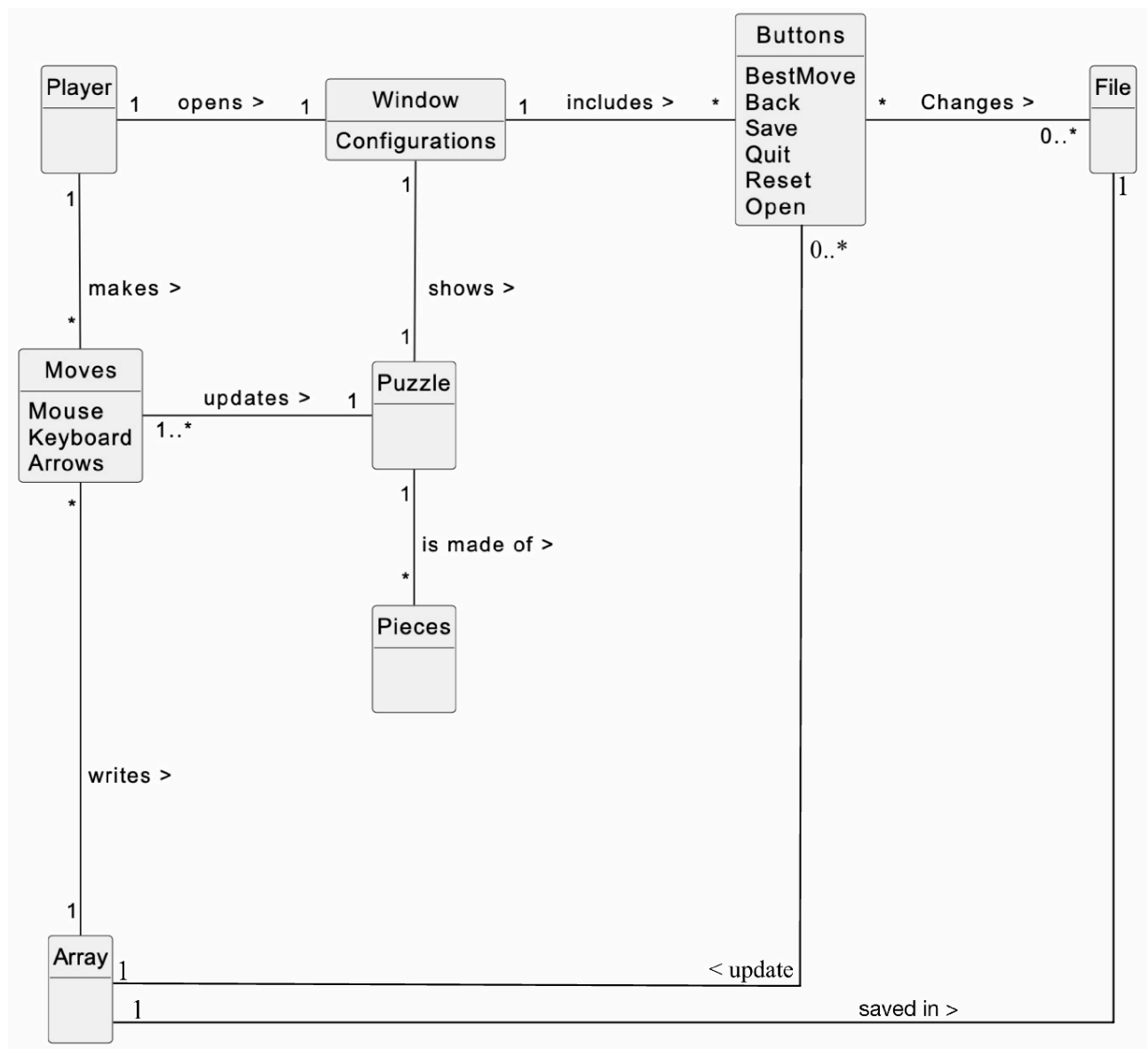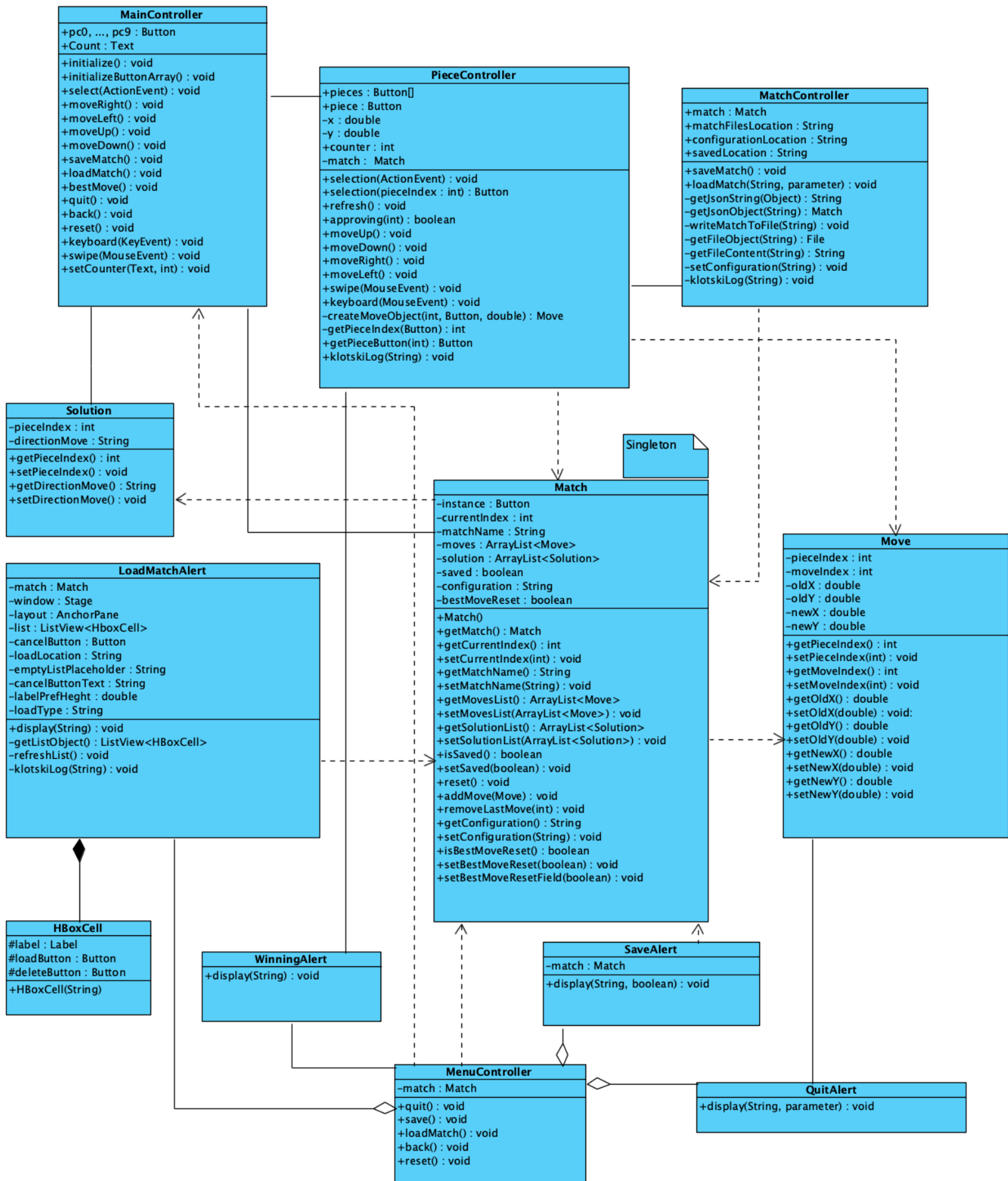# Domain Model:
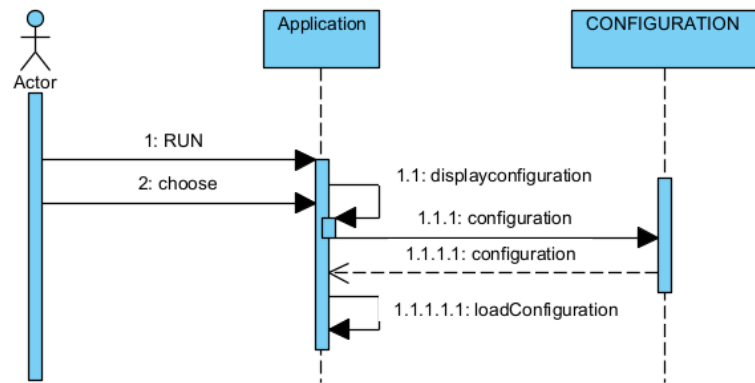

