TIC TAC TOE GAME USING PYTHON

A Mini Project-2 Report submitted
In partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

By

1. **19B01A0508 – A. Bhavya Sri**
2. **19B01A0511 – A. Sravanthi**
3. **19B01A0533 – Ch. Vijaya Deepika**
4. **19B01A0545 – G. Sravya Sree**
5. **19B01A0554 – J. Sri Pranathi**

Under the esteemed guidance of
Mr. P. Sunil
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)

(Approved by AICTE, accredited by NBA & NAAC, Affiliated to JNTU Kakinada)

BHIMAVARAM – 534202

2021– 2022

SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)

(Approved by AICTE, accredited by NBA & NAAC, Affiliated to JNTU Kakinada)

BHIMAVARAM – 534202

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

*This is to certify that the Mini Project-1 entitled “**TIC TAC TOE GAME USING PYTHON**”, is being submitted by **A. Bhavya Sri, A. Sravanthi, Ch. Vijaya Deepika, G. Sravya Sree, J. Sri Pranathi,** bearing the **Regd. No. 19B01A0508, 19B01A0511, 19B01A0533, 19B01A0545, 19B01A0554** in partial fulfilment of the requirements for the award of the degree of **“Bachelor of Technology** in **Computer Science & Engineering”** is a record of bonafide work carried out by her under my guidance and supervision during the academic year **2021–2022** and it has been found worthy of acceptance according to the requirements of the university.*

Internal Guide **Head of the Department**

ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance has been a source of inspiration throughout the course of this project. I take this opportunity to express our gratitude to all those who have helped us in this project.

We wish to place our deep sense of gratitude to **Sri. K. V. Vishnu Raju, Chairman of SVES**, for his constant support on every progressive work of mine.

We wish to express our sincere thanks to **Dr. G. Srinivasa Rao, Principal of SVECW** for being a source of an inspirational and constant encouragement.

We wish to express our sincere thanks to **Dr. P. Srinivasa Raju, Vice-Principal of SVECW** for being a source of an inspirational and constant encouragement.

We wish to place our deep sense of gratitude to **Dr.P.Kiran Sree, Head of the Department of Computer Science & Engineering** for his valuable pieces of advice in completing this project successfully.

Our deep sense of gratitude and sincere thanks to the PRC members: **Mrs.P.R.Sudha Rani(Associate Professor), Dr.M.Prasad(Professor),** and Project coordinator: **Mr.G.V.S.S. Prasad Raju(Assistant professor)** for their encourage and support in making the project success.

We wish to place our deep sense of gratitude to **Mr.P.Sunil, Assistant Professor** for his valuable pieces of advice in completing this project successfully.

Project Associates

1. 19B01A0508 – A. BHAVYA SRI
2. 19B01A0511 – A. SRAVANTHI
3. 19B01A0533 – CH. VIJAYA DEEPIKA
4. 19B01A0545 –G. SRAVYA SREE
5. 19B01A0554 –J. SRI PRANATHI

Abstract

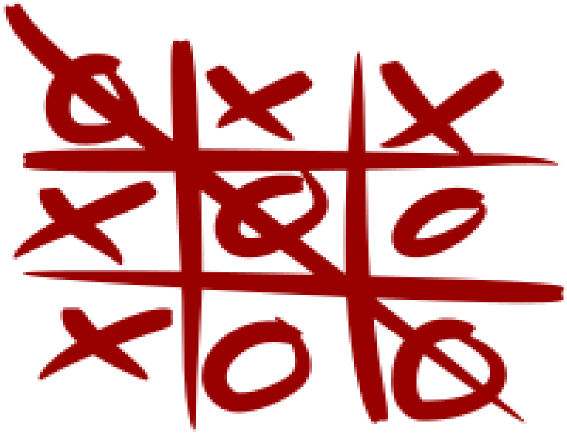
Tic Tac Toe Game, implemented with Python program, is a two-player game where both players must place a X or O marking the spaces in a 3×3 grid board. To win the game players have to get 3 of her marks in a row (up, down, across, or diagonally). The project purpose is to have an entertainment and past time after some hard time from work. The objective of this tic-tac-toe game is to build a tic-tac-toe game so user can play it without wasting paper and improve concentration.

The project is played in a NxN grid board where the players take turn after placing a marker on the board. Traditional game is modified with interesting features, which we proposed in our project. The game can be played by multiple players depend on what user choose in the options.
We included background music while playing the game. The user can also select the game mode which is player vs player or player vs computer.

Contents:

1.INTRODUCTION

Tic-tac-toe is a game that is traditionally played by using pen and paper, and it can be played on a computer also. Tic-tac-toe is a simple, two-player game, if played optimally by both players, will result in a tie. The game is also called noughts and crosses or Xs and Os. In this project, we are going to develop an interactive game where two players will be able to play against each other and also against with computer in suitable GUI



The X player usually goes first. The player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the game. After completion of all valid moves the game will be draw. In the game mode of computer vs player by default player1 is human and player2 is computer. Its purpose is to have an entertainment and past time after some hard time for work.

Objectives:

- The user can play without wasting paper and pen.
- It is implemented using GUI.

2. SYSTEM ANALYSIS

2.1 Existing System

Tic-tac-toe is played on a three-by-three grid by two players, who alternately place the marks X and O in one of the nine spaces in the grid.

DRAWBACKS:

- It is quite simple where more options are not available
- Existing game is boring with no music
- GUI features are not implemented and most of the implementations are console based.

2.2 Proposed System

- User-friendly Interface
- Implementation of Tic Tac Toe with Python GUI.
- More options like selection of game mode, player names and moves etc.
- Background music while playing the game.
- Alert notification when the board size is invalid.
- Winning and draw sound is produced along with the message box.

2.3 Feasibility Study

Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies.
A feasibility study should be relatively cheap and done at the earliest possible time
Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration.

Facts considered in the feasibility analysis were:

1.Technical Feasibility
2. Economic Feasibility
3.Behavioral Feasibility

1.Technical Feasibility:

- It includes whether the technology is available in the market for development and its availability.
- The assessment of technical feasibility must be based on an outline design of system requirements in terms of input, output, files, programs and procedures.
- This can be qualified in terms of volumes of data, trends, frequency of updating, cycles of activity etc
- The proposed system "Tic Tac Toe Game” is planned to run on Python

2.Economic Feasibility:

- The technique of cost benefit analysis is often used as a basis for assessing economic feasibility.
- In this project “Tic Tac Toe Game” is developed on current equipment, existing software technology.

3. Behavioural Feasibility:

- This analysis involves how it will work when it is installed.
- It will accept broad audience from around the world.
- This proposed system “Tic Tac Toe Game” has much behavioural feasibility because users are provided with better options and good design

3. System Requirements Specification

3.1 Software Requirements:

- Front End: Python3
- Back End: Python 3 Interpreter
- Graphical User Interface (GUI) Toolkit
- Programming language: Python

3.2 Hardware Requirements:

- Processor – Intel core i5 or above
- Hard Disk – 128 GB
- Memory – 1 GB RAM
- Operating system – Windows

3.3 FUNCTIONAL REQUIREMENTS

Functional requirements are the requirements which deals with the operational requirements of the system and the requirements that are requested by the user.

Few of its functional requirements are as follows:

- The project includes knowledge on the PYTHON programming language, that would be used in the building of the entire game.
- Proper knowledge of modules like pygame, tkinter etc.

- Good Design of the game

NON-FUNCTIONAL REQUIREMENTS:

A non-functional requirement specifies the process that can be used to check the operations of a system. They are contrasted with the functional requirements that define specific behaviour or functions.

• The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

Some of them are:

• Robustness to cope with errors during execution and cope with erroneous input.

• Performance which tells us whether the input is valid or not and obtained outputs.

4.SYSTEM DESIGN

4.1 INTRODUCTION

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

System Analysis:

It is the process that decomposes a system into its component pieces for the purpose of defining how well those components interact to accomplish the set requirements. The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from problem domain to the solution domain.

The design activity is often divided into two separate phase-system design and detailed design. System design, which is sometimes also called top-level design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired result.

A design methodology is a systematic approach to creating a design by application of set of techniques and guidelines. Most methodologies focus on system design. The two basic principles used in any design methodology are problem partitioning and abstraction.

4.2. UML DIAGRAMS

UML (Unified Modelling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. It is a method for describing the system architecture in detail using the blueprint. We use UML diagrams to portray the behaviour and structure of a system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

- Use case diagram
- Class diagram
- Sequence diagram
- Activity diagram
- Collaboration diagram.

