ANKARA UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

INTERNSHIP REPORT

GENETIC ALGORITHM APPROACH TO 8 QUEENS PROBLEM

Tuğçe Nur ŞAHİN

15290122

04.09.2020

ABSTRACT

This report covers what has been done and learned in the four-week online internship program in HAVELSAN. Internship; It includes online tutorials and webinars, in-depth learning of the Linux operating system and working on genetic algorithms. The report consists of 3 main sections. These are the introduction, study areas so the 8 Queens Problems and the conclusion sections.

First of all, various informative trainings were provided during the internship. This online training included trainee awareness, cybersecurity, company employee policies, and occupational health and safety. Thanks to these trainings and webinars, many ideas were obtained in the field of computer engineering, apart from an internship project.

Genetic algorithm, which is one of the algorithms frequently encountered in artificial intelligence technology, is among the solutions that add a different dimension to the solution by combining genetics and mathematics. The genetic algorithm, which is the most important part of the internship, has some known problems. Examples of these are the travelling salesman problem and the 8 queens problem. The internship project aimed to understand the genetic algorithm and to develop one of the 8 queen problems, one of the genetic algorithm problems. The 8 queens problem aims that N queens cannot be superior to each other on the NxN chessboard. The project involves solving the problem and designing the interface to that problem. In the first days of the internship, Linux was determined as the operating system and in-depth information about the directory structure of this operating system was obtained.

During the internship, business planning was made by contacting the mentor. Reports and projects have been given in line with these plans. During the internship, how the business works in the company was observed and helped form ideas about project management, meeting processes and basic needs.

This report ends with an overview of the company internship program and general impressions of the work experience. Finally, the report is concluded by showing the bibliography and appendices after the conclusion section.

INSTITUTION INFORMATION

Institutions;

Name : HAVELSAN

Deparment : Training and Simulation Technologies Deputy General Manager

Address : Mustafa Kemal Street 2120 Neighborhood No:39 P.C.: 06510 Çankaya/Ankara

Telephone : 0312 219 5787

E-mail : info@havelsan.com.tr

Web Page (if exists) : https://www.havelsan.com.tr/

HAVELSAN is founded in the year 1982 as a corporation owned by and affiliated to the Turkish Armed Forces Foundation.
Aside from being accepted as one of the largest and leading technology firms of Turkey, HAVELSAN is also a leading trademark in the international market as well with its deep-rooted experience, competent and specialized employees, and high-technology based software-intensive solutions and products.
HAVELSAN does not only develop and produce technology in defence, security and information sectors, along with high technology and software solutions developed in-house, but also combines its own solutions and products with the solutions and products of other firms included in its business ecosystem, thus being able to present end to end solutions to its customers.
Focused on software-intensive systems, HAVELSAN is offering and putting at the disposal of the Turkish Armed Forces (TSK), public administrations and entities, overall private sector and its own international customers its solutions comprised of genuine and original products and systems in the following fields:

- Defense
- Simulation
- Information and Communication
- Homeland Security and Cyber Security

[1]

TABLE OF CONTENTS

1. INTRODUCTION

Artificial intelligence is a spoken field in today's technology. In fact, artificial intelligence, which is considered as the technology of the future, includes various algorithms in its formation. Although the operation of each algorithm is different, it aims to lead to a specific result. One of these algorithms, the genetic algorithm, adopts the closest to the ideal result, not the ideal result.

The genetic algorithm includes expressions such as population, chromosome and gene encountered in biology. While the solution set is called the population, each solution in the population is a chromosome. If we think of chromosomes as parents, new generations are formed as a result of the passage of chromosomes. Thanks to this transition, different solutions emerge. Genetic algorithm is an algorithm that searches not the ideal solution but the closest solution to the ideal. The image below is used to fully understand the structure.

Figure 1.1. Genetic algorithm structure

The genetic algorithm selects the chromosomes closest to the result and crossover them to produce the closest results, but crossover alone is not enough. With the selection part, the best individuals in the population are selected and those who do not come close to the ideal result are excluded by selection. The extinction of short giraffes by natural selection in biology is an example of this. Another factor as effective as the crossover is mutation. Mutation can be defined as the change or loss of a gene in a chromosome or the addition of a new gene. Repetitive solutions as a result of a mutation in new generations are avoided.

The crossover, mutation, and selection are not random, but in a specific order. The order determined for this is shown in the table below.

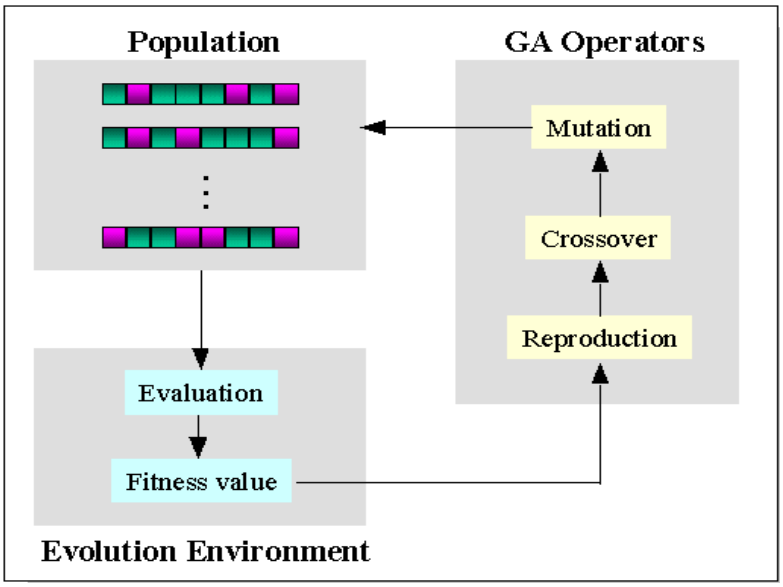

Figure 1.2. Genetic algorithm evolutioon flow

In the genetic algorithm, which has many problems, 8 queens problems in the Linux operating system are solved. This problem, which involves the inability of N queens to dominate each other on the NxN chessboard, is called 8 queens instead of N queens because the chessboards are 8x8.

The two main titles of the project are the game and the sample section. The sample section shows the possible positions of the queen on the chessboard. The game part is based on placing the player in boxes, the queens must not beat each other. At the end of the insertion process, if it fails, the screen displays "Game over!", if successful, "You win!". There is a refresh button in case of unwanted movements during the game. It can be applied in any desired chessboard size within the 2 sections described.

The Linux operating system on which the application is developed differs from Windows and Mac with its directory structure. Linux has a singular hierarchy structure. Other operating systems use directory structure rather than disk usage. This operating system can be embedded in computers or run through a virtual machine. Virtual machine is preferred for the internship.

The internship process that prepares students for business life prepares the student both physically and mentally for the work environment and the desired project. This situation has been reinforced by training and webinars during the internship. If some of the trainings will be given as an example; With occupational health and safety trainings, it was learned what to do in dangerous situations in business life or how to prevent danger. The situations that employees should pay attention to within the company were conveyed through intern awareness trainings. If we give an example from the content of this training; With cyber security trainings, problems that may occur with a USB connection to a computer in the company or problems that may occur over the internet were explained. In addition to many such trainings, webinars were also given. These are not just about engineering, for example, a webinar was held on details of preparing a resume and mistakes made during the interview. Thanks to this webinar, the mistakes made and what needs to be done were learned. These trainings and seminars were given to students at regular intervals and prepared the student for the project. The detailed list of this section is given in Appendix 1.

