

Checkers

Group 14:

Blake Galbavy

Austin Metz

Galen Pogoncheff

Richard Poulson

Use Case-03: Play Against User

- Primary Hosting User and Joining User connect
- Board is displayed for both users
- Primary User has first move and selects a checker piece
- Primary User chooses where to move the selected piece
- Board updates for both users
- Joining User selects move
- Game continues playing from the pattern above

The image shows a Kali Linux desktop environment with two terminal windows open. Both windows are running a Python script named 'Checkers.py'. The left window shows the initial setup where the user chooses to play against AI. The right window shows the game board visualization, which is a 10x10 grid with pieces and a score. The desktop background is black, and the taskbar at the bottom contains various application icons.

Terminal 1 (Left):

```

kali@kali: ~ — python Checkers.py — 204x127
Kali@kali-Pro: tests gallopopenriffs python Checkers.py
Welcome to Checkers!

Would you like to (1) login or (2) create a new account?
> 1
user name: test123
password: password
NAME: test123

(1) Play Against AI
(2) New Game
(3) Join Game
(4) View System Rankings
(5) View System Usage
(6) Help
(7) Quit

Enter Selection: 2
Enter address for game hosting: 127.0.0.1
Enter Port for game hosting: 8080
received connection from 127.0.0.1:54172
waiting for user to join...
waiting for user to join...
received connection from 127.0.0.1:54173

```

Terminal 2 (Right):

```

kali@kali: ~ — python Checkers.py — 204x127
Kali@kali-Pro: tests gallopopenriffs python Checkers.py
Welcome to Checkers!

Would you like to (1) login or (2) create a new account?
> 1
user name: test123
password: password
NAME: test123

(1) Play Against AI
(2) New Game
(3) Join Game
(4) View System Rankings
(5) View System Usage
(6) Help
(7) Quit

Enter Selection: 2
Enter address for game hosting: 127.0.0.1
Enter Port for game hosting: 8080
received connection from 127.0.0.1:54172
waiting for user to join...
waiting for user to join...
received connection from 127.0.0.1:54173

```

The game board visualization in both terminals is a 10x10 grid. The pieces are represented by numbers 1 through 7. The score is displayed at the bottom of the board. The board is as follows:

```

1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10

```

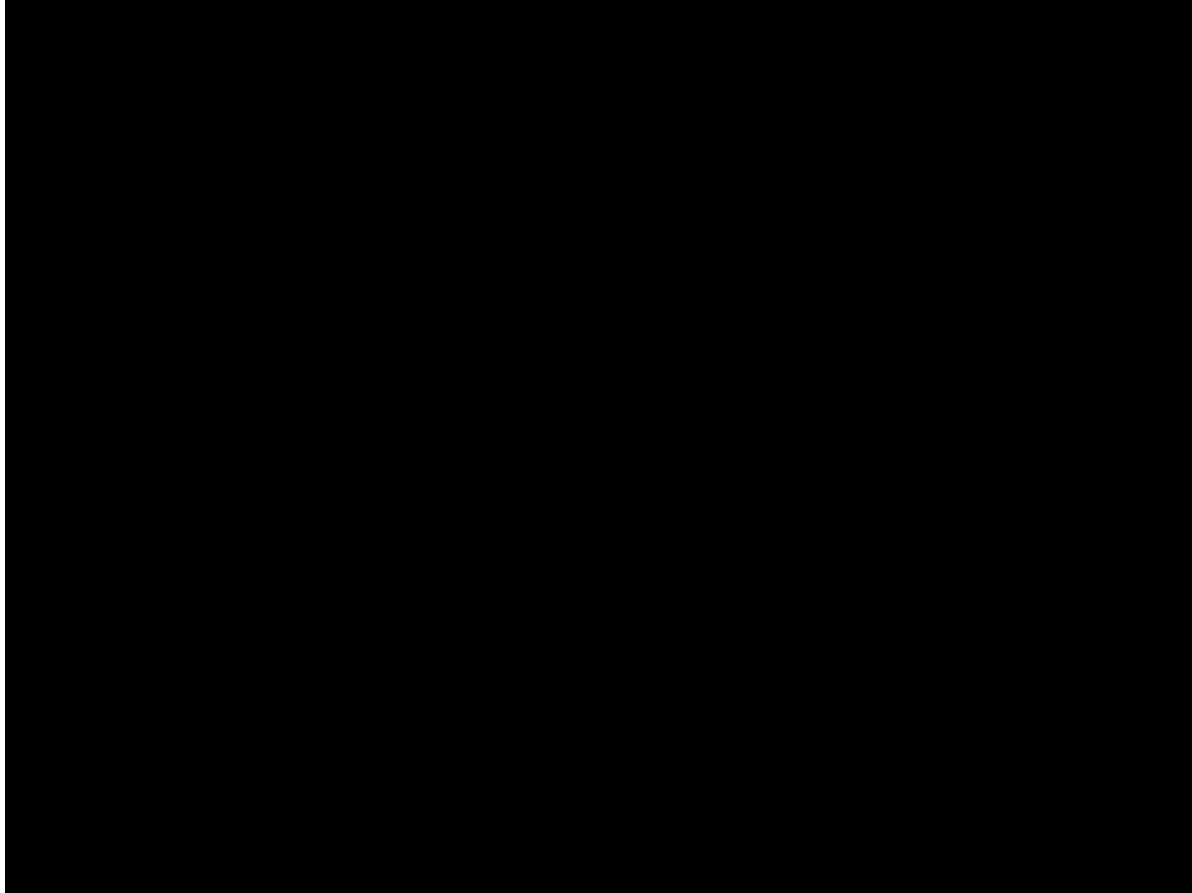
The score is displayed at the bottom of the board. The score is 0.00.

