CMPT 399 - Monopoly

Cody Moorhouse
Patryk Kaczmarkiewicz

Solo Classes

Monopoly Package:

1. GameMaster
2. Die
3. Player
4. TradeDeal

GUI Package:

1. InfoFormatter
2. Main
3. GameBoardUtil

Parent Classes

Monopoly Package:

1. Card
2. TradeDialog
3. RespondDialong
4. Cell
5. GameBoard
6. MonopolyGui

GUI Package:

1. JDialog
2. JPanel
3. CellInfoFormatter
4. JFrame

Implementing Classes

Monopoly Package:

- n/a

GUI Package:

1. MainWindow (MonopolyGui)
2. GUIRespondDialog (TradeDIalog)
3. GUITradeDialog (TradeDIalog)

Design

- Classes extend a parent class and are prefixed with the parents’ name
- Tests are combined within the package contents
- Using old java utilities that have been deprecated

The game implements MVC pattern:

Model:	GameBoard
View:	GUI Package
Controller:	GameMaster

Card Class

Deals with the Cards from the Monopoly Game

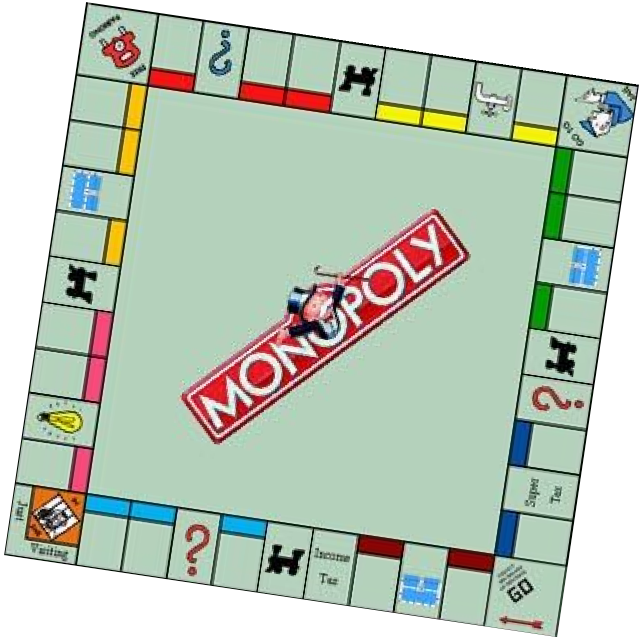
- JailCard
  - “Go to Jail” Card
- MoneyCard
  - “Give a player money” Card
- MovePlayerCard
  - “Move a player” Card

Chance
GO DIRECTLY TO JAIL
DO NOT PASS GO, D
© 1936 PARKE
Community Chest
YOU INHERIT $100
© 1936 P
Chance
ADVANCE TO GO
GO
(COLLECT $200)

Cell Class

Represents the different Cells on the Monopoly Board that a Player can land on

- UtilityCell
- CardCell
- RailRoadCell
- PropertyCell
- JailCell
- FreeParkingCell
- GoCell
- GoToJailCell
- PropertyCell



GameBoard Class

The model for the Monopoly game that sets up the board to be used

- GameBoard14
- GameBoardFreeParking
- SimpleGameBoard
- GameBoardCCLoseMoney
- GameBoardCCGainMoney
- GameBoardCCJail
- GameBoardCCMovePlayer
- GameBoardUtil
- GameBoardRailRoad
- GameBoardJail
- GameBoardFull

Bad Smells

- GameMaster class is doing too much
  - Deals with dice logic, player movements, trades, gui logic
  - Some functions have multiple definitions (e.g. movePlayer is defined twice)
- Long methods
- Inconsistent / poor naming conventions
- Old libraries are used
- Unused variable declarations
- Duplicated Code
- Message Chains (this.that.something.next(...))
- Lack of documentation

Conclusion

1. Start with fixing the test packages
2. Remove all the warnings and convert to current Java libraries
3. Simplify the GameMaster Class
4. See where we go from there

See you next presentation!