FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ

A PROJECT REPORT ON “BINGO GAME USING MULTILINKED LISTS”

SUBMITTED BY

NAME	:	AHMAD AL ALI
ID	:	2221251365
COURSE	:	DATA STRUCTURES

Acknowledgment

I would like to extend my deep gratitude to **Dr. Öğr. Üyesi Berna KİRAZ** and the assistant **Arş. Gör. Zeliha KAYA** for providing us with this insightful project on MultiLinkedList. Their guidance and support throughout the previous weeks have been invaluable in helping us understand the intricacies of linked lists. Their dedication to teaching and mentoring has significantly enhanced our understanding of the topic and has contributed immensely to our learning experience. We are truly grateful for their expertise, patience, and encouragement.

Thank you once again for your unwavering support and commitment to our academic growth.

Sincerely,

Ahmed Al - Ali

[illegible]

Technologies Used :

UI components design : Figma , Java Swing UI (Netbeans)

Game UI implementation : Java Swing UI library

Language : **Java**

Design Overview:

The Bingo Game project is built with a focus on simplicity, flexibility, and efficiency. It starts by a welcoming screen asking for the two players names and then the game starts by generating random numbers according to certain rules (Specified in the next page). The concept of the applications is to keep pressing the **Generate button** button that does that functionality.

At the core of the game's design is how the Bingo card is represented. Using basic structures called MultiNode and MultiLinkedList, the game creates a grid-like layout for the Bingo card. Each MultiNode represents a number on the card, and the MultiLinkedList puts them together, making it easy to organize and use the card.

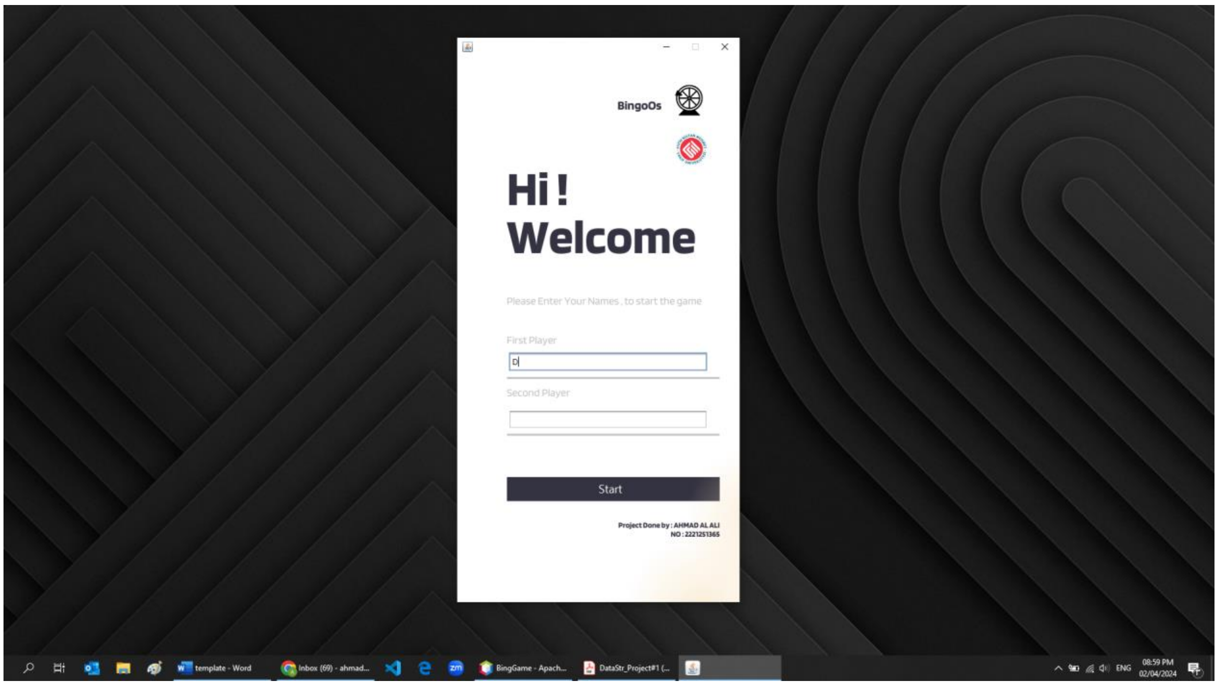
Implementation -Game Mechanics Management:

The game has a set of tools to manage how it works. This includes ***generating random*** numbers for Bingo, **updating what players see on the screen**, and **keeping track of when someone wins.** Functions like **generateNumber** make sure numbers appear randomly, while others like **labelchangerforgame** and **checkAndModifyLabels** change what players see on the card .These methods have functionalities like crossing numbers (making them red), making the generated numbers appear on screen etc. These methods help the game run smoothly.