The desktop environment is Kali Linux, and the taskbar at the bottom contains various application icons.

Use Case-05: Play AI-Player

- After choosing to play again AI, a new instance of CheckersBoard is created and set as the current game.
- The computer player sets the heuristic function that it will use for its AI strategy, then initializes its strategy as well.
- Both the human and computer players are notified, the human player makes the first move, and both players continue until a winner is decided.

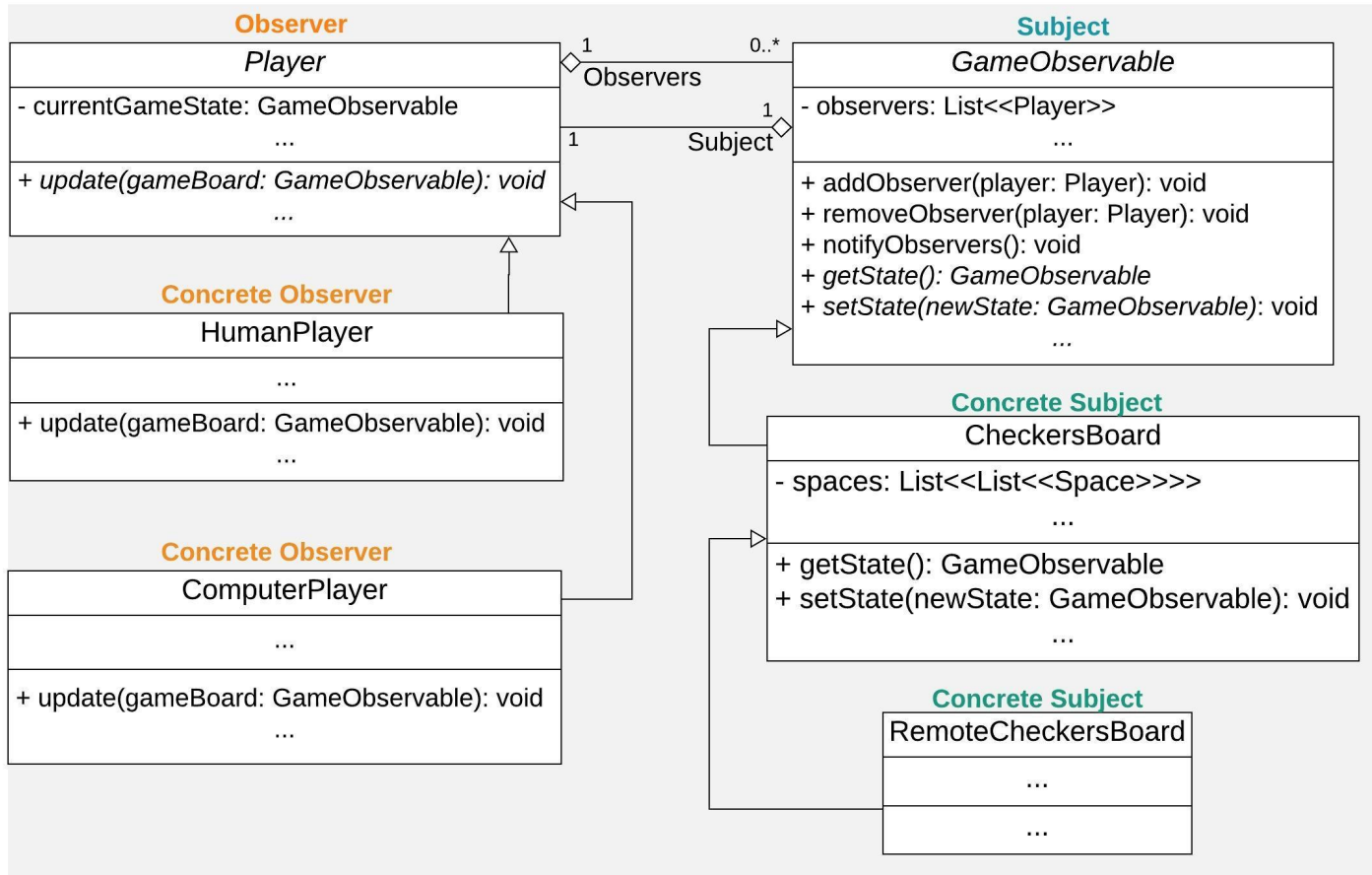
Use Case-05: Demo



Using the Observer Pattern

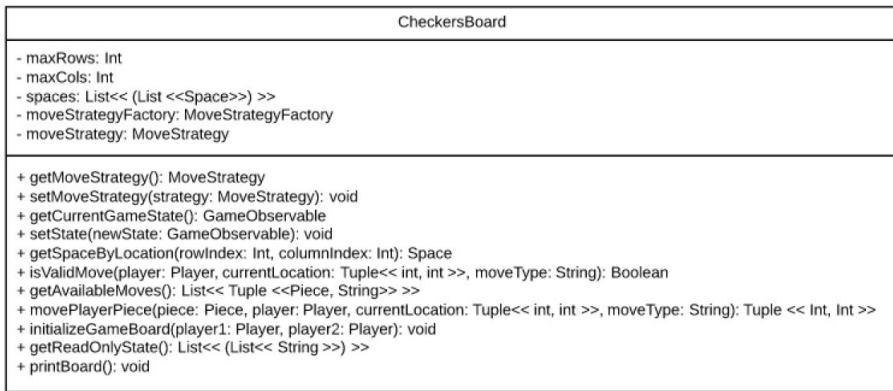
- Subject (publisher) is abstract class `GameObservable`, extended by the `CheckersBoard` and `RemoteCheckersBoard` classes. Player objects are the observers in the implementation.
- When a `CheckersBoard/RemoteCheckersBoard` is initialized, two Player objects are passed as arguments, which are attached to the board.
- After a Player makes a move, the board notifies all the attached Players, causing them to update. Updating the Player objects causes them to update their saved state of the board.
- The `ComputerPlayer` uses this updated game state to calculate possible moves in its following turn.