Basic Flow for Klotski Game:

1. **Player** opens game **window** and chooses a starting configuration from the three available.
2. Program updates the **puzzle** with the chosen configuration.
3. Player **moves** the pieces on the board using mouse, keyboard or arrow buttons which updates the puzzle.
4. The window also includes **buttons** that can update the array.
5. The moves are saved in an **array** that is loaded into the **file** when needed.
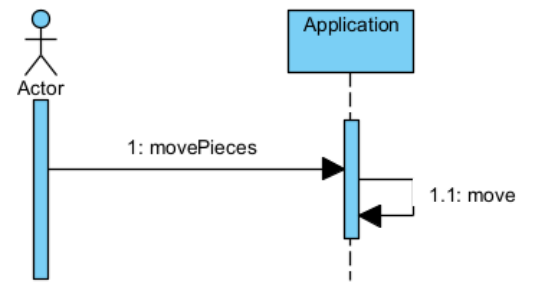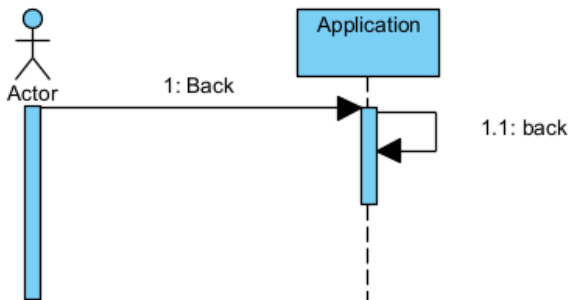
Design class model:

MainController
```
+pc0, ..., pc9 : Button
+Count : Text
```
```
+initialize() : void
+initializeButtonArray() : void
+select(ActionEvent) : void
+moveRight() : void
+moveLeft() : void
+moveUp() : void
+moveDown() : void
+saveMatch() : void
+loadMatch() : void
+bestMove() : void
+quit() : void
+back() : void
+reset() : void
+keyboard(KeyEvent) : void
+swipe(MouseEvent) : void
+setCounter(Text, int) : void
```

PieceController
```
+pieces : Button[]
+piece : Button
-x : double
-y : double
+counter : int
-match : Match
```
```
+selection(ActionEvent) : void
+selection(pieceIndex : int) : Button
+refresh() : void
+approving(int) : boolean
+moveUp() : void
+moveDown() : void
+moveRight() : void
+moveLeft() : void
+swipe(MouseEvent) : void
+keyboard(MouseEvent) : void
-createMoveObject(int, Button, double) : Move
-getPieceIndex(Button) : int
+getPieceButton(int) : Button
+klotskiLog(String) : void
```

MatchController
```
+match : Match
+matchFilesLocation : String
+configurationLocation : String
+savedLocation : String
```
```
+saveMatch() : void
+loadMatch(String, parameter) : void
-getJsonString(Object) : String
-getJsonObject(String) : Match
-writeMatchToFile(String) : void
-getFileObject(String) : File
-getFileContent(String) : String
-setConfiguration(String) : void
-klotskiLog(String) : void
```

Solution
```
-pieceIndex : int
-directionMove : String
```
```
+getPieceIndex() : int
+setPieceIndex() : void
+getDirectionMove() : String
+setDirectionMove() : void
```

LoadMatchAlert
```
-match : Match
-window : Stage
-layout : AnchorPane
-list : ListView<HBoxCell>
-cancelButton : Button
-loadLocation : String
-emptyListPlaceholder : String
-cancelButtonText : String
-labelPrefHeght : double
-loadType : String
```
```
+display(String) : void
-getListObject() : ListView<HBoxCell>
-refreshList() : void
-klotskiLog(String) : void
```

Singleton

Match
```
-instance : Button
-currentIndex : int
-matchName : String
-moves : ArrayList<Move>
-solution : ArrayList<Solution>
-saved : boolean
-configuration : String
-bestMoveReset : boolean
```
```
+Match()
+getMatch() : Match
+getCurrentIndex() : int
+setCurrentIndex(int) : void
+getMatchName() : String
+setMatchName(String) : void
+getMovesList() : ArrayList<Move>
+setMovesList(ArrayList<Move>) : void
+getSolutionList() : ArrayList<Solution>
+setSolutionList(ArrayList<Solution>) : void
+isSaved() : boolean
+setSaved(boolean) : void
+reset() : void
+addMove(Move) : void
+removeLastMove(int) : void
+getConfiguration() : String
+setConfiguration(String) : void
+isBestMoveReset() : boolean
+setBestMoveReset(boolean) : void
+setBestMoveResetField(boolean) : void
```

Move
```
-pieceIndex : int
-moveIndex : int
-oldX : double
-oldY : double
-newX : double
-newY : double
```
```
+getPieceIndex() : int
+setPieceIndex(int) : void
+getMoveIndex() : int
+setMoveIndex(int) : void
+getOldX() : double
+setOldX(double) : void:
+getOldY() : double
+setOldY(double) : void
+getNewX() : double
+setNewX(double) : void
+getNewY() : double
+setNewY(double) : void
```

HBoxCell
```
#label : Label
#loadButton : Button
#deleteButton : Button
```
```
+HBoxCell(String)
```

WinningAlert
```
+display(String) : void
```

SaveAlert
```
-match : Match
```
```
+display(String, boolean) : void
```

MenuController
```
-match : Match
```
```
+quit() : void
+save() : void
+loadMatch() : void
+back() : void
+reset() : void
```

