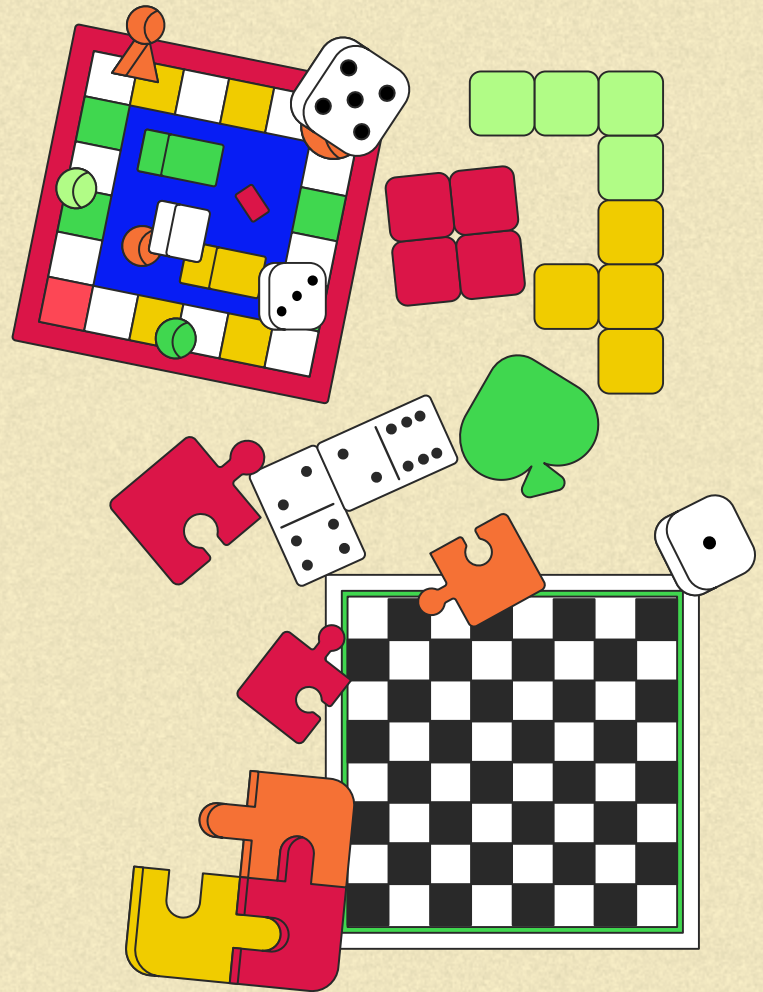


# **Project presentation The game - Connect 4**

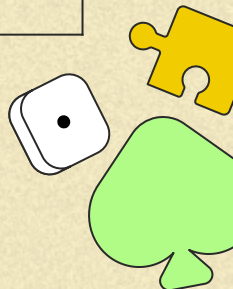
Bakhtiyor Karimov – 201ADB007



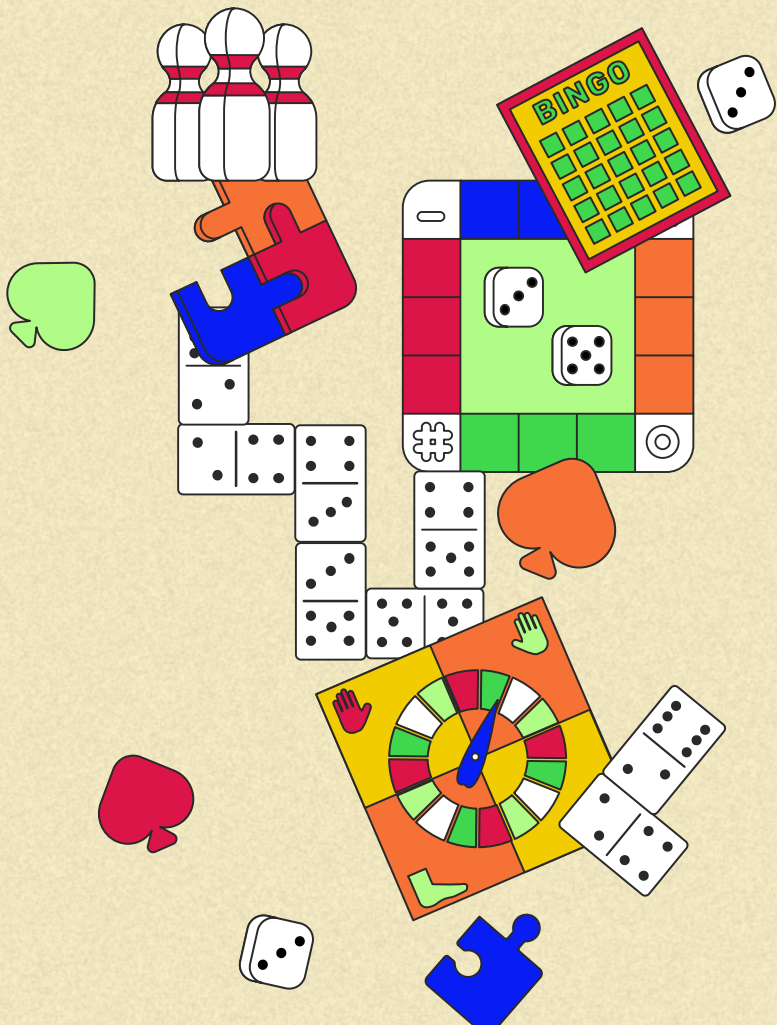


# Contents of the presentation

<b>Introduction</b>	Brief description of the project
<b>Requirements</b>	Description of the requirement assumptions
<b>Design</b>	Design options and selection, selection justification
<b>Test</b>	Insight on testing phase
<b>Lessons learned</b>	What was learnt/obtained during the process
<b>DEMO</b>	Demo – what to show







# Introduction

**The Connect 4 Game is a two-player game where players start the game with a disc color (blue or yellow).**

**The objective is to be the first player to stack four discs of their own color in a row, either horizontally, vertically, or diagonally on a grid.**

## Used Technology:

- **Python**

