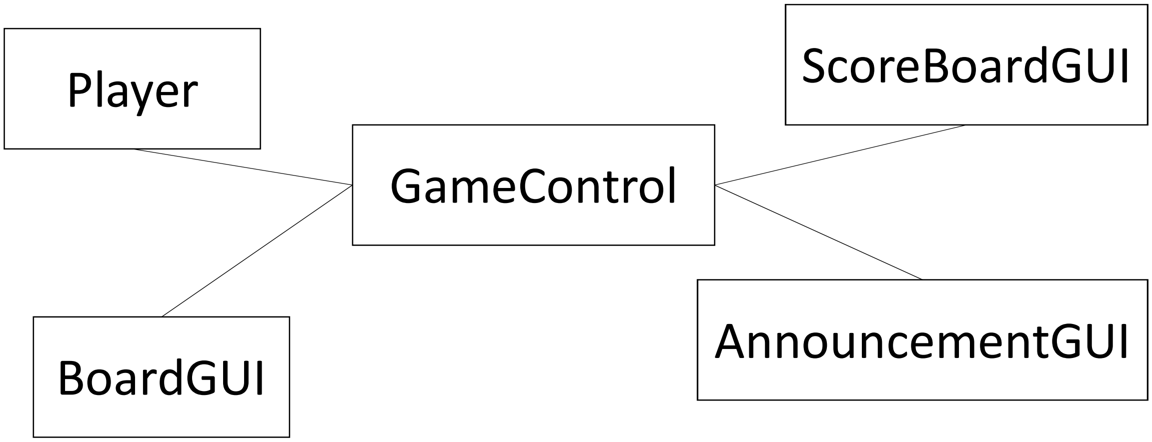
Building Tic Tac Toe with SWING API Part 2

CSCI 2300

Last time: Need a new class to maintain game state

- GameControl
  - Maintains and updates game state (implements scenarios 1 & 2)
    - Who is the current player
    - What moves have been made (state of the game board)
  - Needs access to:
    - ScoreBoardGUI
    - AnnouncementGUI
    - Both players
    - Board buttons

Tight coupling: classes have a high degree of dependence of classes on each other. If one class changes, the other may have to change too



How can we fix it?

- Use Interfaces
- IBoard, IScore, IAnnouncement, IPlayer
- Delegate responsibilities to classes that implement these interfaces
- Example:
  - IBoard – implements methods for keeping the state of the board
  - IScore – implements methods for keeping track of score
  - etc

Semi-independent layers of the application

GUI layer – user interface

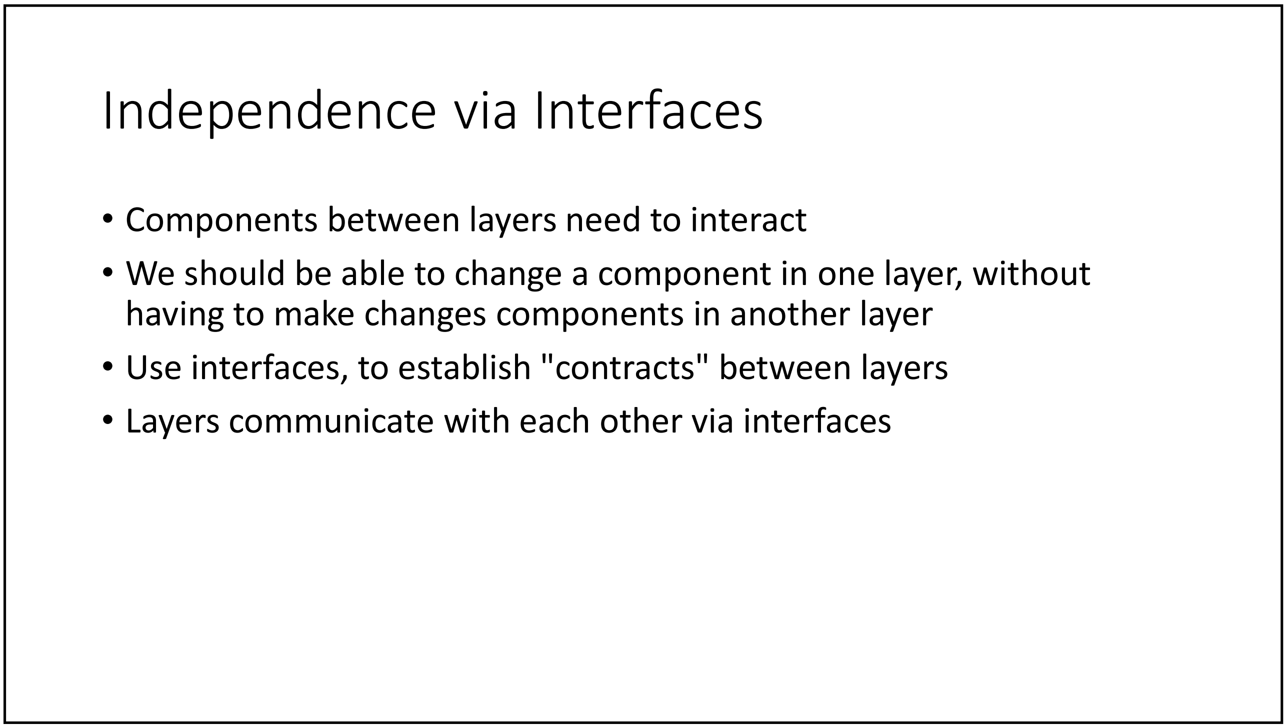
Game state layer – the current state of the game