QuitAlert
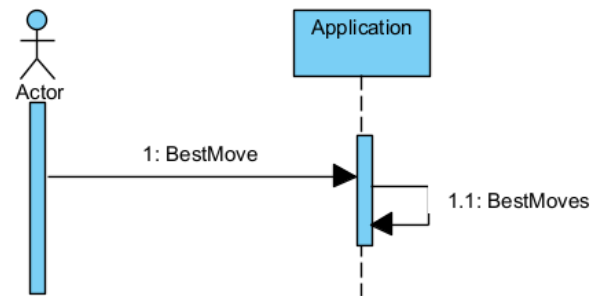```
+display(String, parameter) : void
```
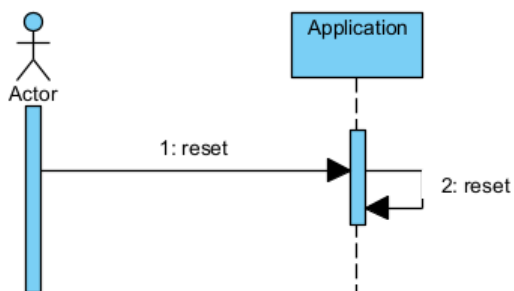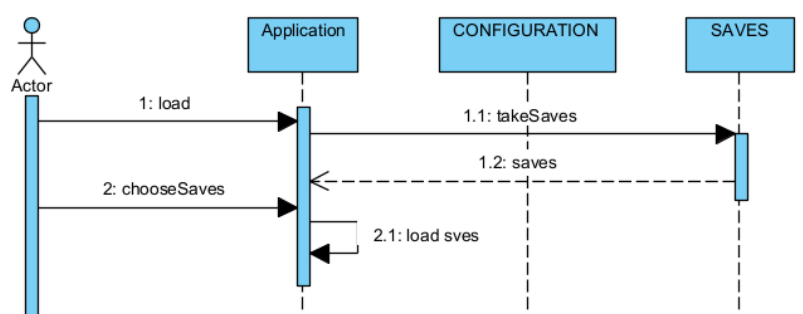