USECASE DIAGRAM:

A Use Case Diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

Use cases:

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

Actors:

An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

Associations:

Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modelled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line.

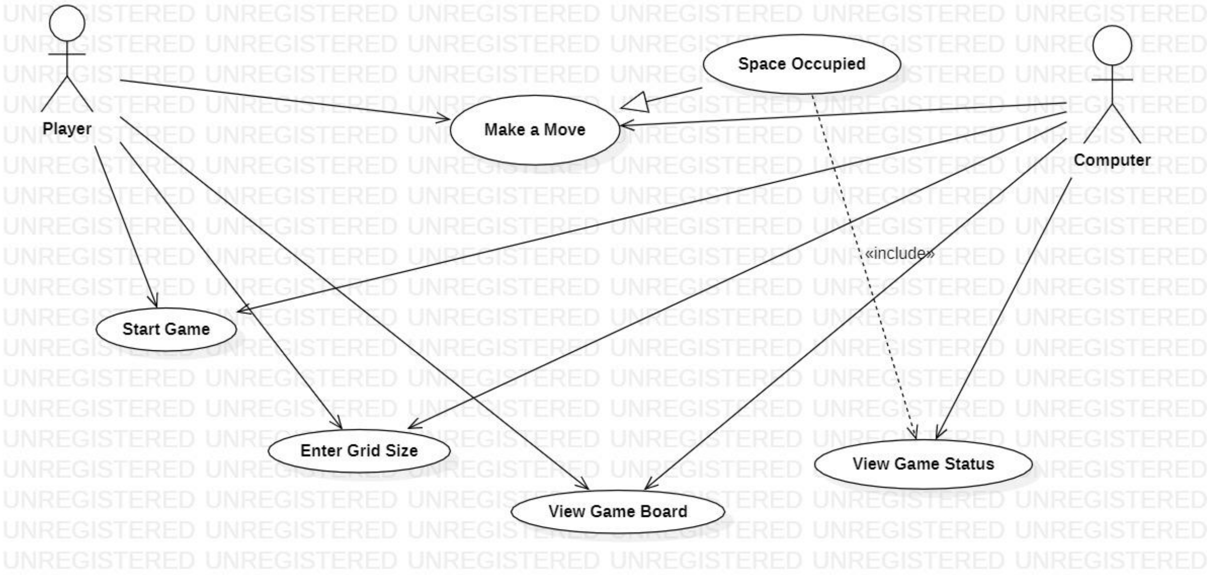


Fig.1: Use case Diagram for Computer vs Player

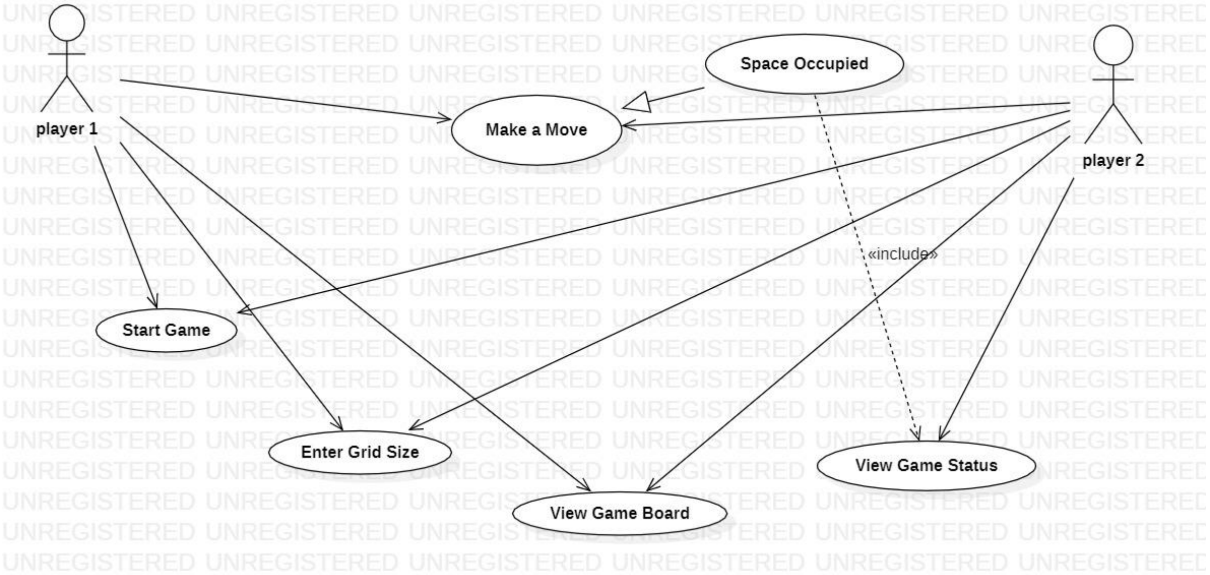


Fig2: Use Case Diagram for Player vs player

SEQUENCE DIAGRAM:

A sequence diagram is the most used interaction diagram.
Interaction diagram:
An interaction diagram is used to show the interactive behaviour of a system. Since visualizing the interactions in a system can be a cumbersome task, we use different types of interaction diagrams to capture various features and aspects of interaction in a system.

Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations:

