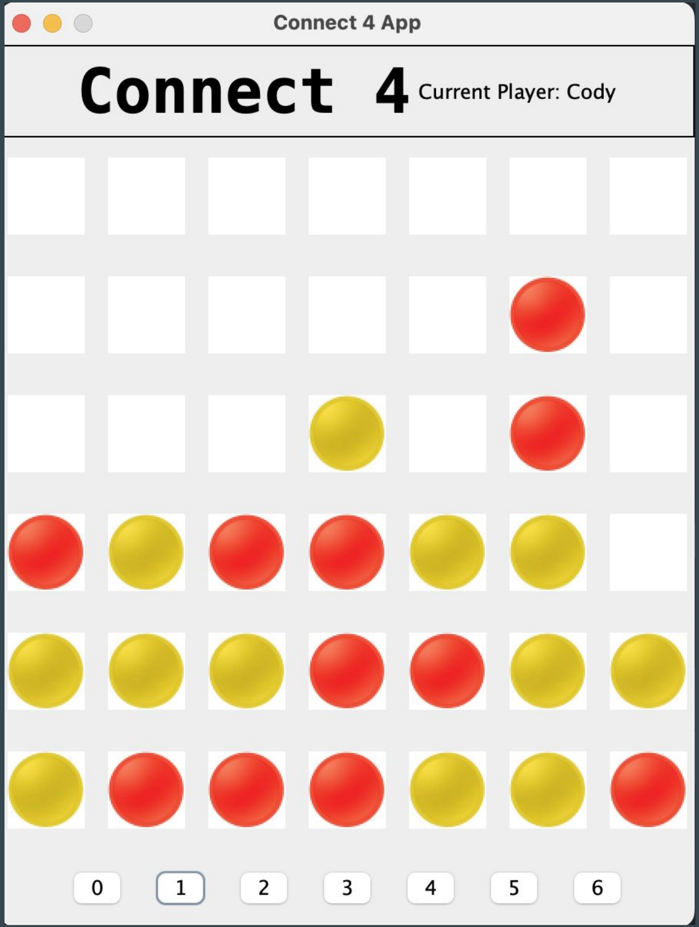
Connect 4

Jake Snitily
Cody Kesselring
Tu Do
Alex Sautereau

Project Overview

- Connect four
    - Usable GUI
    - Proper board logic
    - Two player compatibility
    - Intuitive design

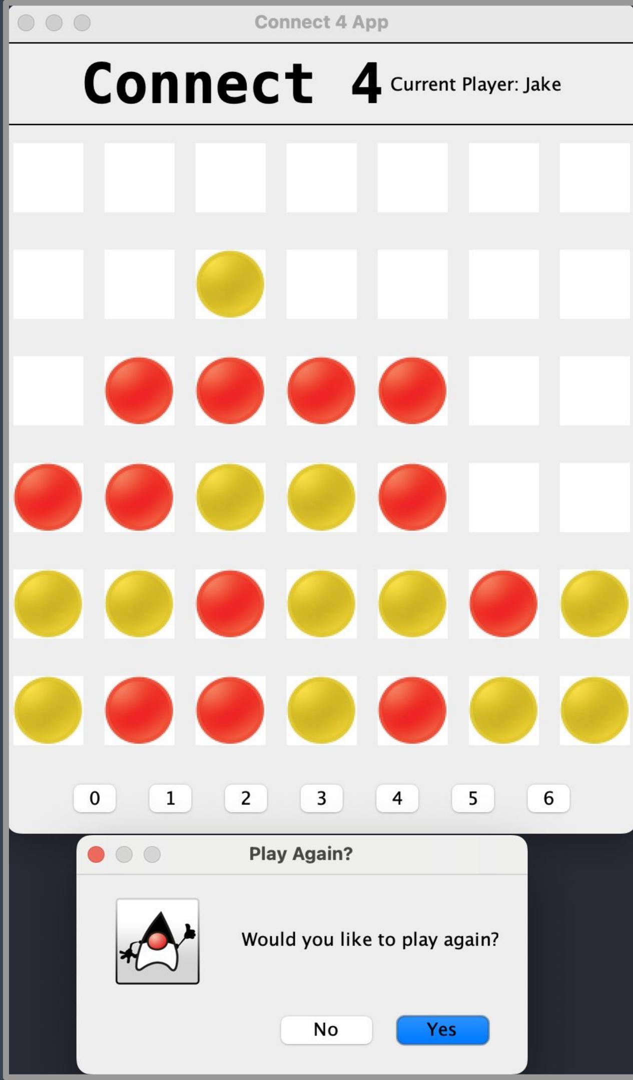


Connect 4 App

Connect 4 Current Player: Cody

Description

- 6x7 grid board
- 2 players take turns dropping pieces into the grid columns
- Goal is to connect four pieces in any linear direction
- Game is over once four pieces are connected
- Ties are possible



