NEW

DTU

Matador

DTU

2014 Edition

Group 51

Martin Nielsen s123064 - Rúni Egholm Vørmadal 134004 - Christian Budtz s134000

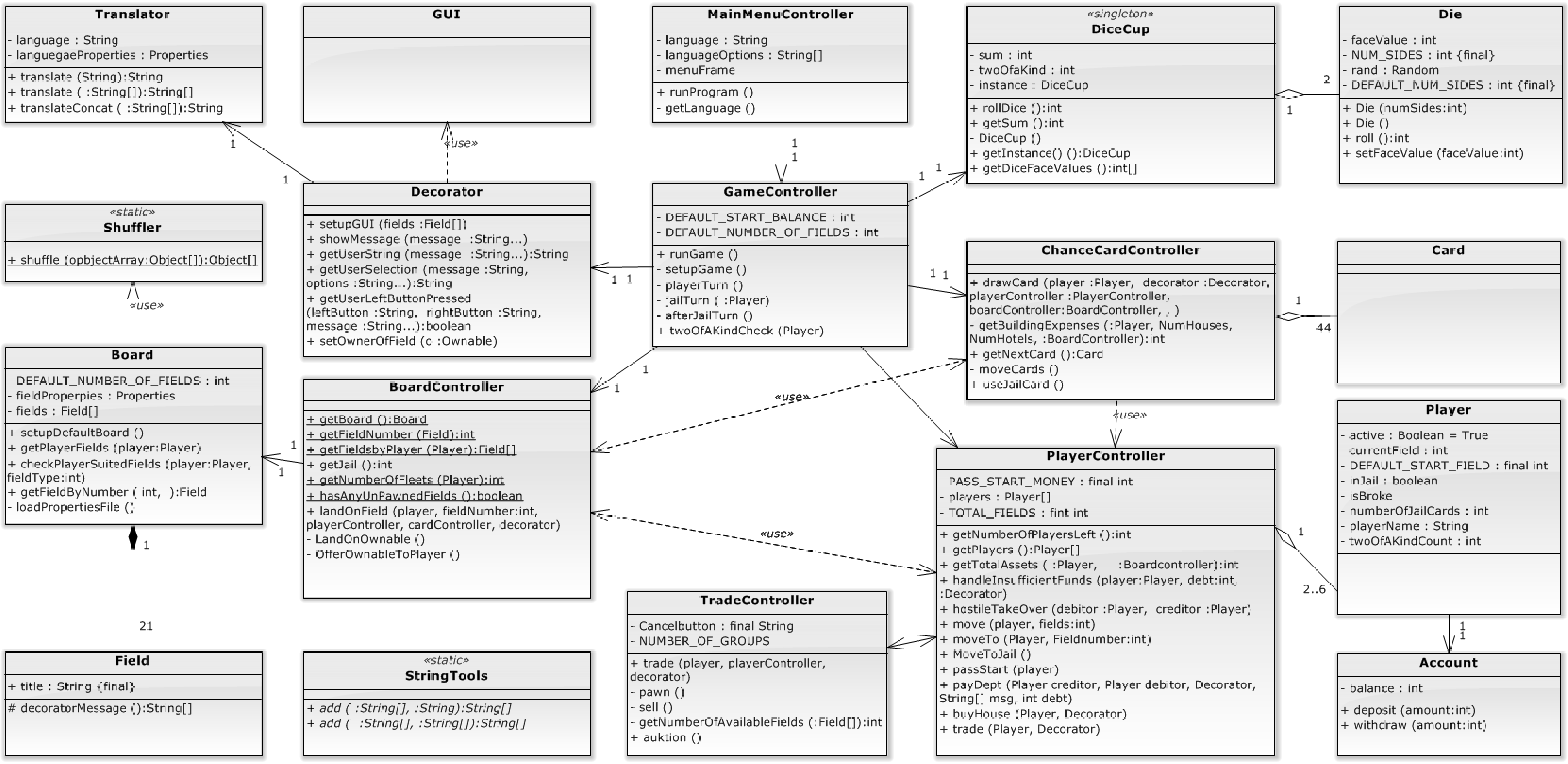
Overview

- Advanced handling of missing payment using exceptions
- Fully implemented Buying/Selling and Pawning of deeds
- Fully implemented Buying/Selling of Houses
- 44 chance cards
- Thematical soundtrack
- 3 fully integrated languages

Coded with:

- Focus on keeping high supportability
- Strict adherence to the GRASP and BCE patterns
- Close adherence to the original board game

An Area-51 production



Translator
- language : String
- languegaeProperties : Properties
+ translate (String):String
+ translate (:String[]):String[]
+ translateConcat (:String[]):String

GUI

MainMenuController
- language : String
- languageOptions : String[]
- menuFrame
+ runProgram ()
- getLanguage ()

