1		0.1	0.1					0.1			0.1									0							
ى		0.7					0.1	0.1																									
خ	0.1	0.2	0	0.1				0.1											0.1														
إ		0.3																															
ز	0.1	0.1	0		0.1															0													
ث	0.1	0.1	0	0.1												0.1																	
ذ		0.2											0.2																				
ض	0.1		0		0.1		0.1			0.1																							
غ		0.1																															
ئ	0.3						0.1																										

Fig. 7: The extracted Frequency of Arabic Digraphs (First letter column, second letter row)

II. Professional Practice Constraints

- Economic Constraints; To find optimal solution the AI should be run for days, but I had to settle with few hours.

- Social Values Constraints; Field study and users' feedback is harder than I thought, I asked for the help of individuals to test the resulting keyboard layout on their devices but most refused, it's hard to conduct research in the current Jordanian society, still I found some friends that wished to help and others who I knew from Facebook groups.

III. System Architecture and Design

The final implementation is going to be an Arabic smartphone keyboard app with the layout that achieves great mixture of typing speed, memorability as well as enhancing the overall experience of the Arabic keyboard on smartphones, the layout for which our AI component of the system has arranged its letters.

Output of AI is one of the Inputs of the Android Keyboard.

Final verdict will be shown in presentation day (users feedback).

IV. Software Implementation

Description of the algorithm used:

Any Genetic Algorithm with Evolutionary Optimization parameters can be represented in high-level pseudo-code, like this:

```
set up parameters
initialize a population of random solutions with seeds if available
loop until done
  select two good parent solutions
  use them to generate two child solutions
  place children into population
  immigrate a new solution into population
end loop
return best solution found
```

Evolutionary Optimization is a meta-heuristic, which means that it's a realistic set of guidelines that can be used to design a specific algorithm. There are many design choices that I had to make for example placement of some special keys like the space and enter, others like the choice of seeds (parents)