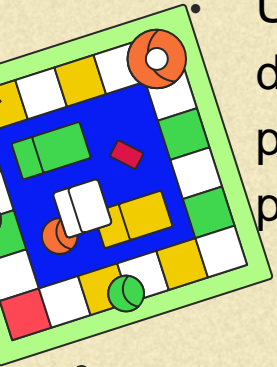
## Modules:

- **numpy**
- **pygame**
- **sys**
- **math**



# Requirements

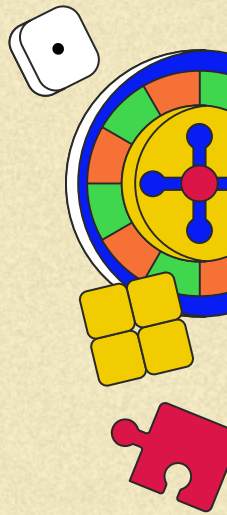
- Game should be for 2 players
- The game should have a grid to place the discs and 21 blue and 21 yellow discs
- The game must comply with the rules of winning - stacking 4 discs in a row horizontally, vertically, and diagonally.
- Users click the start the game button and the game starts with a blue disc for the client's brief. Then they drop the disc over the grid aiming to place 4 of them in a row. Moves are implemented sequentially between players - one after another turn by turn.





# Requirements 2

- Game must display the grid and colored discs
- Players are able to drop their disc from a particular column
- Game checks for the row of discs to determine the winner
- Once the game is over, display the winner