1. Actors
2. Lifeline
3. Messages

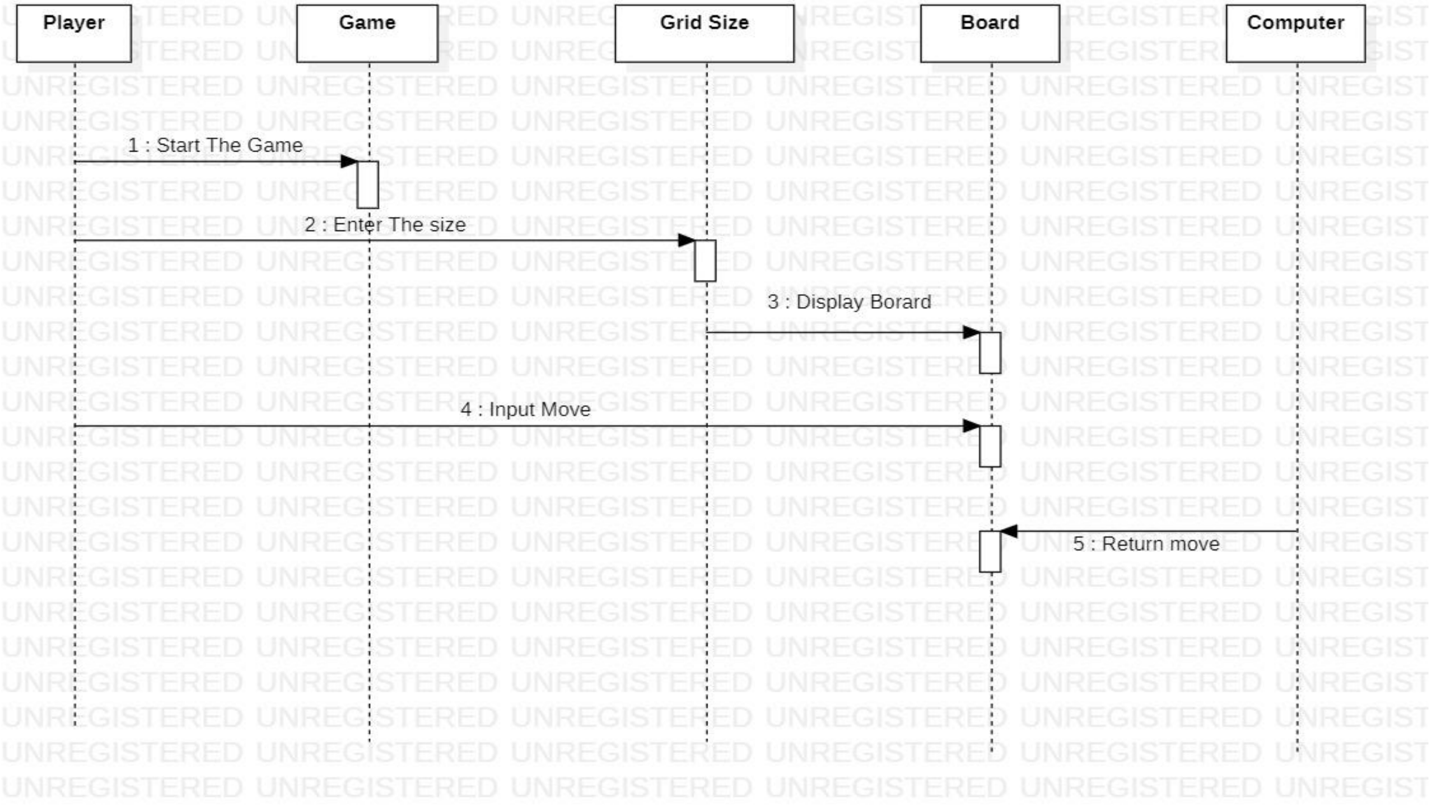


Fig3: Sequence Diagram for Computer vs Player

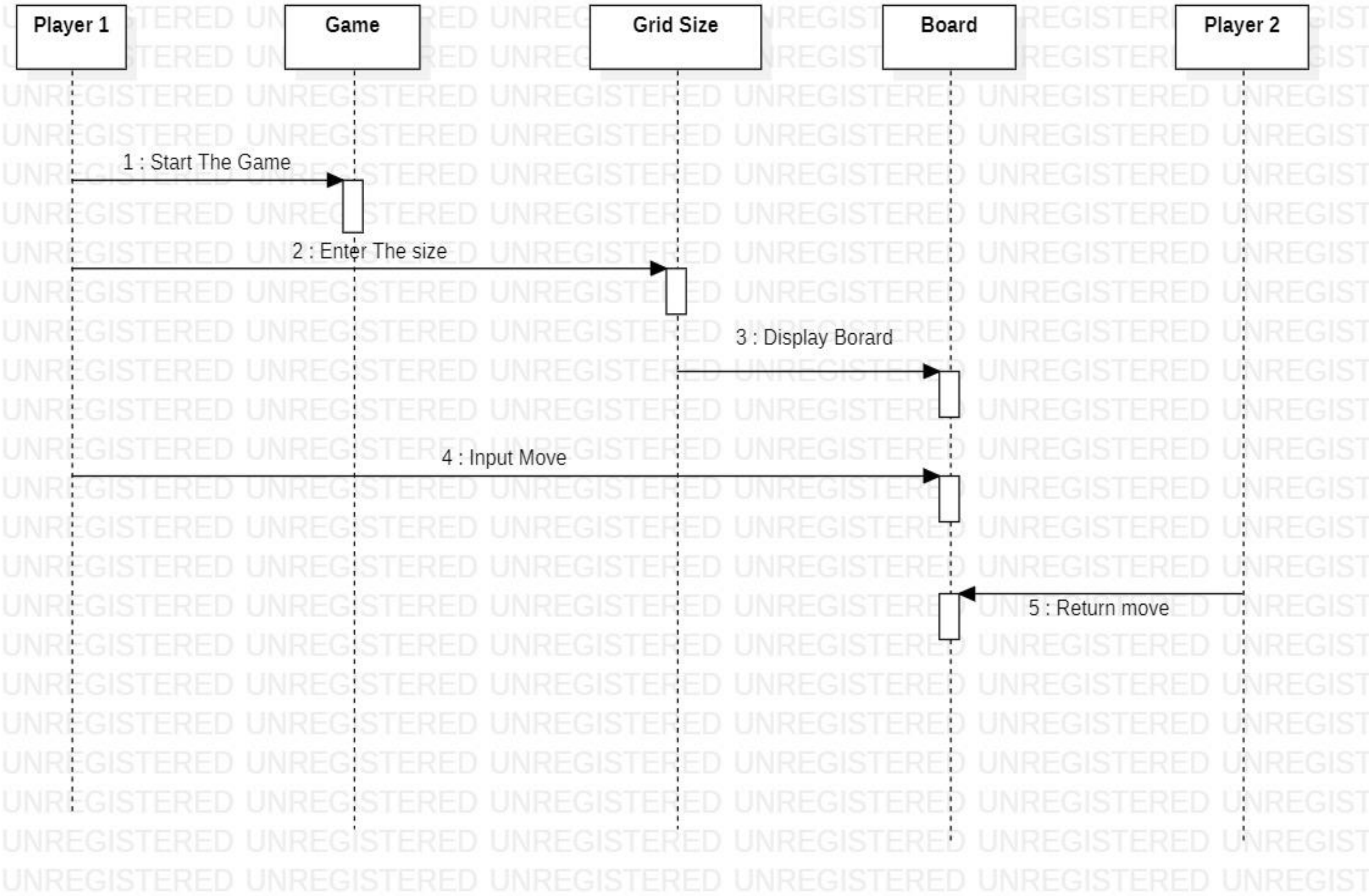


Fig4: Sequence Diagram for Player vs Player

ACTIVITY DIAGRAM:

The basic purposes of activity diagrams are like other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another, but activity diagram is used to show message flow from one activity to another.

The purpose of an activity diagram can be described as:

- Player start’s the game by entering the details of player and size of grid
- Choose the mode of game
- Result is declared after all the moves are completed or any one of the player wins

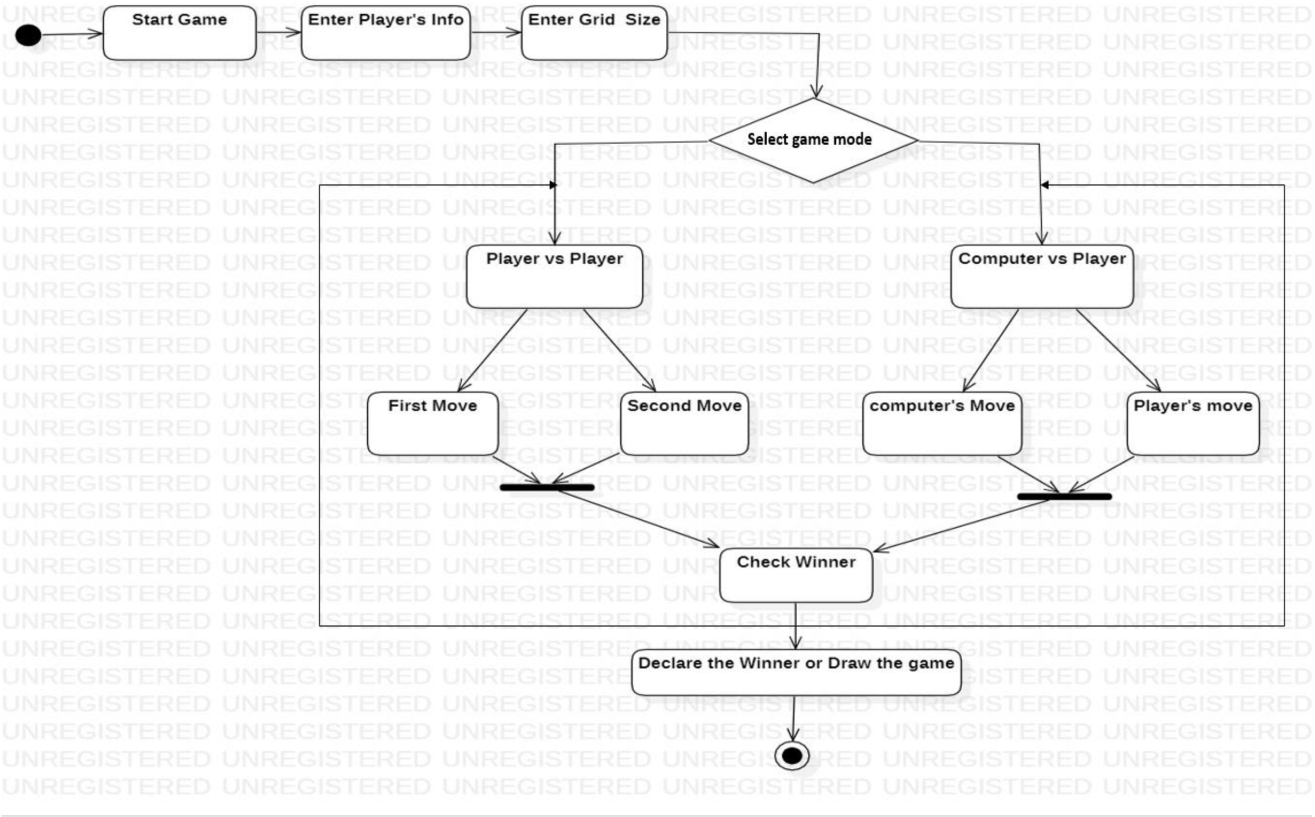


Fig5: Activity Diagram.

DEPLOYMENT DIAGRAM

Deployment Diagram is a type of diagram that specifies the physical hardware on which software system will execute. It maps software pieces of a system to the device that are going to execute it.

