

Refactoring von GameController und BlackJackGame (Commit Hash = 72d4902b)

Es wurden die Klassen GameController und BlackJackGame refactored und überarbeitet. Diese Klassen wurden aufgrund der Folgenden Code Smells ausgewählt und überarbeitet:

Code-Duplizierung: Es gab mehrere Stellen im Programmcode, wo duplizierter Code geschrieben und verwendet wurde.

Lange Klasse: Die Klasse BlackJackGame war vor dem Refacotring 299 Zeilen lang. Die Klasse GameController war vor dem Refactoring 448 Zeilen lang. Diese beiden Klassen waren zu lang und unübersichtlich.

Klassenzweck: Die Klasse BlackJackGame hat sehr viele nicht zusammenhängende Aufgaben übernommen. Die Klasse GameController hat ihre Aufgabe als Controller des kompletten Spieles nicht richtig erfüllt.

Die Klasse BlackJackGame wurde im Zuge des Refactorings in mehrere Klassen aufgeteilt. Es wurden die Klassen EvaluationHandler, MoneyHandler und CardHandler erstellt. Diese Klassen haben einzelne dedizierte Zwecke und erledigen nur diese Zwecke. EvaluationHandler dient nur dem Evaluieren des Status des Spieles- Der CardHandler dient nur dem Ausführen von Aktionen mit Karten und dem Kartendeck. Der MoneyHandler dient dem Verändern des Geldstatus des Spielers. Im Zuge des Refactorings wurden alle Methoden und Aufgaben der Klasse BlackJackGame in andere Klassen umgezogen, weshalb die Klasse BlackJackGame entfernt wurde.

Die Klasse GameController dient nun als Hauptcontroller des Spieles und delegiert den Spielablauf. Innere Klassen aus GameController wurden in eigene Dateien und normale Klassen ausgezogen. Diese inneren Klassen wurden weitergehend auch Refactored und angepasst um Funktionen aus BlackJackGame, welche in den inneren Klassen besser aufgehoben sind, zu übernehmen.