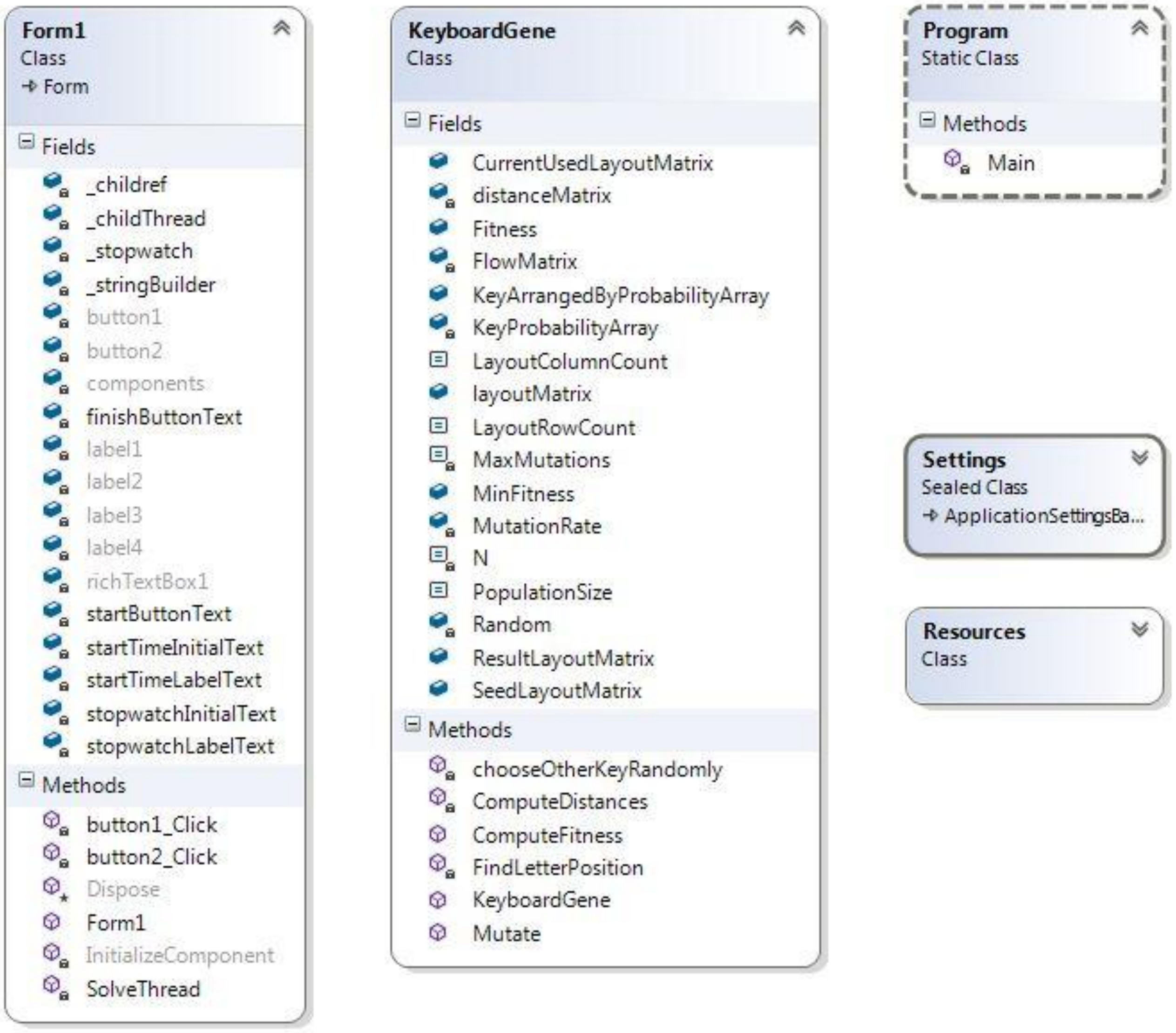


Fig. 8: Class Diagram of the AI

Mainly the program window has buttons that call functions and shows output, a thread is created to do the algorithm.

