Háskóli Íslands

Viðmótsforritun (HBV201G)

# Skilaverkefni 6

Tafl

*Höfundur:*

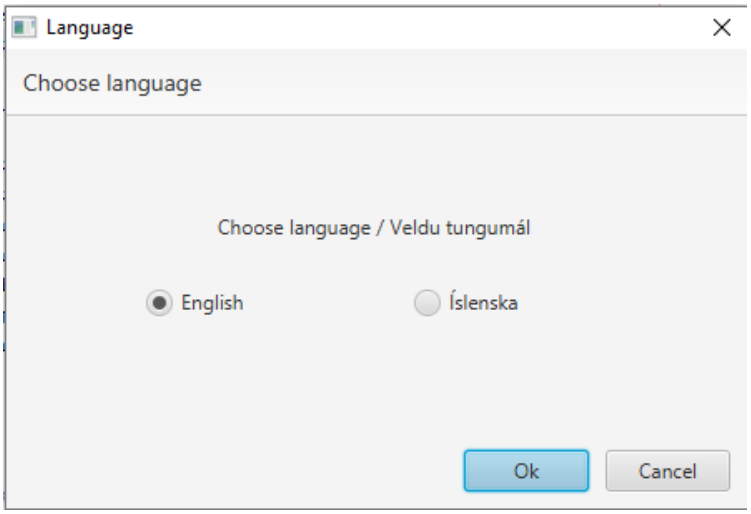
*Daníel Þór Guðmundsson* – dthg7@hi.is

*Kennari:*

*Ebba Þóra Hvannberg*

15. apríl 2018

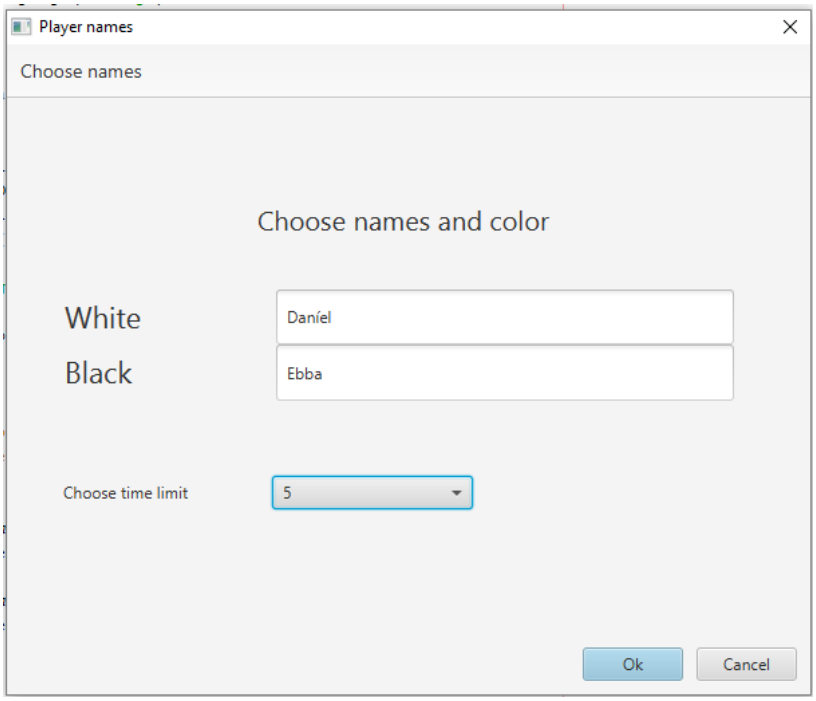
## Skjölun




Þegar forritið er keyrt er byrjað á því að spyrja um tungumál. Hægt er að velja milli ensks og íslensks viðmóts. Allir strengir eru síðan geymdir í properties skrám.