Connect 4 App

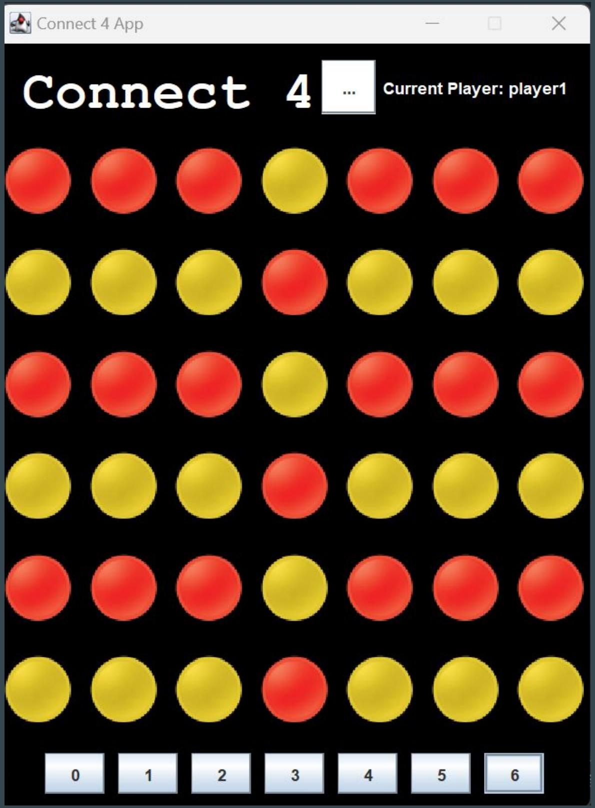
Connect 4 Current Player: Jake

Play Again?

Would you like to play again?

Project requirements

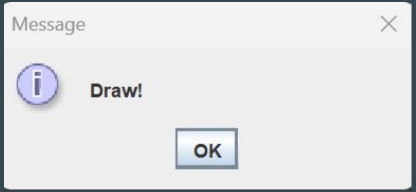
- Title screen with name input
- Usable board GUI that simulates physical Connect Four
- Place piece in columns that fall to the bottom
- Check for wins
- Notify winner and end game



Connect 4 App

Connect 4

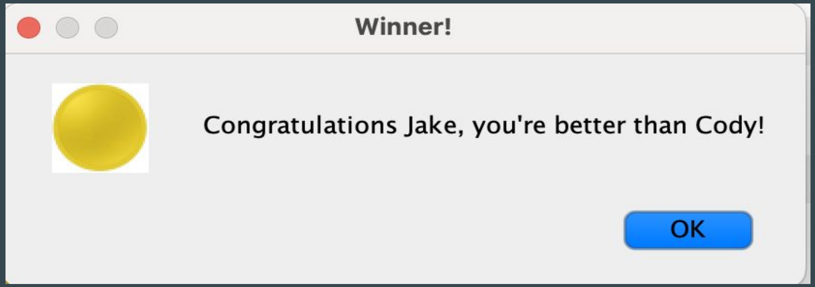
Current Player: player1



Message

Draw!