«singleton»
DiceCup
- sum : int
- twoOfaKind : int
- instance : DiceCup
+ rollDice ():int
+ getSum ():int
- DiceCup ()
+ getInstance() ():DiceCup
+ getDiceFaceValues ():int[]

Die
- faceValue : int
- NUM_SIDES : int {final}
- rand : Random
- DEFAULT_NUM_SIDES : int {final}
+ Die (numSides:int)
+ Die ()
+ roll ():int
+ setFaceValue (faceValue:int)

«static»
Shuffler
+ shuffle (opbjectArray:Object[]):Object[]

Decorator
+ setupGUI (fields :Field[])
+ showMessage (message :String...)
+ getUserString (message :String...):String
+ getUserSelection (message :String, options :String...):String
+ getUserLeftButtonPressed (leftButton :String, rightButton :String, message :String...):boolean
+ setOwnerOfField (o :Ownable)

GameController
- DEFAULT_START_BALANCE : int
- DEFAULT_NUMBER_OF_FIELDS : int
+ runGame ()
- setupGame ()
- playerTurn ()
- jailTurn (:Player)
- afterJailTurn ()
+ twoOfAKindCheck (Player)

ChanceCardController
+ drawCard (player :Player, decorator :Decorator, playerController :PlayerController, boardController:BoardController, ,)
- getBuildingExpenses (:Player, NumHouses, NumHotels, :BoardController):int
+ getNextCard ():Card
- moveCards ()
+ useJailCard ()

Card

Board
- DEFAULT_NUMBER_OF_FIELDS : int
- fieldProperpies : Properties
- fields : Field[]
+ setupDefaultBoard ()
+ getPlayerFields (player:Player)
+ checkPlayerSuitedFields (player:Player, fieldType:int)
+ getFieldByNumber (int,):Field
- loadPropertiesFile ()

BoardController
+ getBoard ():Board
+ getFieldNumber (Field):int
+ getFieldsbyPlayer (Player):Field[]
+ getJail ():int
+ getNumberOfFleets (Player):int
+ hasAnyUnPawnedFields ():boolean
+ landOnField (player, fieldNumber:int, playerController, cardController, decorator)
- LandOnOwnable ()
- OfferOwnableToPlayer ()

Player
- active : Boolean = True
- currentField : int
- DEFAULT_START_FIELD : final int
- inJail : boolean
- isBroke
- numberOfJailCards : int
- playerName : String
- twoOfAKindCount : int

PlayerController
- PASS_START_MONEY : final int
- players : Player[]
- TOTAL_FIELDS : fint int
+ getNumberOfPlayersLeft ():int
+ getPlayers ():Player[]
+ getTotalAssets (:Player, :Boardcontroller):int
+ handleInsufficientFunds (player:Player, debt:int, :Decorator)
+ hostileTakeOver (debitor :Player, creditor :Player)
+ move (player, fields:int)
+ moveTo (Player, Fieldnumber:int)
+ MoveToJail ()
+ passStart (player)
+ payDept (Player creditor, Player debitor, Decorator, String[] msg, int debt)
+ buyHouse (Player, Decorator)
+ trade (Player, Decorator)

TradeController
- Cancelbutton : final String
- NUMBER_OF_GROUPS
+ trade (player, playerController, decorator)
- pawn ()
- sell ()
- getNumberOfAvailableFields (:Field[]):int
+ auktion ()

Field
+ title : String {final}
# decoratorMessage ():String[]

«static»
StringTools
+ add (:String[], :String):String[]
+ add (:String[], :String[]):String[]