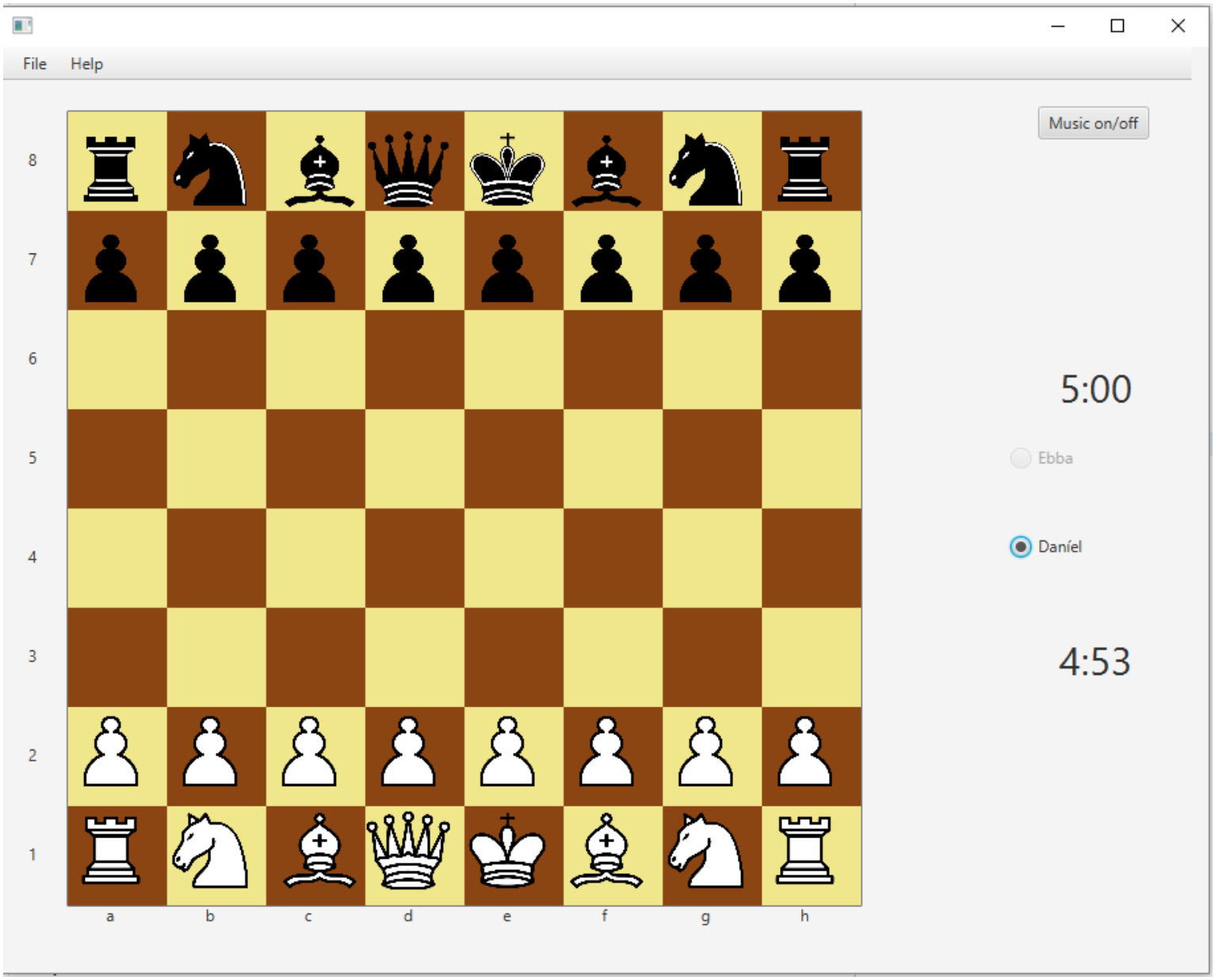
```
76      @FXML
77      public void languageDial(){
78          DialogPane p = new DialogPane();
79          languageDialog.setVisible(true);
80          english.setSelected(true);
81
82          p.setContent(languageDialog);
83
84          Dialog d = new Dialog();
85
86          d.setDialogPane(p);
87
88          d.setHeaderText("Choose language");
89          d.setTitle("Language");
90
91          ButtonType ok = new ButtonType("Ok",
92                  ButtonBar.ButtonData.OK_DONE);
93          d.getDialogPane().getButtonTypes().add(ok);
94          ButtonType cancel = new ButtonType("Cancel",
95                  ButtonBar.ButtonData.CANCEL_CLOSE);
96          d.getDialogPane().getButtonTypes().add(cancel);
97
98          d.showAndWait();
99      }
```

```
58      @FXML
59      private void languageHandler(ActionEvent event) {
60          RadioButton b=(RadioButton)event.getSource();
61          if(b.getText().equals("Íslenska"))
62          {
63              strings=ResourceBundle.getBundle("is.hi.view.Language.text", new Locale("is"));
64          }
65          else
66          {
67              strings=ResourceBundle.getBundle("is.hi.view.Language.text", new Locale("en","GB"));
68          }
69
70          mainController.setLanguage(strings);
71
72      }
```