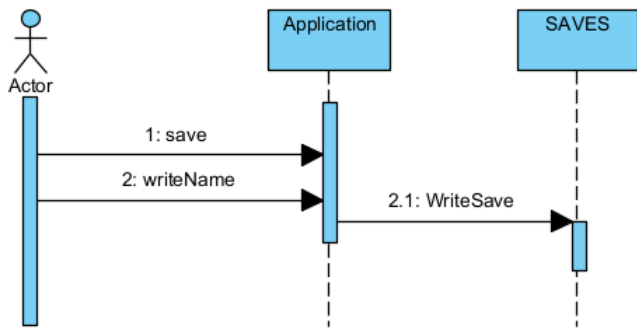# System sequence diagram:
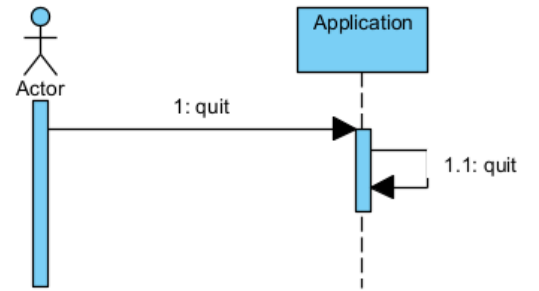


1. Start