A deployment diagram consists of the following notations:

- A node
- A component
- An artifact
- An interface

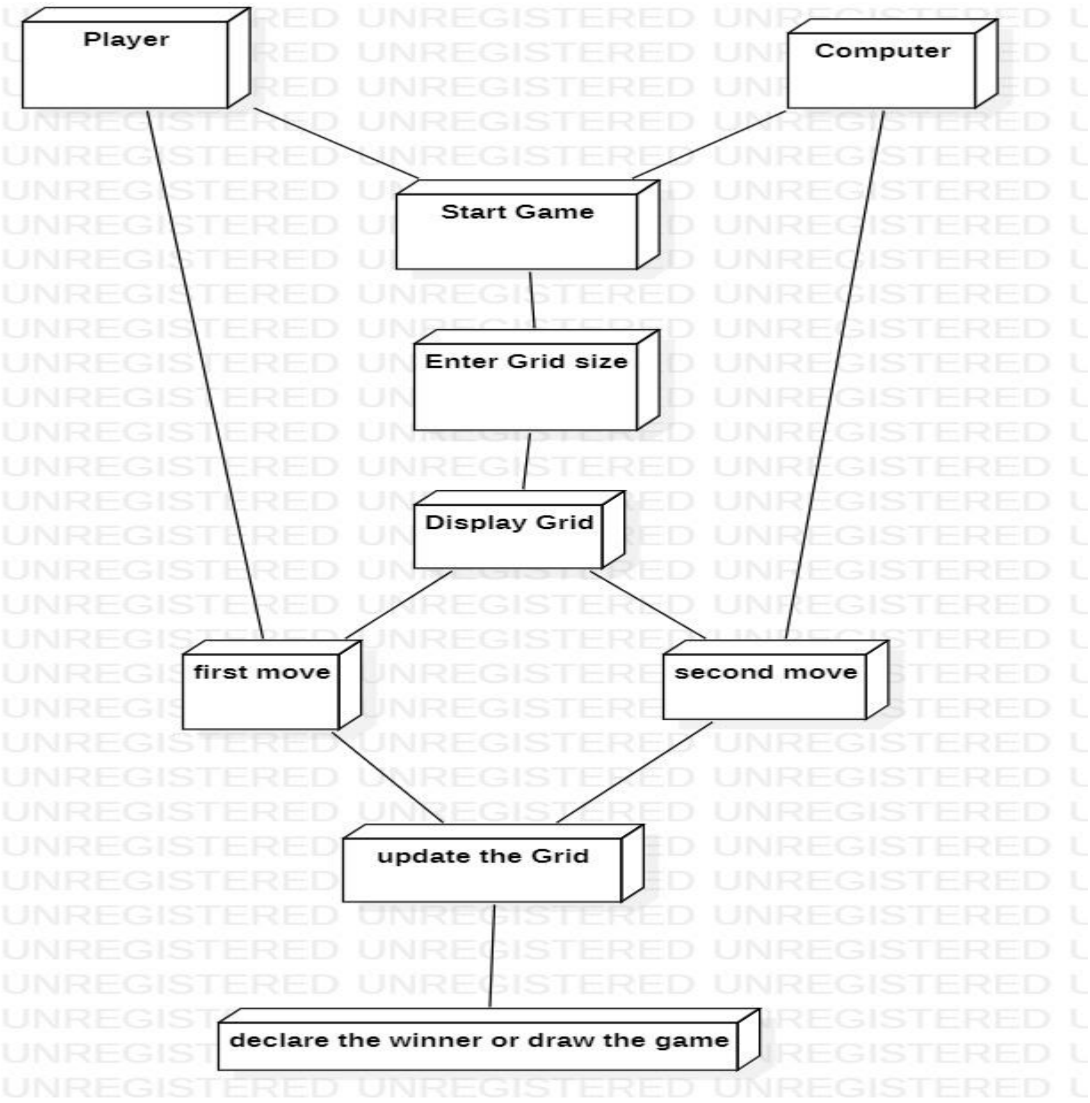


Fig6: Deployment Diagram for Computer vs Player

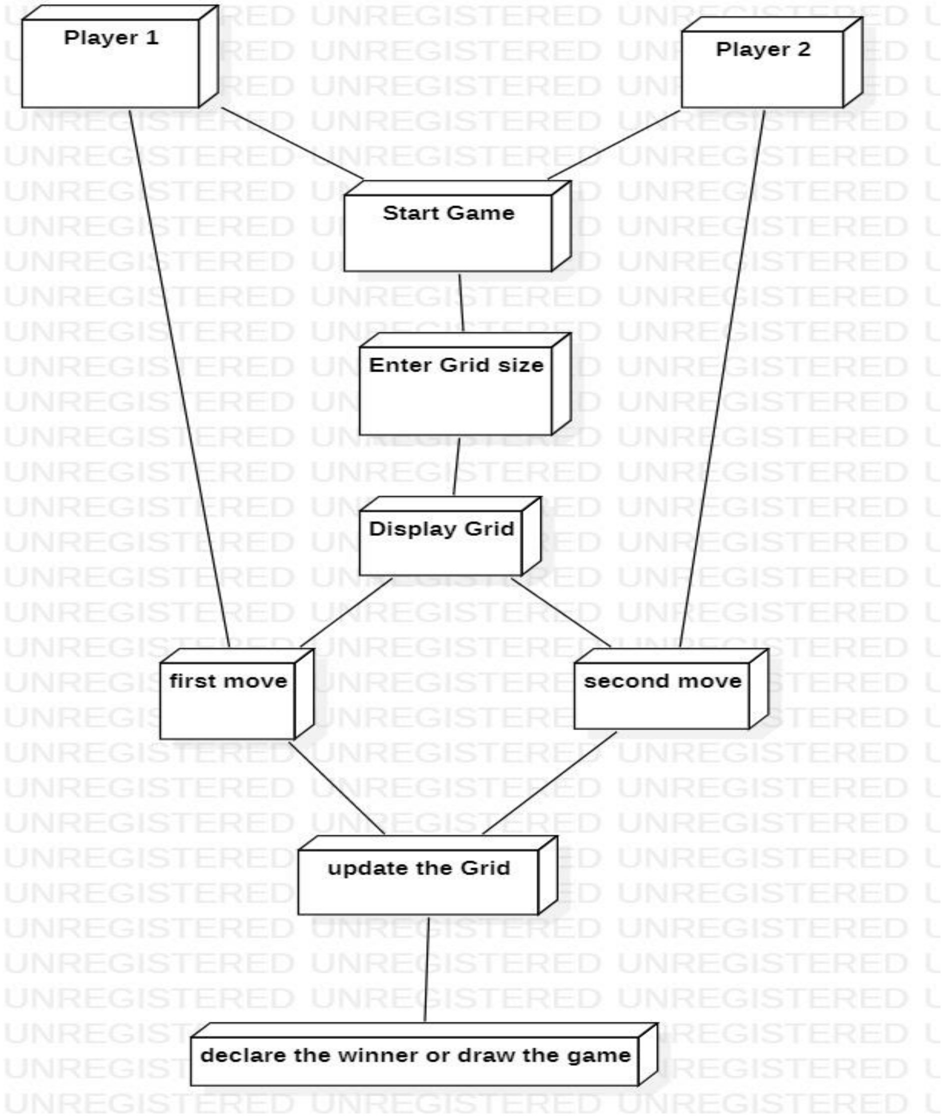


Fig7: Deployment Diagram for Player vs Player

5. System Implementation

5.1 Introduction:

The purpose of system implementation can be summarized as follow making the new system available to the prepared set of users (the deployment), and positioning ongoing support and maintenance of the system within the performing organization (the transaction). At a finer necessary to educate the consumer on the use of system, placing the newly developed system into production, confirming that business functions that interacts with the system and functioning properly.

Transitioning the system support responsibilities involve changing from a system development to the system and maintenance mode of operation, with ownership of the new system moving from the project team to the performing organization. A key difference between system implementation and all other phases of lifecycle is that all project activities up to this point have been performed in safe, protected and check your environments.

It is through the careful planning, execution and management of system implementation activities that the project team can minimize the likelihood of these occurrences and determine appropriate contingency plans in the event of the problem.

Our project Tic Tac Toe game is implemented with NxN grid with 2 modes i.e., player vs plyer and computer vs player. It is user friendly where user can choose any mode to play.