2. 8 QUEENS PROBLEM

The 8 Queens puzzle was first proposed in 1948 by Max Bezzel, a professional chess player. The problem, which was studied over the years by leading mathematicians such as Gauss, was first presented by Franz Nauck in 1950. Also, Franz Nauck was generalized. The problem with the NxN card and it turned it into the N queen problem.[2]

To explain the problem is that N queens are positioned so that they do not eat each other on an NxN chessboard. The queen can eat all the stones on the horizontal, vertical and diagonal he can afford. In the image below, the situations where the queen can prevail are explained.

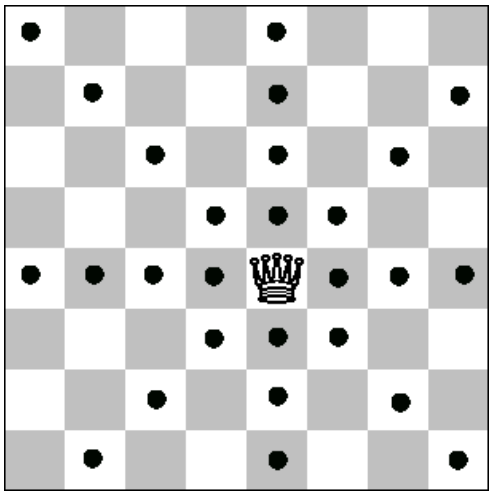

Figure 2.1. Stone situations the queen can eat

The solution of the determined problem was produced with the genetic algorithm in the Linux operating system and the interface was designed. For the code written in Python programming language, the PyQt5 library was used in the interface.

2.1. Operating System Used

Linux is a freely distributable, multitasking, multiuser variant of the UNIX operating system. Linux is an operating system that is developed jointly by many interested and curious people on the Internet and can run on many platforms, especially IBM-PC compatible personal computers, and it is a free operating system. Computers using the Linux operating system are widely used in many areas, especially for private use.

Separating itself from Windows or Mac operating systems with its directory structure, Linux uses the directory structure instead of disk usage in other operating systems. In other operating systems, these structures are called folders. Directories or structures used to collect various files, directories or folders. Thanks to the directory used in the operating request, it provides ease of access and recovery.

2.2. Genetic Algorithm

Genetic algorithms are an intuitive research method based on the principle of generating new sequences of chromosomes in order to realize the solution of complex ordered problems. Holland compiled his studies in 1975 in his book "Adaptation in Natural and Artificial Systems" and Holland used the laws of evolution for optimization problems within genetic algorithms.[3]

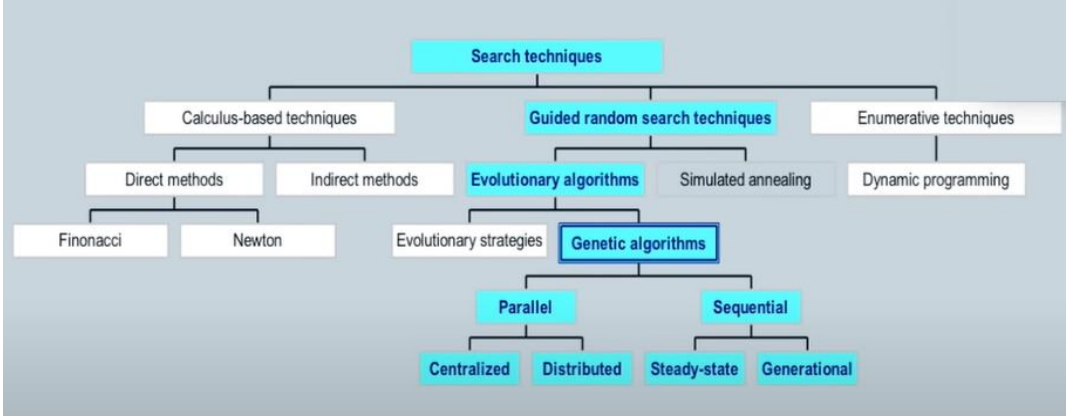

Figure 2.2.1. Genetic algorithms in search techniques

Genetic algorithm has contributed to artificial intelligence projects in terms of usage. While the genetic algorithm, which we can call the intersection of genetics and mathematics, mathematically targets global optimization; takes into account biological evolution in genetic aspects. In order for genetic algorithms to be applied in problem solving, a population of structures (or organisms) with possible solutions to each problem is presented as the solution area of the problem.

To explain the population; while it is used as the name given to a community in biology, it is the name given to the community of individuals that express possible solution information in the genetic algorithm. When examined in more detail, the unit that carries the smallest information about the problem in its structure is called a

gene. Sequences that form part of the information about the solution of the problem by coming together of one or more gene structures are called chromosomes. The genetic algorithm sees each individual as a chromosome.

There are certain conditions that must be met in order to use genetic algorithms effectively:

- Research Space Parameters: Chromosome-gene order should be provided with a certain length by coding in binary system (0,1) or alphabetically.
- Number of Appropriate Solutions: The number of solutions should be reduced with a fitness function in line with the purpose of the problem.
- Parallel and Global Research: In order to serve both products, the research space should be scanned at every point.
- Probabilistic Transition Rule: This rule should be used in order to achieve development between research spaces.

In the genetic algorithm, the solution is not the most ideal but the closest to the ideal is sought. More than one population is possible for suitable solutions in the algorithm. In the algorithm for these

- Combining the features of previously known multiple solutions,
- Changing the order of a previously known solution in a random (coincidental) way,
- Within the population, giving priority,
- It may have mechanisms to push some solutions out of the population.

The functioning scheme of the genetic algorithm is as follows.

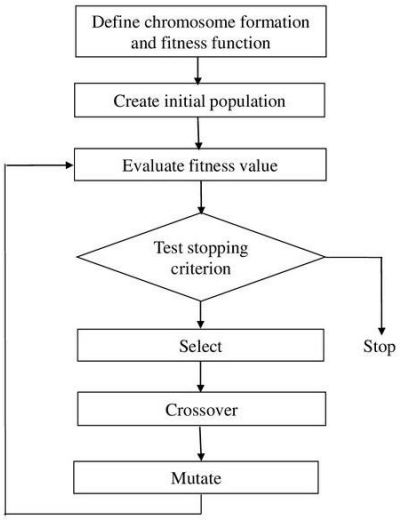

Figure 2.2.2 The flow chart of genetic algorithm

Initially, chromosomes are tested and scored to reach the ideal chromosome. The new population determined is taken into a rule structure and new chromosomes are created from these chromosomes. In this new generation, the successful are re-derived and the unsuccessful ones are expected to move away from the population. This process continues up to a certain point. It is the most successful chromosome left when finished. This is actually the closest solution to the most ideal solution in the most successful chromosome.

The most important factors in deriving new generations are mutation and crossover. It is the process of creating new children by exchanging certain genes in parents. Mutation may also be needed in terms of the reduction of repeating chromosomes as a result of the continuous production of the new generation. As a result of mutation, new generation individuals may be in a better structure than their parents.

The crossover can be divided into single-point crossover and two-point crossover. These distinctions are named as follows.