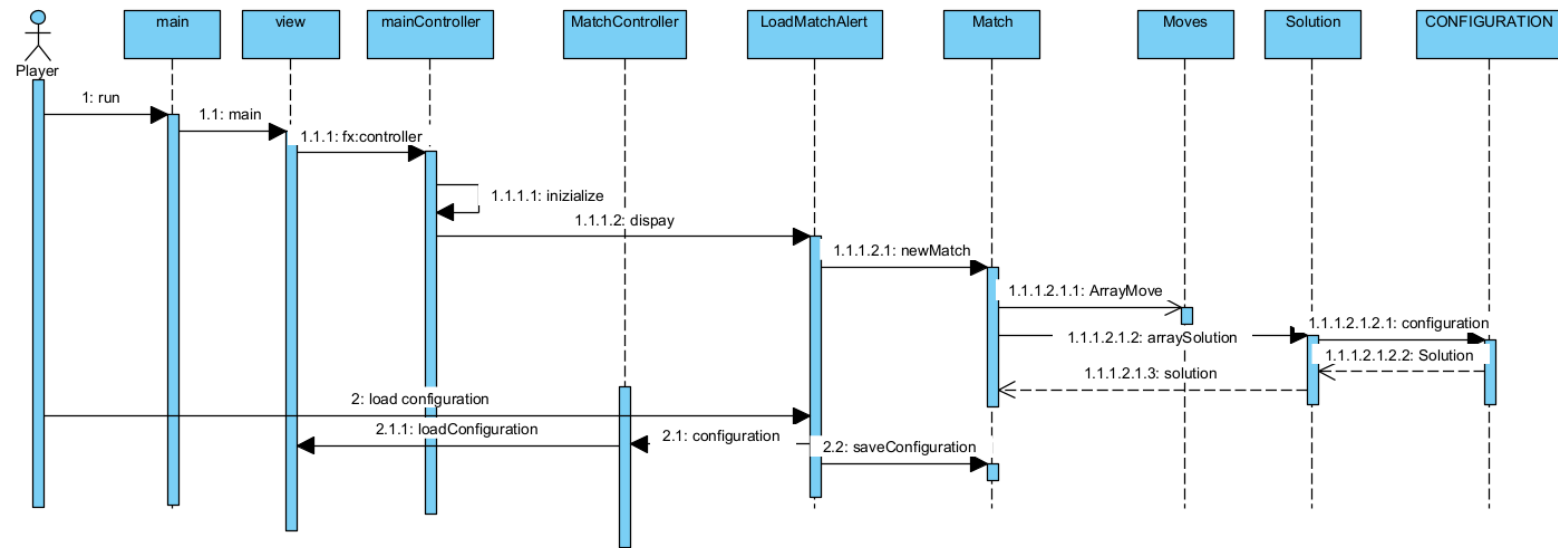2. Move


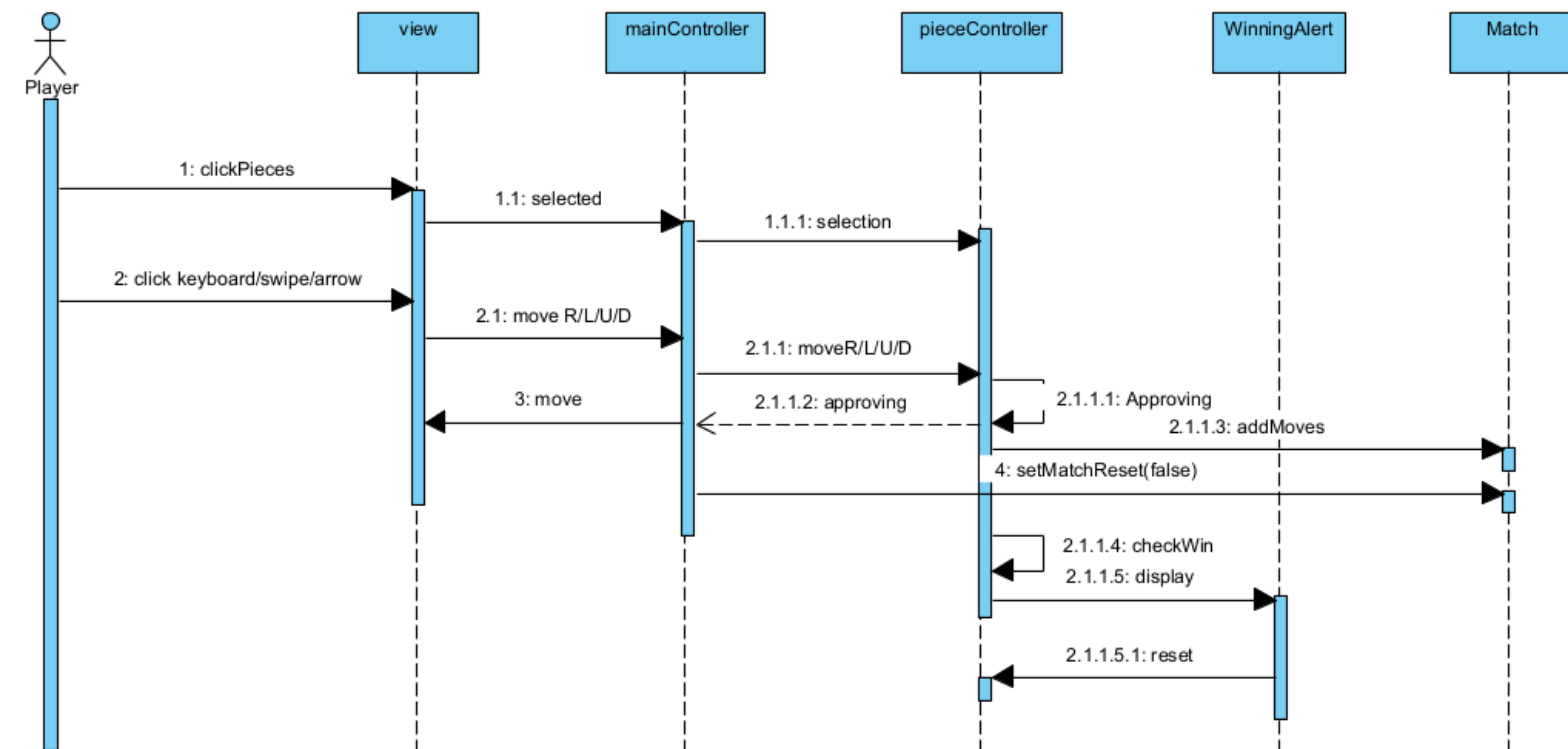
3. Back



4. Best Move



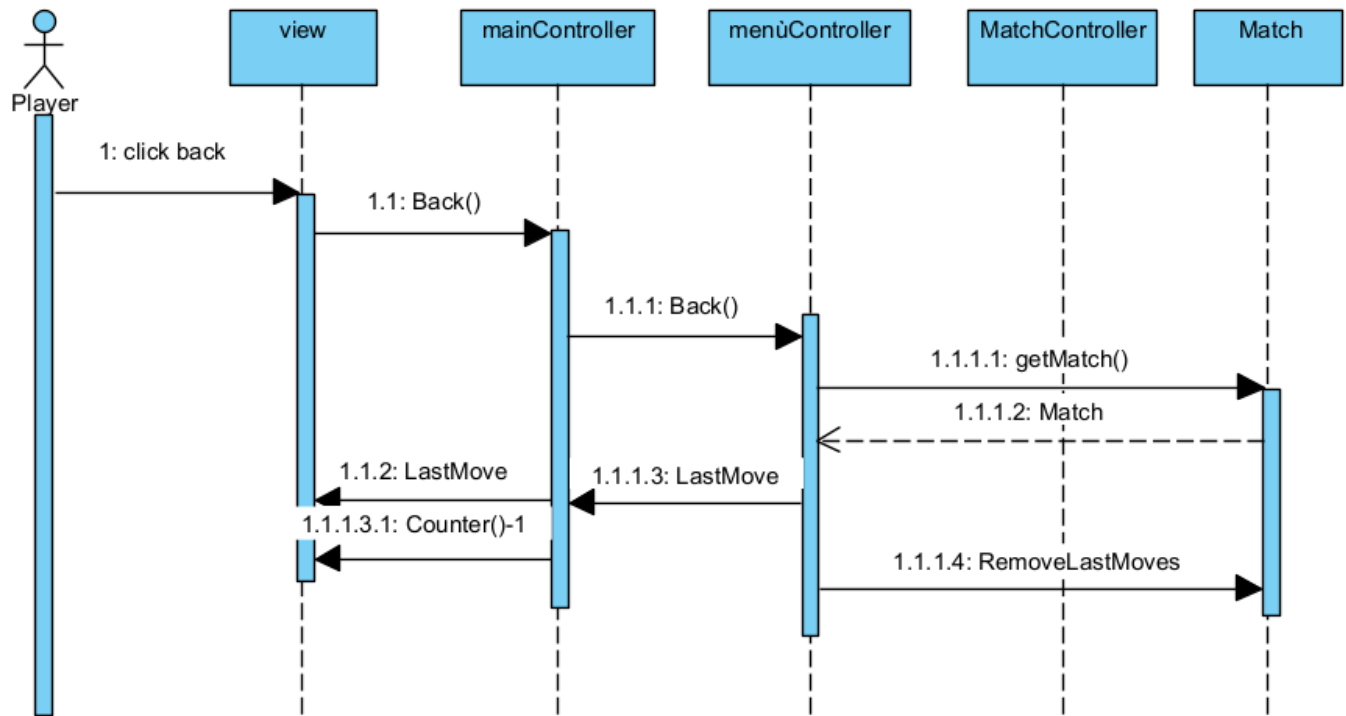5. Reset



