Programowanie w języku Java - projekt

Temat: Reversi - gra z wykorzystaniem relacji klient-serwer.
Autorzy: Bednarski Jakub, Wolszczak Weronika
Grupa: 2ID11A

1. **Wykorzystane technologie.**

   Prosta gra zaimplementowana w Javie. Interfejs graficzny powstał w oparciu o technologię Swing. Serwer działa w oparciu o Sockety na bazie protokołu TCP. Grafika została stworzona w programie Gimp - ilustracje 30x30px.

2. **Najważniejsze fragmenty kodu.**

   Sprawdzanie ruchu gracza w trybie singleplayer.

```
@Override
public void mousePressed(MouseEvent e) {
        String who = "X";
        if(turns==ReversiColor.BLACK) {
                opponent = ReversiColor.WHITE;
                who = "CZARNY";
        }
        else if(turns==ReversiColor.WHITE) {
                opponent = ReversiColor.BLACK;
                who = "BIAŁY";
        }
        boolean movePossible = false, blocked = false;
        if(canIMove(turns,opponent)){
                int x, y;
                immobilityBlocker = 0;
                x = (int) e.getX();
                y = (int) e.getY();
                if(y>=20 && x<240 && y<260) {
                        x = Math.floorDiv(x, 30);
                        y = Math.floorDiv(y-20, 30);
                        if(plansza[x][y]==ReversiColor.EMPTY){
                                boolean[] directions = new boolean[8];
                                  replace(turns,opponent,x,y,directions);           {
                                }
                                else {
                                        infoOGrze = who + " - Ten ruch nie jest możliwy!";
                                        repaint();
                                }
                        }
                        else {
                                infoOGrze = who + " - Miejsce zajęte!";
                                repaint();
                        }
                }
                else
                {
                        infoOGrze = "Kliknąłeś poza planszę";
                        repaint();
                }
        }
        else
        {
                infoOGrze = who + " - Brak możliwych ruchów!";
                blocked = true;
        }
        repaint();
        if(movePossible) {
                endTurn();
        }
```

```
        else
        {
                infoOGrze = "Kliknąłeś poza planszę";
                repaint();
        }
}
else
{
        infoOGrze = who + " - Brak możliwych ruchów!";
        blocked = true;
}
repaint();
if(movePossible) {
        endTurn();
}
if(blocked) {
        immobilityBlocker++;
        endTurn();
}
```

Sprawdzanie ruchu gracza w trybie Multiplayer.

```
@Override
public void mousePressed(MouseEvent e) {
        if(!myTurn){
                infoOGrze = "Jeszcze nie...";
                repaint();
                return;
        }
        if(canIMove(myColor,opposite())){
                int x, y;
                immobilityBlocker = 0;
                x = (int) e.getX();
                y = (int) e.getY();
                if(y>=20 && x<240 && y<260) {
                        x = Math.floorDiv(x, 30);
                        y = Math.floorDiv(y-20, 30);
                        if(plansza[x][y]==ReversiColor.EMPTY){
                                boolean[] directions = new boolean[8];
                                if(replacePossible(myColor,opposite(),x,y,directions)){
                                        plansza[x][y] = myColor;
                                        replace(myColor,opposite(),x,y,directions);
                                        infoOGrze = "Czekaj na przeciwnika...";
                                        myTurn = false;
                                        sendField();
                                }
```

```
                        }
                        else {
                                infoOGrze = "Ten ruch nie jest możliwy!";
                        }
                }
                else {
                        infoOGrze = "Miejsce zajęte!";
                }
        }
        else
        {
                infoOGrze = "Kliknąłeś poza planszę";
        }
}
else
{
        infoOGrze = "Brak możliwych ruchów!";
        myHandler.getWriter().println("Can't move");
        myTurn = false;
        immobilityBlocker++;
}
repaint();
if(Settled())
        endGame();
```

Podstawową różnicę pomiędzy sprawdzaniem ruchu gracza dla obu trybów jest konieczność czekania na ruch przeciwnika w przypadku gry przez serwer, natomiast w trybie singleplayer gracze wykonują ruchy naprzemiennie na jednej planszy. Kiedy okno gry zostanie zamknięte podczas gry multiplayer serwer wyśle informacje o tym do drugiego klienta i tryb multiplayer zostanie zakończony.