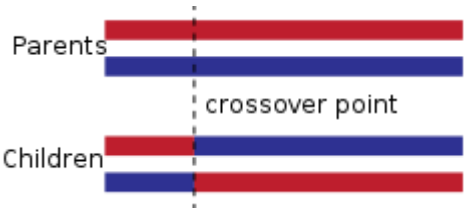

Figure 2.2.3 Single-point crossover

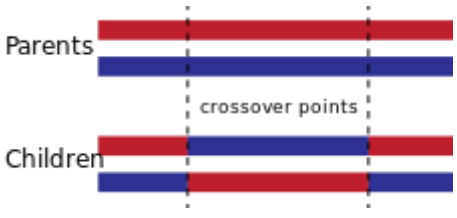

Figure 2.2.4 Two-point crossover

The mutation used for diversity is divided into four as inversion, insertion, deletion and replacement.

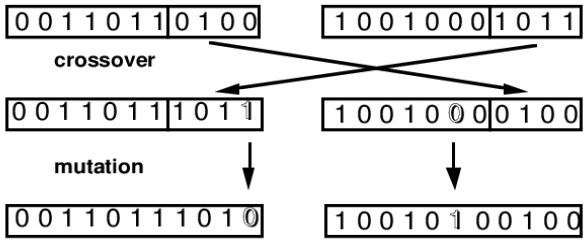

Figure 2.2.5 Mutation and crossover in genetic algorithm

There are parameters for how often to do as a result of crossover and mutation and to restrict flow. These are crossover probability, mutation probability, population size and fitness function.

2.3. 8 Queens Problem Solutions

Various ways have been followed to adapt the 8 queens problems to the genetic algorithm. The position number of the queens in each column on the chessboard will represent the gene. The chromosome will be made up of the positions of all queens on the chessboard. This is clearly understood in the example given below.

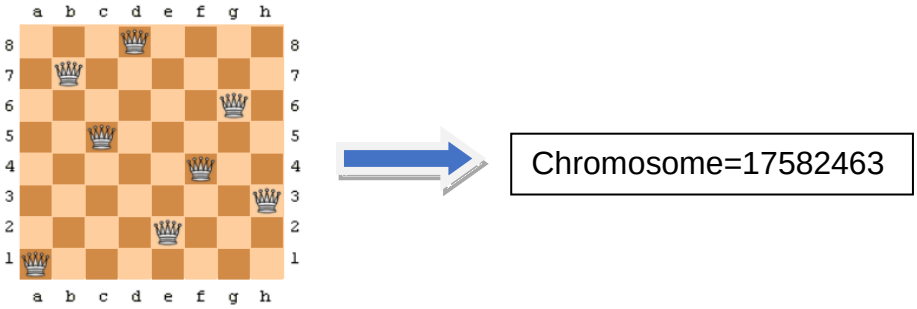

Figure 2.3.1 Gene sequence in the chromosome

The difference between individuals occurs with different gene sequences. A function is needed to show the success rate of these differences in the gene sequence of each individual. This function is problem specific and calculates the fit-to-solution value of each individual in the model. 8 According to the queen problem, this function is the controls function. The controls function is shared in the Appendix 5. In this function, all boxes on the right, left and diagonal are controlled from where the queen is. If a queen is found in controlled places, it is subtracted from the optimum value max_control. max_control gives the probability that the queen is properly placed. Its calculation is N*(N-1)/2. Since it is based on the position of being able to come to

other places after the queen is placed, the open state of the account is: 7 + 6 + 5 8 + 4 + 3 + 2 + 1 = 28 the suitability value is

determined accordingly. In short, the lowest number indicates that many queens are able to eat each other, while the number equal to max_control indicates the desired solution.

The best among these control values determined for each value in the population is selected. This situation is an example of natural selection encountered in biology, while the good survive, individuals unfit for nature die. Individuals who are the best after selection are put into the crossover process.

The best of these control values determined for each value in the population is chosen. This is an example of natural selection in biology where those who are not fit for nature die while living well. After the election, the best people are taken into the transition process.

Although the selection and transition processes are made, after a while the population begins to repeat itself, as the population ensures the survival of the best. To prevent this, the mutation process is applied. The mutation is determined as assigning another value randomly to a specified location on the determining chromosome. When a certain part of the population mutates, a decrease in repeats will be observed.

The program prepared limited the individuals in the population to 100 and mutation depends on the N value. The larger the mutation rate N value, which is a small value in small panel sizes, the higher the value is given to give faster results.

The whole population, individuals in the population and control values are saved in population.txt so that generations can be observed.

2.4. 8 Queens Problem Interface

The interface is very important in terms of better understanding of the program and transferring it to all ages. The problem is reflected quite well by providing examples and providing solutions to the player. In the interface built on the solution part, there is a main page, 2 screens connected to the buttons and 2 warning screens. The entire interface was written in Python and the PyQt5 library, which has a very large content, was imported.

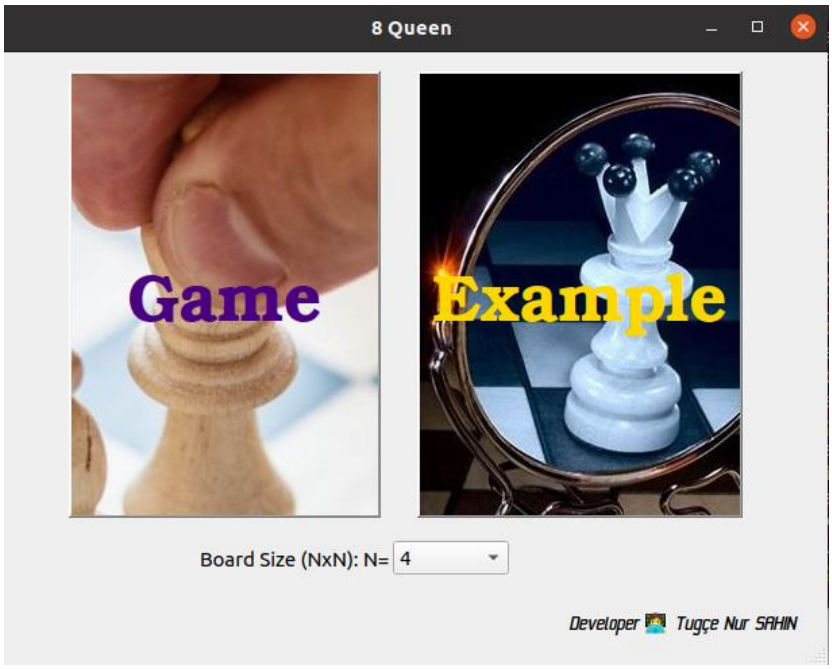

Figure 2.4.1 Main page view

There is a comboBox and two buttons on the main page. It is determined by the N comboBox, which shows the size of the chessboard and the number of queens. The ComboBox contains values 4,5,6,7,8. It is not included in the problem because it is 0 in 1, 2, 3 N values. After determining the N value, the user can move on to the example or game part.

2.4.1. Example Part

Example screen, opened by pressing the sample button after giving the N number, gives results suitable for 8 queens problem. N numbers can be given for all numbers from 4 to 8. It offers a new solution by containing different populations each time with the genetic algorithm study from the main section. The number of solutions is different for each N number. Before applying the genetic algorithm, although there are N! number of solutions, this number drops significantly after the algorithm is applied.