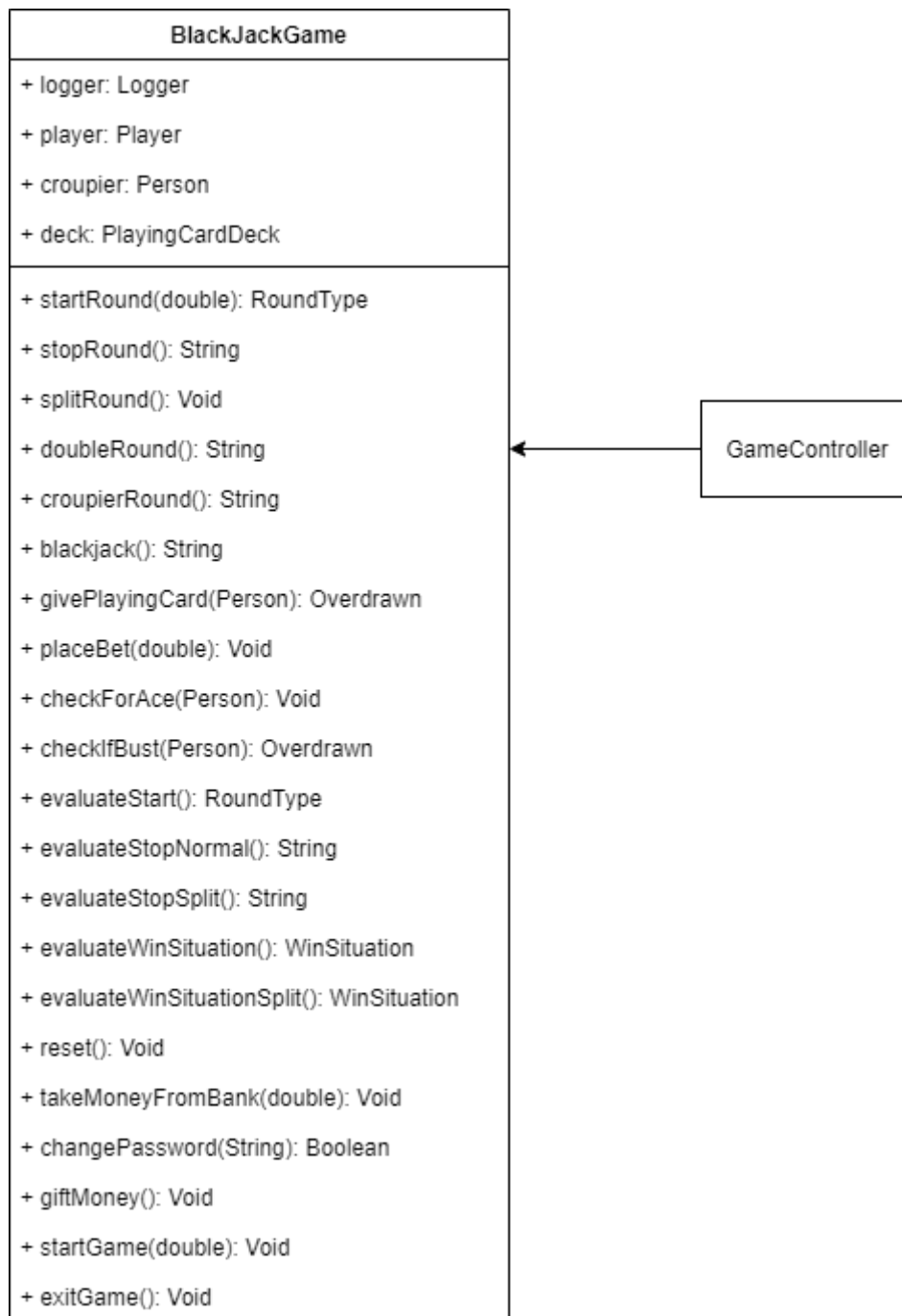


Abbildung 1,UML BlackJackGame vor dem Refactoring

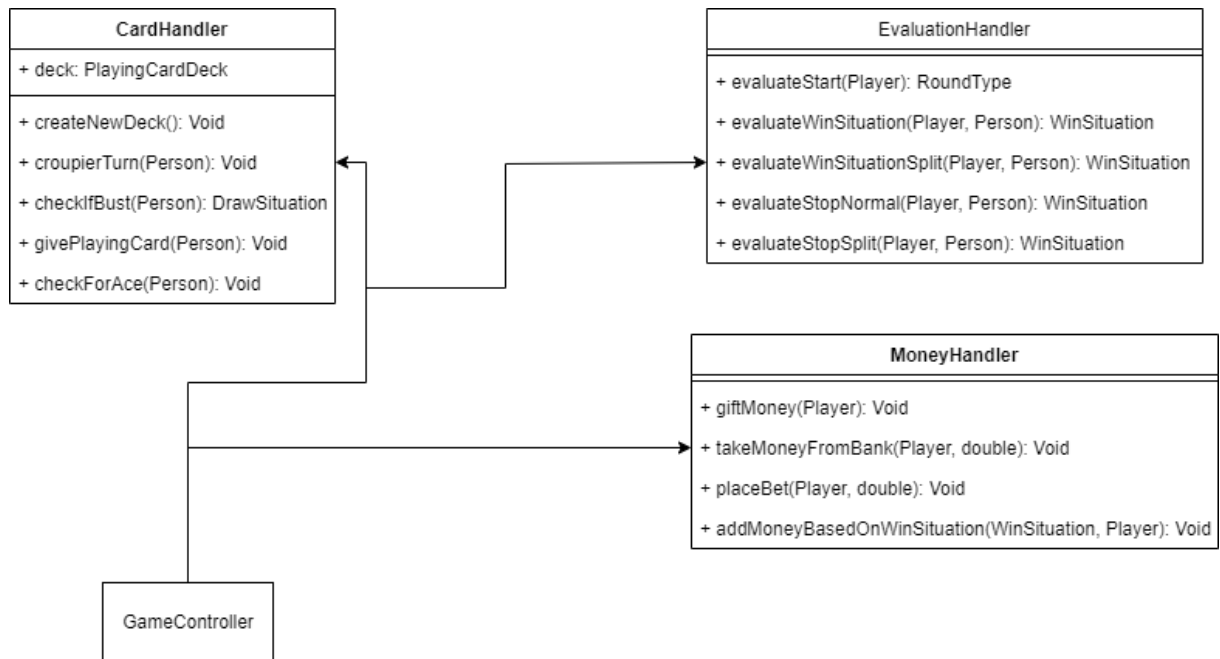


Abbildung 2, UML BlackJackGame nach dem Refactoring

GameController
+ logger: Logger + game: BlackJackGame + gameView: GameGUI + croupier: Person + player: Player + stateS: Boolean + stateD: Boolean + blackjack: Boolean
+ showView(): Void + addListener(): Void + placeCroupierCards(): Int + switchButtonsToStartMode(): Void
StartActionListener
+ actionPerformed(ActionEvent): Void
NewCardActionListener
+ actionPerformed(ActionEvent): Void
StopActionListener
+ actionPerformed(ActionEvent): Void
SplitActionListener
+ actionPerformed(ActionEvent): Void
DoubleActionListener
+ actionPerformed(ActionEvent): Void
TakeMonexFromBankActionListener
+ actionPerformed(ActionEvent): Void
ChangePasswordActionListener
+ actionPerformed(ActionEvent): Void
AddMoneyActionListener
+ actionPerformed(ActionEvent): Void
ProfileGameStartActionListener
+ actionPerformed(ActionEvent): Void
MenuExitListener
+ actionPerformed(ActionEvent): Void
MenuProfileListener
+ actionPerformed(ActionEvent): Void
MenuHelpListener
+ actionPerformed(ActionEvent): Void
MenuGameListener
+ actionPerformed(ActionEvent): Void
DefaultFocusListener
+ focusGained(FocusEvent): Void + focusLost(FocusEvent): Void