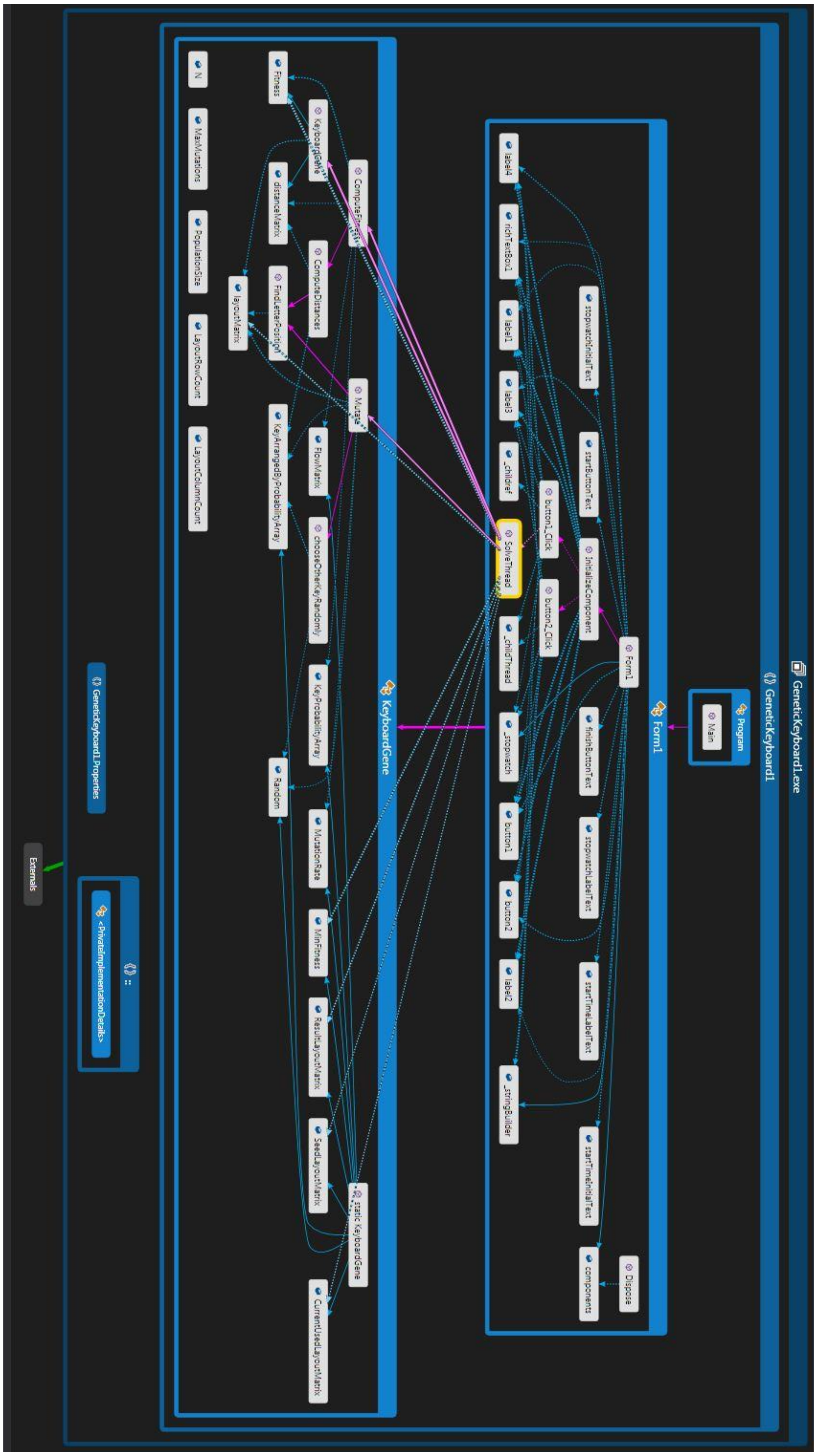


Fig. 9: Code Map of the AI (Fields and Methods dependencies, calls and creations)

Bibliography

[1] Tareq M. Malas, Sinan S. Taifour, and Gheith A. Abandah.: Toward Optimal Arabic Keyboard Layout Using Genetic Algorithm. Proceedings of (MESM 2008), Aug 26-28, Amman, Jordan

[2] Md. Abul Kalam Azad, Rezwana Sharmeen, Shabbir Ahmad and S. M. Kamruzzaman.: Completely Enhanced Cell Phone Keypad. Manarat International University. (IMB2005).

[3] Goldberg, David (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley Professional. ISBN 978-0201157673.

[4] Connell, B. R., Jones, M., Mace, R., Mueller, J., Mullick, A., Ostroff, A., Sanford, J., Steinfeld, E., Story, M. and Vanderheiden, G.: The Principles of Universal Design. The Center for Universal Design, NC State University. 1997. also at http://universaldesign.ie/What-is-Universal-Design/The-7-Principles/

[5] Hiraga Y.; Ono Y.; and Yamada-Hisao. 1980.: An Analysis of theStandard English Keyboard. In Proc. 8th Conf. on Computational Linguistics (Tokyo, Japan). 242-248.

[6] Paul Lee, Shumin Zhai. 2004.: Top-down learning strategies: can they facilitate stylus keyboard learning? International Journal of HumanComputer Studies, Vol 60 (5-6), 585-598.

[7] Zhai, S., M. Hunter, & Smith, B. A. (2000). The Metropolis Keyboard - an exploration of quantitative techniques for virtual keyboard design, Proc. of UIST: the 13th Annual ACM Symposium on User Interface Software and Technology. San Diego, California: ACM. 119-128. 25. Zhai, S., Hunter, M., & Smith, B. A. (2002). Perform.

[8] Smith, B. A., & Zhai, S. (2001). Optimised Virtual Keyboards with and without Alphabetical Ordering - A Novice User Study, Proc. of INTERACT 2001: Eight IFIP Conference On Human-Computer Interaction, Tokyo, Japan, (July 9-13), 92-99.

[9] Atkinson, R. C., & Raugh, M. R. (1975). An application of the mnemonic keyword method to the acquisition of Russian vocabulary. Journal of Experimental Psychology: Human Learning and Memory, 104,126-133.

[10] Reicher, G. M. (1969). Perceptual recognition as a function of meaningfulness of stimulus material. Journal of Experimental Psychology, 81, 275-280.

[11] Bi, X., Smith, B. A., & Zhai, S. (2012). Multilingual touchscreen keyboard design and optimization. Human–Computer Interaction,27(4), 352–382.

