

Kod studenta: Marcin Hanas 29345

Inspekcję zrealizował: Rafał Szczepanik 293471

1)

```
private void ballClicked(ImageView ball) {
    if (boardModel.getState() == GAME_ENDED)
        return;
    if (ballPos == null)
        return;
    int[] ballPos = boardView.getBallPos(ball);
    int x = ballPos[0];
    int y = ballPos[1];

    if (boardModel.getState() == CHOOSE_FIRST_BALL) {
        boardView.setIsChosen(x, y, true);
        boardModel.chooseFirstBall(x, y);
    } else if (boardModel.getState() == CHOOSE_SECOND_BALL) {
        Step lineStep = boardModel.chooseBallLine(x, y);
        if (lineStep != null) {

            int myLineLength = boardModel.getLineLength();
            int[] firstBallPos = boardModel.getBallPos(0);
            int[] lastBallPos = boardModel.getBallPos(myLineLength - 1);

            boardView.setChosenLine(firstBallPos[0], firstBallPos[1], lineStep.dx(), lineStep.dy(), myLineLength);

            List<Step> listOfMoves = boardModel.createListOfMoves();
            if (listOfMoves.size() == 0) {
                boardModel.endTurn();
                boardView.setNotChosenLine(firstBallPos[0], firstBallPos[1], lineStep.dx(), lineStep.dy(),
myLineLength);
                boardModel.startTurn();
                Alerts.makeTurnAlert(MY_COLOR);
            }
            for (Step move : listOfMoves) {
                ImageView moveButton = boardView.addMoveButton(lastBallPos[0], lastBallPos[1],
                    new ImageMove(move.dx(), move.dy()));
                moveButton.setOnMouseClicked(e -> moveButtonClicked(move, lineStep));
            }
        }
    }
}
```

Powyższy kod nie jest przejrzysty. Funkcja realizuje wiele operacji pod skompilowanymi warunkami.

1) dodać komentarze objaśniające kod

2) stworzyć funkcje realizujące fragmenty powyższego kodu (stworzono dwie przykładowe)

3) uprościć zapis warunków

```

private void ballClicked(ImageView ball) {

    //jeśli gra została zakończona
    if (boardModel.getState() == GAME_ENDED)
        return;

    //wybór pola bez kulki
    if (ballPos == null)
        return;

    int[] ballPos = boardView.getBallPos(ball);
    int x = ballPos[0];
    int y = ballPos[1];

    //wybór pierwszej kulki
    if (boardModel.getState() == CHOOSE_FIRST_BALL) {
        firstBallClicked(x,y);
    }

    //wybór drugiej kulki
    if (boardModel.getState() == CHOOSE_SECOND_BALL) {
        secondBallClicked(x,y);
    }

}

//funkcja wykonywana przy wybraniu pierwszej kulki
private static void firstBallClicked(int x, int y){
    boardView.setIsChosen(x, y, true);
    boardModel.chooseFirstBall(x, y);
}

//funkcja wykonywana przy wybraniu drugiej kulki
private static void secondBallClicked(int x, int y){
    Step lineStep = boardModel.chooseBallLine(x, y);
    if (lineStep != null) {

        int myLineLength = boardModel.getLineLength();
        int[] firstBallPos = boardModel.getBallPos(0);
        int[] lastBallPos = boardModel.getBallPos(myLineLength - 1);

        boardView.setChosenLine(firstBallPos[0], firstBallPos[1], lineStep.dx(), lineStep.dy(), myLineLength);

        List<Step> listOfMoves = boardModel.createListOfMoves();
        if (listOfMoves.size() == 0) {
            boardModel.endTurn();
            boardView.setNotChosenLine(firstBallPos[0], firstBallPos[1], lineStep.dx(), lineStep.dy(),
myLineLength);
            boardModel.startTurn();
        }
    }
}

```

```

        Alerts.makeTurnAlert(MY_COLOR);
    }
    for (Step move : listOfMoves) {
        ImageView moveButton = boardView.addMoveButton(lastBallPos[0], lastBallPos[1],
            new ImageMove(move.dx(), move.dy()));
        moveButton.setOnMouseClicked(e -> moveButtonClicked(move, lineStep));
    }
}

```

2)

```

public class ImageMove {
    private int dx;
    private int dy;
    private int rotation;

    public ImageMove(int dx, int dy) {
        this.dx = dx;
        this.dy = dy;
        this.rotation = getRotation(dx, dy);
    }

    private static int getRotation(int dx2, int dy2) {
        if (dx2==2) return 0;
        else if (dx2==1 && dy2==1) return 60;
        else if (dx2==1 && dy2==0) return 90;
        else if (dx2==0 && dy2==1) return 120;
        else if (dx2==0 && dy2==0) return 180;
        else if (dx2==1 && dy2==1) return 240;
        else return 300;
    }

    public int getDx() {
        return dx;
    }
}

```

Nazwy zmiennych w powyższym kodzie nie są samokomentujące się. Użyte w dwóch miejscach zmienne oznaczają to samo a mają inne nazwy.

- 1) dodać komentarze
- 2) ujednolicić nazwy zmiennych
- 3) nadać zmiennym nazwy samokomentujące

```

public class ImageMove {
    //przesunięcie wzdłuż osi X
    private int xShift;
    //przesunięcie wzdłuż osi Y
    private int yShift;
    private int rotation;
}

```

```

public ImageMove(int xShift, int yShift) {
    this.xShift = xShift;
    this.yShift = yShift;
    this.rotation = getRotation(xShift, yShift);
}

private static int getRotation(int xShift, int yShift) {
    if (xShift==2) return 0;
    else if (xShift==1 && yShift==1) return 60;
    else if (xShift==-1 && yShift==1) return 120;
    else if (xShift==-2) return 180;
    else if (xShift==-1 && yShift==-1) return 240;
    else return 300;
}

public int getXShift() {
    return xShift;
}

```

3)

```

public class Main extends Application {

    public static void main(String[] args) {

        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws Exception {
        try {
            Parent root = FXMLLoader.load(getClass().getResource("window.fxml"));
            Scene scene = new Scene(root, 600, 600);
            primaryStage.setScene(scene);
            primaryStage.setTitle("Gracz "+MY_COLOR);
            primaryStage.show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Parametry funkcji przekazywane "na sztywno w kodzie". Niepotrzebne "ENTRY".

1) dodać zmienne i użyć ich jako parametrów wywołania funkcji

2) usunąć niepotrzebne "ENTERY"

```
public class Main extends Application {

    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws Exception {
        int width = 600;
        int height = 600;
        String stageTitle = "Gracz "+MY_COLOR;

        try {
            Parent root = FXMLLoader.load(getClass().getResource("window.fxml"));
            Scene scene = new Scene(root, width, height);
            primaryStage.setScene(scene);
            primaryStage.setTitle(stageTitle);
            primaryStage.show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```