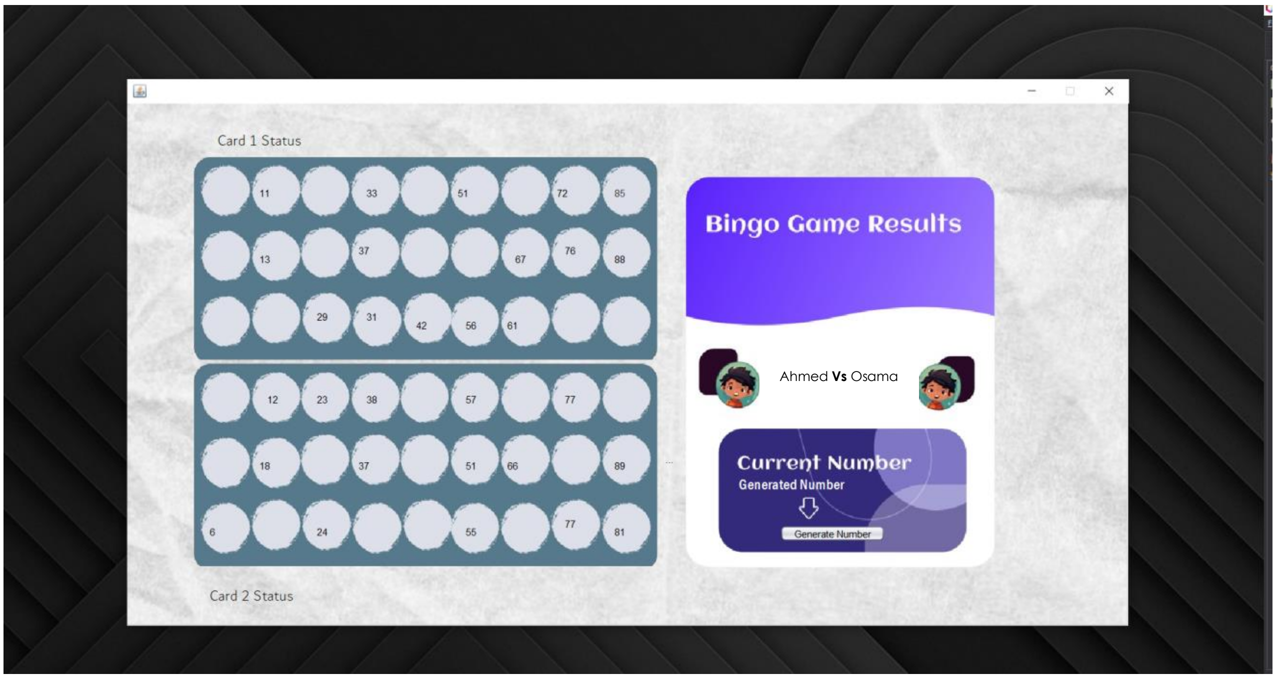
User Interface Integration:

Making the game easy to use is important. By designing simple buttons and menus, players can interact with the game easily. When a player clicks a button, the game responds by **doing things like generating new numbers** or updating what's shown on the screen. This interaction keeps players engaged and makes the game more enjoyable.