5.2 Project Modules:

These are the step to build Tic-Tac-Toe game using python:

1.Import modules

- To implement this game, Pygame and Tkinter modules need to be imported.
- From tkinter, message box module, simple dialog etc are imported.
- Pygame should be initialized here and quit at the end**.**

2.Initialize window

- Tk () is use to initialize window
- Title () used to set the title of the window

3.Function to check result

- In this function, the result will be checked by checking which player makes n value of their marks in a row, column or diagonal.

4.Function to check the winner

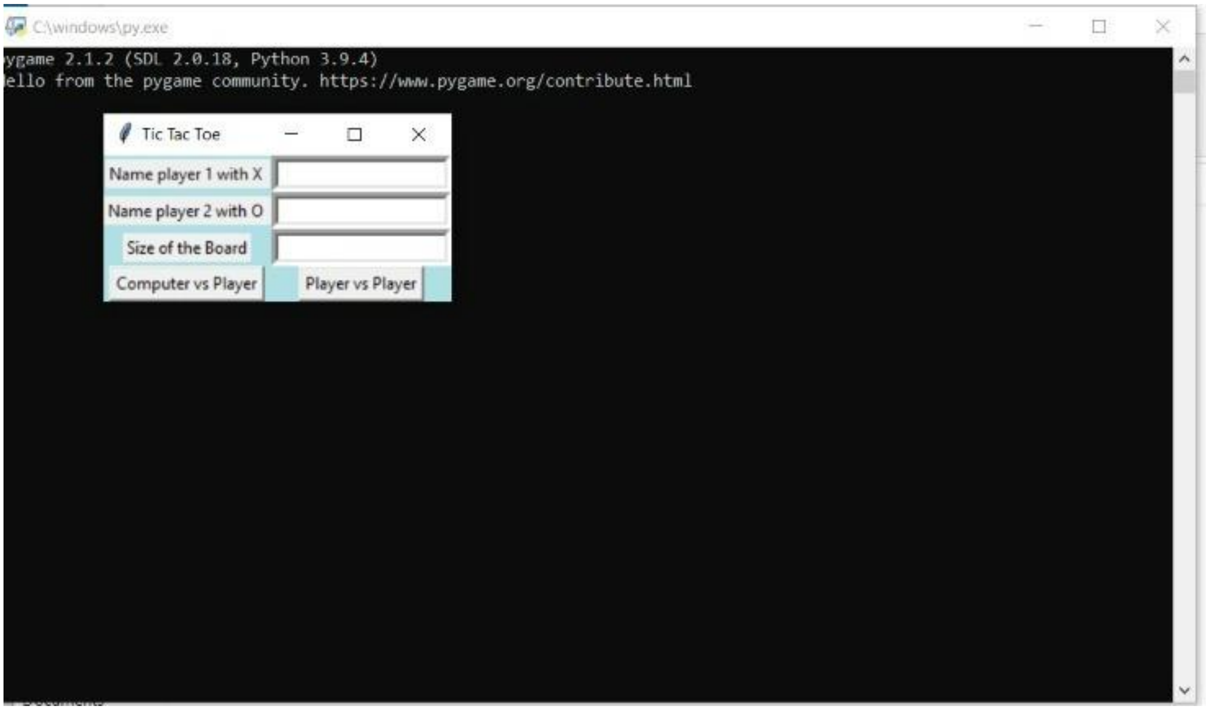
- Players have a total of n*n = n^2(3*3 means 9) clicks to play the game. Each time the player clicked, one chance will decrease by increasing the value of count by 1 if the value of count is greater than (n^2-1) then the result of game is tie.
- If the value of count is even then player1 will play else player2 will play.

5.Define labels and buttons

- Label () widget used to display text that users aren't able to modify.
- Button () widget display button
- Mainloop () method executes when we want to run our program.

5.3 Screens

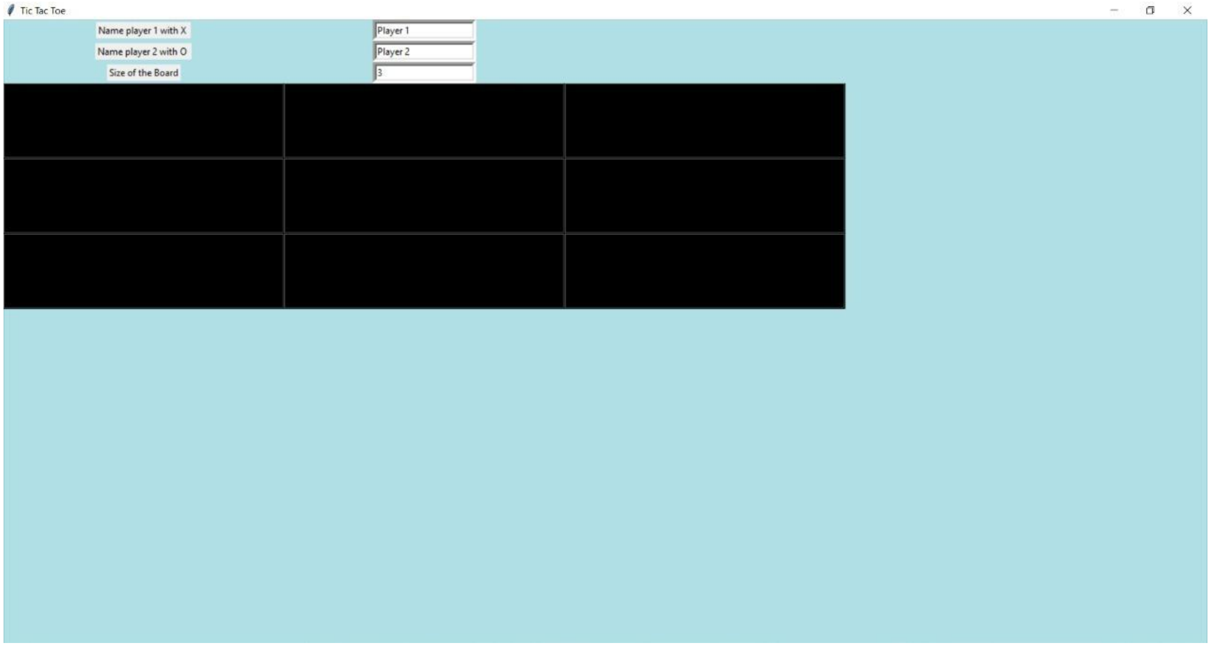
Input:



```
ygame 2.1.2 (SDL 2.0.18, Python 3.9.4)
ello from the pygame community. https://www.pygame.org/contribute.html
```