Sprawdzenie czy warunek zwycięstwa został spełniony.

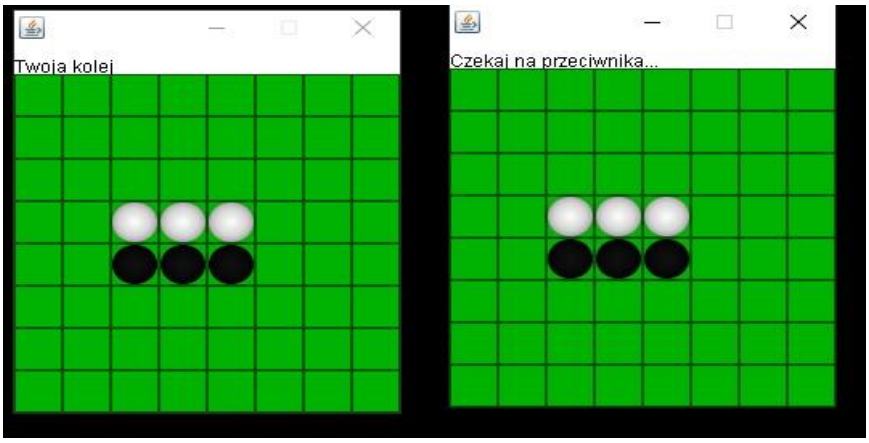
```
public boolean Settled()
{
        if(fullyFilled()||immobilityBlocker>1){
                win = checkWhoPrevailed();
                switch(win){
                        case BLACK: SingleGameOver.score="CZARNY wygrał!";
                                    SingleGameOver.blackWins++;
                                    break;
                        case WHITE: SingleGameOver.score="BIAłY wygrał!";
                                    SingleGameOver.whiteWins++;
                                    break;
                        case DRAW:      SingleGameOver.score="Mamy remis!";
                                    SingleGameOver.draws++;
                                    break;
                        case NOT_YET: return false;
                }
        }

        else{
                win = whoIsEliminated();
                switch(win){
                        case BLACK: SingleGameOver.score="BIAłY wygrał!";
                                    SingleGameOver.whiteWins++;
                                    break;
                        case WHITE: SingleGameOver.score="CZARNY wygrał!";
                                    SingleGameOver.blackWins++;
                                    break;
                        case DRAW:      SingleGameOver.score="Plansza pusta!";
                                    break;
                        case NOT_YET: return false;
                }
        }
        return true;
}
```

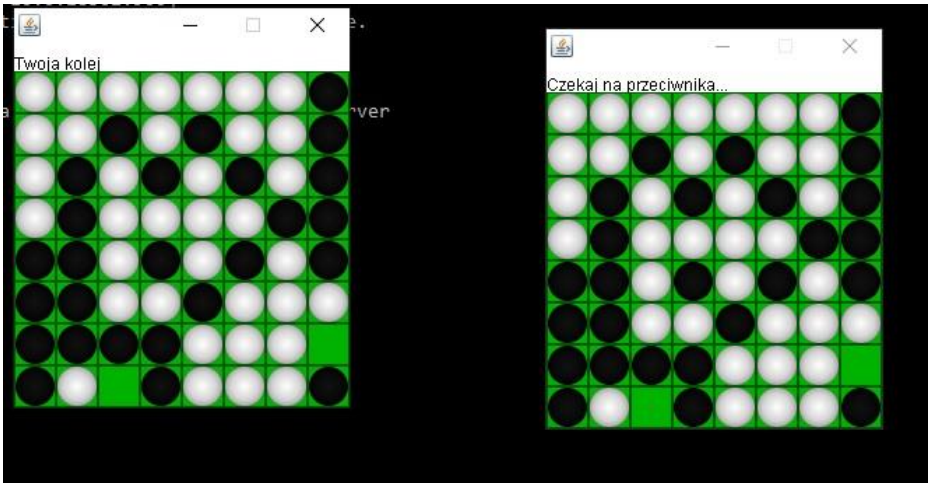
Metoda sprawdza wynik meczu, jeśli żaden z graczy nie ma więcej możliwych ruchów liczy pól którego koloru jest więcej, w przeciwnym przypadku sprawdza który gracz przejął wszystkie pola przeciwnika.