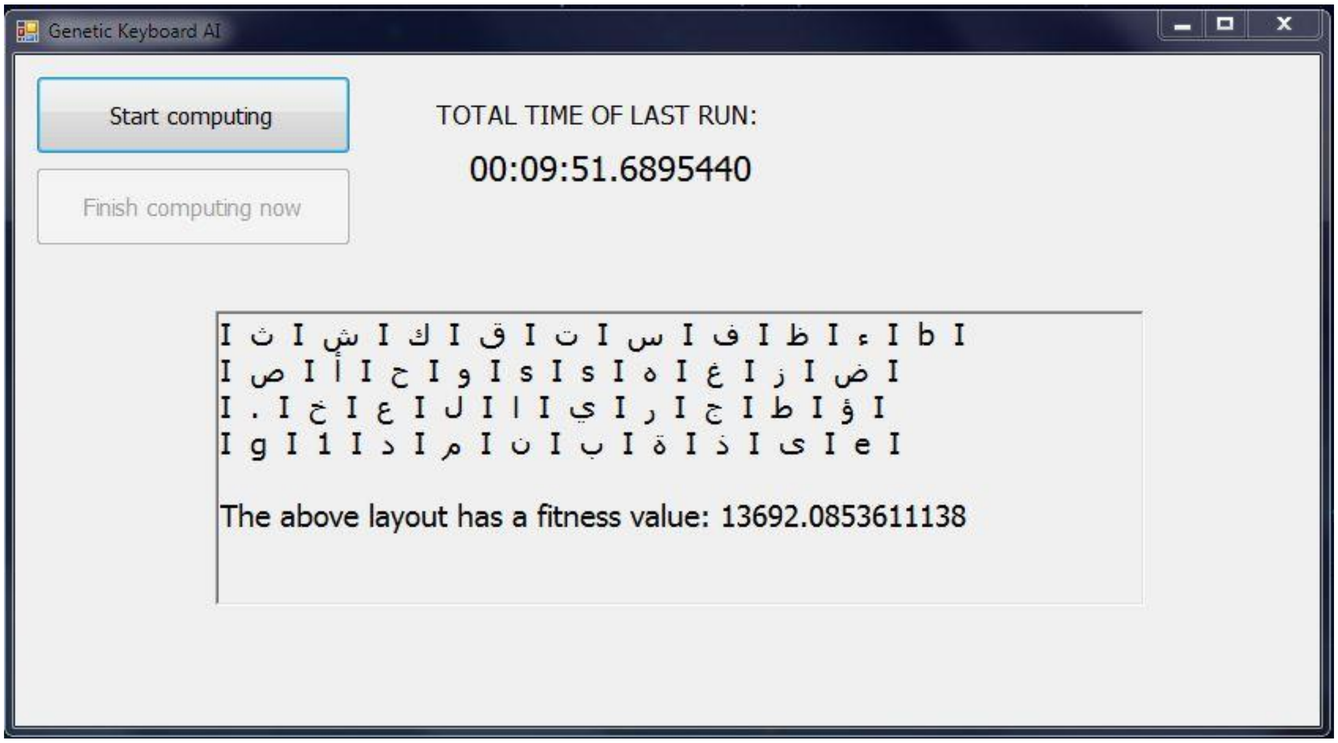
[12] del Galdo, E. M., & Nielsen, J. (Eds.). (1996). International users interface. New York: John Wiley & Sons.

[13] Shneiderman, B. (2000). Universal usability. Communications of the ACM. 43(5), 84 – 91

And many other websites

Appendix A

Samples of the programs:



TOTAL TIME OF LAST RUN:

00:09:51.6895440

```
I ث I ش I ك I ق I ت I س I ف I ظ I ء I b I
I ص I أ I ح I و I s I s I ه I غ I ز I ض I
I . I خ I ع I ل I ا I ي I ر I ج I ط I ؤ I
I g I 1 I د I م I ن I ب I ة I ذ I ى I e I

The above layout has a fitness value: 13692.0853611138
```

Fig. 10: A run of the Artificial Intelligence in a windows C# forms application

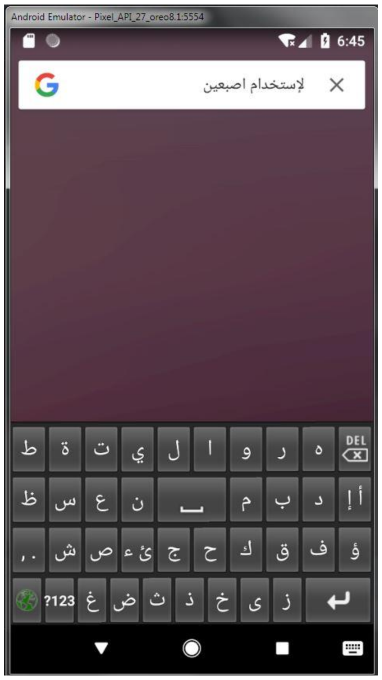


Fig. 11: Two fingers Android keyboard

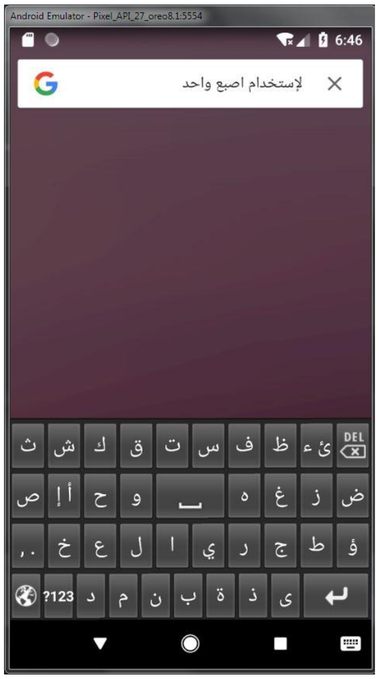


Fig. 12: One finger Android keyboard