Grid Formation

3x3 Grid



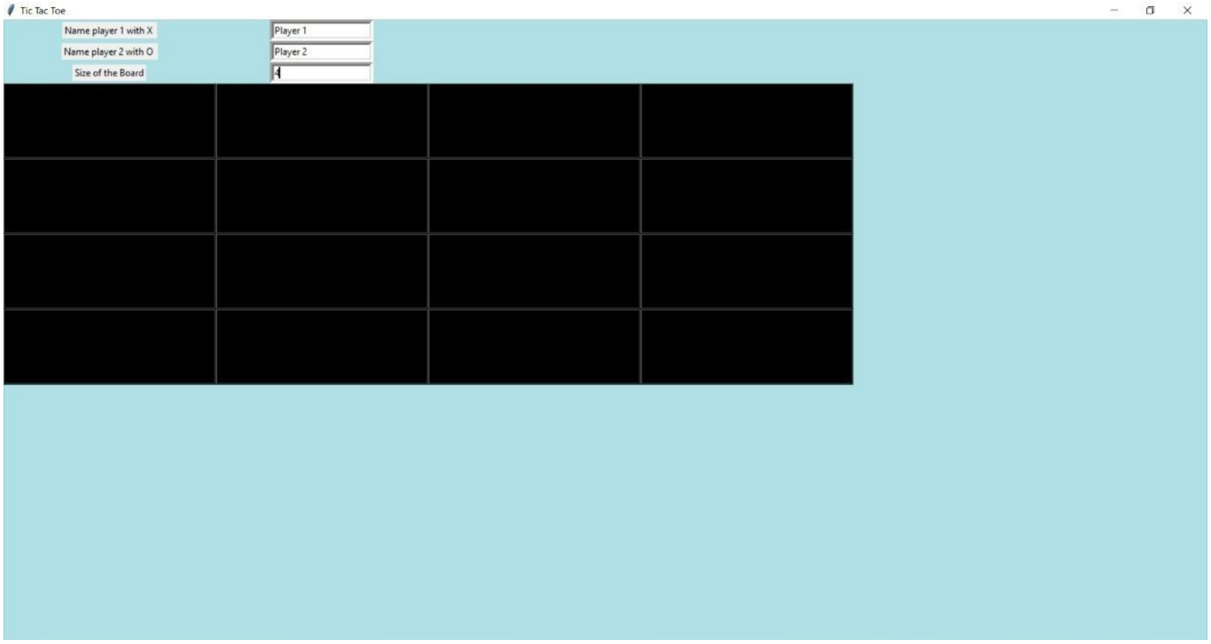
Tic Tac Toe

Name player 1 with X: Player 1

Name player 2 with O: Player 2

Size of the Board: 3

4x4 Grid



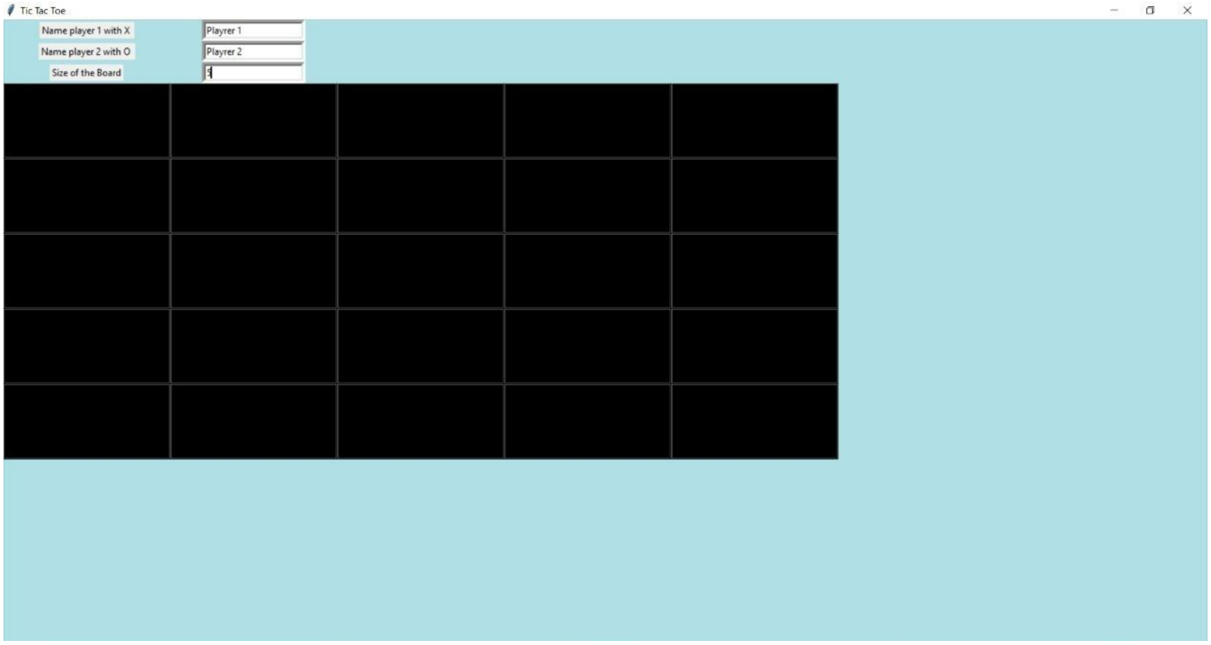
Tic Tac Toe

Name player 1 with X: Player 1

Name player 2 with O: Player 2

Size of the Board: 4

5x5 Grid

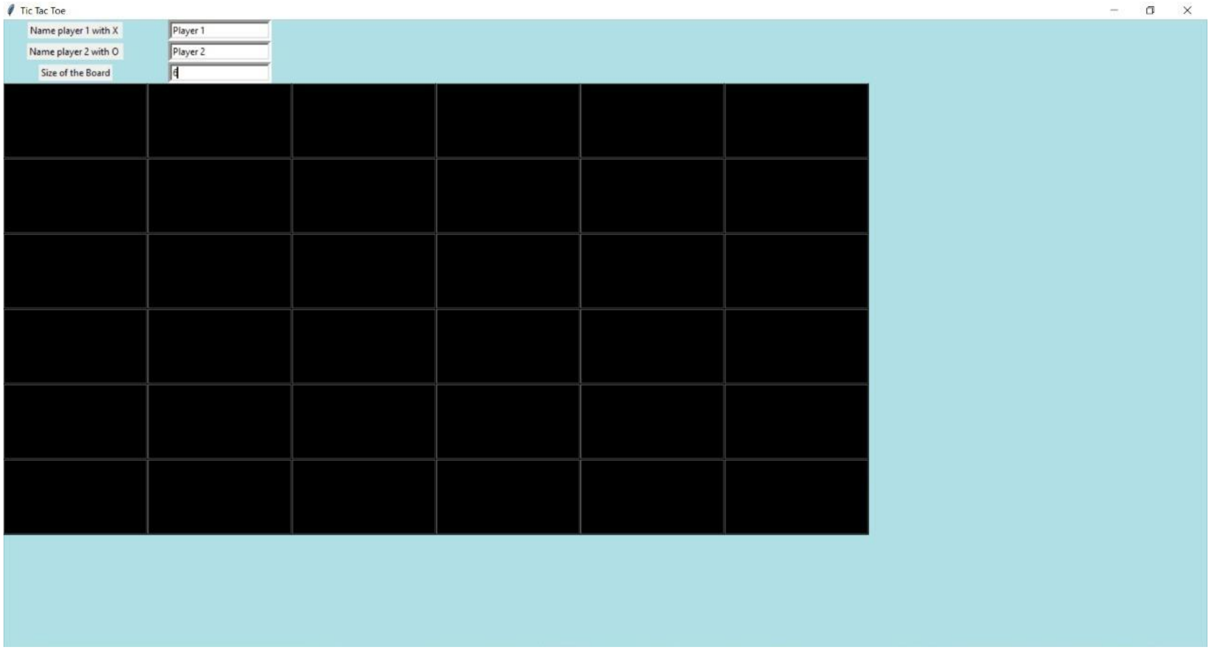


Name player 1 with X: Playrer 1

Name player 2 with O: Playrer 2

Size of the Board: 5

6x6 Grid

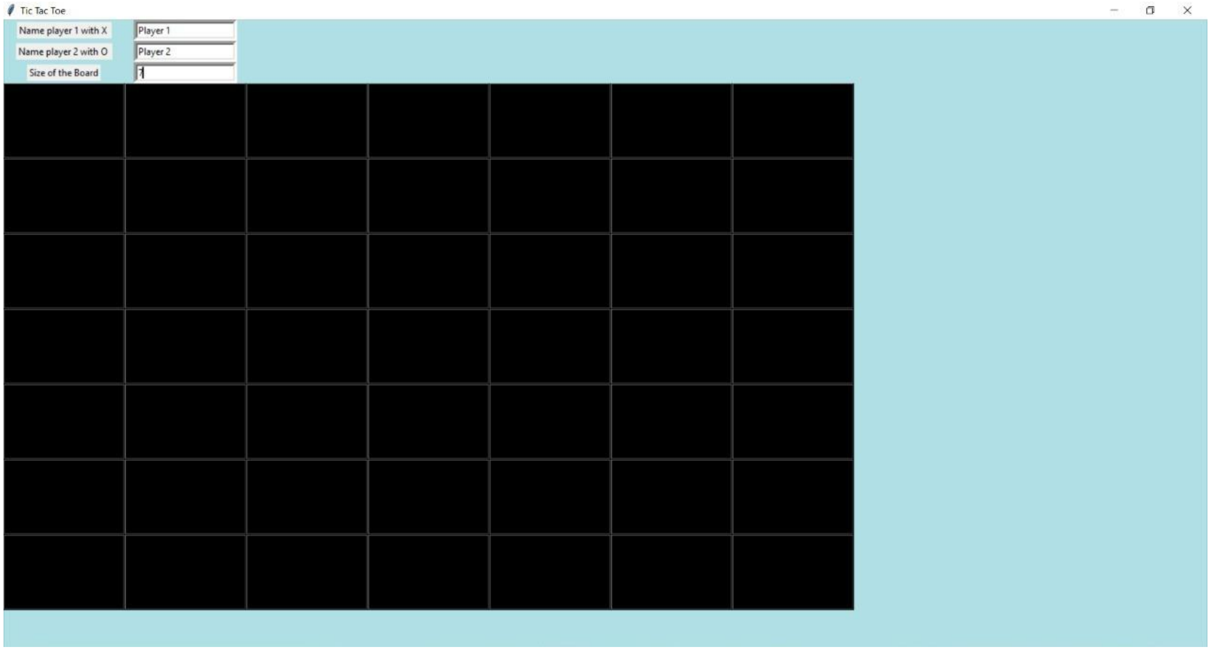


Name player 1 with X: Player 1

Name player 2 with O: Player 2

Size of the Board: 6

7x7 Grid



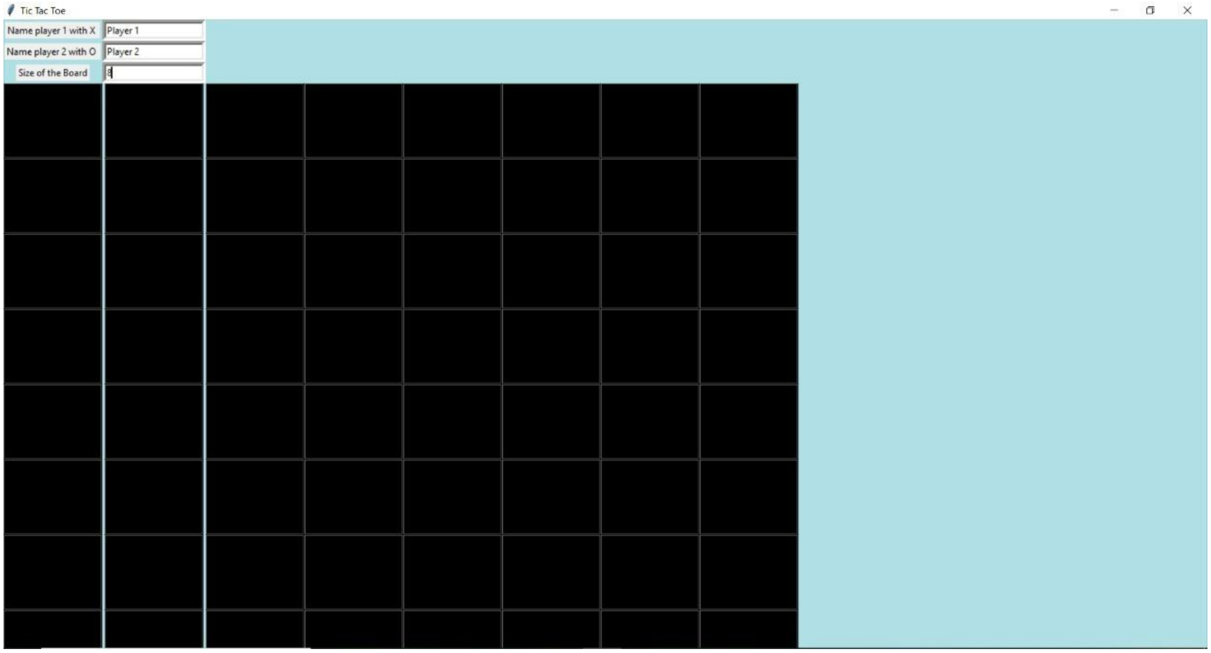
Tic Tac Toe

Name player 1 with X: Player 1

Name player 2 with O: Player 2

Size of the Board: 7

8x8 Grid



Tic Tac Toe

Name player 1 with X: Player 1

Name player 2 with O: Player 2

Size of the Board: 8

6. SYSTEM TESTING

6.1. INTRODUCTION:

Software Testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for III planned through testing.

TESTING OBJECTIVES