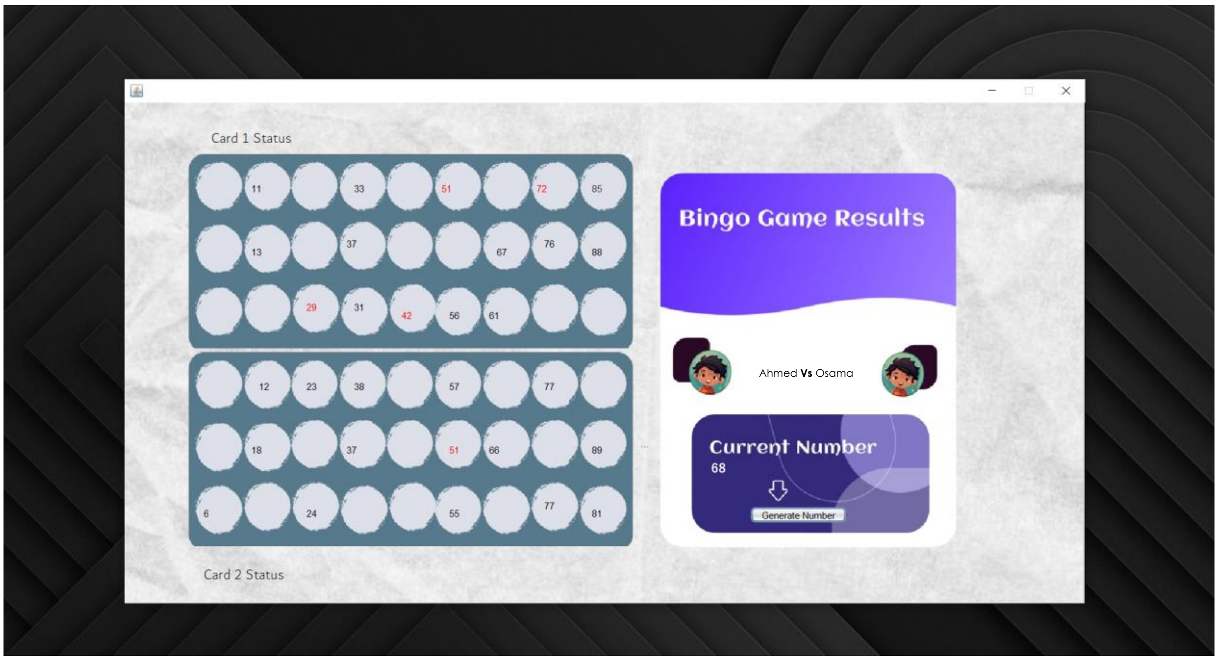
AN EXEMPLARY SCENARIO


BingoOs
Hi !
Welcome
Please Enter Your Names , to start the game
First Player
D
Second Player
Start
Project Done by : AHMAD AL ALI
NO : 2221251365

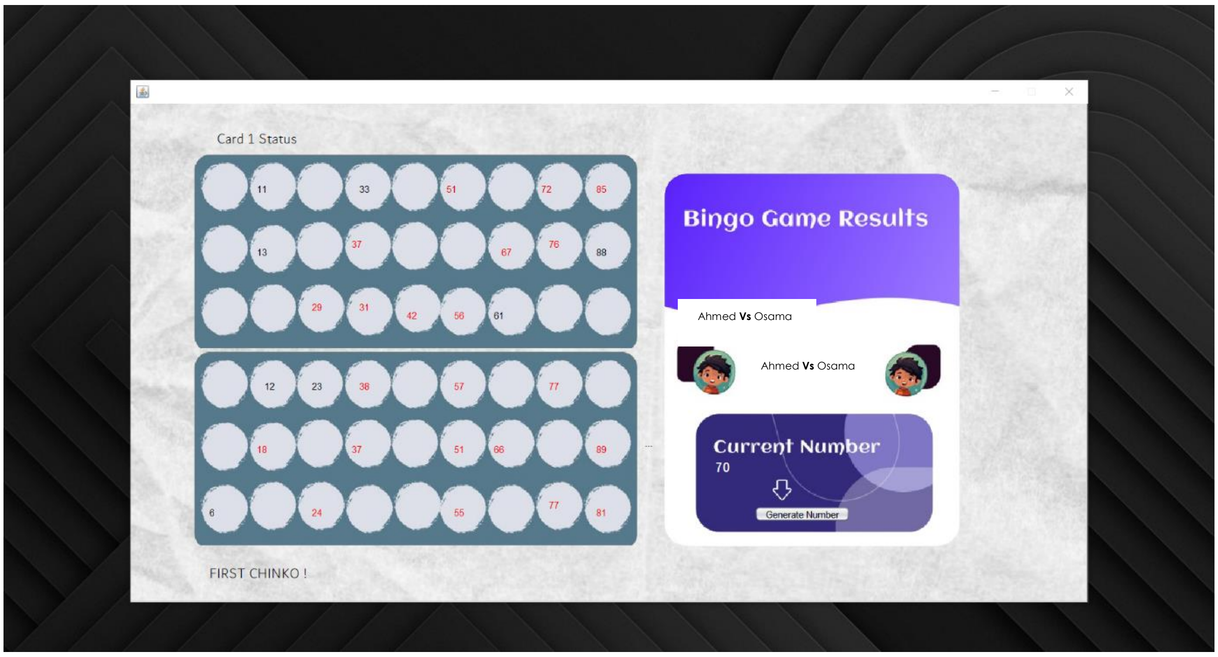
The User is asked to enter two player names. If any of the fields is empty "*An error message appears".*