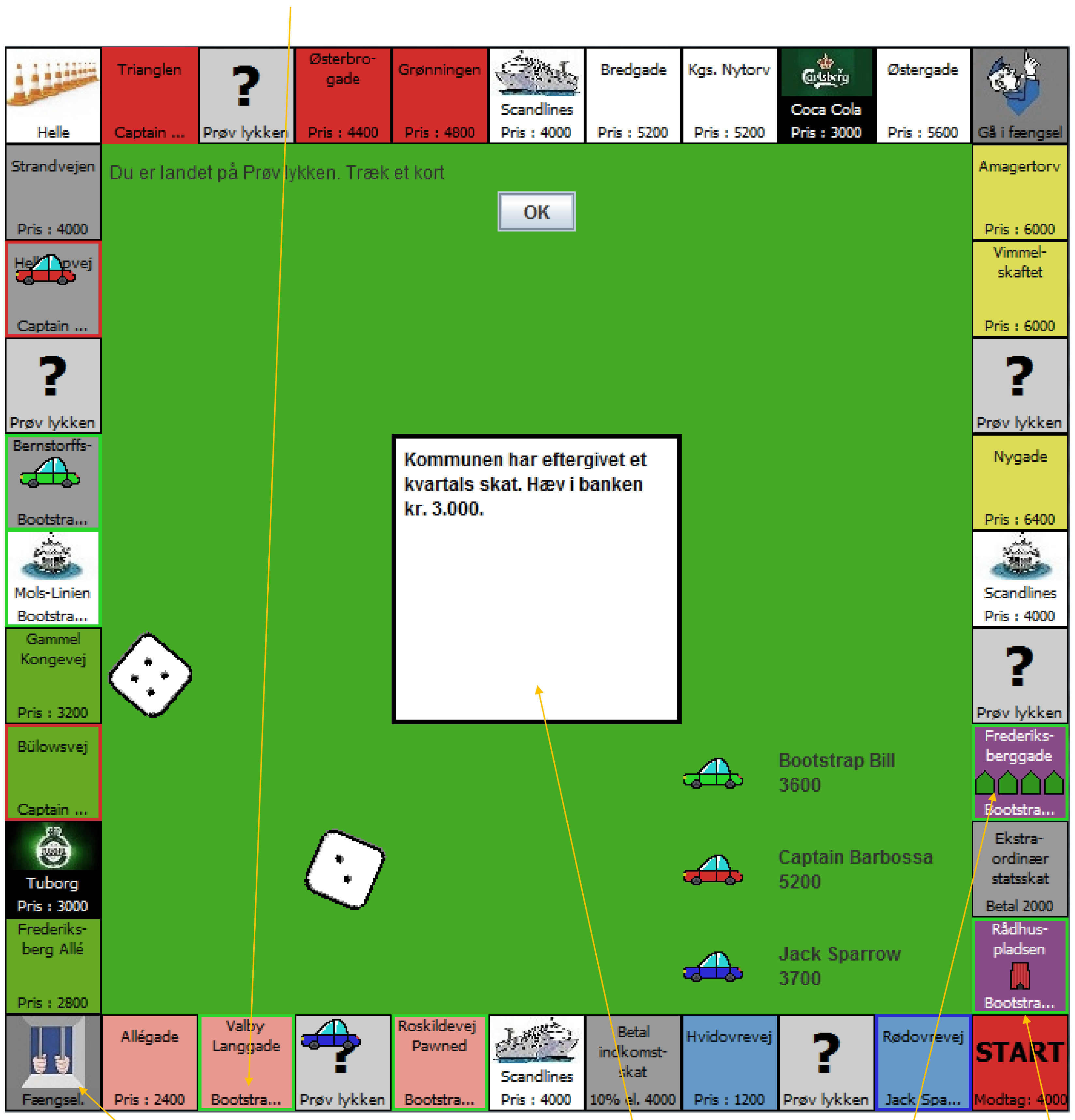
Account
- balance : int
+ deposit (amount:int)
+ withdraw (amount:int)

Design

Strong adherence to BCE
Separated entities
GRASP:
High cohesion
Low coupling
Polymorphic code
Information expert
Controller pattern
Creator pattern
Indirection

Singleton pattern
Static helper class and methods

Name of Owner



Du er landet på Prøv lykken. Træk et kort

OK

Kommunen har eftergivet et kvartals skat. Hæv i banken kr. 3.000.

Bootstrap Bill 3600

Captain Barbossa 5200

Jack Sparrow 3700

Fully implemented jail

Chance Card

Houses

Hotel

Implementering: payDebt()

```
PlayerController.java
public boolean payDebt(Player debitor, Player creditor, Decorator decorator,
           String[] msg, int debt) {
  boolean debtPayed = false;
  while (!debtPayed){
    decorator.showMessage(msg);
    try {
      debitor.getAccount().withdraw(debt);
      if (creditor != null) {
        creditor.getAccount().deposit(debt);
      }
      debtPayed = true;
    } catch (InsufficientFundsException e) {
      //insufficient funds to pay player
      System.out.println("Insufficient funds for transaction");
      //If player has any unpawned fields, he is forced to pawn them
      if(BoardController.hasAnyUnPawnedFields(debitor)){
      String[] msg1 = new String[]{"YouMustAtLeastPawnAllFields"};
      decorator.showMessage(msg1);
      handleInsufficientFunds(debitor, debt, decorator);
      } else {
        //Else he can choose between trying to raise money
        String[] messageString = new String[] {"TradeOrGoBroke"};
        String[] buttons = new String[] {"Trade","Bankrupt"};
        int selection = decorator.getUserButtonPressed(messageString,
           buttons);
        if (selection == 0){
          handleInsufficientFunds(debitor, debt, decorator);
        } else {
          hostileTakeOver(creditor, debitor);
          break;
        }
      }
    } catch (IllegalAmountException e) {
      System.err.println("IllegalAmount LandOnOwnable");
      e.printStackTrace();
      break;
    }
  }
  return debtPayed;
}
```