Nú valdi ég enskt viðmót svo allir textar eru á ensku. Núna eru leikmenn beðnir um að velja sér nöfn, lit og tímamörk á leik.

```
68      public Players ChessDialog(){
69          DialogPane p = new DialogPane();
70          chessDialog.setVisible(true);
71
72          p.setContent(chessDialog);
73
74          Dialog d = new Dialog();
75
76          d.setDialogPane(p);
77
78              d.setHeaderText(strings.getString("dialogHeader"));
79              d.setTitle(strings.getString("dialogID"));
80              wLabel.setText(strings.getString("dialogWhite"));
81              bLabel.setText(strings.getString("dialogBlack"));
82              namesLabel.setText(strings.getString("dialogNameColor"));
83              timeLabel.setText(strings.getString("dialogTime"));
84
85          ButtonType ok = new ButtonType(strings.getString("dialogOK"),
86                  ButtonBar.ButtonData.OK_DONE);
87          d.getDialogPane().getButtonTypes().add(ok);
88          ButtonType cancel = new ButtonType(strings.getString("dialogCancel"),
89                  ButtonBar.ButtonData.CANCEL_CLOSE);
90          d.getDialogPane().getButtonTypes().add(cancel);
91
92          final Node confirmation = p.lookupButton(ok);
93          confirmation.disableProperty()
94                  .bind(whiteText.textProperty().isEmpty()
95                          .or(blackText.textProperty().isEmpty()));
96
97
98  Optional<ButtonType> outcome = d.showAndWait();
99  if (outcome.isPresent() && (outcome.get()
100 .getButtonData() == ButtonBar.ButtonData.OK_DONE)) {
101 return new Players(whiteText.getText(),
102 blackText.getText());
103
104     }
105     return null;
106 }
```




Nú hefst leikurinn. Eins og venjulega byrjar hvítur.



Nú reynir Daníel að draga peðið sitt út fyrir borðið. Það á ekki að vera hægt. Peðið er því sent til baka á þann reit sem það var á og villuskilaboð sett á skjáinn.

Í ChessTable.java

```
258         if (x < 50 || x > 650 || y < 50 || y > 650) {
259             mainController.displayErrorMessage(strings.getString("outsideTable"));
260             putPieceBack(piece);
261         }
```

```
341     public void putPieceBack(Piece piece){
342         piece.getPieceView().setX(piece.getOldX());
343         piece.getPieceView().setY(piece.getOldY());
344     }
```

Það sama mun gerast ef peð er fært á reit sem er með peð af sama lit eða ef færslan reynist ólögleg.

```
281                     //Piece of same color on spot
282                     else if (tiles[i][j].getPiece().getColor() == piece.getColor()) {
283
284                         mainController.displayErrorMessage(strings.getString("occupied"));
285
286                         putPieceBack(piece);
287                     }
```

```
308                     //The move turned out to be illegal
309                     else {
310
311                         System.out.println(strings.getString("illegalMove"));
312                         putPieceBack(piece);
313                     }
```



Nú ætlar Ebba að drepa peð. Þá hverfur það af borðinu.