N Number	4	5	6	7	8
Number of Solutions	2	10	4	40	92

Table 2.4.1.1 Table of solutions to number N

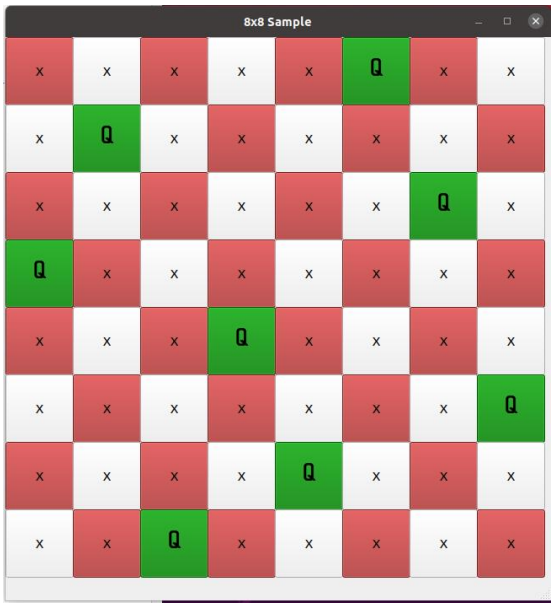

Figure 2.4.1.1 Example window for 8x8

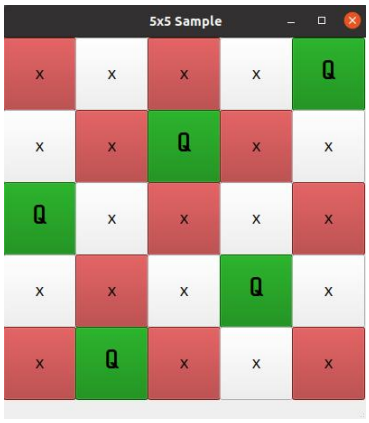

Figure 2.4.1.2 Example window for 5x5

In the examples given above, the numbers 8 and 5 are given to the N number. The Queen's locations have been made clearer with the green background. The symbol 'Q', which is the initial of the word Queens, and its green color distinguish it from other boxes.

2.4.2. Game Part

Designed to present the problem to the user in a fun situation, the game button tests the accuracy of 8 Queens problems. Working with the control function described in the main section, this section works for each N number. The control function has a system that deals with all the right, left and diagonals boxes, checking if another queen has come to those parts. What is required of the player is to place the queen on the right, left and diagonals of the queen without encountering the queen or queens. Since the boxes are designed as buttons, gene production is provided at the back every time the box is pressed. When the process is over, the 'Play' button is pressed and the created genes become chromosomes. The gene value of each button pressed in the terminal section is printed on the terminal screen. If an incorrect button is thought to be pressed, the moves used are undone by clicking the 'Refresh' button and the game starts again.

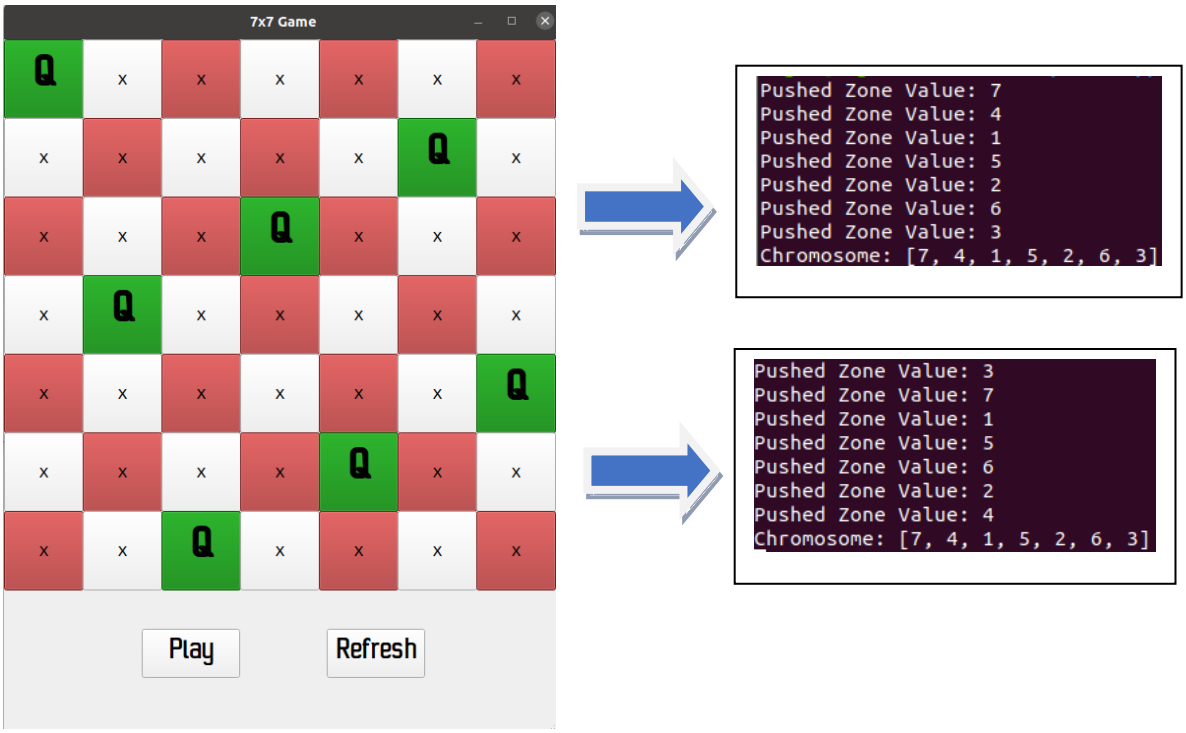

Figure 2.4.2.1 When the value of N is 7, displaying the game part and the terminal view when the boxes are pressed

In the example given above, the buttons are pressed both sequentially and randomly. In both cases, the gene sequence within the chromosome is unchanged. This is because the chessboard is thought of as a sequence, then a one-dimensional sequence is created to create the chromosome.

When each box, that is, the button is pressed, the color of the box turns green to be more distinctive than the other parts. At the same time, the letter 'Q' is prominently displayed. This was made for the convenience of the player.

There are 2 different situations that can be encountered when clicking the 'Play' button in the game. First, max_control and control values will be equal if the rules are followed. When this equality is achieved, "You win!" text will appear. On the contrary, if the rules are not provided, the max_control value will not equal the control value. In this case, "Game over!" text appears on the screen. These emoticons were supported with emojis to make the game fun.

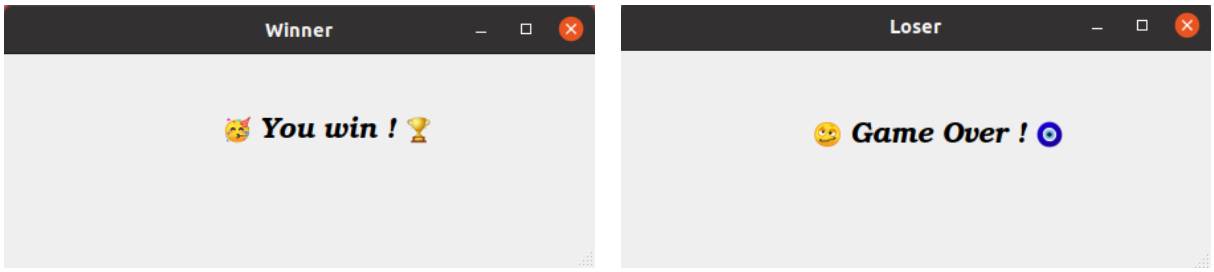

Figure 2.4.2.2 The windows image when the game is won and lost

The game will restart after the two windows shown above. If the game is requested on a different sized chessboard, it is necessary to return to the main screen. The game can be started again by selecting the N number from the main screen.

