

PlayerController.java

```

94     public boolean payDebt(Player debtor, Player creditor, Decorator decorator, String[] msg,
95     int debt) {
96         boolean debtPayed = false;
97         while (!debtPayed){
98             decorator.showMessage(msg);
99             try {
100                 debtor.getAccount().withdraw(debt);
101                 if (creditor != null) {
102                     creditor.getAccount().deposit(debt); //Should be handled in a seperate
103                     statement - future
104                 }
105                 debtPayed = true;
106             } catch (InsufficientFundsException e) {
107                 //insufficient funds to pay player
108                 System.out.println("Insufficient funds for transaction");
109                 //If player has any unpawned fields, he is forced to pawn them
110                 if(BoardController.hasAnyUnPawnedFields(debtor)){
111                     String[] msg1 = new String[]{"YouMustAtLeastPawnAllFields"};
112                     decorator.showMessage(msg1);
113                     handleInsufficientFunds(debtor, debt, decorator);
114                 } else {
115                     //Else he can choose between trying to raise money
116                     String[] messageString = new String[] {"TradeOrGoBroke"};
117                     String[] buttons = new String[] {"Trade", "Bankrupt"};
118                     int selection = decorator.getUserButtonPressed(messageString, buttons);
119                     if (selection == 0){
120                         handleInsufficientFunds(debtor, debt, decorator);
121                     } else {
122                         hostileTakeOver(creditor, debtor);
123                         break;
124                     }
125                 }
126             }
127         }
128     } catch (IllegalAmountException e) {
129         System.err.println("IllegalAmount LandOnOwnable");
130         e.printStackTrace();
131         break;
132     }
133     return debtPayed;
134 }

```