Í ChessTable.java

```
287                     }
288                     //Kill another piece
289                     else {
290
291                         System.out.println(strings.getString("legalMove"));
292                         Piece p = tiles[i][j].getPiece();
293                         if (p.getType().equals("whiteKing")) {
294                             mainController.gameOver(2);
295                         }
296                         if (p.getType().equals("blackKing")) {
297                             mainController.gameOver(1);
298                         }
299
300                         this.getChildren().remove(tiles[i][j].getPiece().getPieceView());
301                         tiles[i][j].setPiece(piece);
302                         tiles[piece.getTileX()][piece.getTileY()].setAvailable(0);
303                         setPieceOnTile(piece, i, j);
304
305                         mainController.nextRound();
306                     }
307                 }
```

Aðeins er hægt að hreyfa peð þess aðila sem á að gera. Hin peðin eru „fryst“

(Í Chess.java)

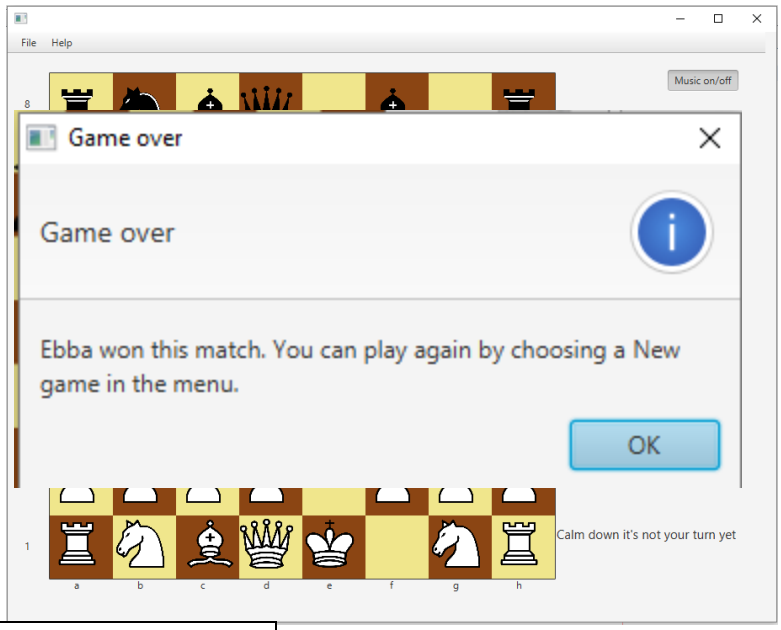
```
55  public void disableBlack() {
56      Tile [][] tiles=chessTable.getTiles();
57      for(int i=0;i<8;i++)
58      {
59          for(int j=0;j<8;j++)
60          {
61              Piece piece=tiles[i][j].getPiece();
62              if(piece!=null)
63              {
64              String type=piece.getType();
65              String isBlack=type.substring(0,5);
66              if(isBlack.equals("black"))
67              {
68                  piece.setYourTurn(false);
69              }
70              }
71          }
72      }
73  }
74
75  public void disableWhite() {
76      Tile [][] tiles=chessTable.getTiles();
77      for(int i=0;i<8;i++)
78      {
79          for(int j=0;j<8;j++)
80          {
81              Piece piece=tiles[i][j].getPiece();
82              if(piece!=null)
83              {
84              String type=piece.getType();
85              String isWhite=type.substring(0,5);
86              if(isWhite.equals("white"))
87              {
88                  piece.setYourTurn(false);
89              }
90              }
91          }
92      }
93
94  }
```

```
100 public void enableWhite() {
101     Tile [][] tiles=chessTable.getTiles();
102     for(int i=0;i<8;i++)
103     {
104         for(int j=0;j<8;j++)
105         {
106             Piece piece=tiles[i][j].getPiece();
107             if(piece!=null)
108             {
109             String type=piece.getType();
110             String isWhite=type.substring(0,5);
111             if(isWhite.equals("white"))
112             {
113                 piece.setYourTurn(true);
114             }
115             }
116         }
117     }
118 }
119
120 public void enableBlack() {
121     Tile [][] tiles=chessTable.getTiles();
122     for(int i=0;i<8;i++)
123     {
124         for(int j=0;j<8;j++)
125         {
126             Piece piece=tiles[i][j].getPiece();
127             if(piece!=null)
128             {
129             String type=piece.getType();
130             String isBlack=type.substring(0,5);
131             if(isBlack.equals("black"))
132             {
133                 piece.setYourTurn(true);
134             }
135             }
136         }
137     }
138 }
139
```