The interfaces used are written with the 'First Code' approach. However, before this process, QtDesiger was used to learn interface design and understand its logic. This program allows us to create new windows using the PyQt5 library. After learning the interface development with this program, the interface coding suitable for the solution of the problem was written manually, independent of QtDesigner.

3. CONCLUSION

In HAVELSAN, the internship has been completed for the intern in a structure that develops in vision and professional field. In addition to being a very enlightening internship, the internship also contributed in terms of work experience. This process gave strong ideas about the progress of business life. Although the internship was completed over the internet, it provided the opportunity to learn about business ethics and disciplines and to see how the teams work with each other. Thanks to the online trainings, many vocational trainings were received and internship projects were carried out with the gains from these trainings. These experiences will shed light on business life.

Artificial intelligence is shown as the technology of the future. For this reason, the fact that the project is about artificial intelligence has increased the interest even more. The related project proceeds through the decision-making mechanism, which is one of the basic components of artificial intelligence. The genetic algorithm, which is the combination of biology and mathematics, functions as decision-making within the project. This algorithm aims to find the closest solution to the solution, not the solution. For such reasons, working on the project and personal development is both enjoyable and important.

Since the project has been determined to be based on the Linux operating system since the construction phase, the Linux operating system hierarchical structure and commands were examined and reported in detail in the first two weeks of the internship. These reports were shared with the mentor weekly and their approval was obtained. After these stages, the genetic algorithm and its possible problems were investigated. With the identification of the problem, the solution process and interface design started. The interface is designed to appeal to all users and thanks to the project, both a product to be offered to the end user and the logic of the genetic algorithm are well understood. All of these are explained in detail in the report. Although internship is perceived as transforming theoretical knowledge into practice, this internship has been used as theoretical and practical knowledge.

I would like to thank my teacher Hacı Bayram KARAKURT for their contribution to the internship project during the internship and the HAVELSAN Family for this opportunity.

BIBLIOGRAPHY

Özçakar, N., 1998, Genetic Algorithm, I.U. İşletme Fakültesi Journal, 27, 69-82.

Pulat, M., DEVECİ KOCAKOÇ, İ.,2016, Case-Based Analysis of Crossing Operators in Solving the Traveling Salesman Problem Using Genetic Algorithms, Operations Research Industrial Engineering 36th National Congress

[1]https://www.havelsan.com.tr/

[2]https://tr.wikipedia.org/wiki/Sekiz_vezir_bulmacas%C4%B1

[3]https://tr.wikipedia.org/wiki/Genetik_algoritma

http://www.belgeler.org/lis/archive-tlkg-lis-5.11.html

https://build-system.fman.io/pyqt5-tutorial

https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm)

https://www.mehmetemineker.com/8-vezir-probleminin-genetik-algoritma-ile-cozumu/

https://medium.com/@AlameddinC/genetik-algoritma-1-fbbc2c4d001d

http://kergun.baun.edu.tr/20172018Guz/YZ_Sunumlar/Genetik_Algoritmalar_Busra_Guracar.pdf

https://www.researchgate.net/publication/310463101_Gezgin_Satici_Probleminin_Genetik_Algoritmalar_Kullanarak_Cozumunde_Caprazlama_Operatorlerinin_Ornek_Olaylar_Bazli_Incelenmesi

https://pchocasi.com.tr/8-vezir-problemi-nedir-8-queens-problem-22094/

APPENDICES

Appendix 1. <u>List of training provided by the institution</u>

- Basic Occupational Health and Safety Training
- Intern Awareness
    - Data security
    - Havelsan Presentation
    - Quality Awareness
    - Information security
    - Physical Security
    - HR Orientation
    - Subcontractor Environmental Management System
    - Information Security Basic Training
    - Web Security
    - E-Mail Security Training
    - Effective Business Correspondence and Official Correspondence
    - Professional Email Formula
    - Common E-mail Mistakes
    - Personal Data Protection Law in Details
    - Are You Ready for the KVKK Compliance Program?
    - Malware
    - Social Networks Security
    - Password Security
    - Mobile Security
- Cyber Threats and Alerts - 1
- Cyber Threats and Alerts - 2
- Data Science and Analysis
- Data Science and Analysis 2
- Block Chain Technology
- Autonomous Systems
- 5G and Applications
- Software Testing Basics