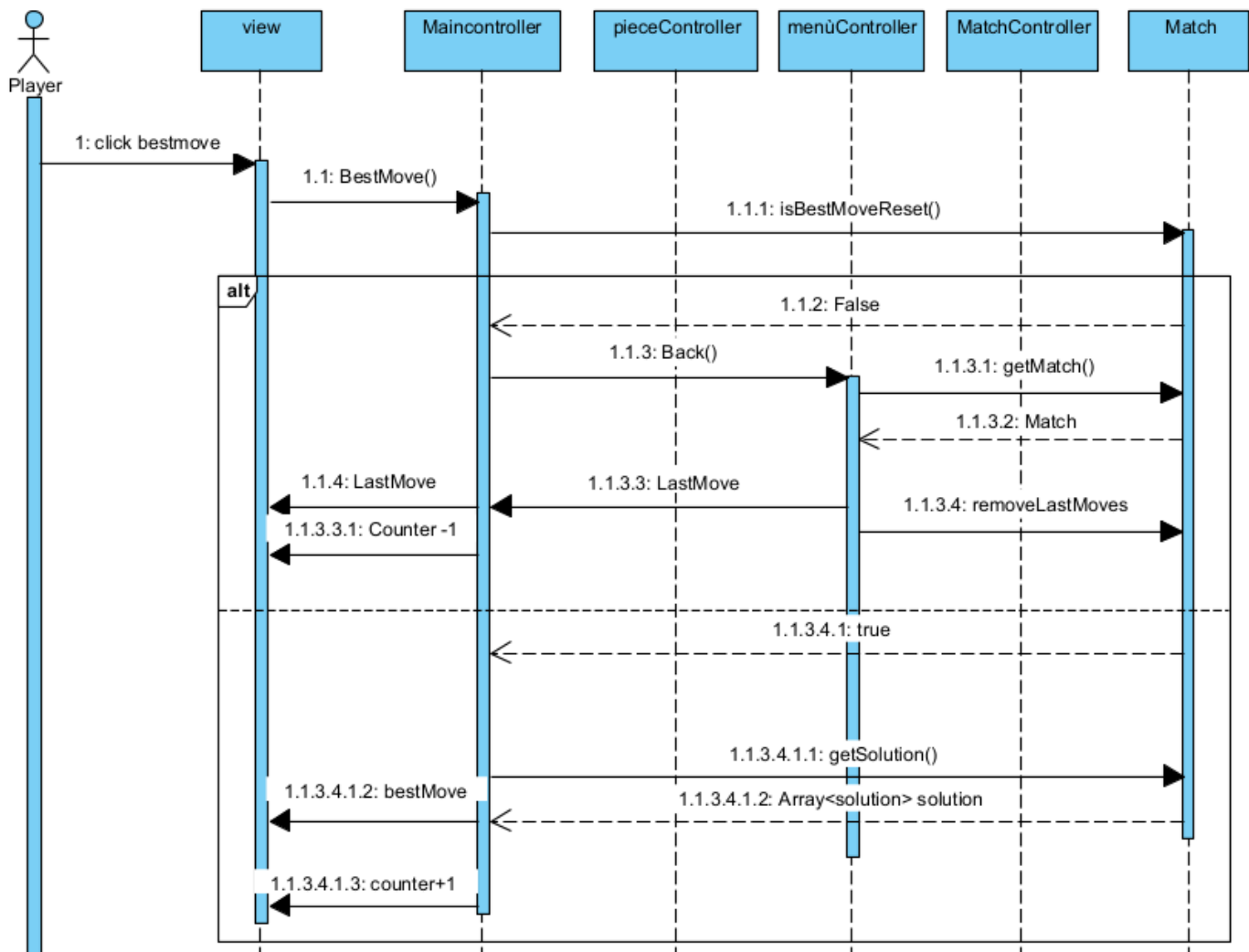6. Load

7. Save



8. Quit
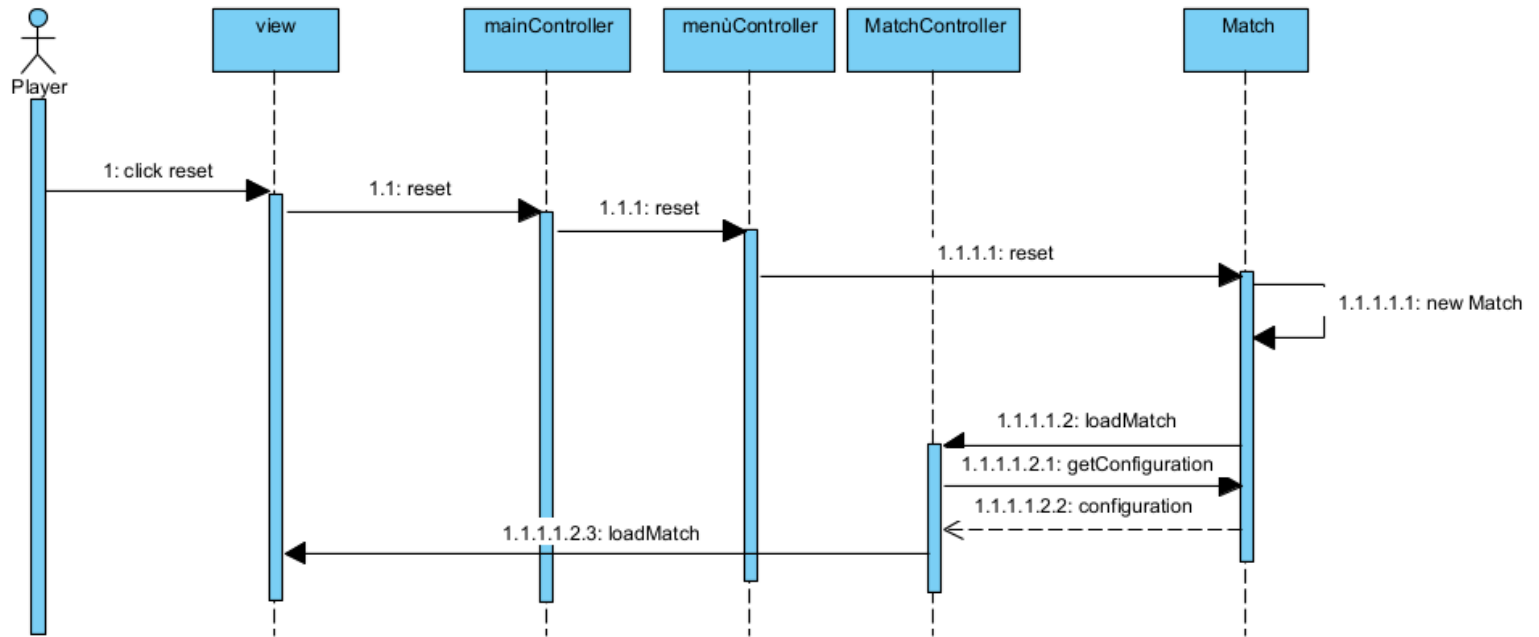
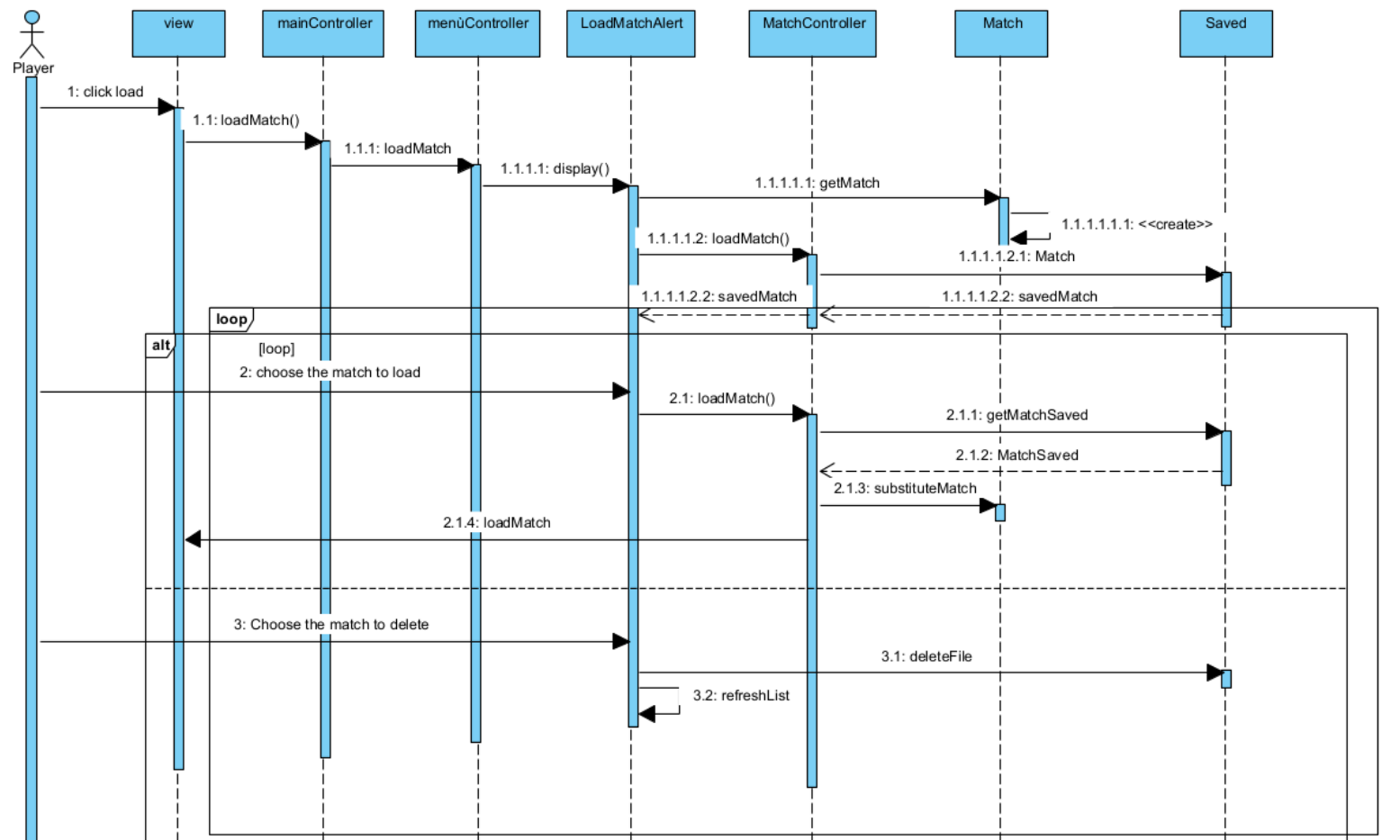# Internal sequence diagram:
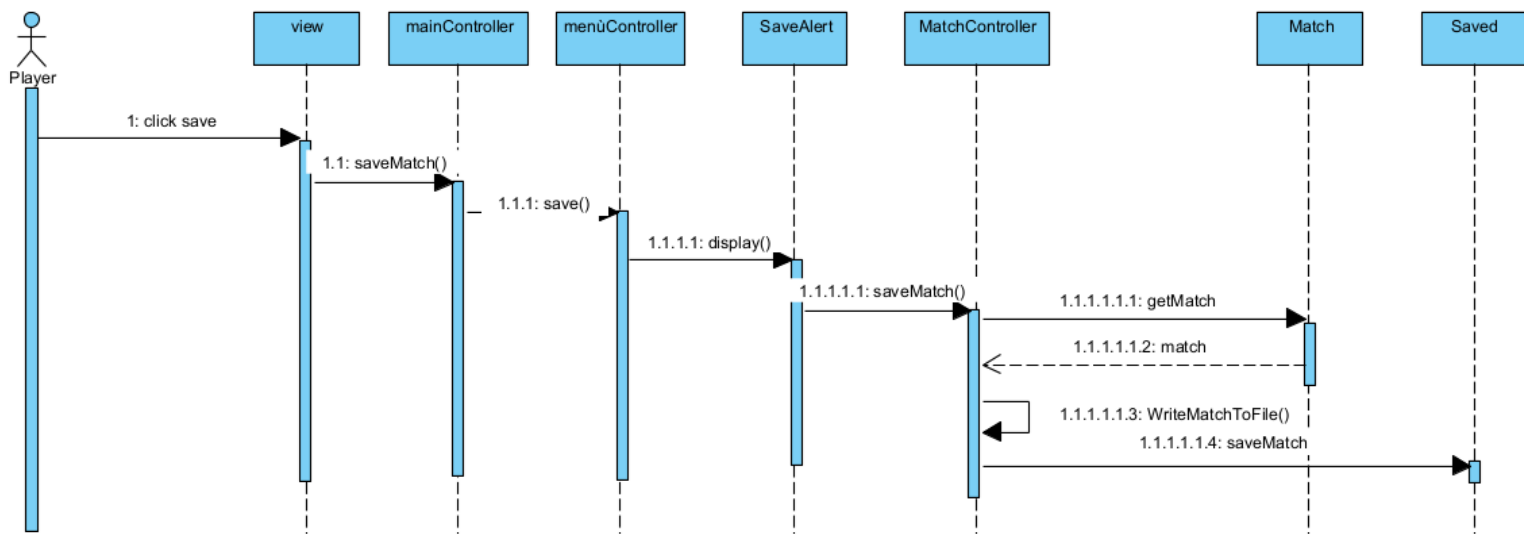
## 1. Start



## 2. Move pieces

# 3. Back



# 4. Best move

# 5. Reset



# 7. Load

# 6. Save



# 8. Quit