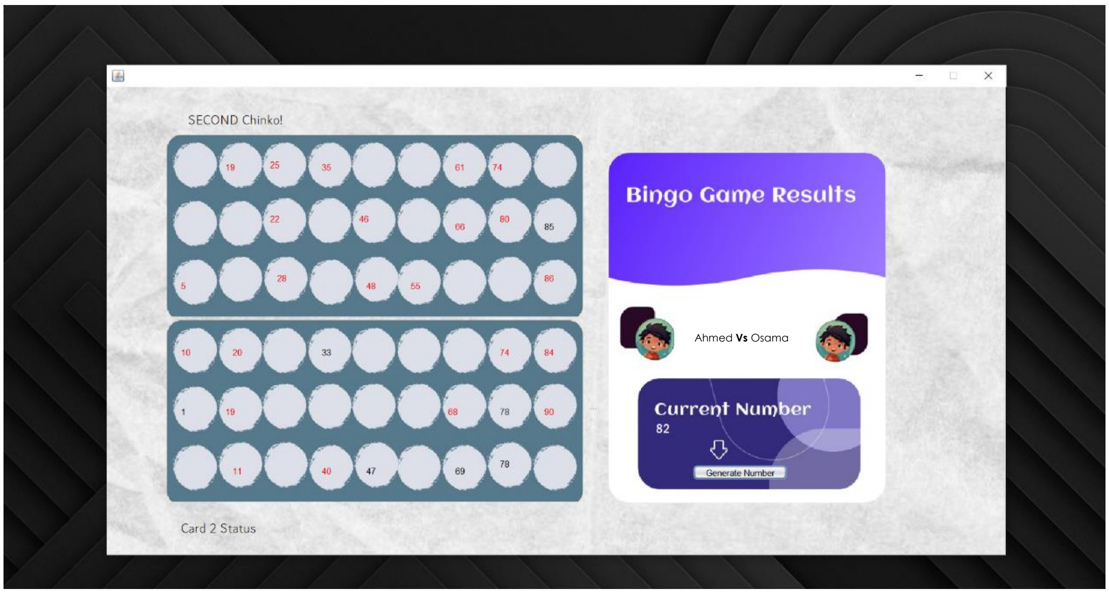
The game starts with random numbers from 1-90 (not repeated) on each card. There are places which are blocked and shown to be as blank.