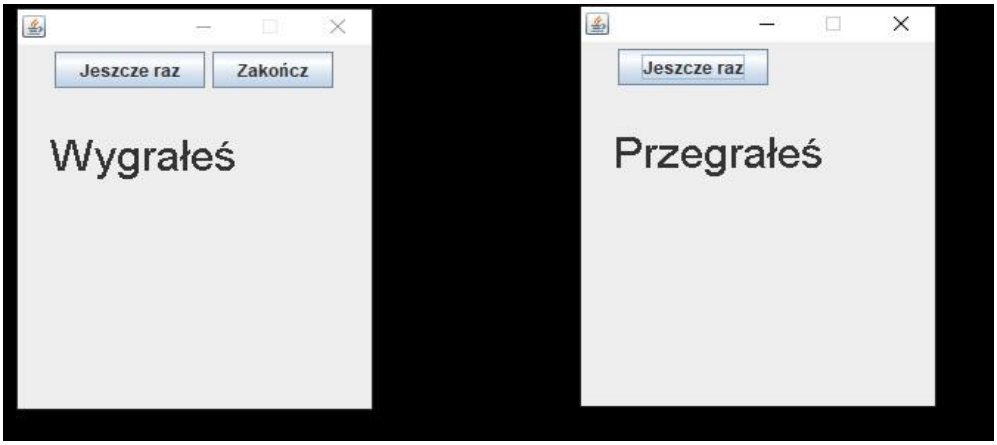
3. **Opis działania.**

Celem gry jest wypełnienie planszy większą liczbą własnych pionów niż przeciwnik. Gra kończy się, gdy jeden z graczy straci wszystkie swoje piony.

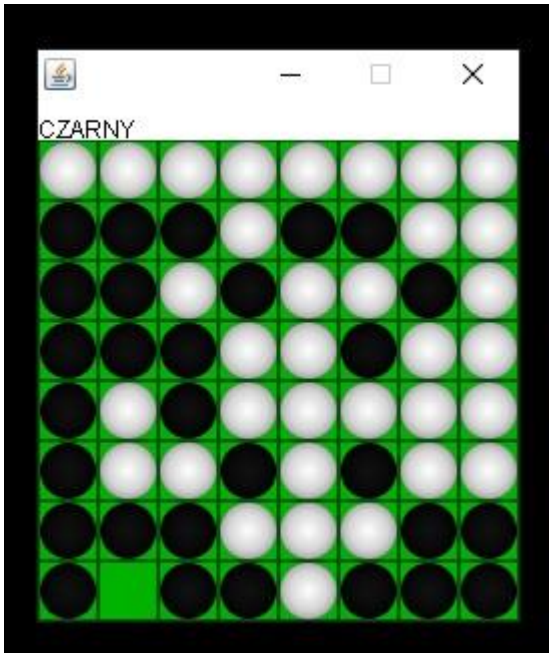


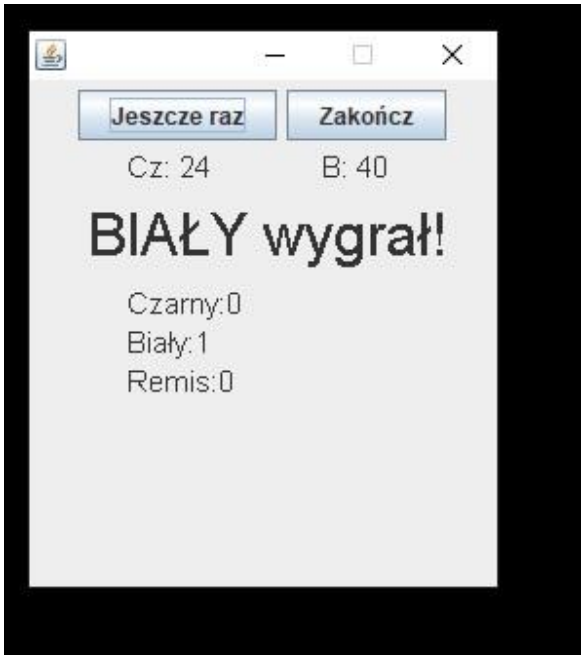
Powyższy obraz przedstawia klientów dwóch graczy grających przeciw sobie. Dalej pokazany zostaje przykładowy przebieg gry:

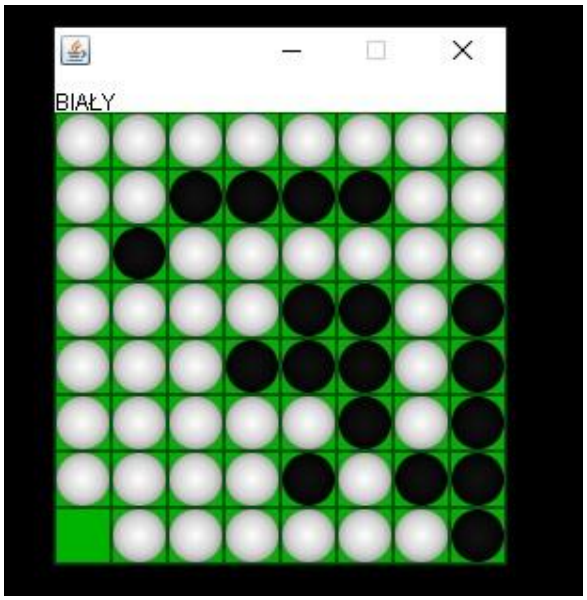


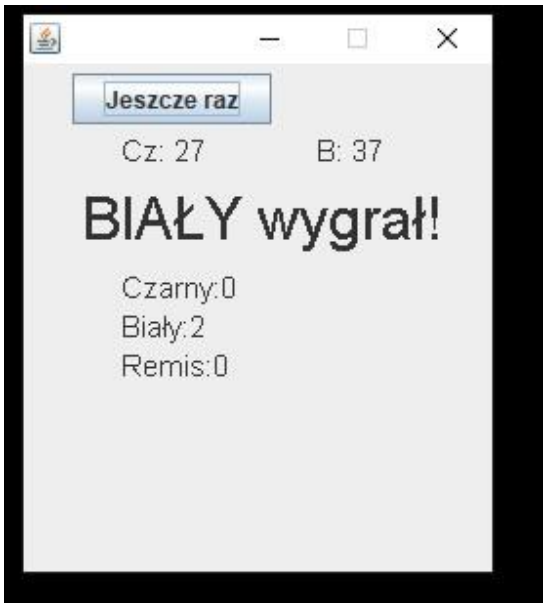


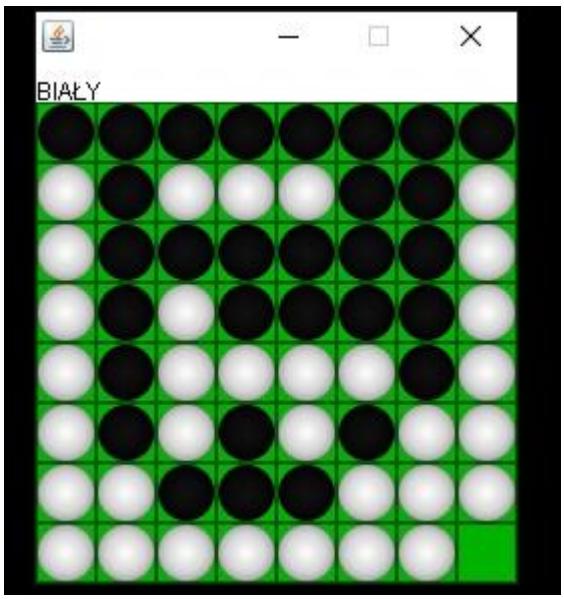
W trybie rozgrywki singleplayer gracze wykonują ruchy naprzemiennie na jednej planszy, natomiast wynik każdej kolejnej gry jest sumowany co będzie widoczne na poniższych zdjęciach z przykładowych trzech rozgrywek.











Na ekranie końcowym widać komunikat o zwycięzcy danej tury. Ponad nim są opcje zagrania ponownie lub zakończenia gry a także zsumowane pola obu graczy zajęte przez nich na koniec gry. Poniżej natomiast sumowane są zwycięstwa obu graczy oraz remisy ze wszystkich dotychczasowych tur.