Nú vill svo til að báðir  aðilar eru orðnir mjög þreyttir á tónlistinni. Þeir geta þá slökkt á henni með því að ýta á takkan uppi til hægri.

(Í MainController)

```
240     @FXML
241     private void musicHandler(ActionEvent event) {
242         if(!pause)
243         {
244             mediaPlayer.pause();
245             pause=true;
246         }
247         else
248         {
249             mediaPlayer.play();
250             pause=false;
251         }
252 
253     }
```


Nú er kóngur Ebbu drepinn svo leiknum lýkur með sigri Daníels

```
287                         }
288                         //Kill another piece
289                         else {
290
291                             System.out.println(strings.getString("legalMove"));
292                             Piece p = tiles[i][j].getPiece();
293                             if (p.getType().equals("whiteKing")) {
294                                 mainController.gameOver(2);
295                             }
296                             if (p.getType().equals("blackKing")) {
297                                 mainController.gameOver(1);
298                             }
299
300                             this.getChildren().remove(tiles[i][j].getPiece().getPieceView());
301                             tiles[i][j].setPiece(piece);
302                             tiles[piece.getTileX()][piece.getTileY()].setAvailable(0);
303                             setPieceOnTile(piece, i, j);
304
305                             mainController.nextRound();
306                         }
307                     }
```

```
200     void gameOver(int i) {
201         String winner;
202         if(i==1)
203         {
204             winner=blackRadio.getText();
205
206         }
207         else
208         {
209             winner=whiteRadio.getText();
210         }
211         chess.disableBlack();
212         chess.disableWhite();
213         whiteRadio.setDisable(true);
214         blackRadio.setDisable(true);
215         Alert gameOver = new Alert
216         (Alert.AlertType.INFORMATION);
217
218         gameOver.setTitle(strings.getString("alertID"));
219         gameOver.setHeaderText(strings.getString("alertHeader"));
220         gameOver.setContentText(winner +" " + strings.getString("alertText"));
221
222         gameOver.setOnHidden(evt -> gameOver.close());
223         gameOver.show();
224         //gameOver.showAndWait();
225     }
```

Leiknum lýkur einnig ef annar aðilinn rennur út á tíma.

(Í White- eða BlackClockController)

```
38      @Override
39      public void initialize(URL url, ResourceBundle rb) {
40          int t = 1; // Time between actions
41          int howManyTimes = 100000000; //Number of actions
42          clock = new Timeline(new KeyFrame(Duration.seconds(t), new EventHandler<ActionEvent>() {
43
44              @Override
45              public void handle(ActionEvent event) {
46
47                  //Timeline clock=getClock();
48                  time--;
49                  chessClock.setText(time/60+":"+time%60);
50                  if(time==0)
51                  {
52                      clock.pause();
53                      mainController.gameOver(1);
54                  }
55
56                  i++;
57              }
58          }));
59
60          clock.setCycleCount(howManyTimes);
61
62      }
```