Appendix 2. Main page and N Number

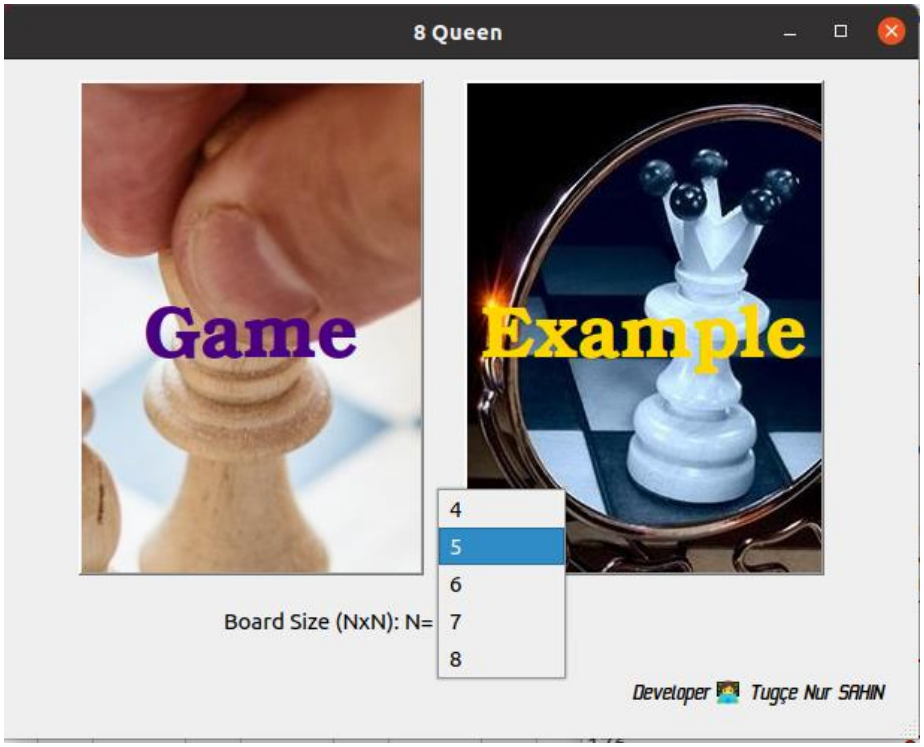

Figure 2.1 Main page

Appendix 3. Windows of the example button depending on the number of N

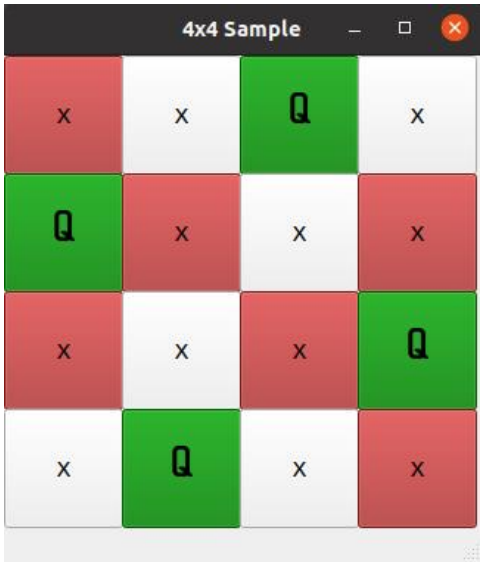

Figure 3.1 4x4 example window

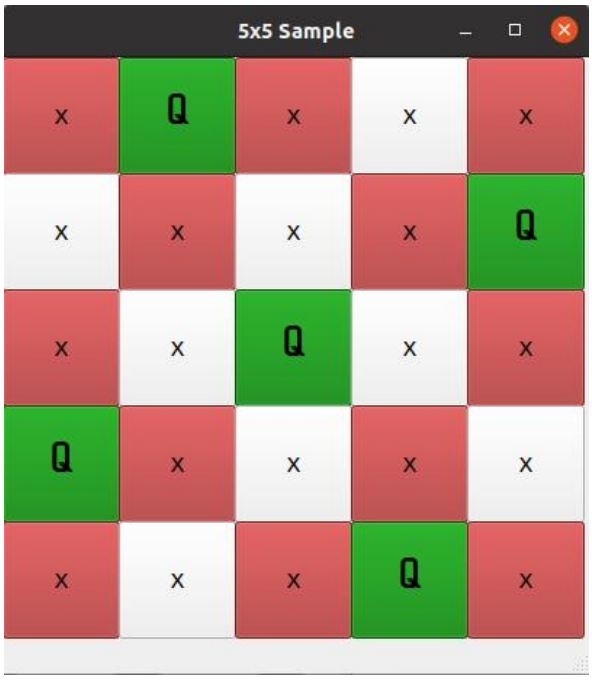

Figure 3.2 5x5 exaple window

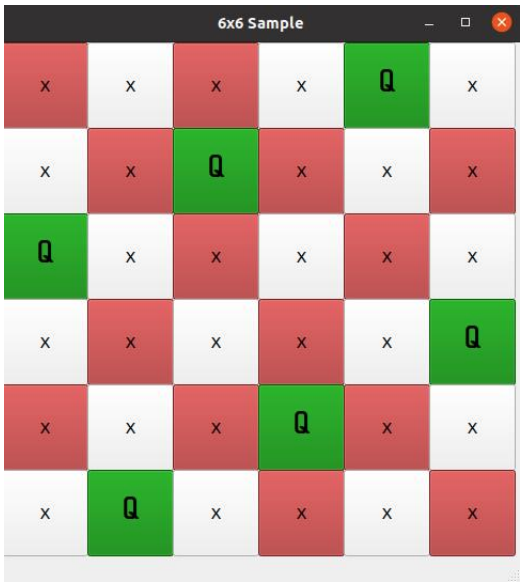

Figure 3.3 6x6 example window

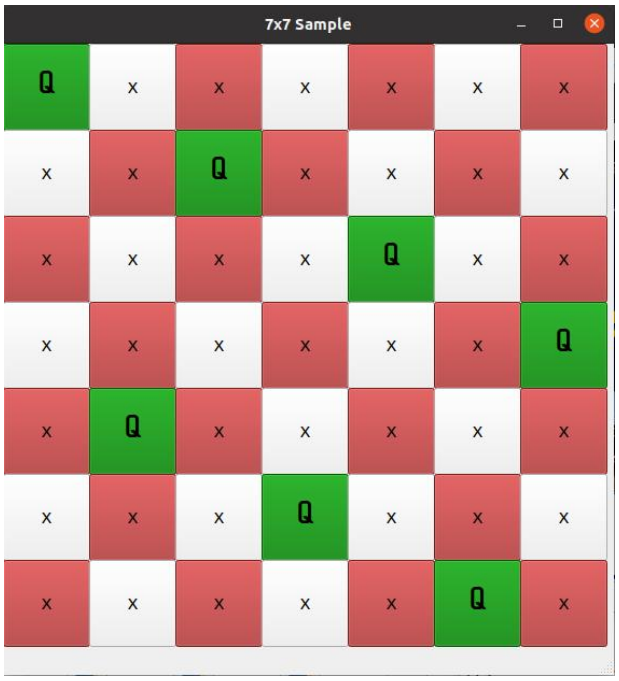

Figure 3.4 7x7 example window

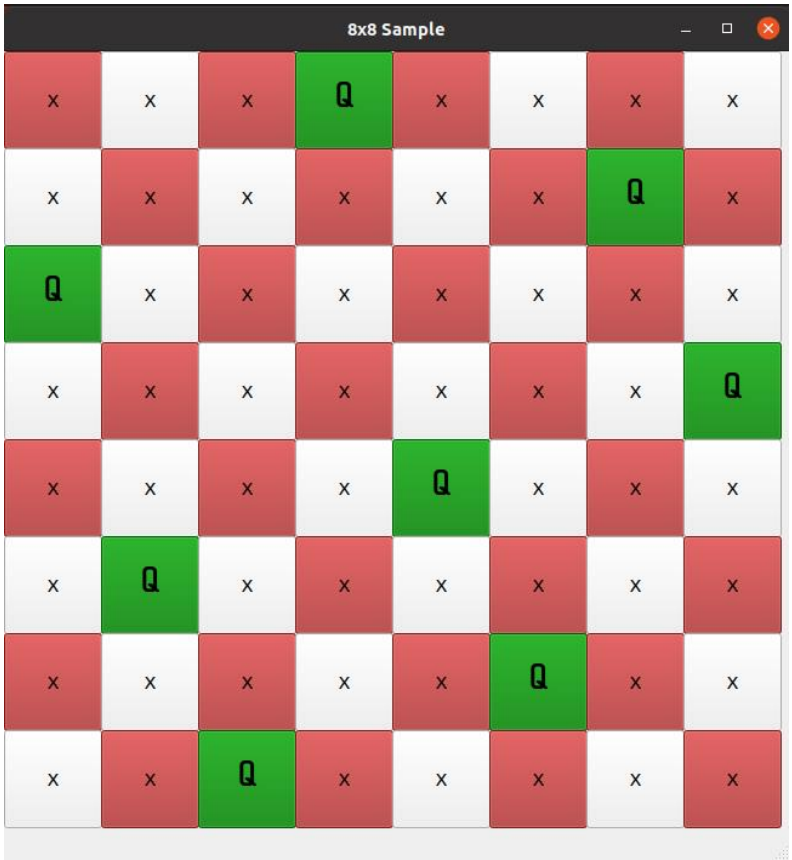

Figure 3.5 8x8 example window

Appendix 4. <u>Windows of the game button depending on the number of N</u>