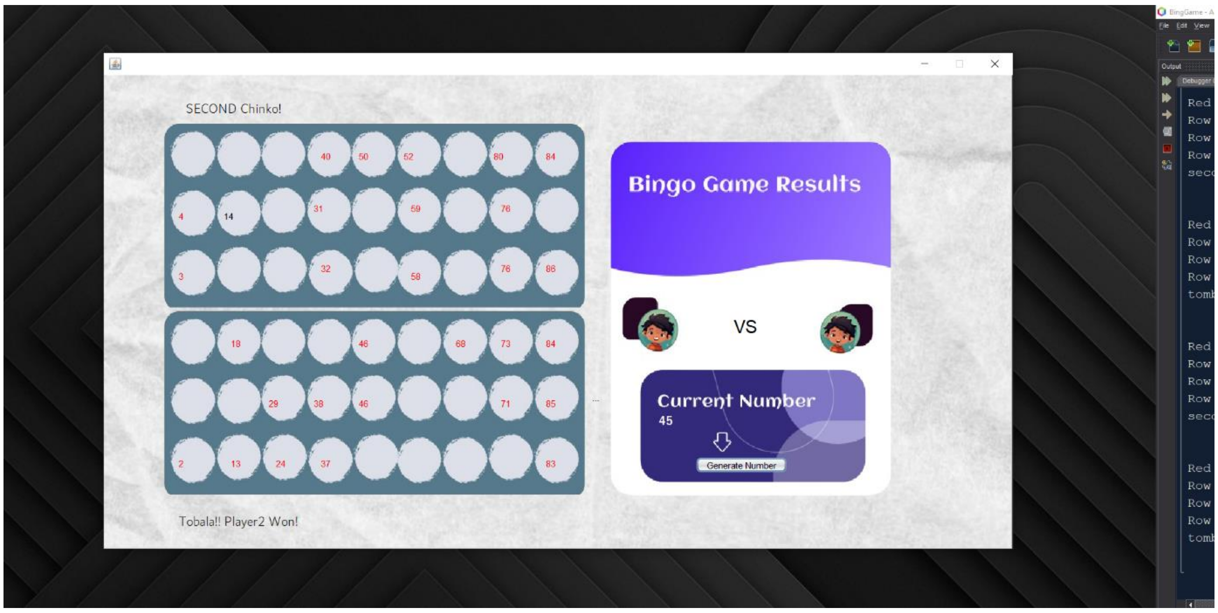


The game starts by pressing generate number that generates numbers also from 1-90 (not repeated) and checks if any of the cards has the generated number. If it finds a number on any of the cards it set as red color.

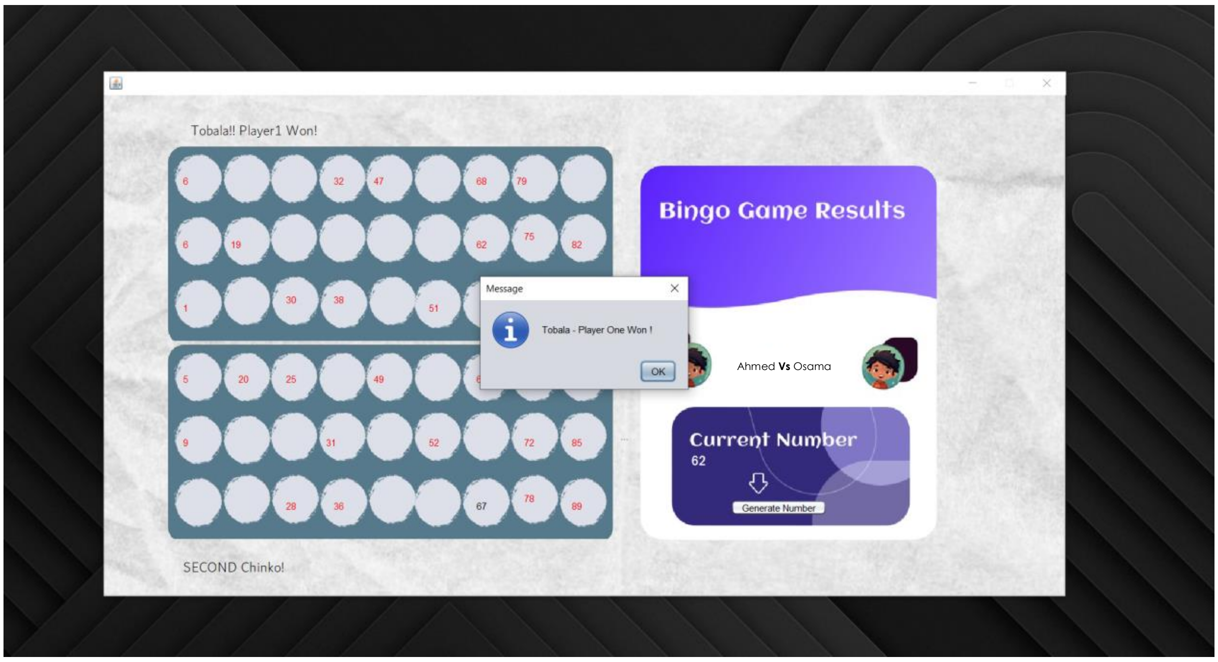


If a full row is set to be as red (Crossed out) the text under/ above the card is set to say (First Or Second Chinko).





If any of the cards has all the numbers crossed out. Then that certain player wins and the text is set to say : "Tobala". A message also appears.


Tobala!! Player1 Won!
Bingo Game Results
Message
Tobala - Player One Won !
OK
Ahmed Vs Osama
Current Number
62
Generate Number
SECOND Chinko!