These are several rules that can save as testing objectives:

- Testing is a process of executing program with the intent of finding an error.
- A good testcase is one that has a high probability of finding an undiscovered error.

Test Levels

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or darkness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2. TESTING METHODS

6.2.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application.

6.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

6.2.3 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

6.2.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test.

6.2.5 White Box Test

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.2.6 Black Box Test

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

6.2.7 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

6.2.8 Integration Testing

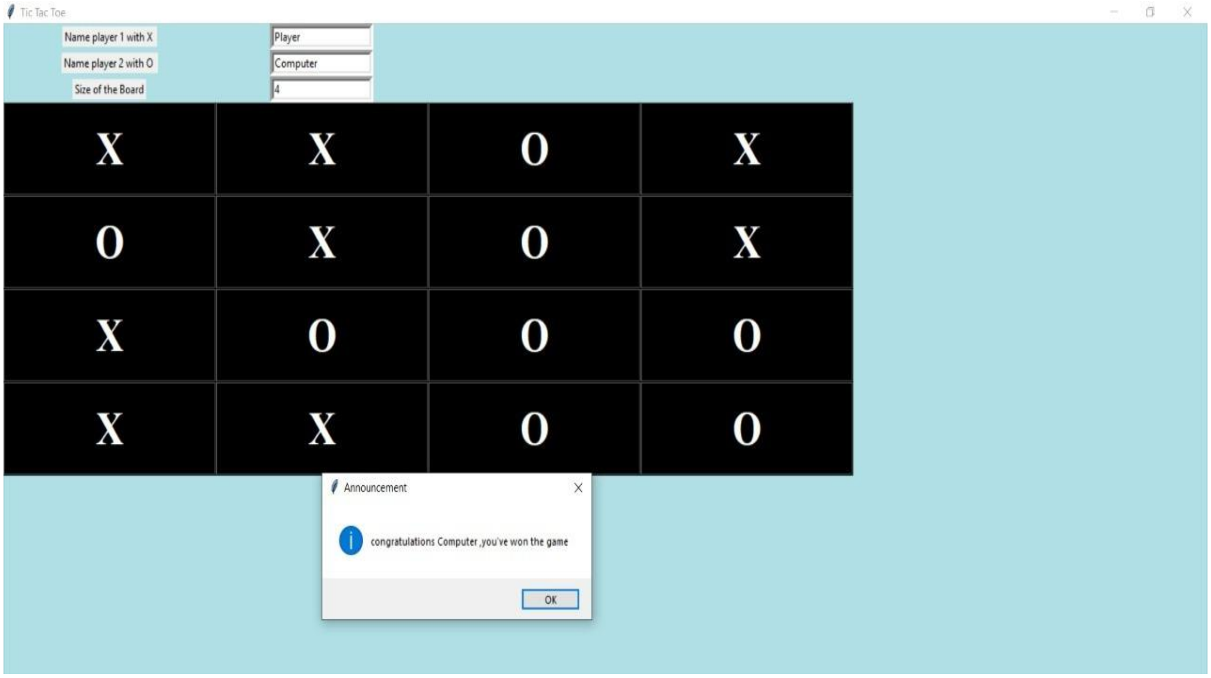
Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

6.2.9 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user.