**DESIGN**



## • **Classes**

- Players
- Disc
- Grid
- Button

## • **Attributes for Players**

- Player number, Color

## • **Attributes for Disc**

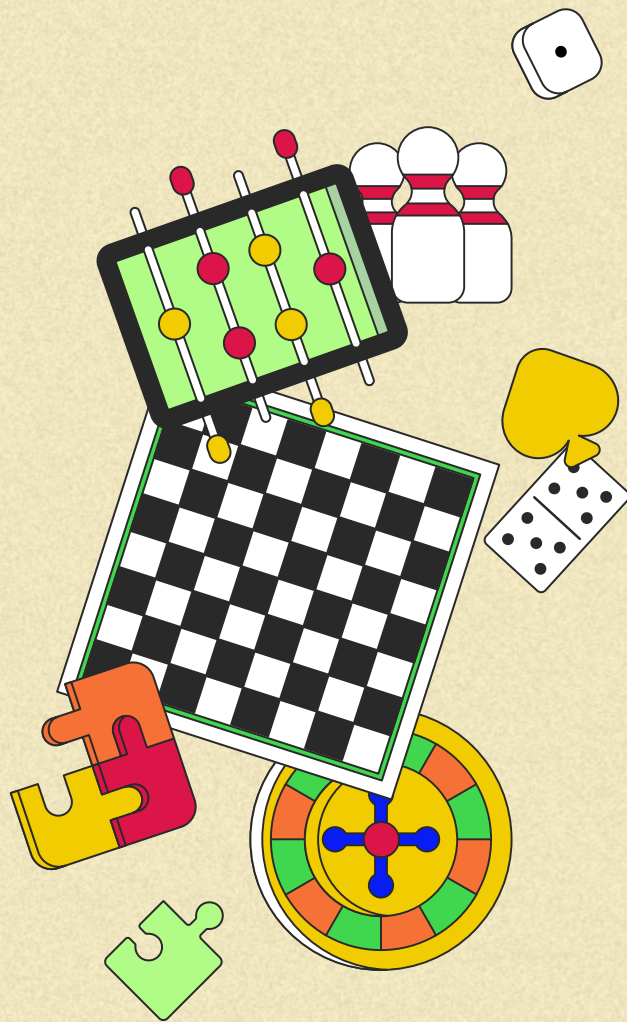
- Player

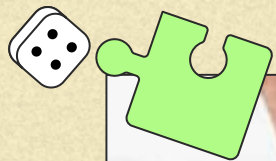
## • **Attributes for Grid**

- Rows, Columns, Height, Radius, board

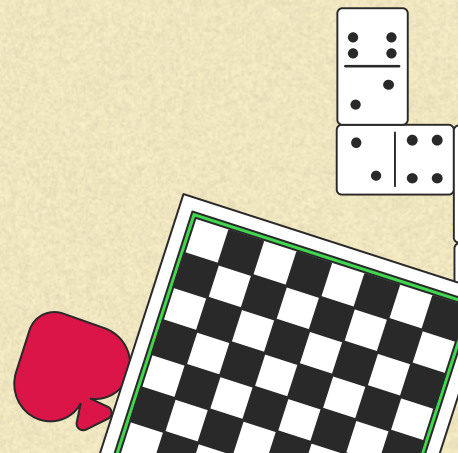
## • **Attributes for Button**

- Text, width, height, pos, elevation

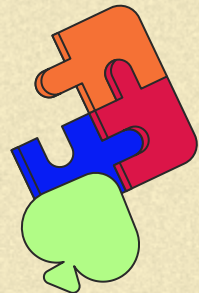




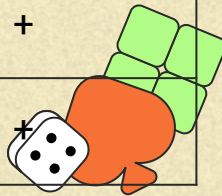
# TESTING

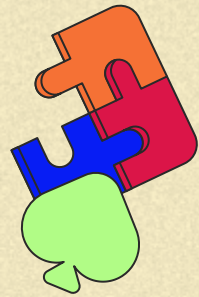






# Schedule

Test ID	Description	Expected Results	Actual Results
01	Start a game with button	The pop-up window closes after clicking on the button, game starts to play	+
02	Game status show colors	Game status shows player disc color	+
03	Disc throw in a column for user 1	Player 1 selects a column and puts the disc	+
04	Game status update for a player	Game status is updated respectively in turn of playing player	+
05	User is unable to place a disc to full column	Column remains unchanged	+
06	Game Over condition when 4 disc in row	Game status shows winner	+
07	Pop-up window when game is over	"Game Start" window pops-up	



- **Board Class:**

- Equivalence Classes:

- - Valid board dimensions (e.g., 6 rows x 7 columns)
    - - Invalid board dimensions (e.g., 0 rows x 0 columns)

- Boundary Values:

- - Minimum and maximum row and column counts

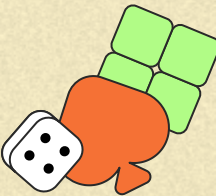
- **Unit testing**

- Several functions have been written to conduct unit testing on the program code

- Check if a Player object is correctly created with the correct ID and color.