Logic layer – the transition between states

1. GamePiece – represents an X or an O
2. BoardGUI – visual representation of the board
3. BoardButtonListener – responds to player clicking a button on the board
4. GameControl – implements scenarios 1 and 2
5. Board - has 3x3 matrix of GamePiece objects

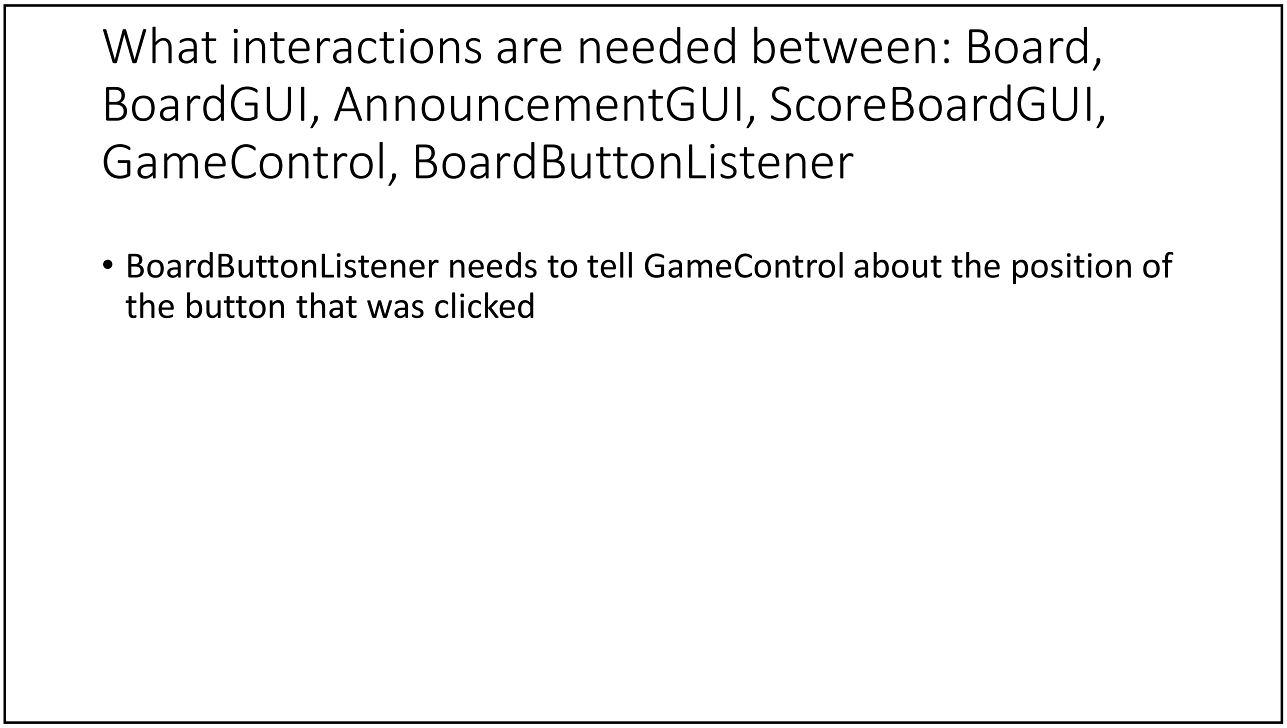
Classify the components above as (G)UI, (L)ogic or (S)tate