Project Solution Approach

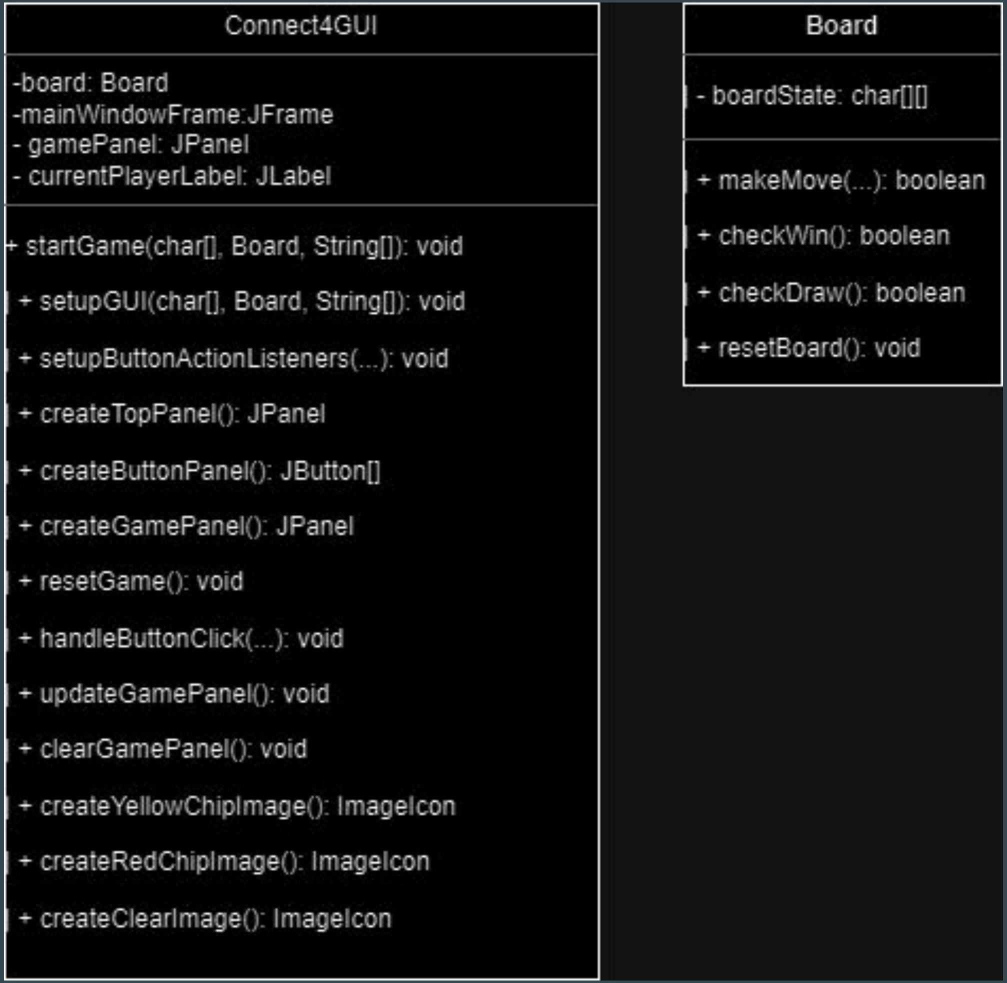


Winner!

Congratulations Jake, you're better than Cody!

- Major design components-
  - **MainGame** class: Initiates the GUI & originally launched terminal based prototype
  - **Connect4GUI** class: Build up the playable GUI & manage GUI events interaction with game
  - **Board** class: Manages the player movement, keeps track of pieces, win & draw checking
- What game/ui features did you identify and work towards in your game?
  - **Player Interaction:**
    - We wanted players to input their names through a dialog box before the game starts.
    - Players pick the columns of their piece drop on every turn
  - **GUI Features:**
    - GUI dynamically reflects each move on the game board.
    - Current player updates after every valid move
  - **Game Logic:**
    - Win & Draw detection, checked after every move by the game board
    - Manages player moves and updates the board accordingly.
    - Replayability resets the board & GUI, keeping winning players turn

UML Design



Connect4GUI

-board: Board
-mainWindowFrame:JFrame
- gamePanel: JPanel
- currentPlayerLabel: JLabel

+ startGame(char[], Board, String[]): void
+ setupGUI(char[], Board, String[]): void
+ setupButtonActionListeners(...): void
+ createTopPanel(): JPanel
+ createButtonPanel(): JButton[]
+ createGamePanel(): JPanel
+ resetGame(): void
+ handleButtonClick(...): void
+ updateGamePanel(): void
+ clearGamePanel(): void
+ createYellowChipImage(): ImageIcon
+ createRedChipImage(): ImageIcon
+ createClearImage(): ImageIcon

Board

- boardState: char[][]

+ makeMove(...): boolean
+ checkWin(): boolean
+ checkDraw(): boolean
+ resetBoard(): void

Team collaboration

- Discord
- No github problems
- No branches
- Coded 100% independently
- Next time should set more due dates to get the project rolling sooner

Testing, Validation, and acceptance Plan

Unit tests: Board class: initialization, moves, win conditions, tie games, and board reset.

Integration Tests: Win conditions.

Functional Tests: Evaluate holistic game functionality—player turns, win/tie conditions

#1 way we'll be able to say "Our project is deliverable"?

achieved a fully functional Connect 4 game with all essential features implemented.

Live Demo!

Lessons learned / Experiences we’ve had

As a team project we have learned:

Communication Essentials:

Clear & Consistent: Vital for tracking progress, addressing challenges, and acknowledging contributions.

Regular Check-ins: Ensures prompt problem-solving and maintains a collaborative team spirit.

Feedback Loop: Encourage open discussions to refine ideas, resolve conflicts, and drive continuous improvement.