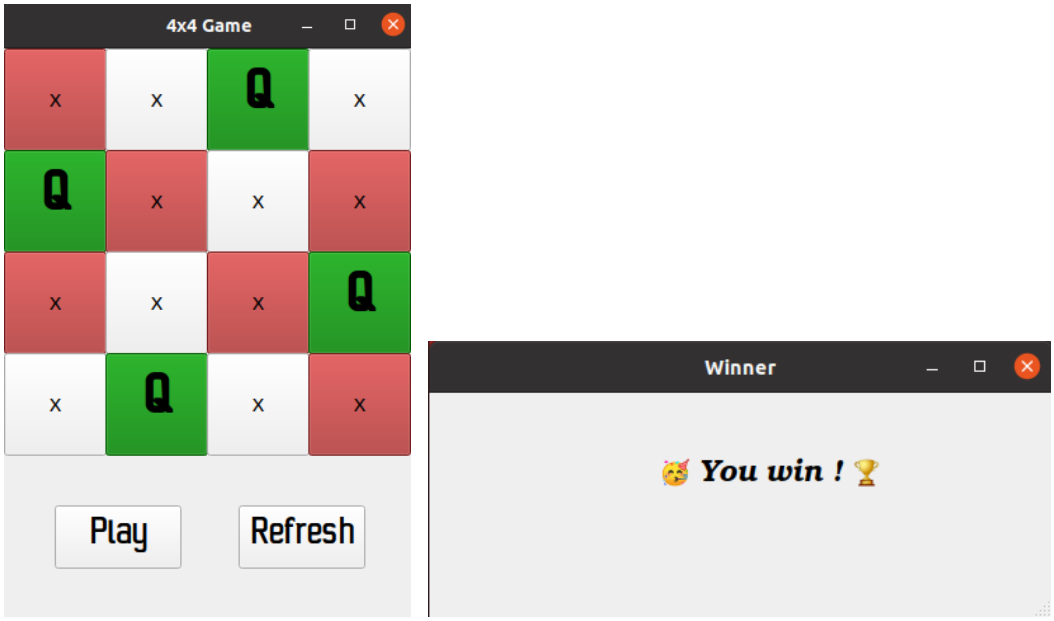

Figure 4.1 When the 4x4 game is won

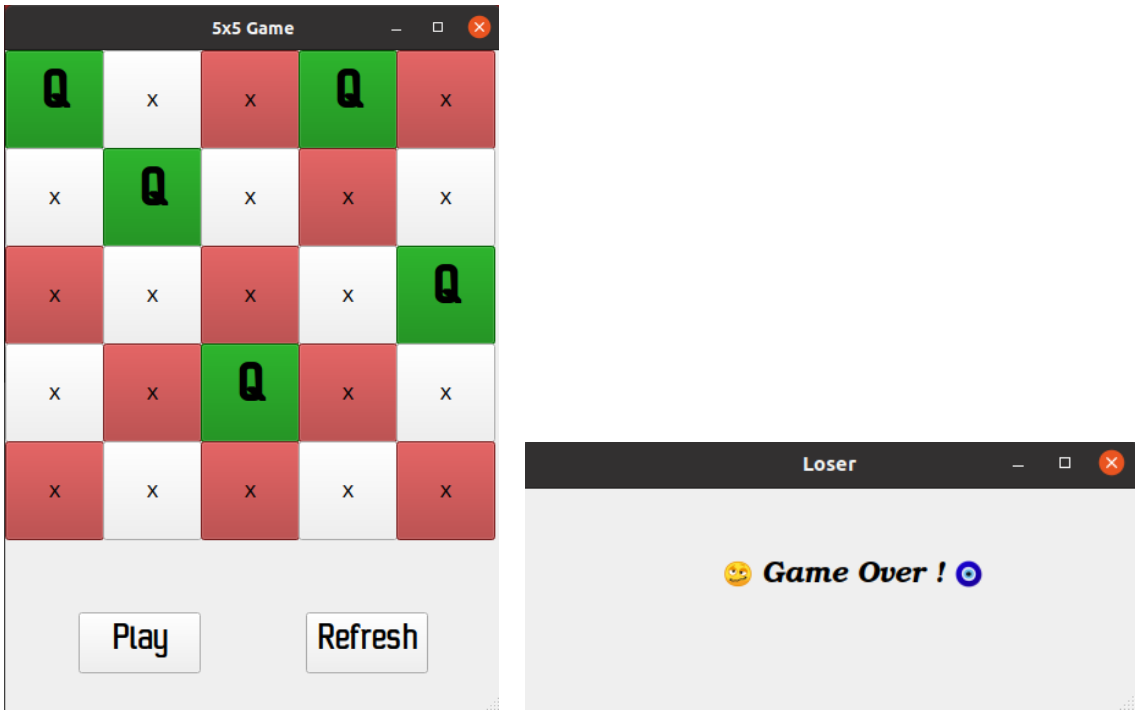

Figure 4.2 When the 5x5 game is lost

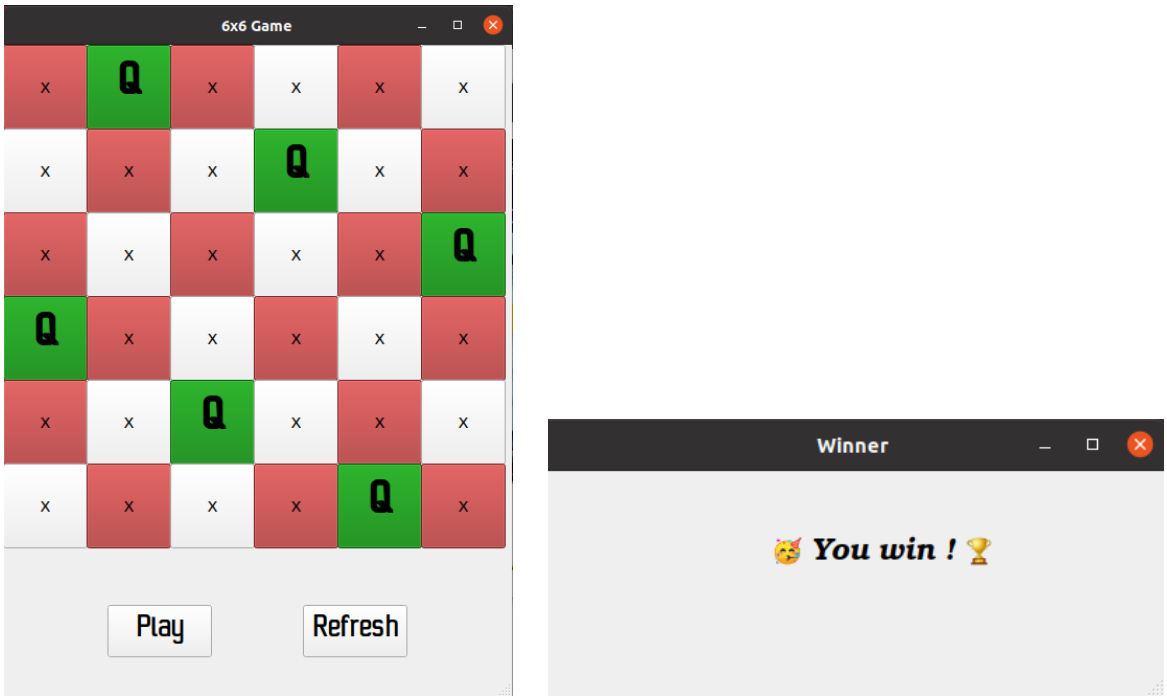

Figure 4.3 When the 6x6 game is won

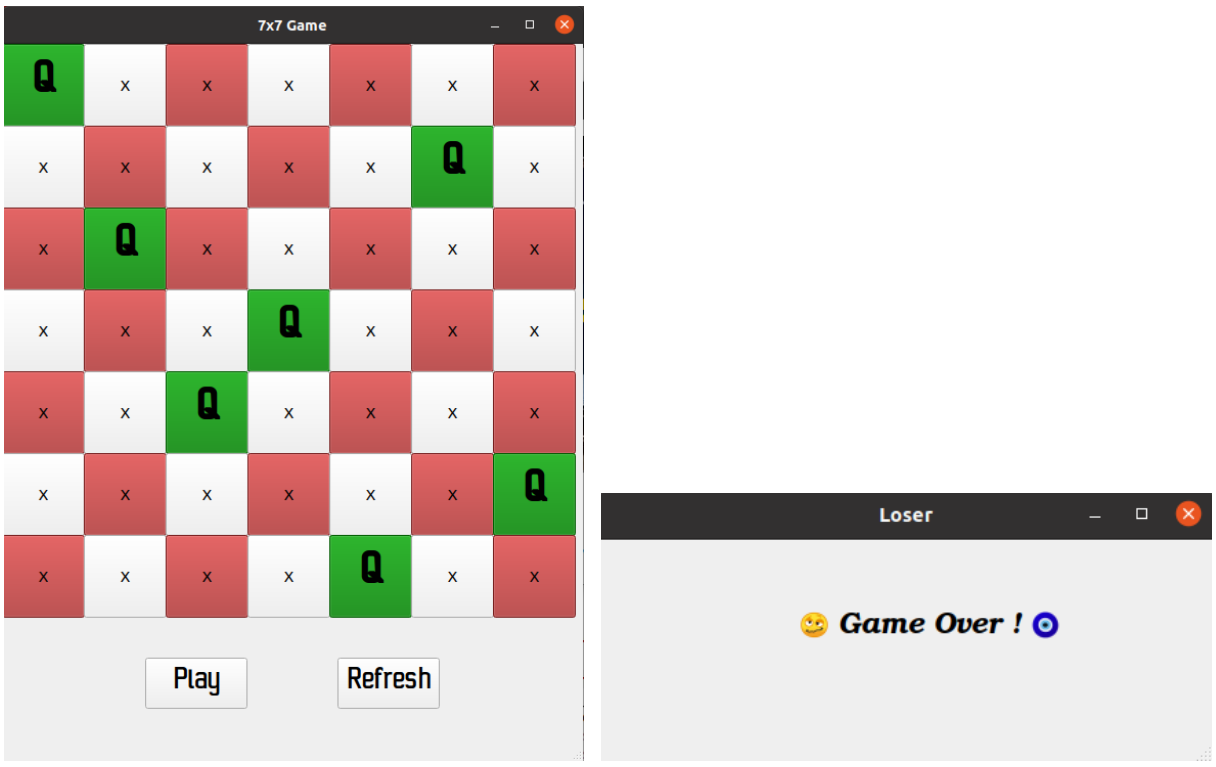

Figure 4.4 When the 5x5 game is lost

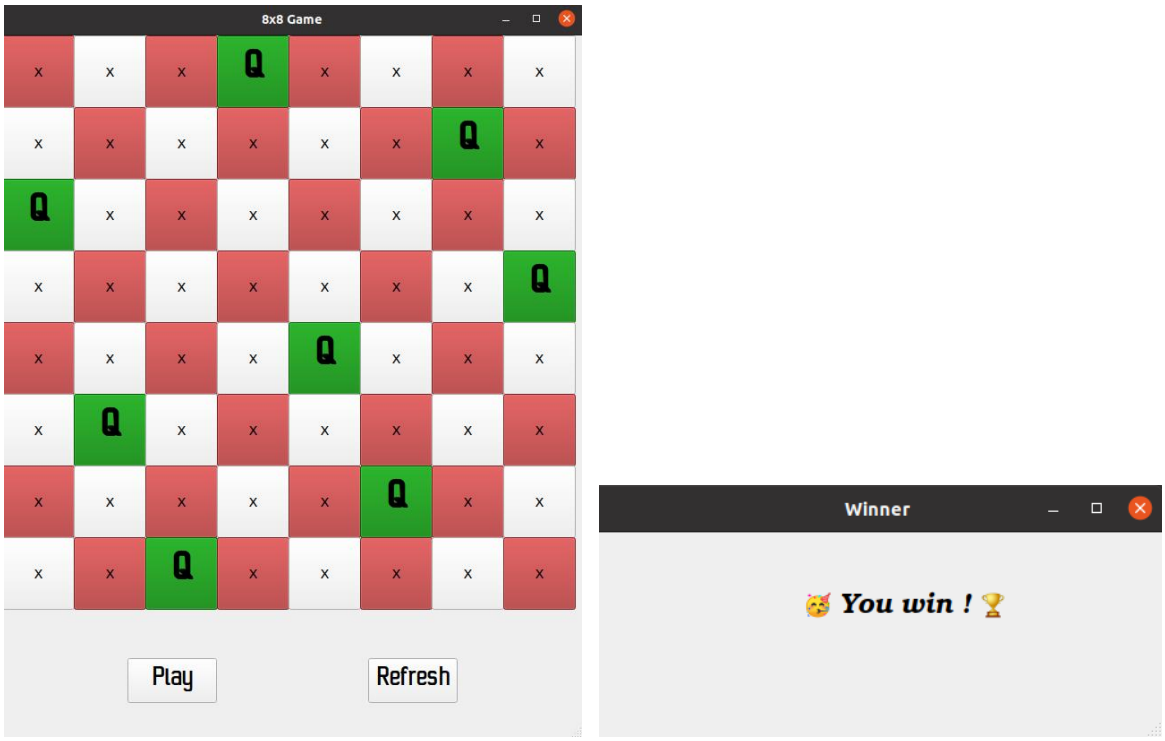

Figure 4.5 When the 8x8 game is won

Appendix 4. Controls function

```
def controls(kromozom):
    yatay_cakisma= sum([kromozom.count(queen)-1 for queen in kromozom])/2
    capraz_cakisma = 0
    uzunluk = len(kromozom)
    sol_x = [0] * 2*uzunluk
    sag_x = [0] * 2*uzunluk
    for i in range(uzunluk):
        sol_x[i + kromozom[i] - 1] += 1
        sag_x[len(kromozom) - i + kromozom[i] - 2] += 1
    capraz_cakisma = 0
    for i in range(2*uzunluk-1):
        flag = 0
        if sol_x[i] > 1: flag += sol_x[i]-1
        if sag_x[i] > 1: flag += sag_x[i]-1
        capraz_cakisma += flag / (uzunluk-abs(i-uzunluk+1))
    return int(max_c - (yatay_cakisma + capraz_cakisma))
```

The above mentioned code section is chosen as an example from the project. For all source codes of the project:

https://github.com/TugceNurSAHIN/Genetic_Algorithm_8Queen_Game