A. 1G 2G 3G 4L 5S
B. 1S 2G 3G 4L 5S
C. 1L 2G 3L 4L 5S
D. 1S 2G 3L 5S 5S



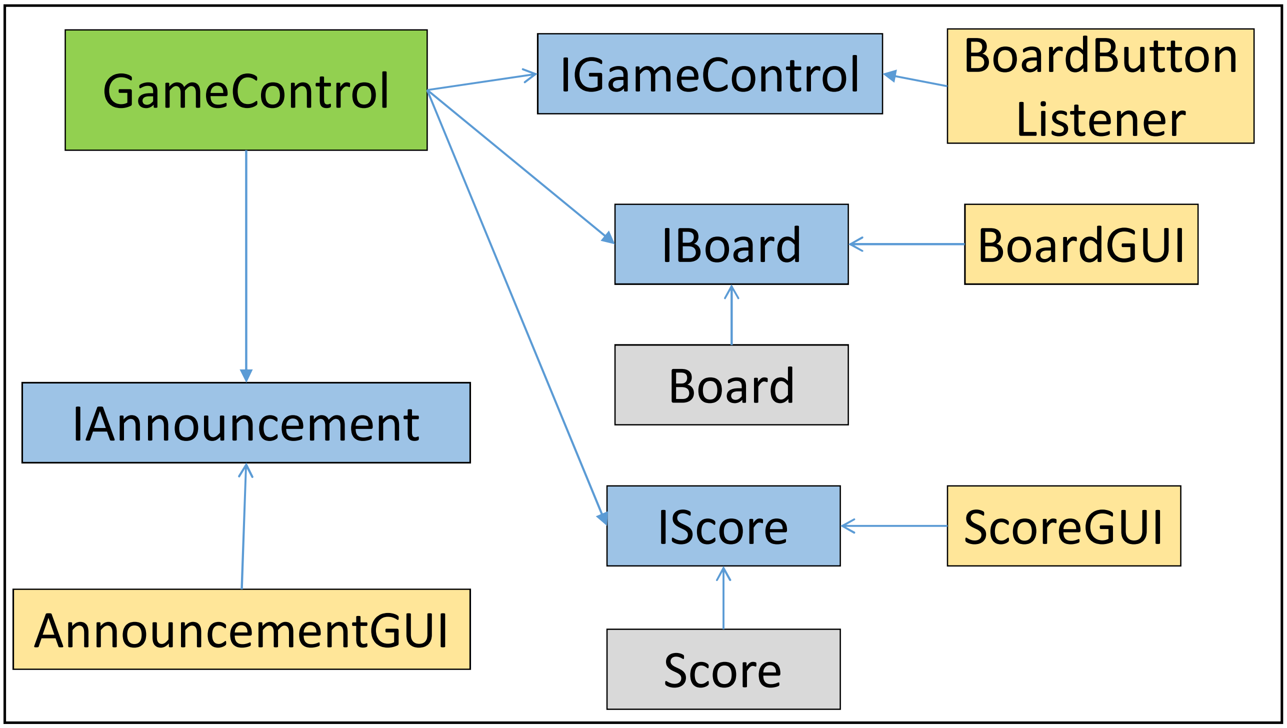
Independence via Interfaces

- Components between layers need to interact
- We should be able to change a component in one layer, without having to make changes components in another layer
- Use interfaces, to establish "contracts" between layers
- Layers communicate with each other via interfaces



What interactions are needed between: Board, BoardGUI, AnnouncementGUI, ScoreBoardGUI, GameControl, BoardButtonListener

- BoardButtonListener needs to tell GameControl about the position of the button that was clicked



GameControl

IGameControl

BoardButton Listener

IBoard

BoardGUI

Board

IAnnouncement

IScore

ScoreGUI

AnnouncementGUI

Score

Lab 11: tic_tac_toe/v2

- Added GameControl class
- Added several interfaces

TODO:

- have existing GUI classes implement the new interfaces
  - Update access modifier in interfaces to be **public**
  - ScoreBoardGUI implements IBcore
  - BoardGUI implements IBoard
  - AnnouncementGUI implements IAnnouncement
- Instantiate GameControl object in the main() method