Abbildung 3, UML GameController vor dem Refactoring

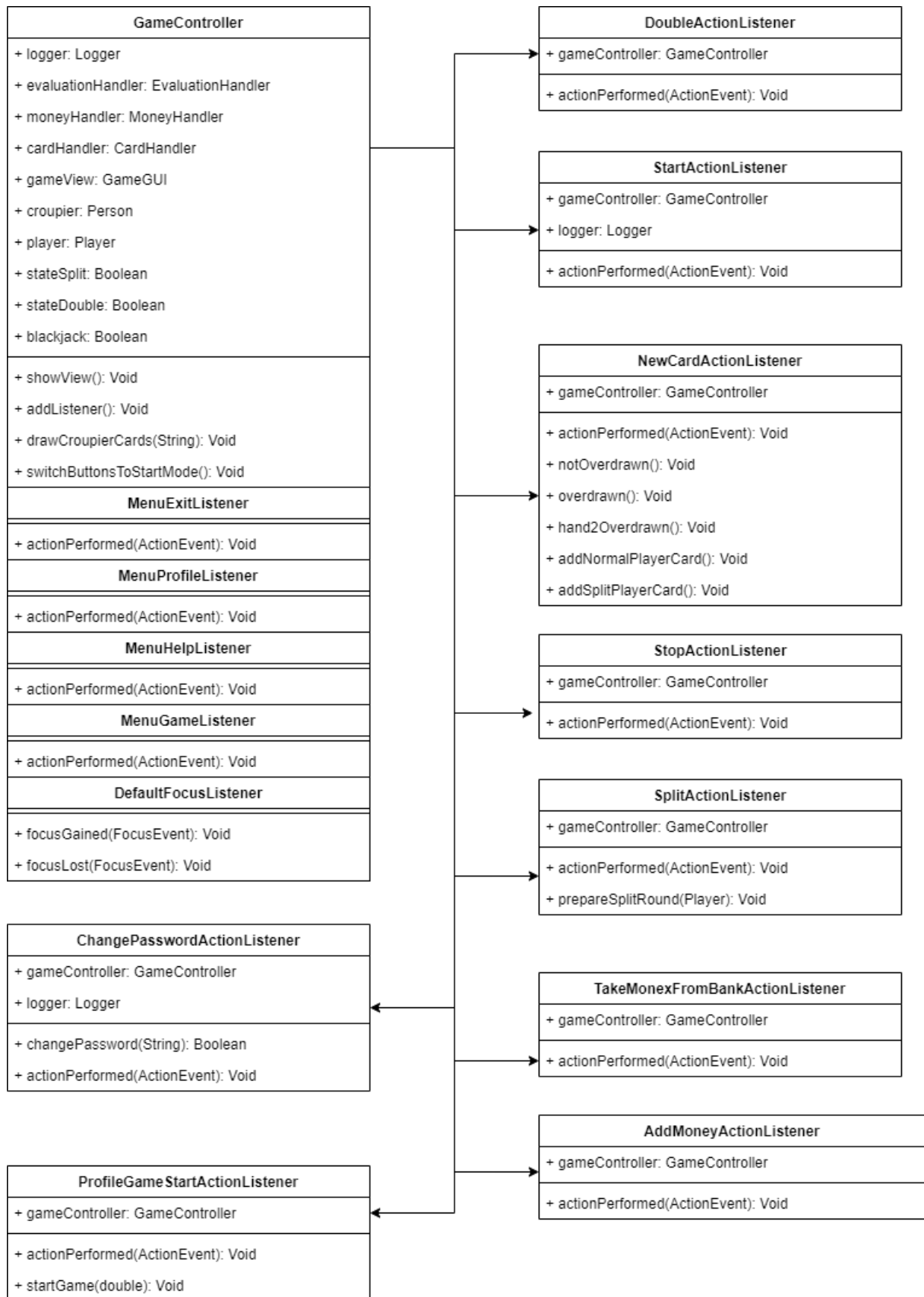


Abbildung 4, UML GameController nach dem Refactoring