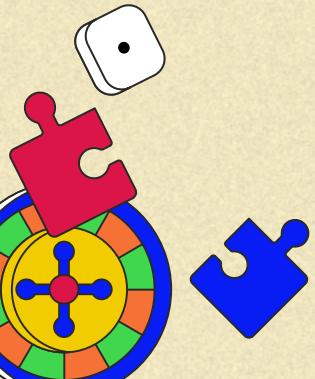
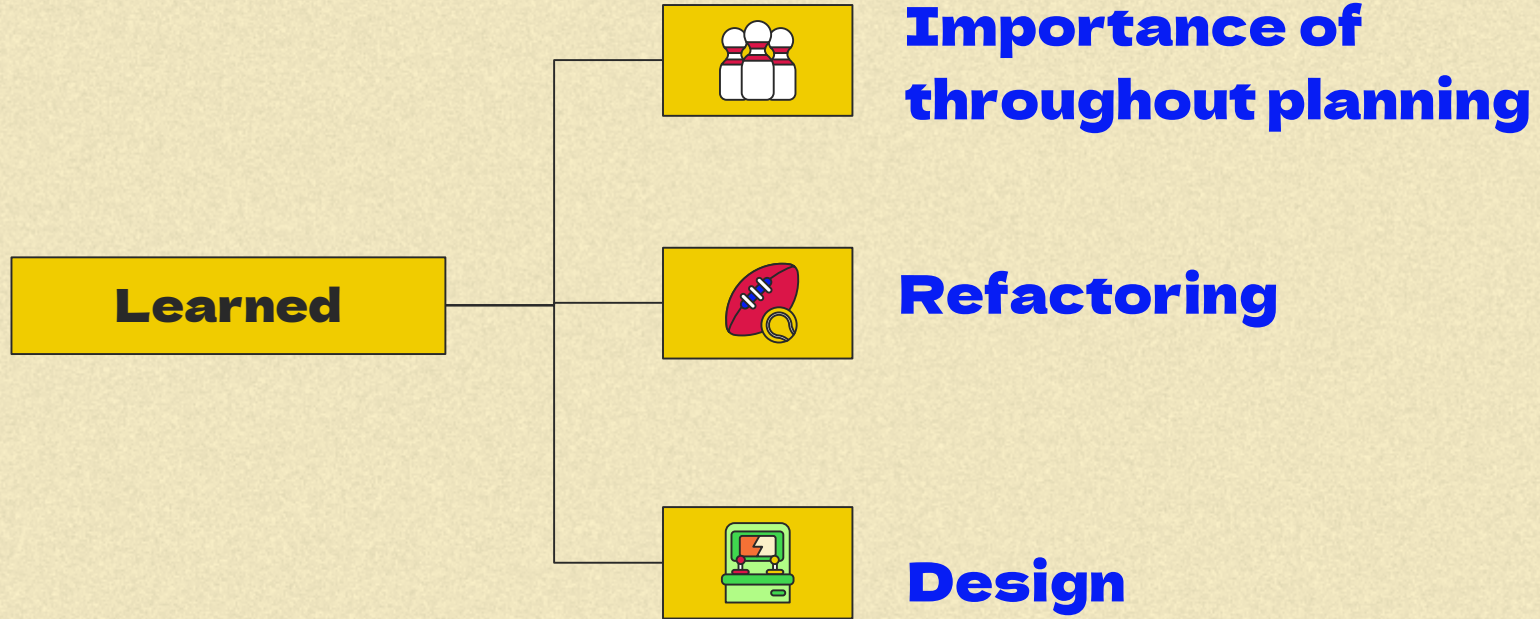
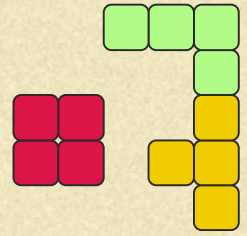
- Check if a Disc object is correctly related to a Player object.

- Check if a Board object is correctly created with the correct row and column numbers count.



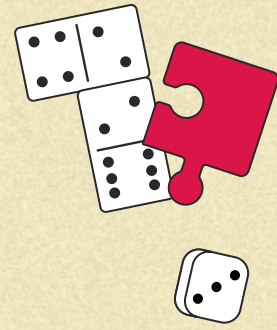


# Lessons learned

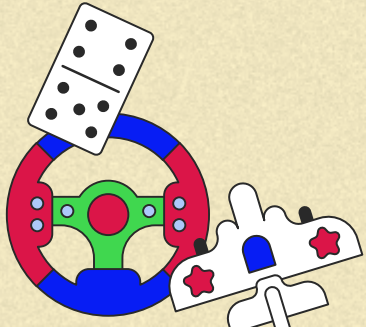




# DEMO



The listed requirements have been  
implemented as functioning features  
and they will be demonstrated





# Thanks!

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