6.3. Test Cases

Computer vs Player



Name player 1 with X: Player

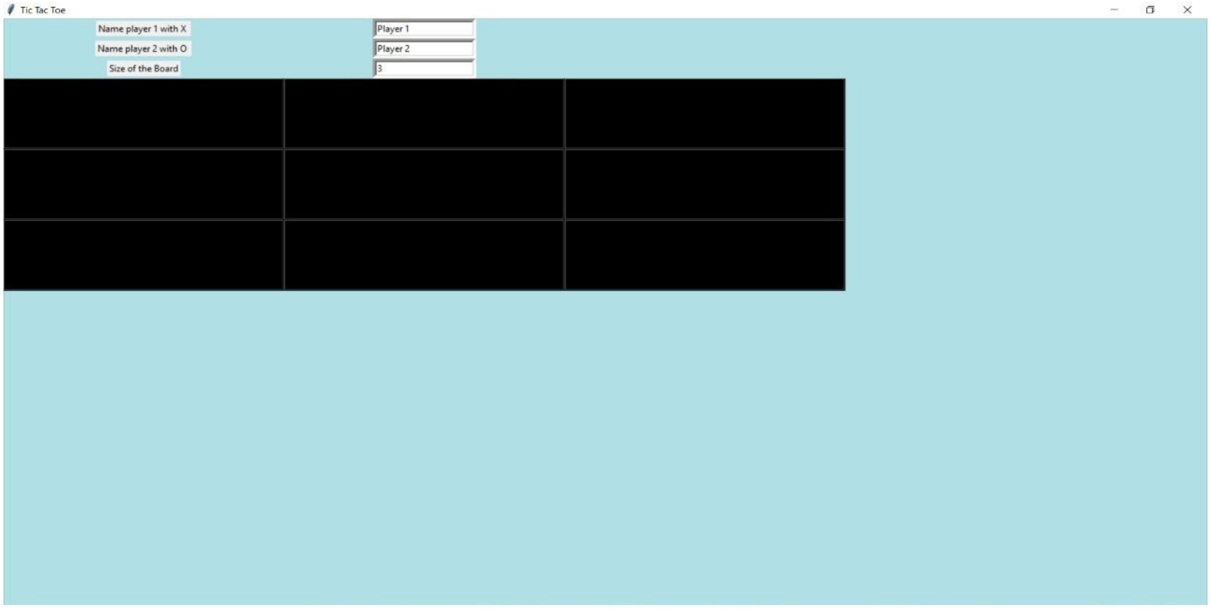
Name player 2 with O: Computer

Size of the Board: 4

X	X	O	X
O	X	O	X
X	O	O	O
X	X	O	O

congratulations Computer ,you've won the game

Player vs Player

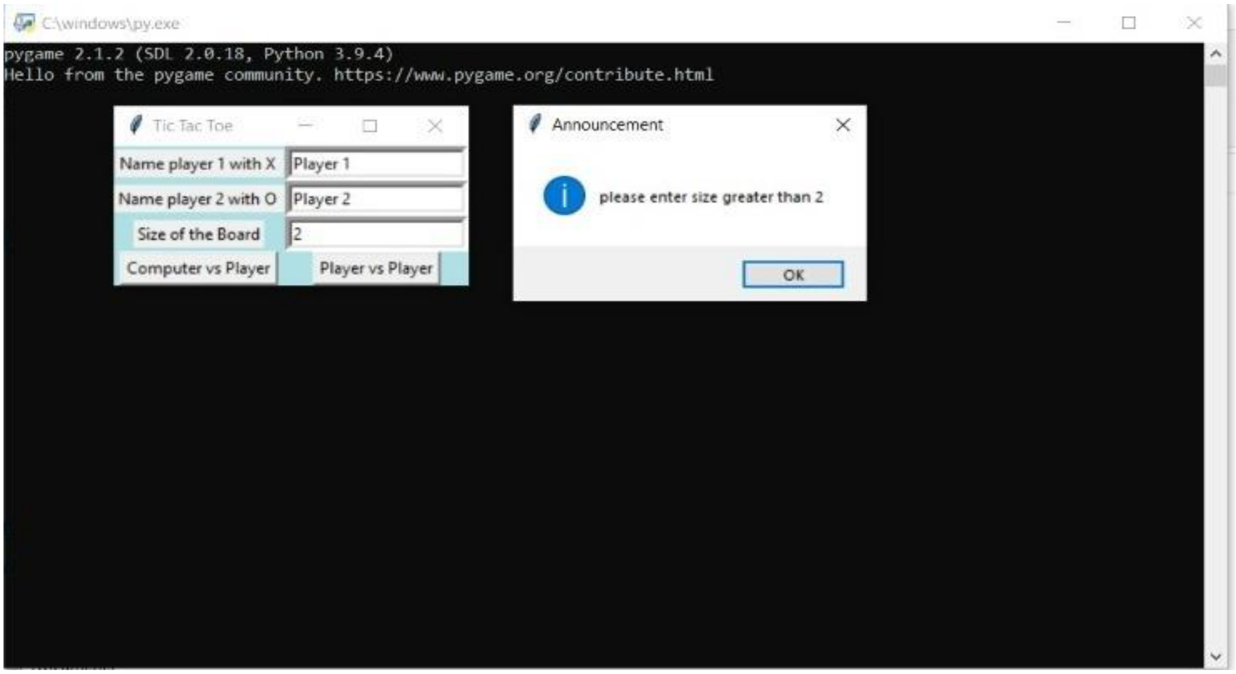


Name player 1 with X: Player 1

Name player 2 with O: Player 2

Size of the Board: 3

Minimum Grid Size Alert



```
pygame 2.1.2 (SDL 2.0.18, Python 3.9.4)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

Name player 1 with X: Player 1

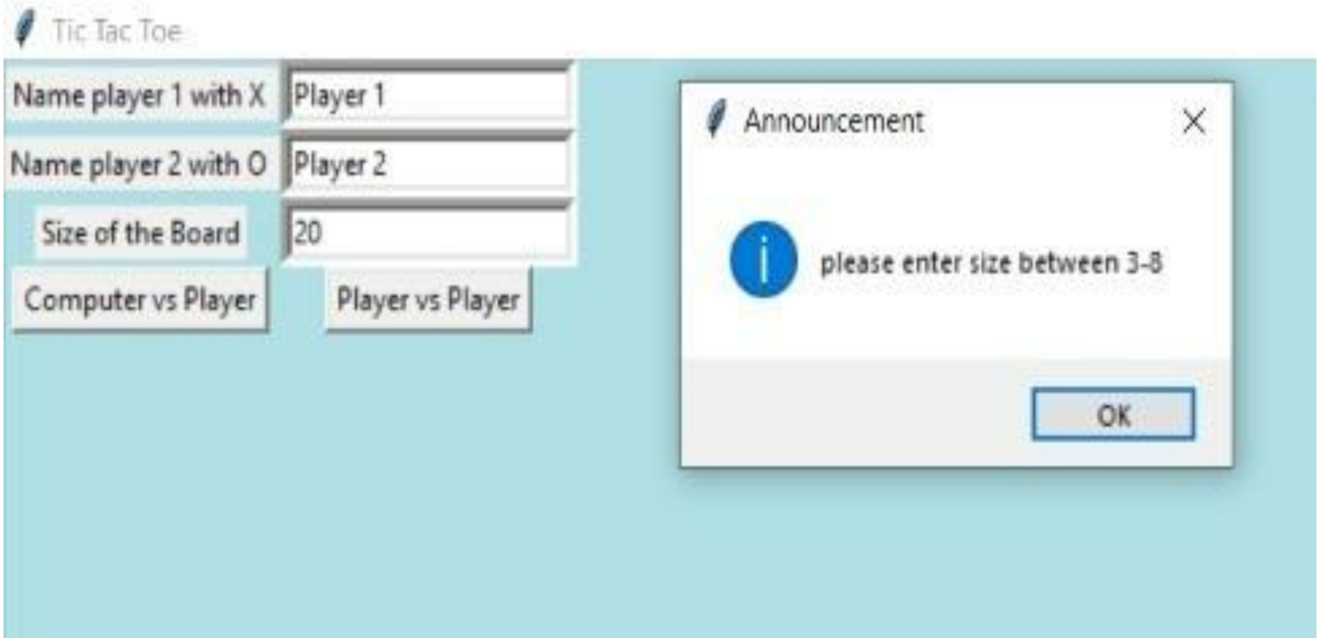
Name player 2 with O: Player 2

Size of the Board: 2

please enter size greater than 2

Maximum Grid Size Alert

In this test case horizontally the grid size is displaying but vertically it is not because of Screen Resolution which the code is being executed



Name player 1 with X: Player 1
Name player 2 with O: Player 2
Size of the Board: 20

please enter size between 3-8

7. Conclusion

Tic tac toe is a game which has been computed successfully and was also tested successfully by including “test cases”. It is easy to play and an interesting game.

Our project has been designed keeping in view of an user where the player select the size of grid with the alerts being displayed for limits of size of grid, it is very flexible.

This game has the benefits of choosing modes of play which are Player vs Player and Computer vs Player and music being played in the background   makes the game user friendly.

However, this is our one of the earliest efforts in making such a software on such platform, we hope for future betterment.

8. Bibliography

1.https://www.google.com/amp/s/www.geeksforgeeks.org/python-implementation-automatic-tic-tac-toe-game-using-random-number/amp/

2. https://youtu.be/xx0qmpuA-vM

3. https://youtu.be/9UUG9ErbOig

4. https://youtu.be/HYTwXvzxeSE

5. https://www.geeksforgeeks.org/tic-tac-toe-game-with-gui-using-tkinter-in-python/

6. https://docs.python.org/3/library/tk.html

7. https://youtu.be/eoppKOt-jdc

9.APPENDIX

9.1 INTODUCTION:

About Python:

- Python is a widely used general-purpose, high level programming language. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.
- Python is a programming language that lets you work quickly and integrate systems more efficiently.
- It is used for web development, software development, mathematics, system scripting.
- Python can be used on a server to create web applications, can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files, handle big data and perform complex mathematics
- Python can be used for rapid prototyping, or for production-ready software development.
- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

About Tkinter:

Tkinter is the most commonly used library for developing GUI (Graphical User Interface) in Python

We can make windows, buttons, show text and images amongst other things.

About Pygame:

- Pygame is a platform set of Python modules which is used to create video games.
- It consists of computer graphics and sound libraries designed to be used with the Python programming language.