Using the Observer Pattern



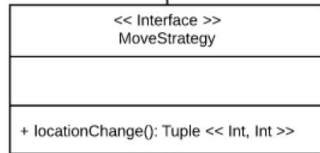
Using the Strategy Pattern

Context

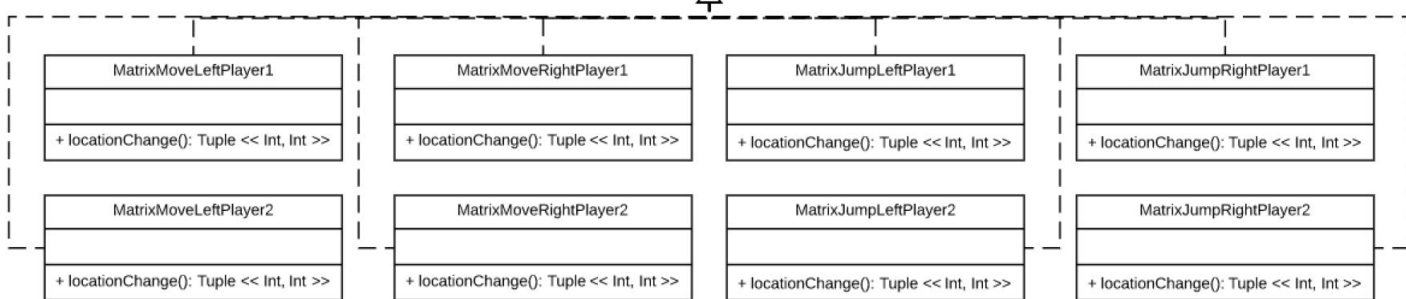


1 - strategy

Interface

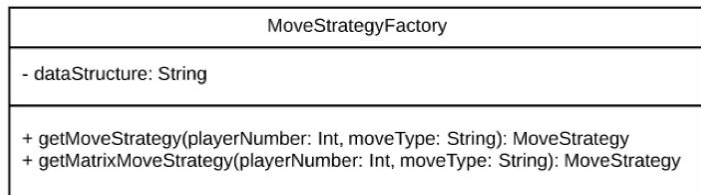


Implementations

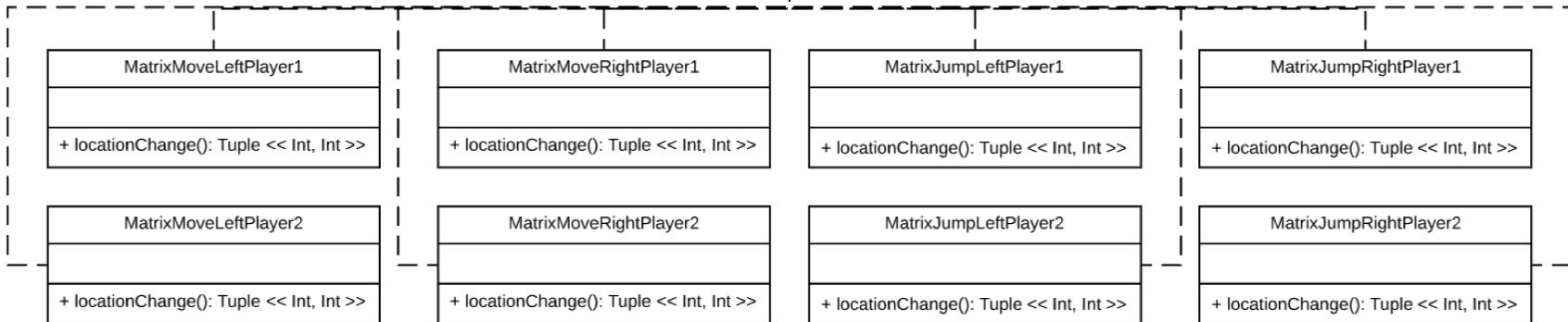
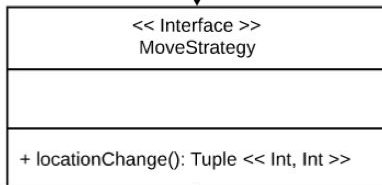


Using the Factory Pattern to create Strategy Objects

ConcreteFactory



returns new MoveStrategy



Link to Project Demo

<https://drive.google.com/open?id=1I2MmiAySGg3ZqhvDI5JgiLdmTpVHQBHR>